3D-Equalizer Version 3 Release 5 Manual

© 2005 Science.D.Visions
Heiliger Weg 60
D-44135 Dortmund
Germany

Email: info@sci-d-vis.com
Web: www.sci-d-vis.com

Acknowledgements

We would like to thank Alias|Wavefront, Softimage, Side Effects, NewTek, Kinetix, Voss TV and SAR for their support in the development of the export modules. Thanks also to the beta-testers of 3D-Equalizer for useful discussions.

We would also like to thank Trix (Belgium), Silicon Studio LA (USA), FrameStore (UK), Video Efecto (Spain) and CA Scanline (Germany) for providing tutorial sequences and additional image material.

Very special thanks to Matthias, Denis, the Andys, Dave, Mario, Seamus, all the fine people at VPJ, Alan, Hugh, Ron, Charlie, Jeff, Petra & Remo, Wayne, Sam, Lee, Michael, Daniel, and Albrecht.

February, 2005

Although every attempt is made to ensure the correct working of the application 3D-Equalizer and the correctness of the information contained in this manual some errors may exist. Science.D.Visions will not accept responsibility of any kind for damages or other consequences resulting from the use of 3D-Equalizer or this manual.

Contents

Overview

The present manual of 3D-Equalizer V3 consists of five different sections:

1. Introduction. The introduction contains the definitions of the most important terms which occur in the context 3DE and describes the situations in which 3DE is commonly used.
2. Tutorials. Below you find a brief description of their content.
3. Selected topics. In these sections you will find background information about some of 3DE's features, which are not explained in the tutorials, in order not to disturb the didactical concept of the latter.
4. Reference Manual. This chapter gives you information about all of 3DE's user interaction components, listed window by window and menu by menu.
5. Appendix. The appendix describes, how to use 3DE's Tcl-interface, gives an overview of the math library (in Tcl) which is delivered as part of the software package.

Tutorials

There are 12 tutorial sections. Probably you do not want to run through each of them, however, some of them deserve your interest before you start doing your own projects, since they present many functions, the knowledge of which will make it easier for you to use 3DE. The first one is quite comprehensive and touches many of the important functions of 3DE. In any case we recommend you to start with this tutorial before you do your own projects. Tutorial 2 demonstrates the use of 3DEqualizer's automatic tracking engine. Tutorials 3 and 4 are also important, because they describe techniques which you will encounter in every project. Tutorial 5 illuminates a typical effect of set arrangements which do not seem to contain enough depth information for a proper computation, and explains how this effect can be removed ("stabilizing"). The next major tutorial is probably tutorial 6, which describes the so-called "Fixed Camera Position" mode, a calculation method which has turned out to be of major interest in practice. Another important feature, namely the so-called reference frames are subject of tutorial 7. Tutorials 8 and 9 deal with features of V3 which were not available in V2, namely multiple pointgroups and multiple sequences. Tutorial 10 explains how to determine the so-called lens distortion, both in presence of grid shots and without grid shots. The methods for measuring and handling this kind of camera parameters have been enhanced considerably in contrast to V2. Tutorial 11 is dedicated to the treatment of zoomed shots. Finally, the last tutorial demonstrates how to setup a basic motion capturing project.

Introduction

In recent years, the process we call “Matchmoving” (also referred to as “3-D Camera Tracking”) has come to play an increasingly crucial role at the foremost edge of the digital graphics world. Matchmoving is made possible by analytical processes applied to original motion picture sequences, by which we define and recreate authentic 3D motion data for all secondary visual elements.

Our matchmoving system, 3D-Equalizer (“3DE”), is an elegant, powerful, easy-to-operate solution for synthetically reconstructing critical motion and structural data from practically any existing quantifiable image elements. Exploring all the possibilities of your 3DE postproduction system is guaranteed to trigger a widening range of exciting creative opportunities, and also to maximize workflow efficiency.

3DE, now in its 3rd generation, is a stand-alone tool which automatically resolves the extensive challenges and specific problems of matchmoving, based on its ingenious scientific calculation core and supported by an intelligent, highly adaptive user interface.

As confirmed by our system users, the industry’s foremost image creators, 3DE V3 is designed and tested from the ground up to be the most powerful and efficient matchmoving tool you will find anywhere.

Most matchmoving scenarios involve a combination, or more precisely, a composition of real elements and virtual elements based on time-dependent spatial relationships between your cameras and the objects being recorded by these cameras.

3DE is your principal ally throughout this mathematically critical process. 3DE automates complex time-space-motion computation, then seamlessly transfers all the motion data essential to further processing stages for your modeling, animation and compositing modules via specially-created export filters. This manual describes in detail all of 3DE’s V3 stand-alone functions, controls and characteristics as well as 3DE system workflow in relation to other applications. Please contact us with any questions or suggestions regarding further developments.

Matchmoving is increasingly used in the fields of commercial, industrial and feature film production, as well as for visualization in science and architecture, in video reenactments and in virtual studios. The basic situations of 3DE usage are categorized as follows:

a) Virtual (or “CG”) objects placed in a real visual environment. In architectural previsualization, CG is often used to integrate partial or entire structural illustrations into real-life photographs or video sequences. Using related methods, CG prosthetics and “virtual make-up” can be applied to alter the features of actors or artifacts in a motion picture scene.

b) Real objects placed in a virtual (or CG) environment. An example of this would be a real actor moving among computer-generated scenery. In TV studio situations, presentations often involve “virtual sets” in a process normally requiring lots of expensive hardware to accomplish - however not if it can be done in postproduction, because now the entire motion reconstruction process can be worked out quickly using 3D-Equalizer.

c) Real objects placed within real environments. Hybrid images combining different real elements do not work convincingly if scales and perspectives from all the various elements are too divergent. Remember “The Incredible Shrinking Man”? You won’t have to, when you’re using 3DE! Your real environment is recorded in the conventional way, then all motion information is reconstructed in 3DE and the data fed into a motion control rig which shoots other real objects to be composited into the primary real environment.

d) Stabilizing creative optical illusions. In “frozen moment” shots (also variously known as “bullet time” or “slice of life”) which are based on a geometric combination of simultaneously triggered still cameras, 3DE is used for calculating the exact position and orientation of each individual camera and a point-based geometric model reconstruction of the set. Your motion data and the point model exported from 3DE define and compensate for all and any random fluctuations of position and orientation of the group of cameras, so that your shot will then be flawlessly composited.

Conventions

3DE’s graphical user interface is designed in accordance with the SGI Indigo Magic Style Guide. All 3DE console colours mentioned in this manual are selections from the Indigo Magic desktop colour scheme. When you wish to enter something in a text field, just place the mouse pointer over the field and enter your text. Removing the mouse pointer from the text field then confirms the entry.

Terminology

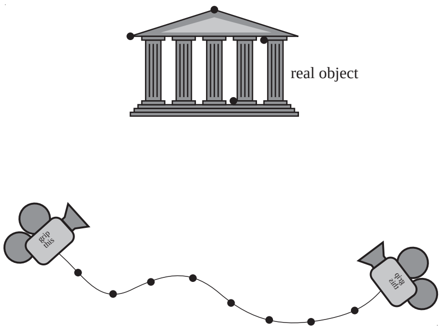
Within 3DE and this manual certain specific expressions are used, most of which will be familiar to everyone experienced in modelling or compositing. In the following section we will review some of these expressions.

Basic geometric expressions

When a scene is recorded by a camera, for each moment in time camera characteristics are defined by *position* and *orientation*. When we fix coordinates in space, we can represent this position by three values (x,y,z). The orientation of the camera, essentially the direction it looks in and the rotation around this view axis, can be represented in many different ways, most familiar of which the reader will be with angular representation (rx,ry,rz). The explicit choice of this representation is not so important here. In a similar sense we can characterize a recorded object by its position and orientation, provided we associate an origin and some coordinate axis to this object. Let us, however, consider only the case of a moving camera in the following.

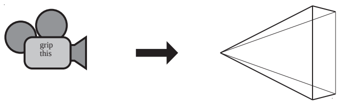
Cameras and Images

Recording something with a film or video camera is a process which evolves in time, it has a beginning and an end. Hence, it is clear, what we understand by a *camera motion path* of a sequence: it is the set of all camera positions/orientations for a given shot. We can easily visualize such a camera motion path.



Camera motion path

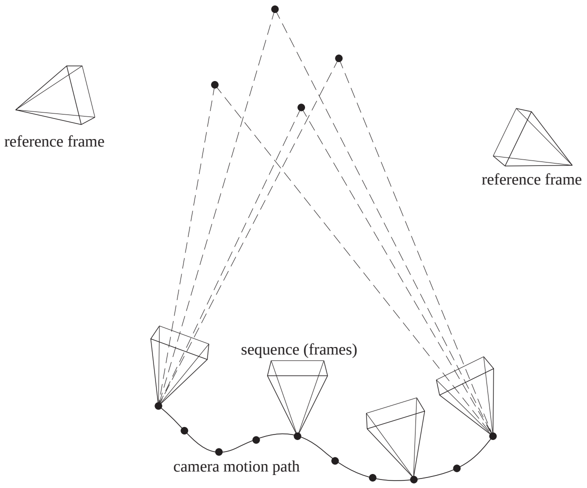
For simplicity, we will represent a camera by a pyramid, which is also the representation within our software (e.g. Orientation Window) and also within many animation packages. This corresponds to a very simple pinhole camera model. Until we begin to deal with non-linear effects like lens distortion, this model is sufficient for our purposes.



Camera pyramid

The geometric data of a camera relevant to 3DE (except for lens distortion) are contained in the shape of this pyramid: the focal length, filmback width and filmback height. Another important parameter is the pixel aspect, but this is more associated with the digitizing process than to the camera optics itself. The camera geometry and digitizing process is explained in the section “Camera Adjustment”, chapter “Selected Topics”. The precise definition of our lens distortion model is explained in the appendix (see also tutorial “Lens Distortion”).

Let us assume we want to attach some CG-content to the scenery so as to get the correct perspective throughout the shot. The CG-content is rendered by a virtual camera, as you already know from your rendering software. If it is defined correctly according to the original footage, that is, if the virtual camera performs exactly the same movement as the real camera, the perspective of CG objects and real objects will always be correct. Therefore the real camera movement must be known or calculated from the footage, and this is what 3DE does.



Sequence and reference frames

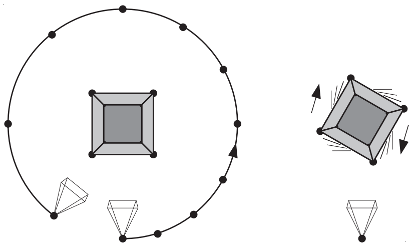
The basic input for 3DE consists of image files, the so-called *frames*. These frames can be divided into two classes, namely *sequence frames* and *reference frames*. A (real) *sequence* is a set of frames, recorded with a camera, which fulfill the following condition: the image content varies more or less continuously, which reflects the continuity of time. In other words, neither the camera positions and orientations nor the position and orientation of the recorded objects “jump” from one frame to the next frame. In contrast to a sequence, reference frames are usually single, isolated recordings of the real scenery or some real objects. They could e.g. consists of photographs. The task which 3DE performs is to calculate the motion path of a camera/object for a sequence. The purpose of reference frames is to provide additional information to 3DE. Usually, in reference frames, the image content does not exhibit any kind of “continuity” like a sequence would.

As mentioned, the frames represent the basic input for 3DE, but the 3d-calculations of 3DE are not based on evaluating the entire image, only some very restricted subsets of information. Again, let us consider the situation in which a camera is recording scenery. The user identifies this scenery with a *point group*. Such a point group consists of *points*, objects which essentially represent a 3d-position in space, but which are equipped with a lot of other properties in the context of 3DE. Usually the user designates striking points in the scenery which can easily be recognized in the images. These striking points happen to be in the scenery "naturally" or are placed intentionally (e.g. markers).

In the course of the sequence and also in the reference frames, the points are identified by the user within the frames. For sequences this process is referred to as *motion tracking*. The expression "Tracking" makes sense, because, as we mentioned above, the image content varies continuously. The result is a so-called *tracking curve*, a two-dimensional curve (e.g. expressed in pixel coordinates). In practice, a tracking curve often will not represent the motion of the point during the entire sequence. On the contrary, the curve will be truncated when a point in the real scenery is not yet (or no longer) visible, or covered by obstacles. However, for simplicity, we can assume here that the points are tracked during the entire sequence.

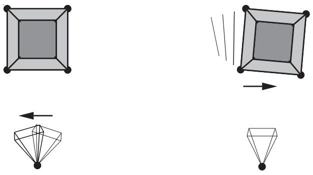
Camera point group and object point groups

When a camera is recording a single object, 3DE calculates the geometric relationship between camera and object. We will illustrate in the following how camera movement and object movement relate to each other. In order to do this, we consider a circle-shaped camera motion around a cube. For the calculation kernel of 3DE this situation is completely equivalent to a fixed camera and a rotating cube.



Circle-shaped camera motion vs. rotating object

On the other hand, imagine a camera fixed to a tripod, rotating around its vertical axis, and a fixed cube. This situation is equivalent to a totally fixed camera and a moving cube. It would also be possible to interpret both camera and cube as moving objects.



rotating camera vs. translating object

In practice, it may happen that both types of information are required: the absolute movement of the camera with respect to the scenery and the absolute movement of the object with respect to the scenery. As an example, imagine a street with a moving car. The camera as well as the car are moving with respect to the street. The task is to replace the real car by some CG-car moving on the street. You want to add realistic lighting (e.g. from the sun) and realistic shadow-casting on the street. So, it would be very useful to know the absolute movement of the car with respect to the street/scenery.

Object pointgroup: car - Camera pointgroup: street

In order to solve this type of problem, 3DE provides methods for calculating several point groups within one project. The two important expressions in this context are “camera group” and “object group”, which we define below.

In order to calculate the motion of the camera with respect to the street one would choose some tracking points on the street. These points form the so-called *camera group*. The car, however, is a moving object, and we want to determine its (absolute) movement with respect to the street. Again, one chooses a few tracking points on the car, and these points form a point group called the *object group*. There can be at most one camera group, but there could be in principle an arbitrary number of object point groups.

As you see, object groups and camera groups now clearly distinguish between moving cameras and moving objects. It would make no sense to talk about a street moving with respect to a fixed camera.

Deviation

When 3DE calculates a camera motion path (or object motion path) by evaluating the tracking curves, the positions of the points in three-dimensional space are also calculated. In a proper arrangement of camera positions and point positions the points in space lie on the rays from the camera position through the image plane as in figure “Sequence and reference frames”. In practice however, there will always be a slight discrepancy between the rays and the points. This discrepancy can be projected back to the image plane and is usually expressed in pixel. We call it the *deviation* in the context of 3DE and in this manual. So, the deviation is a number associated to a frame and a point group (or to a frame and a point). In 3DE’s Status Window you can examine the deviation for an entire sequence and a point group as a curve. The precise definition of the deviation is not so straightforward, and we shall omit the details here. It may be sufficient to say that it expresses the average difference between the “is”-position of the 3d-points projected onto the screen from their “should-be”-position given by the tracking curves of the respective points.

Calculation methods

Let us imagine a situation in which the motion of camera is to be calculated by 3DE. In the current version, 3DE distinguishes two situations: (1) the camera is translating and rotating in three-dimensional space, (2) the camera is rotating but its position is fixed, e.g. when the camera is fixed to a tripod. The attribute “fixed or not” only makes sense for a camera group, and it is associated to a sequence (Frames Window::Modify...). The method for processing this kind of sequence is called the *fixed camera position mode* (FCPM), and is enabled by activating the “fixed camera positition constraint”.

Workflow

In the figure below you see the typical workflow of a 3DE project for a simple project. In the tutorials the single steps are carried out in detail. As mentioned, the basic input when setting up a project consists of frames, namely reference frames and/or a sequence. It is possible to build a project that only consists of reference frames, e.g. if you want to use 3DE in the original sense of photogrammetry, i.e. surveying a three-dimensional object by means of recorded images from this object. However, in practice many projects contain only a sequence with, in some situations, reference frames being added.

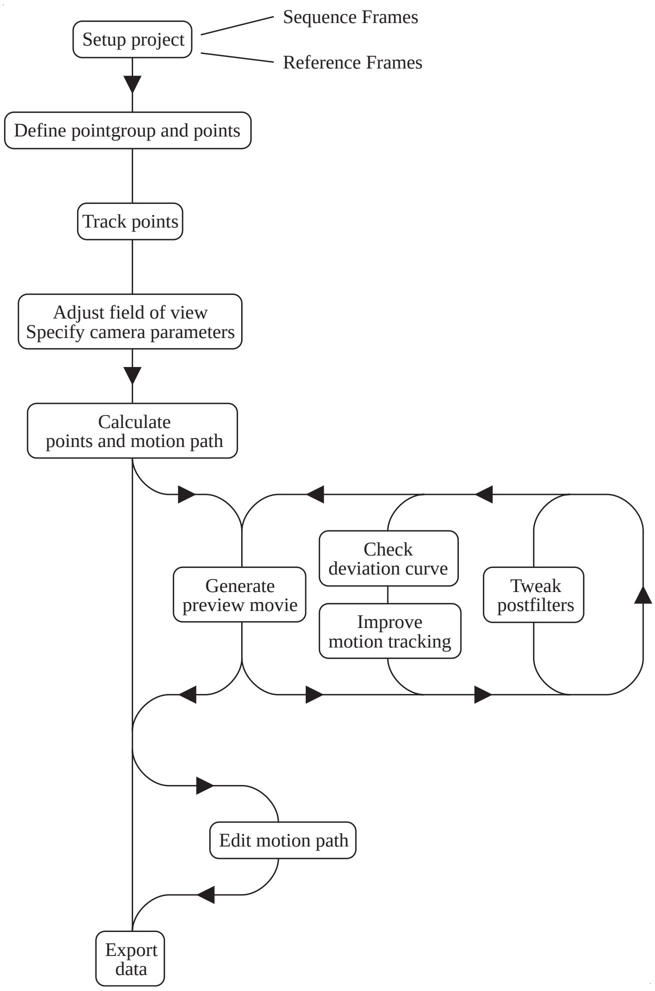
The next step is to create a point group. The expression “point group” is new in version 3. Since in contrast to version 3 of 3DE, version 2 handled only a single point group it has now become necessary to distinguish different point groups by name, to interpret them as objects. In the tutorials you can find examples of projects with more than one point group. The points are then tracked in the sequence (and placed in the reference frames if possible). What we simply called “track points” in our diagram is in practice an interactive procedure in which several functions of the software can be involved: choosing the most precise method, placing key-frames, tracking forward and backward, using image controls when the image material has low contrast, etc.

We have now gathered the basic information about the object or the scenery. However, we have also to specify in what way, the real world is “projected” into the image files, i.e. we need to specify the camera geometry and the digitizing parameters. The camera geometry consists of the focal length and the filmback width and height (or for video cameras: CCD chip size). There are many ways to express these data, e.g. in terms of horizontal and vertical aperture angle or, also, horizontal angle and film aspect ratio. It is up to the user to choose the most convenient way.

The digitizing parameters are essentially the field of view width and height and the pixel aspect. All the parameters mentioned here are in some way dependent on one another, and this dependency structure defines the special appearance of our Camera Adjustment Window. Usually, it is not necessary to enter all parameters because, as mentioned, some of the information is redundant and unknown parameters can be calculated by 3DE. Apart from the basic geometry of the camera it may be necessary to compensate for non-linear effects of the camera optics. These effects are usually referred to as “lens distortion”. Please read the tutorial “Lens Distortion” for more detailed information.

Often, some of the camera parameters are unknown or only approximately known, others are known or can be guessed. The camera parameters which are unknown are calculated by the software. This also includes a calculation of the points in space and the camera motion path. In the diagram we have simply summed up this procedure in the field “Calculate points and motion path”.

When all 3D calculations are done, you could, in principle, export all data to the animation software (or to a compositing system that is able to deal with 3d-objects). However, it might be necessary to render a preview within 3DE and then improve the quality of the calculation. The Status Window displays a curve which expresses the deviation of a point group for a sequence, resolved in frames. It is also possible to examine this curve for each point, and these curves can be used to detect errors in the motion tracking. This is the working cycle “check deviation curve -> improve motion tracking”. In some situations (see tutorials) it may also be necessary to use the postfilter in order to smooth out the motion path. The working cycle usually is “tweak postfilter->generate preview”. Some users prefer to use the final rendering output in high resolution to check the result. In this case the procedure reads “tweak postfilter->export data->render and compose->tweak again” or similar.



Workflow

There are projects (particularly when not enough points are available because the image material does not allow you to track a sufficient number of points) in which it may even be necessary to edit the camera/object motion path. We consider this function optional, and it should be the last step (apart from rendering a preview movie) before exporting the data to the animation package. Everything else such as rendering and compositing is done outside 3DE. If your sequence exhibits lens distortion, you may use the external tool WarpDistort in order to distort the rendered images or to remove distortion from the real images. WarpDistort reads the project file of 3DE and constructs an image operator according to the specified distortion parameters.

Differences between 3DE V2 and V3

The project files are upwardly compatible, i.e. version 2 files are understood by version 3, but not vice versa.

The most important difference between version 2 and verion 3 is that the latter one has “more of everything”. In other words, it is possible to handle more than one point group (as mentioned, the expression point group did not make sense in version 2), more than one sequence and more than one set of reference frames, associated to different cameras.

Version 3 is able to deal with so-called “infinite points”, points which are too far away to be triangulated properly, and which are therefore declared to be located at “infinity”. These points are particularly important when the foreground of the recorded scenery is changing quickly, making it difficult to find a sufficient number of points with a sufficient overlap in time.

The lens distortion model of version 3 consists of five parameters and enables users to handle anamorphic lenses, while the single-parameter model of version 2 was insufficient for this type of lens.

Finally, in version 3 there are a limited number of constraints which allow the user to force the camera to move only in certain limits.

Tutorial

In the following chapter the main features of 3D-Equalizer are presented by means of several examples. The 3D-Equalizer project files that belong to these examples can be found in the directory “<tutorial install directory>/projects/”.

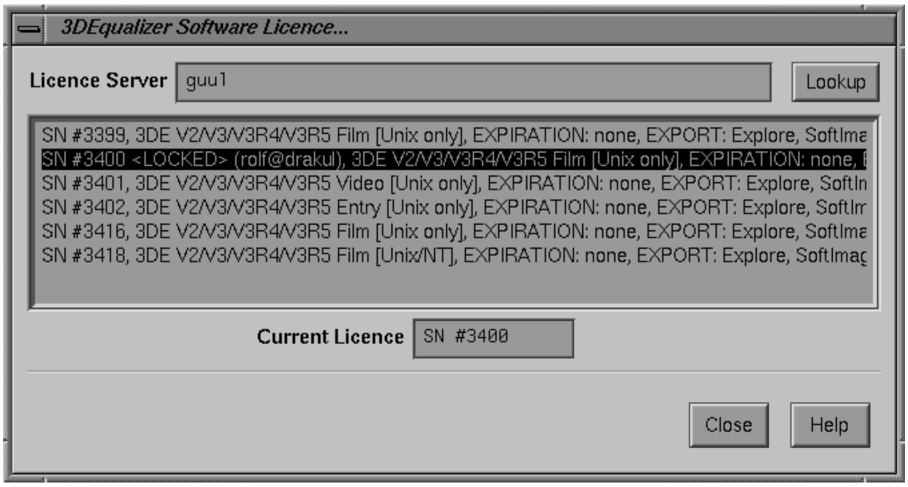
Tutorial 1: Introductory project “ball_bluebox1”

The matchmoving process, determining 3D motion data out of a 2D image sequence can be summed up as follows. The “Environments concept” section explains the efficient handling of 3D-Equalizer’s user interface. In the sections “Creating a new project” and “Creating a new sequence” we describe how to set up a project, including the acquisition of the sequence frames. The sections “Creating a new pointgroup” and “Creating a new point” describe how to enter points which are going to be tracked in the sequence frames. Section “Tracking markers” describes the various methods of tracking the points on the screen. Afterwards, in section “Defining camera parameters”, the camera geometry and motion path are reconstructed. This enables us to take a first look at the preview movie, which is generated as described in the sections “Editing dummy objects” and “Generating a preview movie”. In order to get in contact with your preferred animation package the reconstructed points and the reconstructed camera path are exported. The various exportfilters are mainly described in the “Reference Manual”. This completes the basic procedure of 3D-Equalizer.

Getting started

- *Please start 3D-Equalizer by double-clicking on its icon in the icon catalog window (SGI) or by entering “3dequalizerV3” in a Unix shell window (Linux/SGI).* After a few moments, the requester 3DEqualizer Software Licence opens.

- *Enter the name or IP address of the Flc server workstation (see installation) in the text field Licence Server in order to get a list of the currently available licences. Choose one of them by double-clicking or by selecting and pressing the button Close.*



Licence Server: guu1

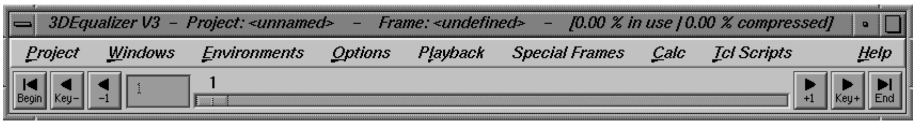
```
SN #3399, 3DE V2/V3/V3R4/V3R5 Film [Unix only], EXPIRATION: none, EXPORT: Explore, SoftIma
SN #3400 <LOCKED> (rolf@drakul), 3DE V2/V3/V3R4/V3R5 Film [Unix only], EXPIRATION: none,
SN #3401, 3DE V2/V3/V3R4/V3R5 Video [Unix only], EXPIRATION: none, EXPORT: Explore, SoftIn
SN #3402, 3DE V2/V3/V3R4/V3R5 Entry [Unix only], EXPIRATION: none, EXPORT: Explore, SoftIm
SN #3416, 3DE V2/V3/V3R4/V3R5 Film [Unix only], EXPIRATION: none, EXPORT: Explore, SoftIma
SN #3418, 3DE V2/V3/V3R4/V3R5 Film [Unix/NT], EXPIRATION: none, EXPORT: Explore, SoftImag
```

Current Licence: SN #3400

Requester: 3DEqualizer Software Licence...

The Main Window

When the Software Licence requester closes, a single, long window opens at the bottom of the desktop. This window is called the Main Window. It mainly contains a few buttons, a textfield and a slider for specifying the current frame. It also contains several menus hosting many functions. 3D-Equalizer's workflow usually starts with functions of the Main Window's menus.



Main Window

Editing 3D-Equalizer's preferences

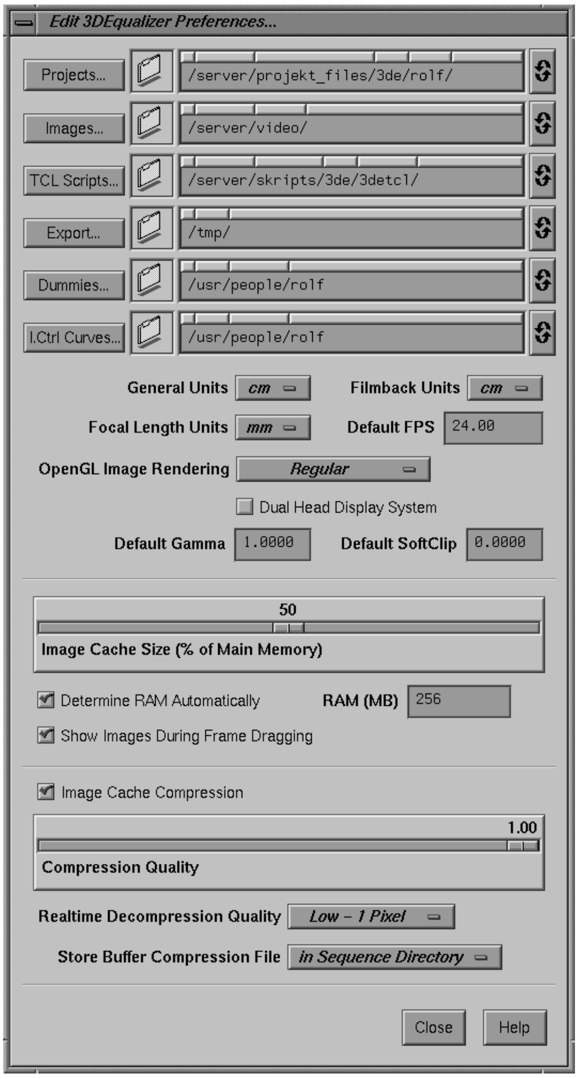
- *Now please select the menu item Main Window::Options::3DEqualizer Preferences.*

The requester Edit 3DEqualizer Preferences opens. The upper text fields contain default search paths for various groups of files, like 3D-Equalizer projects, image files, etc. The button on the left opens a standard file requester in which you can specify the default directory.

The option menu General Units allows you to define the unit system, which is used in various 3D-Equalizer textfields, in order to express distances or positional values.

> **Important note:** In order to enable a smooth workflow, 3D-Equalizer employs several OpenGL graphics functions. It turned out that the quality of OpenGL drivers is extremely different on many operating systems / graphics hardware. You can improve 3D-Equalizer's performance (especially in its Zoom Window) if you try out all available settings for the option menu "OpenGL Image Rendering". "Regular" often seems to be a good setting.

- *Modify the settings of the Edit 3DEqualizer Preferences requester according to your needs, and confirm your edits by pressing the Close button.*



Edit 3DEqualizer Preferences...

Projects... /server/projekt_files/3de/rolf/

Images... /server/video/

TCL Scripts... /server/skripts/3de/3detcl/

Export... /tmp/

Dummies... /usr/people/rolf

I.Ctrl Curves... /usr/people/rolf

General Units: cm

Filmback Units: cm

Focal Length Units: mm

Default FPS: 24.00

OpenGL Image Rendering: Regular

□ Dual Head Display System

Default Gamma: 1.0000

Default SoftClip: 0.0000

Image Cache Size (% of Main Memory): 50

☑ Determine RAM Automatically

RAM (MB): 256

☑ Show Images During Frame Dragging

☑ Image Cache Compression

Compression Quality: 1.00

Realtime Decompression Quality: Low - 1 Pixel

Store Buffer Compression File: in Sequence Directory

Requester: Edit 3DEqualizer Preferences...

Environments concept

Beside 3D-Equalizer's Main Window, there are several other Windows which can be opened (or raised) by the functions of the menu Main Window::Windows. Each window provides a special set of functions, such as handling images, handling points or displaying statistical information.

- *Please select the menu items Windows::Frames, Windows::Zoom, Windows::Overview and Windows::Pointgroups.*

The Frames Window, the Zoom Window, the Overview Window and the Pointgroups Window open up.

- *Arrange the newly opened windows so that they do not overlap each other.*

As you see, opening and arranging just a small number of windows can already take some time. For the different stages in 3D-Equalizer's workflow, different arrangements of windows are useful. 3D-Equalizer is able to recall all kinds of window arrangements. These window arrangements are called "environments" within 3D-Equalizer.

- *Choose the menu item Main Window::Environments::Motiontracking.*

All formerly opened windows are closed, and another set of windows is opened and arranged within a short moment. By using the functions of the Main Window's Environments menu, you can switch very quickly between various environments.

> **Important note:** In order to make use of the environments feature, the "Auto Window Placement" feature of the desktop should be activated (SGI/Linux).

- *Play around with the different environments of the Environments menu. Finally, open the Orientation environment.*

Having the desktop filled up with many windows, can sometimes be distracting if you want to do some other tasks outside of 3D-Equalizer.

- *Please press the iconify button of the Main Window.*

All 3D-Equalizer Windows including the Main Window are closed, and a single icon, representing the entire application, is shown.

- *Click on the remaining icon.*

As you see, all formerly opened windows open up again.

Creating a new user-defined environment

- *Please choose the “Single Window” environment from the Environments menu.*

All windows close except for the Main Window.

- *Select the menu item Main Window::Environments::Configure.*

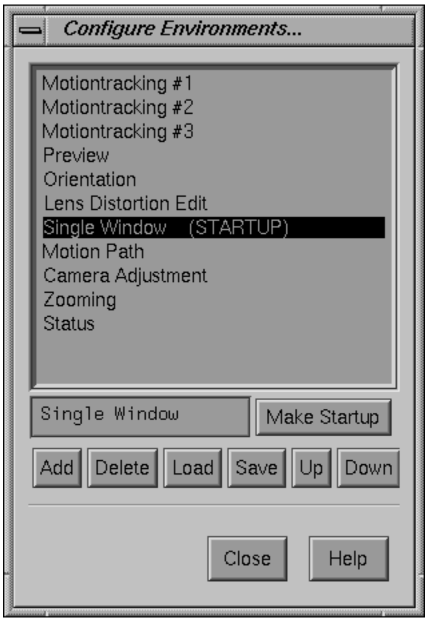
The Configure Enviroments requester opens. By the use of this requester’s functions, you can create, delete and modify the environments of your user account.

- *Press the Add button, in order to create a new environment.*

A new entry “<unnamed>” is added to the bottom of the list.

- *Click on the Environments label in the Main Window’s menu bar in order to display the contents of the Environments menu.*

As you see, there is already a new menu item “<unnamed>”, which represents the newly created environment. Let’s continue to configure it.



Requester: Configure Environments...

- *Please select the newly created environment "<unnamed>" in the Configure Environments requester's list widget. Change its name in the textfield below the list widget into "Custom".*

- *Now, open the Zoom Window, the Overview Window and the Pointgroups Window. Activate the toggle button Main Window::Playback::Show Playback Controls, in order to display the Main Window's playback controls. Arrange the windows, so that they do not overlap each other.*

- *Press the button Save in the Configure Environments requester.*

The last action has stored the current window arrangement, including the Main Window's playback controls toggle button status in the currently selected environment "Custom".

- *Close the Configure Environments requester by pressing its close button.*

- *Choose the environment "Motiontracking" from the Main Window's Environments menu. Thereafter, please choose the newly configured environment "Custom" from the Environments Menu.*

As you see, 3D-Equalizer's user interface is very easy to configure. The position and size of its windows, including the Main Window, can be stored in the environments, as can the various toggle button states.

> **Important note:** In the course of this tutorial we will make extensive use of the default environments which are shipped with 3D-Equalizer. If you have modified some of these default environments, you may want to reset them to their initial state. In order to do that, please select the item Main Window::Environments::Reset To Defaults. Attention, all user-defined environments will be deleted by this command!

Creating a new project

Now it's time to start working on some project. In the following, we will reconstruct the motion of a camera in 3D space by tracking a sequence of 129 frames.

- *Please select the item Main Window::Projects::New.*

3D-Equalizer's database is reset. All objects are deleted, and the user interface is updated. Because we did not create any project related objects up to now, nothing special happens.

- *Choose the "Motiontracking" environment from the Environments menu.*

The "Motiontracking" environment is a good starting point for setting up a new project, because almost all "standard" windows are opened through it.

Creating a new sequence

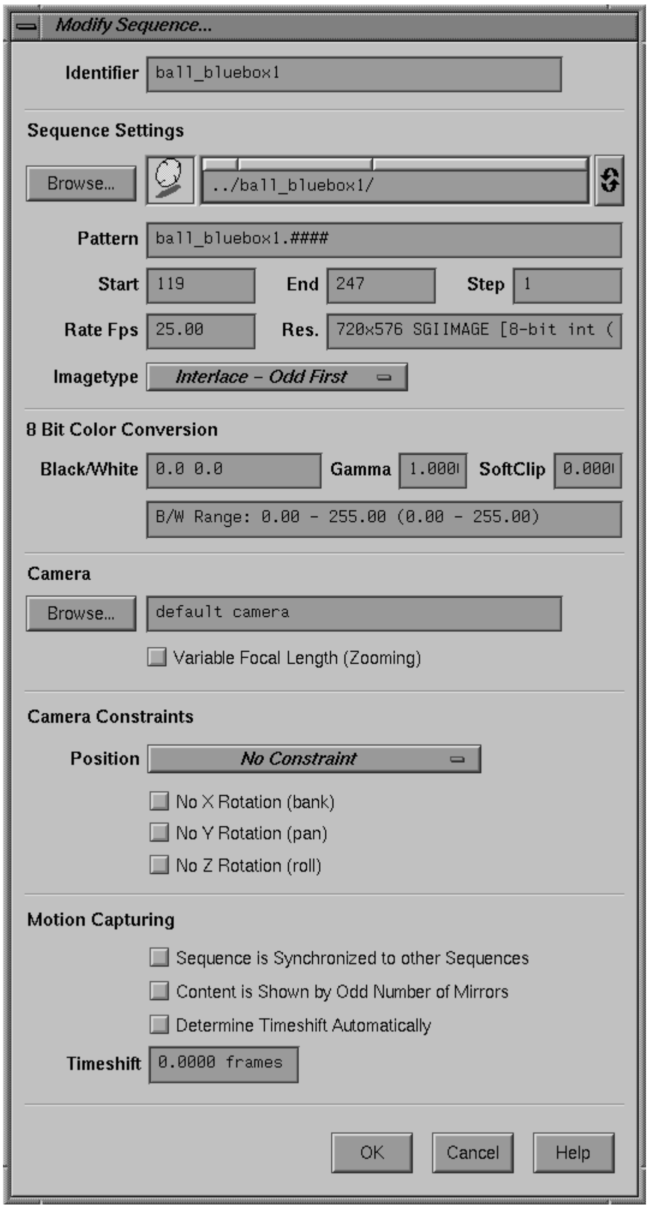
3D-Equalizer is able to handle multiple image sequences within one project. We will extensively discuss this feature in a later tutorial. For this step creating one single sequence, which is the most common case, will be suffice.

- *Please click on the Add button of the Frames Window which is in the lower left corner of the desktop.*

A requester opens, asking you whether you want to create a sequence or a reference frame.

- *Click on the “Add Sequence” button.*

Another requester opens, which allows you to define all the settings associated to the new sequence object to be created.



Modify Sequence...

Identifier: ball_bluebox1

Sequence Settings

Browse... ../ball_bluebox1/

Pattern: ball_bluebox1.####

Start: 119 End: 247 Step: 1

Rate Fps: 25.00 Res.: 720x576 SGIIMAGE [8-bit int (

Imagetype: Interlace – Odd First

8 Bit Color Conversion

Black/White: 0.0 0.0 Gamma: 1.000 SoftClip: 0.000

B/W Range: 0.00 - 255.00 (0.00 - 255.00)

Camera

Browse... default camera

☐ Variable Focal Length (Zooming)

Camera Constraints

Position: No Constraint

☐ No X Rotation (bank)

☐ No Y Rotation (pan)

☐ No Z Rotation (roll)

Motion Capturing

☐ Sequence is Synchronized to other Sequences

☐ Content is Shown by Odd Number of Mirrors

☐ Determine Timeshift Automatically

Timeshift: 0.0000 frames

Requester: Create Sequence...

- *Press the button Sequence Settings::Browse.*

A standard file requester opens.

- *Go to the directory "<tutorial install directory>/ball_bluebox1/" containing the images of the sequence to be created. Double-click on the first image file "ball_bluebox1.0119".*

3D-Equalizer will automatically scan the directory, starting with the selected image file, for other image files associated to the image sequence. Finally the Identifier, Pattern, Start and End textfields are filled with meaningful values. The resolution and the image type of the frames should appear in the textfield Res.

> **Important node:** If the Res. textfield still displays the label "<undefined>", please check the permission modes of the image files and change them, if necessary, by applying the Unix shell command "chmod" (SGI/Linux).

The textfield "Identifier" is filled with the base name of the image files. The contents of this textfield is used to identify the sequence object to be created in the course of 3D-Equalizer's workflow. Furthermore, it is used to identify 3D camera objects during the export process. You can change the sequence's name at any time.

The tutorial sequence is interlaced video at a frame rate of 25 frames per second.

- *Please enter a value of 25 in the Rate Fps textfield.*

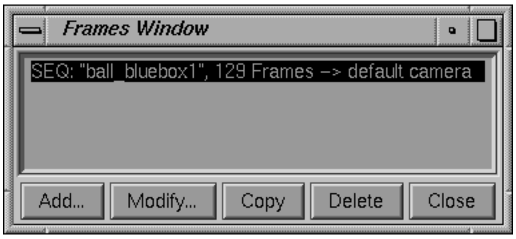
Each interlaced frame contains two so-called fields, time-displaced by 1/50 second. There are two possible settings for interlaced frames: Interlace - Even First and Interlace - Odd First. You may specify here which of the two fields will be used as sequence frame.

- *For the tutorial sequence please set the Imagetype option menu to Interlace - Odd First.*

- *Now press the Ok button, in order to create the new sequence object.*

The Create Sequence requester is closed and a new entry, representing the created sequence object, is displayed in the Frames Window. Our new sequence automatically becomes the current one, thus its entry in the Frames Window is highlighted. Furthermore, the first sequence frame is loaded and displayed in the Overview Window and in the Zoom Window. The frame filename is displayed right below the image in the Overview Window's display area.

To modify the properties of our sequence, press the Modify button of the Frames Window, or simply double-click on its entry.



SEQ: "ball_bluebox1", 129 Frames -> default camera

Frames Window

Navigation within the sequence

There are various methods for navigating through the sequence. The Main Window contains several interaction components to change the current frame.

- *Please click several times on the "+1" button of the Main Window, enter a frame number into the Main Window's textfield, drag the frame silder, press the buttons "End"/"Begin" and watch the results.*

- *Move the mouse pointer over the display area of the Zoom Window, and press the Cursor-left, then the Cursor-right key of the keyboard.*

As you see, you can scrub through the sequence frame by frame. This feature is very useful during the motiontracking procedure.

- *Activate the toggle button Main Window::Options::Frame Keyboard Grid::10 Frames. Move the mouse pointer over the display area of the Zoom Window, and press the Cursor-left, then the Cursor-right key of the keyboard.*

Now you can jump back and forth to every 10th frame. This feature is useful in conjunction with very long image sequences. However, we will not need this in the following.

- *Activate the toggle button Main Window::Options::Frame Keyboard Grid::Off.*

Building up a cache movie

Associated to each 3D-Equalizer project file are two movie archives. These movie archives contain the cache movies and the preview movies of the project's sequences and reference frames. The purpose of preview movies is explained later in this tutorial. We will now concentrate on the cache movies.

Generally speaking, cache movies are mainly used to speed up the workflow within 3D-Equalizer. They help you to create preview movies and streamline the frame navigation, especially in conjunction with high resolution image sequences. To make it even more clear, it is not necessary to use this feature, but it can make your work much easier. In order to use it, you must work on an existing (saved) 3D-Equalizer project.

- *Please choose the menu item Main Window::Projects::Save As. A standard file requester opens. Enter a name for the project to be saved then press the Ok button.*

While saving the project, a requester opens and the mouse pointer changes into a clock symbol.

- *Select the item Main Window::Playback::Create Cache Movie. A requester opens up. Please click on the “Create only Current Cache Movie” button.*

3D-Equalizer starts to build up the cache movie for the formerly created sequence “ball_bluebox1”. This process will take some time. Please wait until it is completed.

Using the cache movie feature

We will no experience how the cache movie feature is used.

- *Please open the “Preview” environment.*

This environment is useful to play back preview and cache movies.

- *Activate the toggle button Overview Window::View::Display::Cache Movie.*

The Overview Window now displays the contents of the cache movie.

- *Select the item Main Window::Playback::Output::All Windows, and press the button “Play”.*

3D-Equalizer starts the playback process. Instead of the preview movie, the cache movie is played back in the Overview Window. The Main Window’s title bar displays some statistical information about playback speed, etc.

- *Press the “Stop” button in order to interrupt the playback, and choose the “Motiontracking” environment from the Environments menu.*

- *Drag the frame slider back and forth through the sequence.*

If there is enough RAM in your machine, the entire image sequence (129 frames) has been loaded into memory while building up the cache movie. In this case you can scroll through the sequence and instantly watch the images updated in the Zoom- and Overview Window. However, if you do not have enough memory, some frames appear to be black unless you release the mouse button. Especially if you work on high resolution image sequences, only a few frames are kept in memory at the same time. In this case, the cache movie can help you to examine the sequence.

- *Activate the toggle button Overview Window::View::Show Cache while Dragging, and once more drag the frame slider back and forth through the sequence.*

As you see, while dragging the slider the Overview Window displays the contents of the cache movie. As soon as you release the mouse button, the Overview Window switches back to its regular display mode.

- *Activate the toggle button Overview Window::View::Display::Cache Movie.*

The Overview Window now continuously displays the contents of the cache movie.

> **Important note:** You can force 3D-Equalizer to load as many images as possible into memory, by selecting the function Main Window::Playback::Fill Imagebuffers. If you work on video resolution images and if you have enough RAM, this feature can speed up the entire motiontracking process tremendously!

Creating a new pointgroup

So far, we described simple preparations. In the next sections we will actively perform steps to determine the camera's motion in 3D space.

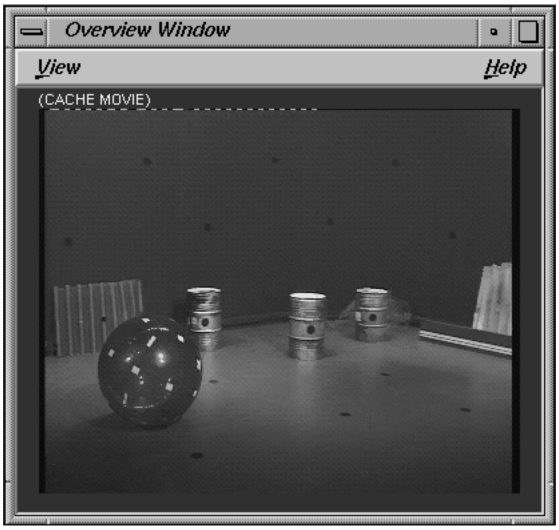
The basic procedure is always the same. A few feature points in the sequence are tracked two-dimensionally on the screen using the so-called point objects. Each point is associated to a certain feature point shown in the footage. This can be the edge of a window, the tip of an actor's nose or an intentionally placed marker. The regular requirements for reconstructing the motion of a camera or an object are as follows:

1. All features, tracked by points, to be used for the reconstruction have to be rigid with respect to each other.
2. In each frame, at least four points should be placed on the screen.
3. There must be at least six points.
4. In at least two of the frames these six points must be placed on the screen, they are called rootframes.

You can define the two rootframes manually, or let 3D-Equalizer choose them automatically. To guarantee a successful reconstruction process, the two rootframes should show the set or the object from different points of view! In the simplest case, there are four common points in all remaining frames and these four points also appear in the rootframes. In practice it often happens that four points are not enough. In fact, there can be very complicated situations containing points that go in and out of frame during the entire sequence. In the past, there have been projects with over 100 tracking points. A situation of this type, featuring 16 tracking points, is described in the tutorial “Reference frames”. In this tutorial example we will create and track six points, and let 3D-Equalizer choose the rootframes automatically.

Another great feature of 3D-Equalizer is the ability to reconstruct the 3D motion of moving objects and the camera at the same time. For instance with our example sequence “ball_bluebox1” it is possible to determine the motion of the ball and the motion of the camera separately within the same project, and export them to an animation package. We will demonstrate this in the tutorial “Multiple pointgroups”. For now, we will concentrate on the camera motion only.

For each three-dimensional motion to be reconstructed, there must be a so-called pointgroup object, associated either to a camera or an object. Each pointgroup has its “own” set of points that are used for the reconstruction process. As already mentioned above, the features, tracked by points of a pointgroup, always have to be rigid with respect to each other.



Overview Window displaying a cache movie

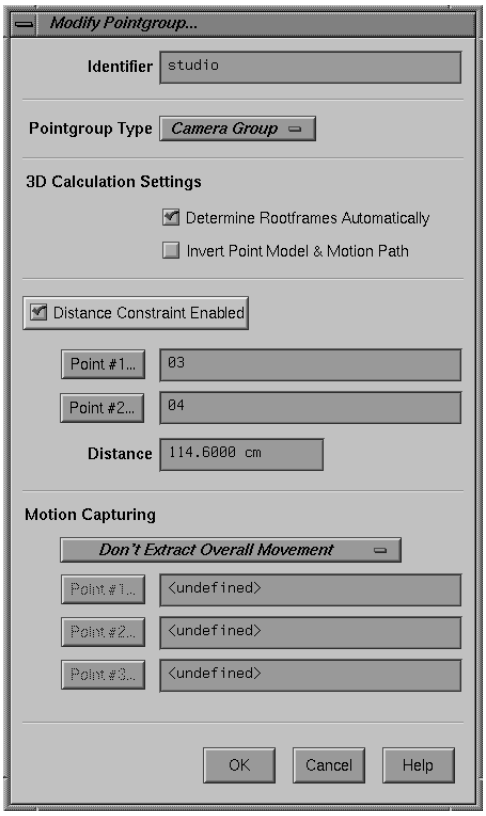
Let us examine the tutorial sequence.

- *Please open the “Motiontracking” environment from the Environments menu and activate the toggle button Overview Window::View::Show Cache while Dragging.*

- *Pick the frame slider of the Main Window and scroll back and forth through the sequence, while watching the Overview Window.*

The camera seems to be mounted on a dolly and moves from left to the right, shooting a bluebox studio, while a blue shiny ball with white markers attached rolls on the studio floor from the right to the left. On the studio floor and on the bluebox walls are several markers. We will track six of these markers, four on the floor and two on the wall, in order to determine the camera’s motion relative to the studio. First of all we have to create a pointgroup object.

- *Select the menu item Pointgroups Window::Pointgroup::Add Pointgroup.*



Modify Pointgroup...

Identifier: studio

Pointgroup Type: Camera Group

3D Calculation Settings

☑ Determine Rootframes Automatically

□ Invert Point Model & Motion Path

☑ Distance Constraint Enabled

Point #1...: 03

Point #2...: 04

Distance: 114.6000 cm

Motion Capturing

Don't Extract Overall Movement

Point #1...: <undefined>

Point #2...: <undefined>

Point #3...: <undefined>

Requester: Create Pointgroup...

The requester Create Pointgroup opens. The most interesting attribute of a pointgroup is the “Pointgroup Type” parameter. There are camera pointgroups and object pointgroups. A camera pointgroup’s points are used to reconstruct the motion of a camera, whereas an object pointsgroup’s points are used to reconstruct the motion of a moving object (like the blue ball in our sequence).

- *Set the Pointgroup Type option menu to “Camera Group”, change the identifier of the pointgroup to-be-created to “studio” and press the button OK.*

The requester closes and a new entry representing the newly created pointgroup appears in the upper list widget of the Pointgroups Window.

Creating a new point

- *Please press the button Add in the Pointgroups Window, to create a new point associated to our pointgroup “studio”.*

The requester Create Point opens. The text field Identifier enables you to enter a name for the point. Points may be given names like A, B, C or 1, 2, 3. You can also use more complicated names like “near_to_hair“, “eyebrow“, “near_to_nose“ and so on. Please try to avoid blanks in the names. Using meaningful, long names makes it easier to identify the points in more complicated projects. 3D-Equalizer will provide a unique name (“01”) automatically, so you don’t have to change it.

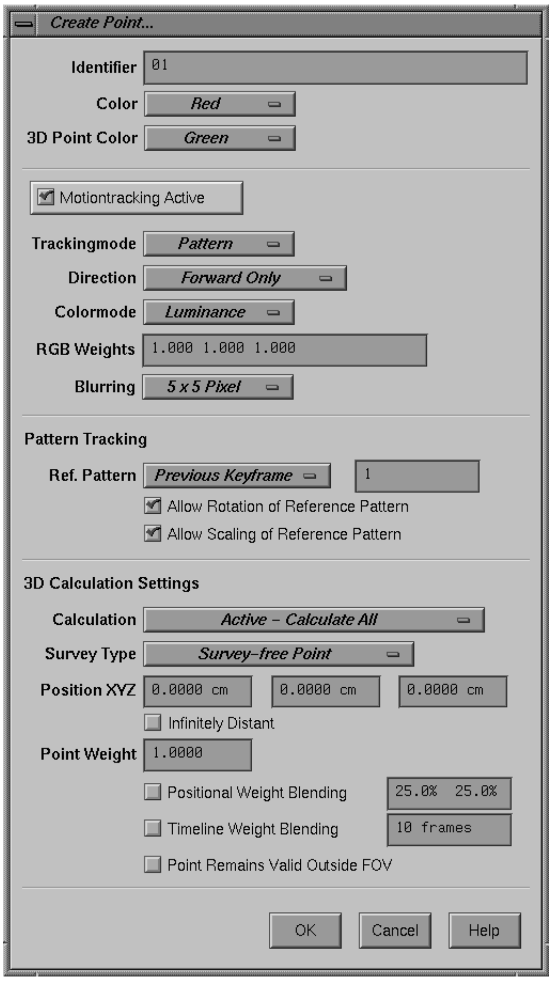
Below the identifier field you see an option menu Color. This defines the color in which the point will be represented on the screen. You should choose this color according to the contents of the footage in order to achieve greater contrast. In our case red is quite good. If you like green better, feel free to set the selector to “Green”.

3D-Equalizer mainly features three different tracking modes. The “Pattern” tracking mode is a standard method by which your are able to track all types of features found in the footage. 3DE’s “Marker” tracking mode is specialized on tracking markers (unicoloured spots) only. Finally, the “Edge/Corner” tracking mode is specialized on tracking corners, build out of straight edges that come together. 3DE’s different tracking modes are explained in detail in the tutorial “Tracking points on the screen.”.

- *Set the option menu Trackingmode to “Marker”.*

This enables the markertracking mode for the to be created point.

- *Press the button OK, in order to close the requester and confirm your settings.*



Create Point...

Identifier 01

Color Red

3D Point Color Green

☑ Motiontracking Active

Trackingmode Pattern

Direction Forward Only

Colormode Luminance

RGB Weights 1.000 1.000 1.000

Blurring 5 x 5 Pixel

Pattern Tracking

Ref. Pattern Previous Keyframe 1

☑ Allow Rotation of Reference Pattern

☑ Allow Scaling of Reference Pattern

3D Calculation Settings

Calculation Active – Calculate All

Survey Type Survey-free Point

Position XYZ 0.0000 cm 0.0000 cm 0.0000 cm

□ Infinitely Distant

Point Weight 1.0000

□ Positional Weight Blending 25.0% 25.0%

□ Timeline Weight Blending 10 frames

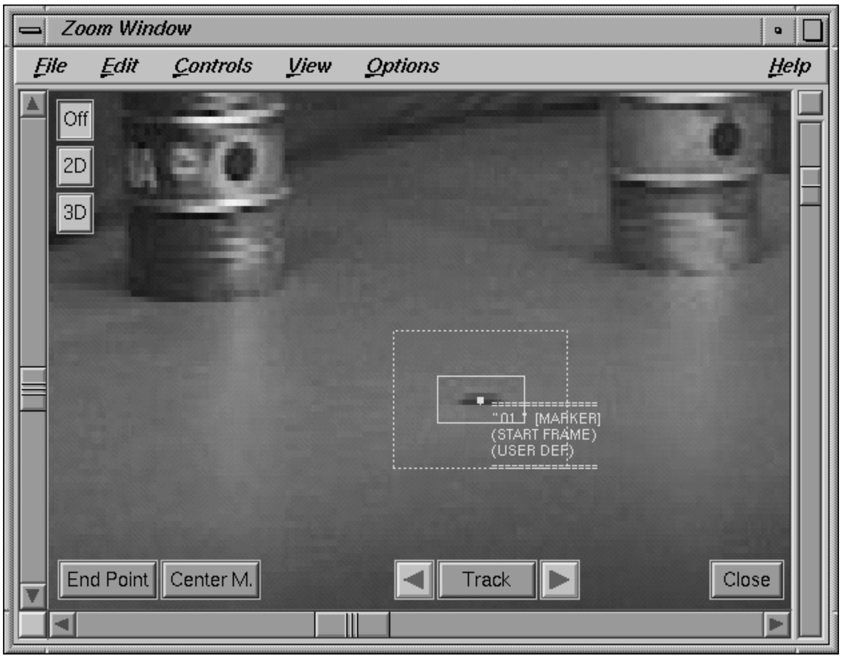
□ Point Remains Valid Outside FOV

Requester: Create Point...

The point is created and a new entry appears in the lower list widget of the Pointgroups Window. If you wish to change the points attributes again, select its entry in the Pointgroups Window and press Modify or simply double-click on it. A requester similar to the requester Create Point appears.

Tracking markers

The next task is to track six markers in the footage, thus producing a two-dimensional motion curve for each point. For this purpose, please take a closer look at the Zoom Window. This window enables you to track and edit the point's 2D motion curves.



Zoom Window

Before you start tracking the 2D motion curve of point "01" it is useful to get familiar with the functionality of this window. In the display you see the currently selected frame, or at least a cut-out part of it.

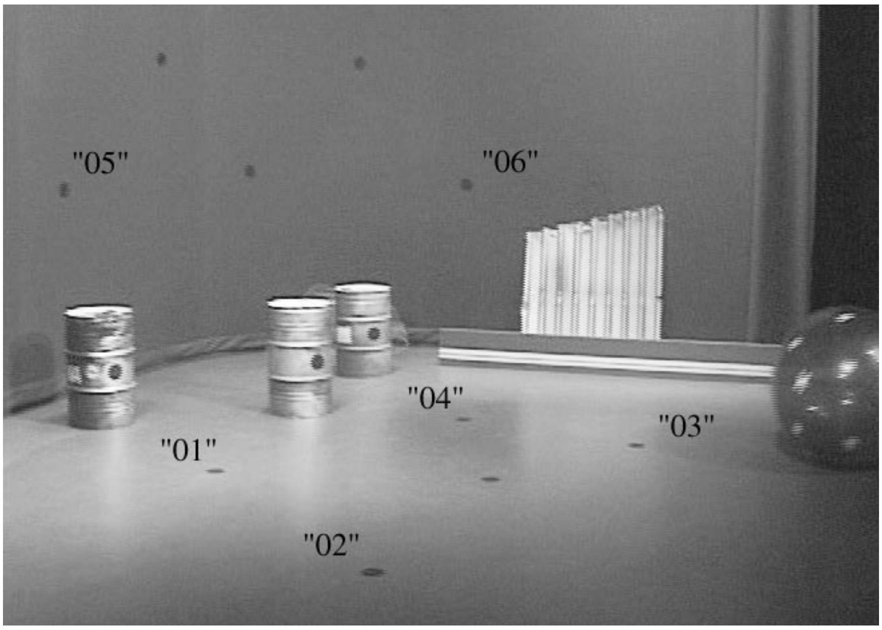
- *Please press the middle mouse button while the mouse pointer is in the Zoom Window's display area, and drag the mouse around.*

While the middle mouse button is pressed, the mouse pointer changes into a cross symbol, and the cutout area can be moved.

- *Press the right mouse button while the mouse pointer is over the Zoom Window's display area, and drag the mouse back and forth.*

The cutout area is zoomed in and out. The position of the mouse pointer in the display at the moment when the mouse button is pressed specifies the point of interest, i.e. the point in the frame whose position remains unchanged while zooming. Pressing the right mouse button in the display and moving down will cause a zoom in, while moving up will lead to a zoom out. Positioning the cut-out part and zooming can also be achieved by using the sliders left, right and underneath the display.

We shall now start to track our recently created point “01”, thus creating a 2D motion curve for it. Thereafter, we will track five additional markers on the studio’s floor and walls as indicated by the picture below.



Six to-be-tracked markers

- *Go to frame 1 using the Main Window’s interaction components.*

- *Zoom and move the cutout area so that the left-most marker located on the studio floor is clearly visible in the middle of the Zoom Window.*

- *Make sure that point “01” is selected in the Pointgroups Window and click on the center of the marker with the left mouse button.*

You have now defined the starting point (a new keyframe) of the 2D motion curve to be tracked. Two rectangles appear around the starting point: the inner one is solid, and the image cut-out within this square is called search pattern. The outer square is dotted and the area marked by this square is called search area.

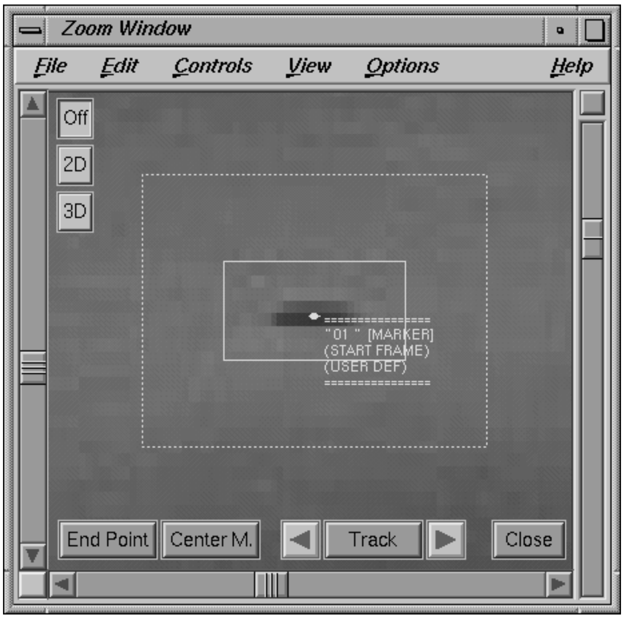
- *Hold down the CTRL-key and click-drag on a border of the inner square (search pattern).*

As you see, the search pattern's size is modified. The search area's square can be modified in the same way.

- *Hold down the CTRL-key and click-drag into the interior of the search area.*

Search pattern and search area are moved relative to the starting point.

- *Now please adjust the search pattern so that the marker is entirely contained in the solid square and only the marker and the unicoloured surface lie within the solid square (see figure below). Leave a small border of at least two pixel between the marker and the inner tracking box. Adjust the search area to, say, twice as big as the search pattern.*

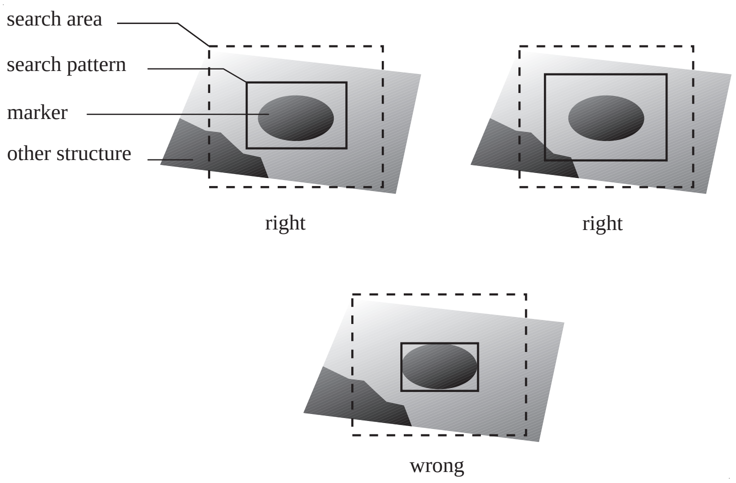


Properly adjusted tracking areas

- *Press the button Center M. ("center marker") several times.*

3D-Equalizer analyses the search area’s contents, seeks the center of the marker, and places the starting point on it. As long as the “Center M.” button is pressed, a white line is drawn around the marker, which indicates how clearly 3DE is able to distinguish the marker from its surrounding background.

This function is very useful for finding out whether a marker will be clearly identified by the software or not. There are several reasons for which 3D-Equalizer might not be able to find a marker’s center in markertracking mode. We will discuss them in detail in the tutorial “Tracking points on the screen”. However, the most common problems are badly adjusted tracking areas, for instance search patterns which are so small that markers do not fit into them (see figure below).



Properly and badly adjusted search patterns

Now that we have ensured that 3D-Equalizer is able to find the marker’s center, we can continue with the actual tracking process.

- *Press the right-arrow button located on the right-hand side of the button “Track”.*

The next frame (frame 2) is loaded into memory, the marker’s center is located for this frame, and the point’s 2D motion curve is updated accordingly.

- *Press the right-arrow button several times in order to track the next few frames manually.*

This procedure is not very comfortable, and should only be used in situations where 3D-Equalizer has problems tracking a feature.

- *Press the button “Track”, in order to start the tracking procedure.*

The label of the button changes to “Stop” and 3D-Equalizer starts tracking the marker, frame by frame. While the tracking procedure is in progress, the Zoom Window navigation controls are still active.

- *Modify the Zoom Window’s cutout area in order to follow the marker being tracked, by click-dragging the middle and/or right mouse button.*

Approximately at frame 64, the blue ball entirely obscures the marker, and can no longer be tracked. The red dot which indicates the 2D motion curve’s position in the current frame, jumps around, out of control.

- *Press the button “Stop” (former “Track”).*

The button changes its label back to “Track”, and the tracking procedure stops. Since the marker simply is not visible in the following frames, we have to disable this point’s 2D motion curve up to the frame in which the marker appears again.

- *Drag the Main Window’s frame slider back to frame 63, the last frame in which the marker is entirely visible.*

- *Activate the toggle button Zoom Window::View::Show MotionCurve.*

A red line is displayed in the Zoom Window area, representing the point’s 2D motion curve.

- *Press the button “End Point” and watch the 2D motion curve displayed in the Zoom Window as well as the contents of the Timeline Window.*

By pressing this button we have declared the current 2D motion curve’s position as an end point (and a new keyframe). Thus, in all subsequent frames the point’s 2D motion curve becomes invalid unless a new start point (keyframe) is defined. The valid part of the 2D motion curve is drawn as a solid line whereas the invalid part is drawn as a dotted line. Pressing the button “End Point” a second time would cause 3D-Equalizer to change the end point back to its initial state and thus make the 2D motion curve valid again during the entire sequence.

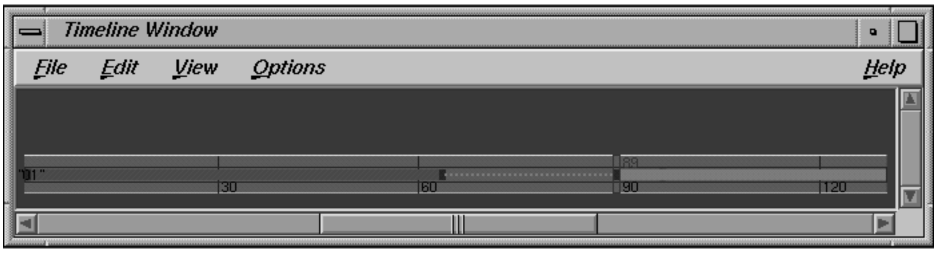
The Timeline Window displays a schematic overview for the 2D motion curves of a pointgroup’s points. Valid motion curve parts are displayed by a grey rectangle, invalid parts are displayed by a dotted line.

- *Drag the Main Window’s frame slider forward and watch the cache movie being displayed in the Overview Window, until the marker is completely visible again (approximately at frame 89).*

Since the marker is now visible again, we can continue to track it until the end of the sequence.

- *Move the Zoom Window’s cutout area so that the marker is clearly visible, and define a new start point (keyframe) by pressing the left mouse button over the marker’s center.*

The 2D motion curve becomes valid again, starting from the current frame. Its representations within the Zoom- and Timeline windows are updated accordingly (see image below).



Schematic representation of point 01's 2D motion curve in the Timeline Window

- *Press the button "Center M." several times, in order to make sure that 3D-Equalizer is able to identify the marker's center clearly.*

- *Re-activate the tracking procedure, by pressing the button "Track".*

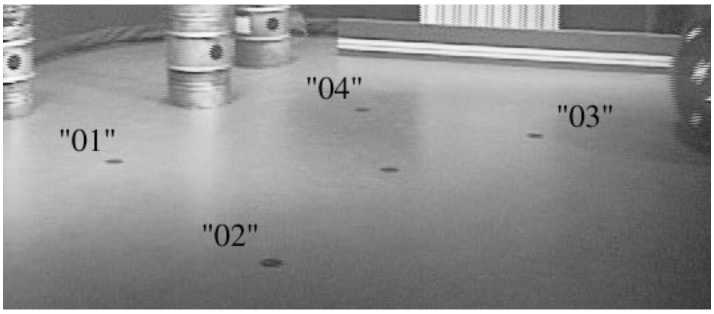
It is sometimes laborious to follow the marker being tracked manually in the Zoom Window.

- *Activate the toggle button Zoom Window::Options::Auto Center, without stopping the tracking procedure.*

The current position of the point's 2D motion curve is now automatically centered in the Zoom Window's display area, which is much more convenient.

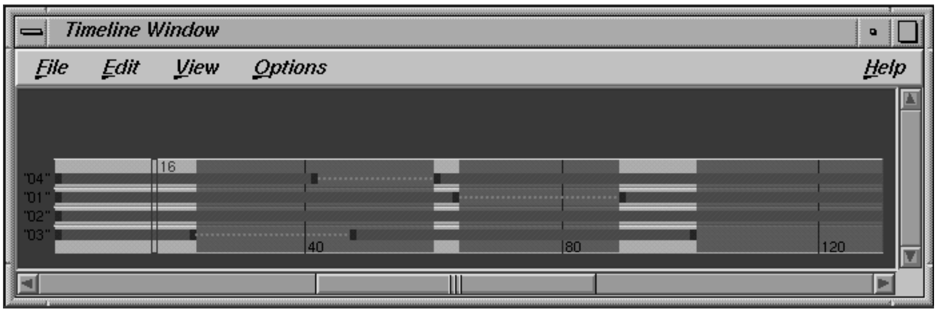
Watch the marker being tracked, until the end of the sequence is reached. Now, tracking point "01" has been completed.

- *Please continue your work by tracking three additional markers on the studio floor. Create the points "02", "03", "04" (markertracking mode) and track their associated markers using the techniques you learned in this section (see picture below).*



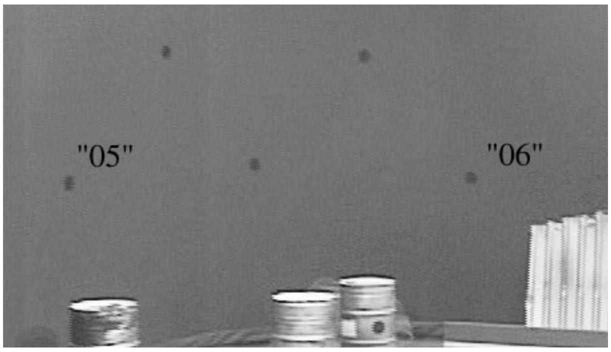
Four to be tracked markers located on the floor

After you have finished tracking the three additional points, the Timeline Window gives you a pretty good overview of the current situation. In some frames all four points have been tracked, in others only three or two 2D motion curves are valid. The frames in which not enough motion curves are valid are indicated in red (see picture below).



Timeline Window displaying 2D motion curves of four points

- *Now please create and track the two remaining points “05” and “06”. As already mentioned, a good choice are two markers on the bluebox walls (see picture below).*



Markers to be tracked using point “05” and “06”

After you have finished tracking these markers, each frame of the sequence contains at least four 2D motion curves which are valid, and thus the Timeline Window does not show any red frames. The actual motiontracking process is finished, and we can now start to perform the steps to reconstruct three-dimensional motion data out of the 2D motion curves we have just tracked.

Adjusting the field of view

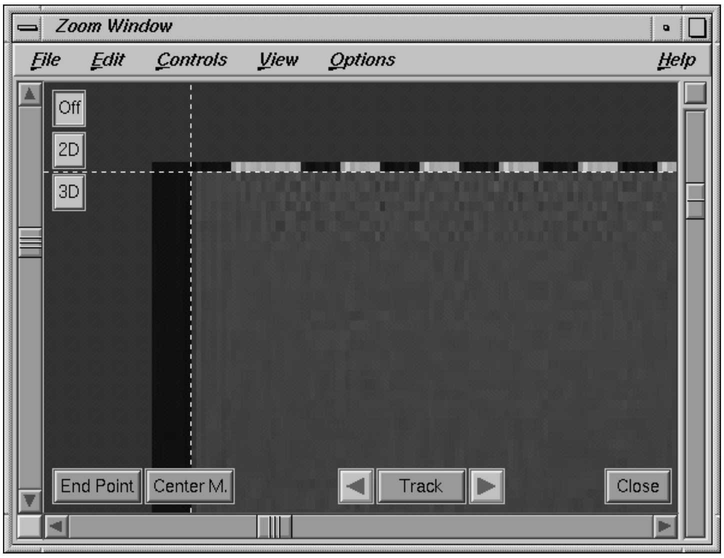
The process of calculating the 3D motion path requires, that we are able to specify, how images from the real world are projected and digitized in the image files. This means we have to specify the position of the filmback (film gate) and, if available, the center of projection axis within the digitized image.

- *Please open the environment “Motiontracking #3” from the environments menu, and go to frame 80.*

- *Zoom out the frame, so that all of it is visible in the Zoom Window’s display area.*

As you see, the frames of the tutorial sequence contain black areas on the left and right sides which do not belong to the image itself. In the process of digitizing black areas can appear above and below the image as well. The rectangular area of the image without the black parts is called the field of view. The field of view borders are represented as dotted white lines in the Zoom Window display. The center of the field of view is represented by a cross of dotted lines. It actually represents the center of projection axis of the camera model which 3D-Equalizer uses for the reconstruction process.

You should now shift the white lines so that they frame the field of view as precisely as possible.



Adjusting the field of view

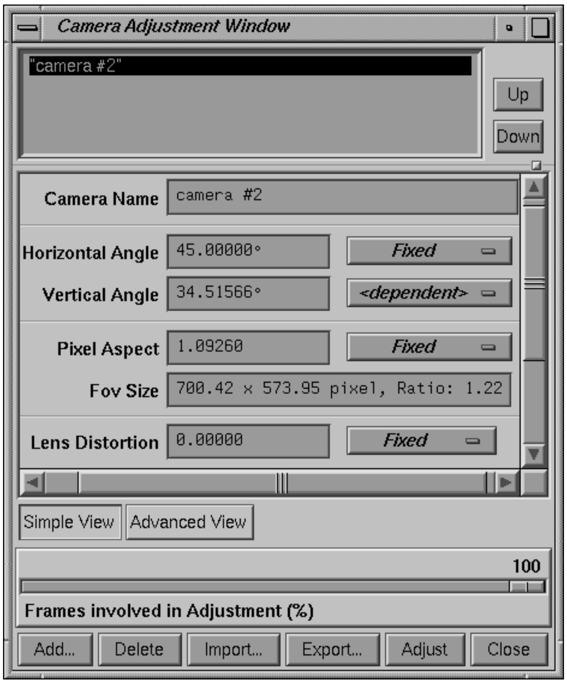
- *Zoom the upper part of the frame. Hold down the CTRL-key and click-drag on the upper white line with the left mouse button.*

As you see, you are able to shift the line up and down.

- *Adjust all four FOV lines, as precisely as you can.*

Defining camera parameters

We have finished the labour-intensive two-dimensional motiontracking task, and now we are going to reconstruct the camera’s 3D motion path. For this purpose, we have to specify the attributes of the camera used for recording the sequence. These attributes are, for instance: focal length, lens distortion, etc. If we knew all the camera parameters, we could simply enter them in the Camera Adjustment Window’s textfields and then let 3D-Equalizer calculate the 3D camera motion path. However, very often some of the camera parameters are missing. 3D-Equalizer can calculate the missing data for you.



Camera Name: camera #2
Horizontal Angle: 45.00000° Fixed
Vertical Angle: 34.51566° <dependent>
Pixel Aspect: 1.09260 Fixed
Fov Size: 700.42 x 573.95 pixel, Ratio: 1.22
Lens Distortion: 0.00000 Fixed
Frames involved in Adjustment (%): 100

Camera Adjustment Window (simple view)

- *Please open the environment "Camera Adjustment" from the Main Window's Environments menu.*

On the right-hand side there is the Camera Adjustment Window. It allows you to create, delete and modify so-called camera objects. A camera contains the above mentioned camera parameters such as, focal length, lens distortion, etc.

There are no three-dimensional data such as camera position and camera rotation assigned to the camera object! Such information is assigned to sequences, reference frames and pointgroups only. Each sequence or reference frame must be linked to a camera, in order to define its camera parameters.

Now, let's try to familiarize ourselves with the Camera Adjustment Window's handling and let's try to link a new camera to our sequence.

- *Press the button Add in the lower left corner of the Camera Adjustment Window.*

You have created a new camera, named "camera #2". It is represented by a new entry in the upper list widget. Obviously, this list widget contains all available cameras of the current project. The attributes of the currently selected camera are displayed in the textfields below.

- *Select the entry “default camera” and press the button Delete.*

You have deleted the camera named “default camera”. Its entry in the list widget disappears and all textfields are cleared.

- *Select the entry of the recently created camera “camera #2”.*

The textfields of the Camera Adjustment Window display the camera’s attributes. Now we will link the sequence to it.

- *Make sure the sequence is selected in the Frames Window, and press the button Frames Window::Modify.*

The requester Modify Sequence opens.

- *Press the button Camera::Browse.*

Another requester opens, displaying a list of all available cameras in the current project.

- *Select the entry “camera #2” and press the button OK.*

The camera “camera #2” is now linked to the sequence. The Modify Sequence requester indicates this in the textfield right beside the Camera::Browse button.

- *Press the button Ok of the Modify Sequence requester in order to confirm your edits.*

> **Important Note:** You should use as few cameras as possible within a project. If different frames have been recorded with the same camera, using the same focal length, you should create one camera with these settings and link all frames to it. 3D-Equalizer actually benefits greatly from the information such as that different frames have been recorded with the same camera (settings).

Now let’s take a closer look at the camera’s attributes. Each camera consists of 14 different camera parameter values. It can sometimes be difficult to keep the overview, especially for new users to the matchmoving field. Therefore, the Camera Adjustment Window hides all advanced camera parameters which are not strictly necessary.

- *Please click on the button Advanced View.*

3D-Equalizer now shows every editable camera parameter. You see several textfields containing predefined camera parameters. Next to some of the fields there are selectors that can assume the values Fixed, Fine Adjust, Unknown and <dependent>. The selector for Lens Distortion can assume the values Fixed, Fine Adjust and Unknown.

In the textfield Fov Size the size of the field of view, as you adjusted it in the previous section, is displayed. In this tutorial sequence we are given an interlaced video sequence (as you specified in the requester Create Sequence requester). Each frame consists of two „half frames“ (fields) shifted by one pixel, i.e. half of the vertical resolution is not displayed. Please keep in mind that the value of Fov height refers to the height of the entire frame, not of a half frame.

The interaction components of this window are described in detail in the chapter “Selected Topics”, section “Camera adjustment”. We also have another extensive tutorial titled “Camera adjustment”.

In this tutorial we limit ourselves to the most important functions. The general principle of handling this window is as follows: A camera, as we understand it in the context of 3D-Equalizer, is determined by four parameters. These are the parameters of which the textfields are displayed in pink. All other parameters (grey) depend mathematically on the pink ones. You can choose which of the parameters are employed by 3D-Equalizer for determining the camera. When you play around with the selectors you will notice that only few of all imaginable combinations of settings are possible. This is because 3D-Equalizer will ensure that the camera parameters are always consistent. You cannot set a selector to <*dependent*> because 3D-Equalizer will calculate this as necessary.

If a parameter is unknown, set its selector to Unknown. In this case you cannot enter a value in the corresponding text field, since you already signalled to 3D-Equalizer that you do not know the value. When you know one of the parameters approximately, set its selector to Fine Adjust and enter the value. If you are sure about a parameter, set its selector to Fixed and enter the value.
In most of the cases you will probably not have to deal with the advanced settings, therefore let’s hide them.

- *Press the button Simple View, in order to hide the advanced settings.*

In principle, there are only two parameters (plus lens distortion), that have to be defined for a proper camera. First we have to specify the pixel aspect of the image files of our sequence. This value mainly depends on the hardware used for digitizing the images. Here is a short list of common image material including resolution and pixel aspect:

Video, PAL	720x576	1.06667
Video, PAL	768x576	1.0
Video, NTSC	720x486	0.9
Film, scanned	any	1.0

Common pixel aspects

In our case, the image size is 720x576 pixels, interlaced video, PAL system. It was digitized using a SGI O2 workstation. Unfortunately, the datasheet of the O2 says that pixel aspect at a resolution of 720x576 is non-standard: 1.0926.

- *Set the option menu beside the Pixel Aspect field to Fixed, and enter the value 1.0926 into the textfield.*

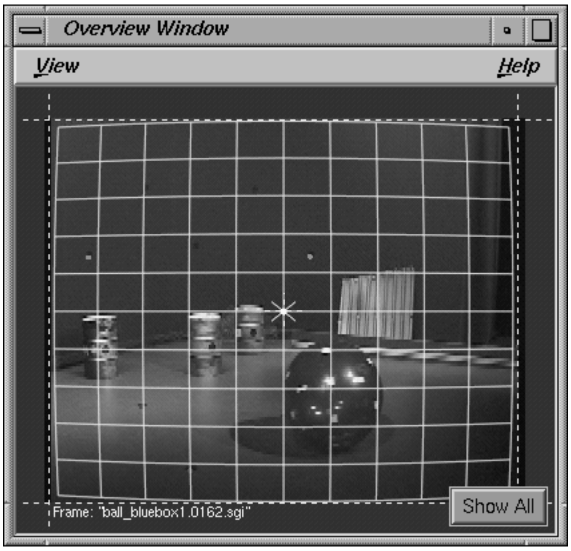
Next, let’s take care of the lens distortion parameter. Real world camera lens systems sometimes distort the filmed images in a non-linear manner. That means, the image content is “bent” towards the edges.

- *Activate the toggle button Overview Window::View::Show Distortion Grid.*

A red grid is now displayed over the current frame in the Overview Window. It visualizes the lens distortion of the camera linked to the current sequence.

- *Enter several different values, ranging from -0.05 to 0.2, into the Lens Distortion textfield and watch the Overview Window’s distortion grid.*

A value of 0.0 yields no lens distortion effect. Values smaller than 0.0 results in a pillow-type, whereas values greater than 0.0 results in a barrel type distortion grid. Typical values of lens distortion for a video camera are 0.02 to 0.08. Good film camera lenses will often exhibit no visible lens distortion at all, so that the value may be set to zero.



Distortion Grid

A consumer video camera has been used to shoot the sequence. There are some minor lens distortion artefacts visible.

- *Enter a lens distortion value of 0.05 and set the respective option menu to Fixed.*

Finally, we have to define the type of camera lens used for recording the sequence. In principle, there are two different ways in to do that in 3D-Equalizer. You can specify a focal length, a horizontal or a vertical camera angle.

Usually, information is available about the focal length of the camera lens. Unfortunately, this parameter depends highly on the size of the film gate (size of the negative). For instance a 10mm lens for a video camera is totally different from a 10mm lens for a 35mm film camera, because the CCD chip (equal to film gate) of a video camera is much smaller than the film gate of a 35mm film camera.

If one knew the exact size of the film gate and the focal length, then the camera lens could be properly defined. In addition there is another problem. Images are often cropped during the digitizing process. That means, the digital image may represent only a fraction of the actual film gate size. In this case the respective focal length must be properly adapted to the "cropped" film gate size, which can be quite difficult.

It is much more intuitive to define the camera lens by its horizontal or vertical camera angles. These parameters are not dependent on film gate size. For instance, the 45.0 ° horizontal camera angle of a video or a film camera will produce the same kind of image. A horizontal angle greater than 50.0 ° represents a wide angle camera lens, and a horizontal angle smaller than 30.0 ° represents a narrow angle (zoom) camera lens (see pictures below).

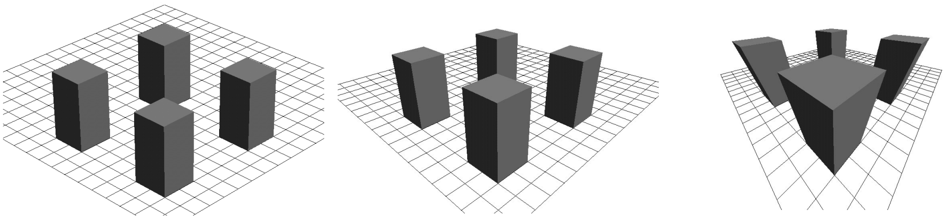


Image shot by a narrow angle (zoom), a regular angle and a wide angle camera lens

Unfortunately, information on the camera angle is not often available, and cropping the images causes the camera angle to be changed, too.

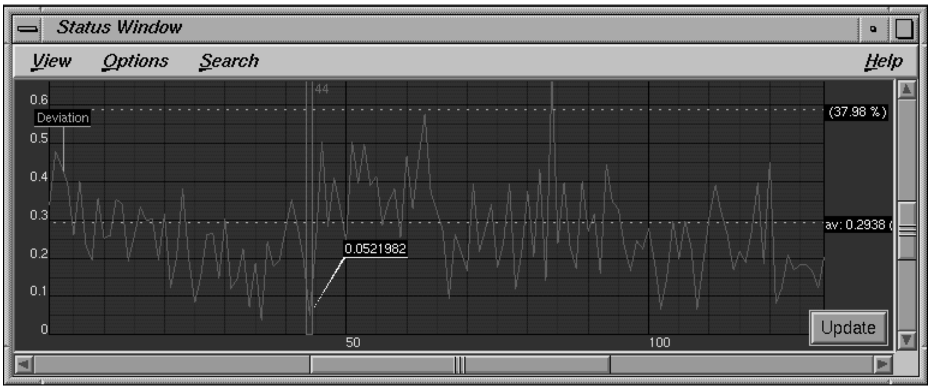
Therefore, it is often useful to let 3D-Equalizer establish the camera lens for you. In our example, we have no information about the camera lens. The image materal appears to be taken by a regular lens.

- *Set the option menu beside the horizontal angle textfield to Unknown. Check the Pixel Aspect textfield, and if necessary set its option menu to Fixed and enter a value of 1.0926.*

The camera is now prepared for the adjustment process.

- *Press the Adjust button, in order to start the adjustment process.*

Several messages will appear on the screen, indicating the current status of the calculation. 3D-Equalizer will try to calculate the parameters declared to be Unknown, then trim the parameters with setting Fine Adjust. You can terminate each level of calculation by pressing Cancel in the progress requester. The parameters yielding the best value up to that moment will then be used.



Status Window

3D-Equalizer finishes this adjustment process and calculates a horizontal angle of about 44.88°.

Now, please take a look at the Status Window. You see a green curve, which represents the reconstruction quality of each single frame. It expresses the deviation of the 2D motiontracked points from the reconstructed 3D points displayed in pixel. There is a dotted line with a black label "av:" and a number. This number represents the average deviation of the entire sequence. It should be around 0.35 pixel, which is already quite a good value.

In order to learn more about adjusting a camera please take a look at the tutorial "Camera Adjustment".

The project "ball_bluebox1.3de" found in the directory "<tutorial install directory>/ projects/", contains all steps we have taken up to now. Please feel free to review it at any time.

Error analysis using the Status Window

In the following section we will present how you can identify tracking errors by means of the Status Window.

- *Please open the project "ball_bluebox2.3de". You can find it in the directory "<tutorial install directory/projects/".*

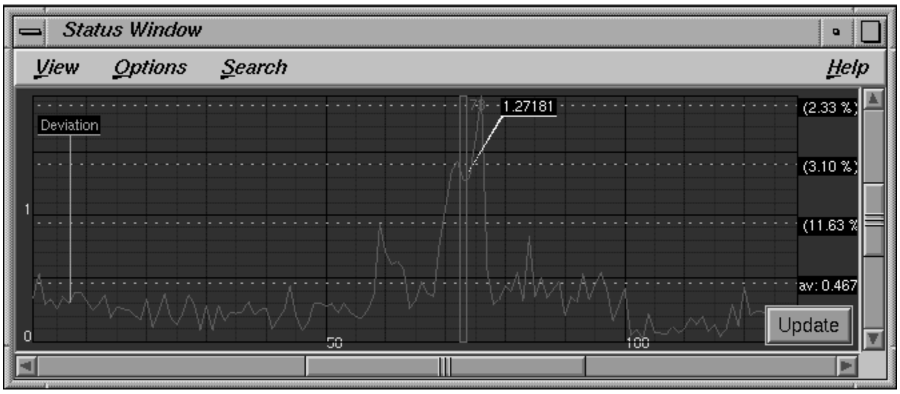
Basically, this project is similar to what we covered in the previous sections. There is a sequence object "ball_bluebox1" and a camera pointgroup "studio" with six points already tracked in the sequence. However, the tracking data are not so precise as the previous project, which will cause some problems.

- *Choose the menu item "Camera Adjustment" from the Main Window's Environments menu, and take a look at the Camera Adjustment Window.*

As you see, the camera is already well defined. The horizontal angle is set to 46.6, pixel aspect is set to 1.0926 and lens distortion is set to 0.06. So, it is not necessary to let 3D-Equalizer adjust the camera, you just have to calculate a 3D camera motion path out of the 2D tracking data.

- *Select the function Main Window::Calc::Calc All Objects from Scratch.*

3D-Equalizer starts the regular calculation process, which takes some time to complete. After the calculation has been finished, please take a look at the status window. There is a big peak in the green deviation curve. The average deviation value is 0.467, which is almost twice as high as in the previous section.



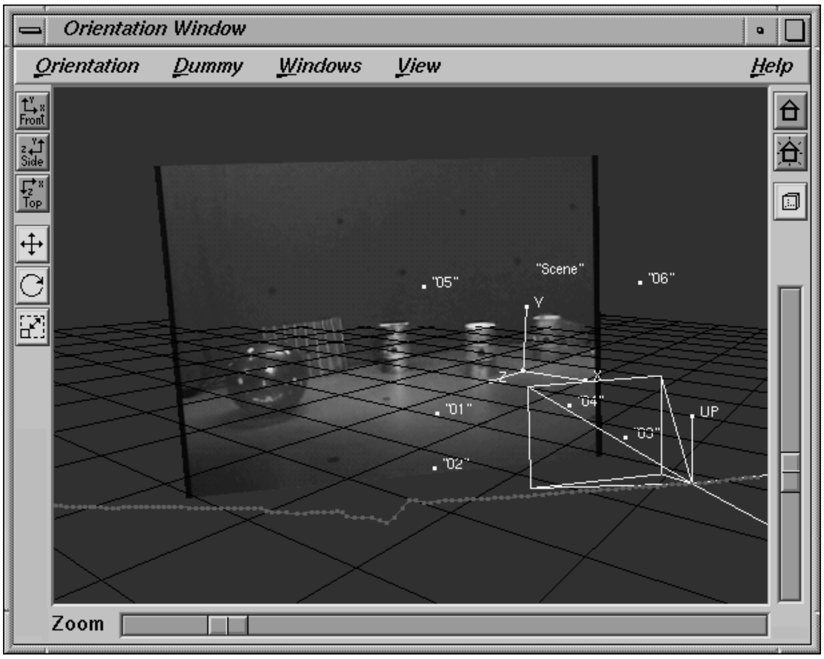
Peak in deviation curve

- *Select the “Orientation” environment from the Environments menu.*

A couple of windows are opened. In the middle of the desktop is the Orientation Window. It is used to display all kinds of three-dimensional objects such as cameras, points, motion paths, etc. The main purpose of this window is to allow you to align the reconstructed 3D data to the global coordinate frame and to edit simple 3D dummy objects. These dummy objects are used to render the preview movie. Now, let’s try to take a look at the reconstructed 3D camera motion path.

- *Activate the toggle button Orientation Window::View::Show Camera Path.*

A red line is displayed near the camera symbol, which represents the 3D camera motion path for the current sequence. It looks like a straight line, however in the middle there is a noticeable peak.



Peak in camera motion path

This peak is due to some tracking errors, which we will eliminate in the following.

- *Open the Status environment from the environments menu.*

Let’s become familiar with handling the Status Window.

- *Hold down the middle mouse button, and drag the mouse.*

The mouse cursor symbol is changed into a cross icon, and you are able to move around the contents of the Status Window’s display.

- *Hold down the right mouse button, and drag the mouse.*

As you see, the mouse cursor symbol is changed into a scale icon, and you are able to scale the contents of the Status Window’s display independently in both dimensions.

- *Select the menu item Status Window::View:Reset View.*

The 2D coordinate frame is automatically scaled and moved, so that it entirely fills the Status Window’s display area.

- *Hold down the CTRL-key, and click-drag the right mouse button in the middle of the display area.*

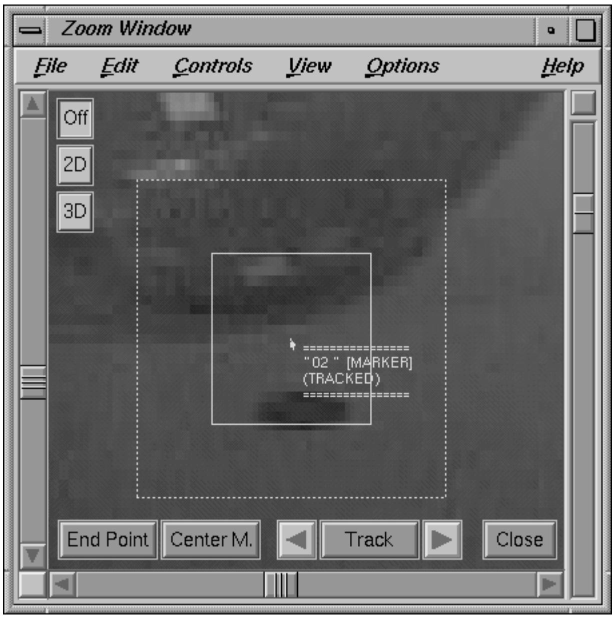
A rubber box is drawn. The area marked by the rubber box is zoomed in, so that it fills the entire display area.

- *Hold down the CTRL-key and click with the left mouse button into the display area.*

As you see, you are able to move the orange rectangle which represents the currently selected frame, to the click position. You are able to change the current frame within the Status Window.

- *Select the menu item Status Window::View::Reset View, hold down the CTRL-key and click with the left mouse button on the middle of the deviation curve's peak (approximately at frame 70).*

Now the current frame should be around frame 70. The peak in the deviation curve indicates that, in these frames, one or more points are probably not tracked precisely enough. We will now try to find these points, in order to improve their tracking curves.



Badly tracked point

- *Activate the toggle button Zoom Window::Options::Auto Center and zoom closer into the image.*

- *Move the mouse over the Zoom Window's display and press the Cursor Up-key several times. Check the tracking position of each point in the current frame.*

Each time you press the Cursor Up-key, the next point in the pointgroup is selected and automatically centered in the Zoom Window's display. All point positions seem to be ok, except for point “02” (see picture above).

- *Move the mouse over the Zoom Window's display and press the Cursor Left- and Cursor Right-key several times, in order to go back and forth through the sequence. Find out in which frames the tracking of point “02” is bad.*

As you see, between frame 58 and 76, the tracker of point “02” is confused by the blue ball which is located near to the point's marker at that time. It seems that the point's search pattern rectangle is too big, so that parts of the blue ball (and its shadow) are moving inside it, which causes problems. Now let's try to fix the tracking curve of point “02”.

- *Go to frame 57 and decrease the size of the inner tracking box (search pattern), so that the blue ball's shadow does not lie within it. Press the button Center M. several times to make sure that the tracker can clearly identify the marker's center.*

- *While the mouse cursor is over the Zoom Window's display, press the Cursor Right-key several times.*

As you see, the search pattern rectangle is increased more and more with each frame. The size of the tracking boxes is splined from keyframe to keyframe. By modifying the search pattern size in frame 57, you have designated this frame of the tracking curve as a keyframe. The search pattern of the next keyframe seems to be much bigger, and therefore the search pattern rectangle becomes bigger with each frame. To get rid of this problem, let's decrease the size of the next keyframe's search patten as well.

- *Press the button Key+ in the Main Window.*

The next keyframe of point “02”‘s tracking curve is searched, and becomes the current frame (frame 77).

- *Modify the search pattern of frame 77 as well, so that it has similar size as in frame 57.*

- *Press the button Track of the Zoom Window.*

The tracking procedure is started. 3D-Equalizer recognizes the areas of the tracking curve to be re-tracked. These areas are indicated by the Timeline Window in a light grey.

- *Watch point “02” being tracked and make sure that it is tracked correctly this time.*

- *After the tracking procedure has been completed, please take a look at the Status Window.*

As you see, the green deviation curve still indicates this big peak around frame 70 and the average deviation value is still 0.467 pixel. The deviation curves must be updated manually.

- *Please press the button Update in the Status Window.*

After a few seconds, the deviation curves are updated. The peak has gone, and the average deviation should be around 0.25.

- *Please open the "Orientation" environment and take a look at the 3D camera motion path.*

As you see, there is still a noticable peak in the camera motion path. It is necessary to update the camera motion path as well. Now it would be possible to execute the function Main Window::Calc::Calc All Objects From Scratch, like you did at the beginning of this section. However, this function takes some time to compute, because it calculates everything completely from scratch. We already have a more or less "good" camera motion path which only needs to be finetuned a little bit.

- *Select the menu item Main Window::Calc::Finetune Selected Pointgroup.*

Some messages appear on the screen. After only a few seconds, the camera motion path is updated, and the peak has gone.

> **Important note:** The functions to update deviation curves and three-dimensional motion paths are very useful, especially on projects with many frames and points. However, it is necessary to have at least a rough solution to operate on. If these functions fail, try the function "Calc All Objects From Scratch" instead.

Aligning the reconstructed 3D data to the global coordinate frame

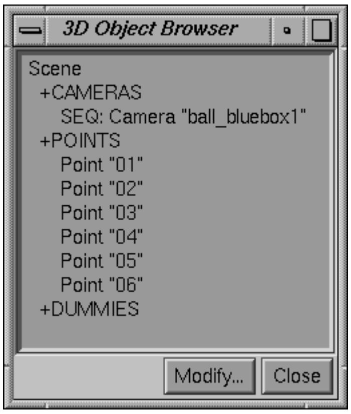
After successfully reconstructing the camera motion path, the next step is to align the resulting solution to the global coordinate frame. In principle it is not necessary to perform this step if you don't care about the camera's alignment in 3D space.

First of all let's get familiar with the handling of the 3D Object Browser Window.

- *Please open the project "ball_bluebox1.3de". from the directory "<tutorial install directory>/projects/".*

- *Select the "Orientation" environment from the Environments menu, and take a look at the 3D Object Browser Window on the right side of the desktop.*

The 3D Object Browser Window contains a list of all 3D objects displayed in the Orientation Window. It is mainly used for selection purposes.



3D Object Browser

Scene
+CAMERAS
SEQ: Camera "ball_bluebox1"
+POINTS
Point "01"
Point "02"
Point "03"
Point "04"
Point "05"
Point "06"
+DUMMIES

3D Object Browser Window

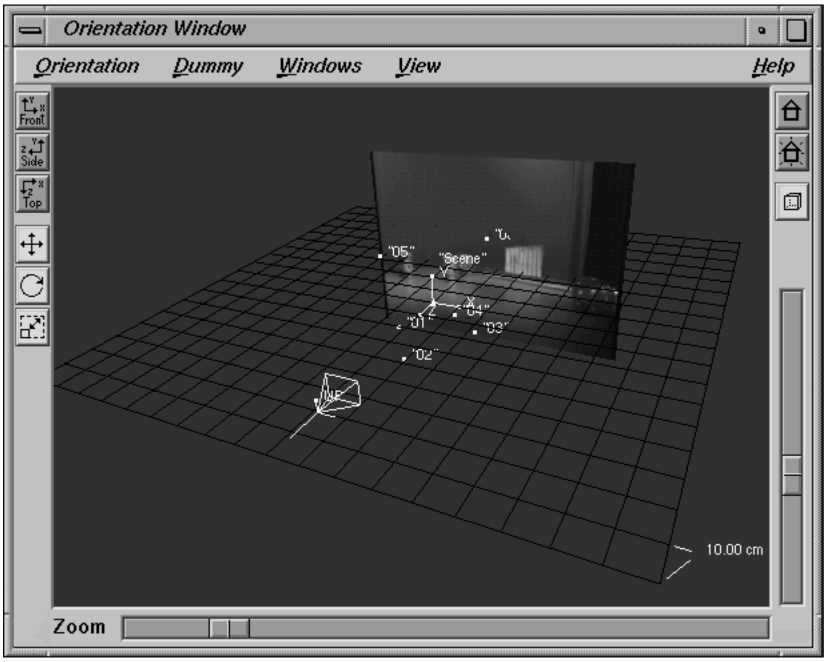
- *Click with the left mouse button on the Scene entry in the 3D Object Browser Window.*

The Scene entry becomes selected and the coordinate frame labeled “Scene” is drawn in red-green-blue in the Orientation Window, which indicates that it is selected as well.

This 3D object is called the “scene node”. It represents a coordinate frame under which all other relevant 3D objects are grouped. Aligning the 3D data to the global coordinate frame means transforming the scene node!

- *Select some other objects in the 3D Object Browser Window, and watch the Orientation Window’s display.*

Now, let’s take care of the Orientation Window’s handling.



Orientation Window

- *Hold down the Alt-key and click-drag the left mouse button.*

The mouse cursor changes into a rotation symbol. Dragging the mouse, you are able to rotate the view around the displayed 3D objects.

- *Click on the Top, Side and Front buttons found on the left border one after another, and watch the results in the Orientation Window.*

The 3D objects are shown from the top, from the right and from the front, respectively.

- *Hold down the Alt-key and click-drag the middle mouse button.*

The mouse cursor changes into a pan symbol. Dragging the mouse, you are able to pan the view in a horizontal or vertical direction.

- *Hold down the Alt-key and click-drag the right mouse button. If a menu should open up, please hold down the Shift- and Alt-key instead.*

The mouse cursor changes into a dolly symbol. Dragging the mouse, you are able to move into or out of the view.

- *Drag the vertical slider located on the right border of the Orientation Window.*

As you see, dragging this slider you are able to move into and out of the view, as well. Let's continue to modify an object.

- *Select the scene node in the 3D Object Browser Window. Activate the scale toggle button, located on the left border, and click-drag the left mouse button in the Orientation Window's display.*

As you see, by moving the mouse to the left and to the right, the scene node is scaled uniformly. Since all other objects are grouped under the scene node, they are scaled as well.

- *Activate the rotate toggle button, located on the left border, and click-drag the left mouse button in the Orientation Window's display.*

The scene node and all other relevant objects are rotated around its local x-axis.

- *Click-drag the middle or the right mouse button in the Orientation Window's display.*

The scene node is rotated around its local y-axis or its local z-axis, respectively.

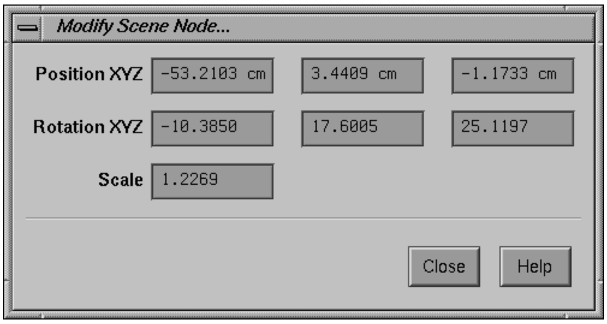
- *Hold down the Shift-key and click-drag the left mouse button in the Orientation Window's display.*

The scene node is rotated around the x-axis of the global coordinate frame.

- *Please play around with the remaining translate toggle button and the Shift-key.*

Using these mouse functions, you can modify every 3D object in the Orientation Window. In addition, there is another method to modify attributes of 3D objects.

- *Double-click on the scene node entry of the 3D Object Browser Window, or simply click on its Modify button.*



Modify Scene Node...

Position XYZ	-53.2103 cm	3.4409 cm	-1.1733 cm
Rotation XYZ	-10.3850	17.6005	25.1197
Scale	1.2269		

Requester: Modify Scene Node

The requester Modify Scene Node opens. It contains several textfields in which you can enter the scene node's position, rotation and scale as numerical values. This requester does not block 3D-Equalizer, so you can leave it opened as long as you want.

- *Modify some of the values, and watch the Orientation Window's display.*

- *Select the menu item Orientation Window::Orientation::Reset.*

As you see, the scene node's attributes are reset to their inital state. Now, let's continue to align the reconstructed 3D data to the global coordinate frame. First of all, we will specify the scale.

- *Double-click on the camera pointgroup entry "studio" in the Pointgroups Window.*

The requester Modify Pointgroup opens. It is similar to the Create Pointgroup requester discussed a few sections ago. After shooting the sequence, we measured the distance of two markers on the studio floor using a measuring tape. Between the markers now tracked with point "03" and point "04", we measured a distance of 114.6 cm. We will now define a so-called distance constraint, in order to fix the scale of the scene node.

- *Activate the toggle button "Distance Constraint Enabled". Click on the button "Point #1", and select point "03" from the list. Click on the button "Point #2", and select point "04" from the list. Enter "114.6 cm" in the Distance textfield, and press Ok to confirm your edits.*

As you already will have noticed, not only has 3D-Equalizer reconstructed a 3D camera motion path. Each tracking point's position in 3D space has been calculated as well. By defining a distance constraint between two points using a real-world measurement, the scale of the reconstructed 3D data becomes fixed and it is properly related to the real studio location. Looking at the Orientation Window's display, you will notice that the scene node and all objects grouped under it have been scaled up considerably. The scene node's scale has been automatically modified, so that the distance between the reconstructed 3D positions of point "03" and point "04" becomes exactly 114.6 cm.

- *Move out of the view using the Alt-key and the right mouse button, so that you can see all 3D objects again.*

- *Select the scene node in the 3D Object Browser Window. Activate the scale toggle button, located on the left border, and click-drag the left mouse button in the Orientation Window's display.*

As you see, you are no longer able to scale the scene node, because you have defined a distance constraint.

The black grid, which represents the xz-plane of the global coordinate frame, is quite small in respect to the remaining 3D objects. Let's modify its scale as well.

- *Select the menu item View::Set Grid Size.*

The requester Change Grid Size opens.

- *Enter "100cm" into the Grid Size textfield, and confirm your edits by pressing the Ok button.*

The black grid looks better now, however the camera and its imageplane have become much bigger. Let's fix that as well.

- *Select the camera entry "SEQ: Camera" in the 3D Object Browser Window. Activate the scale toggle button, located on the left border, and scale the camera down.*

As you see, the 3D camera object and its imageplane are scaled simultaneously.

Let's continue to align the reconstructed 3D data to the global coordinate frame. The markers tracked by the points "01"-"04" are all located on the studio floor, which we will assume now to be the black grid.

- *Select the menu item View::Center On Origin.*

The Orientation Window's view is moved so that the origin of the global coordinate frame is located in the center of the display. The origin is equal to the center of the grid.

- *Press the lowest button located on the right border of the Orientation Window.*

The icon of the button changes from a perspective cube symbol into an orthogonal cube symbol, which indicates that the Orientation Window's viewing mode is switched from perspective projection to orthogonal projection.

- *Press the button Side, located on the left border of the Orientation Window, and take a look at the points "01"-"04".*

The points "01"-"04" seem to lie in one plane. This plane however is not parallel to the grid, nor do any of the points lie on it.

- *Select the points "01", "03" and "04" in the 3D Object Browser Window and choose the menu item Orientation::Align 3 Points::XZ-Plane.*

The scene node is transformed, so that the plane formed by the selected points is oriented parallel to the xz-plane of the global coordinate frame (equal to grid).

- *Press the button Top, located on the left border of the Orientation Window, and take a look at the points "01"-"04" again.*

The points are not aligned to neither the x-axis nor the z-axis of the global coordinate frame (grid).

- *Select the points "01" and "04" in the 3D Object Browser Window and choose the menu item Orientation::Align 2 Points::X-Axis.*

The scene node is transformed, so that the axis formed by the two selected points is oriented parallel to the x-axis of the global coordinate frame.

- *Press the button Front, located on the left border of the Orientation Window, and take a final look at the points “01”-”04”.*

The points are parallel to the grid, but they are not located on it.

- *Select point “01” in the 3D Object Browser Window and choose the menu item Orientation::Move 1 Point::Origin.*

The scene node is transformed so that the selected point is located in the origin of the global coordinate frame.

- *Press the lowest button located on the right border of the Orientation Window.*

The viewing mode switches back to perspective projection.

- *Rotate the view around the objects, in order to examine the aligned 3D data.*

The points “01”-”04” are exactly located on the xz-plane of the global coordinate frame, which is represented by the black grid. The scene node is located near the origin. It seems to have rotated a bit around the three axis.

> **Important note:** Due to the nature of 3D-Equalizer’s application, there are many constraints between different 3D objects. For instance, the relationship between a 3D camera and the respective 3D points cannot be destroyed as result of a user action. Since the constraints are taken into account, editing a 3D object often results in changing the scene node.

Editing dummy objects

The second purpose of the Orientation Window is to create and edit simple three-dimensional test objects called “dummies”. They are used to give you an impression of the visual quality of your matchmove.

- *Please select the menu item Orientation Window::Dummy::Create::Cube.*

A blue cube appears in the middle of the black grid and in the Overview Window’s display. A new entry “Dummy Cube #1” is created in the 3D Object Browser Window, as well.

- *Hold down the Ctrl-key and click on the newly created dummy with the left mouse button.*

A red outline is drawn around the cube, indicating its selection state. The dummy’s entry in the 3D Object Browser Window has been selected as well.

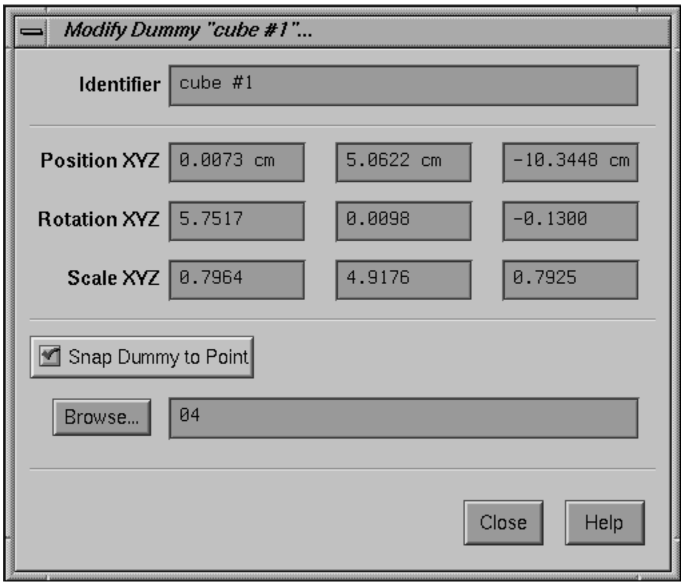
- *Transform the selected dummy with the mouse, using the toggle buttons located on the left border of the Orientation Window, as already learned in the previous section.*

As you see, you are able to transform the dummy independently from the scene node, so the scene node is not affected at all when a dummy is edited.

In order to get the best possible impression of the matchmove's quality, let's try to put the cube exactly onto the studio floor.

- *Make sure the dummy is selected in the 3D Object Bowser Window and click on the Modify button.*

The requester Modify Dummy opens. It contains textfields for editing the dummy's attributes.



Modify Dummy "cube #1"...

Identifier: cube #1

Position XYZ	0.0073 cm	5.0622 cm	-10.3448 cm
Rotation XYZ	5.7517	0.0098	-0.1300
Scale XYZ	0.7964	4.9176	0.7925

☑ Snap Dummy to Point

Browse... 04

Requester: Modify Dummy...

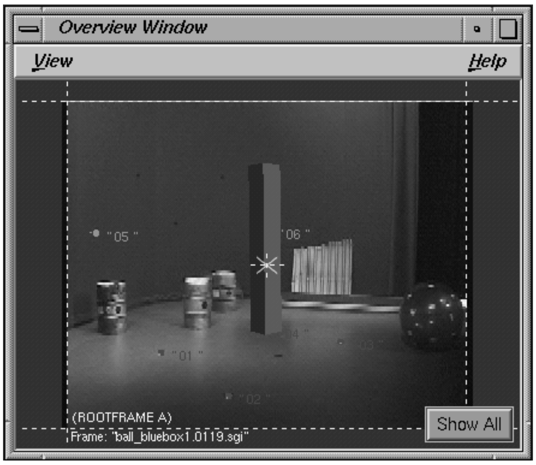
- *Activate the toggle button "Snap Dummy to Point", click on the Browse button and select point "04" from the list. Then, press the button Close to close the requester.*

The cube's position is now constraint to the reconstructed position of point "04", and you are not able to move it in 3D space any longer.

- *Select the menu item Dummy::Reset All.*

The dummy's rotation and scale are reset and it becomes aligned to the grid.

- *Activate the Scale toggle button on the left of the Orientation Window's border and scale the dummy so that it becomes a thin, tall column (see picture below).*



Dummy object

Let's get a first impression of the matchmove.

- *Drag the frame slider of the Main Window back and forth, and watch the Overview Window.*

You can also synchronize the Orientation Window's view to the reconstructed 3D camera object, which is sometimes useful while editing the dummies.

- *Activate the toggle button Orientation Window::View::Camera View, drag the Main Window's frame slider back and forth and watch the Orientation Window's display.*

The Orientation Window's view is "snapped" to the reconstructed 3D camera position and rotation each time the current frame changes. Now, let's create an additional dummy.

- *Make sure the recently created dummy is selected. Select the menu item Dummy::Copy.*

The dummy is copied to the clipboard.

- *Select the menu item Dummy::Paste.*

A new cube shaped dummy is created and displayed in the Orientation and Overview Window, similar to the one you edited before.

- *Please make sure the newly created dummy is selected in the 3D Object Browser Window, and press the Modify button. In the opened Modify Dummy requester align the dummy to point “02” and close the requester by pressing the button Close.*

Generating a preview movie

Having edited a few dummy objects, we can now generate a preview movie in order to see, whether the camera has been successfully reconstructed.

- *Please open the Preview environment from the Main Window’s environments menu.*

- *Make sure the toggle button Main Window::Playback::Playback Output::Preview Movie is activated and press the button Play.*

3D-Equalizer starts to generate the preview movie. After all frames have been rendered, the preview is played back in realtime (if possible) giving a good impression of the matchmove’s quality. The dummy objects are simply drawn on the image. If you take a look at the edges of the dummies, you can see some aliasing artefacts.

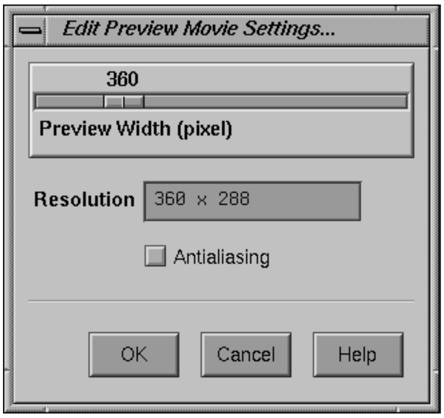
> **Important note:** For better performance, the preview movie is rendered directly on the screen. Please make sure that no other window is located over the Overview Window while the movie is being rendered.

- *Press the button Stop.*

The preview playback stops.

- *Select the menu item Options::Preview Settings.*

A requester Edit Preview Movie Settings opens.



Edit Preview Movie Settings...

360

Preview Width (pixel)

Resolution 360 x 288

□ Antialiasing

Requester: Edit Preview Movie Settings...

- *Activate the toggle button Antialiasing, and confirm your edits by pressing the Ok button.*

You now have enabled the antialiasing render mode.

- *Press the button Play again.*

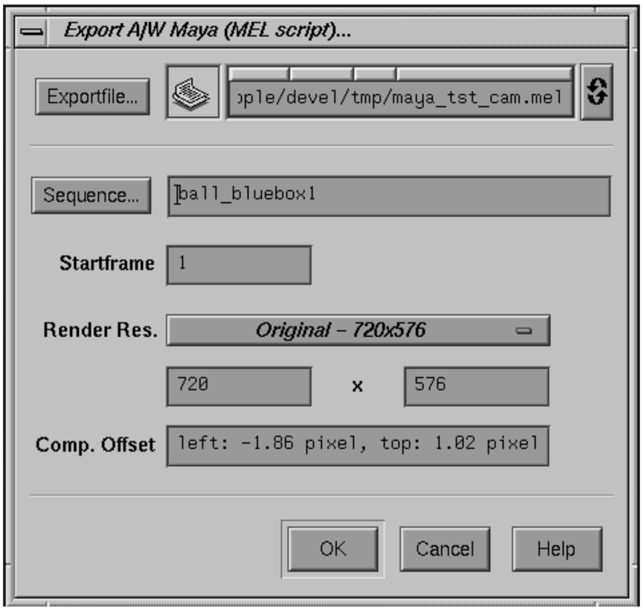
3D-Equalizer starts generating the preview movie again. The rendering of the two dummy objects looks much better now. The antialiasing render feature enables you to better identify the quality of your matchmoves.

The project “ball_bluebox3.3de” found in the directory “<tutorial install directory>/projects/”, contains all steps we have done up to now. Please feel free to review it.

Exporting the reconstructed 3D data

Usually the final step of 3D-Equalizer’s workflow is exporting the reconstructed 3D data, 3D camera motion path and 3D point positions, to your preferred animation package. In the current release of 3D-Equalizer the packages A|W PowerAnimator, SoftImage/XSI, Discreet Logic Flint/Flame/Inferno, NewTek Lightwave, 3D Studio Max 3, A|W Maya and Kodak Cineon are supported.

Please open the corresponding export requester for your preferred animation package by selecting Main Window::Project::Export.



Export A/W Maya (MEL script)...

Exportfile... ɔple/devel/tmp/maya_tst_cam.mel

Sequence... ball_bluebox1

Startframe 1

Render Res. Original – 720x576

720 x 576

Comp. Offset left: -1.86 pixel, top: 1.02 pixel

Requester: Export A|W Maya...

The mode of operation of the respective export requester is described in the chapter “Reference Manual”, section “Main Window”.

Important note: The wire frame representation of your models in the respective animation package often does not take into account the pixel aspect, and therefore in general the real and the reconstructed points will not coincide in this representation mode.

Tutorial 2: Automatic Motiontracking

In the following section 3DEqualizer's capabilities for automatic feature tracking are demonstrated. For this purpose, we will matchmove the image sequence of the previous tutorial one more time. Several specific aspects of automatic tracking, such as autotracking regions, elimination of badly tracked points and spline area mattes, will be discussed.

- *Please start 3D-Equalizer as described in the section "Getting started" and select the menu item Main Window::Projects::New to create a blank project.*

First of all, we will setup our project for the automatic tracking procedure.

- *Select the environment "Motiontracking" from the Main Window's environments menu.*

- *Please click on the Add button of the Frames Window.*

A requester opens and asks you whether you want to create a sequence or a reference frame.

- *Click on the "Add Sequence" button to create a new sequence object.*

The Create Sequence requester opens, which allows you to define all the settings associated to the new sequence object.

- *Press the button Sequence Settings::Browse and go to the directory "<tutorial install directory>/ball_bluebox1/". Double-click on the first image file "ball_bluebox1.0119".*

3D-Equalizer will scan the directory, starting with the selected file, to determine the number of image files associated to the sequence. After that, the Identifier, Pattern, Start and End textfields are filled with well-defined values.

The tutorial sequence is interlaced PAL video at a frame rate of 25 frames per second.

- *Please enter a value of 25.0 in the Rate Fps textfield, set the Imagetype option menu to Interlace - Odd First and press the Ok button, to create the new sequence object.*

Now let's modify the field of view (FOV), as you already did in the previous tutorial.

- *Go to frame 80 and adjust all four FOV lines in 3DE's Zoom Window as precisely as you can. To move a FOV line, hold down the CTRL-key and click-drag the left mouse button.*

Finally, we need to create a new pointgroup object that will be filled with new points by 3DEqualizer's autotracking engine.

- *Select the menu item Pointgroups Window::Pointgroup::Add Pointgroup, and simply confirm the Create Pointgroup requester that opens, by clicking on the button Ok.*

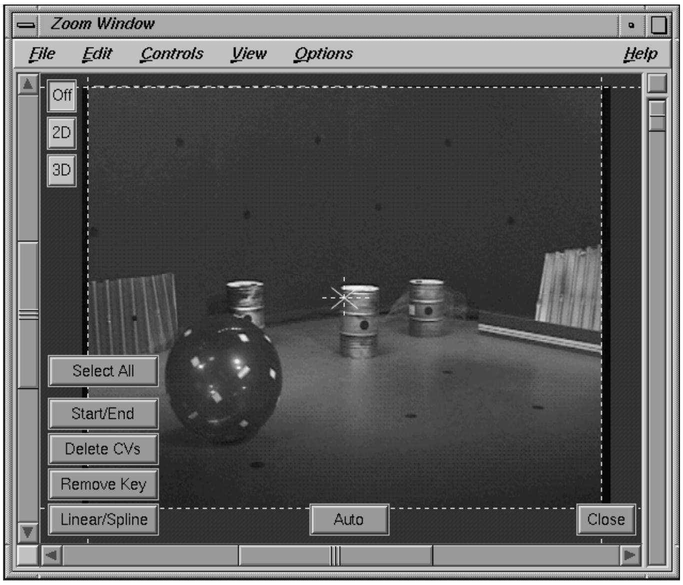
A camera pointgroup object named "pgroup #1" is created, and a new entry, which represents it, appears in the Pointgroups Window (see also project "auto_track1.3de").

Basic concepts of automatic tracking

Our project is now prepared for automatic tracking. Before we actually start tracking, let’s have a look at 3DE’s autotracking related user interface functions.

- *Select the environment “Autotracking” from the Main Window’s environments menu.*

The Timeline-, Frames-, Pointgroups-, Zoom- and Autotracking Regions Windows open up. In addition, the “Autotracking Controls” of 3DEqualizer’s Zoom Window are enabled (see figure below).



Zoom Window with Autotracking Controls enabled

3DEqualizer’s autotracking engine can be configured due to the functions provided by its Autotracking Regions Window. This window contains a list of so-called “autotracking region” objects, which are associated to the currently selected sequence. An autotracking region usually defines a two dimensional area of the FOV, inside of which 3DEqualizer tracks points during the automatic tracking procedure. A two dimensional area can be defined by a matte image sequence or by a spline area matte or it can simply be the entire FOV. Each autotracking region has its own set of parameters and thus can be configured separately.

There is already a default autotracking region object in the Autotracking Regions Window. So, let’s examine its parameter settings.

- *Make sure autotracking region "region #1" is selected and click on the button Autotracking Region Window::Modify.*

The requester Modify Autotracking Region opens (see figure below). In this requester all settings of a given autotracking region can be modified.

The option menu Tracking Mode, allows you to specify which autotracking engine (marker or pattern) is going to be be used.

Points created by an autotracking region are stored in a pointgroup object. You can link an autotracking region to a specific pointgroup by using the button Pointgroup.

To speed up autotracking of high resolution image sequences, you can reduce image size by modifying option menu Image Scale.

3DEqualizer's autotracking procedure can be subdivided into three different steps. In step A, 3DE searches for good features and creates a new point object for each good feature found. Each point, created during step A, is tracked frame by frame, as long as it remains "active" (step B). Finally, 3DE decides which of the active points should no longer be tracked, and thus become "inactive" (step C). Steps B and C are always performed for every frame. However, to improve performance, Step A is usually done every 15th frame. You can configure this behaviour by modifying the parameter Search Interval. During step A, 3DE tries to find enough features so that 60 active points (including already existing ones) become available. If it turns out that in a frame only less than 30 active points are available, 3DE is forced to perform step A in order to "fill up" the set of active points. You can configure this behaviour by modifying the parameter # Points (min/max). Finally, inactive points that are tracked for less than 10 frames are deleted, because 3DE usually cannot benefit from them. This behaviour can be configured by parameter Min. Track Length.

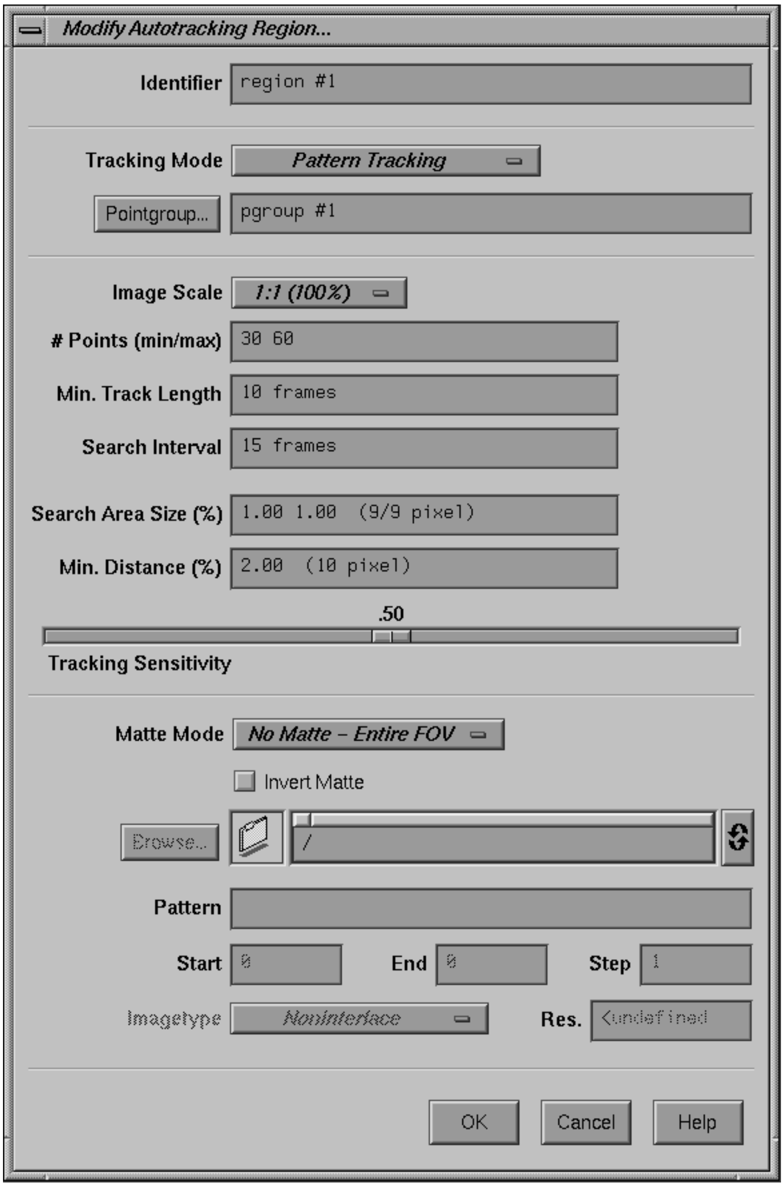
The size of features to be tracked, is defined by parameter Search Area Size. Active points that move too close together during tracking, become inactive automatically. You can configure the allowed minimum distance by the parameter Min. Distance.

The parameter Tracking Sensitivity allows you to define, how soon a feature is declared to be lost. A value of 1.0 will force 3DE to lose features very soon, thus generating points with relatively high quality, but rather short tracking curves (low number of frames). A value of 0.0 will result in points with relatively low quality, but rather long tracking curves (high number of frames).

Each autotracking region has its own two dimensional area of the FOV inside which points are tracked during autotracking procedure. The characteristics of such an area can be configured by the lower user interface elements starting with the option menu Matte Mode. Generally, an area can be defined by an image sequence, a spline matte, or it can be the entire FOV.

The default settings of autotracking region #1 are already very good, so let's start tracking.

- *Press the button Ok, to close the Modify Autotracking Region requester.*



Modify Autotracking Region...

Identifier: region #1

Tracking Mode: Pattern Tracking

Pointgroup...: pgroup #1

Image Scale: 1:1 (100%)

Points (min/max): 30 60

Min. Track Length: 10 frames

Search Interval: 15 frames

Search Area Size (%): 1.00 1.00 (9/9 pixel)

Min. Distance (%): 2.00 (10 pixel)

Tracking Sensitivity: .50

Matte Mode: No Matte – Entire FOV

□ Invert Matte

Browse...: /

Pattern:

Start: 0 End: 0 Step: 1

Imagetype: NonInterlace Res.: <undefined

Requester: Modify Autotracking Region...

- *Click on the button Zoom Window::Auto.*

3DEqualizer's automatic tracking procedure is started, and the button label is changed into "Stop". Several red points appear in the FOV while 3DE advances through the sequence frame by frame. After a while, some red points turn yellow, which indicates that their tracking curves have now reached the minimum track length value (10 frames). From time to time, the FOV is filled up with new (red) points. In order to keep track of what's going on, let's configure the display.

- *Disable the toggle menu item Zoom Window::View::Show Point Names, without stopping the automatic tracking procedure.*

Now, each point is only represented by a small, red or yellow colored dot, which allows you to get a better overview. Finally, 3DEqualizer reaches the last frame. The button label changes back to "Auto". The Pointgroups Window indicates that 101 points have been automatically created. Let's have a look.

- *Press the button Main Window::Begin in order to go to the first frame. Use the Main Window's frame slider to go through the sequence.*

You will see several "badly" tracked points, mainly located on or around the blue ball. Nevertheless, let's try to calculate a 3D solution. In order to do so, we have to define some camera parameters, first.

- *Select the environment "Camera Adjustment" from the Main Window's environments menu.*

From the previous tutorial we already have a good idea about the correct camera values for this sequence. To speed things up, let's enter them manually.

- *Set Horizontal Angle to 45.0, Pixel Aspect to 1.0926 and Lens Distortion to 0.04.*

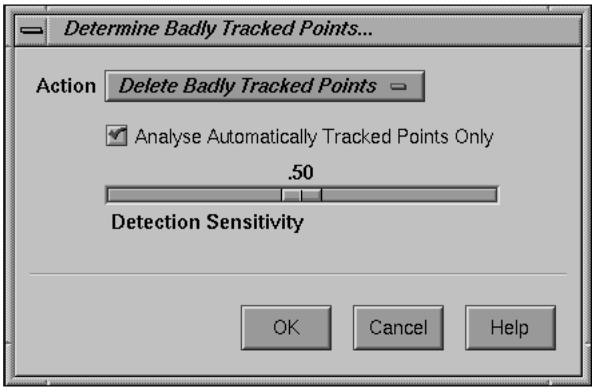
- *To calculate a 3D solution, please select the menu item Main Window::Calc::Calc All Objects From Scratch.*

3DEqualizer starts the regular calculation process to compute a 3D camera motion path and 3D positions for all tracking points. After some time the calculation aborts because a number of errors occurred. Obviously, the badly tracked points confuse 3DEqualizer, so we must get rid of them. We could search for these bad points frame by frame and delete them manually, but there is a more elegant way.

- *Select the menu item Main Window::Calc::Determine Badly Tracked Points.*

A small requester opens (see figure below). As its name already indicates, this function will search for points that seem to be badly tracked.

- *Set option menu Action to Select Badly Tracked Points, increase Detection Sensitivity to 1.0 and click on the button Ok to start the determination process.*



Determine Badly Tracked Points...

Action: Delete Badly Tracked Points

☑ Analyse Automatically Tracked Points Only

Detection Sensitivity: .50

Requester: Determine Badly Tracked Points...

After some time, 3DEqualizer has finished the determination process, and several points have become selected. Let’s have a look at the result.

- *Please select the environment “Motiontracking” from the Main Window’s environments menu.*

- *Press the button Main Window::Begin in order to go to the first frame. Use the Main Window’s frame slider to go through the sequence.*

As you see, most of the selected points are located on or around the blue ball. Let’s get rid of them.

- *Press the button Pointgroups Window::Delete, and confirm the security requester by clicking on the button Delete Selected Points.*

The Pointgroups Window indicates that approximately 80 points have survived. Now, let’s try to calculate a 3D solution again.

- *Please select the menu item Main Window::Calc::Calc All Objects From Scratch.*

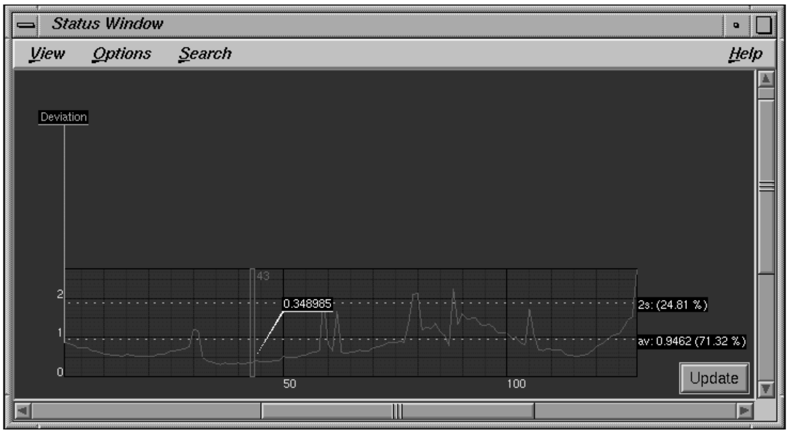
This time, 3DEqualizer is able to compute a 3D camera motion path and 3D point positions, without any problems.

> **Important note:** Under certain circumstances it may happen that the function “Determine Badly Tracked Points” finds not enough or even too many “bad” points. In these cases, modify the Detection Sensitivity value and/or apply the function several times, until you get a satisfying result.

Let’s examine the result more in detail.

- *Select the environment "Status" from the Main Window's environments menu and examine the average deviation ("av") value.*

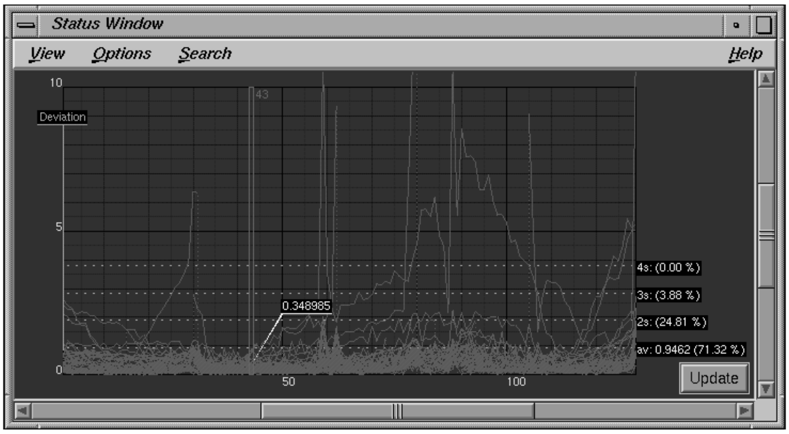
You will notice that the average deviation is about 0.95 pixel (see figure below), which is about 4 times higher compared to the value achieved in the previous tutorial.



Status Window displaying average deviation curve

- *Please enable the toggle menu item Status Window::View::Show Point Deviations::All.*

The deviation curves of all points will now be displayed at once (see picture below). As you see, the deviation curves of a small number of points is very high in comparison to the average deviation curve. These points seem to be "bad", as well. So, we should get rid of them.



Status Window displaying deviation curves of all points

- *Hold down the ALT-key and draw a rubberbox from the upper left corner of the Status Window's display area to the right side by click-dragging the left mouse button. Make sure that the bottom edge of the rubberbox does not reach below the "3s:" dotted white line (second line from above).*

As you might have noticed, you have selected those points which have a very high deviation curve.

- *Press the button Main Window::Begin in order to go to the first frame. Use the Main Window's frame slider to go through the sequence while watching the Zoom Window.*

Each selected point performs an unintended movement in a number of frames. Either it is "carried away" with the blue ball, or it jumps from one feature to another. Simply delete these points, because they falsify the entire result.

- *Press the button Pointgroups Window::Delete, and confirm the security requester by clicking on the button Delete Selected Points.*

The Pointgroups Window indicates that approximately 70 points have survived. Let's check the result of this action, again.

- *Please select the menu item Main Window::Calc::Calc All Objects From Scratch, and examine the average deviation value displayed in 3DEqualizer's Status Window after the calculation has been finished.*

The average deviation has been greatly improved to a value of 0.39 pixel, which is already an acceptable figure.

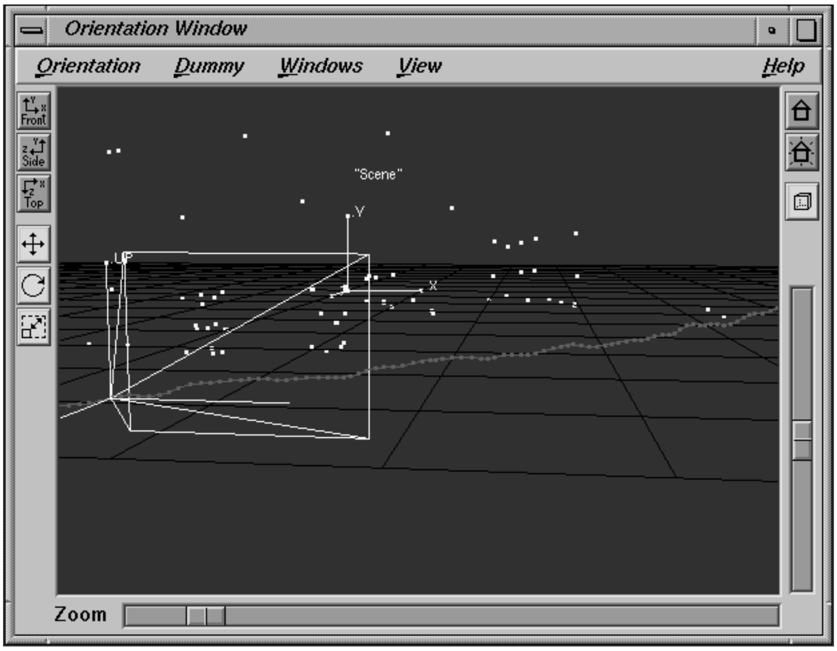
Techniques for refining automatically tracked projects

In the previous section we discussed basic concepts for using 3DEqualizer's automatic tracking features. Now we shall learn how to further improve the result.

- *Please select the environment "Orientation" from the Main Window's environments menu.*

- *Disable the toggle menu item Orientation Window::View::Show Point Names, enable the toggle menu item Orientation Window::View::Show Camera Path, and take a look at the 3D camera motion path.*

All in all, the 3D solution seems to be ok, however the camera motion path looks a bit bumpy which it wasn't in reality (see figure below). The reason is that all points, 3DEqualizer's autotracking engine has generated, are more or less located in the background of the bluebox studio. There are a few markers in the foreground, but they appear to be displayed with low constrast. We will learn in one of the following tutorials that points tracked in the foreground, can significantly improve the quality of a matchmove.



Bumpy camera motion path

For now, let’s add a manual tracking point in order to stabilize the camera motion path.

- *Please select the environment “Motiontracking” from the Main Window’s environments menu.*

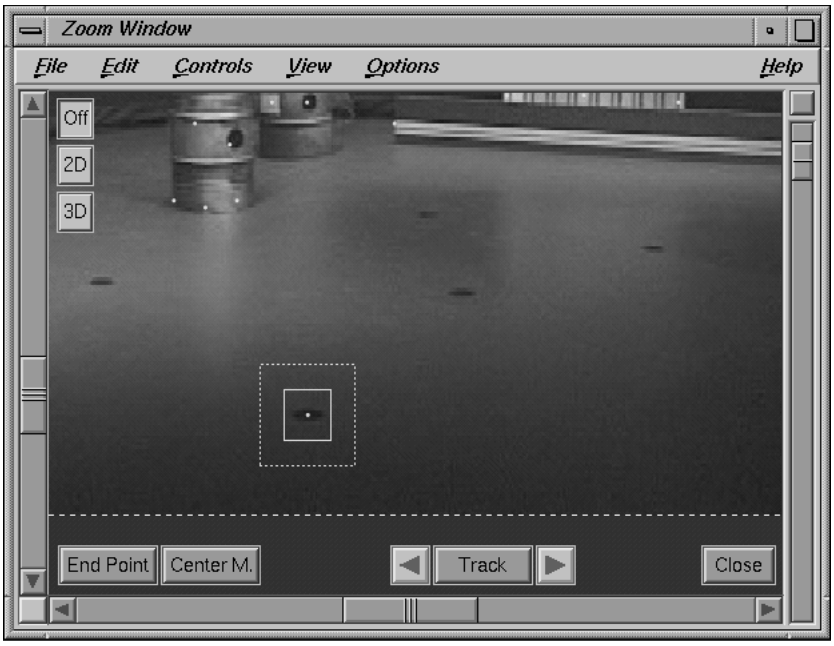
- *Click on the button Pointgroups Window::Add to create a new point, named “01”.*

The requester Create Point opens.

- *Set option menu Trackingmode to Marker and confirm your edits by clicking on the button Ok.*

For a maximum effect you should track the marker which is closest to the camera.

- *Make sure the newly created point “01” is selected, go to frame 1 and click with the left mouse button on the bottommost marker (see figure below). After that, press the button Zoom Window::Center M. several times to make sure that 3DE is able to identify the marker clearly, then enable the toggle menu item Zoom Window::Options::Auto Center and finally click on the button Track.*



Tracking a marker manually

As you see, 3DEqualizer tracks the marker throughout the entire sequence without any flaws. Let's examine the impact of this additional point on the resulting matchmove.

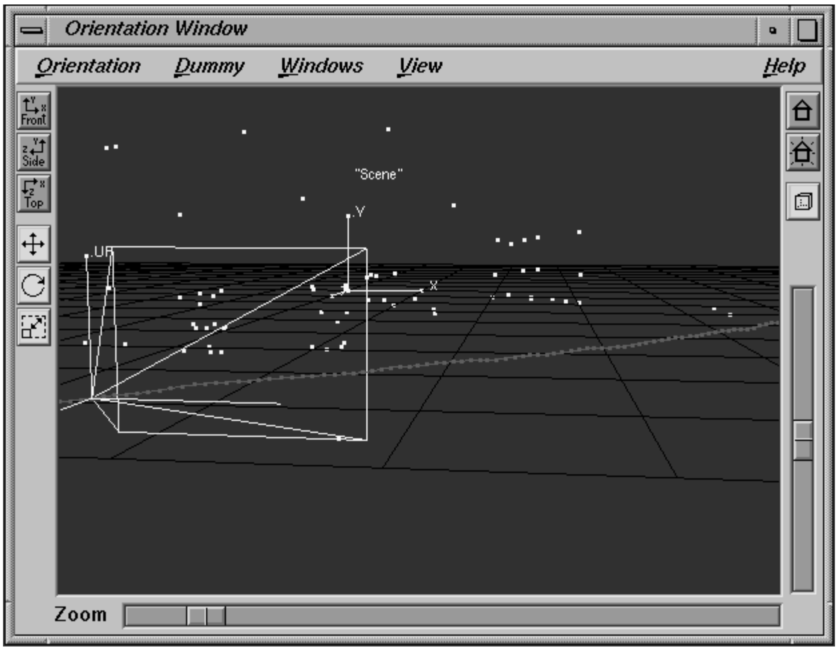
- *Please select the menu item Main Window::Calc::Calc All Objects From Scratch, then select the environment "Status" from the Main Window's environments menu and examine the average deviation value.*

The average deviation is still around 0.39 pixel and thus hasn't really improved.

- *Select the environment "Orientation" from the Main Window's environments menu and examine the camera motion path again.*

The camera motion path looks much smoother now (see figure below). Our manually tracked "stabilizing" point has indeed improved the matchmove significantly.

Important node: The previous steps are a good example for a typical workflow that can happen when matchmoving a shot using automatic tracking. We do not understand 3DEqualizer's autotracking engine as a "fire and forget" solution. It is a great tool to create many tracking points in short time, but it is not meant to replace your capability of solving problems in a creative way.



Smooth camera motion path

- *Select the environment “Status” from the Main Window’s environments menu and make sure the toggle menu item Status Window::View::Show Point Deviations::All is enabled.*

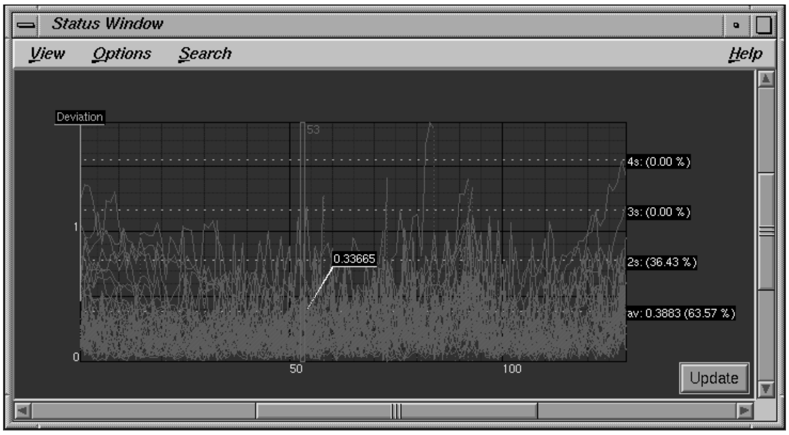
The deviation curves of some points are still relatively high in comparison to the average deviation curve (see figure below). Let’s see what happens if we delete these points, as well.

- *Hold down the ALT-key and draw a rubberbox from the upper left corner of the Status Window’s display area to the right side by click-dragging the left mouse button. Make sure that the bottom edge of the rubberbox does not reach below the “3s:” dotted white line (second line from above).*

All deviation curves that were located within the rubberbox turn orange, which indicates that the corresponding point objects are now selected.

- *Press the button Pointgroups Window::Delete, and confirm the security requester by clicking on the button Delete Selected Points.*

The Pointgroups Window indicates that approximately 60 points have survived. Finally, let’s check the result of this action one more time.



Status Window displaying deviation curves of all points

- *Please select the menu item Main Window::Calc::Calc All Objects From Scratch, and examine the average deviation value displayed in 3DEqualizer's Status Window after the calculation has been finished.*

The average deviation has been improved to a value of approximately 0.30 pixel, which is a good result (see also project “auto_track2.3de”).

How to use Spline Area Mattes

We will now demonstrate how to use spline area mattes in automatically tracked projects. A spline area matte is always associated to exactly one autotracking region. It usually consists of a set of control vertices (“CVs”), which precisely define its outline. Each CV can be animated using simple keyframing techniques. During the automatic tracking procedure, points are only tracked inside or outside of a spline area matte, depending on the spline area's configuration. In other words, by means of spline area mattes 3DEqualizer can be forced to track points only in certain areas of the FOV.

In the following section we will autotrack the already well known sequence “ball_bluebox1”. This time, we will use an animated spline area matte that “covers” the blue ball, so that 3DEqualizer is forced not to generate any “bad” points on it.

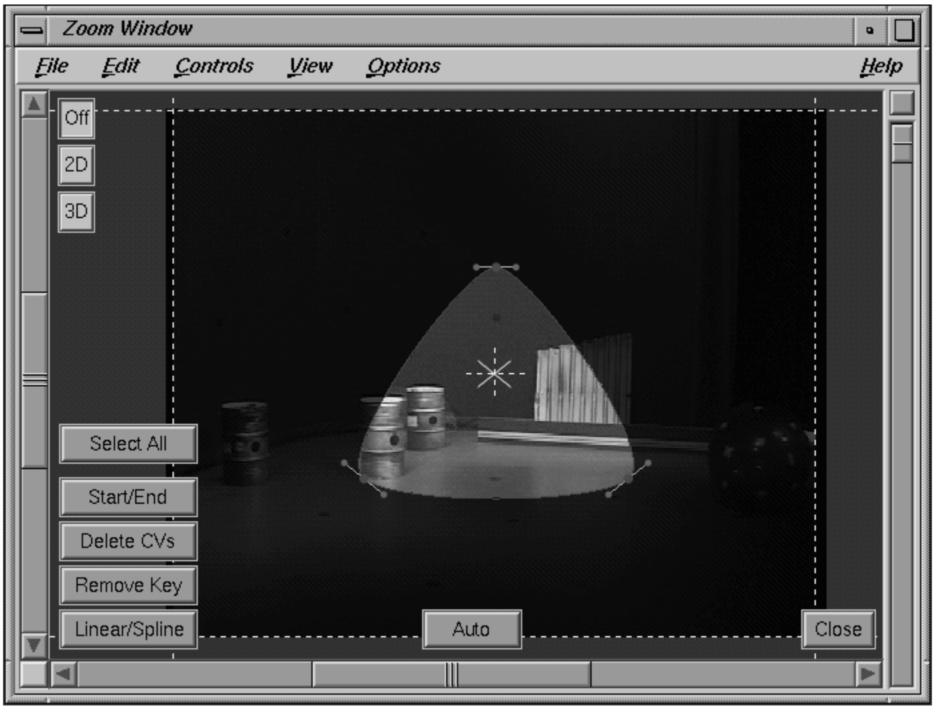
- *Please open the project “auto_track1.3de” located in the directory “<tutorial install directory>/projects/”.*

- *Select the environment “Autotracking” from the Main Window's environments menu.*

The project only consists of a sequence object and an empty camera pointgroup object. The sequence is already setup with images of the “ball_bluebox1” shot. Finally, the FOV has been adjusted to fit the black borders so that we can instantly start to create a spline area matte.

- *Double-click on the “region #1” entry of the Autotracking Regions Window to open the Modify Autotracking Regions requester. Set option menu Matte Mode to “Spline Area matte”, and confirm your edits by clicking on the button Ok.*

In 3DEqualizer’s Zoom Window you will see three, red-colored CVs, connected by a red line, which form a rounded triangle. The part of the frame, located outside of the triangle is displayed much darker than usually (see figure below). Dark parts of the frame indicate areas in which 3DE will be forced not to track any points during the automatic tracking procedure.



Default spline area matte

The idea is to place the spline area matte over the ball throughout the sequence, in order to force 3DE to track points only outside of it. To prevent 3DE from tracking any points on the blue ball, it makes sense to invert the effect of the spline area matte so that the inner part of the triangle becomes dark. Let’s modify the autotracking region to achieve this.

- *Double-click on the “region #1” entry of the Autotracking Regions Window to open the Modify Autotracking Regions requester again. Enable the toggle button “Invert Matte”, and confirm your edits by clicking on the button Ok.*

As you see, the mask becomes inverted. Now the inner part of the triangle, formed by the CVs, is displayed much darker.

A spline area matte is constructed out of its CVs. To edit/animate a spline matte its CVs have to be edited/animated. So, let’s get familliar with this business.

- *Click on a CV with the left mouse button.*

The CV becomes yellow, which indicates that it is now selected.

- *Click on a different CV with the left mouse button.*

The previously selected CV becomes unselected, and the newly clicked CV becomes selected.

- *Hold down the Shift key and click on different CVs multiple times.*

Each time you click on a CV, its selection state is reversed. That means, formerly selected CVs become unselected and vice versa.

- *Hold down the Alt key and click-drag the left mouse button to span a rubberbox.*

CVs located inside the rubberbox become selected, CVs located outside the rubberbox become unselected.

- *Click-drag a CV with the left mouse button.*

As you see, you are able to move the CV around. Once you release the mouse button, the mask (dark area) is updated.

- *Hold down the Shift key, click on a CV to select it, then click-drag on a different CV.*

Both CVs are moved around.

- *Click-drag a CV’s green control point with the left mouse button.*

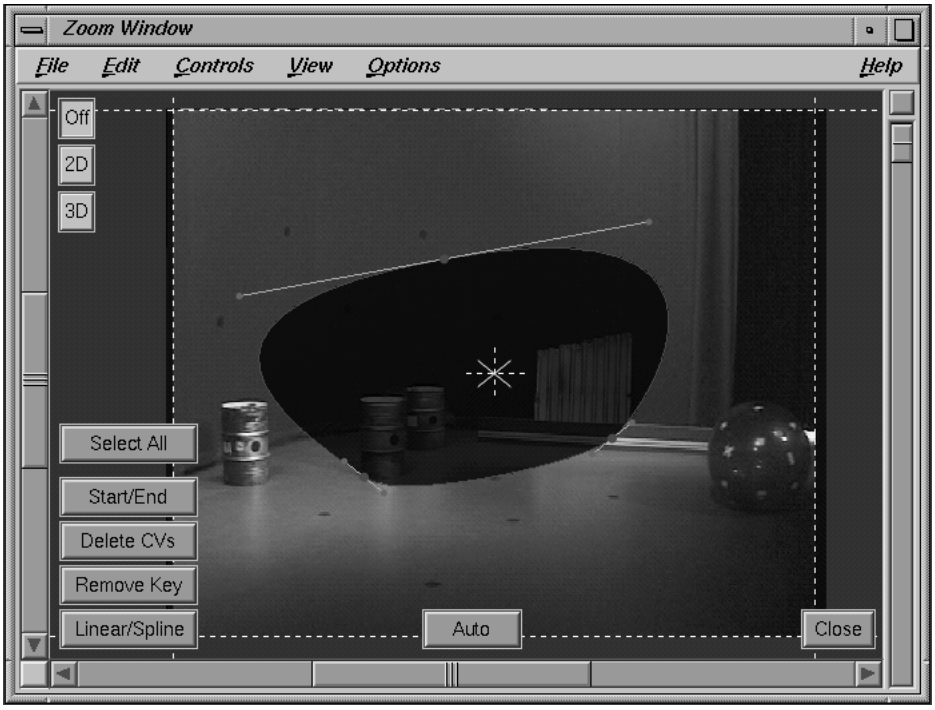
You are able to move the control point around, which changes the tangent of the respective CV and thus the shape of the spline matte (see figure below).

- *Click on a CV with the left mouse button to select it. Then, press the button Zoom Window::Delete CVs.*

The selected CV is deleted.

- *Hold down the Ctrl key and click inside the FOV several times.*

Each time you click inside the FOV, a new CV is created. It is automatically inserted between those two CVs that are located nearby.



Modifying a CV’s tangent

Now, that you have become famillar with 3DE’s basic CV editing functions, it’s time to animate a spline area matte that covers the blue ball.

- *Press the button Zoom Window::Select All.*

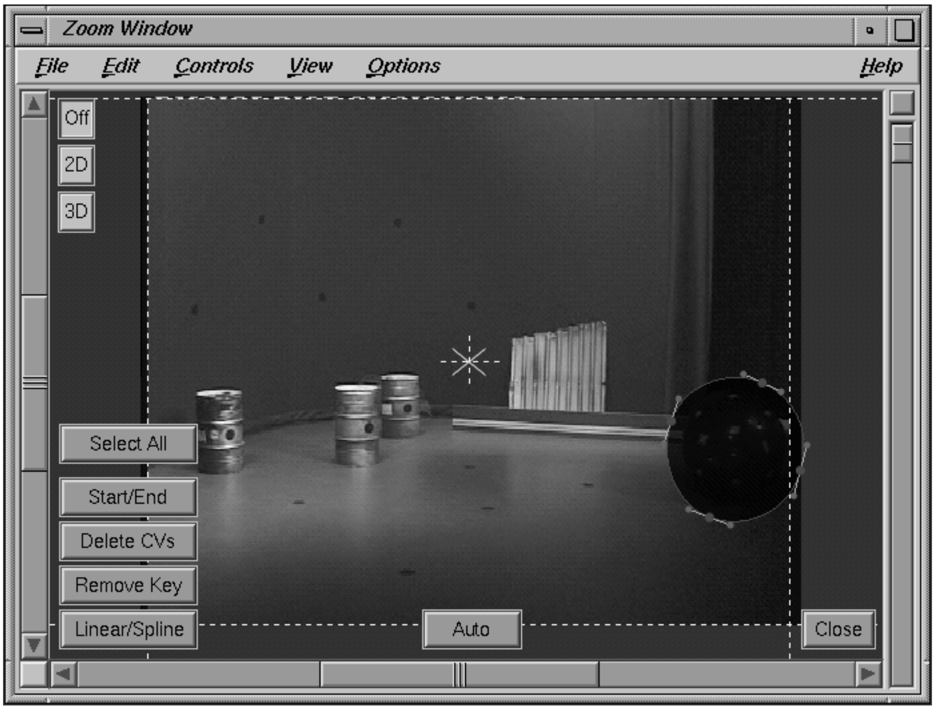
All CVs become selected.

- *Press the button Zoom Window::Delete CVs.*

All CVs are deleted, and thus no dark area will be drawn at all.

- *Create 4 CVs by holding down the Ctrl key while clicking four times inside the FOV. Make sure to distribute the CVs loosely around the blue ball. Modify the CV’s tangents so that a roundish spline shape is created.*

In frame 1 the blue ball is now covered by our newly created spline matte (see figure below). Let’s start to animate it.



Blue ball covered by spline area matte

- *Go to frame 10 using the Main Window’s navigation functions.*

You will notice that the spline matte’s outline, CVs and control turn white. This indicates, that the spline matte is not keyframed in the current frame. In addition, the blue ball has moved outside the inner area of our spline matte, so let’s re-adjust it.

- *Press the button Zoom Window::Select All to make sure all CVs are selected, hold down the Shift key and move the entire Spline Matte back over the blue ball, by click-dragging the left mouse button.*

As soon as you start to move the spline area matte’s CVs, its outline becomes red and the CV’s control points become green, again. This indicates that the spline matte has become keyframed in the current frame. Let’s check this.

- *Drag the Main Window’s frame slider back and forth between frame 1 and frame 10.*

The spline matte smoothly changes its position between frame 1 and frame 10.

- *Press the button Main Window::Begin, then press the button Main Window::Key+.*

The current frame number will be 10. While Zoom Window's Auto Tracking Controls are enabled, the buttons Main Window::Key- and Main Window::Key+ allow you to jump from keyframe to keyframe of the currently selected spline area matte. Let's continue to animate the spline matte.

- *Select the toggle menu item Main Window::Options::Frame Keyboard Grid::10 Frames. Make sure the mouse cursor is located over the display area of 3DE's Zoom Window, then press the Cursor-Right key.*

Now, the current frame number will be 20. The blue ball has moved outside the inner area of our spline matte, again.

- *Press the button Zoom Window::Select All to make sure all CVs are selected, hold down the Shift key and move the entire Spline Matte back over the blue ball, using the left mouse button.*

The blue ball should now be covered by the spline matte, again. The red outline and green control points are indicating that the spline matte has become keyframed in the current frame, as well.

- *Animate the Spline Area Matte by creating three additional keyframes in frames 30, 40 and 50.*

You will notice that the shape of our spline area matte is too small to cover the entire blue ball in frame 50.

- *Increase the shape of our spline matte by click-dragging all CVs and their control points separately, so that it becomes significantly bigger than the blue ball (see figure below).*

The spline matte is now big enough to cover the entire blue ball.

- *Continue to animate the spline area matte by creating a new keyframe every 10 frames. At the end of the sequence the blue ball becomes smaller again, so that you can decrease the size of our spline matte, accordingly.*

Before we start 3DEqualizer's automatic tracking procedure, we should make sure that the blue ball is properly covered by our spline matte in every frame.

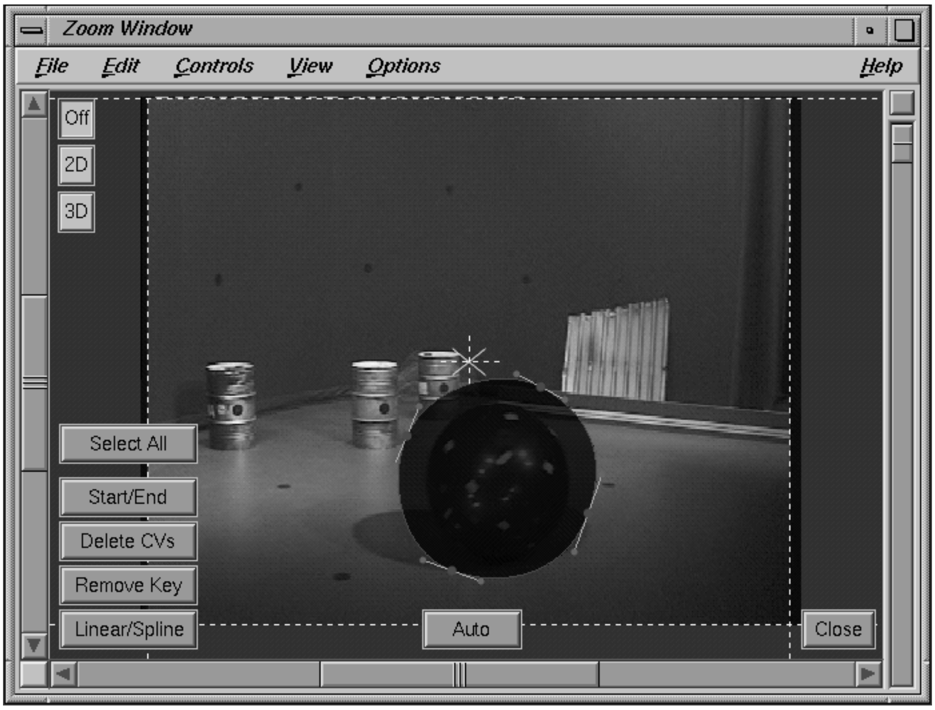
- *Please select the menu item Main Window::Playback::Fill Imagebuffers. If a requester opens and asks you to free the used imagebuffers, click on the button "Free Used Buffers".*

3DEqualizer starts loading the image files of sequence "ball_bluebox1" into memory.

- *Disable the menu toggle button Zoom Window::View::Show Tracking Region Mask.*

The mask (dark area) is no longer being displayed. This will make it easier to see whether the blue ball is still covered by the spline matte or not.

- *Go through the sequence by dragging the Main Window's frame slider and examine the spline area matte. Whenever the blue ball is not covered by the spline matte in a particular frame, modify some of the CVs until the blue ball is properly covered again.*



Increased spline area matte

Editing the spline area matte is now complete, so we can start tracking some points (see also project "auto_track3.3de").

- *Please press the button Zoom Window::Auto to start the automatic tracking procedure.*

The button's label is changed into "Stop" and 3DEqualizer tracks the sequence frame by frame, creating new point objects. As soon as a point comes too close to the spline area matte, it is switched off. After some time, the automatic tracking procedure is finished and 101 new points have been created. Let's check the result by calculating a 3D solution. In order to do so, we need to define some camera parameters, first.

- *Select the environment "Camera Adjustment" from the Main Window's environments menu.*

As in the previous section we will enter the correct camera values for this sequence, manually.

- *Set Horizontal Angle to 45.0, Pixel Aspect to 1.0926 and Lens Distortion to 0.04.*

- *To calculate a 3D solution, please select the menu item Main Window::Calc::Calc All Objects From Scratch.*

3DEqualizer starts its regular calculation process to compute a 3D camera motion path and 3D positions for all tracking points. In contract to the previous section, the calculation finishes without any problems! Finally, let's take a look at the result. Although we didn't search for any badly tracked points, the average deviation is already around 0.58 pixels, which is not a bad value. You will be able to improve the result even further, if you apply the techniques you learned from the previous sections.

Tutorial 3: Manual Motiontracking

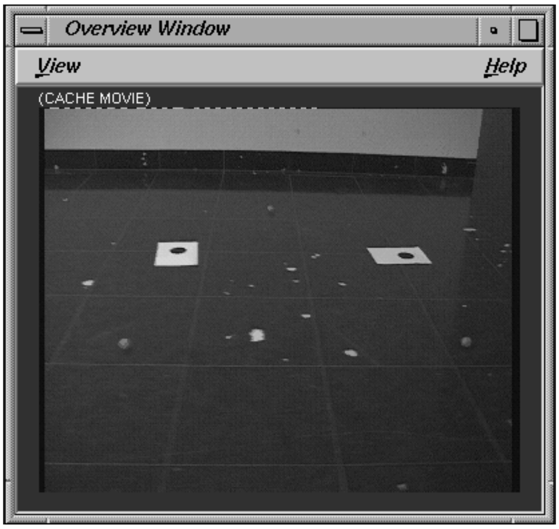
In principle, there are three different manual tracking modes in 3DEqualizer. Patterntracking is the standard method that can be applied to almost every situation. In some rare cases it can suffer from error accumulation, which causes tracking quality to decrease more and more during the tracking process. Marker- and cornertracking are specialized to certain situations and therefore cannot be used in all cases. However, they are immune against error accumulation. The purpose of this section is to present several situations which can occur in connection with pattern- and markertracking. In the first section we deal with markertracking (tracking of simple, symmetric, unicolored objects). The use of the Image Controls Window is illustrated in the second section by means of a realistic example. In the last two sections, regular patterntracking and its difference from markertracking are explained.

Markertracking

- *Please start 3D-Equalizer as described in the section "Getting started" and open the project "tracking1.3de" in the directory "<tutorial install directory>/projects/".*

- *Select the environment "Motiontracking" from the Main Window's environments menu.*

On the floor you see two markers and three little balls. In the course of the animation, the camera moves around these objects.



Overview Window displaying cache movie

- *If you wish to get an impression of the sequence you can create a cache movie by executing the function Main Window::Playback::Create Cache Movie. If there is enough RAM installed in your machine, you can also select the function Main Window::Playback::Fill Imagebuffers and go through the sequence using the Main Window's frame slider.*

There is already a pointgroup named "floor", but you need to create a point object.

- *Please press the button Add in the Pointgroups Window in order to create a new point.*

The requester "Create Point" opens.

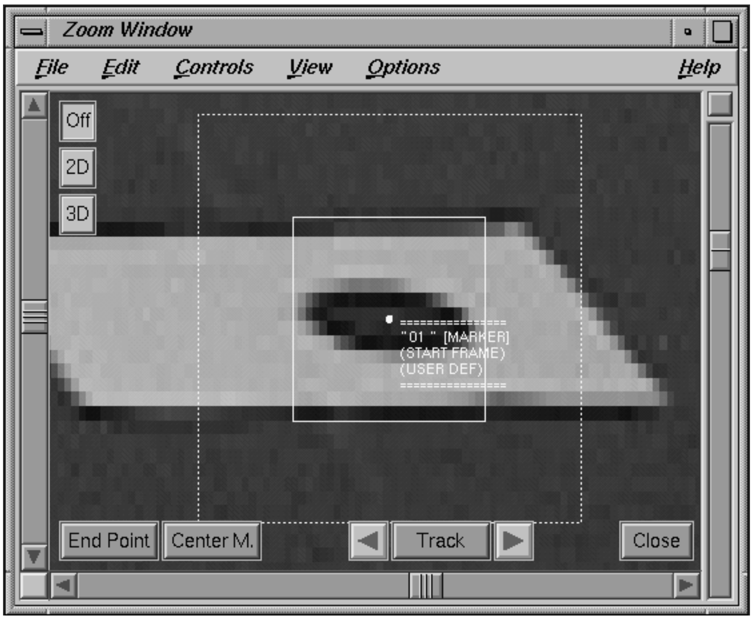
- *Change the selector Trackingmode from Pattern to Marker.*

This activates the markertracking mode for this point. All other settings can be left as they are.

- *Confirm the entries by pressing the button Ok.*

In the first example we shall see what happens if the two tracking areas of the screen point are not adjusted properly.

- *Go to frame 1 and place the tracking point over the right marker in the Zoom Window (without modifying the motiontracking areas), as shown in the figure below.*

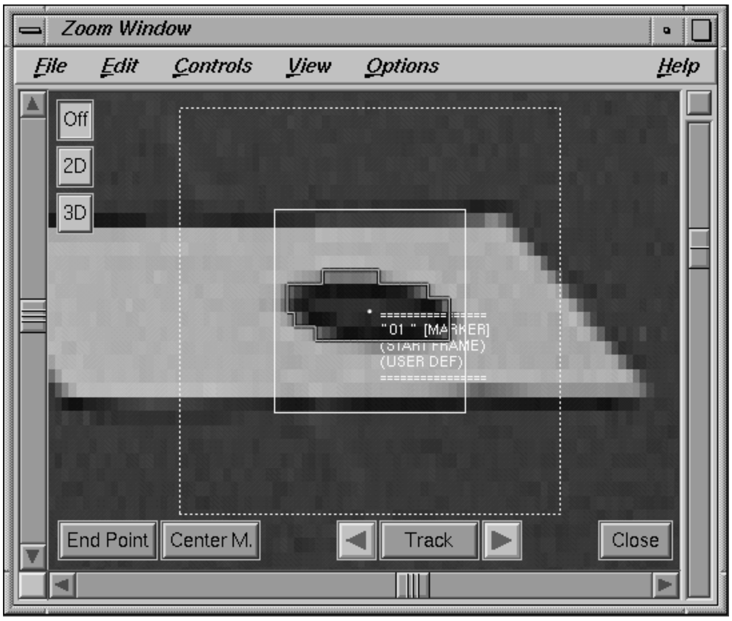


Markertracking

By doing this you specify a keyframe for this point.

- *As already mentioned in the first tutorial, press the button Center M. several times to let 3D-Equalizer find the center of the marker.*

Each time you press and hold down the button Center M, a white outline is displayed in the Zoom Window. This outline shows the outer border of the marker that 3DEqualizer has identified (see figure below). This mechanism is a great help to find out whether a marker can be tracked correctly or not.



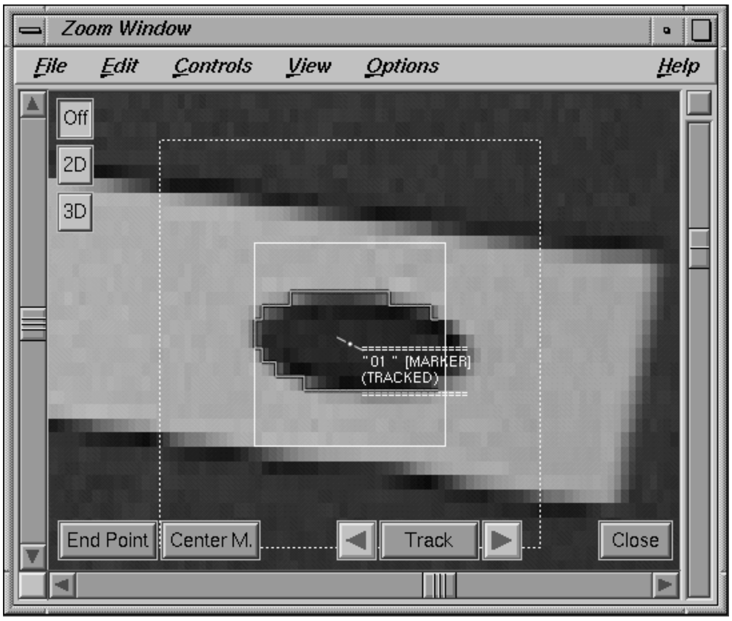
Marker outline identified by 3DEqualizer

Since 3DE is able to locate the center of the marker correctly, we are ready to start the tracking procedure.

- *Activate the toggle Zoom Window::Options::Auto Center. Press the button Track Forward a few times and make sure that the tracking system finds the marker center in the following frames as well.*

- *Start the tracking procedure by pressing the button Track in the Zoom Window, and watch the marker being tracked throughout the entire sequence.*

At frame 23 approximately, the system will not be able to find the center of the marker correctly. The reason is that the size of the marker is constantly growing from frame to frame. At a certain point the marker no longer fits in the search pattern (see figure below), and the markertracking fails (see also chapter “Selected Topics”, section “Markertracking”).



Search pattern too small

We shall now demonstrate how such situations can be resolved.

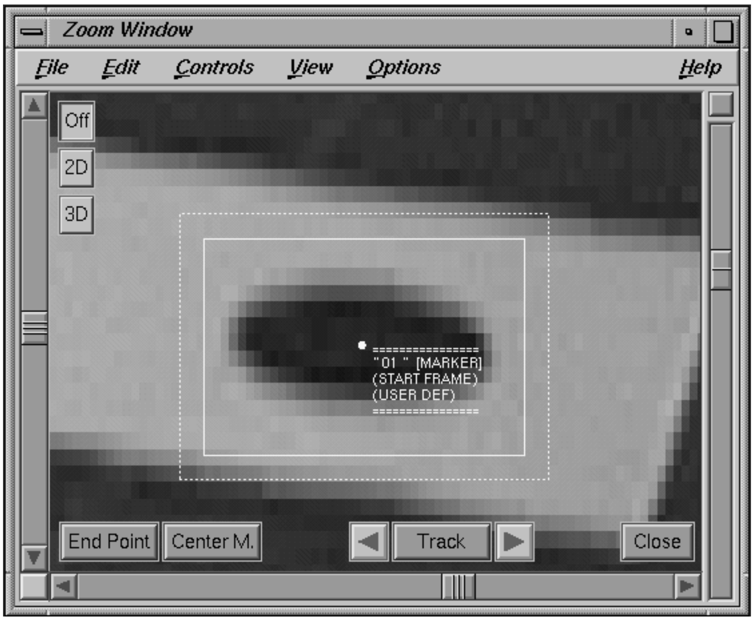
- *Delete the entire motion curve by pressing Zoom Window::Edit::Delete Curve::All. Go to frame 1, place the tracker point on the marker’s center and adjust the search pattern so that the marker does not touch its borders. Then, press the button Track to restart the tracking procedure.*

- *Whenever there are less than two pixels of bright background between the marker and the edge of the search pattern, interrupt the tracking procedure by pressing the button Stop. Adjust the search pattern so that it covers the marker, and so that the bright background surrounds the marker within the search pattern, then continue the tracking by pressing the button Track again.*

Important note: The best way to get a precise motion curve is to supervise the tracking process visually. If the point of interest is not tracked correctly, you should interrupt the process immediately and adjust the tracking areas accordingly. This technique saves time and guarantees the highest tracking quality!

We have seen the effect of resizing the search pattern appropriately. In the next example you will see what happens if the search area (outer, dotted line rectangle) is too small.

- *Please choose the function Zoom Window::Edit::Delete Curve::All in order to delete the motion curve. Go to frame 26, place the tracker point on the marker's center and adjust the search pattern correctly as described above. However, set the search area to its minimum size as shown in the figure below.*



Search area too small

- *Press the button Center M. several times, to ensure that the system identifies the center of the marker correctly. Deactivate the Auto Center mode in the menu Options.*

- *Now please press the button Track Forward one time.*

As you see, the following frame is not tracked correctly. The reason for this problem is that from frame 26 to frame 27 the marker moves very fast. The tracking system tries to find the marker in frame 27 within the search area specified in frame 26. The search area is too small and the marker is not contained in the search area and hence the tracking procedure fails.

- *In order to avoid this please go back to the previous frame 26 and enlarge the search area slightly. Press the button Track Forward again.*

The marker now can be tracked correctly in the following frame. You see that - depending on the type of camera movement - a certain minimum size of the search area is required, but please keep in mind that increasing the search area requires more processing time.

Image Controls Window

As we have seen in the last section, if the markermode is used for tracking, there have to be well-defined markers on a contrasting uniform background. In practice these conditions are not always fulfilled. In the following example a technique for improving the tracking quality is illustrated. The sequence for this section was made available by Trix, Belgium.

- *Please start 3D-Equalizer as described in the section "Getting started", and open the project "tracking2.3de" in the directory "<tutorial install directory>/projects/".*

- *Open the environment "Motiontracking #2" from the Main Window's environments menu.*

You see a big studio with several yellow tennisball markers on the floor.

- *If you wish to get an impression of the sequence you can generate a cache movie by selecting the function Main Window::Playback::Create Cache Movie. If there is enough RAM installed in your machine, you can also select the function Main Window::Playback:: Fill Imagebuffers and go through the sequence using the Main Window's frame slider.*

As in the previous example there is already a pointgroup named "studiofloor", however it is necessary to create a point object.

- *Press the button Add in the Pointgroups Window.*

A requester "Create Point" opens.

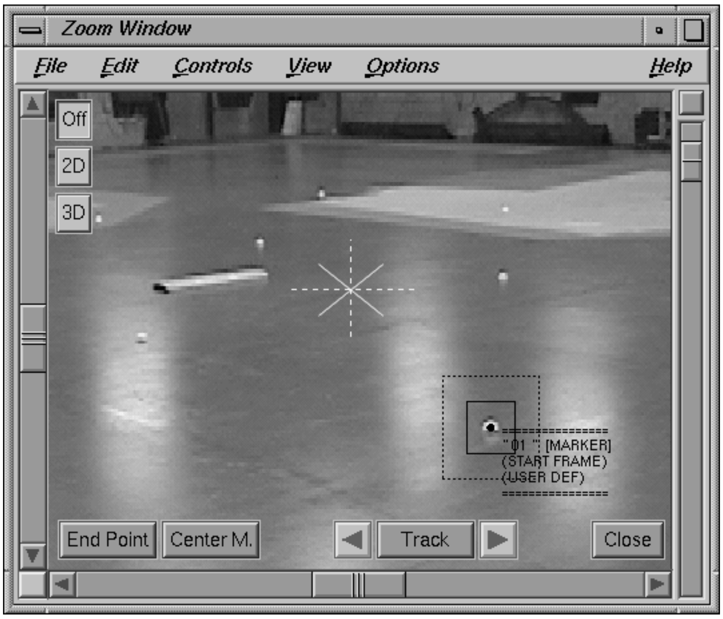
- *Please change the selector Trackingmode to Marker.*

This activates the markertracking mode for this point. All other settings can be left as they are.

- *To confirm your edits, press the button Ok.*

A new point object is created.

- *Activate Zoom Window's autocenter mode by choosing the menu function Zoom Window::Options::Auto Center. Jump to the first frame and place the tracker point on the tennisball marker which is located approximately in the middle of the frame (see figure below).*



Studio floor with several tennis ball markers

- *Please press the button Center M. several times, to ensure that the tracking system identifies the center of the marker correctly. Activate the tracking procedure by pressing the button Track, and watch the tracking procedure until the end of the sequence.*

You will note that from frame 8 to frame 19 in the area around the marker there is a reflection from one of the light sources. During this period, 3DEqualizer becomes confused and has problems to focus on the tennisball marker. However, by means of the Image Controls Window this problem can be avoided.

- *Please jump back to frame 1 and choose the function Zoom Window::Edit::Delete Curve::All in order to delete the motion curve.*

The functions of the Image Controls Window allow you to manipulate the images in order to improve contrast or, in this case, eliminate a reflection. You can figure out the image control curves in order to eliminate the reflection, however to get started it might be better to import a prepared set of curves.

- *Select the menu function Image Controls::File::Load Curves. Please load the curves file "Trix2.crv" stored in the directory "<tutorial install directory>/projects/".*

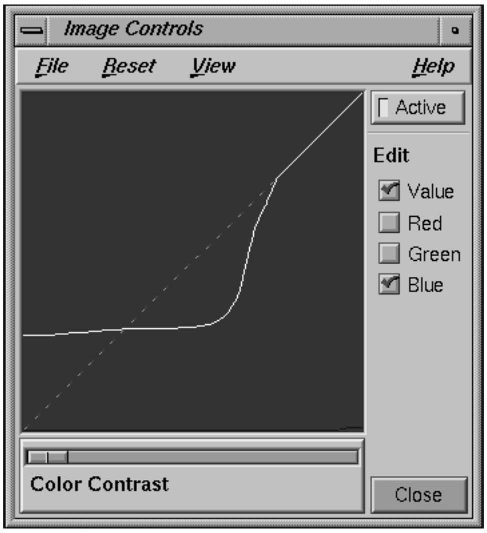


Image Controls Window

- *Activate the toggle button Active.*

3D-Equalizer modifies the image according to the loaded curves. As you can see the colored image has become greyscaled.

- *Jump to frame 14, and take a look at the Zoom Window.*

The dark shadow as well as the reflection on the floor are eliminated.

- *Now, jump to frame 1 and place the tracker point on the tennisball marker again. Press the button Center M. a few times, to ensure that 3D-Equalizer identifies the center of the marker correctly. Please start the tracking procedure, and watch the marker being tracked until the end of the sequence.*

Now that the misleading parts of the images are removed using 3DEqualizer's image controls, tracking will be performed successfully.

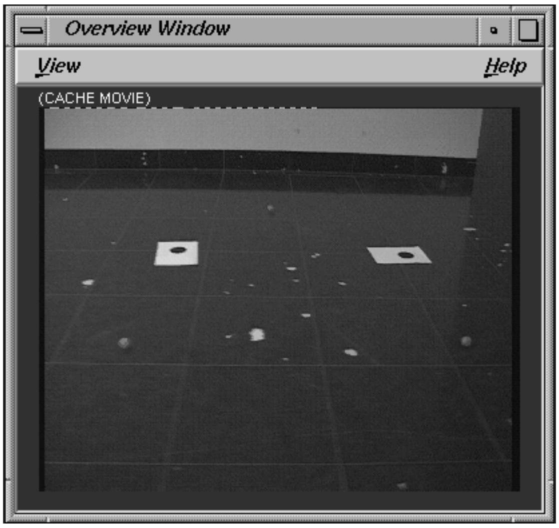
Patterntracking: Rotating and Scaling the Reference Pattern

Up to now, during all tutorials, 3DEqualizer's markertracking has been used exclusively. However, it often occurs that markers are not available. If this happens, points can be tracked using patterntracking. In the following section, we will discuss fundamental principles of patterntracking.

- *Please open the project "tracking3.3de" located in the directory "<tutorial install directory>/projects/".*

- *Open the environment "Motiontracking" from the Main Window's environments menu.*

You see the same sequence that has already been used for the first section of this tutorial (see figure below).



Overview Window displaying cache movie

- *For a clear impression of the sequence you can generate a preview movie by using the function Main Window::Playback::Create Cache Movie. If there is enough RAM installed in your machine, you can also select the function Main Window::Playback::Fill Imagebuffers and go through the sequence using the Main Window's frame slider.*

There is already a pointgroup named "floor", as well as a point object named "01". First we shall see what happens if markertracking is applied in situations in which it should not be applied at all.

- *Double-click on the entry “01” of the Pointgroups Window.*

The requester “Modify Point” opens.

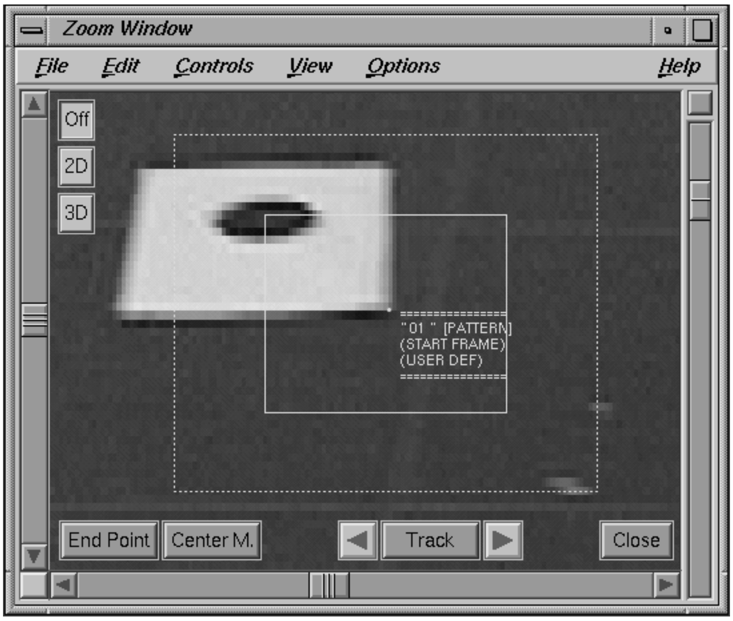
- *Please change the option menu Trackingmode to Marker and confirm your edits by pressing the button Ok.*

This activates markertracking mode for this point (which will soon turn out to be the wrong mode in this example). Point “01” is already placed on the lower right corner of the left sheet of paper (see figure below).

- *Press the button Center M. several times.*

The tracker point is jumping around and finally remains inside the white sheet of paper. As described in the previous section the situation here does not fulfill the conditions required for markertracking (see also chapter “Selected Topics”, section “Markertracking”). We have to use patterntracking here!

- *Click on the lower right corner of the left sheet of paper to place the tracking point over it again (see figure below).*



Patterntracking

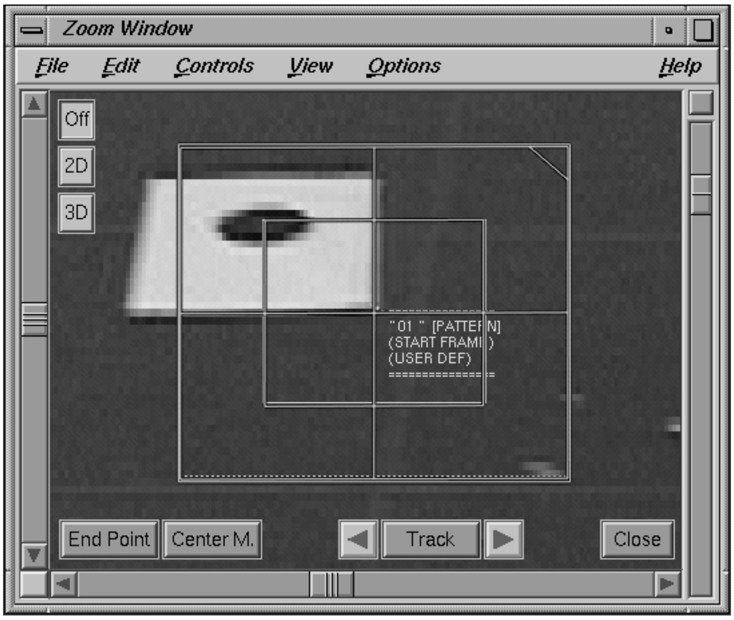
- *Double-click on the entry “01” of the Pointgroups Window to open the requester Modify Point. Change the option menu Trackingmode to Pattern.*

You have now activated patterntracking mode for this point. The fundamental principles of patterntracking are explained in the following.

During the tracking process, 3DEqualizer tries to find the so-called reference pattern in each to be tracked frame. The reference pattern is defined by the point’s search pattern in a certain frame. It can be defined by the first keyframe, the previous keyframe, the previous frame or by an explicit, user-defined frame. You can configure this behaviour with the option menu Ref. Pattern.

To identify the reference pattern in a given frame, 3DEqualizer is able to rotate and scale it. Scaling is performed in both dimensions separately. You can force 3DE not to scale and/or rotate the reference pattern by disabling the toggle buttons Allow Rotation/Scaling of Reference Pattern.

- *Please confirm your edits by pressing the button Ok. Activate the toggle Zoom Window::Options::Auto Center and press the button Track to start 3DEqualizer’s tracking procedure.*



White grid representing reference pattern

As you see, the white corner is tracked very well throughout the entire sequence without any problems. Let's examine the tracking result more in detail.

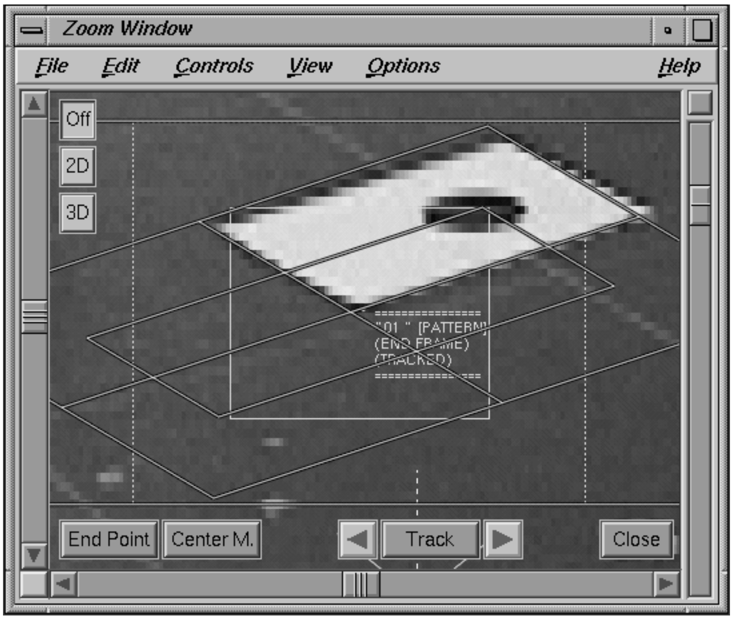
- *Go to frame 1 and press the button Center M for a few seconds.*

A white grid is displayed in the Zoom Window (see figure above). It consists of a big and a small box. The small, inner box represents the reference pattern, after it has been scaled and rotated to track the current frame. Since Frame 1 defines a reference pattern, the white boxes exactly match the point's search area and search pattern.

> **Important note:** In contrast to markertracking, the function Center M does not alter the tracking point postion when applied to a patterntracking point. It only displays the white grid so that you can verify the result of the patterntracking process.

- *Go to the last frame of the sequence and press the button Center M, once again.*

The white grid has completely changed in comparison to the first frame. It is rotated and scaled to compensate the change in perspective, applied to the white sheet of paper throughout the course of the sequence (see figure below).



Reference pattern being scaled and rotated

In some rare situations the tracking quality of the patterntracking engine can be improved by forcing 3DE not to rotate and/or scale the reference pattern. However, these situations do not happen very often, and therefore you shouldn't modify these settings until it becomes absolutely necessary. Let's examine what happens, if we do not allow 3DE to rotate the reference pattern in this example.

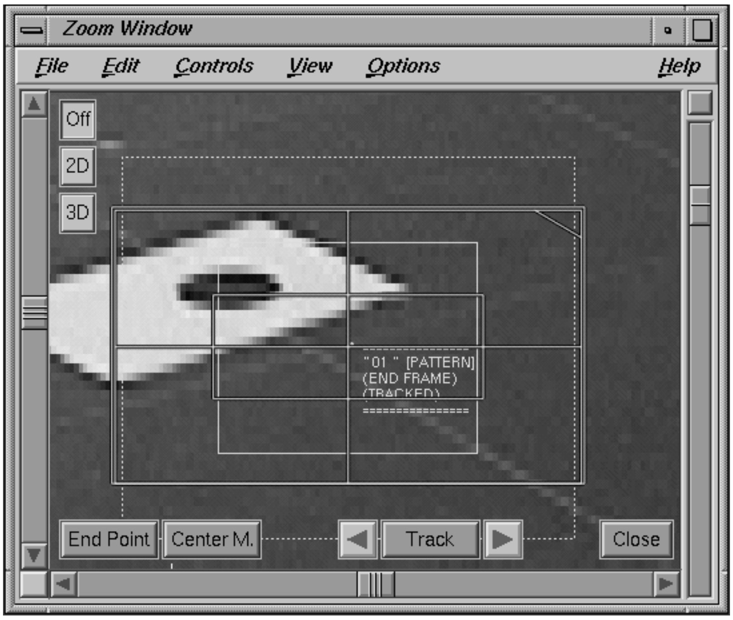
- *Please double-click on the entry "01" of the Pointgroups Window to open the requester Modify Point. Disable the toggle button Allow Rotation of Reference Pattern and confirm your edits by pressing the button Ok.*

- *Make sure that menu toggle Zoom Window::Options::Auto Center is activated and press the button Track to start the tracking procedure.*

As you see, 3DEqualizer is not able to track the lower right corner of the white sheet of paper. Frame by frame, the tracking point slowly loses its target and finally ends up approximately at the upper right corner of the white sheet of paper.

- *Go to the last frame of the sequence and press the button Center M.*

The white grid indicates that the reference pattern has only be scaled down vertically and slightly scaled up horizontally (see figure below). Disabling rotation makes it impossible for 3DE to track the reference pattern successfully!



Badly transformed reference pattern

Patterntracking: Forward & Backward Tracking

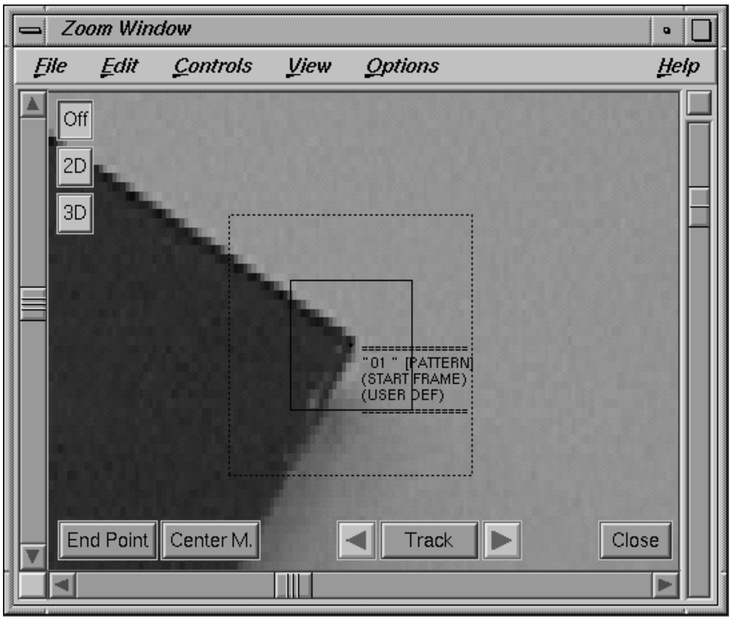
In the following section we will demonstrate what happens, if it becomes necessary to force a point to take its reference pattern always from the previous frame. You will learn the effect of “error accumulation” and what you can do to improve tracking quality in these cases.

The example we consider here, can of course also be tracked as described in the previous section, where we took the reference pattern from the previous keyframe. We use it here because the camera movement is very smooth and the effect of error accumulation becomes apparent. In practice, the technique of taking the reference pattern from the previous frame is applied in situations where pattern tracking fails because the reference pattern and the search area in the tracked frame differ too much.

- *Please start 3D-Equalizer as described in the section “Getting started” and open the project “tracking4.3de” in the directory “<tutorial install directory>/projects/”.*

- *Open the environment “Motiontracking” from the Main Window’s environments menu.*

You see a red painting hanging on a wall. In this section of the tutorial, we will track the upper right corner of the red painting.



Patterntracking point

- *For a clear impression of the sequence you can generate a preview movie by using the function Main Window::Playback::Create Cache Movie. If there is enough RAM installed in your machine, you can also select the function Main Window::Playback::Fill Imagebuffers and go through the sequence using the Main Window's frame slider.*

There is already a pointgroup named "wall", but you have to create a new point object.

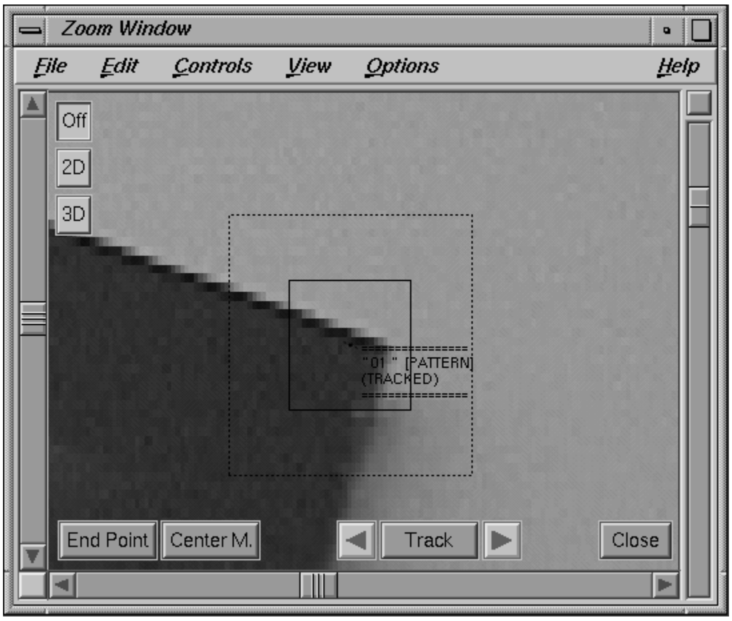
- *Press the button Add in the Pointgroups Window.*

A requester "Create Point" opens.

- *Set option menu Ref. Pattern to Previous Frame and confirm the creation of the new point by clicking the button Ok.*

You have created a new patterntracking point, which always takes its reference pattern from the previous frame.

- *Jump to frame 1, activate the Auto Center mode, by selecting Zoom Window::Options::Auto Center and place the tracker point on the upper right corner of the painting, as shown in the figure above.*



Point tracked with accumulated error

- *Start the tracking procedure by pressing the button Track and watch the tracker point during this process.*

At approximately frame 25, the tracked point deviates by about two pixels from the corner of the painting (see figure below).

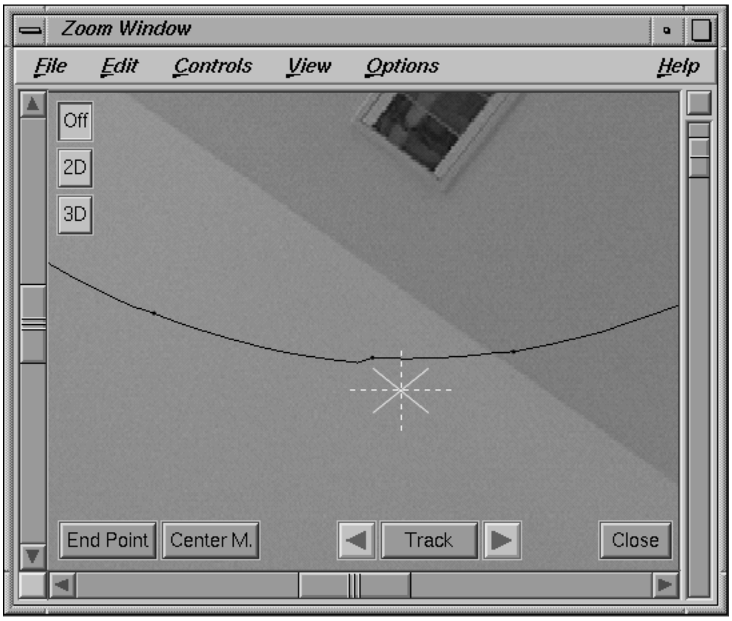
- *Please interrupt the tracking at this time, by pressing the button Stop.*

What is going on? We configured the point so that it takes its reference pattern always from the previous frame. In each step the deviation of the tracked point from its real position becomes bigger and bigger due to numerical inaccuracies and the lack of information in pixeloriented images. We shall refer to this phenomenon as “error accumulation”.

- *In order to “work around” the error accumulation, reposition the screen point at the corner of the painting, thus generating another keyframe.*

- *Please continue tracking until the end of the sequence. At approximately every 20 frames, you should stop the tracking process for manual repositioning.*

- *Make 3D-Equalizer displays the motion curve by selecting the menu function Zoom Window::View::Show Motion Curve. Zoom out so that you can examine the entire curve (see figure below)*



Motion curve with “jaggies” due to error accumulation

You will note that the motion curve exhibits little “jaggies” at the intermediate keyframes. For 3D-Equalizer’s calculation procedures it is quite important to avoid these jaggies, because all properties of the motion tracking curves have an impact on the three-dimensional camera motion path. We shall now show how this inaccuracy can be avoided.

- *Press the button Modify in the Points Window.*

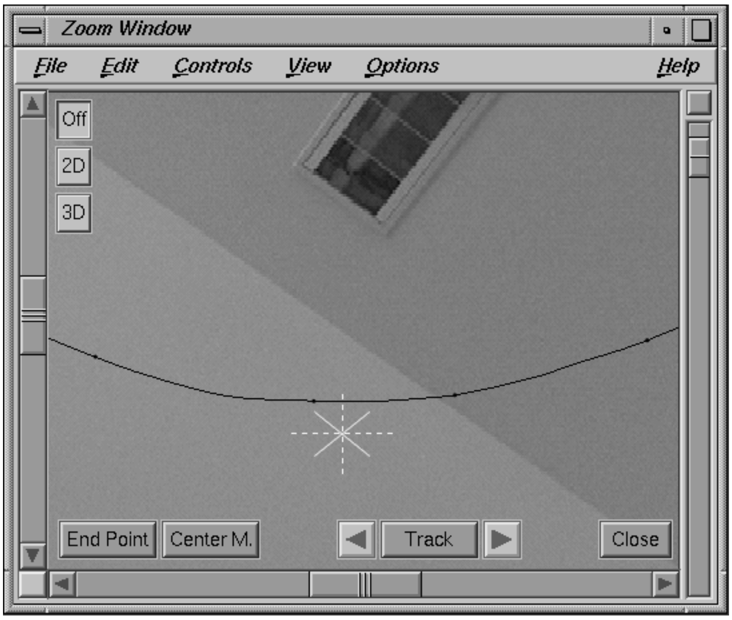
The requester Modify Point opens.

- *Change the Direction setting from Forward to Forward & Backward. Close the requester by pressing the button OK. Another requester opens which asks you to decide whether the motion curve should be set to obsolete or not. Since you wish to replace the failed tracking curve with a better one, press the button Set Obsolete in this requester.*

For the trackingmode Forward & Backward it is recommended to designate at least the first frame and the last frame of the tracking curve as keyframes.

- *Go to frame 1. Since there are already a few keyframes from the previous track, you just start the tracking process as usual by pressing the button Track.*

On the first pass, 3D-Equalizer tracks the corner from the beginning until the end of the sequence, producing exactly the same motion curve as before. Having arrived at the last frame, a requester opens which tells you to define the last keyframe.



Motion curve smoothed out by “Forward & Backward” motiontracking

- *Press Cancel to interrupt the tracking process and define the last keyframe by clicking on the right corner of the painting as precisely as you can.*

- *Start the tracking procedure again.*

Now the corner is tracked backwards from the end to the beginning. During this pass the curve is smoothed out and the jaggies are eliminated (see figure above).
What has happened? During the backward tracking procedure 3D-Equalizer produces another motion curve, which is combined with the forward motion curve using a splining prescription so that the resulting curve becomes smooth.

Tutorial 4: Camera Adjustment

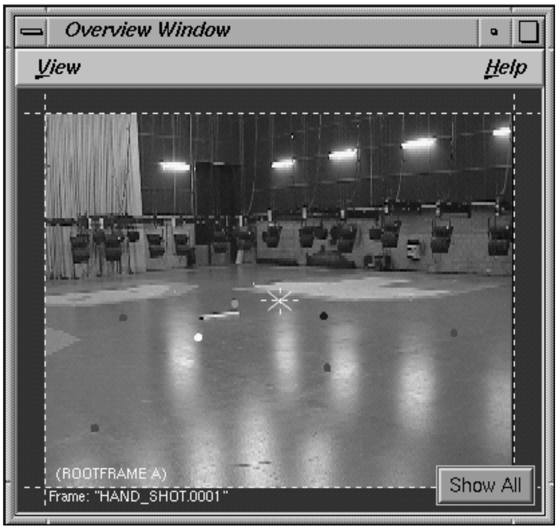
The following tutorial demonstrates how to use the functions of the Camera Adjustment Window in order to minimize the calculation time for camera adjustment procedures.

In general the pre-conditions for using the camera adjustment feature are: precise motiontracking curves and a noticeable amount of camera movement, i.e. the camera position must change considerably during the course of the sequence. If this is not the case, reference frames can be used. (see also Tutorial Reference Frames).

- *Please start 3D-Equalizer as described in the section “Getting started” and open the project “cam_adjust.3de” in the directory “<tutorial install directory>/projects/”.*

- *Open the environment “Motiontracking” from the Main Window’s environments menu.*

You see a studio with several yellow tennis ball markers on the floor (see figure below).



Studio floor with several tennis ball markers

- *Please take a look at the sequence by clicking a few times on the left hand side of the Main Window’s frame slider.*

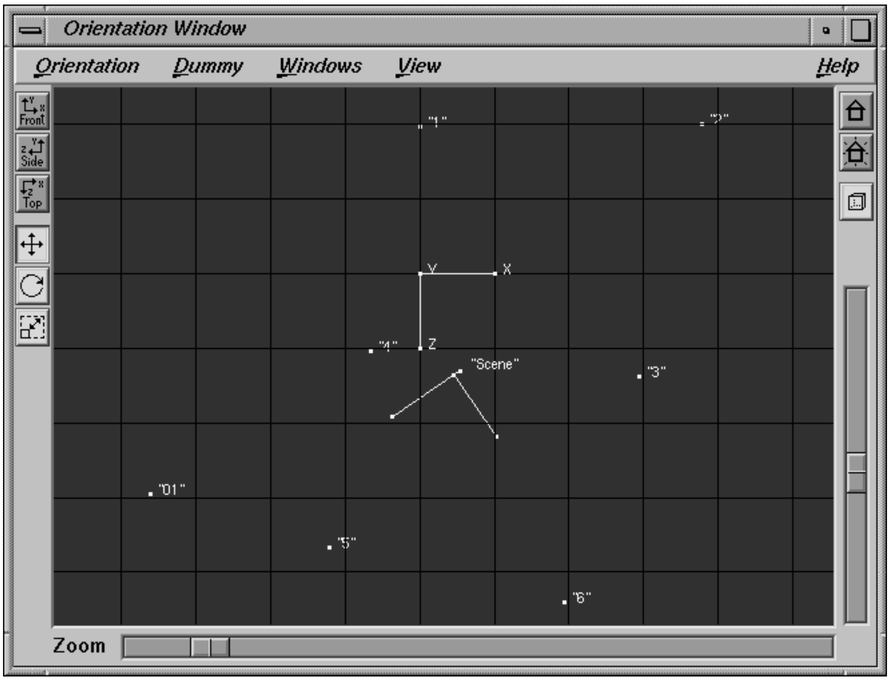
The camera is moving around the tennisball markers. There is already a sufficient number of tracked points, so we can now concentrate on the camera adjustment procedure only.

- *Open the Camera Adjustment environment.*

There is a camera named “<default camera>” which is already linked to the image sequence. You will find standard values for the camera parameters. First of all, let us try to calculate a camera motion path including points in 3D space using these standard values.

- *Open the Orientation environment and choose the menu item Main Window::Calc::Calc All Objects From Scratch.*

After a few seconds the 3D point positions are displayed in the Orientation Window’s display. As you can easily see from the image sequence, the tennisball markers form squares on the floor. However, if you examine the calculated point positions you notice that they do not appear to form squares. The points are distorted due to incorrect camera parameters (see figure below).



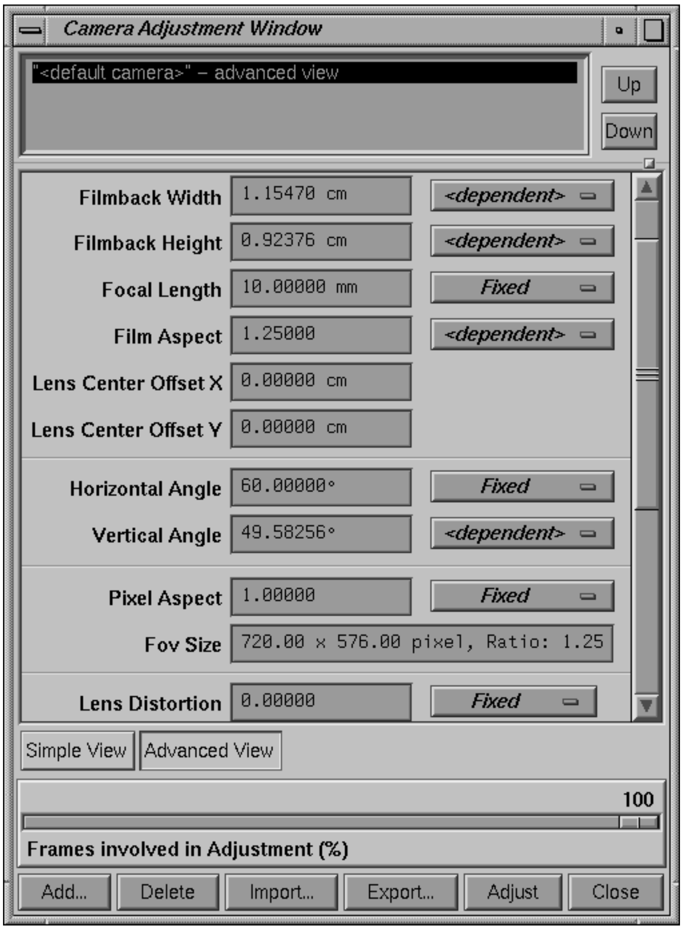
Distorted point model due to wrong camera parameters

- *Open the Status Window and take a look at the average deviation value.*

The average deviation is about two pixel, which is too high if you take into account that the tracking of this project is quite precise. It seems that the camera parameters are not precise, so let’s try to improve the quality of the solution by performing a camera adjustment procedure.

Rough camera adjustment

- *Please open the Camera Adjustment environment again. To get a full overview of all camera parameters, click on the Advanced View button.*



Camera Adjustment Window

"<default camera>" – advanced view

Filmback Width 1.15470 cm <dependent>

Filmback Height 0.92376 cm <dependent>

Focal Length 10.00000 mm Fixed

Film Aspect 1.25000 <dependent>

Lens Center Offset X 0.00000 cm

Lens Center Offset Y 0.00000 cm

Horizontal Angle 60.00000° Fixed

Vertical Angle 49.58256° <dependent>

Pixel Aspect 1.00000 Fixed

Fov Size 720.00 x 576.00 pixel, Ratio: 1.25

Lens Distortion 0.00000 Fixed

Frames involved in Adjustment (%) 100

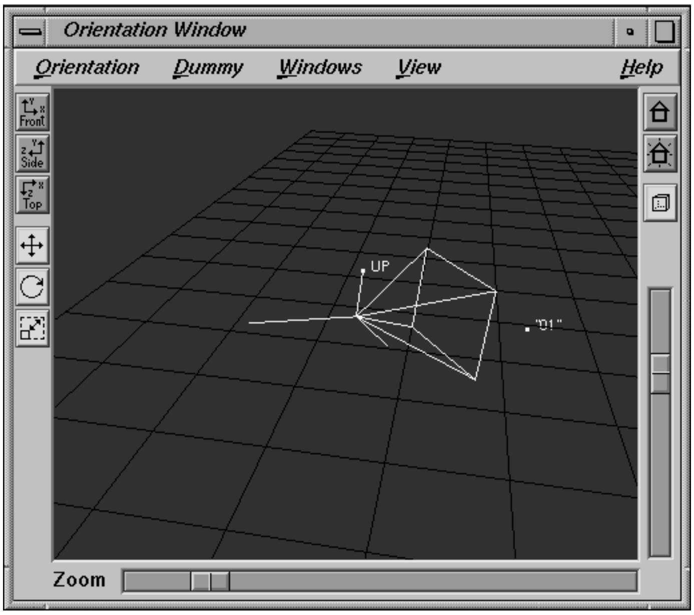
Camera Adjustment Window - Advanced View

Let us assume that we have no information about the camera parameters. Of course, we could set every possible selector to Unknown and then simply start the adjustment procedure, but this would probably take about one or two hours to compute. Although we do not know any of our camera parameters, it is still possible to aquire some useful information. In this section, we will try to get a rough idea of our camera parameters, and in the next section we will “refine” the result.
If you examine the image sequence, you will discover no “extraordinary” distortion, that means, as in almost every production, the lens distortion will probably be between 0.0 and 0.1. This amount of lens distortion has only a minor effect on the camera path calculation, so we can ignore it for our rough camera adjustment.

- *Enter 0.0 into the lens distortion field and set the selector to Fixed.*

You can see that the image sequence has been filmed with a video camera. The film aspect of standard video material is always 4:3 (1.3333).

- *Set the selector Film Aspect of the Camera Adjustment Window to Fixed and enter 1.3333 in the respective field.*



Default camera

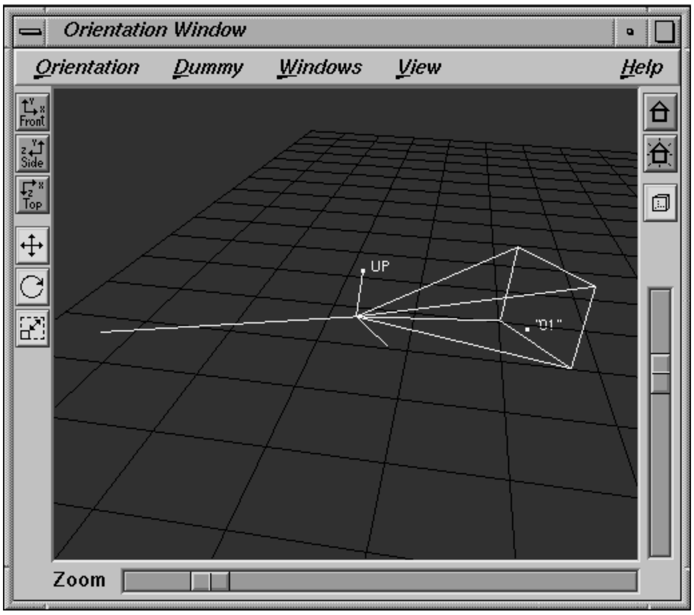
The FOV was already very well adjusted. The Pixel Aspect has been automatically changed to <dependent> and approximately 1.067, which is the known value for PAL video.

Important note: To guarantee a consistent camera, either the pixel aspect or the film aspect is <dependent>. As long as you do not have any other information, you fit in with the following rules. If you work with Video or Telecined material, adjust the FOV and set the Film Aspect selector to Fixed and the respective field to 1.3333. If you work with scanned (film) images, set the Pixel Aspect to Fixed and the respective field to 1.0.

- *Open the Orientation Window and activate the toggle button View::Show Camera. Use the navigation functions of the Orientation Window such that the camera object becomes visible (see figure above).*

- *Now modify the Horizontal Angle value in the respective field of the Camera Adjustment window to 30.0.*

If you examine the camera object again, you will notice that the shape of the camera pyramid has changed (see figure below).



Camera shape changed

- *Enter the original value of 60.0 into the Horizontal Angle field of the Camera Adjustment Window.*

- *Now modify the Focal Length value to 5.0.*

This time the camera pyramid has not changed its shape. In general, there are always three non-dependent camera parameters in the Camera Adjustment Window (and in addition the Lens Distortion parameter). In almost every case, two of them will influence the shape of the camera pyramid, while the remaining one does not. However, for calculating the camera motion path, only the shape of the camera pyramid is important. In our case, the Focal Length parameter has no influence on the camera shape.

- *Leave the Focal Length field as it is, and set the respective selector to Fixed.*

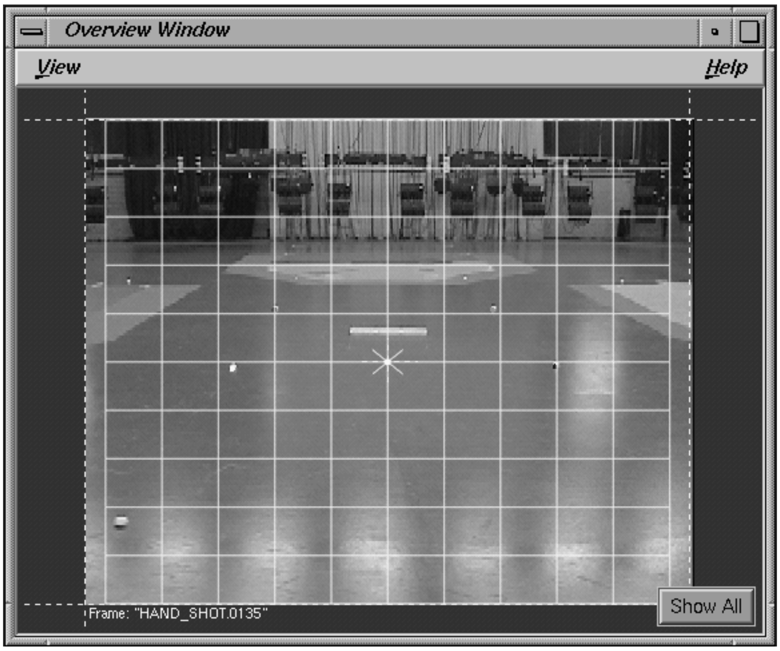
- *Set the remaining selector "Horizontal Angle" to Unknown and press the Adjust button to start the camera adjustment procedure.*

After some time, 3D-Equalizer has calculated a horizontal angle of 42.86. The Status Window indicates an average deviation of 0.37 pixel, which is already an acceptable value.

Finetuning the camera parameters

- *Please activate the toggle Overview Window::View::Show Distortion Grid and go to frame 135.*

As you see in the Overview Window, the edges of the floor and the wall seem to be slightly curved in comparison to the distortion grid (see figure below). This is probably a lens distortion effect. We can improve our entire reconstruction even more if we take lens distortion into account.



Lens distortion not matching

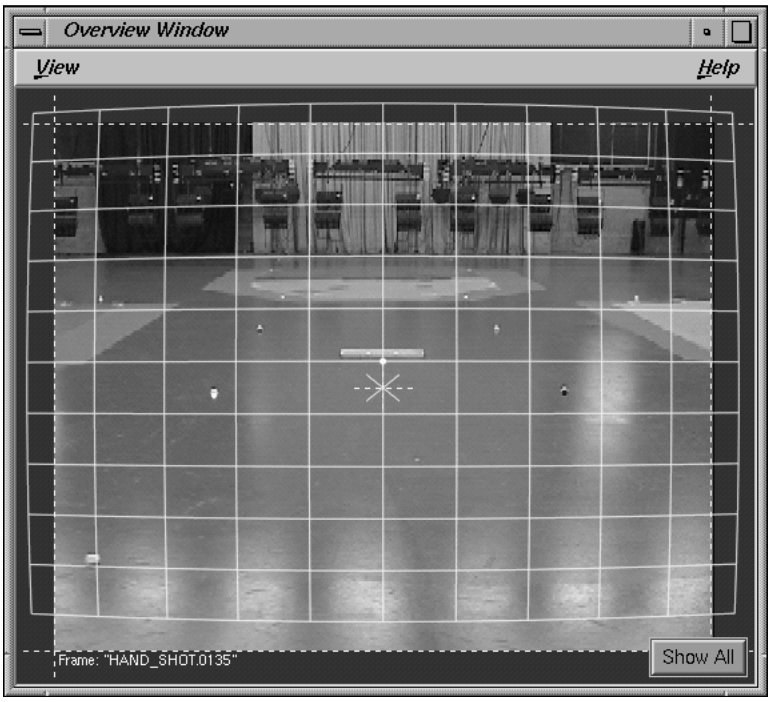
- *Set the Lens Distortion selector of the Camera Adjustment Window to Fine Adjust and press the Adjust button.*

After some time, 3D-Equalizer has calculated a new lens distortion value of 0.014. The average deviation of the Status Window has changed to 0.34 pixel. However, if you take a look at frame 135 again, you will notice that there is still a mismatch between the curvature of the floor-wall edge and the lens distortion grid.

The lens distortion and the "shape" of the camera are closely related to each other. That means, if one of them is modified the other has to be modified too.

- *Set the Horizontal Angle and the Lens Distortion selectors to Fine Adjust, and press the Adjust button.*

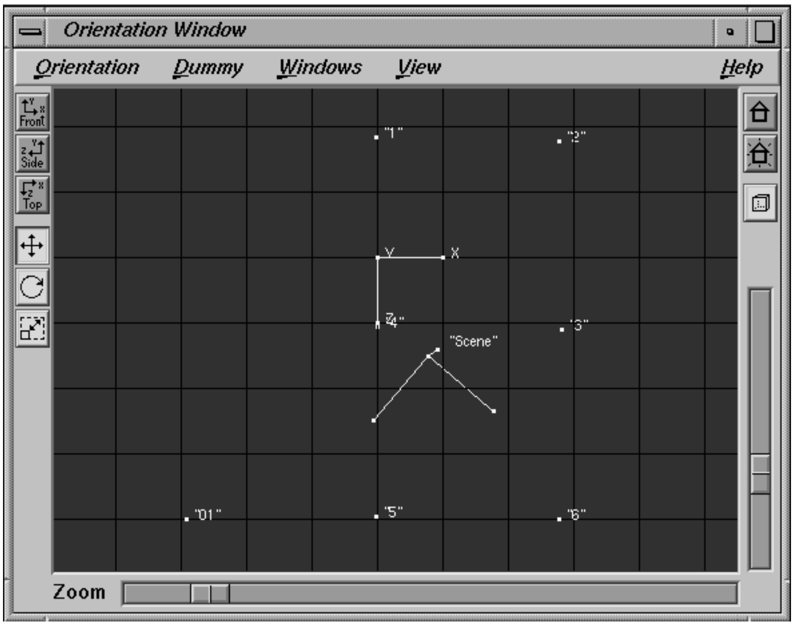
3D-Equalizer finally calculates a Lens Distortion of 0.052 and a Horizontal Angle of about 47.26. The average deviation has been reduced to 0.15 pixel.
Please take a last look at frame 135. As you see, the curvature of the floor-wall edge and the lens distortion grid are now matching quite well (see figure below).



Matching lens distortion

- *Open the Orientation environment and examine the 3D point positions again.*

They form squares like the tennis ball markers did in reality, because the camera parameters are now very precise (see figure below).



Well calculated points because of correct camera parameters

Tutorial 5: Stabilizing

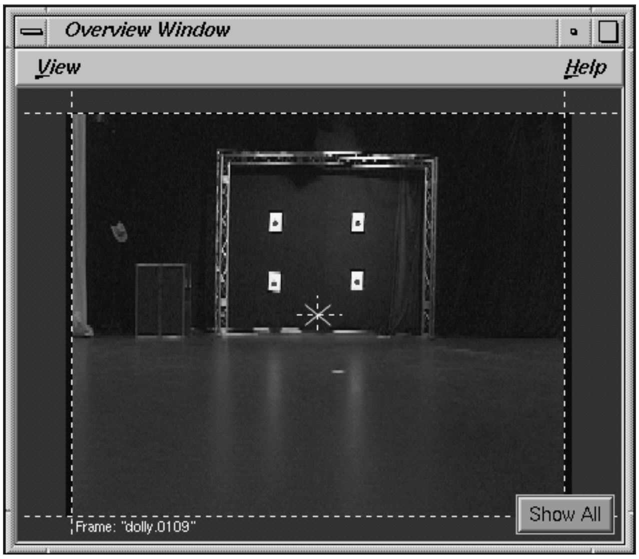
The purpose of this section is to point out several techniques for improving the quality of camera reconstruction. The situation presented below represents one of the most problematical types of matchmoving projects. The camera motion path is improved by the technique of stabilizing points, by editing the three-dimensional motion curves and by using the postfilter functionality.

Stabilizing Points

- *Please start 3D-Equalizer as described in the section “Getting started”, and open the project “dolly.3de” in the directory “<tutorial install directory>/projects/”.*

- *Select the environment “Motiontracking” from the Main Window’s environments menu.*

You see a black wall with four markers attached to it. The camera seems to be mounted on a dolly. It moves from the right to the left while facing the black wall. The project is already prepared, the four points on the wall are tracked. The fifth and the sixth point are defined in the first and the last frame, the camera geometry is correct, so that in principle the camera motion path can be calculated (see figure below).



Overview Window

- *Open the “Orientation” environment, and let 3D-Equalizer reconstruct the camera motion and point positions in 3D space by choosing the function Main Window::Calc::Calc All Objects From Scratch.*

After one or two minutes the calculation is over and the points appear in the Orientation Window display.

- *Open the Status Window and have a look at the average deviation value.*

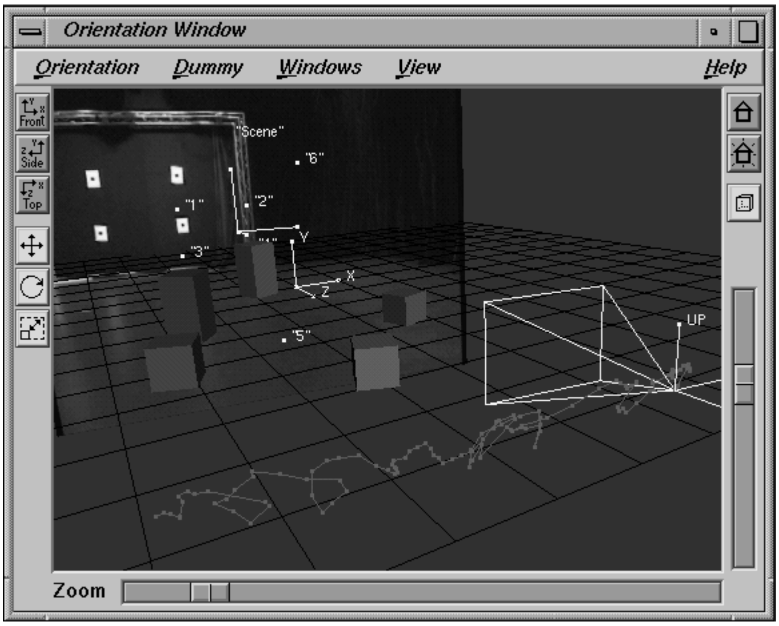
The average deviation is about 0.17 pixel. From the first tutorial you know that this value is quite good, therefore it seems that the camera motion path is consistent with the tracking curves.

- *Now open the “Preview” environment and press the button Main Window::Play.*

As you see, the preview looks absolutely terrible, the objects are fluctuating in a strange manner, although the recorded sequence looks very smooth and uniform.

- *Interrupt the preview playback by pressing the button Main Window::Stop.*

- *Open the “Orientation” environment again and activate the toggle Orientation Window::View::Show Camera Path. Use the functions of the Orientation Window in order to examine the path (see also figure below).*



Curly camera motion path

In this sequence the camera has been moved in a smooth path by means of a dolly, but the reconstructed camera performs strange curly movements. What is going on? In this project the optical axis of the camera is almost perpendicular to the black wall on which the markers have been mounted, i.e. there is not enough perspective information available. From a mathematical point of view this case is extremely sensitive to slightest fluctuations in the motiontracking curves. In other words, it is impossible to work precisely enough to achieve a reasonable camera motion path! In the following we shall see how the camera motion path can be stabilized. The basic principle is to supply more perspective information, e.g. by adding points which do not lie within the perpendicular plane.

- *Open the "Motiontracking" environment from the Main Window's environments menu.*

The marker of point "5" is located on the studio floor a few meters in front of the black wall. Let us try to stabilize the camera motion by tracking it.

- *Select point "5" in the Pointgroups Window and jump to the first frame of the sequence. Activate the toggle Zoom Window::Options::Auto Center.*

The Timeline Window indicates, that point "5" is invalid between the first and the last frame. We wish to track this point for the entire sequence, therefore you must make the point's tracking curve valid.

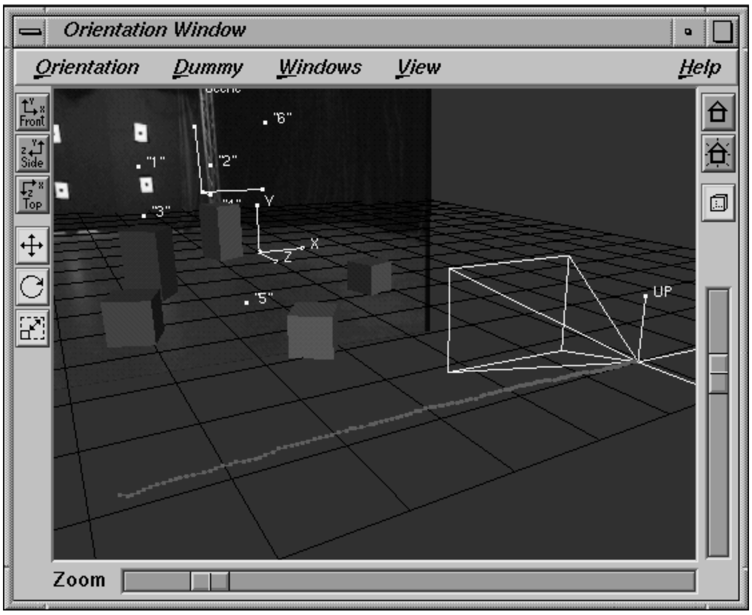
- *Press the button Zoom Window::End Point.*

The point's status in frame 1 is changed from endframe to startframe, making its tracking curve valid. The search area and the search pattern are already adjusted correctly and therefore we can start to track.

- *Press the button Zoom Window::Track in order to track point "5" until the last frame (the object on the floor is tracked as a marker).*

- *Choose the "Orientation" environment from the Main Window's environments menu and let 3D-Equalizer reconstruct the camera motion path again (Main Window::Calc::Calc All Objects From Scratch).*

After a minute, the points and the reconstructed camera motion path appear. The curly camera path has now become an almost straight line (see figure below).



Straight camera motion path

- *Open the "Preview" environment and and let 3D-Equalizer generate a preview movie again.*

Since the camera path has been stabilized by point "5" the movie looks much better now. However the cubes, in particular the blue one in the foreground, still perform some trembling movements. Obviously there is still a precision problem to be solved.

Important note: Since the preview movie has a lower resolution than the original images, it is useful to place the dummies such that inaccuracies in the reconstructed camera motion path are amplified. This can be done as demonstrated in this tutorial, i.e. by placing some of the dummies close to the camera but remote from the markers.

The next step is to check whether the tracking of point "5" has been performed correctly. Let's take another look at the Status Window.

- *Open the Status Window and examine the green deviation curve.*

Unfortunately, the green curve does not identify any problem areas. In some rare situations this can happen. In order to identify tracking errors you can then use the preview movie playback instead.

- *Set the playback range to 60 and 80, activate the toggle button Playback::Playback Mode::Swing and press Play.*

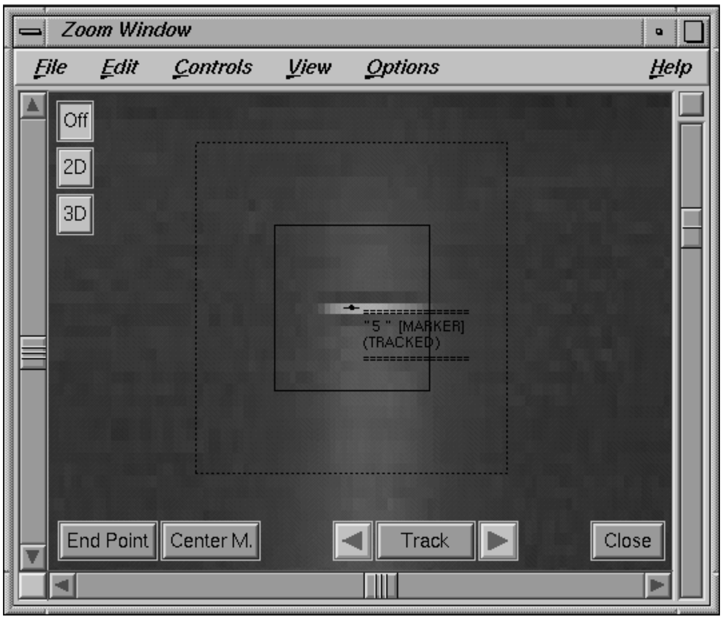
As you see in the middle of this frame range there is some jaggy movement of the objects.

- *In order to find out precisely which one of the frames causes this behaviour stop the playback, go to frame 60 and click on the button "+1" several times to examine the sequence frame by frame.*

As you see, from frame 69 to 70 the dummies move in a strange manner.

- *Open the "Motiontracking" environment, go to frame 70 and take a look at point "5".*

As you see, point "5" has not been tracked correctly in this frame (see figure below).



Point not tracked correctly

- *Simply correct the tracking point by clicking with the left mouse button on the center of the white marker.*

The tracking curve of point "5" in frame 70 has been corrected. In order to actually see the result we must let 3D-Equalizer reconstruct the 3D data. Editing the tracking of a point is just a minor modification, so it is not necessary to calculate everything from scratch.

- *Select the function Main Window::Calc::Finetune Selected Pointgroup.*

After a few seconds 3D-Equalizer has finished, and we can examine new results.

- *Open the “Preview” environment one more time, and press the button Play.*

After some processing time, the preview movie is generated between frame 60 and frame 80 as you specified by means of the playback range sliders. At least the jaggy movement at frame 70 has gone. As you see, this technique can be used for improving the quality of individual bad frames.

Editing the three-dimensional camera motion path

In the last section we learned how to improve the quality of a camera motion path by means of a stabilising point. However, what if there is no feature/marker visible in the foreground throughout the entire sequence? In this case it is possible to edit the three-dimensional motion path manually in order to improve its quality. In this section we demonstrate how this can be managed.

Let’s assume the marker, tracked by point “5”, isn’t visible between frame 40 and frame 79 and thus the point’s tracking curve is invalid in this frame range.

- *Please open the “Motiontracking” environment from the Main Window’s environment menu.*

- *Hold down the Alt-key, and click-drag the left mouse button between frame 40 and frame 79 on the tracking curve representation of point “5”, displayed in the Timeline Window.*

Point “5”’s tracking curve becomes selected between frame 40 and frame 79, indicated by an orange rectangle. These partially selected tracking curves are referred to as curve segments.

- *Choose the function Timeline Window::Edit::Delete Curve Segment.*

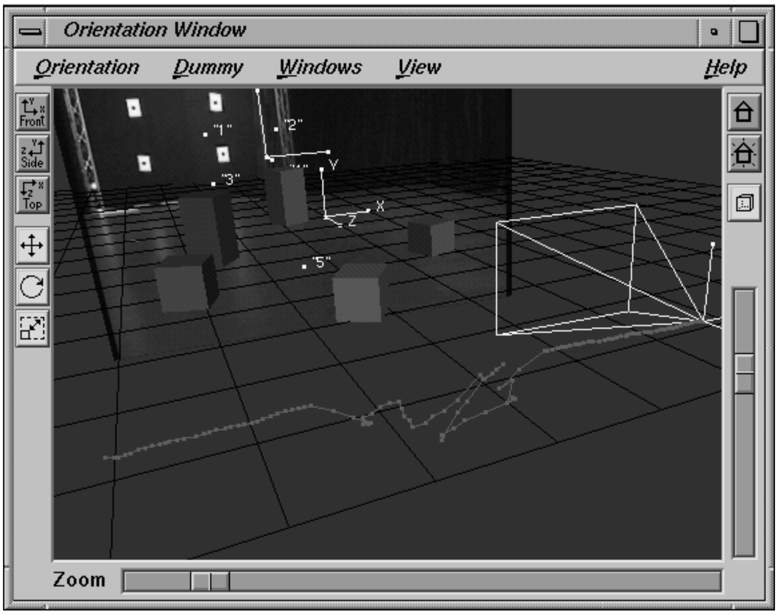
The selected curve segment is deleted from the point’s tracking curve. Point “5” is now invalid between frame 40 and frame 79.

- *Select the function Main Window::Calc::Calc All Objects From Scratch.*

3D-Equalizer reconstructs the camera motion path and the point positions in 3D space. Let’s take a look at the result.

- *Open the “Orientation” environment and activate the toggle button View::Show Camera Path.*

As you see, the camera motion path starts with a straight line, in the middle there is some curly movement and at the end it becomes a straight line again (see figure below).



Camera motion path partially corrupted

Obviously, 3D-Equalizer has problems reconstructing the camera motion path for those frames in which point “5” is invalid. (Please see also project “dolly2.3de”).

- *Open the “Preview” environment. Set the frame range sliders from frame 1 to frame 111 and press the button Main Window::Play.*

3D-Equalizer generates the preview movie, and afterwards plays it back. As you see, in the middle of the sequence the dummies perform some trembling movements, whereas at the end and at the beginning the motion is fine.

Now, let’s try to get rid of the trembling motion by directly editing the three-dimensional camera motion path.

- *Open the “Motion Path” environment from the Main Window’s environments menu.*

The large grey window labeled Motion Path Window allows you to edit the three-dimensional motion path of the currently selected pointgroup tracked in the currently selected sequence. For this purpose a coordinate frame in which you can separately edit six component curves is displayed. These component curves build a motion path in 3D space. In particular there are component curves for position x, position y, position z, rotation x, rotation y and rotation z. The basic usage of the Motion Path Window is very simillar to the Status Window, which we discussed in the first tutorial. With the middle mouse button you can move the coordinate frame around, with the right mouse button you can scale the coordinate frame, etc. For more information please take a look at the reference section.

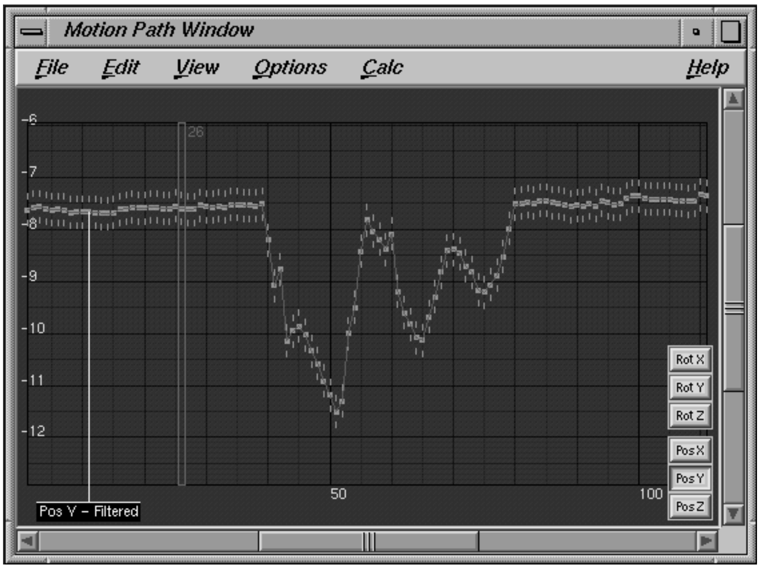
Let's start to edit a component curve. On the upper right edge of the window there are six toggle buttons labeled Rot X - Pos Z. These toggle buttons specify the component curve to be edited. Only one component curve can be edited at a time.

- *Click on each toggle button one after another and watch the Motion Path Window's coordinate frame.*

With each click another component curve appears in the window. On each frame of these curves there is a keyframe. Let's start to modify the motion path by editing the positional component curves.

- *Click on the Pos Y toggle button.*

The position y component curve becomes selected for editing and is displayed in the coordinate frame (see figure below).



Component curve position y

As you see between frame 40 and frame 79 the curve is quite jaggy, whereas outside of this frame range it is more or less straight. Let's get rid of the jaggy keyframes.

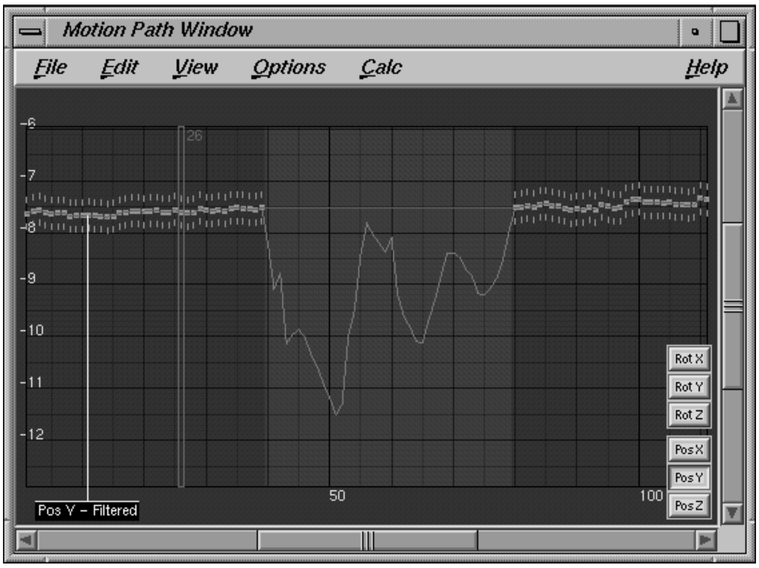
- *Activate the toggle button View::Show Raw Curves.*

- *Select the keyframes between frame 40 and frame 79 by click-dragging with the left mouse button.*

The selected keyframes become yellow to indicate their selection status.

- *Choose the function Edit::Delete Keyframes or simply press the Del-key on your keyboard.*

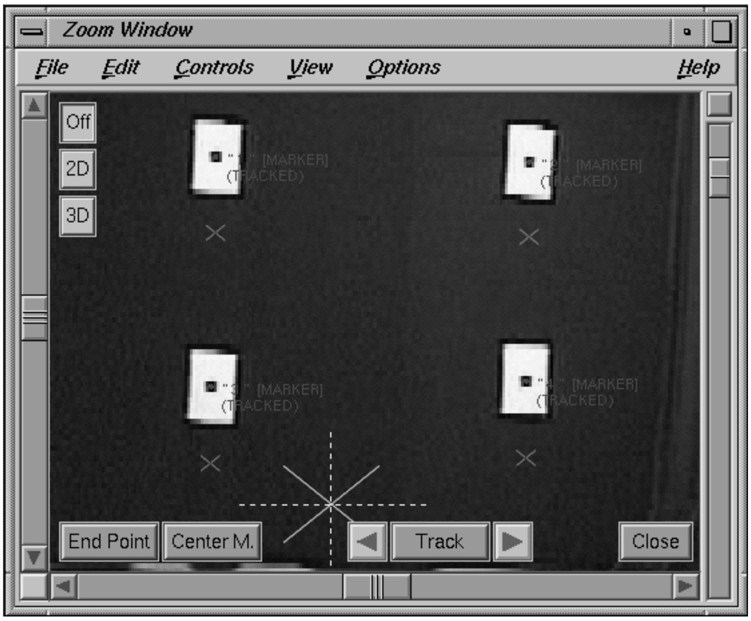
The selected keyframes are removed and the component curve becomes a straight line in this frame range. A grey curve becomes visible, which looks exactly like the component curve before it was edited. This grey curve is referred to as the raw curve. It serves as a reference for you to know how the original component curve looked before any editing action was applied. Finally, frames in which the component curve has been edited are marked in grey (see figure below).



Edited component curve postion y

- *Go to frame 65, activate the toggle button Zoom Window::View::Show Reconstructed Points::Distortion and take a look at the Zoom Window's display.*

As you see, for each valid tracking point a green cross is displayed. These crosses represent the reconstructed three-dimensional point position that is "seen" through the reconstructed camera in the current frame, taking lens distortion into account. For a precisely reconstructed camera each cross must sit almost exactly over its tracking point. However in our example, the green crosses are at least 30 pixels away from the respective tracking points (see figure below).



Misplaced reconstructed 3D points

What is going on? There is a tight relationship between the positional components and the rotational components of a motion path. By editing the position y curve, we have “injured” this relationship, so that the reconstructed camera motion path in this frame range became bad. We have to “fix” the relationship between the positional and rotational components.

- *Activate the toggle button Motion Path Window::View::Show Rotation Curves::All.*

Three additional curves appear on the coordinate frame, representing the component curves rotation x, rotation y and rotation z.

- *Choose the function Motion Path Window::Calc::Make Rotation Curves Consistent and take a look at the rotation curves as well as the Zoom Window’s display.*

The rotational component curves are now modified. The green crosses, displayed in the Zoom Window, are now located almost exactly over their tracking points, and thus the relationship between the positional and the rotational components of the camera motion path have been restored. This function is essential for editing 3D motion paths!

Now, let us continue to explore the curve editing functions provided by the Motion Path Window.

- *Make sure the position y curve is selected, hold down the Alt-key and click with the left mousebutton in the middle of the greyed area in the coordinate frame, approximately at frame 60.*

A new keyframe is inserted, and the position y curve is modified accordingly. By means of this editing action the relationship between the positional and the rotational curves is injured again, and the green crosses are not perfectly located over their tracking points.

- *Click-drag the newly created keyframe and move it around several times, while watching the resulting behaviour of the green crosses displayed in the Zoom Window.*

Each time you release the left mouse button, 3D-Equalizer updates the green crosses in the Zoom Window.

- *Activate the toggle button Options::Auto Consistency.*

- *Again click-drag the newly created keyframe and move it around several times. Watch the resulting behaviour of the green crosses displayed in the Zoom Window as well as the rotational curves displayed in the coordinate frame of the Motion Path Window.*

Each time you release the left mouse button, the rotational curves are updated, thus automatically restoring the relationship between them and the positional curves. This causes 3D-Equalizer to keep the green crosses positioned over their respective tracking points.

Now let’s continue to edit the two remaining positional component curves that build the 3D camera motion path.

- *Delete the recently created keyframe by selecting it with the left mouse button and pressing the Del-key on your keyboard.*

- *Activate the toggle button View::Show Rotation Curves::Selected.*

The rotational curves disappear from the coordinate frame.

- *Click on the Pos X toggle button.*

The position x component curve becomes selected and is displayed in the coordinate frame of the Motion Path Window.

- *Select and delete the “bad” keyframes between frame 40 and frame 79, like you did before with component curve position y.*

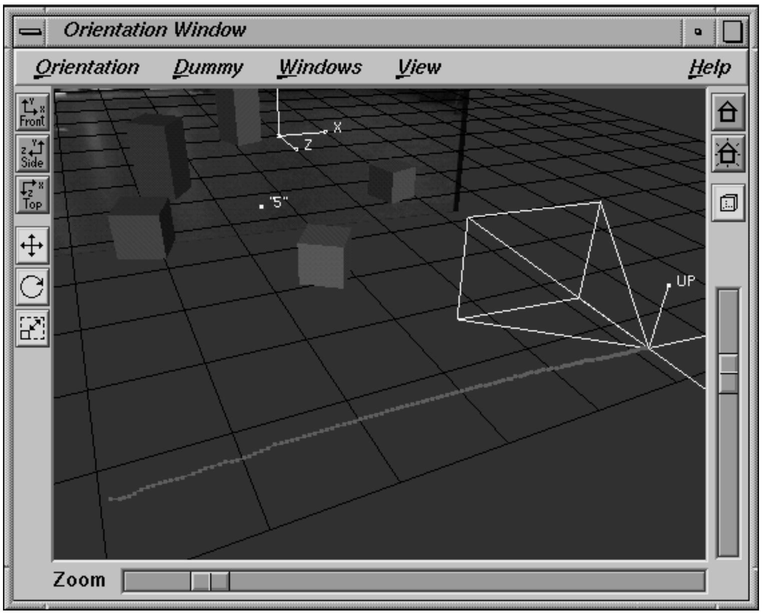
Since the toggle button “Auto Consistency” is still activated, the green crosses remain positioned over their respective tracking points.

- *Repeat the last steps with component curve position z.*

Now you have fixed the three positional component curves of the 3D camera motion path. The consistency functionality of the Motion Path Window, automatically ensured that the rotational component curves are fine as well. Let's take a look at the result.

- *Open the "Orientation" environment, and look at the 3D camera motion path.*

As you see, the resulting camera motion path now looks quite good (see figure below).



Fixed 3D camera motion path

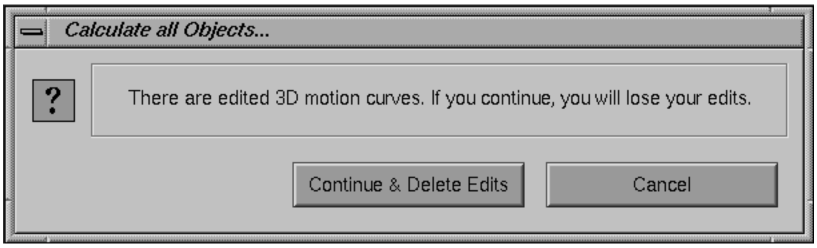
Finally let's examine the preview movie again.

- *Open the "Preview" environment, and press the button Main Window::Play.*

3D-Equalizer generates the preview movie again, and afterwards plays it back. The motion is quite good now. The trembling movements of the dummies in the middle of the sequence have been eliminated. (See also project "dolly3.3de").

- *Please try to calculate the 3D data again, by selecting the function Main Window:: Calc:Calc All Objects From Scratch.*

A requester opens, warning you that calculating the 3D data will 3D-Equalizer cause to delete the edited motion paths (see figure below)!



There are edited 3D motion curves. If you continue, you will lose your edits.

Warning requester

Important note: You should always try and avoid editing any three-dimensional motion paths. However, when it is absolutely necessary to edit one, it must always be the last step in your matchmoving workflow.

Applying the postfilter

In the previous sections we have seen that the reconstruction of a sequence can become jaggy. Now we shall see how this effect can be minimized.

- *Please open the project “dolly4.3de” in the directory “<tutorial install directory>/ projects/”.*

The project consists of the same sequence as in the previous examples. The four points at the wall and the additional stabilizing point on the floor are entirely tracked.

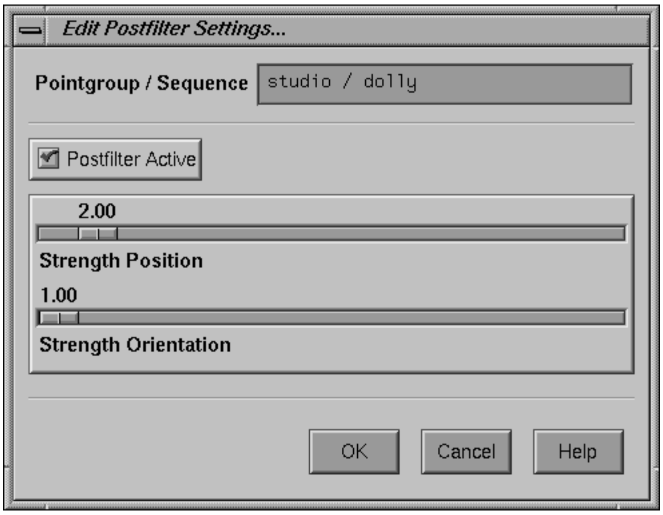
- *Select the environment “Motion Path” from the Main Window’s environments menu.*

- *Open the requester Edit Postfilter Settings by selecting the function Motion Path Window::Options::Postfilter Settings or Main Window::Options::Postfilter Settings.*

This requester allows you to adjust the filter settings for the 3D motion path of the currently selected pointgroup tracked in the currently selected sequence (see figure below).

- *Activate the toggle button Postfilter Active and press the button OK to close the requester.*

3D-Equalizer starts filtering the 3D motion path using the default filter settings, which only takes a few seconds.



Edit Postfilter Settings...

Pointgroup / Sequence: studio / dolly

☑ Postfilter Active

2.00
Strength Position

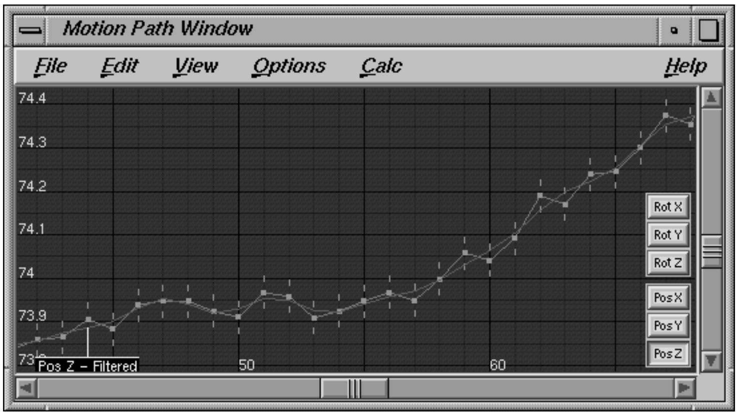
1.00
Strength Orientation

Requester: Edit Postfilter Settings

In the following you can investigate the relationship between strength and resulting filtered camera motion path.

- *Activate the toggle button Pos Z of the Motion Path Window and take a look at the displayed curves.*

As you see, there is a blue and a red curve displayed in the coordinate frame of the Motion Path Window. The blue curve represents the edit curve of the selected component postion z. The red curve represents the filtered edit curve (see figure below). During the filtering process the positional component curves may be extensively modified. Therefore 3D-Equalizer automatically makes the rotational component curves consistent with the positional ones, to avoid any bad “swimming” effects.



Filtered component curve using strength position 2.0

- *Open the "Preview" environment, and generate a preview movie.*

As you see, some of the fluctuations are removed, while others are still visible. That means the postfilter is too weak in this case.

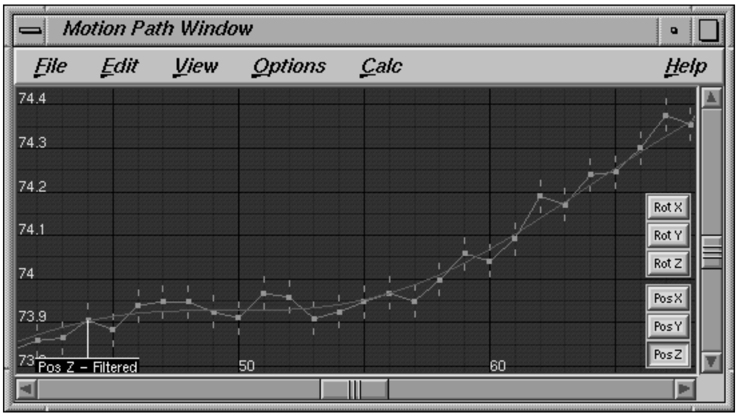
- *Open the "Motion Path" environment from the Main Window's environments menu.*

- *Open the Edit Postfilter Settings requester and change the slider Strength Position to some value, say, between 10 and 15. Press the button OK to close the requester, and take a look at the resulting curve.*

You will note that the filtered curve has become much smoother (see figure below).

- *Open the "Preview" environment, and generate a preview movie one more time.*

As you see the preview looks very smooth with this setting.



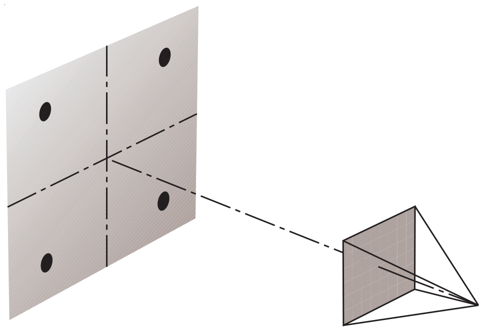
Filtered component curve using strength position 10.0

Important note: The strength of the postfilter to be used depends on the type of camera equipment. If the real camera motion path is smooth, e.g. because a dolly or a crane has been used you can set the strength to some higher value. However, the motion paths should always be checked in the Orientation Window and the Preview Window. If the sequence is recorded by a handheld camera, the strength of the postfilter should not be too high, so that the reconstructed camera motion path is not falsified.

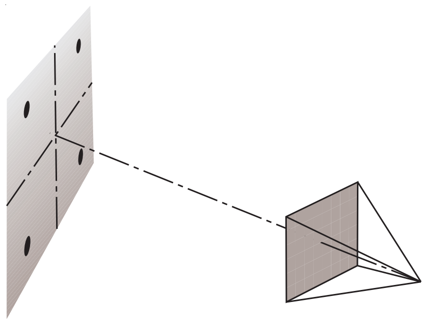
In addition to the position filter there is also a filter function for the orientation. Filtering the orientation means that the 2D tracking curves are smoothed out collectively. You can use it in order to compensate for 2D tracking errors. The effect of filtering the orientation is much lower than the effect of filtering the position. Therefore, when you apply postfiltering, adjust the filter Strength Position first and the filter Strength Orientation afterwards. You can provoke a situation which enables you to study various filter effects by misplacing a tracking point in one single frame by approximately 10 pixels and playing around with the filter settings.

General remarks

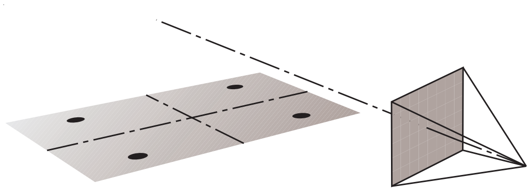
You have seen that the situation in which the optical axis is perpendicular to the plane with markers can bear some problems, and how these problems can be eliminated step by step. In practice you may try to avoid these situations: rearrange the wall and the markers or place the markers in a completely different way as illustrated in the figures below.



Problematic: optical axis perpendicular to markers on a wall



Not problematic: optical axis not perpendicular to markers on a wall



Not problematic: optical axis not perpendicular to markers on the floor

Tutorial 6: Fixed Camera Position

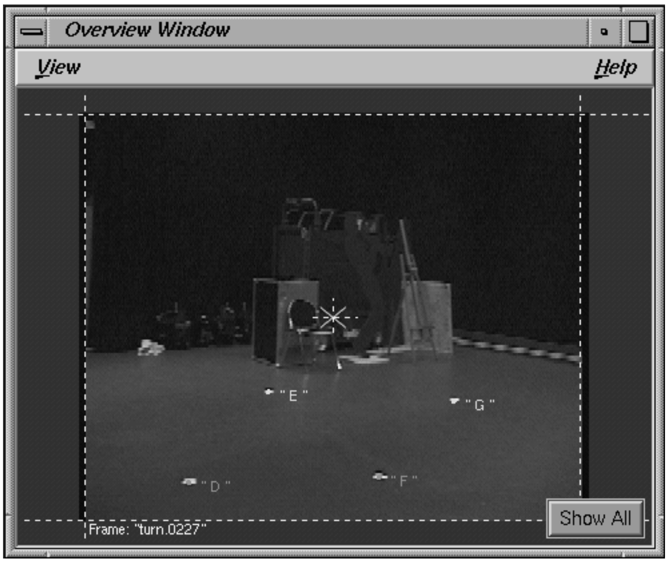
Frequently there are projects in which the camera position does not change over time, e.g. if the camera is attached to a tripod. In these cases the camera is only rotating around a fixed point.

These quite “simple” matchmoving projects can become quite difficult using the regular calculation method because the key condition, a varying camera position during the shot, is not fulfilled! Using reference frames is one possibility to solve the problem (see section Tutorial Reference Frames). In this section however, we will learn another technique for handling this type of project: the fixed camera position mode.

- *Please start 3D-Equalizer as described in the section “Getting started” and open the project “fixed_cam_position.3de” in the directory “<tutorial install directory>/projects/”.*

- *Open the environment “Motiontracking” from the Main Window’s environments menu.*

You see a TV studio with sixteen markers on the floor. The camera is attached to a tripod. In the course of the sequence it turns around by an angle of about 120°. The camera position remains unchanged, or at least almost unchanged. Sixteen points have already been tracked (see figure below).



Overview Window

Now let’s reconstruct the camera motion using fixed camera position mode. This calculation mode is assigned to each separate sequence. That means, you can create projects which contain “fixed camera postion” sequences and “non fixed camera position” sequences at the same time.

- *Open the requester Sequence Settings for the image sequence “turn” by double clicking on its entry in the Frames Window.*

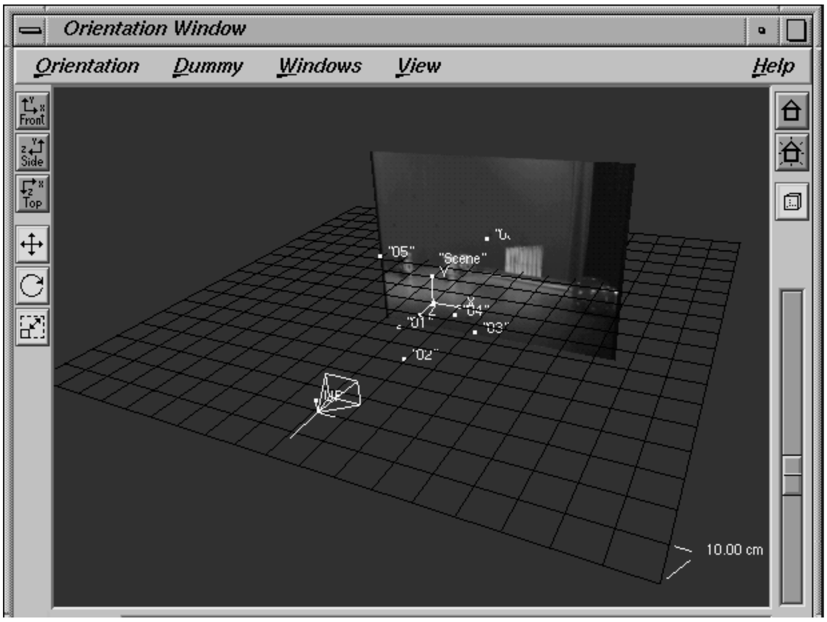
- *Set the option menu Camera Constraints::Position to “Fixed Camera Position Constraint” and press the button Ok to confirm your edits.*

The sequence’s “fixed camera postion” mode is now active, which is indicated by the label “Constraint” in the Frames Window. The camera parameters are already well defined, so that we can reconstruct the camera motion.

- *Select the function Main Window::Calc::Calc All Objects From Scratch.*

3D-Equalizer reconstructs the camera motion, which takes some time to complete. Let’s take a look at the result.

- *Open the “Orientation” environment from the Main Window’s environments menu and take a look at the reconstructed data.*



Points projected on a virtual sphere due to fixed camera position method

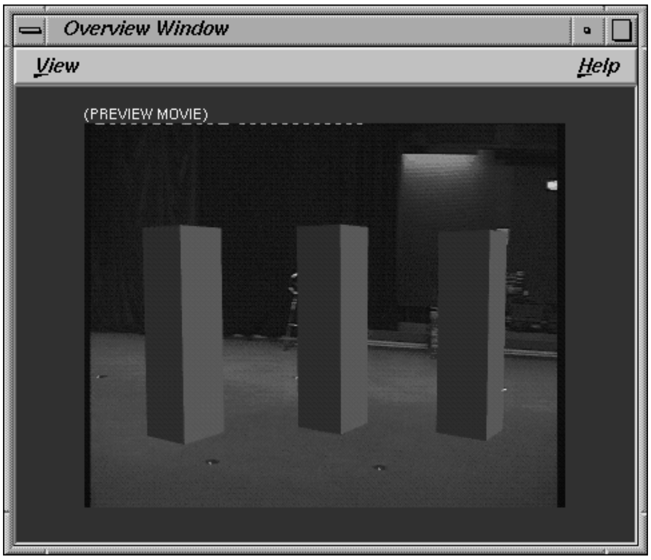
You will notice that something strange is going on. The camera does not change its position, and rotates around its "center of projection" point. However, although all markers are placed on the studio floor and thus lie in the same plane, the reconstructed points are distributed in space. More precise, they lie on the surface of a virtual sphere (see figure above).

In this example the fixed camera position mode does not allow us to calculate "real" 3D point positions, because the studio floor is only filmed from a single point of view. The distance information between the camera position and each point is missing and cannot be recovered.

Fortunately, this lack of depth information does not cause many problems. Due to the fact that the camera position does not change over time, the distance between the dummy objects and the camera does not affect the representation as far as the perspective is concerned. In our example, the dummy objects are placed at different distances and different size. Let's try to prove this.

- *Open the "Preview" environment, and press the button Play.*

After a short period of time, in which 3D-Equalizer generates the preview movie, the sequence is played back in realtime. Although the three dummy objects are located at different distances from the camera the preview looks correct (see figure below).



Preview movie

So, the missing accuracy in the point positions is no problem, since the dummy objects (or the models in the final animation package) can be placed manually "by eye". If, for some reason, it is absolutely necessary to reconstruct the accurate point positions in 3D space, reference frames <u>must</u> be used in addition (see section Tutorial Reference Frames).

Tutorial 7: Reference Frames

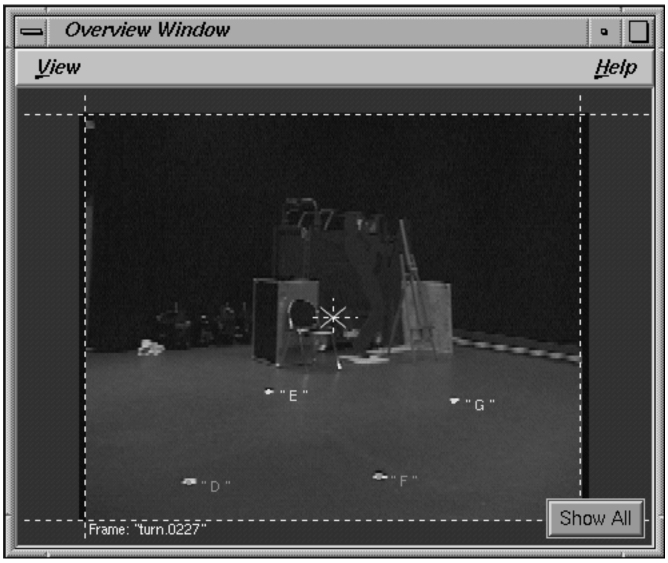
In the following, we again take up the previous example. However, this time we will provide more information to the system in order to obtain accurate point positions in 3D space. We achieve this by introducing the concept of reference frames.

In practice it often happens that the camera used for recording the sequence is heavy, unwieldy or even part of some big installation. In this case it cannot be moved to different points of view without problems. However, reference frames can be taken with an entirely different camera. In the second part of this section, we will present a sequence that contains enough camera movement to obtain a reasonable camera motion path, but no precise point positions. By using reference frames, taken with a 35mm photo camera, the quality of the point model is considerably enhanced.

- *Please start 3D-Equalizer as described in the section "Getting started" and open the project "turn1.3de" in the directory "<tutorial install directory>/projects/".*

- *Open the environment "Motiontracking" from the Main Window's environments menu.*

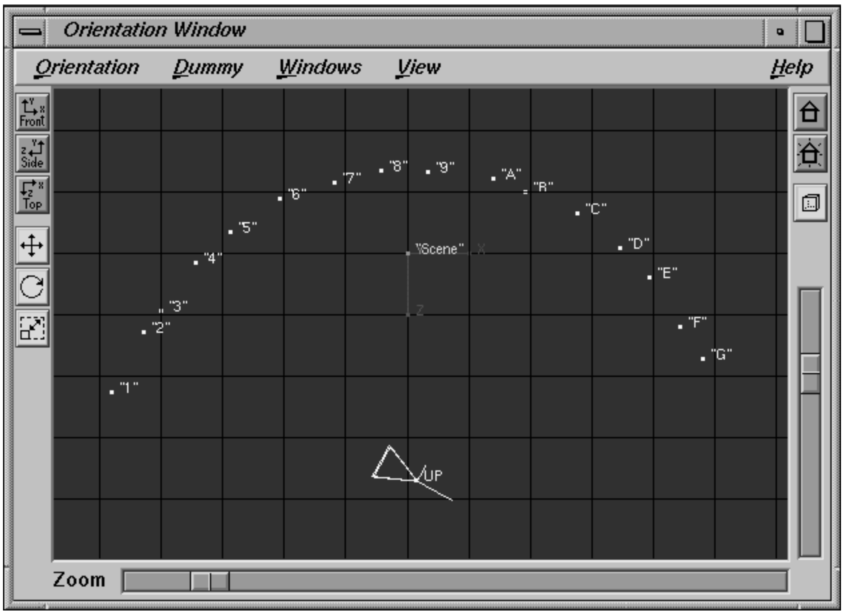
You see a TV studio with sixteen markers on the floor. The camera is attached to a tripod. In the course of the sequence it turns around by an angle of about 120°. The camera position remains unchanged, or at least almost unchanged, and thus the sequence object "turn" is set to fixed camera position mode. Sixteen points have been tracked. The camera parameters are already well-adjusted (see figure below).



Overview Window

- *Open the “Orientation” environment and select the function Main Window::Calc::Calc All Objects From Scratch.*

After a few seconds the calculation is complete and the points appear in the Orientation Window display. As you see, they are placed on a virtual sphere due to the fixed camera position mode that is enabled in the image sequence (see figure below).



Points projected on a virtual sphere due to fixed camera position mode

What if the points positions need to be accurately calculated in 3D space? By the use of reference frames you can supply 3D-Equalizer with the needed “perspective information” to reconstruct proper point positions.

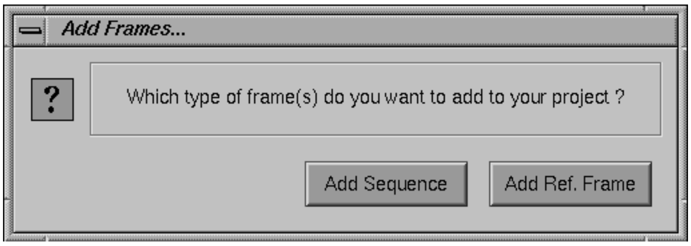
Defining reference frames

Let’s supply 3D-Equalizer with two additional reference frames.

- *Open the “Motiontracking #3” environment.*

- *Click on the button Add of the Frames Window.*

A requester opens, asking if you want to create a new sequence or a new reference frame (see figure below).



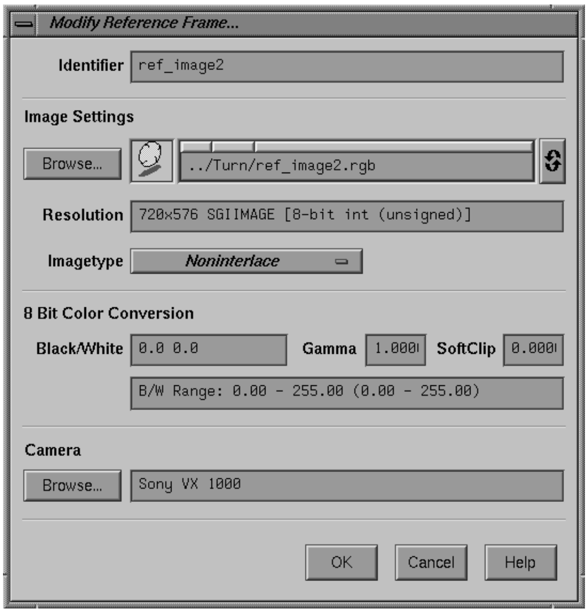
Add Frames...

Which type of frame(s) do you want to add to your project ?

Requester: Add sequence or reference frame?

- *Since we want to create a new reference frame, click on the button Add Ref. Frame.*

The requester Create Reference Frame opens, in which you can define the attributes of the reference frame to be created (see figure below).



Modify Reference Frame...

Identifier: ref_image2

Image Settings

../Turn/ref_image2.rgb

Resolution: 720x576 SGIIMAGE [8-bit int (unsigned)]

Imagetype: Noninterlace

8 Bit Color Conversion

Black/White: 0.0 0.0 Gamma: 1.000 SoftClip: 0.000

B/W Range: 0.00 - 255.00 (0.00 - 255.00)

Camera

Sony VX 1000

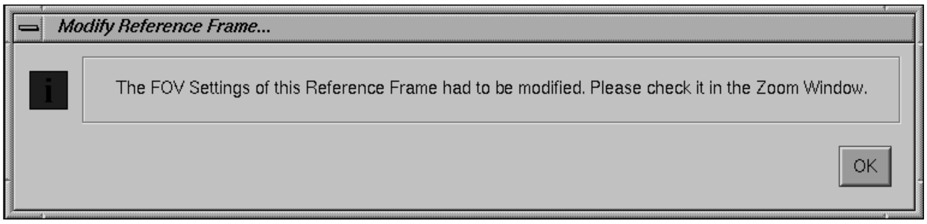
Requester: Create Reference Frame...

- *Click on the button Image Settings::Browse and select the image file “ref_image2.rgb”, found in the directory “<tutorial install directory>/Turn/”.*

The images named “ref_image#.rgb” have been recorded with the same camera as the image sequence, but they are non-interlaced. Therefore the selector Image Type should be left to Noninterlace.

- *Press the button OK, to confirm your edits and close the requester.*

The image is loaded and displayed in the Zoom- and in the Overview Window. Furthermore, an info requester opens, telling you that the FOV setting of the newly create reference frame had to be modified (see figure below).



Modify Reference Frame...

The FOV Settings of this Reference Frame had to be modified. Please check it in the Zoom Window.

Info requester

Why does this happen? Our project now contains two “image objects”, the sequence “turn” and the reference frame “ref_image2”, that are linked to the same camera object (“Sony VX1000”). 3D-Equalizer must ensure that at least the FOV ratio is exactly the same for each image linked to this camera. Therefore it automatically modified the FOV ratio of our newly created reference frame in order to make it “compatible” to the FOV ratio of the already existing sequence.

Let’s continue our work on this project by adjusting the FOV of the reference frame. 3D-Equalizer prevents you from modifying the FOV ratio, because of the reasons pointed out above. Therefore, you can only edit the FOV ratio at the corners of the FOV from now on.

- *Hold down the Ctrl-key, click-drag on the upper-right corner of the reference frame’s FOV and modify it until the right FOV line matches the image’s film gate.*

- *Hold down the Ctrl-key, click-drag on the upper-left corner of the reference frame’s FOV and modify it until the upper FOV line matches the image’s film gate.*

- *Hold down the Ctrl-key, click-drag on the lower-left corner of the reference frame’s FOV and modify it until the left and the lower FOV lines match the image’s film gate.*

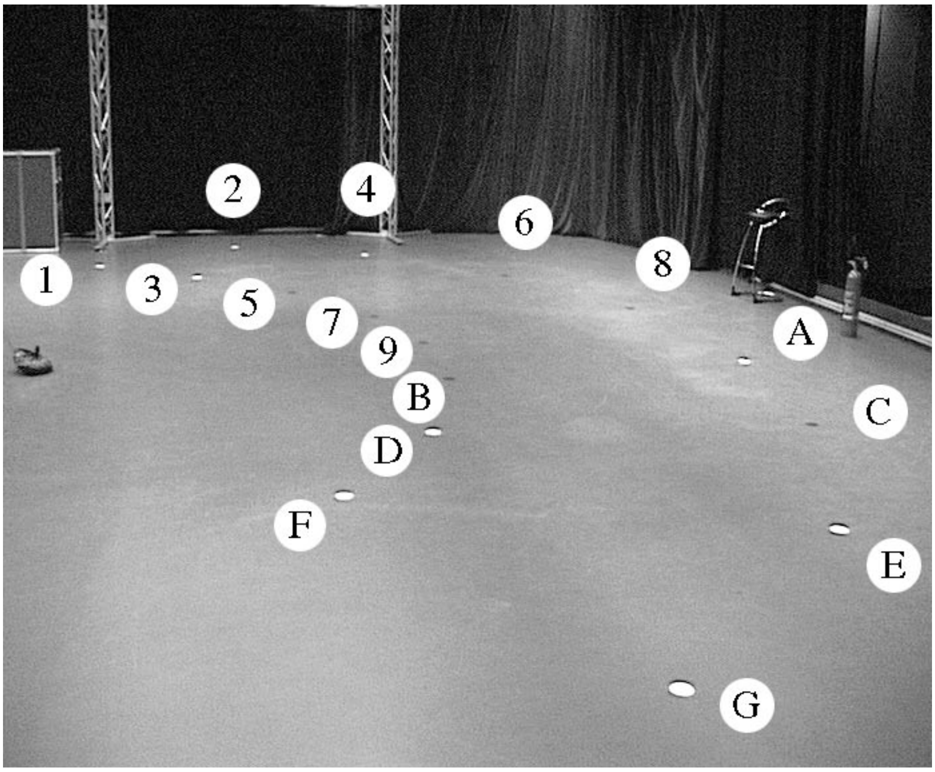
Now, the FOV should be well adjusted. In principle, it is sufficiently to work with “only” one reference frame in conjunction with a sequence. However, introducing a few more reference frames usually gives you more precision.

- *Therefore please repeat the last steps and add the reference frame “ref_image7.rgb”, found in the same directory.*

The two frames are recorded from different points of view. In order to include all markers on the floor the second frame was recorded with a roll angle. A general principle in the concept of reference frames is, that one should try to include as many points as possible in each of the frames, so that more information is provided to the system.

- *Now place the points on the markers for both of the reference frames. In order to let 3D-Equalizer identify the center of a marker, you can still use the button Center M.*

The points cannot be tracked as in the sequence frames, because reference frames do not represent a "continuous" succession in time. Therefore all points must be placed by hand. Take care that the same points are associated to the same markers in both reference frames and sequence frames, as shown in the figure below.



Point object - marker assignment

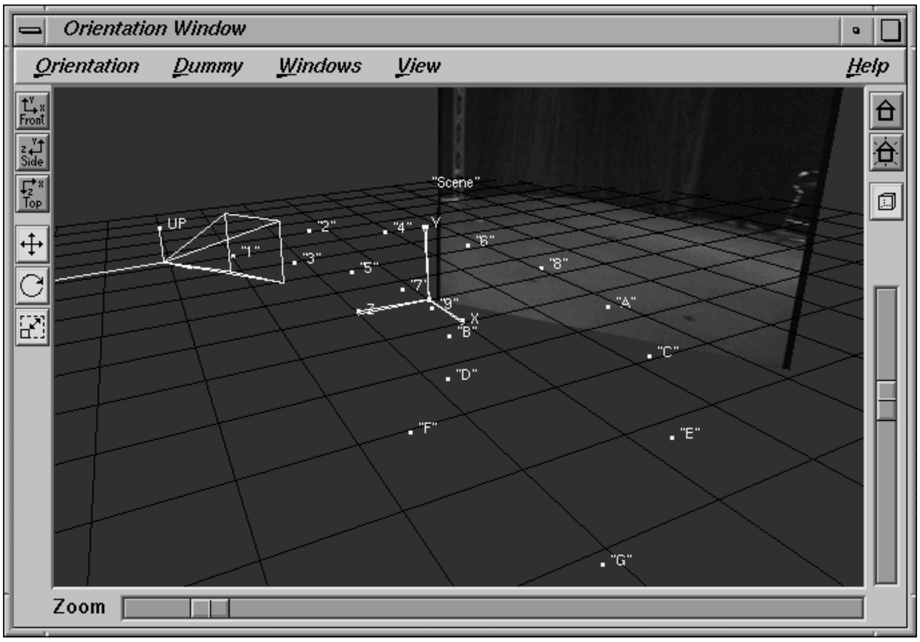
Now you should have a project with two reference frames containing sixteen points. The project "turn2.3de" found in the directory "<tutorial install directory>/projects/", contains all steps we have taken up to now. Please feel free to review it at any time.

Let’s examine the effect of the newly introduced reference frames on our point model.

- *Open the “Orientation” environment from the Main Window’s environments menu.*

- *Select the function Main Window::Calc::Calc All Objects From Scratch and take a look at the Orientation Window’s display.*

After one or two minutes the points appear in the display. In contrast to the original project they now almost lie in a plane because 3D-Equalizer was able to calculate them precisely by means of the reference frames.



Well-calculated 3D point positions due to reference frames

- *Go ahead and place some dummy objects on the virtual floor using the functions of the Orientation Window as described in the first tutorial. If you are not interested in this, you can also open the project “turn3.3de” found in the directory “<tutorial install directory>/projects/”.*

- *Open the “Preview” environment and press the button Play.*

After some time the preview movie is complete and you can examine it. If you have placed the points in the reference frames precisely enough, the movement of the dummies should be smooth and uniform.

Checking the average deviation value

Finally, let’s take a look at the quality of the reconstructed data.

- *Please open the “Status” environment, and take a look at the average deviation value of the sequence.*

As you see the average deviation is about 1.0 pixel. In comparison to other projects this value seems quite high. There are mainly two reasons for that: the fixed camera position mode and a lack of precision due to the reference frames.

- *While the sequence “turn” is selected, press the button Frames Window::Modify in order to open the requester Modify Sequence. Set the option menu Camera Constraint::Position to “No Constraint” in order to deactivate the fixed camera position mode and confirm your edits by pressing the button OK.*

- *Select the function Main Window::Calc::Calc All Objects From Scratch, and take a look at the average deviation value displayed in the Status Window after the calculation process has been completed.*

Now that the fixed camera position mode is deactivated, the average deviation has been reduced to 0.45 pixel. The reason for this is that the fixed camera position mode forces 3D-Equalizer to keep the center of projection of the camera at exactly the same position. However, in reality the rotation center point must not be positioned exactly over the center of projection of the real camera. In our example, this is the case. By disabling the fixed camera position mode, 3D-Equalizer is able to “minimize” the error even more, which results in a slighty animated camera motion path (see figure below). Let’s take a final look at the reference frames.

- *Open the “Orientation” environment and select the two reference frames one after another in the Frames Window while watching the Orientation Window’s display.*

Unfortunately, the reference frames show the studio floor with the attached markers from approximately the same points of view, which results in a fairly good, but not optimum, point model.

- *Open the project “turn4.3de” which you can find in the directory “<tutorial install directory>/projects/”.*

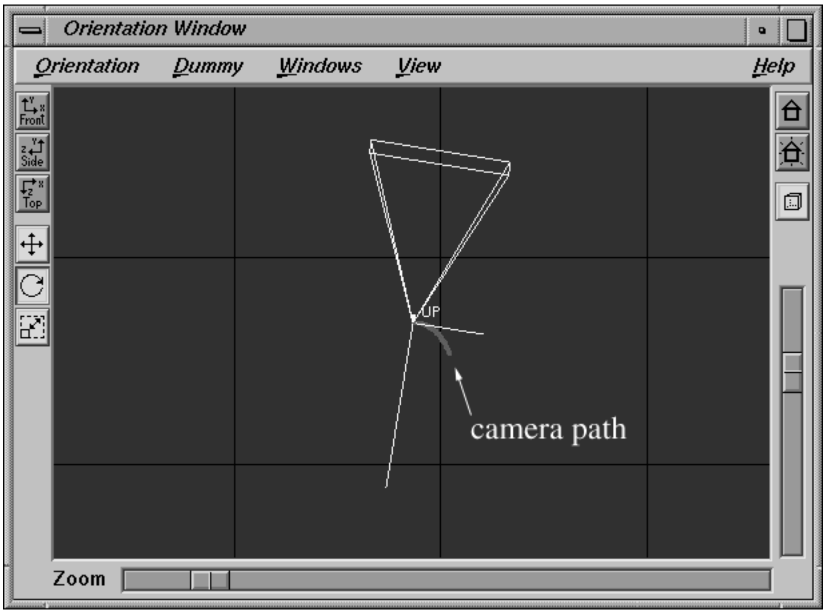
Project “turn4.3de” is nearly the same as the previous project. However, the fixed camera position mode is disabled and one reference frame image has been replaced.

- *Select the two reference frames one after another in the Frames Window while watching the Orientation Window’s display.*

As you see, the 3D cameras of the two reference frames are oriented nearly perpendicular to each other. That means, these reference frames supply 3D-Equalizer with much more “perspective information” than the previous set of reference frames.

- *Open the Status Window, select the sequence “turn” in the Frames Window and take a look at the average deviation value.*

As you see, the average deviation is only 0.27 pixel, which is a much better value.



Slightly animated camera motion path

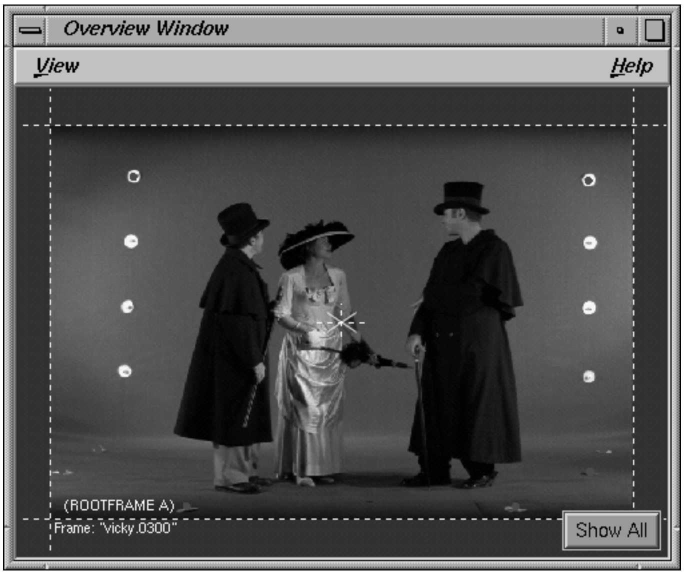
Improving the quality of the point model

We will now investigate a project which, at first sight, contains enough camera movement for doing matchmoving, but which turns out to yield badly calculated point positions. By providing reference frames, this problem is removed.

- *Please start 3D-Equalizer as described in the section "Getting started" and open the project "vicky1.3de" in the directory "<tutorial install directory>/projects/".*

- *Open the environment "Motiontracking" from the Main Window's environments menu.*

Three actors are standing in front of a bluebox environment. Take a look at the sequence by clicking a few times on the right hand side of the Main Window's frame slider. The camera only moves slightly upwards (see figure below).



Overview Window

- *Open the "Camera Adjustment" environment.*

Because there is very little camera movement (change of camera position), 3D-Equalizer will not be able to properly adjust the camera. Let's prove that.

- *Set the horizontal angle value in the Camera Adjustment Window to 45 degree. Select the function Main Window::Calc::Calc All Objects From Scratch and take a look at the average deviation value displayed in the Status Window.*

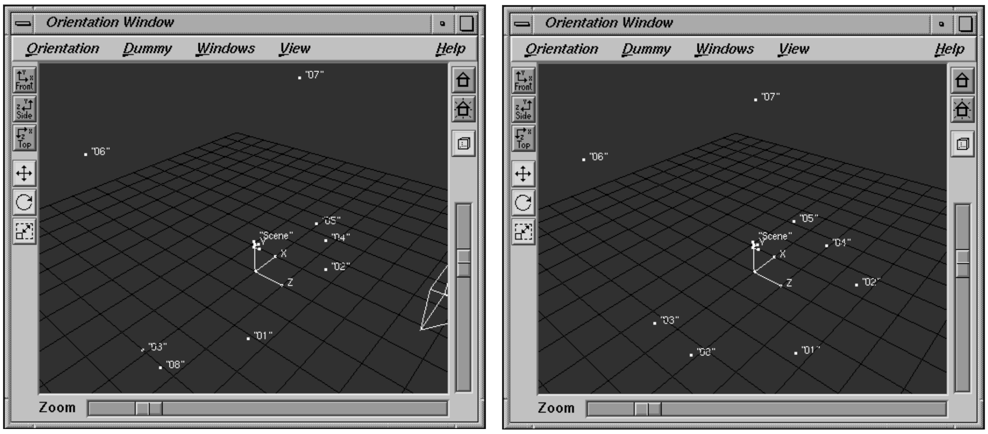
After some time, the calculation process has finished. The average deviation is about 0.187 pixel, which is a very good value. Let’s try a more wide-angle kind of camera.

- *Set the horizontal angle value in the Camera Adjustment Window to 70 degree. Select the function Main Window::Calc::Calc All Objects From Scratch and take a look at the average deviation value displayed in the Status Window.*

3D-Equalizer completes the calculation process with an average deviation value of 0.193 pixel, which is a very good value, too. Let’s try a narrow-angle kind of camera.

- *Set the horizontal angle value in the Camera Adjustment Window to 30 degree. Select the function Main Window::Calc::Calc All Objects From Scratch and take a look at the average deviation value displayed in the Status Window one more time.*

Now the average deviation is about 0.184 pixel. Still a very good value. It seems that almost every camera setting results in an excellent reconstructed solution, if we only take a look at the average deviation value. The reason why 3D-Equalizer will not be able to perform a sucessful camera adjustment is that every camera is “good”. This is because there is only little camera movement. Let’s take a look at the reconstructed point models (see figure below).



Pointmodel reconstructed with 70 degree, respectively 30 degree horizontal angle

The pointmodel reconstructed using a 70 degree horizontal angle looks totally different from the one reconstructed using 30 degree horizontal angle. As already mentioned in the previous tutorial “Camera Adjustment”, to reconstruct a proper pointmodel you need to have correct camera parameters. We will see in the course of this tutorial, that the correct horizontal angle is about 27 degrees. However, we will see that, even if we enter the correct value, the pointmodel gets distorted because there is not enough camera movement.

Adding Reference Frames

In the following section of this tutorial we will insert three reference frames taken with a regular photo camera. First of all, let’s create a new camera object.

- *Please press the button Camera Adjustment Window::Import to open the Import Camera file requester. Select the file “Size_135_Photo.3decam”.*

A new camera object is created and filled with the data stored in the selected file. It represents the standard settings of a regular photo camera (filmback 36mm x 24mm, focal length 50mm). Let’s continue to add a few reference frames.

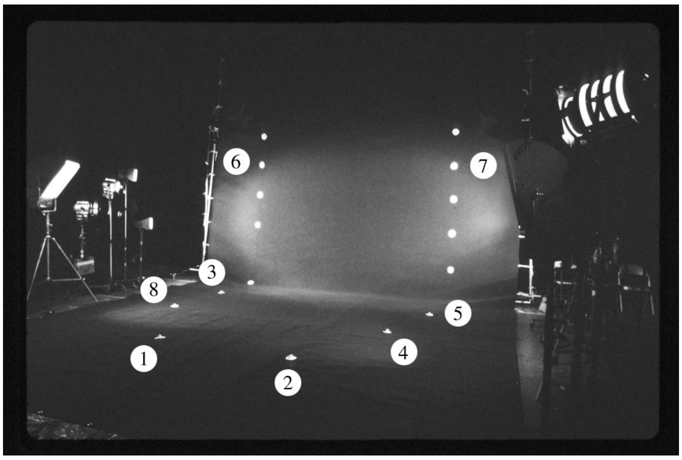
- *Open the “Motiontracking #3” environment for the Main Window’s environments menu.*

- *Press the button Frames Window::Add and click on Add Ref. Frame. Press the Image Settings::Browse button and select the image file “ref_vicky1.jpg” found in the directory “<tutorial install directory>/Vicky/”.*

Finally you have to specify the camera that should be linked to the new reference frame.

- *Press the button Camera::Browse and double-click on the entry “Size 135 Photo”.*

- *To confirm your edits press the button Ok.*



Markers 1-8

A new reference frame is added to your project, linked to the recently created camera object “*Size 135 Photo*”.

- *Repeat the last steps in order to create two additional reference frames “ref_vicky2.jpg” and “ref_vicky3.jpg”.*

Next step is to set half of the tracking points in the reference frames (see picture above).

- *Select each reference frame one after another in the Frames Window. Define the tracking points “03”, “05”, “06” and “07” in each of the reference frames.*

- *Specify the FOV for each reference frame using the left mouse button while holding down the Ctrl-key (see previous section).*

As you have seen, the three reference frames show the bluebox studio from highly different points of view. Therefore the camera object that is linked to them is much more “important” then the camera object linked to the sequence. We have to be careful about this. <u>“Important” cameras must be located above less “important” cameras in the camera list widget of the Camera Adjustment Window.</u>

- *Open the “Camera Adjustment” environment and take a look at the camera list widget.*

As you see the camera “Size 135 Photo” that is linked to our reference frames is located below the camera linked to the sequence. Let’s change that.

- *Select the camera “Size 135 Photo” in the camera list and press the button Camera Adjustment Window::Up.*

The order of cameras is good now. The project “vicky2.3de” found in the directory “<tutorial install directory>/projects/”, contains all steps we have taken up to now. Please feel free to review it at any time.

Distorted point model

In the following section you will see that although both cameras are well defined, the point model gets distorted. First of all, let’s enter the correct camera settings.

- *Make sure the camera “Size 135 Photo” is selected in the Camera Adjustment Window and enter the value 25.0 mm in the Focal Length textfield.*

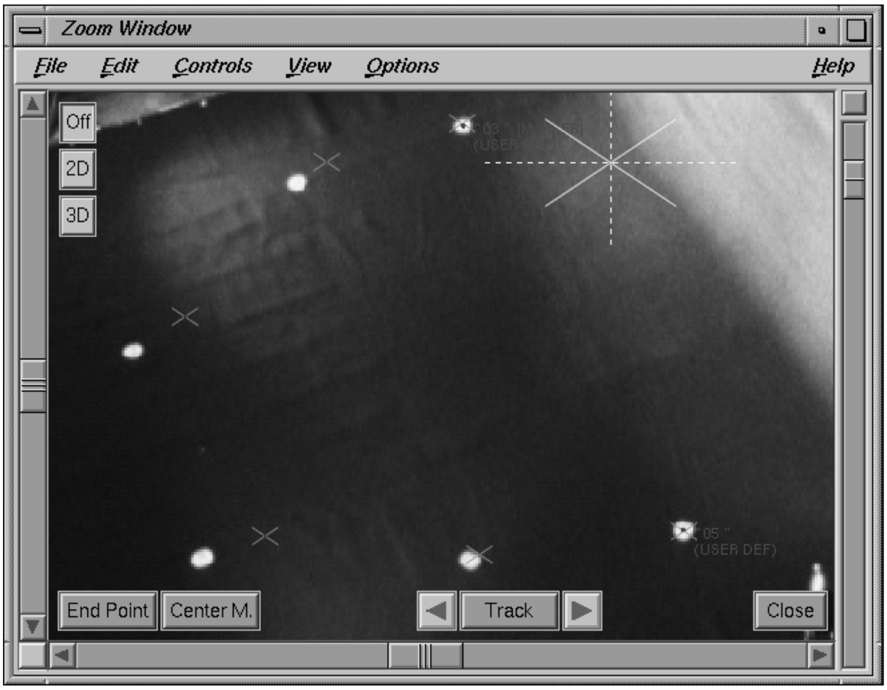
- *Select the camera “35mm Film” and enter 27.0 degree in the Horizontal Angle textfield.*

As you will see later, both cameras are correct now.

- *Select the function Main Window::Calc::Calc All Objects From Scratch, in order to reconstruct the point model.*

- *Open the "Motiontracking #3" environment, select the reference frame "ref_vicky2" in the Frames Window and activate the toggle button Zoom Window::View::Show Reconstructed Points::Distortion.*

As you can see in the Zoom Window's display, the reconstructed three-dimensional point positions (green crosses) of those tracking points defined in the reference frames (points "03", "05", "06" and "07") match the markers quite well. However, the four remaining points are substantially offset from their respective markers (see figure below).



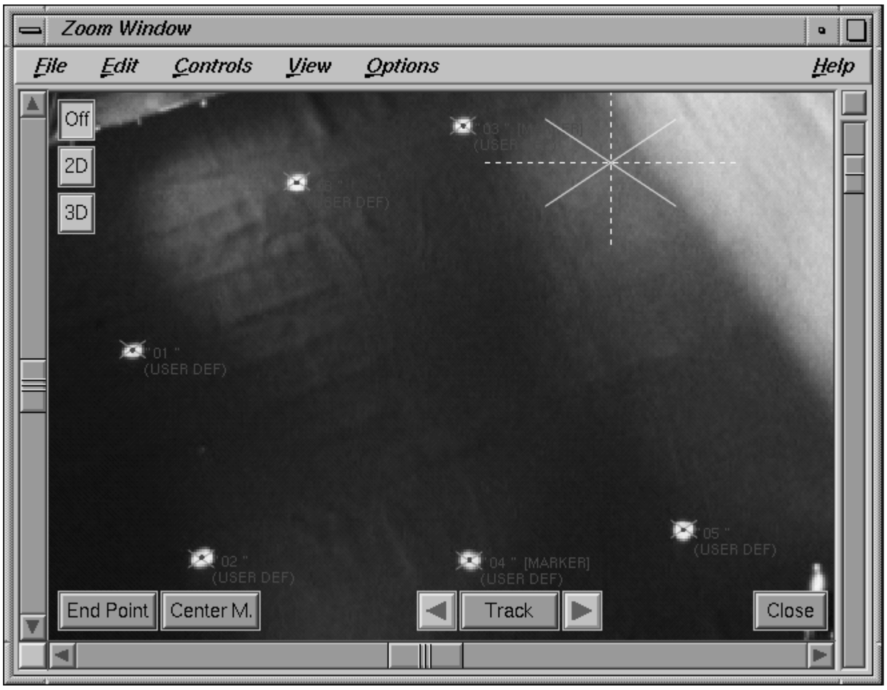
Distorted point model

These points are "only" calculated by means of the sequence frames. As you already know, the sequence camera changes its position only slighty throughout the shot. Therefore the remaining points of the point model become distorted.

- *Select each reference frame one after another in the Frames Window and define the remaining tracking points "01", "02", "04" and "08" in each of the reference frames.*

- *Select the function Main Window::Calc::Calc All Objects From Scratch again and take a final look at reference frame "ref_vicky2".*

As you see, the reconstructed point positions of the remaining points are now also correct, although we haven't changed the camera settings at all (see figure below).



Precise point model

Adjusting the cameras

At the beginning of this tutorial it was not possible to adjust a camera, because there was not enough "perspective information" with the image sequence. In the meantime we have added more information from three reference frames. Let's find out if this is enough information to perform a camera adjustment with 3D-Equalizer.

- *Open the "Camera Adjustment" environment for the Main Window's environments menu.*
- *Make sure the camera "Size 135 Photo" is selected in the Camera Adjustment Window and set the option menu Focal Length to Unknown.*

- *Select the camera “35mm Film” and set the option menu Horizontal Angle to Unknown as well.*

- *Press the button Adjust, in order to start the camera adjustment procedure.*

A requester opens, asking whether you would like to adjust all cameras at once, or if you would like to adjust only the selected one.

- *Press the button “Adjust all Cameras”.*

The camera adjustment procedure is started. After some time, 3D-Equalizer has calculated a focal length of 25.66 mm for the camera “Size 135 Photo” and a horizontal angle of 27.2 degree for the camera “35mm Film” without any problems. When taking a look at the Status Window you will see, that the average deviation value of the sequence is around 0.34 pixel and the deviation of the reference frames is between 0.9 and 0.5 pixel. The project is very precise now. 3D-Equalizer was able to determine the correct camera settings, because we supplied enough “perspective information”.

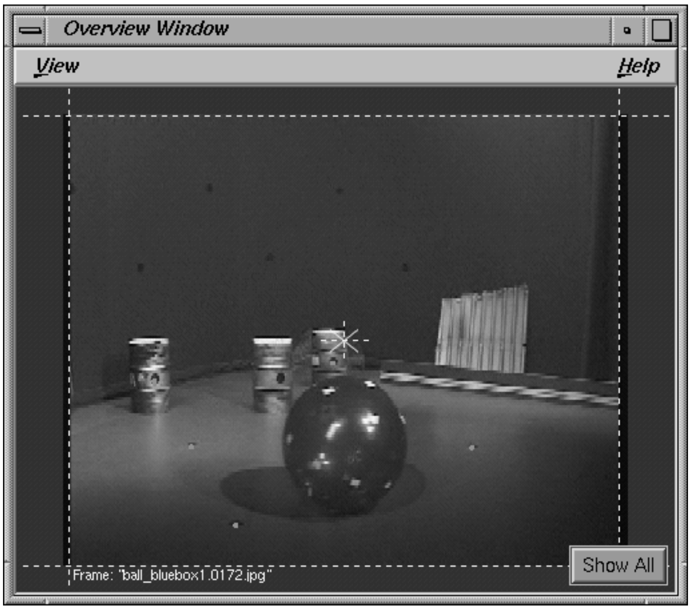
The project “vicky3.3de” found in the directory “<tutorial install directory>/projects/”, contains all steps we have taken so far. Please feel free to review it at any time.

Tutorial 8: Multiple pointgroups

One of 3D-Equalizer's most important features is the ability to track multiple, differently moving real-world objects and the camera simultaneously. In the following tutorial this feature is demonstrated in detail.

- *Please start 3D-Equalizer as described in the section "Getting started" and open the project "two_pgroups1.3de" in the directory "<tutorial install directory>/projects/".*

- *Open the environment "Motiontracking" from the Main Window's environments menu and take a look at the sequence (see figure below).*

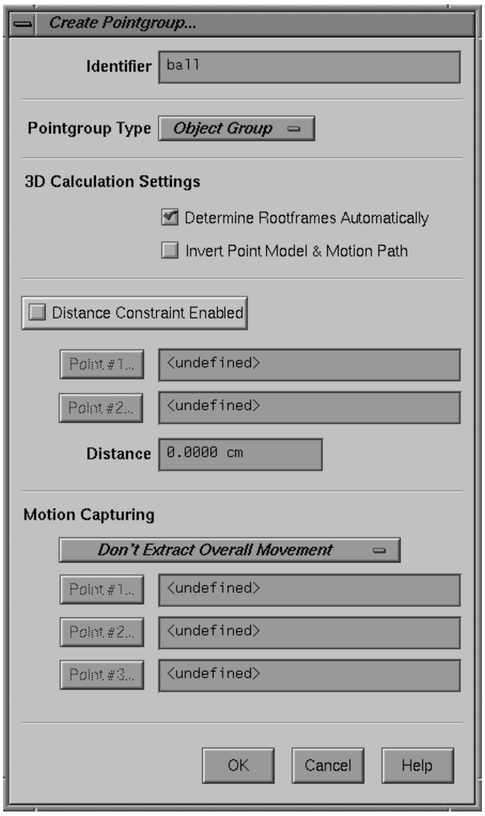


Overview Window

As you can see, the sequence is exactly the same as in the first tutorial. There is a camera pointgroup "studio" containing six points, which are already tracked. This pointgroup is used by 3D-Equalizer to determine the camera motion in 3D space. In addition there is a blue ball with white markers rolling over the studio floor from right to left. Let us begin by creating an additional pointgroup to determine the separate motion of the blue ball.

- *Select the menu function Pointgroups Window::Pointgroup::Add Pointgroup.*

The requester Create Pointgroup opens (see figure below).



Create Pointgroup...

Identifier: ball

Pointgroup Type: Object Group

3D Calculation Settings

☑ Determine Rootframes Automatically

□ Invert Point Model & Motion Path

□ Distance Constraint Enabled

Point #1... <undefined>

Point #2... <undefined>

Distance: 0.0000 cm

Motion Capturing

Don't Extract Overall Movement

Point #1... <undefined>

Point #2... <undefined>

Point #3... <undefined>

Requester: Create Pointgroup...

- *Change the name of the new pointgroup in the textfield Identifier to “ball”, make sure the Pointgroup Type is set to “Object Group” and press the button Ok to confirm your edits.*

A new object pointgroup “ball” is created and, as you can see, its entry is displayed in the respective list widget of the Pointgroups Window.

- *Please select pointgroup “studio” and afterwards select pointgroup “ball” again. Watch the behaviour of 3D-Equalizer’s windows on the screen.*

The pointlist widget in the Pointgroups Window is empty for “ball” but not for “studio”. This is because every pointgroup has its “own” set of points. In the Zoom- and Overview Window only the points of the currently selected pointgroup are displayed. Finally, the Timeline

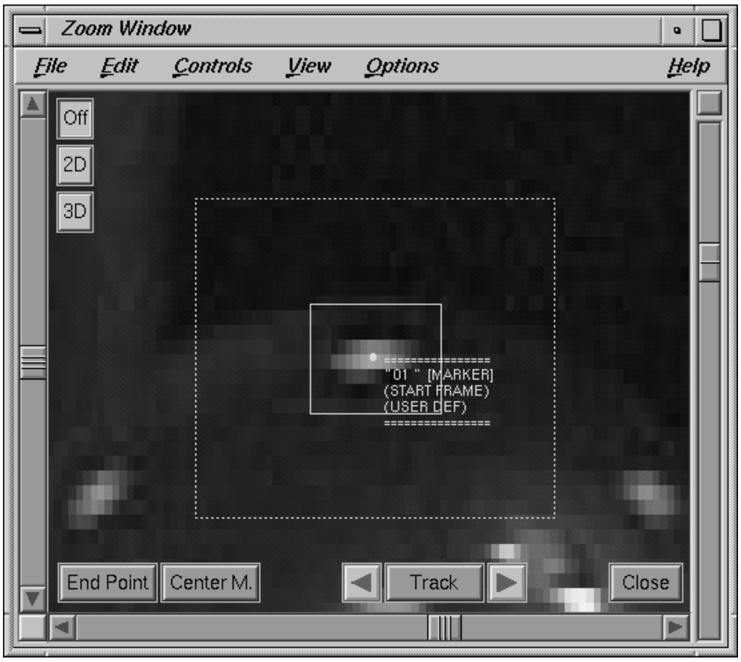
Window only displays the tracking curves of the points in the currently selected pointgroup tracked in the current sequence. Now, let’s create a new point object for pointgroup “ball”.

- *Make sure to select pointgroup “ball” in the Pointgroups Window.*

- *Press the button Pointgroups Window::Add to open the Create Point requester. Set the option menu Trackingmode to “Marker Mode - Forward”, and press the button Ok to confirm your edits.*

A new point is created for pointgroup “ball” and its entry appears in the point list widget. Now, let’s start to track one of the white markers on the blue ball.

- *Go to the first frame of the sequence. Make sure the newly created point object is selected and left-click on the uppermost white marker to define a new startpoint (keyframe) for the 2D trackingcurve. Modify the tracking area so that the marker fits well (see figure below).*



Zoom Window

- *Track the marker on the ball, as you already learned in the first tutorial.*

You may find that tracking these white markers on the ball is not so easy. There are a lot of reflections on the ball that sometimes confuse 3D-Equalizer’s tracker. Furthermore, the markers quickly become obscured by the ball itself, and then are visible again later in the sequence. If a formerly tracked marker is obscured and becomes visible again, it should be tracked with exactly the same point. To identify which marker has already been tracked with which point is difficult, too. Each pointgroup must have its “own” pair of rootframes, so it is necessary to have at least two frames in which six same points are properly tracked. That means there are at least six points needed.

Enabling the function Zoom Window::Options::Auto Center, using the “Track Single Frame” button and a bit of “hand” work are helpful in tracking the white markers.

- *Track enough markers on the ball, so that there are at least four points tracked in each frame. Make sure there are at least two frames available with the same six points tracked.*

If you are not interested in this challenging tracking exercise, simply open the project “two_pgroups2.3de” in the directory “<tutorial install directory>/projects/”.

The pointgroup “ball” is tracked quite well, in this project, using 12 points from the beginning until the end of the sequence.

- *Select the function Main Window::Calc::Calc All Objects From Scratch.*

The camera parameters being already well defined. 3D-Equalizer calculates the 3D motion path and 3D point positions of all pointgroup objects tracked in all sequences and reference frames.

- *Open the environment “Status”, make sure the pointgroup “studio” is selected in the Pointgroups Window and take a look at the average deviation value displayed in the Status Window.*

As you see the average deviation is about 0.24 pixel, which is quite a good value. Since pointgroup “studio” is selected, it represents the reconstruction quality of the 3D camera motion.

- *Select the pointgroup “ball” and take a look at the Status Window again.*

The green deviation curve looks completely different and the average deviation is now about 0.25 pixel, which is a good value too. Since pointgroup “ball” is selected now, it represents the reconstruction quality of the blue ball’s motion in 3D space. All in all, it seems that our project is ok. Let’s take a closer look at the resulting 3D data.

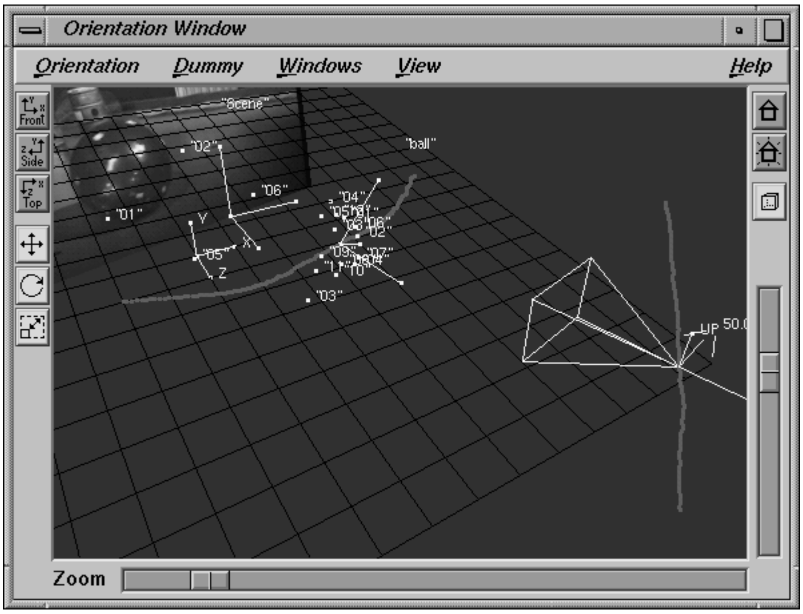
- *Open the environment “Orientation” from the Main Window’s environments menu.*

- *Activate the toggle buttons Orientation Window::View::Show Camera Path and View::Show Pointgroup Path.*

There are two separate, red motion paths displayed in the Orientation Window’s display area (see figure below).

- *Drag the frame slider of the Main Window back and forth, while watching the Orientation Window's display.*

As you see, the camera object and the pointgroup object "ball" (including its point objects) are moving separately from each other.



Two motion paths in 3D space

- *Select the pointgroup object "ball" in the 3D Object Browser Window. Activate the scale toggle button located on the left border of the Orientation Window and click-drag the left mouse button in the display area.*

As you see, you are able to scale the pointgroup object "ball" (including its points) separately from the camera and the pointgroup "studio"'s points. In addition, each time you release the left mouse button the motion path of pointgroup "ball" is modified. What is going on?

- *Activate the toggle button Orientation Window::View::Camera View.*

The Orientation Window's display is changed, so that you are now looking exactly "through" the reconstructed camera. The reconstructed points of both pointsgroups are clearly located on the respective markers of the current image, displayed by the imageplane.

- *Make sure the scale toggle button is activated click-drag the left mouse button in the display area again.*

You are still able to scale pointgroup “ball”, however when looking through the reconstructed camera its points don’t seem to be modified at all! In other words, regardless of the pointgroup’s scale, the matchmove is always correct. This is a strange mathematical “effect”:

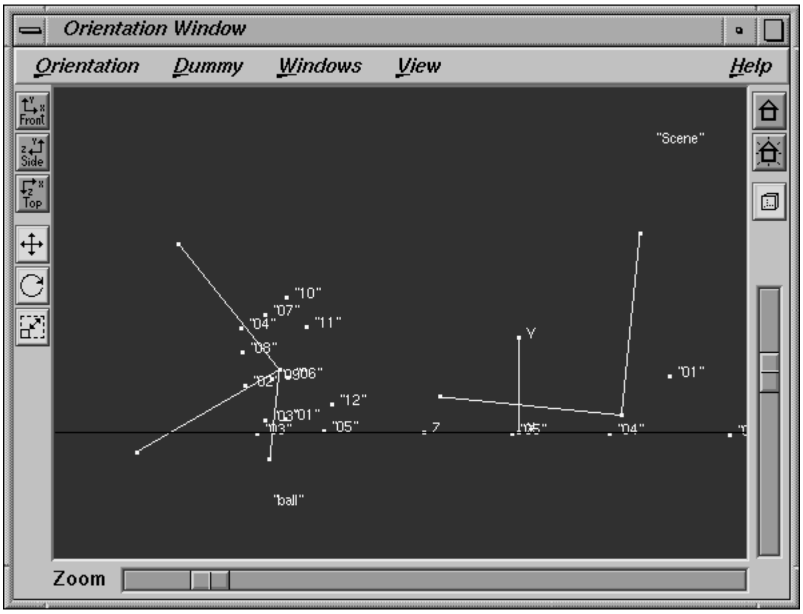
> **Important note:** The scale of different pointgroups to each other cannot be reconstructed automatically. It should be defined “manually” using a separate distant constraint for each pointgroup.

Let’s continue to define a distant constraint for each pointgroup.

- *Deactivate the toggle button Orientation Window::View::Camera View.*

- *Double-click on the “studio” entry of the Pointgroups Window, in order to open the requester Modify Pointgroup. Enable the distant constraint feature between point “04” and point “06” using a distance of 114.6 cm. In order to confirm your edits click on the button Ok.*

- *Double-click on the “ball” entry of the Pointgroups Window, in order to open the requester Modify Pointgroup. Enable the distant constraint feature between point “08” and point “09” using a distance of 25.0 cm. In order to confirm your edits click on the button Ok.*

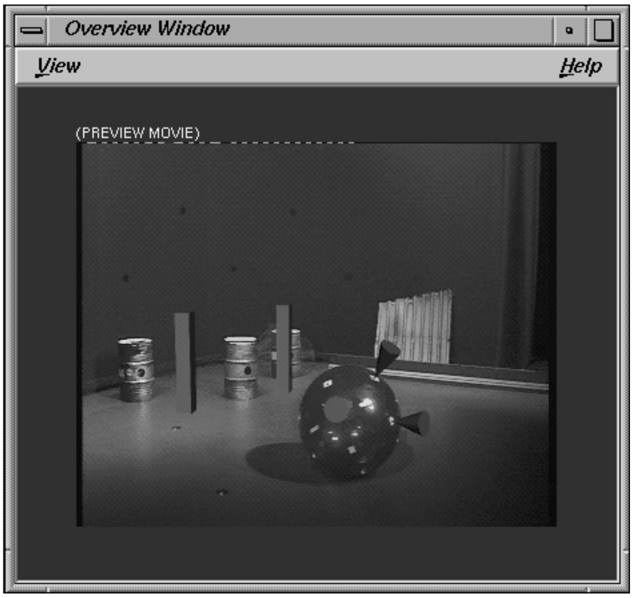


Properly scaled pointgroups due to distant constraints

The Orientation Window's display area is automatically updated. You will notice that you are no longer able to scale the pointgroup "ball" or the scene node. This is because the scale of these objects has already been defined by the distant constraints. When you look at the reconstructed 3D data, you'll find that the pointgroup "ball" is in a reasonable relationship to pointgroup "studio". This is indicated by the fact that the points of "ball" seem to roll exactly over the plane defined by the points of "studio" (see figure above). Finally, let's create a preview movie to examine the result.

- *Create a few dummy objects for pointgroup "studio" and pointgroup "ball", and distribute them in the scene. Open the "Preview" environment and press the button Play.*

3D-Equalizer renders a preview movie, which takes some time. After the last frame of the movie has been rendered, it is played back in realtime. As you see, the motion of the camera as well the motion of the blue ball are properly reconstructed.



Preview movie

The project "two_pgroups3.3de" found in the directory "<tutorial install directory>/projects/", contains all steps we have taken up to now. Please feel free to review it at any time.

Tutorial 9: Multiple sequences

In the following tutorial we will discuss how 3D-Equalizer can help you in handling multiple shots filmed of a specific environment. This feature is helpful in situations where cg models must be composited with different sequences, but should appear at exactly the same position in 3D space. For instance multiple shots of a “set extension” scene for a feature film.

- *Please start 3D-Equalizer as described in the section “Getting started” and select the function Main Window::Project::New to create a blank project.*

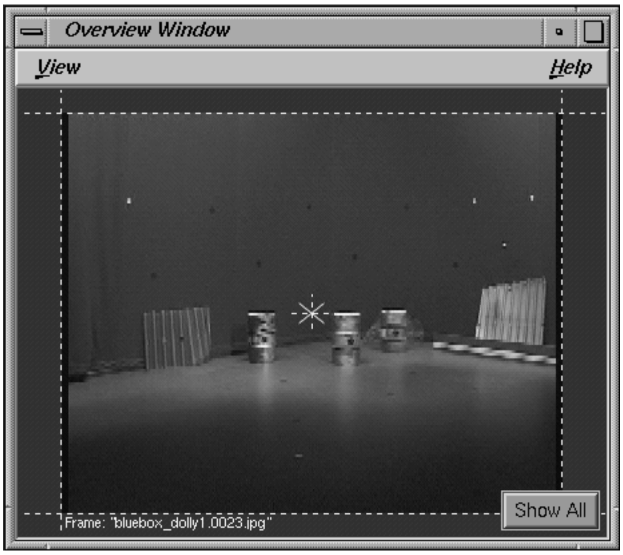
- *Open the environment “Motiontracking” from the Main Window’s environments menu.*

- *Press the button Frames Window::Add and then press the button Add Sequence to create a new image sequence object.*

The requester Create Sequence opens.

- *Click on the button Sequence Settings::Browse and import the image sequence that is located in the directory “<tutorial install directory>/bluebox_dolly_1/”.*

- *Set the option menu Imagetype to “Interlace - Odd First” and confirm your edits by pressing the button Ok.*



Frame: "bluebox_dolly1.0023.jpg"

Sequence “bluebox_dolly1”

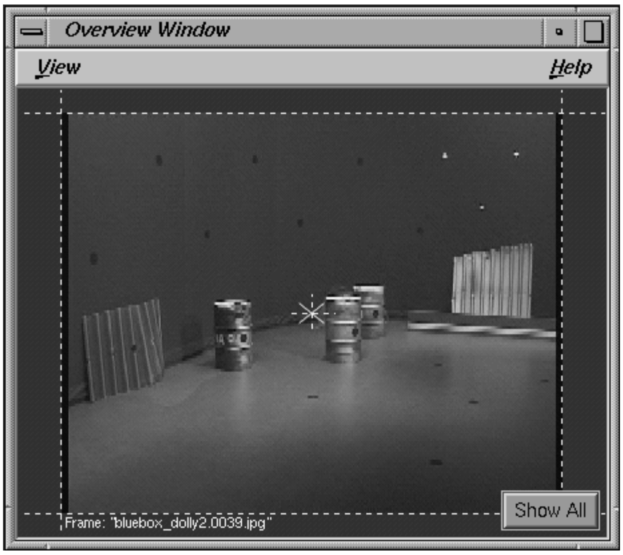
A new sequence object “bluebox_dolly1” is created, its entry is displayed in the Frames Window and the first image is displayed in the Zoom- and in the Overview Window (see figure above).

- *Make yourself familiar with the sequence by clicking a few times on the right side of the Main Window’s frame slider area.*

As you see, the bluebox studio which you already know from some of the previous tutorials is shown. During the course of the sequence the camera moves straight onto the three barrels that stand on the studio floor.

- *Click on the button Frames Window::Add and then on the button Add Sequence to create a second image sequence. The requester Create Sequence opens. The images for this sequence are located in the directory “<tutorial install directory>/bluebox_dolly_2/”. Import them using the function Sequence Settings::Browse and set the option menu Imagetype to “Interlace - Odd First” like you did with the first sequence. Press Ok to close the requester.*

An additional sequence object “bluebox_dolly2” is created, its entry is displayed in the Frames Window and the first image is displayed in the Zoom- and in the Overview Window (see figure below).



Frame: "bluebox_dolly2.0039.jpg"

Sequence “bluebox_dolly2”

- *Make yourself familiar with this new sequence by clicking a few times on the right side of the Main Window’s frame slider area.*

As you see, the same bluebox studio with the same set of markers is shown. However, the camera moves around the three barrels in a totally different way than in the first sequence. Now, let’s track some points.

- *Select the function Pointgroups Window::Pointgroup::Add Pointgroup.*

The requester Create Pointgroup opens.

- *Change the textfield Indentifier to “studio” and leave all other settings as they are. Press Ok to close the requester.*

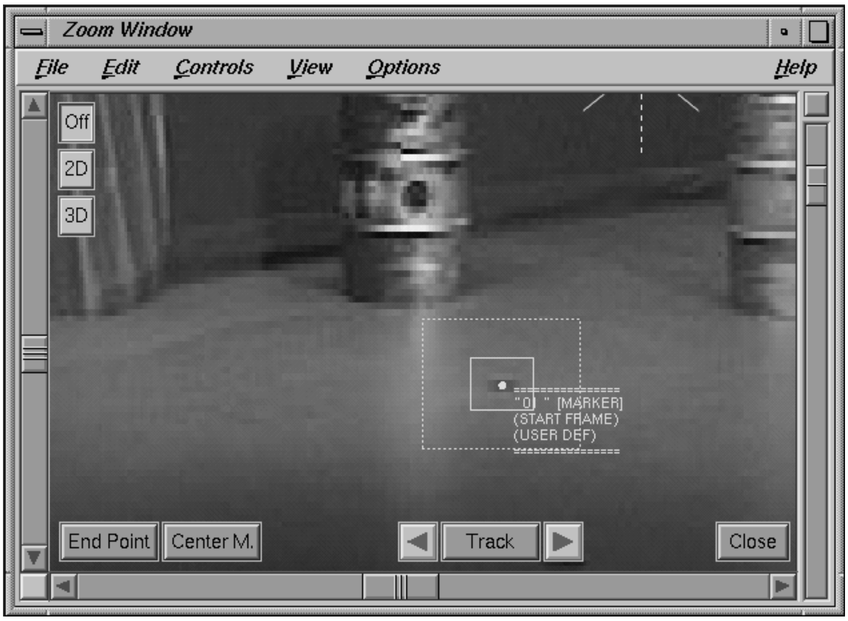
A new camera pointgroup “studio” is created. Its entry is displayed in the Pointgroups Window.

- *Press the button Pointgroups Window::Add.*

The requester Create Point opens.

- *Set the option menu Trackingmode to “Marker” and confirm your edits by pressing the button Ok.*

A new point object “01” is created.



Tracking a marker in sequence “bluebox_dolly1”

- *Select sequence "bluebox_dolly1" in the Frames Window and go to its first frame.*

- *Make sure point "01" is selected and start to track the marker that is located right below the left most barrel (see figure above).*

You should be able to track this marker until it leaves the FOV at frame 78.

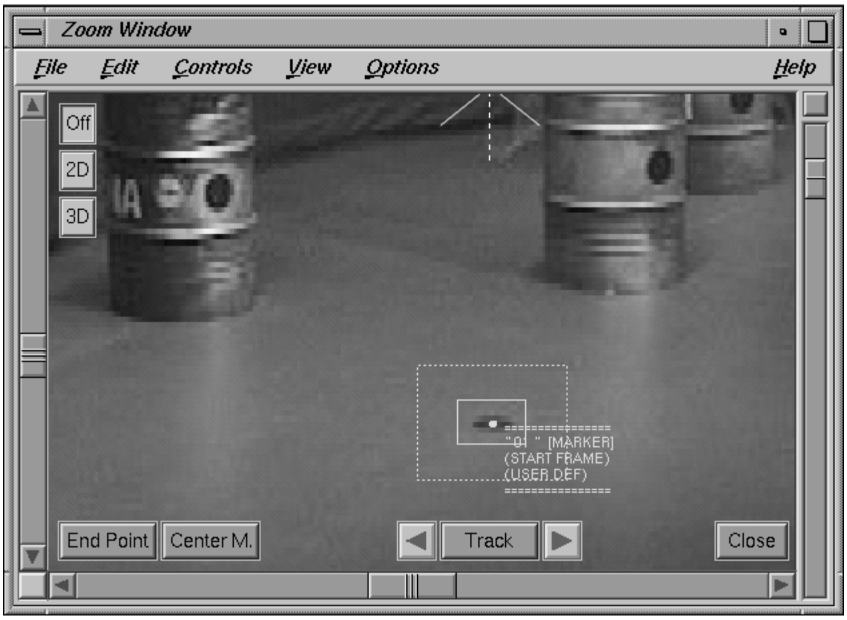
- *Activate the toggle button Zoom Window::View::Show Motion Curve.*

A curve is displayed in the Zoom Window. It represents the two-dimensional tracking curve of point "01" recently tracked in the selected sequence ("bluebox_dolly1").

- *Select sequence "bluebox_dolly2" in the Frames Window and watch the Zoom Window's display area.*

As you see, the tracking curve disappears from the Zoom Window display, because point "01" is not yet tracked in sequence "bluebox_dolly2" (see also Timeline Window).

- *Go to frame 1, make sure point "01" is selected and track exactly the same marker as in the previous sequence (see figure below).*



Tracking a marker in sequence "bluebox_dolly2"

The resulting two-dimensional tracking curve of point “01”, tracked in sequence “bluebox_dolly2”, is of course totally different from the tracking curve tracked in sequence “bluebox_dolly1”. It is important to understand that each point object can “have” a set of multiple tracking curves. As mentioned before, the clue with multiple sequences is to get multiple animated cameras but only one single set of points in 3D space.

- *Please continue to track the sequences, using additional points. Make sure that there are at least four points tracked in both sequences, otherwise 3D-Equalizer is not able to reconstruct a proper set of 3D points.*

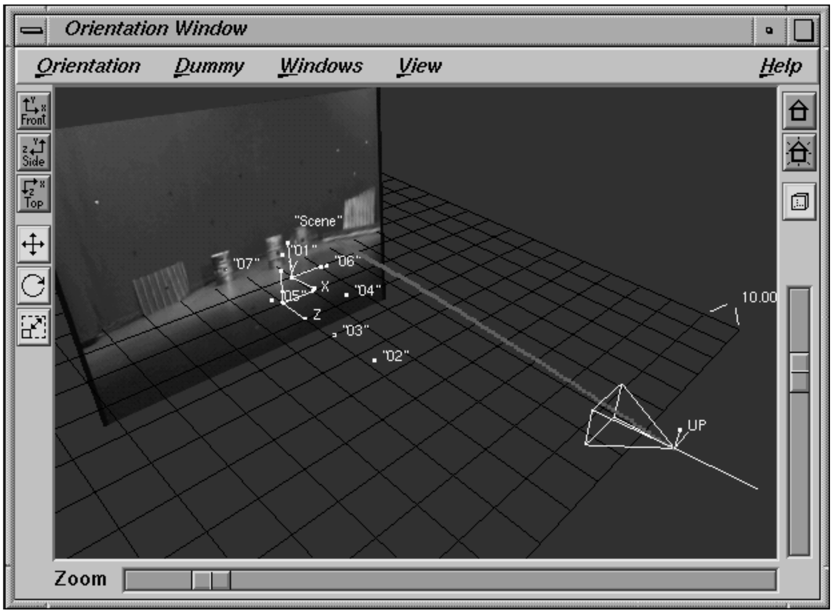
Since there is only one pointgroup, only one pair of rootframes is required. Therefore it is possible to track one sequence using six points and the other sequence using just four points.

- *If you are not interested in tracking the two sequences, please open the project “two_sequences1.3de” found in the directory “<tutorial install directory>/projects/”.*

As you see, there are nine points in total with this project. Six points have been tracked in both sequences, the remaining three points have been tracked in just one.

- *The camera parameters are already very well defined, so let 3D-Equalizer reconstruct the data by selecting the function Main Window::Calc::Calc All Object From Scratch.*

3D-Equalizer finishes the task after some time. Now, let’s take a look at the result.



Reconstructed camera motion path of sequence “bluebox_dolly1”

- *Open the "Status" environment from the Main Window's environments menu. Make sure sequence "bluebox_dolly1" is selected and examine the average deviation value displayed in the Status Window.*

The average deviation is about 0.2 pixel, which is a good value. The green deviation curve represents the reconstruction quality of the currently selected sequence "bluebox_dolly1".

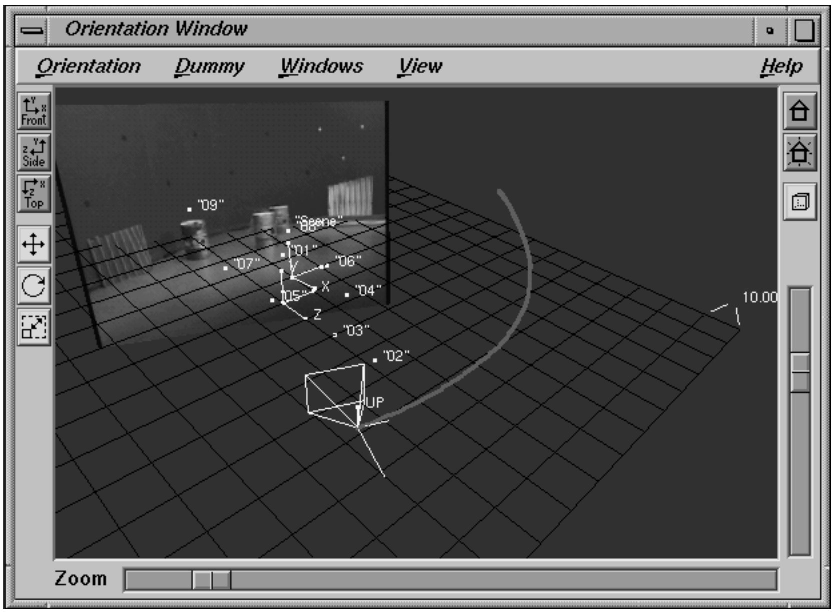
- *Select sequence "bluebox_dolly2" in the Frames Window and take a look at the Status Window again.*

The green deviation curve looks completely different, of course. The average deviation is about 0.23 pixel, again a good value. It seems that our project went well, so let's take a look at the actual reconstructed 3D data.

- *Open the environment "Orientation" and select sequence "bluebox_dolly1" in the Frames Window. Enable the toggle button Orientation Window::View::Show Camera Path and take a look at the display area.*

You will see a set of 3D points and a camera object with its camera motion path (see figure above).

- *Select sequence "bluebox_dolly2" in the Frames Window and take a look at the Orientations Window's display again (see figure below).*



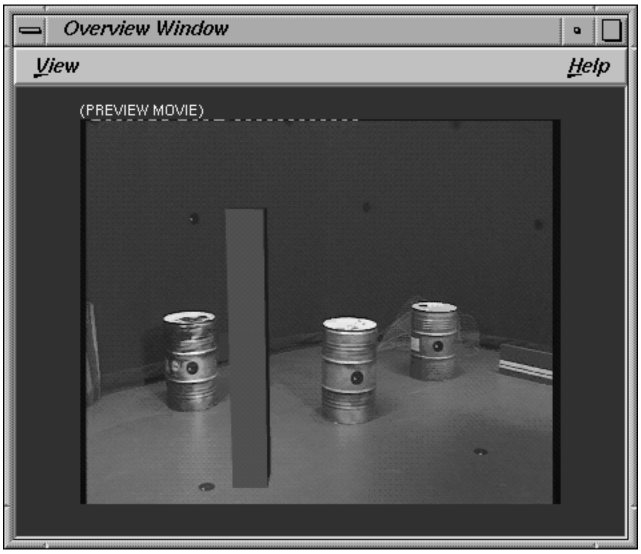
Reconstructed camera motion path of sequence "bluebox_dolly2"

You will notice that the current camera position and the entire camera motion path are modified. However, the set of points in 3D space isn't changed at all. That means there is now one fixed 3D point cloud with two separate 3D camera motions that are perfectly matched to it.

- *Create a few dummy objects and transform them so that they become virtually located on the studio floor. If you are not interested in this, please open the project "two_sequences2.3de" in the directory "<tutorial install directory>/projects/".*

- *Open the "Preview" environment, make sure that sequence "bluebox_dolly1" is selected in the Frames Window and press the button Main Window::Play.*

3D-Equalizer starts generating the preview movie. After this task has been completed 3D-Equalizer plays back the preview in realtime (see figure below).



Preview Movie

- *Press the button Main Window::Stop, select sequence "bluebox_dolly2" and press the button Main Window::Play again.*

3D-Equalizer now generates a preview movie for sequence "bluebox_dolly2" and plays it back after it has been completed. The preview of sequence "bluebox_dolly1" is not lost. The preview and cache movies for all sequences of a 3D-Equalizer project are stored in a movie archive.

- *Press the button Main Window::Stop, select sequence “bluebox_dolly1” and press the button Main Window::Play again.*

As you see, the preview movie of sequence “bluebox_dolly1” still exists because it has been stored in a movie archive.

Tutorial 10: Lens Distortion

The following tutorials describe how distorted images should be handled. First, the lens distortion parameters of 3DE V3 are introduced. Afterwards, we investigate how these parameters can be determined by means of gridshots and also in cases where no gridshots are available. The last section is dedicated to the tool WarpDistort, which allows you to apply or remove distortion to / from real or rendered images.

General remarks

Before an image is processed (by means of the tracking points) by the calculation kernel of 3DE, non-linear distortion effects should be removed so that the simple camera pyramid model is applicable. This is done within 3DE by means of several parameters you specify in the Camera Adjustment Window. In 3DE version 3 the non-linear distortion is modelled by four parameters. These are called lens distortion (as in version 2), anamorphic squeeze, curvature x and curvature y. Additionally there are two parameters called lens center offset x and y. They specify the center of distortion, which is equivalent to the central axis of the camera's lens system. The lens center offset is measured in real length units, since it refers to a real, physical point offset on the filmback.

In the following we will illustrate and describe the effect of modifying these parameters. The default values for the distortion parameters are:

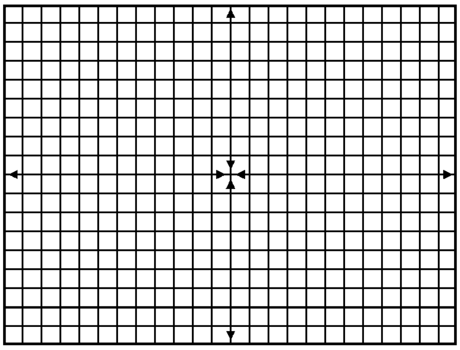
- lens distortion = 0.0
- anamorphic squeeze = 1.0
- curvature x = 0.0
- curvature y = 0.0

Additionally, there is a parameter called quartic distortion, which can be helpful when fisheye lenses are used. The default value for this parameter is:

- quartic distortion = 0.0

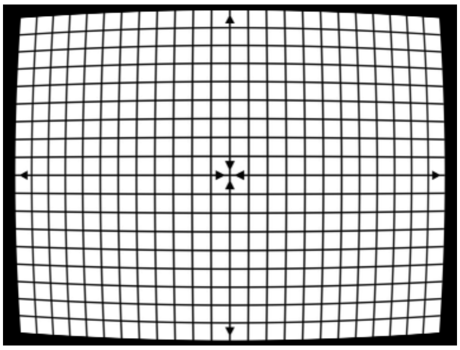
The default value for the lens center offset x/y is 0, which corresponds to the center of the filmback.

When the lens distortion is set to some positive value, 3DE assumes that the images are subject to some radially symmetric barrel type distortion. In the following, we give some examples of distorted images, using different values for the various parameters. The grid we use here has a film aspect of 4:3.

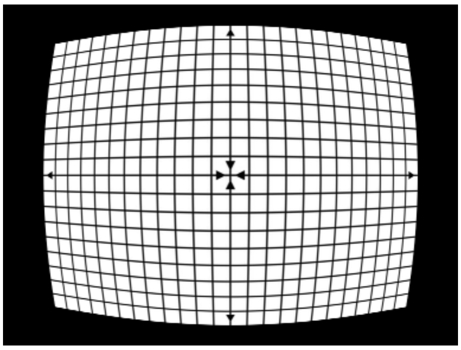


Undistorted grid

In the images below we have distorted a rendered grid shot using a lens distortion value of 0.1 and 0.5, respectively.

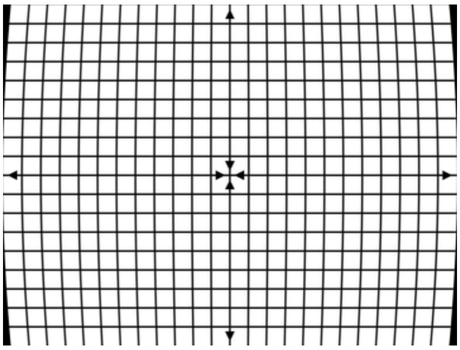


Lens distortion = 0.1

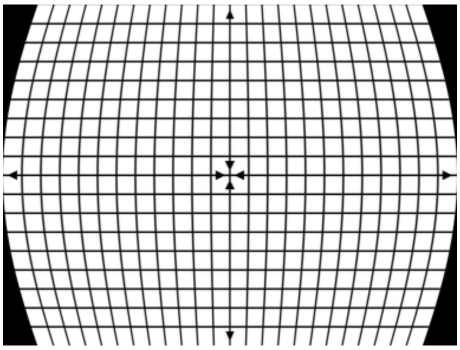


Lens distortion = 0.5

In many cases, it is sufficient to use only one parameter for removing non-linear distortion, since most lens systems in fact exhibit radial symmetry around the view axis. However, when an anamorphic camera is used, this simple model fails. One might think that in this case a two-parameter model consisting of lens distortion and anamorphic squeeze would be sufficient. However, it turned out that in practice the two remaining parameters curvature x and curvature y sometimes need to be adjusted properly. In general we think that this is due to the fact that we cannot say in general <u>where</u> exactly within the lens system the distortion occurs: before or after the anamorphic squeezing (or both? or during?). In other words, the curvature parameters are necessary in order to cope with every imaginable kind of anamorphic lens. On the other hand, it is hard to imagine that a distortion could be handled with the curvature values <u>only</u>, since there is probably no physical lens system which produces this pure type of curvature.



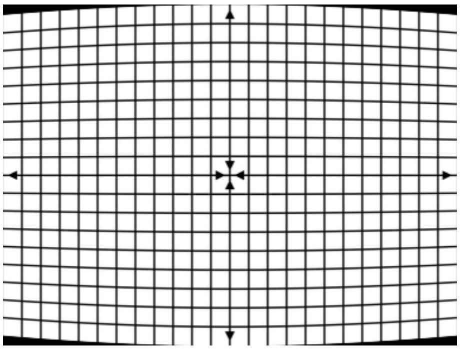
Curvature x = 0.1



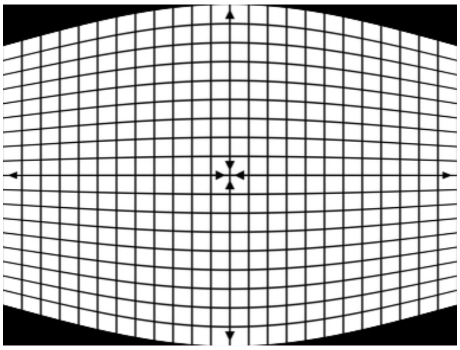
Curvature x = 0.5

Let us have a look at the distortion effects of the anamorphic squeeze and the curvature parameters. In the accompanying figure the distortion values are set to the default, except for the curvature x (0.1 and 0.5). As you see, there is only a horizontal distortion effect, no vertical effect. Note also, that the x-axis is not distorted at all.

The parameter curvature y has a similar interpretation. Below you see two grids in which all parameters are set to the default, except for curvature y. The distortion only affects the vertical direction, and the y-axis itself remains undistorted.

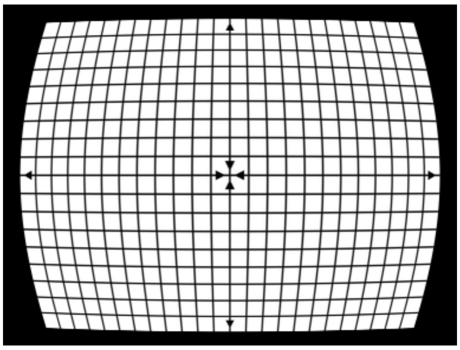


Curvature y = 0.1

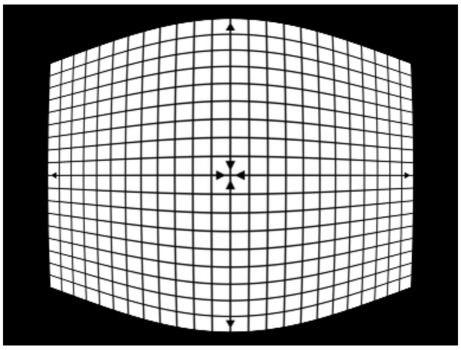


Curvature y = 0.5

Next, let us examine the effect of the parameter anamorphic squeeze. Its interpretation is not so straightforward. First of all: this parameter does not express the squeeze ratio of an anmorphic image. It only expresses the “anisotropy” of the other distortion parameters. That means, when lens distortion, curvature x and curvature y are set to their defaults, anamorphic squeeze does not have any effect. We illustrate the effect of this parameter by means of the two grids below. The lens distortion is 0.3, the anamorphic squeeze values are 0.5 and 2.0, respectively.



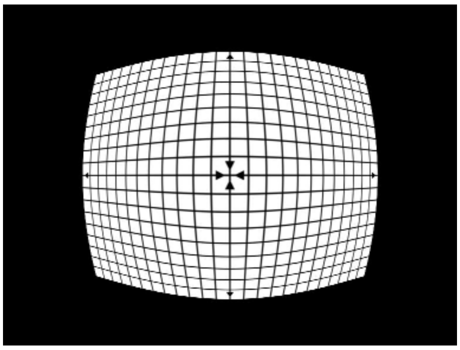
Lens distortion = 0.3
Anamorphic Squeeze = 0.5



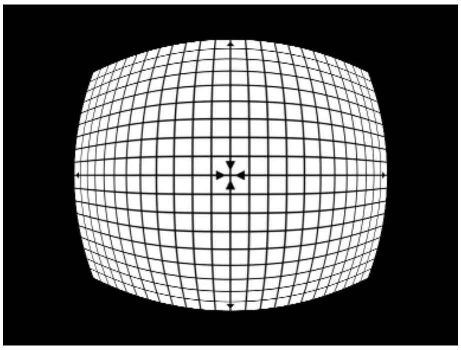
Lens distortion = 0.3
Anamorphic Squeeze = 2.0

As you see, anamorphic squeeze does not really squeeze the image, but only the distortion effect of the other parameters.

Finally, we point out the effect of the parameter quartic distortion. In the right image below we used a value of 5.0 for the quartic distortion and 0.0 for the lens distortion, while in the left image the quartic distortion is 0.0 and the lens distortion is 2.0. The strength of both distortions is comparable, however you may note that quartic distortion has only a minor effect near the lens center whereas the effect at the border of the image is stronger than in the case of lens distortion.



Lens distortion = 2.0
Quartic Distortion = 0.0



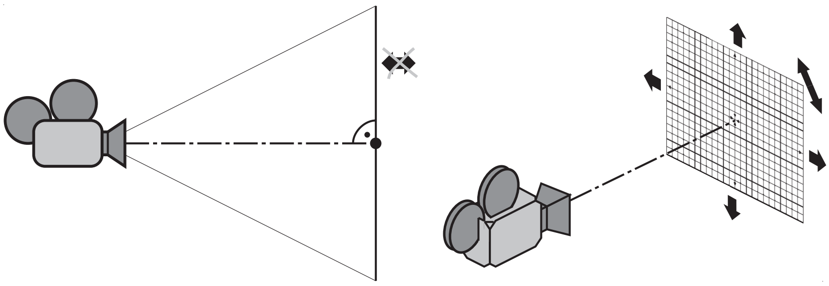
Lens distortion = 0.0
Quartic Distortion = 5.0

In the following tutorial we will not concider the quartic distortion. Therefore, this parameter should remain zero. In most practical situations it will not be necessary to use this parameter at all.

Recording a grid

When you intend to process a sequence that has been recorded with a camera which exhibits a significant lens distortion, you may wish to take a grid shot before using exactly the same camera with the same lenses. Especially in the case of an anamorphic camera which can have a large lens distortion this procedure is recommended. In the Zoom Window of 3DE, you can edit a virtual grid, translate, rotate and scale it, tweak the various distortion parameters and displace the lens center in order to match it to the grid shot. It is usually necessary to translate, rotate and scale the virtual grid, because the geometric relationship between the real grid and the camera is not well-defined, according to our experience, i.e. in practice the grid is slightly rotated, and its center does not coincide with the camera's direction of view. Scaling, rotating and translating the virtual grid does not affect the 3D calculations of 3DE, it is only meant to help you to match the gridshot. The only 'really' important quantities are the four lens distortion parameters (lens distortion, anamorphic squeeze, curvature x/y) and the two lens center parameters.

There is, however, a condition which should be fulfilled in order to be able to match the virtual grid as well as possible to the recorded grid. We have sketched this condition in the figure below: The direction of view (which is equal to the lens center projected onto the filmback) should be perpendicular to the grid plane. This condition enables you to match the grids without knowing the focal length. Things become easier, of course, if you place the direction of view directly on the center of the grid.



How to take a gridshot

Apart from the perpendicularity condition mentioned above, you also need to know the film aspect ratio in the following sense: you certainly have examined 3DE's Camera Adjustment Window. There are text fields called 'Filmback Width', 'Filmback Height' and 'Film Aspect'. When a conventional camera is used, it is clear what these expressions mean. In order to match the grids you should enter the correct value of the film aspect ratio or the filmback width and height (only their ratio is important here). This automatically fixes the text field 'Pixel Aspect'. When an anamorphic camera is used, the question arises, which value has to be entered: the measurable, squeezed film aspect of the camera film back or the un-squeezed filmback (which you will not find as a physical quantity anywhere within the camera). For example, a typical image ratio nowadays for wide-screen is 2.4:1. The filmback in this case may be 0.825 in x 0.690 in [5], which corresponds to an anamorphic squeeze factor of 2.0. When this film strip is scanned by a film scanner with a pixel aspect of 1.0, the resulting image files have a pixel aspect of 2.0. In 3DE's Camera Adjustment Window the value for the filmback height is simply 0.690 in (= 1.7526 cm). However, the filmback width must be entered as if the image were not squeezed, namely 0.825 in x sequeeze factor = 1.650 in (= 4.191 cm). If you enter these values, the pixel aspect is 2.0 and the film aspect is 2.4.

Examples of grid shots

In the projects directory you will find a project named "grid_shots.3de". This project cannot be calculated, and it does not make sense to define or track any points; it is only a collection of grid shots, each associated to its own camera. By means of these grid shots you can play around with the distortion parameters.

There are several circumstances under which the virtual grid will not match the recorded grid exactly:

0. The pixel aspect of the camera of the frame containing the grid is wrong.

1. The recorded grid was not really drawn on a plane surface

2. The grid was inclined when it was recorded, i.e. the grid plane was not perpendicular to the direction of view.

3. The lens distortion is so high, that the four-parameter model is not sufficient (we believe that in practice this does not happen very often).

It may happen that the images are deformed in some really odd, position dependent way for artistic reasons. This situation cannot be handled at all with the present lens distortion model. (It is planned in the future to provide a mechanism within 3DE to cope with this type of distortion as well (two-dimensional displacement map).)

Measuring lens distortion by means of grid shots

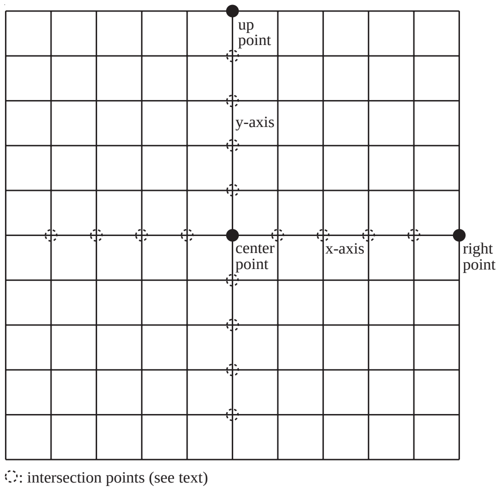
In the following we will see how the distortion measurement by means of a grid shot can be done for a grid which has been recorded using an anamorphic camera. In order to do this we will need the Frames Window, the Zoom Window and the Camera Adjustment Window. You can also select the environment “Lens Distortion Editing” from the menu Main Window::Environments. The Lens Distortion Editing feature is only available in the film version of 3DE.

- *Please select the reference frame “grid 9”, activate the mode Zoom Window:: Controls::Distortion Grid Edit Controls. Reset the grid shape by pressing Zoom Window::Edit::Reset Distortion Grid. Also, reset the camera parameters in the Camera Adjustment Window to their default values, i.e. lens center offset x->0.0, lens center offset y->0.0, lens distortion->0.0, anamorphic squeeze->1.0, lens curvature x->0.0, lens curvature y->0.0.*

The process of matching the distortion grid to the recorded grid can essentially be divided into two passes:

1. Translating, rotating and scaling the grid so that the “most important” points coincide with the corresponding points in the original grid. As mentioned, the operations which allow you to modify the shape of the grid do not influence the distortion behaviour of the camera model, but they are necessary because the translation, rotation and scaling of the recorded grid varies (as you see in the various frames of the project “grid_shots.3de”).

2. Tweaking the lens center offset and lens distortion parameters in order to make the entire grid coincide with the recorded grid.



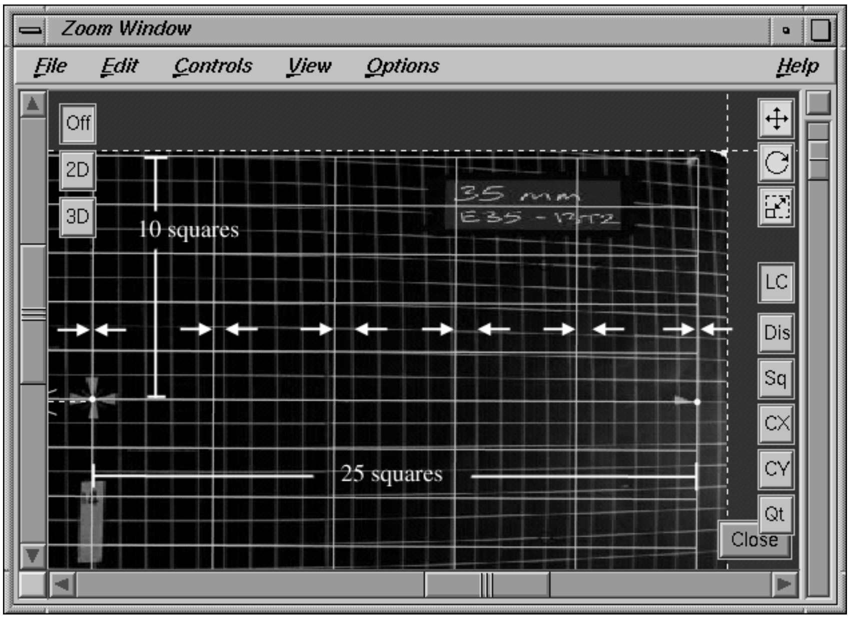
Distortion grid

In practice, it will turn out that these passes cannot be separated so strictly. There are two groups of Distortion Grid Edit Controls (which clearly reflect the two passes we mentioned above): the upper three buttons are used for translating, rotating and non-uniform scaling, while the lower ones correspond to lens center offset and the distortion parameters. Now, please have a look at the distortion grid above. You see three points: one in the center of the grid, one on the right hand side and one at the top. We will call these points "center-point", "right-point" and "up-point" in the following. All scaling and rotating is performed with respect to the center-point. The steps (i) - (vii) below correspond to the frames "grid 9 step (i)" - "grid 9 step (vii)" in the current project.

- *Step (i) - Use the translate button to place the center-point on an intersection point of your choice in the recorded grid, preferably on its designated center.*

- *Step (ii) - Use the scale buttons to magnify the grid, so that it approximately covers the recorded grid.*

- *Step (iii) - Use the rotate button in order to match the x-axis and y-axis of the grid to the axis of the recorded grid. Make sure that up-point and right-point lie on the x- and y-axis of the recorded grid.*



Grid after step (iv)

The next step is a little bit odd: you should try to scale the grid such that there is a reasonable correspondence between the distortion grid square borders and the recorded grid square borders. One possibility would be to match ONE distortion grid square to ONE recorded grid square, but in this case you would lose precision. The best way to try it out is to do it on the example grid here:

- *Step (iv) - Activate the scale button. Move the up-point up and down, until ONE distortion grid square height equals TWO recorded grid square heights. Move the right-point until it coincides with the 25th intersection point. By doing this, you define that one distortion grid square corresponds to 5x2 recorded grid squares.*

We have now defined the positions of the three most important points of the distortion grid. In the following, we will tweak the distortion parameters. 3DE will always try to keep the positions of center-point, right-point and up-point fixed. We start tweaking the most important parameter, i.e. the lens distortion.

- *Step (v) - Activate the distortion button “Dis” and adjust this parameter so that the intersection points of the x-axis and the y-axis coincide as good as possible with the corresponding points in the recorded grid. Afterwards the parameter lens distortion in the Camera Adjustment Window will be around 0.11.*

As you see, the intersection points on the right hand side fit quite well, whereas the intersection points on the left hand side do not. There can be two reasons for this, and we cannot decide from looking only at the grid shot: either the lens center must be offset (very probable), or the grid was inclined during the shot (maybe), or both. We will try to correct the lens center offset in order to make the left hand side fit as well.

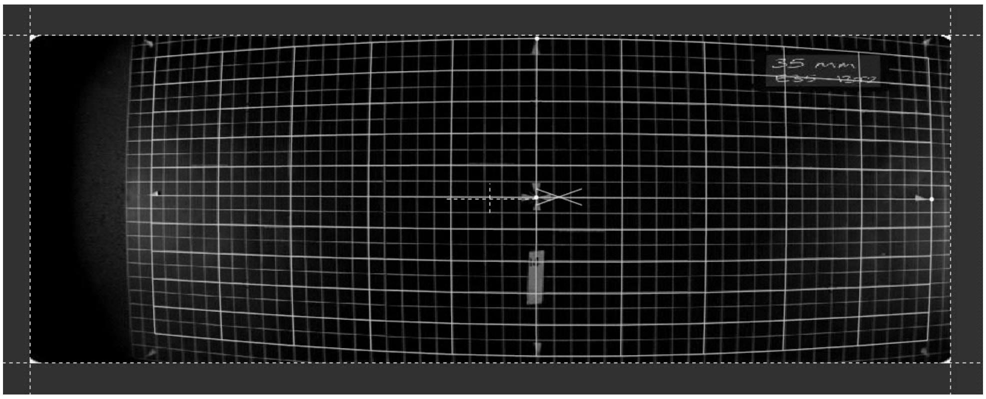
- *Step (vi) - Activate the lens center button “LC” and place the white cross so that the left and the right hand side “look” symmetric. Tweak the distortion parameter and the lens center offset alternately until the intersection points on the x- and y-axis coincide as well as possible.*

We now have a symmetrically positioned grid with coinciding intersection points on the x- and y-axis. There are three parameters left, which we have not touched: anamorphic squeeze, curvature x and curvature y. Sometimes it is hard to decide which parameter should be tweaked, since they can generate similar deformations. On the other hand, the most important thing is to match the grids. It is not so important how you do it. There may be various combinations of anamorphic squeeze and the curvature parameters which produce similar results. In our example we decided to tweak the parameter curvature y, since its effect is easy to understand and produces an acceptable result.

- *Step (vii) - Activate the curvature y button “CY” and tweak it until the horizontal grid lines coincide. The vertical lines already seem to fit, but you can also tweak the parameter “CX” if you like. If necessary, rotate the grid a little.*

Now, the result looks quite good, but it is not perfect. This is due to the fact that we have chosen quite a difficult example with an anamorphic lens, in which we do not know the lens center, how far the grid was declined during the shot, and (to be honest) we are not quite sure about the film/pixel aspect. As mentioned above, the two passes of aligning the grid and modifying the lens parameters cannot be separated completely. In a final pass you can tweak "everything" a little bit and see if you can improve the result. Finally, you will find that the deviation of the edited grid from the recorded grid ranges from 0 and 2 pixel. It is important to know that this does NOT imply that the match move calculations of 3DE using this camera settings will produce an error of two pixel as well! The deviation of the three-dimensional calculations is dominated by the errors of the two-dimensional motion tracking, and the deviation of the distortion grid plays a minor (however not negligible) role.

To sum up, the things which have to be done in any case are translating, scaling and rotating the grid and tweaking the parameter lens distortion. The other parameters may often remain on their default values, or they are known and can be entered. For video cameras the lens center is usually the center of the image and anamorphic squeeze and the curvatures remain at their default values. For movie cameras the lens center can be obtained from the data sheet. Anamorphic squeeze and the curvature parameters only apply for anamorphic lenses and even in this case they should only be used if it really turns out to be necessary.



Final result, mismatch between 0 and 2 pixel

Measuring lens distortion without grid shots

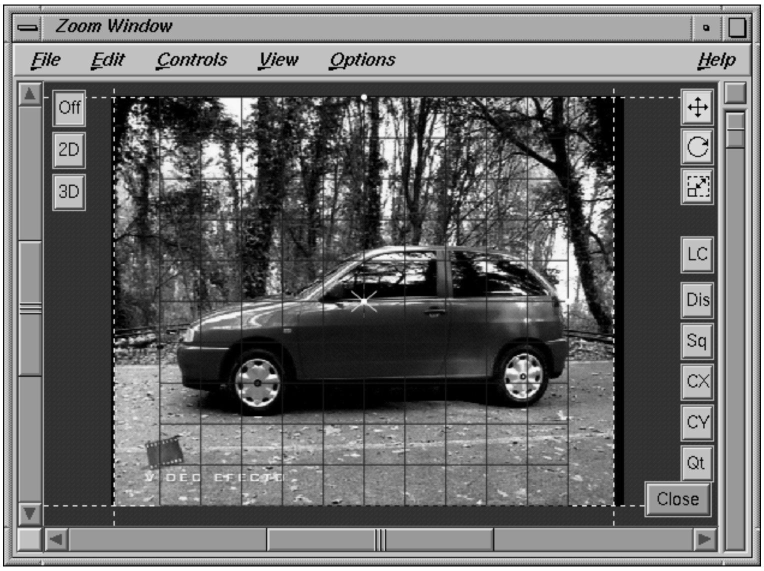
It may happen that you need to process a sequence recorded with a camera which obviously has some kind of distortion, but no grid shots are available. In this case it might be difficult to find out all of the parameters, however it is often possible to determine at least the most important parameter, namely lens distortion. In this case it is recommended that you set all other parameters to their default values.

There are two ways to determine lens distortion from the image material without recourse to grid shots:

1. Use the functionality of the Camera Adjustment Window: set the menu of the field lens distortion to “Fine Adjust” and do a camera parameter optimization as described in the other tutorials. Since usually the aperture angle / focal length is also unknown, this will turn out to be a time-consuming procedure. This method of course requires that the tracking is complete and precise.
2. The other method is much more elegant and (depending of the available depth information) can be even more precise. However, it requires that the image material contains objects with straight lines, e.g. buildings, the horizon, lane markings etc. For this method we need the Distortion Grid Edit Controls of the Zoom Window, only accessible in the film version of 3DE.

While method 1. should be clear from some of the other tutorials, we should have a look at method 2. The basic idea is to use the editable grid in the Zooming Window and match one of its lines to some straight line that happens to occur in the sequence. In order to do this, the grid usually has to be translated, rotated and sometimes also scaled.

- *Please open the project “grid_shots.3de”. In the following we will need essentially the Zoom Window and the Distortion Grid Edit Controls, but you can also use the environment Lens Distortion Editing. Select the frame “Seat” in the Frames Window.*

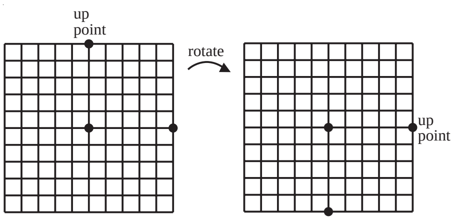


Please have a look at the Zoom Window. You see a lane marking in the lower half of the image. In reality this is a straight line, so we can utilize its curvature in order to determine lens distortion.

- *Please activate the translate button of the grid and move the grid such that the center point touches the upper border of the lane marking, approximately horizontally centred. Use the middle and the right mouse button in order to transpose and zoom the display area of the Window, so that you can control the grid more precise. Finally zoom out again.*

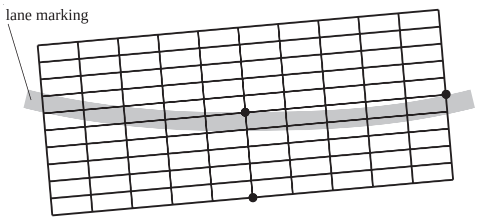
The measurement will be precise if you use the full length of the straight line in the sequence. For reasons we will explain later it is more convenient to match the grid by means of the center-point and the up-point rather than the center point and the right-point, therefore

- *Activate the rotate button and turn the grid around by about 90 degrees clockwise (i.e. drag the mouse to the left).*



- *Activate the scale button of the Zoom Window and scale the grid horizontally until almost the whole width of the image is covered. In order to do this, click the grid near its border with the left mouse button and drag the mouse. It will be useful if the up-point still lies within the image.*

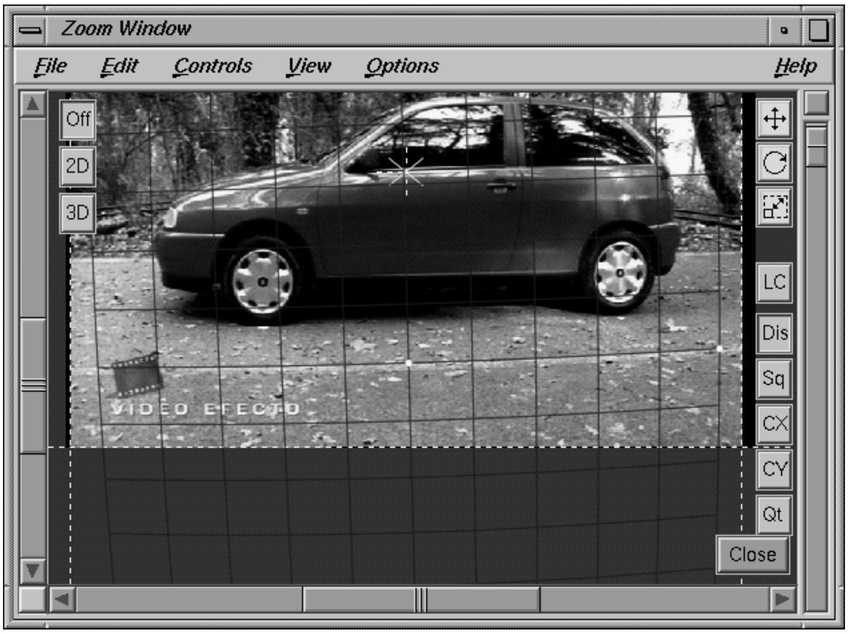
- *Rotate the grid so that the up-point (which is at the right border of the image) touches the upper border of the lane marking (like the center-point). Use the translation and zooming functions of the display area in order to be able to work precisely.*



Center and up point properly arranged

- *Now please activate the button “Dis” which allows you to vary the parameter lens distortion. Click the left mouse button, drag the mouse to the left and to the right and watch the line which connects the center-point and the up-point: you will be able to modify the parameter in such a way, that the entire grid line (also the part on the left hand side of the center point) matches the upper border of the lane marking.*

Now, if you look at the Camera Adjustment Window, the value “lens distortion” should have assumed a value of about 0.05. Your value will not precisely coincide with our value, and also if you do more than one measurement of this kind you will get different results. This is due to the fact that your and our “optical” interpretation of the “upper border of the lane marking” is different. It is absolutely common in every measurement process that the result exhibits some uncertainty (one attempt is enough, but if you want to be sure you can repeat the procedure twice or thrice, write down the results and take the average value). This completes the measurement of the lens distortion.



Final result (lens distortion = 0.053)

Why did we use the up-point instead of the right-point in order to match the line? The reason is that if you tweak the lens distortion or any other parameter, the center-point and the up-point remain exactly at their position, while the right-point does not. For mathematical reasons we cannot maintain the position of all three points, so the choice was met to declare the center- and the up-point as fixed. You can, if you like use the right-point in order to match the grid. This also works, but in this case you will have to tweak lens distortion and the grid rotation alternately.

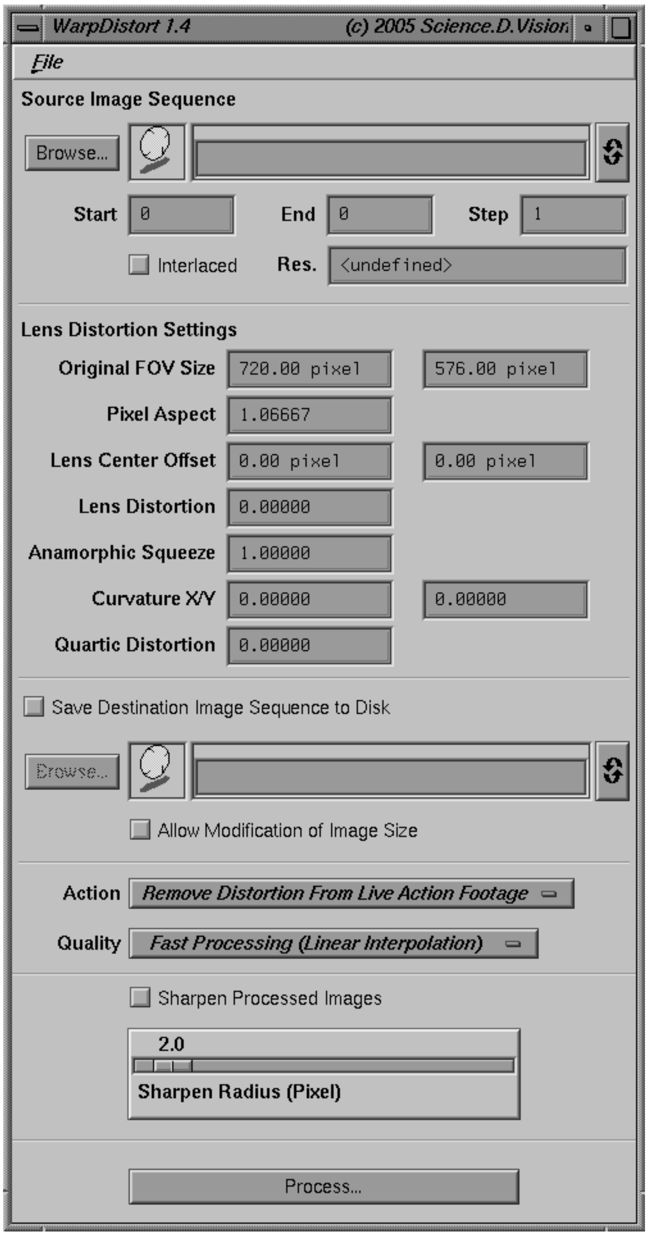
The procedure as we described it here works only if the straight line in the image does not go through the center of the field of view. In fact, it should be as far away as possible from the center, so that its curvature is bigger. A line through the center will not exhibit any curvature and cannot be used to measure the lens distortion.

Fisheye lenses

The distortion of fisheye lenses can be compensated using an appropriate combination of lens distortion and quartic distortion. The other parameters should be left on their default values, since anamorphic fisheye lenses cannot yet be handled properly with the given set of five parameters.

WarpDistort

When you process images with a significant lens distortion you will probably sooner or later run into the problem that your renderer is not able to generate distorted images (Houdini is able to do so, but their distortion model seems to be incompatible to ours. Mental Ray is said to be able, but we haven't checked this). The only general way to solve this problem is to apply an image operator to the recorded images in order to remove the distortion, or to apply an operator to the rendered images in order to create distorted rendered images. This is exactly what the tool WarpDistort (available when at least one 3DE film licence is present on your licence server) does. In the following we will see how WarpDistort works. It is quite easy to use, and we will use it in order to process the image “Seat” from the previous tutorial.



Source Image Sequence

Start 0 End 0 Step 1

□ Interlaced Res. <undefined>

Lens Distortion Settings

Original FOV Size 720.00 pixel 576.00 pixel

Pixel Aspect 1.06667

Lens Center Offset 0.00 pixel 0.00 pixel

Lens Distortion 0.00000

Anamorphic Squeeze 1.00000

Curvature X/Y 0.00000 0.00000

Quartic Distortion 0.00000

□ Save Destination Image Sequence to Disk

□ Allow Modification of Image Size

Action Remove Distortion From Live Action Footage

Quality Fast Processing (Linear Interpolation)

□ Sharpen Processed Images

Sharpen Radius (Pixel) 2.0

WarpDistort Main Window

- *Please start WarpDistort (in the following called WD) by entering "warpdistort" in a shell (SGI / Linux), or select the entry "WarpDistort" from the Start menu (Windows NT).*

Two windows open. One is titled ImageView and the other one is WarpDistort's main user interface window. The Image View Window displays WarpDistort's input and output data (i.e. the image before and after applying the image operator). The workflow is essentially as follows: you tell WD which 3DE project file you intend to process. WD asks you which reference frame or sequence exactly you want to handle, because WD processes one reference frame or sequence at a time. The reference frame or sequence you choose is associated to a camera object within the project file, and the distortion parameters of this camera are loaded into WD.

- *Select File::Import 3D-Equalizer Project. Select the project from the previous tutorial, "grid_shots.3de". When WD asks you to select a frame object, please select the image "Seat".*

Normally, when you select a sequence, the text field Source Image Sequence contains the file pattern of the images of that sequence. In this case it is only a single reference frame. The text field Original FOV Size contains the field of view of the image as it had been defined in the project file. The lens center offset is now given in pixel, not in length units, because we do not really deal with camera geometries here, but merely with image files, i.e. two dimensional objects. The lens center offset (in pixel) indicates how far the camera's direction of view is offset with respect to the center of the image file.

- *Please turn on the toggle "Save Destination Image Sequence To Disk" and enter an appropriate file name in the accompanying textfield, e.g. "/tmp/seat_undistorted.sgi".*

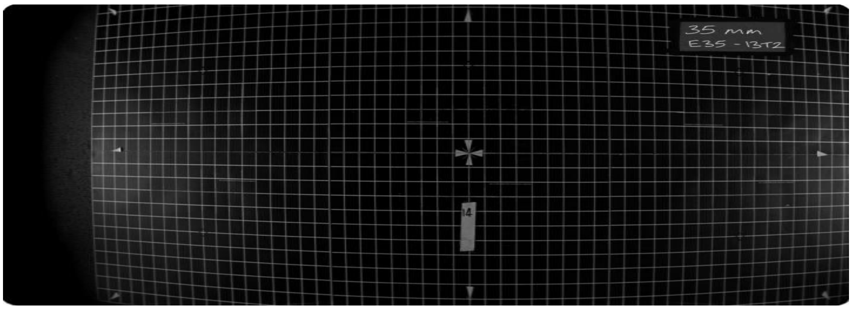
The output file format is the same as the input file format, i.e. WD does not convert file formats. Below the textfield you see a menu "Action". By default, it is set to the value "Remove Distortion...". This means, that WD is going to remove the lens distortion from the images given in Source Image Sequence. Furthermore, there is a menu "Quality", which can be set to "Fast Processing (Lienar Interpolation)" or "Slow Processing (Bicubic Interpolation)". The two settings are directly fed into SGI's image library, and it would not be justified to call the quality of one setting "higher" than the quality of the other setting. They are just, let's say, different. For many purposes, linear interpolation is sufficient. The process of resampling an image (be it linear or bicubic) is accompanied by a slight loss of sharpness. When you activate the toggle "Sharpen Processed Images" you can compensate this effect. The default value of 2.0 pixel for the sharpen radius is probably appropriate, and higher values will take more time.

- *Please press the button "Process...".*

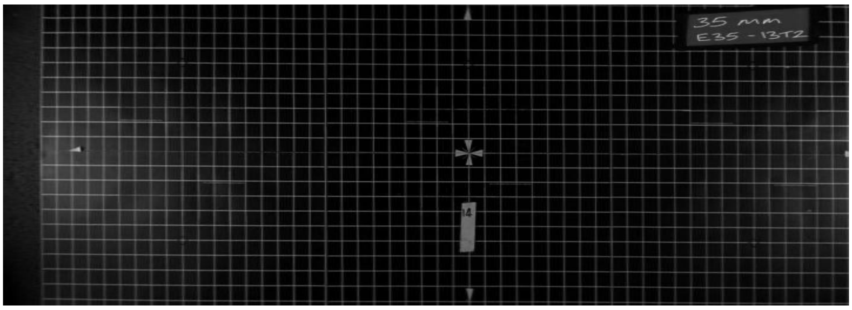
Several requesters appear and indicate the status of the calculation. After a few seconds the calculation is over and the Image View Window shows the undistorted image. You see that the black edges at the left and the right are almost gone. This is due to the fact, that removing lens distortion from an image is done by projecting the colour value of each pixel position to a new position which lies further away from the lens center than its original position (for positive values of the lens distortion). After removing the distortion, the black edges lie outside the valid area and are not part of the image file any more. In order to check the distortion of the new image "/tmp/seat_undistorted.sgi" we suggest the following exercise:

- *Open the project "grid_shots.3de" in 3DE. Activate the Lens Distortion Edit environment. Add a new camera in the Camera Adjustment Window. In the Frames Window, add a reference frame object containing the image file "/tmp/seat_undistorted.sgi" which you generated with WD. The new frame should be associated to the new camera object you created before. Adjust the field of view in the Zoom Window. Try to match the grid to the lane marking as described in one of the previous tutorials.*

When you have done this, the resulting lens distortion in the Camera Adjustment Window should be something around 0.00 or 0.01, i.e. the lens distortion is now removed. If you do the same procedure with an image in which the straight line is a little clearer, you can even get closer to the perfect result of 0.000, when you measure the distortion of an undistorted image.



Original gridshot



Undistorted gridshot

Another suggestion is to do the same exercise with some of the grid shots, e.g. grid 9 (see figure above). As you remember, it was a little bit tricky to match this grid shot because it was recorded with an anamorphic camera, and at least four of the six camera parameters had to be modified.

Applying lens distortion to rendered images

So far, we have removed lens distortion from distorted images. It is also possible to apply distortion to an undistorted image (usually a rendered image sequence). Certainly the rendered sequence is supposed to be composed to a real sequence which you have processed with 3DE. In order to apply lens distortion to the rendered sequence you must open the corresponding 3DE project file. However, replace the Source Image Sequence file pattern by the file pattern of your rendered images and change the menu “Action” to the setting Apply Distortion To CGI Sequence. Use all other interaction components of the WD Main Window as usual. The result will be an image sequence which exhibits the same lens distortion as the original images.

Tutorial 11: Zooming

In this tutorial we demonstrate how to process zoomed sequences, i.e. sequences in which the focal length varies over time. We will have to distinguish essentially two different situations:

1. Sequences which have been recorded with a camera fixed to a tripod, i.e. the camera may rotate around all three axes, but its position is fixed. In tutorials “Fixed Camera Position” and “Reference Frames” a sequence of this type was discussed. Additionally, we will now allow the focal length to vary over time.

2. Sequences with a moving object or a camera which is rotating and changing position. This situation is a little bit complicated, and there are different techniques for handling it.

Zooming and Fixed Camera Position

In this example we will process a sequence which has been recorded by a camera which was fixed to a tripod. The absolute minimum number of points for each frame is two (otherwise 3DE will try to interpolate the missing information). However, we recommend to track at least three points in each frame, since only in this case postfiltering the orientation makes sense.

- *Please open the project “pan_zoom.3de” and activate the environment “Motiontracking”.*

In the Frames Window you see a sequence consisting of 66 frames (interlaced, PAL video resolution). We have already defined a point group called “blue box”, containing four points. All points have been tracked in the course of the sequence, as you can see in the Timeline Window. Each frame contains at least three points.

In order to get a better view of the sequence it will be useful to activate the function Main Window::Playback::Fill Imagebuffers. This takes a moment, but it enables you to scroll forth and back easily within the sequence.

You see that in this project two points are tracked for the entire sequence. It is not a required condition to have the same set of two points in each frame, but in this case it was quite convenient to do this.

The camera is slightly rotating on the tripod during the sequence, and obviously the focal length changes from a small value (i.e. short focal length, i.e. large aperture angle) to a larger value (i.e. long focal length, i.e. small aperture angle).

- *Please select the sequence in the Frames Window (if it is not yet selected) and press “Modify...” in the same window.*

The familiar requester “Modify Sequence...” opens. As you see, the toggle button “Fixed Camera Position” is already turned on. Since this is a sequence with zooming:

- *Turn on the toggle "Variable Focal Length (Zooming)" and press "OK".*

This signals 3DE that the focal length of the images varies over time. It does NOT specify, how 3DE will obtain the focal length: the curve may be entered in the curve editor or it may be calculated by the system. The toggle only expresses the fact that focal length is not necessarily constant.

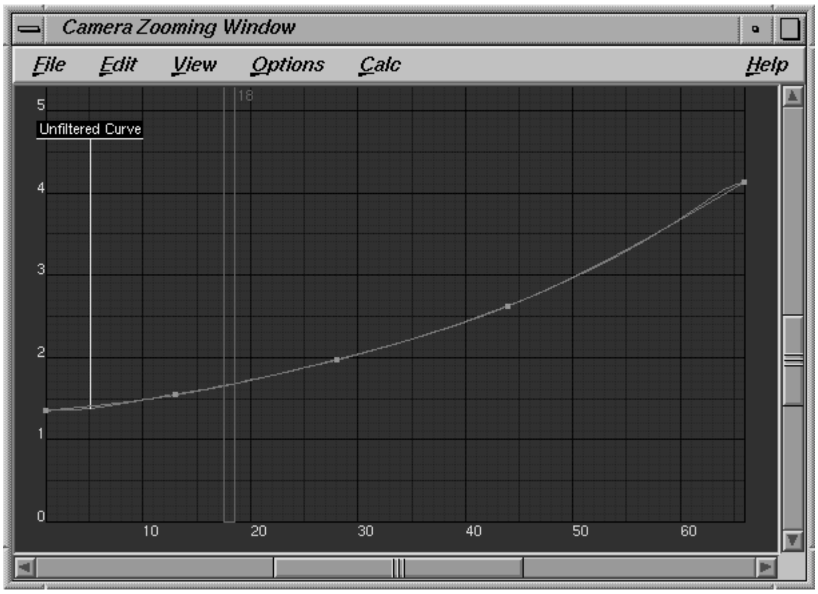
- *Please activate the environment "Zooming". Select the tab "Advanced View" in the Camera Adjustment Window and set the selectors Filmback Width, Filmback Height and Focal Length to "Fixed".*

The most interesting window for our purposes is the Camera Zooming Window. Let us recall how projects without zooming are done in the simplest case: you track as many points as necessary, adjust the field of view, enter the camera parameters and press Main Window::Calc::Calc All Objects from Scratch. If you do this, all camera / object motion paths and pointgroups are calculated. For zoomed sequences (when all the points are tracked and so on) some of the calculations are done in the Camera Zooming Window, namely calculating the zoom curve of the zoomed sequences (but this also contains calculating the camera motion path). Once the zooming curve is calculated, filtered and edited, you can modify the project, e.g. correct the motiontracking curves of the points, correct some of the camera parameters and so on. If you do so, it is not always necessary to calculate everything from scratch. It is sufficient to calculate only the "common stuff" i.e. camera/object motion path by pressing Main Window::Calc::Calc All Objects from Scratch. This saves a lot of time.

However, as you see in the Camera Zooming Window, it seems that no reasonable zoom curve is available. Let us have a look at the display field of this window. The meaning of the horizontal axis should be clear: here the frames are represented ranging from frame 1 (not zero) to frame 66. The vertical axis starts at 0 and goes up to 2. You see a blue, straight, constant line at the value of 1. When this curve represents the focal length, how does this relate to the fact that we also represent the focal length in the Camera Adjustment Window? The answer is that, in the Camera Adjustment Window, the value of the focal length is some kind of "nominal" value, in this case 5.9mm. The focal length curve expresses the factor by which the value in the Camera Adjustment Window is modified. In this example 3DE obviously assumed so far that the focal length remains at 5.9mm for every frame. We will examine the connection of the focal length value again later.

- *Please press Camera Zooming Window::Calc::Calc Zooming Curve From Scratch.*

Several messages appear indicating the current status of the calculation. After a minute the calculation is over and you see a blue curve which represents the time-dependent focal length factor for this sequence. It starts at a value of 1.354 and ends at a value of 4.132. The Status Window shows an average value of about 0.1 pixel which indicates that the tracking was very precise and our calculation went quite well. The display field of the Zoom Window can be moved around and scaled, as can the other function browser and editor windows (Status Window, Motion Path Window).



Zoom curve (fixed camera position)

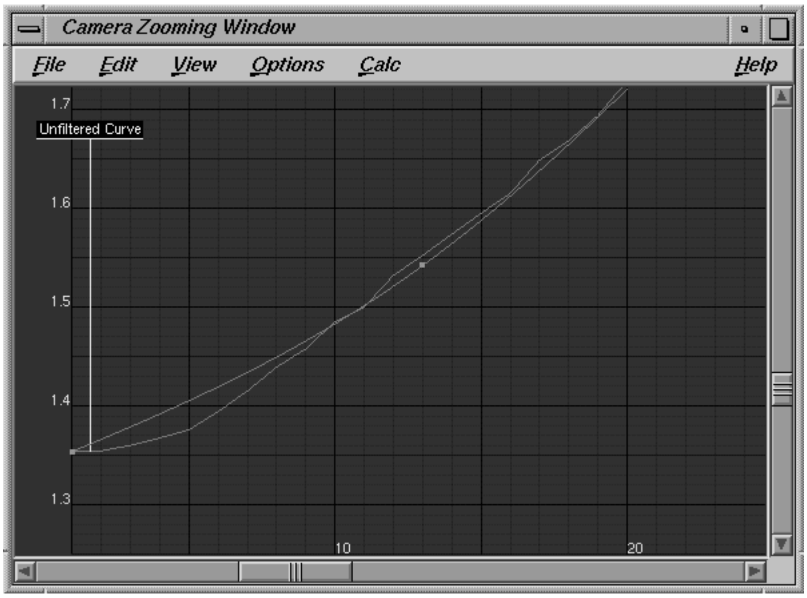
- *Please examine the curve in the frame range 1 to 10. Magnify the curve horizontally and vertically.*

You see a grey line under the blue line. This grey line is, what 3DE originally calculated. It is a little bit jaggy, because the calculation is based on motion tracking data, which naturally exhibit statistical errors. The blue line is obtained from the grey one by filtering, similar to the postfiltering of camera position and orientation, which you already know. Let us examine in detail how the zoomfilter works.

- *Please activate the menu Camera Zooming Window::Options::Zoom Filter Settings.*

The requester Edit Zoom Curve Filter Settings opens. You can now try out different values of the filter and watch the result in the Camera Zooming Window.

- *Drag the knob to the left, say 5.0, release the mouse button and watch the blue curve.*



Originally calculated curve and filtered curve

The blue curve is now much closer to the grey, jaggy one then before, i.e. the filter effect is not as strong as before.

- *Please set the value to 1.0 (left-most position).*

Now, there is no filter effect at all and the blue curve coincides with the grey one. You may have seen that the blue curve is controlled by keyframes. The position of these key frames is calculated by 3DE. For a given zoomfilter setting, the system tries to use as few keyframes as possible (in some sense) in order to represent the curve. For a strong filter less keyframes are required, because the curve is very smooth. For a weak filter many keyframes are required, in the extreme case even every frame can be declared as keyframe. In our example here a zoom filter strength of 15 is too high, because there is already a noticeable, systematic discrepancy between the blue and the grey curve at the beginning and the end of the sequence. We suggest using a value of about 3 to 4. In principle, the data can now be exported to the animation package, or you can generate a preview movie. From here on the procedure is as already demonstrated in the other tutorials.

As mentioned before, the “real” value of the focal length results from multiplying the zoom curve to the focal length in the Camera Adjustment Window. Let us examine the relationship between the focal length value and the zooming curve. Remember that for a value of 5.9mm the curve goes from 1.354 up to 4.132.

- *Please turn on the toggle buttons Camera Zooming Window::View::Show Zoom Values and Display Zoom Values As::Focal Length.*

The cursor in the Camera Zooming Window is now accompanied by a label which shows the frame-dependent focal length value. It ranges from 7.988mm to 24.381mm.

- *Please set the menus of filmback width, height and focal length to fixed. Change the value of the focal length to 11.8mm, but take care that the Filmback Width and Height remain the same. The horizontal angle will change to 24.3 degree.*

You have now multiplied the focal length by 2.0. If you were about to calculate an unzoomed sequence, this would cause a considerable deviation in the Status Window, or the calculation might even fail. But here the situation is different:

- *Press Camera Zooming Window::Calc::Calc Zooming Curve From Scratch.*

After a minute you see the new zooming curve. It now ranges from 0.679 to 2.072, which is approximately half of the values obtained before. As you move the frame slider you will note that the focal length now ranges from 8.011mm to 24.450mm, which is (barring some numerical uncertainties) very close to the result obtained before. As you see, if you double the focal length value, the curve values are halved.

Important note: It is recommended to set the focal length in the Camera Adjustment Window at least roughly to some average value (average with respect to all the values that occur during the zooming).

Note also that the lens distortion parameters should remain at their default value.

Zooming, rotating and translating camera

When the camera is performing the most typical type of movement, namely rotation and translation, we distinguish two different methods in order to calculate the zooming curve. One method requires that reference frames are available, and the other method requires that one has at least a rough idea of how the Zoom curve should look. In the following we will demonstrate how these two methods work.

Reference frames available

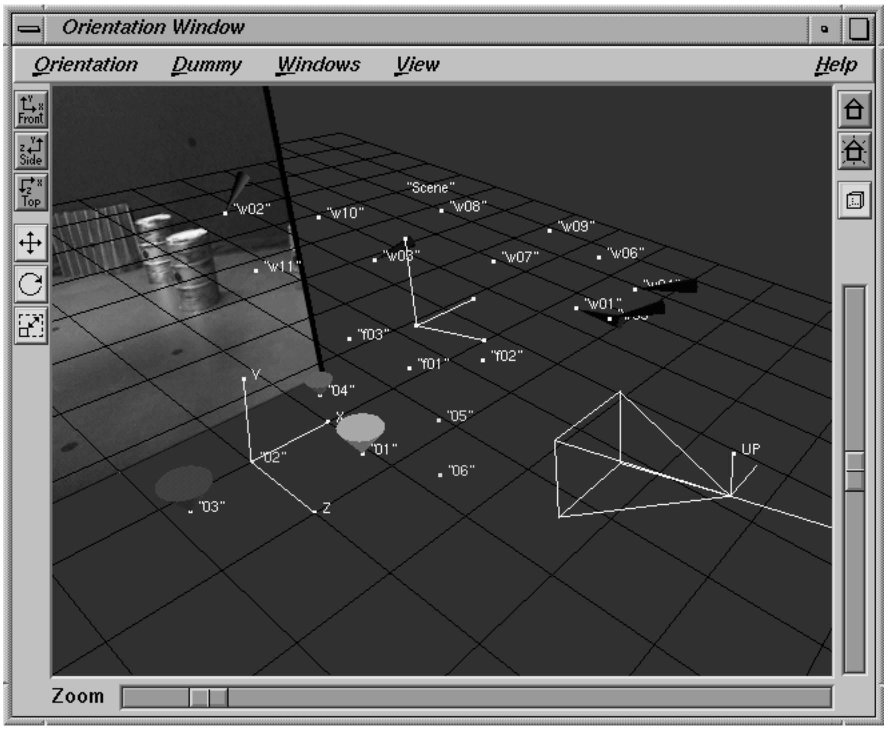
In this section we describe how to process a sequence which has been recorded with a camera which changes position, orientation and focal length over time. The basic idea within 3DE is that one has to provide sufficient information to enable 3DE to calculate a point model, i.e. the position of all points of the pointgroup. Usually this is done by means of reference frames, but the point positions can also be entered by the user. In the following tutorial we will use reference frames in order to calculate the points. Since all aspects of motiontracking and also the use of reference frames have been explained in the other tutorials, we will generally load prepared projects instead of doing every single step.

- *Please open the project "zoom_dolly_ref_only.3de". Open the Frames Window, the Overview Window and the Camera Adjustment Window.*

The projects contains six reference frames. The reference frames are recorded with the same (video) camera and the same focal length, therefore it is permissible to associate them to the same camera object. The field of view is adjusted properly. The horizontal angle, however, is probably wrong (45 degree is only a default value).

- *Set the menu of the horizontal angle to "Fine Adjust" and press "Adjust".*

After a few minutes the horizontal angle is calculated. The result should be about 43.548 degree (if you do not want to wait, you can also enter the value directly and press Main Window::Calc::Calc All Objects From Scratch). In the Orientation Window you can ascertain that the point model has been reconstructed.



Point model

The next step is to create a sequence object and to track as many points as necessary. The minimum number of points in each frame is six (or two regular and two infinite points, if you are using the feature “infinite points”). However, it is recommended to use at least seven or eight points rather than six. It should be obvious that it is quite important to do the motiontracking as precisely as possible, in order to get numerically stable and precise results.

- *Please press Frames Window::Add and add a sequence.*

The requester Create Sequence opens.

- *Click on the button Sequence Settings::Browse and import the image sequence “dolly_zoom_1” located in the directory “<tutorial install directory>/dolly_zoom_1/”. Set the image type to “Interlace - Odd First” and close the requester.*

Now the points which are already in use in the reference frames should be tracked in the sequence as well:

- *Turn on the toggle button Overview Window::View::Show Point Names. Examine the various reference frames and sequence frames. Pick any point you can identify in both, the reference frames and the sequence; e.g. points “f01” - “f03” on the barrels visible in the sequence as well.*

- *Track the points you can identify or open the project “zoom_dolly_ref_camad_seq.3de”.*

If you move the frame slider in the Main Window you can get an overview of the sequence. With a little experience you can also judge by eye that the focal length is varying over time. However, as often in zoomed sequences, it is difficult to distinguish between a zoom and a movement in the camera’s direction of view. This “ambiguity” is sometimes used to create a special effect, which is called the “Trombone Effect” or the “Vertigo Effect”, since it became famous in A. Hitchcock’s “Vertigo” [4]. There, a camera movement along the (vertical) direction of view was accompanied by a zoom in order to visualize the protagonist’s acrophobia.

- *Doubleclick the line containing the sequence name in the Frames Window in order to open the requester Modify Sequence... .*

This is an interlaced video sequence in PAL resolution 720x576 of 247 images. It was in fact recorded with a zoom lens, therefore:

- *Turn on the toggle “Variable Focal Length” and close the requester.*

The field of view was properly adjusted for the reference frames. However, we have not yet done it for the sequence.

- *Please open the Zoom Window. Press the Ctrl key, click and drag the corners of the field of view until all four dotted lines line up with the visible part of the image.*

Since the reference frames and the sequence frames are associated to the same camera (and hence share the same pixel aspect), it is only possible to apply modifications to the field of view which maintain the aspect ratio, and these modifications can only be done by dragging the corners. A systematic way to do this has been demonstrated in some of the previous tutorials, but we will give a short summation here:

- *Press the Ctrl key, drag the upper right corner until the upper dotted line lines up with the top edge of the image.*

- *Press the Ctrl key, drag the lower right corner until the right dotted line lines up with the right edge of the image.*

- *Press the Ctrl key, drag the lower left corner until the lower dotted line lines up with the lower edge of the image. Because of the constraint that fixes the aspect ratio, the left dotted line should now line up with the left edge of the image.*

In the following we will let 3DE calculate the zoom curve, so:

- *Please turn on the environment "Zooming".*

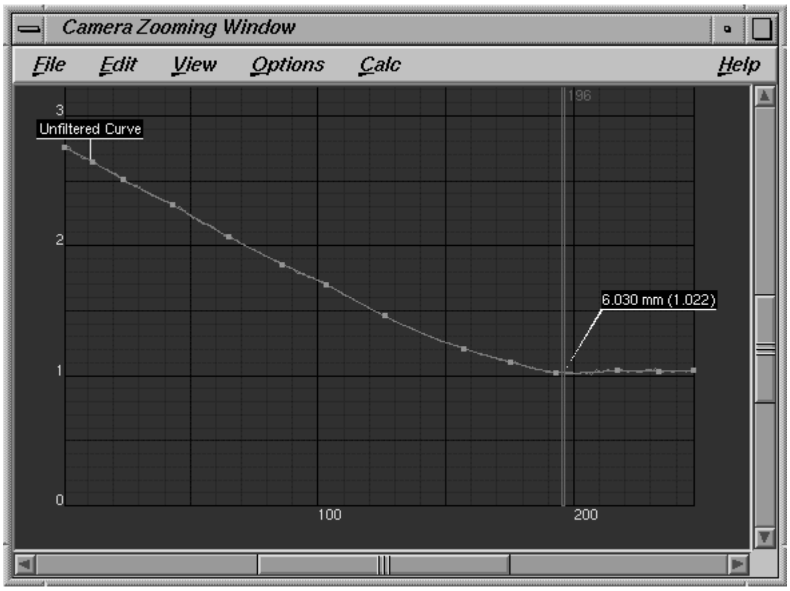
The Zooming Window should now display a blue straight line indicating a constant value of 1. The interpretation of this line is exactly as in the previous tutorial: it is the "zoom factor", a number which is multiplied to the focal length value in the Camera Adjustment Window in order to yield the "true" value of the focal length for a specific frame. The strategy is similar to the case of a fixed camera position: we calculate the zoom curve for the sequence; the postfiltering is done automatically (using a default strength of 15.0). Afterwards it is possible to adjust zoomfilter strength interactively to tweak the tracking, to render a preview, to edit the motion data and so on. It is possible, of course, to modify the tracking curve by hand, but once we have a reasonable zoom curve, it is not necessary to calculate it again and again.

- *Please press Camera Zooming Window::Calc::Calc Zooming Curve From Scratch.*

The system now calculates the point model out of the set of reference frames and inserts the sequence afterwards. This may take a few minutes. The resulting zoom curve will look as illustrated in the figure below.

- *Turn on Camera Zooming Window::View::Show Zoom Values and Display Zoom Values As::Focal Length.*

As you see, for the first frame the zoom value is about 2.76 and for the last frame the value is about 1.03. In fact, we started recording the sequence with a long focal length, and at a certain point (around frame 195) the focal length assumed its minimum value. The label at the cursor displays a focal length of 16.267mm at frame 1 and 6.030mm at frame 247. The reference frames have also been recorded with this short focal length, i.e. a large aperture angle, in order to include as much of the blue box as possible. For the reference frames, the aperture angle was calculated by 3DE, and we used the same value for the sequence as well. Therefore, we could expect that the final value of the zoom curve would tend towards 5.9mm. The fact that we cannot (yet) handle time-dependent (or focal-length-dependent) lens distortion should explain that our final value is 6.03mm instead of 5.9mm.



Final zoom curve

If you magnify the curve in the Zooming Window you may note that the grey curve, which represents the originally calculated curve, is quite jaggy. The blue curve is the post-filtered version of the grey curve. The filter strength in this case is 15, and since the zoom was done by an electric servo, it is justified to assume that in reality the zoom curve is fairly smooth. Let us try another value:

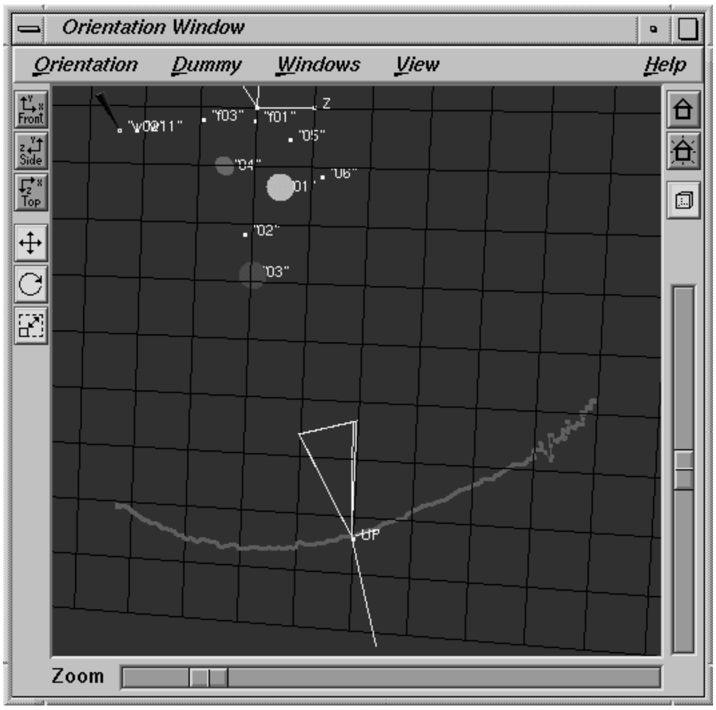
- *Please magnify the curve in the frame range of about 190 to 240. Use the middle and the right mouse button in order to translate and scale the Camera Zooming Window display area. Select the function Options::Zoom Filter Settings and set the parameter Zoom Filter Strength to about 5 or 6. When you drag the knob and release the mouse button, the blue curve in the display field will be updated immediately.*

When you use a value of 5, the blue curve is fluctuating softly at the end of the sequence. However, given the fact, that the zoom value is constant in that range, such a shape cannot be justified. The optimum value is, let’s say, a matter of taste, probably something between 10 and 15. A value of 30 is too high, because this would not correctly reflect the behaviour of our zoom lens in the range around frame 195, where the zoom servo “abruptly” stops.

The relation between zoom curve, camera motion path and status curve

We can also invoke the use of the zoom filter in a different way. To do this, you need the Camera Zooming Window and the Orientation Window. We suggest to follow the procedure below, we will explain the effect afterwards.

- *Please select the environment “Single Window” in order to clean up the desktop. Now open the Camera Zooming Window and the Orientation Window. Activate the toggle Orientation Window::View::Show Camera Path and watch the camera motion path from an above position, as shown in the figure below.*



Camera motion path

- *Select Camera Zooming Window::Options::Zoom Filter Settings and set the filter strength to 1.*

The blue curve in the Camera Zooming Window is updated immediately. There is no filter effect, and the blue curve is identical to the originally calculated grey one.

- *Select Camera Zooming Window::Calc::Make Rot.&Pos. Curves Consistent and watch the camera motion path in the Orientation Window.*

After a few seconds the camera motion path in the Orientation Window becomes very jaggy.

- *Select Camera Zooming Window::Options::Zoom Filter Settings and set the filter strength to 30.*

The blue curve in the Camera Zooming Window is updated immediately.

- *Select Camera Zooming Window::Calc::Make Rot.&Pos. Curves Consistent and watch the camera motion path in the Orientation Window.*

After a few seconds the camera motion path in the Orientation Window becomes more or less smooth, but around frame 192 there is a strange angle-shaped deformation.

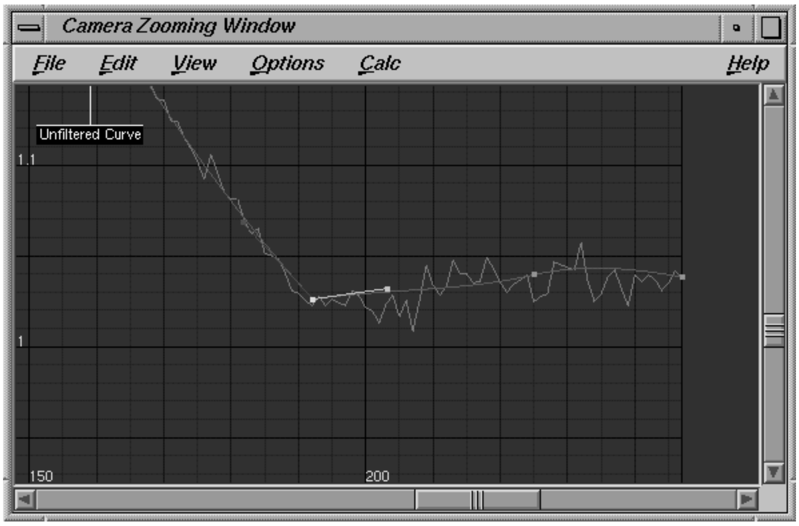
- *Modify the zoom curve by hand, e.g. drag one of the key frames at the beginning of the sequence up or down.*

The curve becomes red, which indicates that it has been edited.

- *Select Camera Zooming Window::Calc::Make Rot.&Pos. Curves Consistent and watch the camera motion path in the Orientation Window.*

You will note that the camera motion path now exhibits a similar deformation to the one you applied to the zoom curve. We have now a bunch of phenomena which should be explained. The central function in this procedure has been “Make Rot.&Pos. Curves Consistent”. Whenever the zoom curve is modified, 3DE’s optimal solution to the match moving problem (which does not necessarily coincide with what the user wants) is falsified. More precisely, there is an ‘inconsistency’ between the zoom curve on one hand and the position and rotation curve on the other hand. When you select the function “Make consistent” you force 3DE to re-establish the relationship between the zoom curve and the camera motion path. 3DE does this by modifying the path. The principle is very similar to that in the tutorial section “Stabilizing”. There, we suggested modifying the positional curves of the camera by hand in the Motion Path Window. Consistency was restored by the function “Make Rotation Curves Consistent” in this window.

As mentioned, it is hard for the system to distinguish zooming from translation along the camera view direction. In the above procedure you have set the filter strength to 1, i.e. effectively removing the filter. The result was a jaggy zoom curve and a jaggy motion path. Every jag in the zoom curve is accompanied by a compensating jag in the motion path. When you set the filter strength to 30, all jags in the zoom curve are removed, even the only one that was correct: the change of zooming velocity at frame 192 that resulted from stopping the zoom servo. So, this was a little too much filtering. 3DE’s reaction was to compensate this by adding an angle to the camera motion path. It is hard to find a perfect number for the filter strength so that the motion path is not jaggy and the angle at frame 192 vanishes. Probably it is necessary to use a high value for the filter and add a jag in the zoom curve by hand, in order to simulate the shutdown of the engine. We have done this in the figure below.



Zoom curve, postfilter strength 30, edited

As in the tutorial section “Stabilizing” you can repeat the above procedure and check the effect in the Zoom Window (activate Zoom Window::View::Show Reconstructed Points). It is also possible to check how “far” you move away from what 3DE believes to be the optimal solution. Remember that 3DE tries to minimize the deviation curve displayed in the Status Window.

- *Please open the Status Window and arrange the status curve in a way so that you see it completely. Modify the zoom curve slightly by dragging one of the key frames or inserting a new key frame (ALT + left mouse button). Do not forget to press Camera Zooming Window::Calc::Make Rot.&Pos. Curves Consistent afterwards. Now press “Update” in the Status Window and watch the status curve.*

After a few seconds the curve is modified, and the frame range you modified probably exhibits some large error. In practice, it does not make too much sense to lower the quality of the calculated result, however, by means of this example you may understand the relationship between the zoom curve, the camera motion path and the status curve. In the following tutorial section we will process a sequence without providing reference frames, but with some zoom data given.

No reference frames available, estimated zoom curve available

Let us begin with a theoretical example: assume you want to process a sequence of 300 frames for which you do not have enough reference frames, or the motion tracking was not precise enough to calculate a reasonable zoom curve. On the other hand maybe you have the information that the focal length started at 150mm in frame 1, goes down to 75mm in frame 200 and remains constant until frame 300. The filmback width and height may be 2.10cm x 1.13cm. Now the camera geometry is defined in principle: you enter the filmback value in the Camera Adjustment Window. You can set the focal length to 75mm. Now it is quite easy to "guess" what the zoom curve looks like. Since you know that it starts at 150mm, the value of the curve at frame 1 assumes the value 2.0. Then the curve goes down to a value of 1 in frame 200 and remains at 1 until the end of the sequence. In order to enter this zoom curve you need two or three keyframes. Let us do a similar procedure for the sequence we already used in the previous section.

- *Please open the project "zoom_dolly_seq_only.3de" and activate the environment "Zooming". Turn on Camera Zooming Window::View::Show Zoom Values and Display Zoom Values As::Focal Length.*

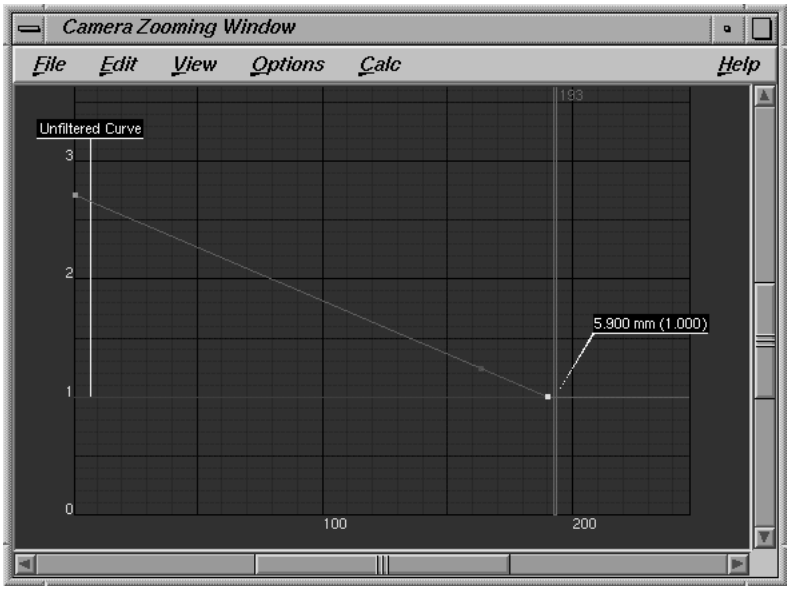
In this project we removed all the reference frames, so that only the sequence frames remain. The zoom curve is constant at a value of 1. We know from some data sheet that the camera has a filmback of 0.508cm times 0.381cm. The minimum focal length is 5.9mm.

- *Please enter these values in the Camera Adjustment Window.*

The director says "we started with a focal length of 16mm, and at frame 190 the focal length was 5.9mm and then the zoom wasn't changed until the end". Now, 16mm corresponds to a zoom factor of 16mm divided by 5.9mm which is 2.7 (some zoom lenses are labelled with the zoom factor directly instead of the absolute focal length value).

- *Please set the frame cursor to frame 1. Place a point in the Camera Zooming Window in frame 1, at a value of 16mm (that is a zoom factor of 2.7). You can do this by pressing ALT and the left mouse button at the position where the point is supposed to appear. In order to work precisely, use the middle and the right mouse button in order to translate and magnify the location of interest. Place another point in frame 190, at a value of 5.9mm (that is a zoom factor of 1.0).*

Since we now have a rough estimate of the zoom curve, we can assign 3DE to analyze it. We do NOT calculate the zoom curve now, since we have already entered it.



Estimated, edited zoom curve

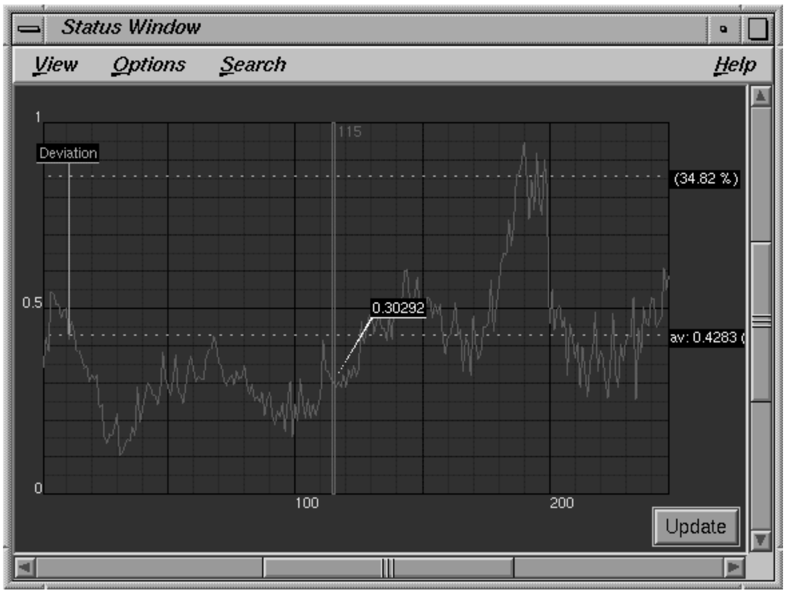
- *Please press Main Window::Calc::Calc All Objects From Scratch*

After a minute the calculation terminates and you can have a look at the Status Window. The average value should be around 0.4 to 0.5, but there are frames with a deviation of about 1 pixel (see figure below).

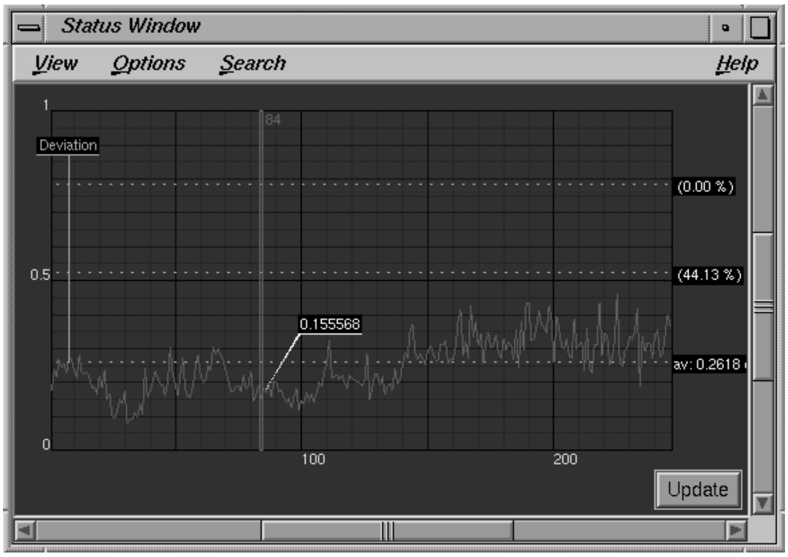
We should try to improve this result. There are two ways to do this: tweaking by hand, or using the finetuning function. When there are enough points and a sufficient amount of perspective information (which is the case here) you can do the following:

- *Press Camera Zooming Window::Calc::Finetune Zooming Curve. A requester will ask you to confirm this action, because the curve including possible editings will be changed.*

3DE performs a few so-called finetuning operations (20 by default, the value can be changed in Main Window::Options::Calculation Settings). After half a minute or so the zoom curve and the status curve are updated, and the status curve should look better now, with an average value of about 0.25 to 0.27 (see figure below). The other means of improving an estimation of the zoom curve is to tweak the curve by hand, press “Make Rot.&Pos. Curve Consistent”, press “Update” in the Status Window and watch the status curve, as we already described in the previous section. When you have done three or four of these working cycles, you should press Main Window::Calc::Calc All Objects From Scratch in order to “refresh” the result.



Status curve before zoom finetuning



Status curve after zoom finetuning

Tutorial 12: Matchmoving Non-Rigid Objects

It has been a fundamental pre-condition for all matchmoving projects we have seen up to now, that all tracking points had to be static with respect to each other. We will now discuss a new technique that allows you to reconstruct the motion of individually moving points in three-dimensional space.

In general, this new technique can be used for two different tasks, namely producing "classical" motion capturing data as well as matchmoving non-rigid objects. A new fundamental pre-condition for this type of projects is that a non-rigid object to be tracked must be recorded with multiple cameras simultaneously, thus producing a separate sequence for each camera. The cameras can operate at different frame rates, and should view the non-rigid object from different angles. Usually the cameras should be totally fixed to the ground, however under certain circumstances it is possible to move at least one of them around.

In the following tutorial you will see how to track the face of a talking actress. For this purpose three fixed cameras have been used.

- *Please start 3D-Equalizer as described in the section "Getting started" and select the menu item Main Window::Projects::New to create a blank project.*

First of all, let's do some basic preparations to setup a 3DE mocap project.

- *Select the environment "Motiontracking" from the Main Window's environments menu.*

We will create three sequences, one for each camera. Two of these sequences have been shot with the same camera model, whereas the third one has been shot with a different camera.

- *Please click on the Add button of the Frames Window.*

A requester opens and asks you whether you want to create a sequence or a reference frame.

- *Click on the "Add Sequence" button to create the first sequence object.*

The Create Sequence requester opens, which allows you to define all the settings associated to the new sequence object.

- *Press the button Sequence Settings::Browse and go to the directory "<tutorial install directory>/mocap/left/". Double-click on the first image file "jo_left.0000.jpg".*

3D-Equalizer will scan the directory, starting with the selected file, in order to determine the number of image files associated to the sequence. After that, the Identifier, Pattern, Start and End textfields are filled with meaningful values.

The sequence is noninterlaced (pro-scan) PAL video at a frame rate of 25 frames per second.

- *Please enter a value of 25.0 in the Rate Fps textfield and press the Ok button, to create the new sequence object.*

You see a woman who has a couple of dark markers attached to her face (see figure below). Let’s continue to import the remaining sequences. The following one has been shot with the 2nd camera and is interlaced PAL video at 25 frames per second.

- *Create another sequence and import all image files located in directory “<tutorial install directory>/mocap/front/”, enter a value of 25.0 in the Rate Fps textfield and set the Imagetype option menu to Interlace - Odd First.*

The last sequence is noninterlaced (pro-scan) PAL video at a frame rate of 25 frames per second, again.

- *Create the final sequence object and import all image files located in directory “<tutorial install directory>/mocap/right/”, enter a value of 25.0 in the Rate Fps textfield and leave the Imagetype option menu at Noninterlace.*

In 3DEqualizer’s Frames Window there will be three entries, each of which represents a separate sequence object. Next, we will setup two camera objects and properly link the sequences to them.

Actress simultaneously filmed from three different points of view

- *Please select the environment “Camera Adjustment” from the Main Window’s environments menu.*

The present release of 3DEqualizer cannot determine camera parameters for mocap projects automatically. Therefore we have to setup both cameras manually.

Important note: If you have no idea about the camera parameters we recommend to setup a regular matchmoving project in order to determine them. For instance, put six markers on a board and move it in front of the camera you’d like to adjust.

- *Rename camera “default camera” to “SONY VX1000”, set Horizontal Angle to 45.0, Pixel Aspect to 1.06666 and Lens Distortion to 0.04.*

- *Click on the button Camera Adjustment Window::Add to create a 2nd camera object. Rename the newly created camera to "JVC DVX8", set Horizontal Angle to 35.9, Pixel Aspect to 1.06666 and Lens Distortion to 0.059.*

The sequences "jo_left" and "jo_right" have been shot with two identical JVC cameras, sequence "jo_front" has been shot with a Sony camera. Let's link the sequence objects to their respective camera objects.

- *Please select the environment "Motiontracking" from the Main Window's environments menu.*

- *Double-click on the entry of sequence "jo_left" in the Frames Window to open the requester Modify Sequence. Then, click on the button Camera::Browse, to open the requester Choose a Camera, select camera object "JVC DVX8", and click on the button Ok. To confirm your edits, click on the button Ok of the Modify Sequence requester.*

Sequence "jo_left" is now linked to camera "JVC DVX8", which is indicated by its entry in 3DE's Frames Window. Sequence "jo_front" is already linked to camera "SONY VX1000", so there is no need to apply any changes. However, sequence "jo_right" is not linked to the correct camera, so let's modify it, as well.

- *Double-click on the entry of sequence "jo_right" in the Frames Window to open the requester Modify Sequence. Then, click on the button Camera::Browse, to open the requester Choose a Camera, select camera object "JVC DVX8", and click on the button Ok. To confirm your edits, click on the button Ok of the Modify Sequence requester.*

As we already discussed above, all sequence objects used in a mocap project have to be synchronized to the same timeline. In addition, the sequences of this tutorial have been shot using totally fixed cameras. Let's edit our sequence objects accordingly.

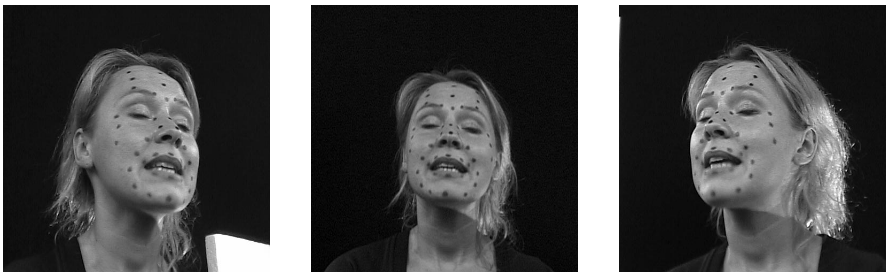
- *Double-click on the entry "jo_left" in the Frames Window to open the requester Modify Sequence. To enable motioncapturing click on the toggle button Sequence is Synchronized to other Sequences. Finally, set option menu Camera Constraints::Position to Fixed Camera Position Contraint, and enable No X-, No Y- and No Z-Rotation toggle buttons. To confirm your edits, click on the button Ok.*

- *Repeat the previous action with both remaining sequence objects "jo_front" and "jo_right", please.*

Finally, we have to tell 3DEqualizer how exactly our sequences are synchronized to each other. In order to do so, you'll have to define a so-called Sync Frame for each sequence separately. The Sync Frames of all sequences should show the non-rigid object in exactly the same moment in time (just taken from different points of view). Usually, a frame is chosen that shows some content which can be easily recognized in all other sequences (e.g. blink of an eye, flashlight, etc.).

- *Select sequence “jo_left” from the Frames Window and use the Main Window’s navigation elements to go to frame 24.*

In frame 24 of sequence “jo_left” the actress has closed her eyes. This action certainly can be found in the remaining sequences very easily (see figure below).

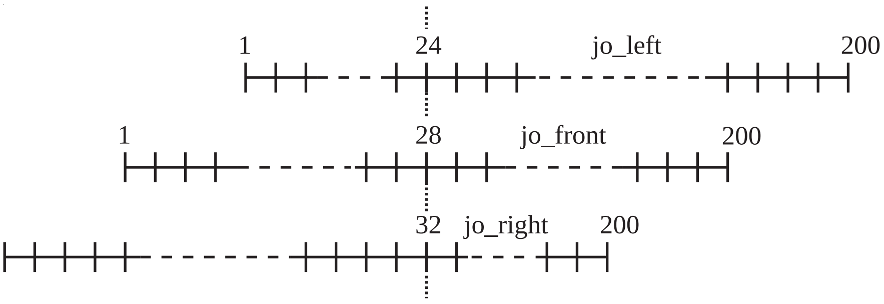


Sync Frames of sequences “jo_left”, “jo_front” and “jo_right”

- *Select the menu item Main Window::Special Frames::Set Sync Frame.*

You have now declared frame 24 to be the sequence’s Sync Frame. This is indicated by a small label in 3DE’s Overview Window and in the Main Window’s title as well as by a yellow bar in the Timeline Window. Let’s continue and define the Sync Frame of both remaning sequences.

- *Please select sequence “jo_front” from the Frames Window, go to frame 28 and select the menu item Main Window::Special Frames::Set Sync Frame. Finally, select sequence “jo_right”, go to frame 32 and select the menu item Main Window::Special Frames::Set Sync Frame one more time.*



Schematic view of synchronized sequences

All sequences should now be properly synchronized to each other (see figure above). As a last step to prepare our mocap project, you’ll have to create two pointgroup objects.

- *Select the menu item Pointgroups Window::Pointgroup::Add Pointgroup.*

The Create Pointgroup requester opens.

- *Make sure the option menu Pointgroup Type is set to Camera Group and press the button Ok.*

You have created an empty camera pointgroup, which is needed to hold some internal data structures. The 2nd pointgroup to be created will actually be filled up with some mocap tracking points.

- *Select the menu item Pointgroups Window::Pointgroup::Add Pointgroup, set option menu Pointgroup Type to MoCap Group and press the button Ok.*

A new mocap pointgroup has been created. The label “MOCAP” indicates that each of its points will be animated separately (see also project “mocap1.3de”).

Tracking points in mocap projects

In the following section we will briefly discuss specific details of point tracking in mocap projects. Generally, the same techniques as in tutorial “Multiple Sequences” are applied. In order to compute a 3D motion path for a given point, it has to be tracked in at least two different sequence objects.

- *Please select the environment “Motiontracking” from the Main Window’s environments menu.*

Let's track a single point to get an idea of the regular workflow.

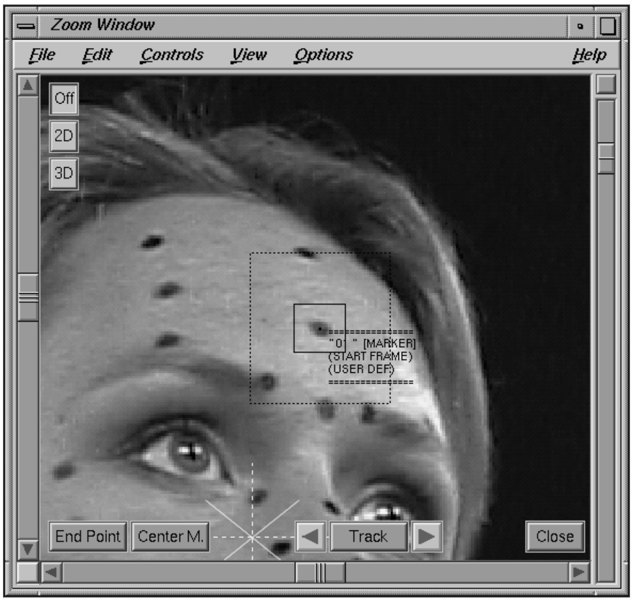
- *Select sequence “jo_left” in 3DE's Frames Window and use the Main Window's navigation elements to go to frame 1.*

- *Make sure mocap pointgroup “pgroup #2” is selected and press the button Pointgroups Window::Add to create a new mocap tracking point.*

The requester Create Point opens.

- *Set option menu Trackingmode to Marker and confirm your edits by clicking on the button Ok.*

A new marker tracking point labeled “01” has been created. Let’s track one of the markers located on the actress’ face.



Tracking a mocap point

- *Use 3DEqualizer's manual motiontracking functions to track the marker that is located in the middle of the actress' forehead (see figure above).*

The actress' forehead marker is now tracked by point "01" in sequence "jo_left". Let's continue and track the same marker in the two remaining sequences.

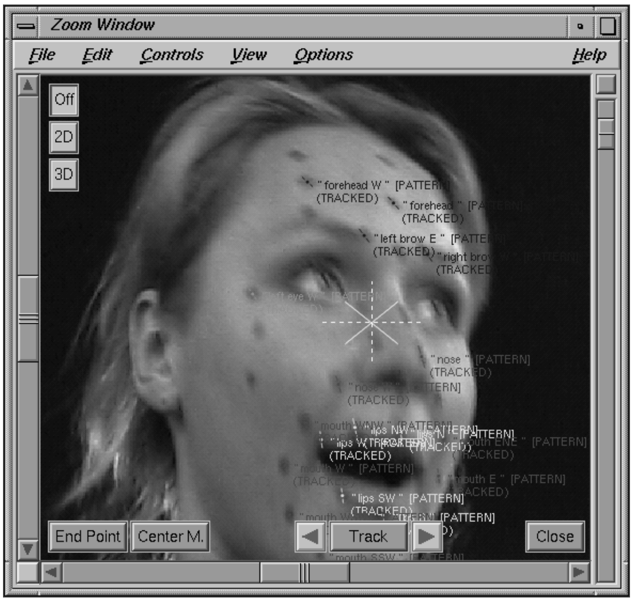
- *Select sequence "jo_front" and go to frame 1. Make sure point "01" is selected and track the actress' forehead marker until the end of the sequence.*

Now, there are two separate tracking curves associated to point object "01". One of them is tracked in sequence "jo_left", and the other one is tracked in sequence "jo_front".

- *Select sequence "jo_right" and go to frame 1. Make sure point "01" is selected and track the actress' forehead marker one more time until the end of the sequence is reached.*

Point “01” is now tracked in all three sequences (see also tutorial “Multiple Sequences”).

> **Important note:** Sometimes it may happen that a point gets tracked on the “wrong” marker in a certain sequence. If this should happen to you, use the function Zoom Window::Edit::Swap Curves to exchange tracking curves between two point objects.


Zoom Window
File
Edit
Controls
View
Options
Help
Off
2D
3D
" forehead W " [PATTERN] (TRACKED)
" forehead " [PATTERN] (TRACKED)
" left brow E " [PATTERN] (TRACKED)
" right brow " [PATTERN] (TRACKED)
" nose " [PATTERN] (TRACKED)
" lips SW " [PATTERN] (TRACKED)
End Point
Center M.
Track
Close

Mocap project containing 27 tracking points

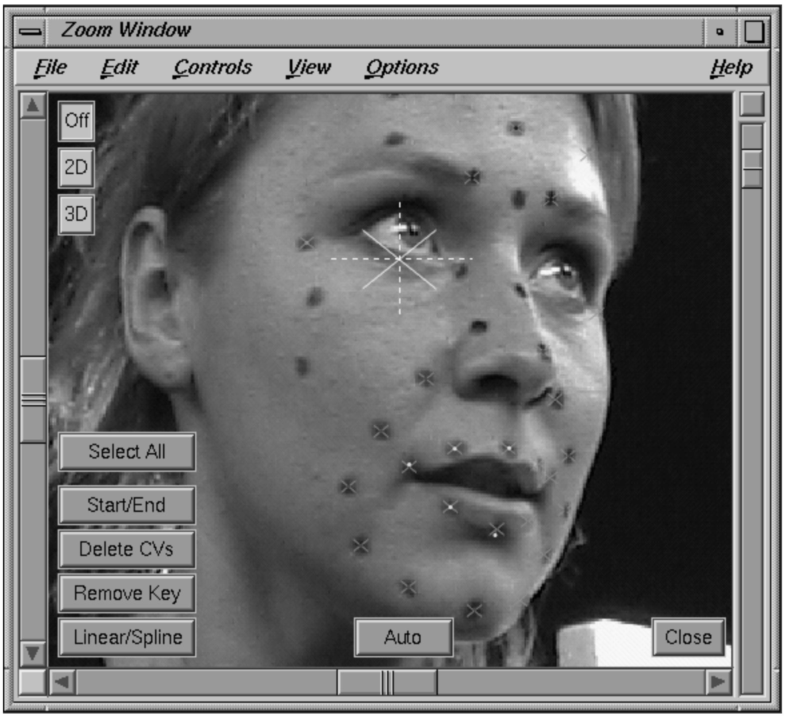
To complete tracking business, it would be necessary to create at least six points, tracked in the same sequences, otherwise 3D-Equalizer would not be able to calculate a solution for our mocap project. We will skip this part, because there is most likely nothing new to learn.

- *Please open the project “mocap2.3de” found in the directory “<tutorial install directory>/projects/”. After that, select the mocap pointgroup “pgroup #2”.*

As you see, there are 27 points in total in this project. Some points have been tracked in two sequences, whereas others have been tracked in all three ones (see figure above). The project is well prepared now, so we can start calculating.

- *Select the menu item Main Window::Calc::Calc All Objects From Scratch.*

3DEqualizer starts calculating. After a short period of time, this task is finished and we can examine the results.



Reconstructed mocap points

- *Please select the environment “Status” from the Main Window’s environments menu. Examine the average deviation value, displayed in the Status Window, of each sequence, by selecting them one after another.*

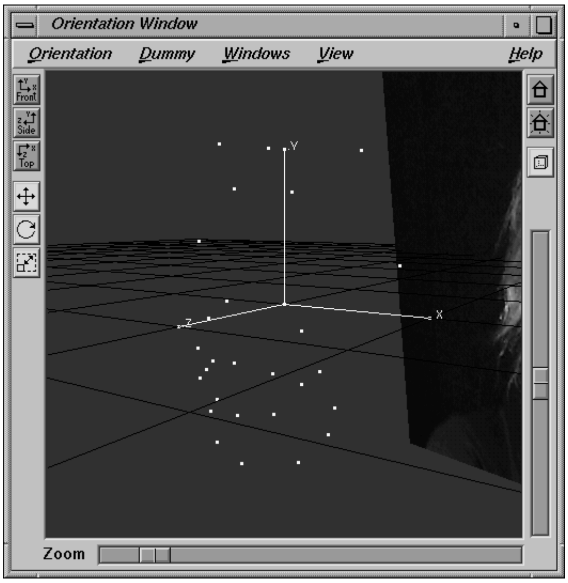
The average deviation values already look quite good. Sequence “jo_left” has an average deviation of 0.40 pixel, “jo_front” has an average deviation of 0.22 pixel and “jo_right” has an average deviation of 0.74 pixels.

- *Disable the menu toggle Zoom Window::View::Show Point Names, enable the menu toggle Zoom Window::View::Show Reconstructed Points::Distortion, go through the sequence using Main Window’s frame slider and watch the Zoom Window’s display.*

The reconstructed three-dimensional point positions are visualized by green crosses. As you can easily see, the reconstructed points match the original tracking data quite well. In addition, they move independently from each other, as intended (see figure above).

- *Please select the environment "Orientation" from the Main Window's environments menu. Disable the menu toggle Orientation Window::View::Show Point Names, go through the sequence using the frame slider and watch the white 3D points, displayed in the Orientation Window.*

The white 3D points reflect the form of the actress' face quite well (see figure below). As we expected, they move individually.



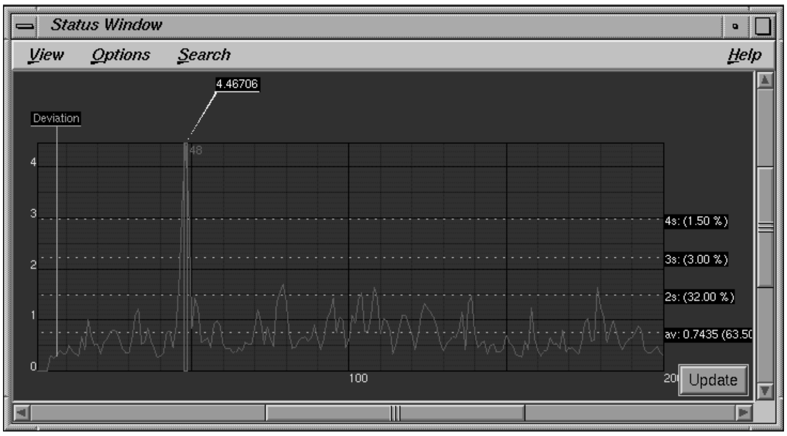
Reconstructed mocap points in 3D space

Compensating for subframe timeshift

In the following section we will discuss how to adjust timeshift between sequences on a subframe basis. By using this technique, it is possible to improve the reconstruction quality of a mocap project.

- *Please choose environment "Status" from the Main Window's environments menu, select sequence "jo_right" and examine the average deviation curve, displayed in 3DE's Status Window.*

You will notice that there is a significant peak in the average deviation curve around frame 48 (see figure below).



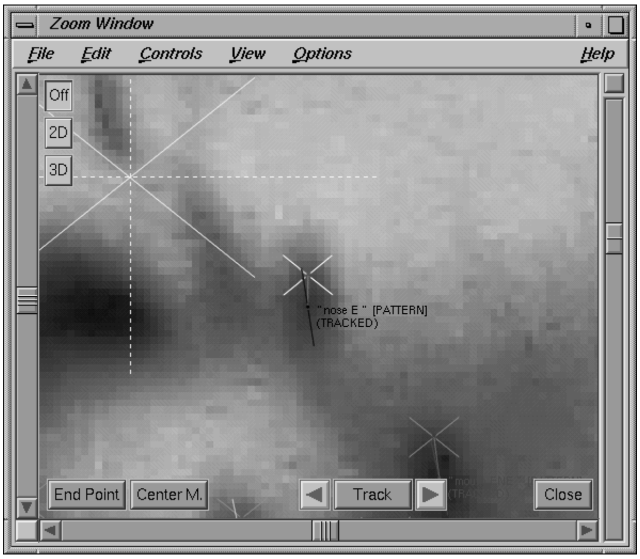
Peak in deviation curve caused by wrong subframe timeshift

There are actually two different reasons why the deviation is so high. If you look at frame 48 you will notice that the actress' head is quite motionblurred because she performs a rapid movement.

First of all, the motiontracking quality decreases for a few frames, caused by blurred markers. Secondly, there seems to be a subframe timeshift between the three sequences. That means, the sequences have been recorded by three video cameras, which were not fully synchronized to each other (e.g. by using a blackburst generator). Their offset in time is less than one frame, but it seems to be enough to falsify the result. Let's check this.

- *Go to frame 48, make sure toggle menu item Zoom Window::View::Show Reconstructed Points::Distortion is enabled and take a look at point "nose E".*

As you see, the reconstructed 3D point (green cross) is located several pixels above the actual tracking point, which tracked the marker more or less precisely (see figure below).



Mismatch between 2D and 3D point position caused by wrong subframe timeshift

You can enter the subframe timeshift of a sequence manually, if you know the respective number. However, most often this number will be unknown (as in our example). Fortunately, 3DEqualizer is able to compute the subframe timeshift value for you.

- *Click on the button Frames Window::Modify to open the requester Modify Sequence for object "jo_right". Enable the toggle button Motion Capturing::Determine Timeshift Automatically and click on the button Ok to confirm your edits.*

- *Double-click on the entry "jo_left" of the Frames Window to open the respective Modify Sequence requester. Again, enable the toggle button Motion Capturing::Determine Timeshift Automatically and click on the button Ok.*

Sequence "jo_front" will act like some sort of time reference to the entire project, so we will not enable its "Determine Timeshift Automatically" mode. The subframe timeshift of the other two sequences will be computed relatively to this one.

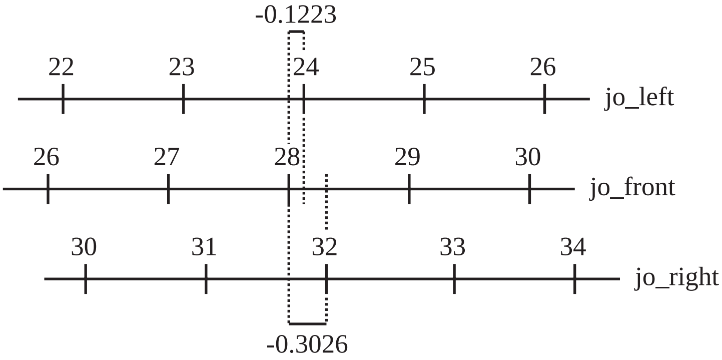
The general rule is that among all synchronized sequences there must be at least one with the Determine Timeshift Automatically toggle turned off and Timeshift parameter set to 0.0.

- *Select the menu item Main Window::Calc::Calc MoCap Timeshift.*

3DEqualizer starts calculating, which is indicated by a small requester. After some time, the calculation process is finished. Let's examine the result.

- *Examine the subframe timeshift value of sequence “jo_left” and “jo_right” displayed in the respective Modify Sequence requesters.*

You will notice that 3DEqualizer has computed a subframe timeshift of -0.1223 for sequence “jo_left” and a subframe timeshift of -0.3026 for sequence “jo_right” (see figure below). To see the full effect you have to recompute everything.



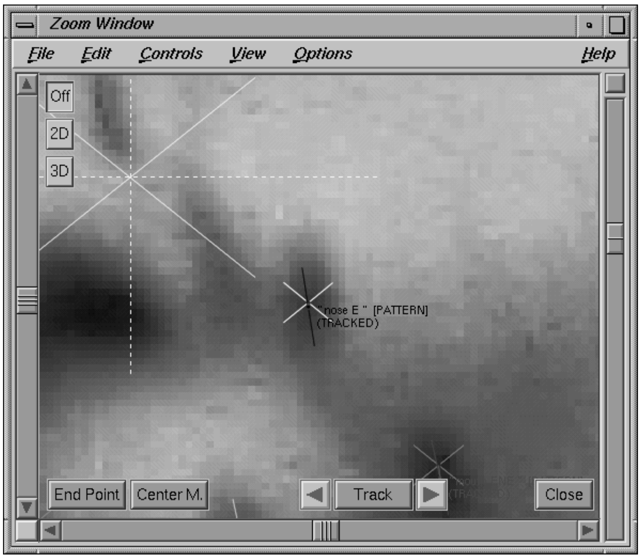
Schematic view of subframe timeshift

- *Select the menu item Main Window::Calc::Calc All Objects From Scratch.*

After 3DE has finished the calculation procedure, the entire result has been greatly improved. The Status Window indicates that the average deviation value of sequence “jo_right” has been improved from 0.74 to 0.40 pixel. Sequence “jo_front” has been improved from 0.22 to 0.16 pixel deviation, whereas sequence “jo_left” has been improved from 0.41 to 0.40 pixel deviation.

- *Select sequence “jo_right”, go to frame 48, make sure toggle menu item Zoom Window::View::Show Reconstructed Points::Distortion is enabled and take a look at point “nose E”.*

The mismatch between the reconstructed 3D point- and its tracking point position has been reduced significantly (see figure below).



Improved reconstructed 3D point position

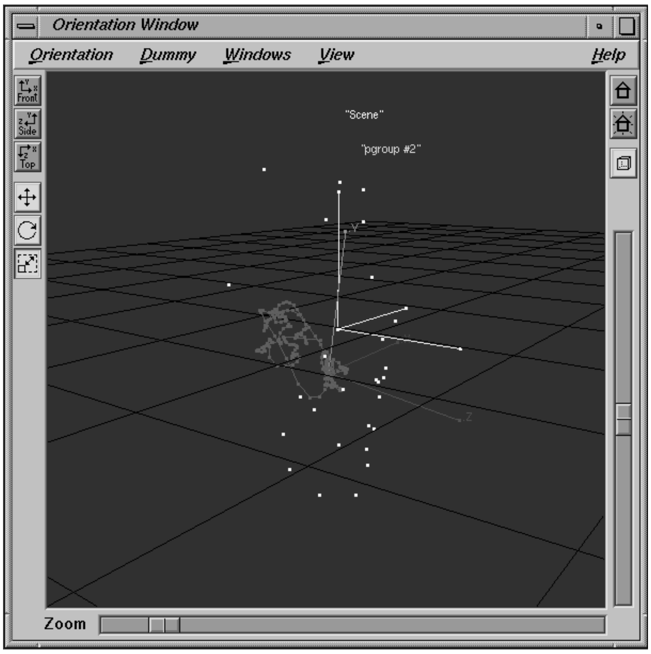
Extracting Overall Movement

In the following section we will learn a new technique to separate global motion, that all points have in common, from each point's specific motion. This feature can make it easier to use motion capturing data, produced by 3DE, in other 3D animation software packages.

- *Please select the environment "Orientation" from the Main Window's environments menu. Disable the menu toggle Orientation Window::View::Show Point Names, go through the sequence using the frame slider and examine the white 3D points, displayed in the Orientation Window.*

The white 3D points move in a quite realistic manner. The motion of the actress' head is created by the mimic of her face, as well as the movement of her head. Her head's movement, which is some kind of global motion, is added to the specific motion of every single point. It would be useful to separate this global motion from the points and instead associate it to the mocap pointgroup.

- *Double-click on the mocap pointgroup's entry "pgroup #2" to open the Modify Pointgroup requester. Set option menu Motion Captuing to Extract Overall Movement Automatically, and confirm your edits by pressing the button Ok.*



Global motion extracted into pointgoup

To see the impact of this modification, you'll have to compute a new solution.

- *Select the menu item Main Window::Calc::Calc All Objects From Scratch.*

After some time, 3DEqualizer finishes the calculation process and you can examine the result.

- *Enable the menu toggle Orientation Window::View::Show Pointgroup Path and take a look at the Orientation Window's display while going through the sequence.*

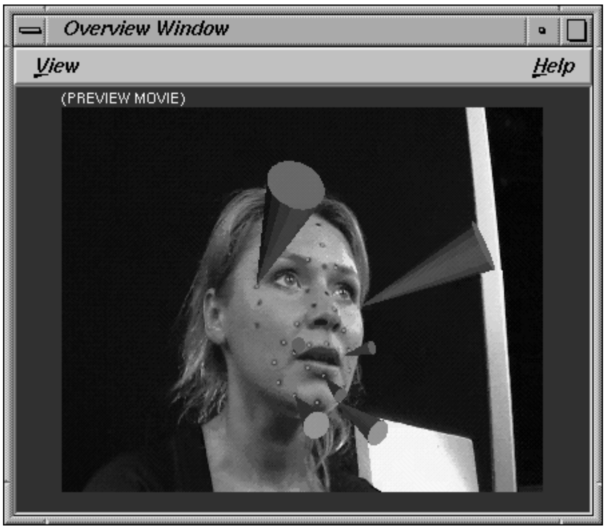
As you see, a red three-dimensional motion path indicates that the pointgroup object has now become animated (see figure above). In fact, its translation, as well as its rotation now change over time. However, the points have kept their individual motion.

> **Important Note:** To achieve the best possible quality for extracting global motion, use the setting "Extract Overall Movement Manually" and specify three points that do not move individually with respect to each other.

You have now reached the end of this tutorial. Let's take a look at the result.

- *Please open the project "mocap3.3de" located in the directory "<tutorial install directory>/projects/". After that, select the environment "Preview" from the Main Window's environments menu and press the button Main Window::Play.*

3DEqualizer starts generating a preview movie and finally plays it back in realtime. You see a few, individually moving dummy cones attached to the actress' head (see figure below).



Preview movie of a mocap project

Selected Topics

Tracking

A point which has been designated in three-dimensional space and which has been entered in the Pointgroups Window should be represented on the screen by screen points for as many frames as possible. This is usually done by setting the screen points for a sufficient number of frames (keyframes) and using tracking techniques in order to determine their position on the screen for the intermediate frames as well. In the current release 3D-Equalizer provides three methods for doing this, namely patterntracking, markertracking and cornertracking. The modes of operation for these methods are similar, but the application is different. Before pointing out the differences and the limits of applicability we shall discuss in detail the mode of operation.

Search pattern and search area

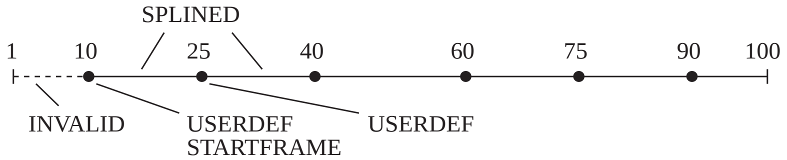
The basic principle of tracking a point can be described as follows. The user fixes the position of a screen point in a given frame, say frame j, by clicking the left mouse button in the Zoom Window. If the toggle Zoom Window::View::Show Tracking Area::All/Selected is activated, the search pattern (solid line) and the search area (dotted line) are displayed in the Zoom Window display. The areas can be modified as decribed in the chapter “Tutorial 1”.

The program tries to localize the search pattern specified in frame j by scanning the search area around the pre-determined point in frame j+1 with the search pattern. In the requester Modify Point... (opened by pressing Pointgroups Window::Modify...) the selector Trackingmode specifies whether marker-, pattern- or cornertracking is going to be performed. When the selector Colormode is set to Luminance, the search pattern and search area are transformed into black-and-white images before being matched. In general this mode is sufficient. Otherwise the selector can be set to Chrominance. In this case all three colour channels are used for matching the patterns, and the results are combined appropriately. In this mode tracking takes more time.

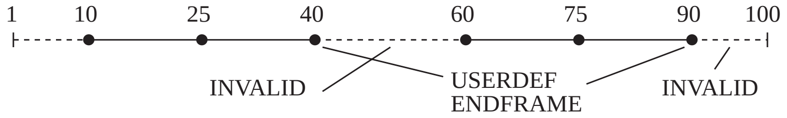
The tracking procedure can be supported by the functionality of the Image Control Window (opened by activating Main Window::Windows::Image Controls). The preconditions for markertracking are described in the section “Markertracking”.

Valid and invalid screen points

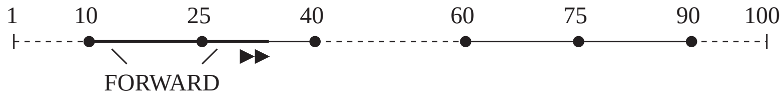
Assume you are given a sequence of 100 frames and you want to track a point of a moving object. For some frames the point may be covered by any obstacle, so that it cannot be tracked. The point may be visible from frames 10 to 40 and from 60 to 90, and invisible for frames 1 to 9, 41 to 59, 91 to 100, for what reason ever. Now you set keyframes for instance at frames 10, 25, 40, 60, 75, 90. The first keyframe of the point is automatically defined as startframe. The situation may be visualized by the following diagram:



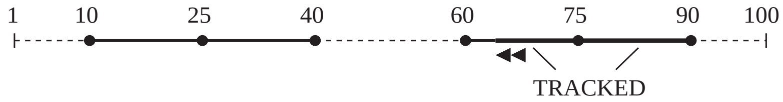
Since in the intermediate frames no information about the screen point is available, 3D-Equalizer uses some interpolation procedure in order to generate a screen point position, thus setting the screen point motion curve to a well-defined state. Since we do not have any information at all about frames 41 to 59 and 91 to 100, the screen points for these frames should be defined as invalid. This can be done by transforming the screen points of frames 40 and 90 into endpoints (see also End Point, Zoom Window). Frame 60 is now a startframe of the point, and the situation is now:



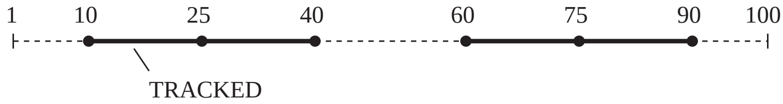
When the point is tracked, the setting of the parameter Direction (requester Create Point..., Modify Point...) is taken into account. We shall consider the case that it is set to Forward & Backward, since all other cases are logically contained therein. When the tracking procedure is started (see also Track, Zoom Window), 3D-Equalizer begins at frame 11, and an intermediate situation of the screen point key-states is:



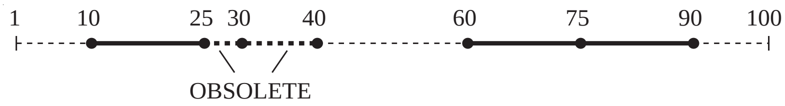
The screen points which are tracked carry now the key-status forward. Having tracked frame 89, the same procedure is done in the reverse direction:



The screen points which carried the key-status forward are now transformed into tracked. The results from tracking forward and tracking backward are combined appropriately. The final situation is:



It may occur that you wish to correct the screen point position for any frame by hand or insert another keyframe, because in the tracking procedure, the program has lost the track. If you insert a keyframe for instance at frame 30 the situation becomes:

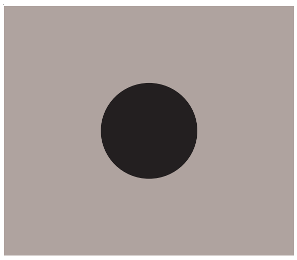


When the tracking procedure is started, 3D-Equalizer will handle frames 26 to 29 and 31 to 39 once more, running again into the final situation mentioned above. A similar behaviour occurs when a userdefined screen point is modified or removed.

Markertracking

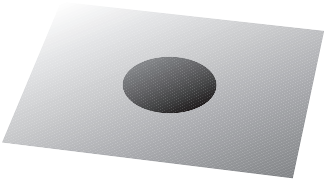
In the following, the peculiarities of the markertracking procedure are described. In order to do this, we have to clarify what we understand by a marker in the framework of 3D-Equalizer. Let us consider a surface with a spot on it. We call this spot a *marker* if

- the spot is unicoloured and point-symmetric,
- the surface is unicoloured and non-curved (i.e. flat).



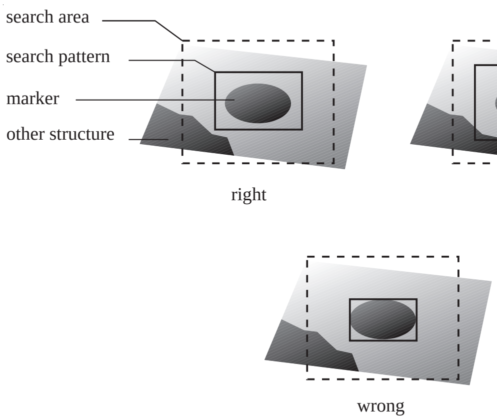
Example of a marker

The surface may be illuminated, so that it does not appear to be unicoloured or to have a uniform luminosity, as shown in the figure below. Nevertheless markertracking can be applied, since 3D-Equalizer is able to compensate non-uniformities of this type within certain limits.



A non-uniformly illuminated marker

When you specify the search pattern for tracking a marker, you should do this as shown in the following figure. The marker must be entirely contained in the search pattern and should not touch the search area borders.



Tracking a marker

Now the procedure is clear: 3D-Equalizer tries to recognize the search pattern of one frame within the search area of the frame currently being tracked. When the area (which contains the marker) is found, 3D-Equalizer calculates the center of the marker, thus completing the tracking procedure of this frame.

Applicability

As you have seen, markertracking is applied in very special situations, namely in presence of unicoloured markers on unicoloured surfaces, whereas patterntracking can be applied to each structured image. So, why should markertracking be used at all?

The advantage of markertracking is the greater precision in contrast to patterntracking. In a patterntracking procedure it can happen that the deviation of the screen point from its „true“ position becomes too big due to error accumulation.

The markertracking procedure is based on template matching of neighbouring frames as well. However, in each frame the center of the marker can be found without recourse to any of the previous frames. One could say markertracking is more stable as time elapses (if the conditions of markertracking are fulfilled, of course).

The tracking procedure may fail under the conditions given in the following. Patterntracking does not work at a satisfying precision under the following circumstances:

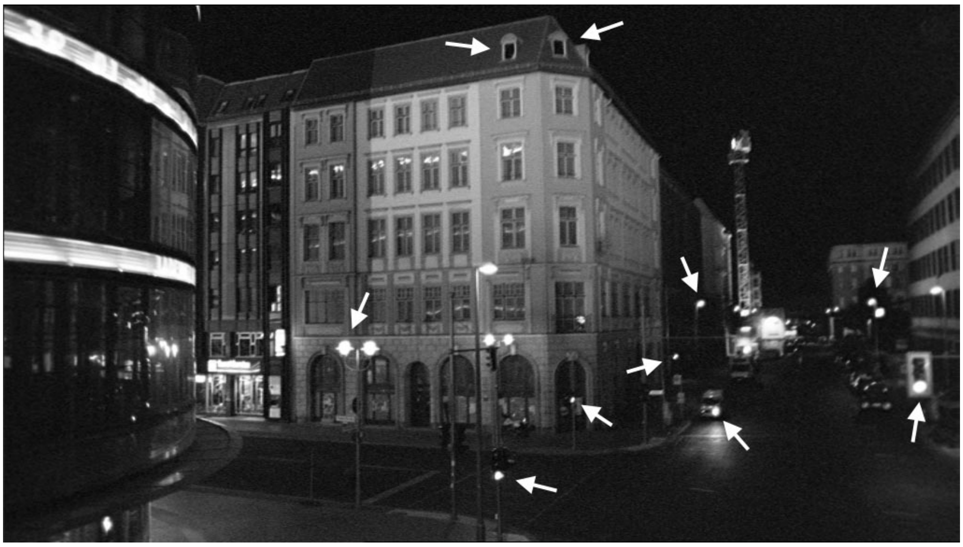
- The luminosity contrast of the search pattern is too weak.
- The image within the search pattern is subject to extreme deformations. This can occur, when the camera or the object at which the point is designated perform rapid rotations around the direction-of-view axis.

Markertracking fails to work at a satisfying precision if:

- the marker does not fulfil the conditions specified in the section “Markertracking”.
- if the luminosity contrast of the marker on the surface is too weak.

“Natural” markers

Often it is possible to track objects as markers which in reality do not fulfill the conditions of markers at all, e.g. street lights, traffic lights, the moon (infinite distance!), etc. In the figure below we have marked a few objects which in fact can be tracked as markers. For some of these objects it is necessary to use the image controls in order to remove disturbing influences.



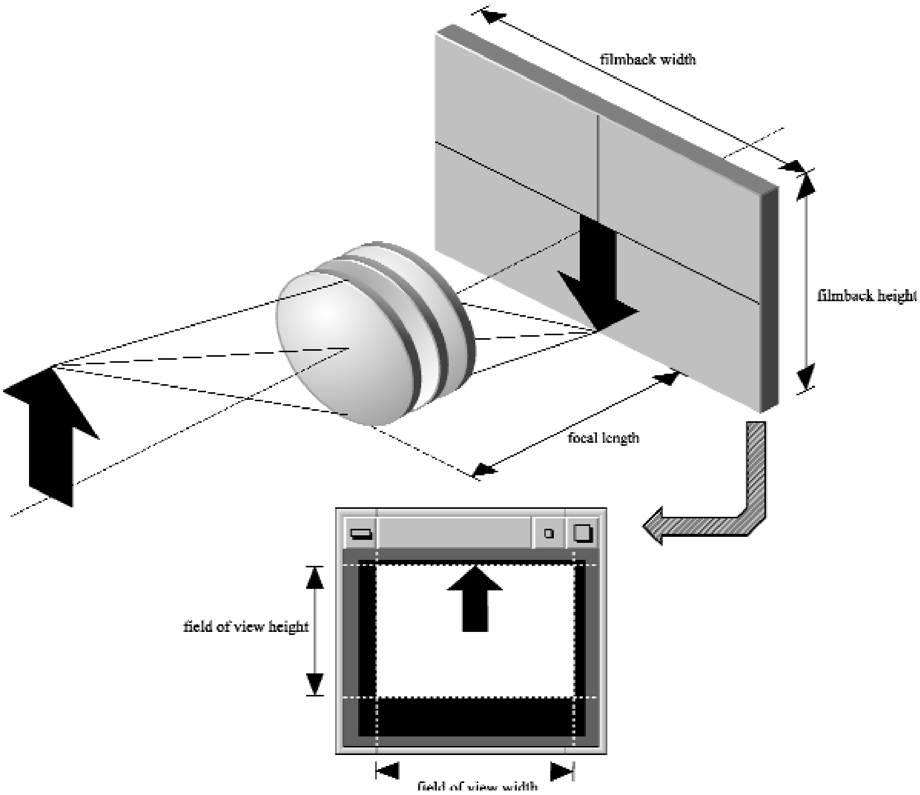
“Natural” markers

Camera Adjustment

In the following the terms *focal length, filmback width, filmback height, lens center offset, field of view width, field of view height* which arise in connection with 3D-Equalizer, camera optics and digitizing are illustrated.

Camera models

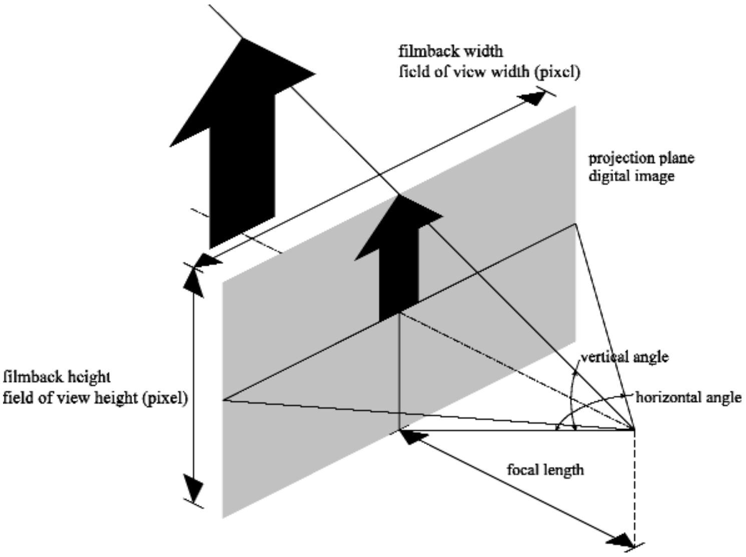
A simple model of a realistic camera is shown in the figure below. It essentially consists of a lens turret and the filmback.



Real camera geometry

In terms of the ray model of light, each point of the recorded object is the origin of a bundle of light rays, which pass the lens turret. If the rays meet in one point on the filmback the turret is well-focused with respect to the object. In the case of a film camera the filmback is a piece of celluloid. In the case of a modern video camera it might be a CCD chip. Abstractly speaking, we visualize the filmback as a rectangular shape in the focal plane of the camera, to which the recorded image is projected. It has a width and a height, measurable in common length units (not in pixel). We do not need to consider here the details of digitizing and data compression and storage.

The next point of interest is the output device (namely the display field of the 3D-Equalizer Overview Window) used for displaying the recorded images. We shall assume here, that the entire image has survived the process of digitizing, i.e. there is not cropping at the borders of the image. On the filmback one can associate a width and a height in length units to the image. The representation on the screen leads us immediately to the term field of view. The field of view is the rectangular area on the screen, which defines precisely the borders of the images. It is measured in pixel by pixel. The parameters filmback width and height are closely related to the parameters field of view width and height. The field of view, i.e. the image displayed in the Overview Window display must correspond to the image on the filmback. In other words, the field of view is the representation in pixel by pixel of the filmback in length unit by length unit.



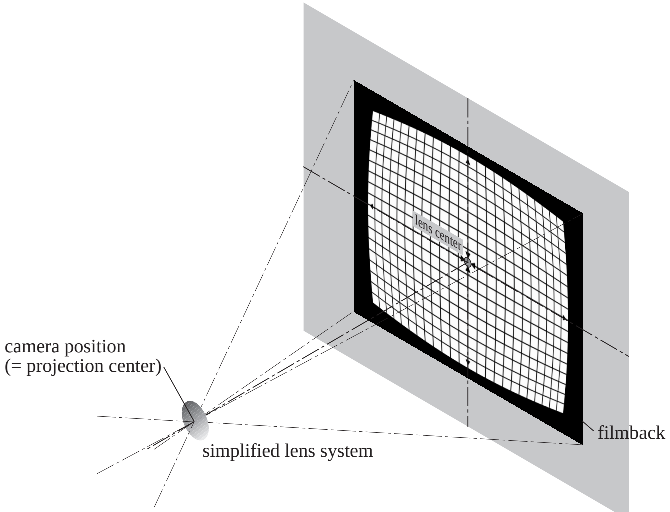
Filmback width/height vs. Field of view width/height

In connection with 3D-Equalizer's reconstruction procedures it is very important to maintain this one-to-one correspondence, and therefore the field of view should be specified with great care, as well as the filmback should be carefully specified by the user or calculated in 3D-Equalizer's automatic camera adjustment procedure.

The realistic camera can be replaced by a simple model as shown in the figure below, which contains the parameters filmback width, filmback height, focal length (the nonlinear distortion is taken into account as well). In the case of well-adjusted field of view and filmback you can think of the image plane as the filmback in length units on one hand and the field of view in pixel on the other hand. Thus, if you sit exactly in front of the computer monitor your eye corresponds precisely to the camera.

An important parameter arising in connection with the filmback and the field of view is called the pixel aspect. In the section "Examples of pixel aspect" you will find some details about this quantity.

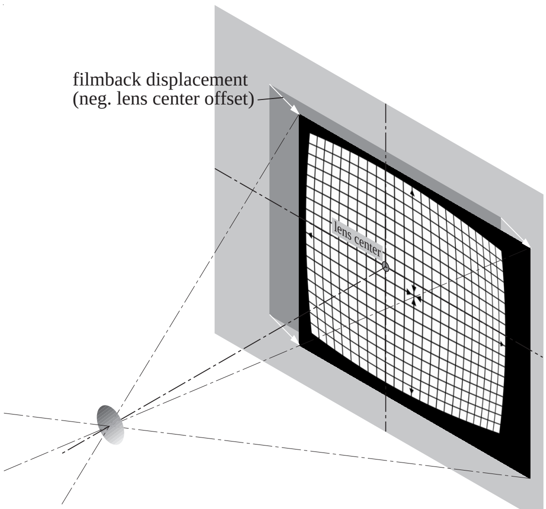
In contrast to version 2, version 3 allows to model camera geometries in which the lens center is displaced with respect to the filmback center. In the camera model we concidered up to now, the direction of view (which is equivalent to the symmetry axis of the lens system) cuts the filmback orthogonally in its center.



Simplified camera model

In terms of digitized images this means, the direction of view coincides with the center of the field of view. In some cameras however, the image projected on the filmback is bigger than the final images, and often the complete filmback is digitzed ("open gate"). In this case the lens center may be displaced, and the exact displacement can be obtained from the camera datasheets. The figure below, illustrates how the standard pinhole camera model of 3DE is modified in this situation.

The lens center plays a particular role in connection with the lens distortion parameters; the lens distortion model of 3DE allows to compensate symmetric, anamorphic distortions. The center of this symmetric deformation is the lens center.



Simplified camera model - lens center offset

Adjustment procedure

One of the most important points in camera tracking is to know the precise geometry of the camera. As you know, there are two different types of frames in 3D-Equalizer, namely sequance frames and reference frames. In the current version both groups of frames can either be recorded with the same camera or with different cameras. The Camera Adjustment Window provides the functionality to adjust both of them, if there is enough perspective information in the respective set of frames. The interaction components are the same for sequence and reference frames, and therefore in the following we shall consider the adjustment procedure for sequence frames in detail.

The Camera Adjustment Window provides two views on its content, namely the “simple view” with a very reduced set of parameters, and the “advanced view” with the full (and redundant) set of parameters. In the following we will consider this full set, so that you get an idea of how this window works.

In the “advanced view” of the Camera Adjustment Window, you see 14 parameters. The first six ones (filmback width, filmback height, focal length, film aspect, lens center offset x, lens center offset y) describe the standard camera geometry, the next two parameters (horizontal angle, vertical angle) represent the shape of the camera pyramid as angles. The pixel aspect (and the field of view) can be considered as digitizing parameters; they have nothing to do with the physical camera itself. The last group of parameters is dedicated to lens distortion. We have discussed them in detail in the tutorial “Lens Distortion”.

You may have noticed, that some of the parameter text fields are pink while others are grey. According to the guidelines used in the entire user interface of 3D-Equalizer you can only enter values in the pink fields. In the Camera Adjustment Window the colours are used to tell apart the camera parameters which determine the camera and those that are dependent of them.

The basic technique to find an appropriate camera is the following: try to find out as much information as possible about the camera and about the digitizing equipment. Set the selectors of the parameters that you do not know to UNKNOWN. The text fields turn into some ugly muddy pink. You may try to enter values in these fields. Since you signalled to 3DE that you do not know the corresponding parameter 3DE will immediately set this field to a value which the software developers found most reasonable.

The Camera Adjustment Window will only allow certain combinations of pink and grey fields, therefore it is not possible to get them all pink. This guarantees that the values in the fields always represent a consistent camera geometry, even if it does not yet necessarily represent the one you used for recording the scene.

Set the selectors right to the parameters that you know precisely to FIXED. You will notice that the corresponding text fields turns into some more friendly pink. You should enter the values in the text field.

Set the selectors of the parameters you are not quite sure about to FINE ADJUST and enter what you think might be the correct value.

By fixing the selectors you specified the parameters on which the camera is based. All remaining grey fields are dependent of the pink ones. For example you will never see that film aspect and pixel aspect are pink (or muddy pink) at the same time: specifying one of them automatically determines the value of the other one.

Here is another example: focal length, filmback height and vertical angle cannot be pink at the same time because each two of them determine the third one. The same is valid for focal length, filmback width and horizontal angle.

> **Important note:** You may be tempted to set as many selectors as possible to UNKNOWN. However, this increases the processing time dramatically. A better strategy is to think about the parameters and follow the above procedure.

The grey text field underneath the camera parameters displays information about what is going to happen when you press the button Adjust. You may read it or ignore it, as you like. If you want to know more about the parameters and their selectors you can try out a few settings of the selectors and see what message appears in the text field.

Let us consider the following example. Set the selector Horizontal Angle to „FINE ADJUST“, the selector Vertical Angle to „UNKNOWN“, the selector Focal Length to „FINE ADJUST“ and the selector Lens Distortion to „UNKNOWN“. Then in the text field the following messages appear:

> *Calculation level „Unknown“ (large search interval):*
> *filmback width   vertical angle   lens distortion*
>
> *focal length will be fixed by definition.*
>
> *Calculation level „Fine Adjust“ (small search interval):*
> *filmback width   vertical angle   lens distortion*

When you press the adjust button, filmback width and vertical angle will be determined roughly, thereafter the lens distortion will be calculated roughly as well. Afterwards the focal length will be fixed by definition (10mm), and finally filmback width, vertical angle and lens distortion will be determined as precise as possible.

The actual adjustment procedure can be described as follows: First you have to decide how precise you want to determine the camera. There are two parameters which determine the level of precision, namely the precision slider in the Camera Adjustment Window and the parameter “Point Finetuning“ in the requester “Edit Global Calculation Settings...“ (select Main Window::Options::Calculation Settings). In order to understand the parameter “Point Finetuning“ imagine the following situation: your sequence consists of many frames, and many points come in and out of the screen. While 3DE calculates the positions of the points it performs some kind of „over all finetuning“ from time to time. The “Point Finetuning“ specifies how many of these finetuning cycles are performed during the procedure. Therefore, if your sequence looks as described above, the number of point finetuning cycles should be increased. For sequences with a minimum of points (six) the value 5 is a quite good default value.

The meaning of the precision slider is quite easy to understand: it simply determines how many frames are taken into account for adjusting the camera. When it is set to the right most position all frames are taken into account. When it is set to the left 3DE chooses a set of frames for the adjustment. As soon as your production comes to an end you should choose the highest precision and perform a fine adjustment.

When you press Adjust a requester that displays the current status of the calculation appears. The form of the message depends on the „calculation level“, whether the system is trying to determine parameters with “FINE ADJUST“ or “UNKNOWN“ setting. As you can see in the grey text field the “UNKNOWN“ parameters are always processed before the “FINE ADJUST“ parameters.

Calculation level “Unknown“

Let us consider the case that one or more selectors are set to “UNKNOWN“. The requester contains the text:

Attempt: x/y

Deviation a.aaaa (Best: b.bbbb)

That means, 3DE has done its x-th attempt out of a total number of y attempts in order to determine the parameters roughly. The result was a pixel deviation of a.aaaa pixel and the best value up to now was b.bbbb pixel. A calculation of two or three unknown parameters can take a long time. However, after 30-50 % of the total number of attempts, 3DE will usually have found a quite acceptable result. If this is the case you can decide if you want the system to go on searching or if you terminate this calculation level and go on with the fine adjustment.

Calculation level “Fine Adjust“

The message for the case of fine adjustment is similar:

Attempt: x

Deviation a.aaaa (Best: b.bbbb)

Again, x indicates howmany attempts have been done. However the overall number of attempts is not yet known to the system. There is a maximum number of attempts in order to keep the processing time finite, but when the system reaches the limit of calculation precision earlier the procedure may also terminate. As soon as you notice that the pixel deviation only changes slightly (e.g. 0.001 or less) you may terminate the procedure by hand. As general rules for the processing time simply keep in mind:

- Three parameters take more time than two parameters take more time than one parameter,
- “UNKNOWN“ parameters take more time than „FINE ADJUST“ parameters,

Examples of pixel aspect

In many cases in which you do not know the pixel aspect there are ways to find it out by „guessing“.

Let us assume your images have a size of 720x576 pixel and come from a PAL/SECAM video camera. In this case your pixel aspect is usually 1.06667. This is the case when you have done the digitizing e.g. with Flint/Flame/Inferno. Or maybe the images have a size of 768x576 pixel. This might be a PAL/SECAM image with a pixel aspect of 1.0. As a third example we consider the case of a size of 720x486 and you are working on NTSC images. Then the pixel aspect will probably be 0.9. Finally, if your images come from a film scanner, the pixel aspect will often be 1.0, although other pixel aspects may occur as well (e.g. Quantel products). If you use the built-in hardware of the SGI O2 in order to digitze PAL images, the pixel aspect is 1.0926.

Again, if you are not quite sure, try to adjust the camera by setting the selector to “FINE ADJUST“ and press adjust.

If you do not have any idea about the pixel aspect, maybe you know the film aspect. Most video cameras have a “film aspect” of 4/3 = 1.333333. HDTV and PAL PLUS images have a film aspect of 16/9 = 1.777777.

Many cameras for taking photographs have a film aspect of 3/2 = 1.5. This could be an important information if you are using digitized photographs as reference frames. When you enter the film aspect the pixel aspect is a <dependent> variable and therefore calculated by 3D-Equalizer.

Adjustment of the reference camera

In practice it is useful to take reference frames with a camera different from the one used for recording the sequence frames. This might be the case when the sequence camera is unwieldy or part of a bigger mechanical installation, so that it would be difficult to move it to another point of view.

Photographs as reference frames

The reference frames can be provided e.g. by means of high resolution digitized photographs. In this case usually you will know many of the parameters of the camera.

The rules for estimating parameters of the sequence camera can be applied to the reference camera as well. The reference camera can be adjusted by 3DE as well as the sequence camera described above. Since usually there are much less reference frames than sequence frames there are limits which and how many of the parameters can be calculated. The general rules are

- The more reference frames are used, the more precise the reference camera and the points can be calculated by 3DE.
- The more common points are contained in the reference frames, the more precise the reference camera and the points can be calculated by 3DE.

Important note: This should, however, not lead to gigantic projects with hundreds of reference frames and thousands of points! A bettter strategy is to add reference frames and points step by step, and to do the 3D calculations from time to time.

Minimum requirements

To use the concept of reference frames you will need at least two of them containing at least six common points. When these minimum requirements are fulfilled the points can be calculated. If you want 3DE to calculate the parameters of the reference camera you should take into account that it is almost impossible to calculate more than one parameter if only the above minimum requirements are fulfilled.

Generally it is problematical (or even impossible) to let 3DE calculate the focal length for the minimum requirements, although we have examples for which this works as well. We have done tests with one of the tutorial sequences. 3DE was able to calculate the following parameters (in case the others are fixed and correct):

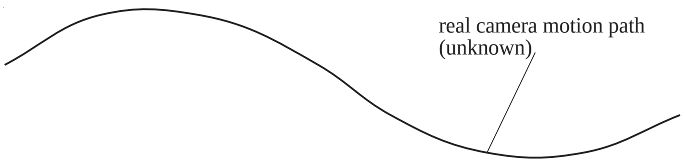
- horizontal angle
- vertical angle (very unprecise)

Important note: If you are using different cameras for the reference and sequence frames you should determine the reference camera first and the sequence camera afterwards.

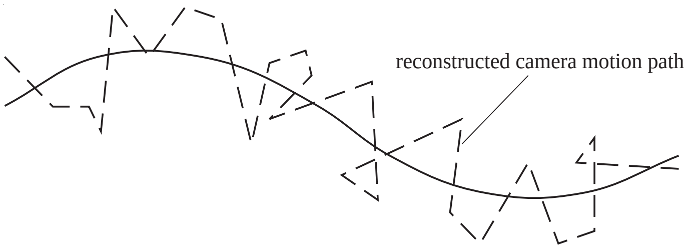
Selected Topics

The Concept of Postfiltering

The purpose of this section is to outline some aspects of 3DE's postfilter functionality. Let us consider the following situation: assume you are recording a real sequence. The camera motion (position) path will be a more or less smooth curve in three-dimensional space. We visualize this path by a solid line as follows:

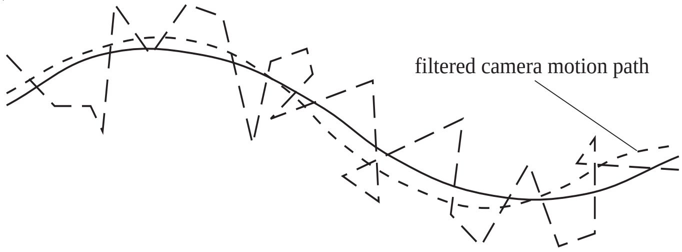


When 3DE reconstructs this motion path by evaluating the tracking curves of the points, the result could look like this:

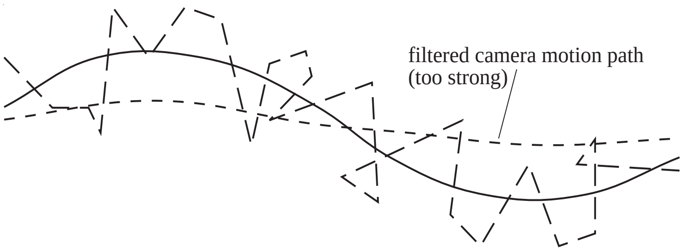


The magnitude of the fluctuations of the reconstructed path depends on the quality of the tracking curves and on the relative alignment of the points in three-dimensional space with respect to the camera. In the tutorial "Stabilizing" you may have seen this dependency in detail. Now the idea of postfiltering is the following: In the reconstruction procedure the information of "smoothness" of the real camera motion path obviously gets lost. The task of a postfilter is to restore this information without falsifying the reconstructed motion path too much. 3D-Equalizer does this by means of a Fourier analysis and synthesis: the camera motion path is considered as a superposition of periodic movements of many different frequencies. When the motion path is jittered, there are a lot of high frequency movements which contribute to the path in an uncontrollable manner. Fourier analysis means to find out the strength (amplitude) of each frequency. Filtering by means of Fourier synthesis means to recombine many of the frequencies, but to leave out the high, disturbing ones. The more higher frequencies are left out, the smoother becomes the filtered motion path. The highest frequency to be taken into account is called the "cut-off"-frequency (because all higher ones are cut off). In the requester Edit Postfilter Settings... the cut-off frequency is represented by the sliders Strength Position/Orientation/Zoom.

The following figure shows an example of a good filtered motion path. However, in practice one cannot know whether the real motion path and the filtered motion path are so close together. This fact represents the difficulty of finding the optimal value for the slider Strength.



The next figure shows a filtered motion path that could have been created with a value for the slider Strength which was too high. The filtered motion path is smooth, but its coincidence with the real camera motion path is even worse than the one of the reconstructed motion path.



Editing and Consistency

In this section we explain the expression “consistency” which occurs in some situations in the workflow of 3DE. In the introduction of the manual we presented the typical workflow of a 3DE project. The last (optional) step in this workflow before exporting the data to the animation package is to edit, if necessary, the motion curves of the camera (or the object). In general, this should not be necessary, because each error in the motion path should originate from errors of the motiontracking or from unprecise camera parameters. However, in practice it turns out that even if neither the motiontracking nor the camera parameters can be improved, the motion path exhibits inaccuracies which the user wishes to eliminate. This can be done in 3DE’s Motion Path Window.

The menus of this window are explained in the reference part of the manual. The menu “Options” essentially contains toggle buttons for turning on and off modes which reflect the functions of the menu “Calc”, so that we can now concentrate on the menu functions Calc::Make Rotation Curves Consistent and Calc::Make Position Curves Consistent.

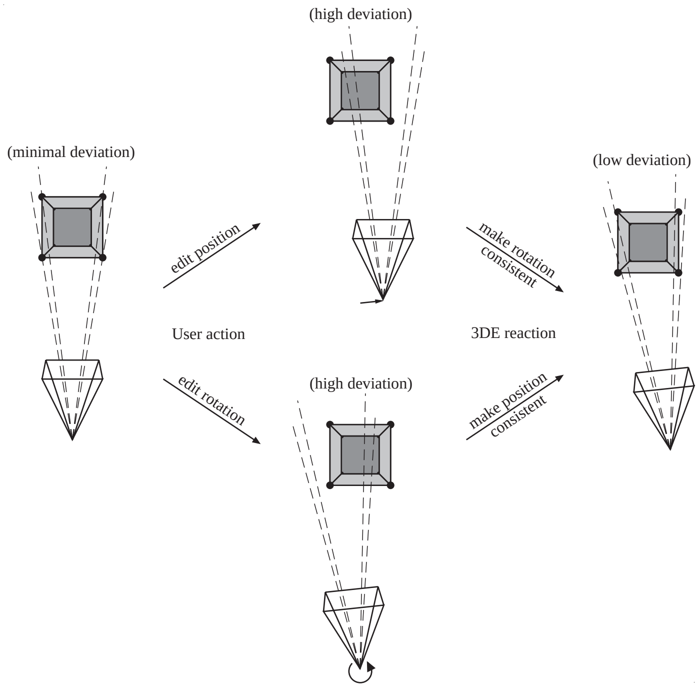
Let us assume you have tracked a sufficient number of points and the camera parameters are correct. Nevertheless, there is (are) one (or more) frame(s) in which the camera position and rotation looks inaccurate, and you decide to correct this manually before exporting. The principle 3DE uses in order to calculate the camera position and rotation is to minimize (essentially) the distances between the rays spanned by the camera position and the tracked point, and the point in three-dimensional space. These distances are closely related to the deviation displayed in the Status Window.

One can expect that any manual correction will increase the deviation value, and the task of 3DE in this case is to reduce the “damage” (sorry) which results from editing. In the figure below you can see the two different situations which can occur as a consequence.

When you modify the position of the camera, the rays totally miss the previously calculated points. If you generate a preview movie now, you will probably see a slight jump of the dummy objects. The strong coherence between position and rotation is destroyed. When you press now Calc::Make Rotation Curves Consistent, 3DE compensates the translation of the camera by an appropriate rotation, again trying to minimize the deviation with the constraint of a fixed, edited position. The opposite situation is analogous: you edit the rotation of the camera, press Calc::Make Position Curves Consistent, and 3DE compensates with an appropriate translation.

You can check the deviation in the Status Window (press ‘update’). After editing, the deviation for the current frame increases. After making the curves consistent it decreases, but not as low as it was before editing, because 3DE had to take into account the edited quantity.

Between these two cases, editing the position and compensating by a rotation will probably happen more often, since the position usually exhibits a much bigger error than the rotation.



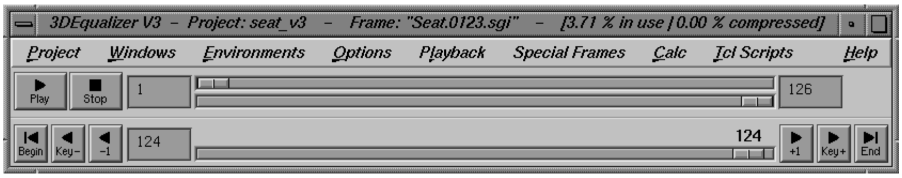
Edit <-> Make Consistent

Reference Manual

The purpose of this chapter is to give a complete description of 3D-Equalizer's features, listed window by window. A menu is opened by pressing ALT-shortcut. A menu item is selected by pressing the shortcut without the ALT-key.

Main Window

The Main Window provides functions for handling sequences, reference frames, the playback features, and it also hosts all major menus from which 3D-Equalizer's main functions can be controlled.



Main Window

Miscellaneous

Window title status line

Contains the name of the currently loaded project, the filename of the current frame and the amount of occupied imagebuffers.

Begin

Jumps to the first sequence/playback or reference frame.

End

Jumps to the last sequence/playback or reference frame.

Key -

Jumps to the previous (sequence) keyframe of the currently selected point.

Key +

Jumps to the next (sequence) keyframe of the currently selected point.

Frame -1, Frame +1

Steps back/forth by one frame in the currently selected sequence or in the list of reference frames.
see also: Frames Window

Frame slider

Used for moving within the currently selected sequence. While moving the slider, the screen points are displayed in the Overview and Zoom Window. The number of the current frame is displayed above the knob. It always ranges from 1 to number of frames, regardless of the numeration of the sequence frame filenames. In the requester Frames Window::Modify Sequence, a sequence may be specified by Start = 50, End = 250, Step = 2, corresponding to the filenames myimage.0050, ..., myimage.0250. In this case there are 101 frames, and frame nr. 1 corresponds to frame filename myimage.0050, and frame nr. 101 corresponds to myimage.0250.
see also: Frames Window

Frame textfield

Displays the current frame number.

Startframe slider

Specifies the first frame of the playback framerange.

Startframe textfield

Displays the first frame of the playback framerange.

Endframe slider

Specifies the last frame of the playback framerange.

Endframe textfield

Displays the last frame of the playback framerange.

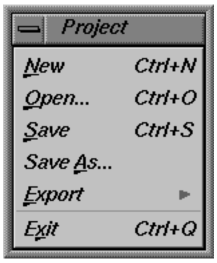
Play

Starts the playback mode. According to the current settings of the Playback menu toggle buttons, the preview movie or the frames of the currently selected sequence are played back. While in playback mode, the current playback speed is displayed in the title bar of the Main Window. This performance output is updated every 15 frames.
see also: Main Window::Playback

Stop

Stops the playback mode.

Menu: Project



Project handling and export functionality.
shortcut: ALT-p

New

Sets up a new project.
shortcuts: n, CTRL-n

Open

Opens a standard file requester Open New Project. The default directory can be specified in Options::3DE Preferences::Projects.
shortcuts: o, CTRL-o

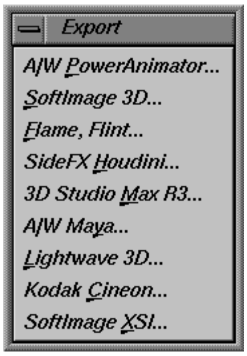
Save

Saves the current project. The default directory can be specified in Options::3DE Preferences::Projects.
shortcuts: s, CTRL-s

Save As

Opens a standard file requester Save Current Project As. In case of existing preview or cache movie(s), the system asks the user if the movies should be copied as well. The default directory can be specified in Options::3DE Preferences::Projects.
shortcut: a

Export



Several export functions.
shortcut: e

A|W PowerAnimator...

Opens a requester Export A|W PowerAnimator...
shortcut: p

SoftImage 3D..

Opens a requester Export SoftImage 3D...
shortcut: s

Flame, Flint...

Opens a requester Export Discreet Logic Flame, Flint...
shortcut: f

SideFX Houdini...

Opens a requester Export SideFX Houdini...
shortcut: h

3D Studio Max R3...

Opens a requester Export Kinetix 3D Studio Max R3...
shortcut: m

A|W Maya...

Opens a requester Export A|W Maya...
shortcut: y

Lightwave 3D...

Opens a requester Export NewTek Lightwave 3D...
shortcut: l

Kodak Cineon...

Opens a requester Export Kodak Cineon...
shortcut: c

SoftImage XSI...

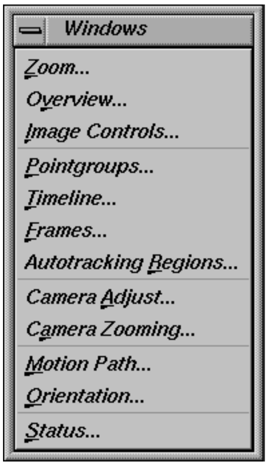
Opens a requester Export SoftImage XSI...
shortcut: c

Exit

Leave the program.
shortcuts: x, CTRL-q

Menu: Windows

Provides functions for opening/raising several windows.
shortcut: ALT-w



Zoom...

Opens the Zoom Window.
shortcut: z

Overview...

Opens the Overview Window.
shortcut: v

Image Controls...

Opens the Image Control Window.
shortcut: i

Pointgroups...

Opens the Points Window.
shortcut: p

Timeline...

Opens the Timeline Window.
shortcut: t

Frames...

Opens the Frames Window.
shortcut: f

Autotracking Regions...

Opens the Autotracking Regions Window.
shortcut: r

Camera Adjust...

Opens the Camera Adjustment Window.
shortcut: a

Camera Zooming...

Opens the Camera Zooming Window.
shortcut: a

Motion Path...

Opens the Motion Path Window.
shortcut: m

Orientation...

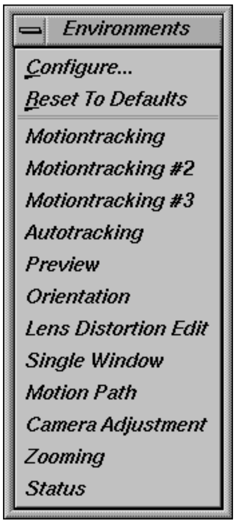
Opens the Orientation Window.
shortcut: o

Status...

Opens the Status Window.
shortcut: s

Menu: Environments

Functions to create, delete, modify and display window configurations, the so-called environments. Each menu entry located underneath the double line represents a certain environment. Selecting one of these menu entries causes 3D-Equalizer to display the respective environment. Displaying an environment means closing all windows first, then opening and positioning the windows according to the configuration of the respective environment. Furthermore, some toggle button settings of the windows are stored in the environments as well. The manual window placement feature of the desktop should be disabled when making use of the Environments feature.
shortcut: ALT-e
see also: all windows



Configure...

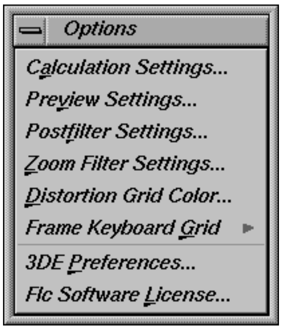
Opens the requester Configure Environments.
shortcut: c

Reset To Defaults

All existing environments of the current user account are deleted and replaced by the set of original environments, that are shipped with 3D-Equalizer.
shortcut: r

Menu: Options

Settings for calculation, preview, postfilter, keyboard, 3D-Equalizer preferences and software license.
shortcut: ALT-o



Project Settings...

Opens a requester Edit Global Calculation Settings...
shortcut: a

Preview Settings

Opens a requester Edit Preview Movie Settings...
shortcut: v

Postfilter Settings

Opens a requester Edit Postfilter Settings...
shortcut: f

Zoom Filter Settings...

Opens a requester Edit Zoom Curve Filter Settings...
shortcut: z

Distortion Grid Color...

Opens a requester Edit Distortion Grid Color...
shortcut: d

Frame Keyboard Grid

Choose among off, 2, 3, 4, 5, 10, 15, 20, 25 or Custom. The frame keyboard grid is taken into account when the CURSOR-keys are pressed for jumping back and forth within the sequence.
shortcut: g

3DE Preferences

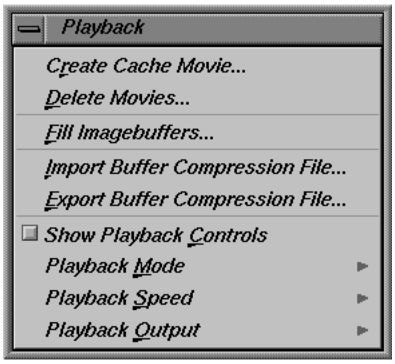
Opens a requester Edit 3DEqualizer Preferences...
shortcut: p

Software License

Opens a requester 3DEqualizer Software License...
shortcut: l

Menu: Playback

Functions to configure 3D-Equalizer’s playback features as well as functions to manage preview and cache movies.
shortcut: ALT-l



Create Cache Movie...

This function creates the cache movie of the currently selected sequence/reference frame only, or it creates all cache movies of all sequences and reference frames. Cache movie frames which have already been created earlier are not updated. The user is able to interrupt the creation process at any time. In order to operate, the Overview Window must be opened at a proper size, so that the cache movie can be displayed.
shortcut: r
see also: Edit Preview Movie Settings

Delete Movies...

If the preview and/or the cache movies are no longer needed, they can be deleted by the use of this function. A requester is displayed, asking for the movies to be deleted. If the user confirms this, the cache (respectively the preview) movies of all available sequences and reference frames are deleted.
shortcut: d

Fill Imagebuffers...

3D-Equalizer’s image buffers are filled with images of the currently selected sequence. This function is very useful on systems with much RAM, while working on video resolution material. Often it is possible to load the entire image sequence into memory, which speeds up the tracking process and general workflow tremendously. A requester asks the user whether the currently occupied imagebuffers should be freed or not. After an option has been chosen, the images of the currently selected sequence are loaded into the image buffers, starting at frame 1. If there are enough free image buffers, no formerly occupied imagebuffers are freed at all. The user is able to interrupt the process at any time.
shortcut: f
see also: Edit 3DEqualizer Preferences

Import Buffer Compression File...

A file, containing compressed image data, associated to the currently selected sequence is imported into 3D-Equalizer's imagebuffers.
shortcut: i
see also: Edit 3DEqualizer Preferences

Export Buffer Compression File...

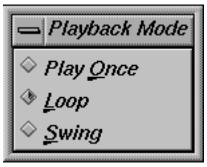
The images of the currently selected sequence are compressed and written into a single file. This file is either stored in the project directory or in the directory where the sequence's image files are located.
shortcut: e
see also: Edit 3DEqualizer Preferences

Show Playback Controls

If this toggle button is enabled, the playback controls (Play button, Stop button, Startframe/Endframe slider & textfield) are displayed. If this toggle is disabled, the playback controls are hidden by reducing the size of the Main Window. The setting of this toggle button is stored in the environments.
shortcut: c

Playback Mode

Contains toggle buttons for selecting the current playback mode.
shortcut: m



Play Once

If activated, a regular forward playback is perfomed once within the currently defined playback range limits.
shortcut: o

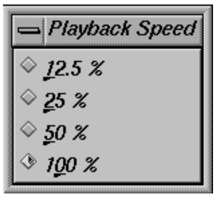
Loop

If activated, the program performs a playback again and again forward in time within the limits given by the playback range sliders, until it is interrupted.
shortcut: l

Swing

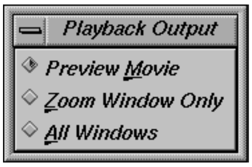
If activated, 3D-Equalizer performs a playback back and forth within the playback range limits until it is interrupted.
shortcut: s

Playback Speed



Contains toggle buttons to configure the current playback speed. If set to 100%, 3D-Equalizer tries to perform a playback of exactly the same framerate that is defined in the Rate Fps textfield of the currently selected sequence's requester in the Frames Window. If set to a lower percentage value, the playback speed is reduced accordingly.
shortcut: s
see also: Frames Window

Playback Output



Contains toggle buttons to define the output medium.
shortcut: o

Preview Movie

If activated, the preview movie of the currently selected sequence is played back while in playback mode.
shortcut: m

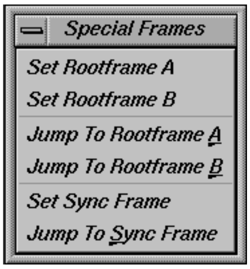
Zoom Window Only

If activated, the currently selected sequence is played back in the Zoom Window only while in playback mode. For this playback feature, the regular imagebuffers are used. That means, this playback performs at the best possible display quality because the original image material is used. While in playback mode, the user is able to use all functions of 3D-Equalizer, such as scaling and translating the zoom area, enabling the autocenter feature or selecting different points. Thus, this function enables the user to check the motion-tracking quality of the points very easily.
shortcut: z

All Windows

If activated, the currently selected sequence is played back simultaneously in the Zoom Window, the Overview Window and the Orientation Window while in playback mode.
shortcut: a

Menu: Special Frames



Provides functions to define and jump to a certain rootframe, as well as to define and jump to the sequence's syncframe.
shortcut: ALT-r

Set Rootframe A/B

Each pointgroup has its "own" set of two special frames, that are used in the early calculation phase of 3D-Equalizer. These frames are called rootframes. It is possible to let 3D-Equalizer define the rootframes automatically, which is the default behaviour. However, there are certain situations in which it might be necessary to define the rootframes manually. If the automatic rootframe determination is switched off in the respective requester of the Pointgroups Window, the functions Set Rootframe A/B can be used. Executing one of these functions causes 3D-Equalizer to declare the currently selected frame of the currently selected sequence or reference frame as a rootframe. If the currently selected frame is a rootframe, the label "ROOTFRAME" is displayed in the window title of the Main Window as well as within the image area of the Overview Window. Furthermore, rootframes are marked with a green bar in the Timeline Window.
see also: Pointgroups Window, Overview Window, Timeline Window

Jump to Rootframe A/B

Jumps to rootframe A (or B) of the currently selected pointgroup.
shortcut: a, b

Set Sync Frame

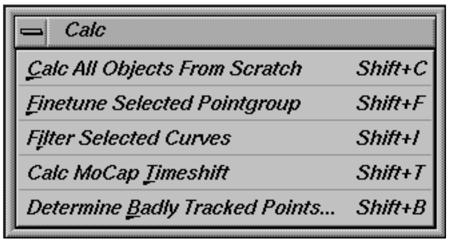
Mocap projects usually consist of multiple sequences which are synchronized in respect to each other. For this purpose, a special frame, named syncframe, has to be defined for each sequence. The syncframes of all sequences should show the content in exactly the same moment in time (just taken from different points of view). Usually, a frame is chosen that shows some content which can be easily recognized in all other sequences (e.g. blink of an eye, flashlight, etc.). This function defines the currently selected frame to be the syncframe of the current (synchronized) sequence. In order to use this function, the toggle Modify Sequence::Motion Capturing::Sequence is Synchronized to other Sequences must be activated. If the currently selected frame is a syncframe, the label "SYNCFRAME" is displayed in the window title of the Main Window as well as within the image area of the Overview Window. Furthermore, syncframes are marked with a yellow bar in the Timeline Window.
see also: Overview Window, Timeline Window

Jump To Sync Frame

Jumps to the syncframe of the currently selected sequence.
shortcut: s

Menu: Calc

Provides various calculation functions.
shortcut: ALT-c


Calc
Calc All Objects From Scratch Shift+C
Finetune Selected Pointgroup Shift+F
Filter Selected Curves Shift+I
Calc MoCap Timeshift Shift+T
Determine Badly Tracked Points... Shift+B

Calc All Objects From Scratch

Start the main calculation procedure of 3D-Equalizer from scratch. All reference frames and sequences are used in order to reconstruct the point positions in space of all available pointgroups. For each sequence and reference frame the 3D camera (respectively object) position(s) and orientation(s) are calculated with respect to the 3D points. The calculation is performed every time the function is activated, regardless of the calculation status. If a 3D motion path has been modified before, using the functionality of the Motion Path Window, 3D-Equalizer will destroy these edits and calculate an entirely new solution.
shortcuts: c, SHIFT-c
see also: Motion Path Window, Edit Global Calculation Settings

Finetune Selected Pointgroup

A special calculation method is started, that finetunes the 3D motion paths and 3D point positions of the currently selected pointgroup in all available sequences and reference frames. This function operates on a much higher speed than the function Calc All Objects From Scratch, because it refines an already existing solution rather than calculating everything from scratch. If a 3D motion path has been modified before, using the functionality of the Motion Path Window, 3D-Equalizer will destroy these edits.
shortcuts: f, SHIFT-f
see also: Motion Path Window, Edit Global Calculation Settings

Filter Selected Curves

The 3D motion path of the currently selected pointgroup in the currently selected sequence is filtered using the settings defined in the Edit Postfilter settings requester.
shortcuts: i, SHIFT-i
see also: Edit Postfilter Settings

Calc MoCap Timeshift

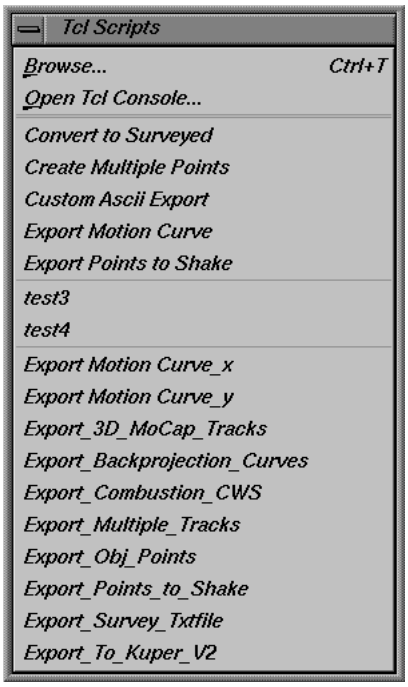
The subframe timeshifts for multiple sequences in a mocap project are calculated. To allow the calculation of the subframe timeshift value for a given sequence, the respective toggle Modify Sequence::Motion Capturing::Determine Timeshift Automatically must be activated.
shortcuts: t, SHIFT-t

Determine Badly Tracked Points...

Opens the requester Determine Badly Tracked Points.
shortcuts: b, SHIFT-b

Menu: Tcl Scripts

Provides functions for the execution of Tcl scripts. During the startup three different directories are scanned for special 3D-Equalizer Tcl scripts with the suffix ".3detcl". The program searches for these scripts in the directories "<3de install dir.>/user_data/tcl_archive/", "~/.3dequalizer/tcl_archive/" and 3DE Prefs::TCL Scripts. For each Tcl script that has been found, a new menu entry is created in the Tcl Scripts pulldown menu. By selecting one of these menu entries, the respective Tcl script is executed.
shortcut: t
see also: Tcl Command Reference

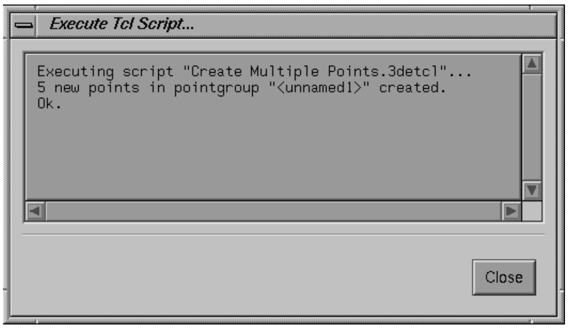


Browse...

Opens a standard file requester Execute Tcl Script. The selected Tcl script will be executed.
shortcut: b, CTRL-t

Open Tcl Console...

Opens the Tcl Console Window, which displays warning and error messages as well as output from the recent executed Tcl script.
shortcut: o

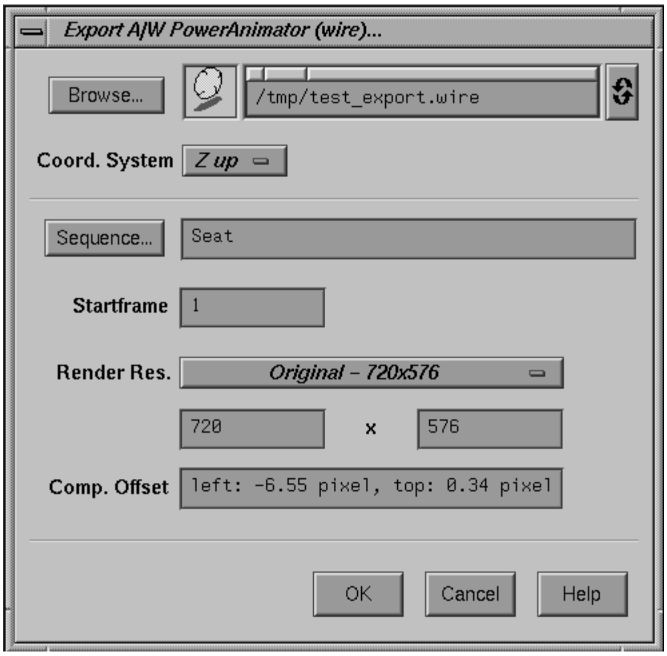


```
Executing script "Create Multiple Points.3detcl"...
5 new points in pointgroup "<unnamed1>" created.
Ok.
```

Tcl Console Window

Requester: Export A|W PowerAnimator...

Exporting the reconstructed camera and points to Alias|Wavefront's PowerAnimator. A wire file is generated that can be easily retrieved starting with PowerAnimator V6 or higher. It may happen that the retrieved points are not visible. In this case, the far clipping plane must be increased (can be found in Power Animator's camera editor).



Export A/W PowerAnimator (wire)...

Browse... /tmp/test_export.wire

Coord. System: Z up

Sequence... Seat

Startframe: 1

Render Res.: Original – 720x576

720 x 576

Comp. Offset: left: -6.55 pixel, top: 0.34 pixel

Requester: Export A|W PowerAnimator...

Browse...
: Opens a file requester. The filename of the wirefile to be generated must be specified.

Coord. System
: Choose among *Y up* and *Z up*. Should be set to the same setting as specified in A|W PowerAnimator.

Sequence...
: Opens a requester which contains the list of currently available sequences and reference frames. The 3D motion paths of all pointgroups in the specified sequence (or reference frame) are exported.

Startframe

This number indicates at which frame the animation begins.

Render Res.

Sometimes it is useful to define a special output render resolution which is different from the resolution of the original image material. The settings of this option menu enable the user to let 3D-Equalizer export at a different image resolution.

Original

If activated, the output render resolution is equal to the original image resolution of the specified sequence (or reference frame).

FOV

If activated, the output render resolution is equal to the size of the FOV setting of the specified sequence (or reference frame).

FOV, Lens Center Offset

If activated, the output render resolution is equal to the minimal resolution needed to cover the entire FOV area. If the Lens Center Offset X/Y setting of the sequence's camera is not equal to (0,0), then it is necessary to increase the output render resolution. During compositing, the rendered layer must be shifted in 2D, in respect to the original image material layer, so that the centers of projection of both layers coincide.

Custom

The user can specify a custom output render resolution in the textfields below the Render Res. option menu.

Comp. Offset

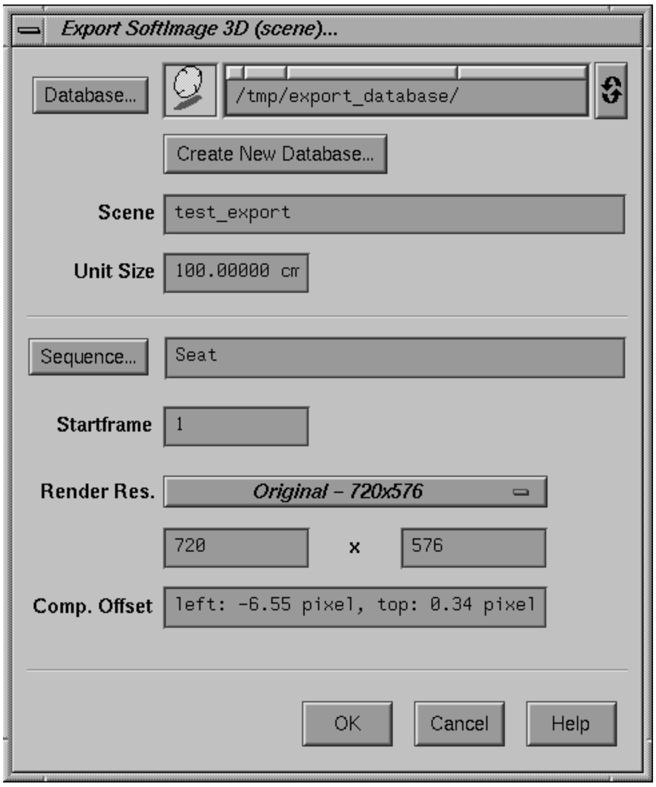
This textfield displays the 2D compositing offset that needs to be applied in order to perfectly match the rendered layer with the original image material layer.

Ok

Causes 3D-Equalizer to generate the export file according to the requested settings.

Requester: Export SoftImage 3D...

Exporting the reconstructed camera and points to SoftImage 3D. An entire scene is generated which can be easily loaded starting with SoftImage 3D 3.5 or higher.


Export SoftImage 3D (scene)...

Database... /tmp/export_database/

Scene: test_export

Unit Size: 100.00000 cm

Sequence... Seat

Startframe: 1

Render Res.: Original – 720x576

720 x 576

Comp. Offset: left: -6.55 pixel, top: 0.34 pixel

Requester: Export SoftImage 3D...

Database...
: Opens a file requester. The path to the database in which the scene should be generated must be specified.

Create New Database...
: Opens a file requester. The user can specify a path to create a new SoftImage database.

Scene
: Name of the scene.

Unit Size

Defines the size of one Softimage length unit in real units. The to be exported reconstructed 3D data are scaled accordingly.

Sequence...

Opens a requester which contains the list of currently available sequences and reference frames. The 3D motion paths of all pointgroups in the specified sequence (or reference frame) are exported.

Startframe

This number indicates at which frame the animation begins.

Render Res.

Sometimes it is useful to define a special output render resolution different from the resolution of the original image material. The settings of this option menu enable the user to let 3D-Equalizer export at a different image resolution.

Original

If activated, the output render resolution is equal to the original image resolution of the specified sequence (or reference frame).

FOV

If activated, the output render resolution is equal to the size of the FOV setting of the specified sequence (or reference frame).

FOV, Lens Center Offset

If activated, the output render resolution is equal to the minimal resolution needed to cover the entire FOV area. If the Lens Center Offset X/Y setting of the sequence’s camera is not equal to (0,0), then it is necessary to increase the output render resolution. During compositing, the rendered layer must be shifted in 2D, in respect to the original image material layer, so that the centers of projection of both layers coincide.

Custom

The user can specify a custom output render resolution in the textfields below the Render Res. option menu.

Comp. Offset

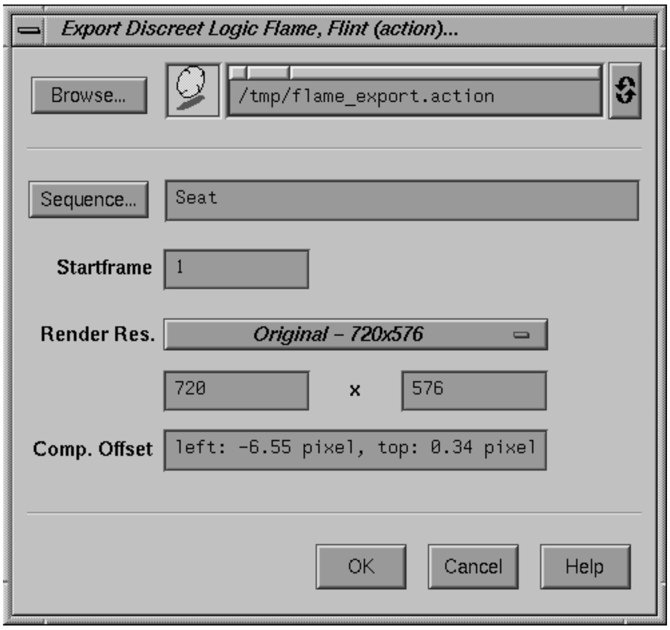
This textfield displays the 2D compositing offset that needs to be applied in order to perfectly match the rendered layer and the original image material layer.

Ok

Causes 3D-Equalizer to generate the export file according to the requested settings.

Requester: Export Discreet Logic Flame, Flint...

Exporting the reconstructed camera and points to Discreet Logic's Flame or Flint. An action file is generated that can be easily opened from within the action module.



/tmp/flame_export.action

Sequence... Seat

Startframe 1

Render Res. Original – 720x576

720 x 576

Comp. Offset left: -6.55 pixel, top: 0.34 pixel

Requester: Export Discreet Logic Flame, Flint...

Browse...
Opens a file requester. The filename of the action file to be generated must be specified.

Sequence...
Opens a requester which contains the list of currently available sequences and reference frames. The 3D motion paths of all pointgroups in the specified sequence (or reference frame) are exported.

Startframe
This number indicates at which frame the animation begins.

Render Res.
Sometimes it is useful to define a special output render resolution different from the resolution of the original image material. The settings of this option menu enable the user to let 3D-Equalizer export at a different image resolution.

Original

If activated, the output render resolution is equal to the original image resolution of the specified sequence (or reference frame).

FOV

If activated, the output render resolution is equal to the size of the FOV setting of the specified sequence (or reference frame).

FOV, Lens Center Offset

If activated, the output render resolution is equal to the minimal resolution needed to cover the entire FOV area. If the Lens Center Offset X/Y setting of the sequence's camera is not equal to (0,0), then it is necessary to increase the output render resolution. During compositing, the rendered layer must be shifted in 2D, in respect to the original image material layer, so that the centers of projection of both layers coincide.

Custom

The user can specify a custom output render resolution in the textfields below the Render Res. option menu.

Comp. Offset

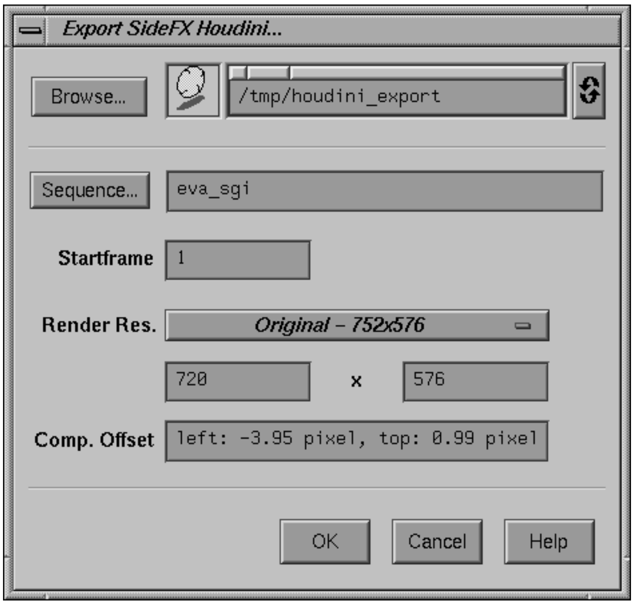
This textfield displays the 2D compositing offset that needs to be applied in order to perfectly match the rendered layer and the original image material layer.

Ok

Causes 3D-Equalizer to generate the export file according to the requested settings.

Requester: Export SideFX Houdini...

Exporting the reconstructed camera and points to SideFX Houdini. A script file is generated that can be easily opened from within the Houdini software.



Export SideFX Houdini...

Browse... /tmp/houdini_export

Sequence... eva_sgi

Startframe 1

Render Res. Original – 752x576

720 x 576

Comp. Offset left: -3.95 pixel, top: 0.99 pixel

Requester: Export SideFX Houdini...

Browse...
Opens a file requester. The filename of the script file to be generated must be specified.

Sequence...
Opens a requester which contains the list of currently available sequences and reference frames. The 3D motion paths of all pointgroups in the specified sequence (or reference frame) are exported.

Startframe
This number indicates at which frame the animation begins.

Render Res.
Sometimes it is useful to define a special output render resolution different from the resolution of the original image material. The settings of this option menu enable the user to let 3D-Equalizer export at a different image resolution.

Original

If activated, the output render resolution is equal to the original image resolution of the specified sequence (or reference frame).

FOV

If activated, the output render resolution is equal to the size of the FOV setting of the specified sequence (or reference frame).

FOV, Lens Center Offset

If activated, the output render resolution is equal to the minimal resolution needed to cover the entire FOV area. If the Lens Center Offset X/Y setting of the sequence's camera is not equal to (0,0), then it is necessary to increase the output render resolution. During compositing, the rendered layer must be shifted in 2D, in respect to the original image material layer, so that the centers of projection of both layers coincide.

Custom

The user can specify a custom output render resolution in the textfields below the Render Res. option menu.

Comp. Offset

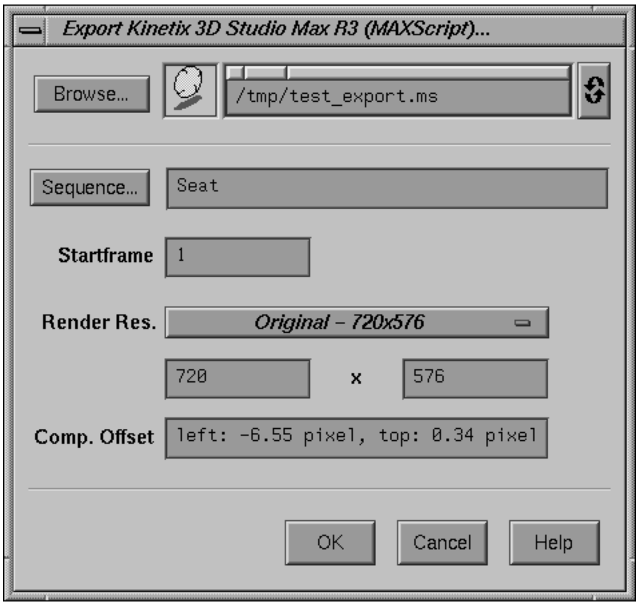
This textfield displays the 2D compositing offset that needs to be applied in order to perfectly match the rendered layer and the original image material layer.

Ok

Causes 3D-Equalizer to generate the export file according to the requested settings.

Requester: Export Kinetix 3D Studio Max R3...

Exporting the reconstructed camera and points to Kinetix 3D Studio Max R3. A MAXScript (ascii) file is generated to be executed within 3D Studio Max R3 using the function Utilities::MAXScript::Run Script.



Export Kinetix 3D Studio Max R3 (MAXScript)...

Browse... /tmp/test_export.ms

Sequence... Seat

Startframe 1

Render Res. Original – 720x576

720 x 576

Comp. Offset left: -6.55 pixel, top: 0.34 pixel

Requester: Export Kinetix 3D Studio Max R3...

Browse...
Opens a file requester. The filename of the MAXScript file that should be generated must be specified.

Sequence...
Opens a requester which contains the list of currently available sequences and reference frames. The 3D motion paths of all pointgroups in the specified sequence (or reference frame) are exported.

Startframe
This number indicates at which frame the animation begins.

Render Res.

Sometimes it is useful to define a special output render resolution different from the resolution of the original image material. The settings of this option menu enable the user to let 3D-Equalizer export at a different image resolution.

Original

If activated, the output render resolution is equal to the original image resolution of the specified sequence (or reference frame).

FOV

If activated, the output render resolution is equal to the size of the FOV setting of the specified sequence (or reference frame).

FOV, Lens Center Offset

If activated, the output render resolution is equal to the minimal resolution needed to cover the entire FOV area. If the Lens Center Offset X/Y setting of the sequence's camera is not equal to (0,0), then it is necessary to increase the output render resolution. During compositing, the rendered layer must be shifted in 2D, in respect to the original image material layer, so that the centers of projection of both layers coincide.

Custom

The user can specify a custom output render resolution in the textfields below the Render Res. option menu.

Comp. Offset

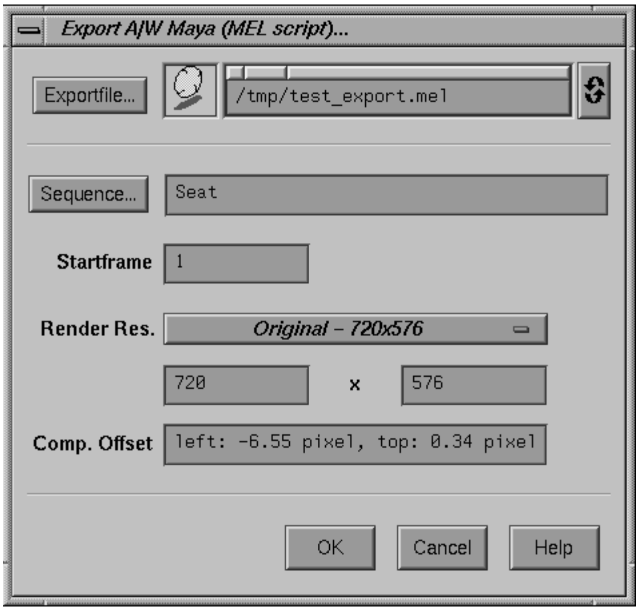
This textfield displays the 2D compositing offset that needs to be applied in order to perfectly match the rendered layer and the original image material layer.

Ok

Causes 3D-Equalizer to generate the export file according to the requested settings.

Requester: Export A|W Maya...

Exporting the reconstructed camera and points to Alias|Wavefront’s Maya. A Mel script file is generated, that can be easily imported into Maya using the function File::Import. It is useful to use exactly the same General Units (see requester Edit 3DEqualizer Preferences) within 3D-Equalizer, as are defined for Maya.



Export A/W Maya (MEL script)...

Exportfile... /tmp/test_export.mel

Sequence... Seat

Startframe 1

Render Res. Original – 720x576

720 x 576

Comp. Offset left: -6.55 pixel, top: 0.34 pixel

Requester: Export A|W Maya...

Browse...

Opens a file requester. The filename of the Mel script file to be generated must be specified.

Sequence...

Opens a requester which contains the list of currently available sequences and reference frames. The 3D motion paths of all pointgroups in the specified sequence (or reference frame) are exported.

Startframe

This number indicates at which frame the animation begins.

Render Res.

Sometimes it is useful to define a special output render resolution different from the resolution of the original image material. The settings of this option menu enable the user to let 3D-Equalizer export at a different image resolution.

Original

If activated, the output render resolution is equal to the original image resolution of the specified sequence (or reference frame).

FOV

If activated, the output render resolution is equal to the size of the FOV setting of the specified sequence (or reference frame).

FOV, Lens Center Offset

If activated, the output render resolution is equal to the minimal resolution needed to cover the entire FOV area. If the Lens Center Offset X/Y setting of the sequence's camera is not equal to (0,0), then it is necessary to increase the output render resolution. During compositing, the rendered layer must be shifted in 2D, in respect to the original image material layer, so that the centers of projection of both layers coincide.

Custom

The user can specify a custom output render resolution in the textfields below the Render Res. option menu.

Comp. Offset

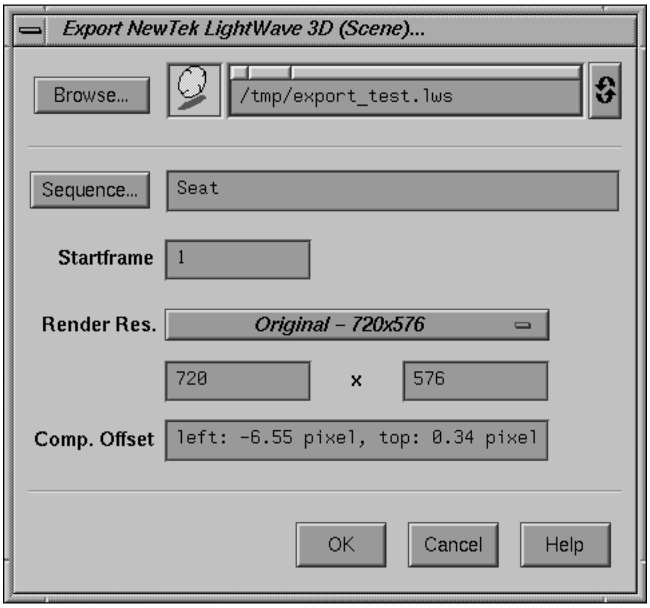
This textfield displays the 2D compositing offset that needs to be applied in order to perfectly match the rendered layer and the original image material layer.

Ok

Causes 3D-Equalizer to generate the export file according to the requested settings.

Requester: Export NewTek LightWave 3D...

Exporting the reconstructed camera and points to NewTek's LightWave 3D. A Scene file is generated which can be opened using the function File::Load Scene...



/tmp/export_test.lws

Sequence... Seat

Startframe 1

Render Res. Original – 720x576

720 x 576

Comp. Offset left: -6.55 pixel, top: 0.34 pixel

Requester: Export NewTek LightWave 3D...

Browse...
Opens a file requester. The filename of the Scene file to be generated must be specified.

Sequence...
Opens a requester which contains the list of currently available sequences and reference frames. The 3D motion paths of all pointgroups in the specified sequence (or reference frame) are exported.

Startframe
This number indicates at which frame the animation begins.

Render Res.
Sometimes it is useful to define a special output render resolution different from the resolution of the original image material. The settings of this option menu enable the user to let 3D-Equalizer export at a different image resolution.

Original

If activated, the output render resolution is equal to the original image resolution of the specified sequence (or reference frame).

FOV

If activated, the output render resolution is equal to the size of the FOV setting of the specified sequence (or reference frame).

FOV, Lens Center Offset

If activated, the output render resolution is equal to the minimal resolution needed to cover the entire FOV area. If the Lens Center Offset X/Y setting of the sequence's camera is not equal to (0,0), then it is necessary to increase the output render resolution. During compositing, the rendered layer must be shifted in 2D, in respect to the original image material layer, so that the centers of projection of both layers coincide.

Custom

The user can specify a custom output render resolution in the textfields below the Render Res. option menu.

Comp. Offset

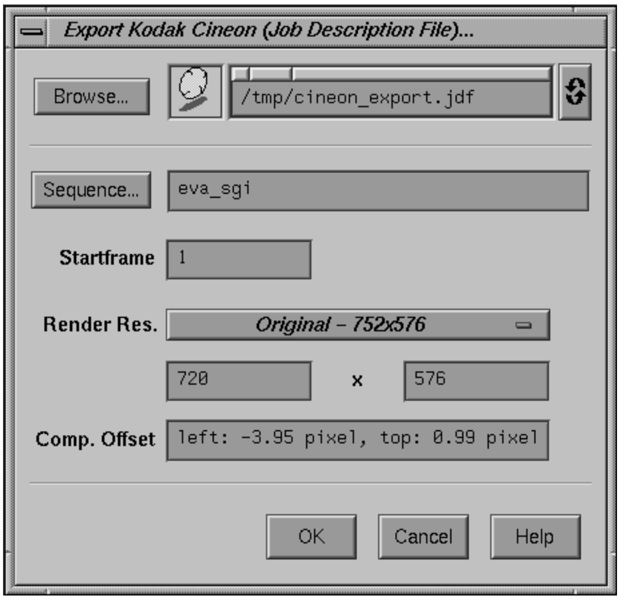
This textfield displays the 2D compositing offset that needs to be applied in order to perfectly match the rendered layer and the original image material layer.

Ok

Causes 3D-Equalizer to generate the export file according to the requested settings.

Requester: Export Kodak Cineon...

Exporting the reconstructed camera and points to Kodak's Cineon. A job description file (jdf) is generated that can be easily opened from within Cineon.



Export Kodak Cineon (Job Description File)...

Browse... /tmp/cineon_export.jdf

Sequence... eva_sgi

Startframe 1

Render Res. Original – 752x576

720 x 576

Comp. Offset left: -3.95 pixel, top: 0.99 pixel

Requester: Export Kodao Cineon...

Browse...
: Opens a file requester. The filename of the jdf to be generated must be specified.

Sequence...
: Opens a requester which contains the list of currently available sequences and reference frames. The 3D motion paths of all pointgroups in the specified sequence (or reference frame) are exported.

Startframe
: This number indicates at which frame the animation begins.

Render Res.
: Sometimes it is useful to define a special output render resolution different from the resolution of the original image material. The settings of this option menu enable the user to let 3D-Equalizer export at a different image resolution.

Original

If activated, the output render resolution is equal to the original image resolution of the specified sequence (or reference frame).

FOV

If activated, the output render resolution is equal to the size of the FOV setting of the specified sequence (or reference frame).

FOV, Lens Center Offset

If activated, the output render resolution is equal to the minimal resolution needed to cover the entire FOV area. If the Lens Center Offset X/Y setting of the sequence's camera is not equal to (0,0), then it is necessary to increase the output render resolution. During compositing, the rendered layer must be shifted in 2D, in respect to the original image material layer, so that the centers of projection of both layers coincide.

Custom

The user can specify a custom output render resolution in the textfields below the Render Res. option menu.

Comp. Offset

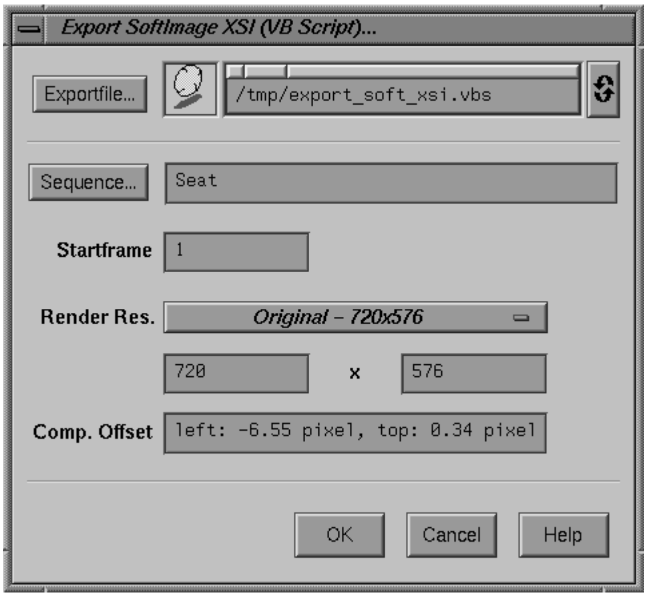
This textfield displays the 2D compositing offset that needs to be applied in order to perfectly match the rendered layer and the original image material layer.

Ok

Causes 3D-Equalizer to generate the export file according to the requested settings.

Requester: Export SoftImage XSI...

Exporting the reconstructed camera and points to SoftImage XSI. A „visual basic“ script (VB) is generated that can be easily executed from within XSI (Script Editor->File->Open..., Run).



Export SoftImage XSI (VB Script)...

Exportfile... /tmp/export_soft_xsi.vbs

Sequence... Seat

Startframe 1

Render Res. Original – 720x576

720 x 576

Comp. Offset left: -6.55 pixel, top: 0.34 pixel

Requester: Export SoftImage XSI...

Browse...
: Opens a file requester. The filename of the VB script to be generated must be specified.

Sequence...
: Opens a requester which contains the list of currently available sequences and reference frames. The 3D motion paths of all pointgroups in the specified sequence (or reference frame) are exported.

Startframe
: This number indicates at which frame the animation begins.

Render Res.
: Sometimes it is useful to define a special output render resolution different from the resolution of the original image material. The settings of this option menu enable the user to let 3D-Equalizer export at a different image resolution.

Original

If activated, the output render resolution is equal to the original image resolution of the specified sequence (or reference frame).

FOV

If activated, the output render resolution is equal to the size of the FOV setting of the specified sequence (or reference frame).

FOV, Lens Center Offset

If activated, the output render resolution is equal to the minimal resolution needed to cover the entire FOV area. If the Lens Center Offset X/Y setting of the sequence’s camera is not equal to (0,0), then it is necessary to increase the output render resolution. During compositing, the rendered layer must be shifted in 2D, in respect to the original image material layer, so that the centers of projection of both layers coincide.

Custom

The user can specify a custom output render resolution in the textfields below the Render Res. option menu.

Comp. Offset

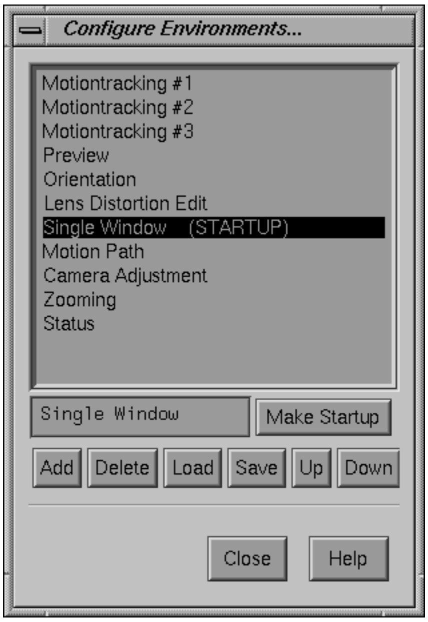
This textfield displays the 2D compositing offset that needs to be applied in order to perfectly match the rendered layer and the original image material layer.

Ok

Causes 3D-Equalizer to generate the export file according to the requested settings.

Requester: Configure Environments...

A requester to create, delete and modify window configurations called environments. The set of current environments is displayed as a list of selectable entries. Clicking on an entry simply selects the respective environment. Double-clicking on an entry selects the environment and loads (displays) it. The name of the currently selected environment can be modified in the textfield below the list. The environment which is automatically loaded during startup time is labeled “STARTUP”.



Requester: Configure Environments...

Make Startup
: The currently selected environment is declared to be the startup environment. It will be automatically loaded each time 3D-Equalizer is started.

Add
: A new environment is created and added to the bottom of the list of existing environments. It automatically becomes the selected one.

Delete
: Deletes the currently selected environment.

Load
The currently selected environment is loaded (displayed).

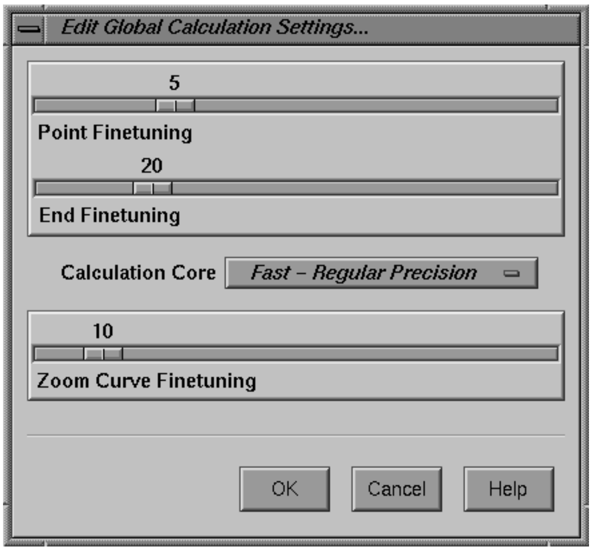
Save
The current window configuration is saved into the currently selected environment.

Up
Moves the currently selected environment up one position in the list of environments.

Down
Moves the currently selected environment down one position in the list of environments.

Requester: Edit Global Calculation Settings...

Global settings concerning the calculation process.



Edit Global Calculation Settings...

Point Finetuning: 5

End Finetuning: 20

Calculation Core: Fast – Regular Precision

Zoom Curve Finetuning: 10

Requester: Edit Global Calculation Settings...

Point Finetuning
Number of finetuning passes after each added point. If many points come in and out of frame quickly, this value should be increased.

End Finetuning

Number of final finetuning passes. Increasing this value can yield better calculation results. It also defines the number of finetuning passes that are performed when the Calc::Finetune Selected Pointgroup function is invoked.

Calculation Core

Optionmenu to configure 3D-Equalizer's calculation core routines.

Fast - Regular Precision

3D-Equalizer computes the project using standard precision. The computation will be fast, but it might happen that under certain circumstances (bad tracking, wrong camera parameters, etc.) the process becomes aborted due to errors.

Slow - Enhanced Precision

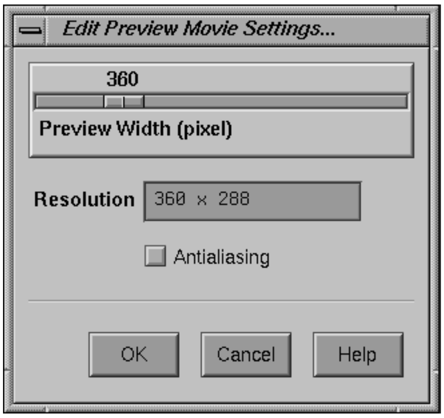
3D-Equalizer computes the project using enhanced precision. The computation will be slow, but it will be more robust against „certain circumstances" (bad tracking, wrong camera parameters, etc.).

Zoom Curve Finetuning

Number of finetuning passes that are performed for calculating the zoom curve.
see also: Camera Zooming Window

Requester: Edit Preview Movie Settings...

Preview and cache movie configuration settings.



Preview Width (pixel): 360

Resolution: 360 x 288

□ Antialiasing

Requester: Edit Preview Movie Settings...

Preview Width (pixel)

This slider is used for specifying the width of the preview and cache movie, in pixels. The height is calculated separately for each sequence (or reference frame).
see also: Resolution

Resolution

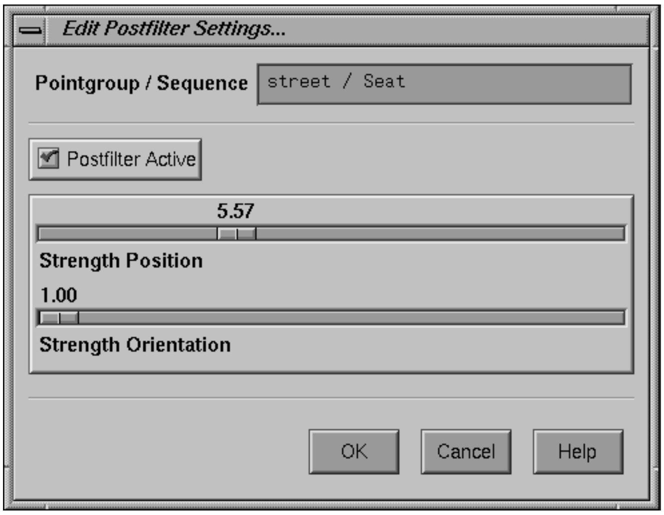
Shows the resolution of the preview and cache movie of the currently selected sequence (or reference frame) in pixels.

Antialiasing

If activated, the dummies are rendered for the preview movie using antialiasing techniques.

Requester: Edit Postfilter Settings...

Provides a functionality for smoothing out the 3D motion path being exported to the animation packages or being used for rendering the preview. Each combination of a pointgroup and a sequence represents a separate 3D motion path. Therefore each combination of pointgroup/sequence has its “own” postfilter setting.



Edit Postfilter Settings...

Pointgroup / Sequence: street / Seat

☑ Postfilter Active

Strength Position: 5.57

Strength Orientation: 1.00

Requester: Edit Postfiltering Settings...

Pointgroup / Sequence

Shows the names of the currently selected pointgroup and sequence.

Postfilter Active

If activated, 3D-Equalizer will filter the 3D motion path of the currently selected pointgroup/sequence combination each time a calculation is performed.

Strength Position

This value specifies the strength of the position postfilter for the currently selected pointgroup/sequence combination. The motion path positions are filtered, the motion path orientation is corrected accordingly. A value of 1.0 means there is no filter effect at all.
see also: Strength Orientation

Strength Orientation

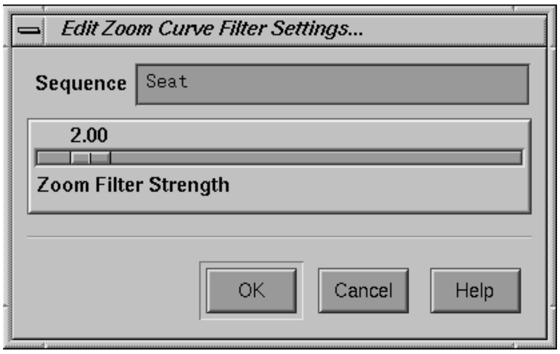
This value specifies the strength of the orientation postfilter for the currently selected pointgroup/sequence combination. This filter can be used for correcting errors of the motiontracking curves. A value of 1.0 means there is no filter effect at all. When you are using the postfilter, the best strategy is the following:

1. Adjust the position postfilter first and leave the orientation postfilter at value 1.
2. Adjust the orientation postfilter afterwards.

see also: Strength Position

Requester: Edit Zoom Curve Filter Settings...

Settings for filtering the zoom curve of the currently selected sequence. Each sequence has its "own" zoom curve filter setting.
see also: Camera Zooming Window



Sequence: Seat

Zoom Filter Strength: 2.00

Requester: Edit Zoom Curve Filter Settings...

Sequence

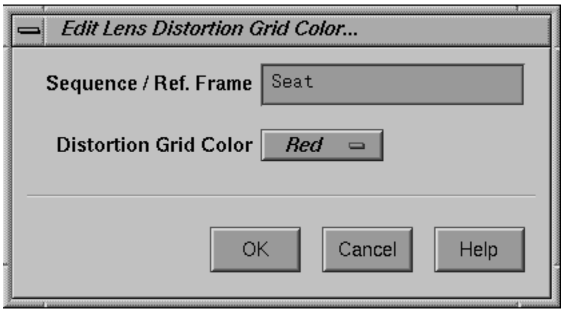
Shows the name of the currently selected sequence.

Zoom Filter Strength

This value specifies the strength of the zoom curve filter of the currently selected sequence. A value of 1 means there is no filter effect at all.

Requester: Edit Lens Distortion Grid Color...

Settings for the color of the distortion grid displayed in the Overview and Zoom Window of the currently selected sequence or reference frame.
see also: Overview Window, Zoom Window



Edit Lens Distortion Grid Color...

Sequence / Ref. Frame: Seat

Distortion Grid Color: Red

Requester: Edit Lens Distortion Grid Color...

Sequence / Ref. Frame

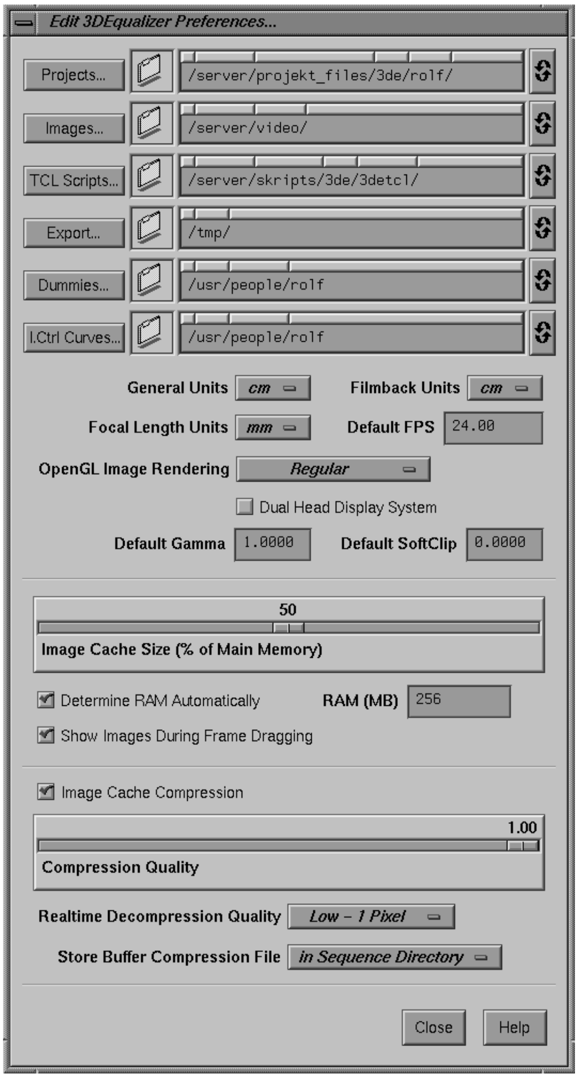
Shows the name of the currently selected sequence or reference frame.

Distortion Grid Color

Allows selection of the color for the distortion grid of the currently selected sequence or reference frame. Choose among *Red, Green, Blue, Yellow, Black, White, Gray*.

Requester: Edit 3DEqualizer Preferences...

Settings for the default directories, default length units and number of imagebuffers.



Edit 3DEqualizer Preferences...

Projects... /server/projekt_files/3de/rolf/

Images... /server/video/

TCL Scripts... /server/skripts/3de/3detcl/

Export... /tmp/

Dummies... /usr/people/rolf

I.Ctrl Curves... /usr/people/rolf

General Units cm

Filmback Units cm

Focal Length Units mm

Default FPS 24.00

OpenGL Image Rendering Regular

☐ Dual Head Display System

Default Gamma 1.0000

Default SoftClip 0.0000

Image Cache Size (% of Main Memory) 50

☑ Determine RAM Automatically

RAM (MB) 256

☑ Show Images During Frame Dragging

☑ Image Cache Compression

Compression Quality 1.00

Realtime Decompression Quality Low - 1 Pixel

Store Buffer Compression File in Sequence Directory

Requester: Edit 3DEqualizer Preferences...

Projects...

The default directory for 3D-Equalizer project files. To integrate environment variables in the path the following syntax must be used: “$(ENV_VARIABLE_NAME)”.

Images...

The default directory for image files. To integrate environment variables in the path the following syntax must be used: “$(ENV_VARIABLE_NAME)”.

TCL Scripts...

The default directory for TCL scripts. To integrate environment variables in the path the following syntax must be used: “$(ENV_VARIABLE_NAME)”.

Export...

The default directory for exporting 3D data. To integrate environment variables in the path the following syntax must be used: “$(ENV_VARIABLE_NAME)”.

Dummies...

The default directory for OBJ Dummy object files. To integrate environment variables in the path the following syntax must be used: “$(ENV_VARIABLE_NAME)”.

I.Ctrl Curves...

The default directory for image control curve files. To integrate environment variables in the path the following syntax must be used: “$(ENV_VARIABLE_NAME)”.

General Units

The default length unit for representing all types of distances.
see also: Orientation Window

Focal Length Units

The default length unit for representing the focal length.
see also: Camera Adjustment Window

Filmback Units

The default length unit for representing filmback width and height.
see also: Camera Adjustment Window

Default FPS

The default frames per second value used for newly created sequence objects.
see also: Modify Sequence Requester

OpenGL Image Rendering

This option menu defines in which way images are drawn in the Overview- and Zoom Window. The update speed of these windows can be dramatically increased by choosing the “correct” option of this menu.

Regular

The to be drawn images are copied from the computer's main memory to the graphics memory, each time the Overview- or Zoom Window is updated. Usually, this option should be activated on all modern SGI/Irix, Linux and Mac OS X workstations.

Integer Zooming

This drawing method is simillar to "Regular". If you are using a SGI/Irix workstation with XZ-, Elan- or Extreme-graphics board this option must be activated. In this case the zoom functionality of the Zoom Window works step wise, not continuously.

Hardware Texturing

The to be drawn images are copied from the computer's main memory to the graphics memory only once. After that, they are drawn using the graphic board's "hardware texturing" features.

Dual Head Display System

If activated, requesters open up in the middle of the left-hand monitor.

Default Gamma

The default gamma value used for newly created sequence objects.
see also: Modify Sequence Requester

Default Softclip

The default softclip value used for newly created sequence objects.
see also: Modify Sequence Requester

Image Cache Size

This silder defines the amount of memory allocated to be used for 3D-Equalizer's image buffers. An image buffer is an area of RAM in which a single image of a sequence or a reference frame image is temporarily stored during a session. The actual number of available image buffers depends on the respective image size. Set this as high as possible in order to work fluidly, but take care not to run out of memory. Usually 50% of the available RAM is a good setting.
see also: Main Window::Playback

Determine Main Memory Size Automatically

If activated, 3D-Equalizer determines the amount of RAM automatically. If deactivated, the amount of RAM can be entered manually in the textfield above.

Show Images During Frame Dragging

If activated, 3D-Equalizer properly updates the Overview- and Zoom Window's background, while the frame slider is dragged. If deactivated, the Overview- and Zoom Window's background keeps to be drawn in black.

Image Cache Compression

If activated, 3D-Equalizer compresses every imported image and stores it in its image buffers. If a formerly compressed image has to be displayed again, it will be decompressed rapidely out of memory, and thus no time-consuming filesystem access must be performed.

Compression Quality

This slider defines the compression quality with which 3D-Equalizer compresses to be imported images. The best quality setting (1.0) is equal to a compression ratio of about 15:1.

Realtime Decompression Quality

Optionmenu to define the image decompression quality while dragging the frame slider.

Low - 1 Pixel

The images will be decompressed with reduced quality: for each 8x8 pixel subimage only 1 pixel will be decompressed.

Full - 8x8 Pixels

The images will be decompressed with full quality.

Store Buffer Compression File

Optionmenu to define in which directory 3D-Equalizer will store the so-called buffer compression files.
see also: Playback Menu

in Sequence Directory

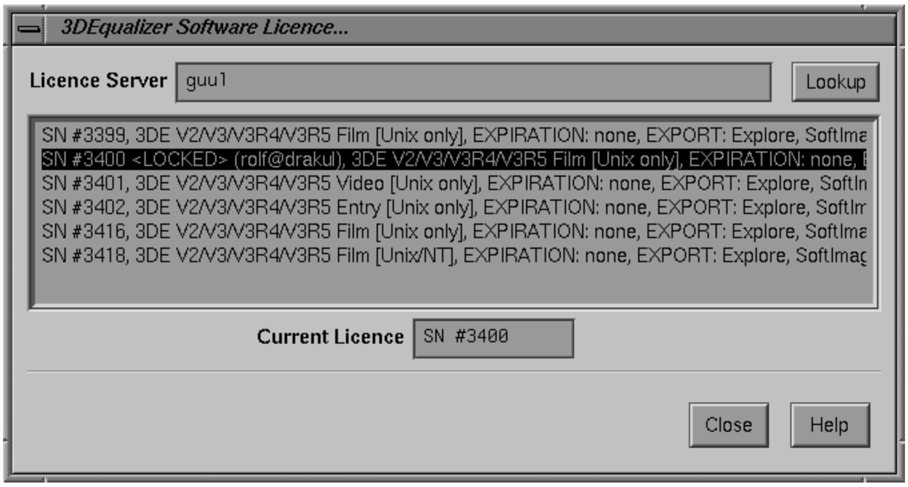
If activated, buffer compression files will be stored in the same directory where the currently selected sequence's image files are located.

in Project Directory

If actiaved, buffer compression files will be stored in the directory where the currently loaded project file is located.

Requester: 3DEqualizer Software Licence...

All available floating licences and the floating licence currently used by the system are displayed. By double clicking a licence entry, the user can grab another licence. The licence in use is automatically unlocked, and can be grabbed by another user on another host.



Licence Server: guu1

SN #3399, 3DE V2/V3/V3R4/V3R5 Film [Unix only], EXPIRATION: none, EXPORT: Explore, SoftIma
SN #3400 <LOCKED> (rolf@drakul), 3DE V2/V3/V3R4/V3R5 Film [Unix only], EXPIRATION: none, E
SN #3401, 3DE V2/V3/V3R4/V3R5 Video [Unix only], EXPIRATION: none, EXPORT: Explore, SoftIn
SN #3402, 3DE V2/V3/V3R4/V3R5 Entry [Unix only], EXPIRATION: none, EXPORT: Explore, SoftIm
SN #3416, 3DE V2/V3/V3R4/V3R5 Film [Unix only], EXPIRATION: none, EXPORT: Explore, SoftIma
SN #3418, 3DE V2/V3/V3R4/V3R5 Film [Unix/NT], EXPIRATION: none, EXPORT: Explore, SoftImag

Current Licence: SN #3400

Requester: 3DEqualizer Software License...

Licence Server

Enter the name (or IP address) of a Flc server workstation here. All available licences are immediately displayed in the list below the text field.

Lookup

Update the list of available licences.

Current Licence

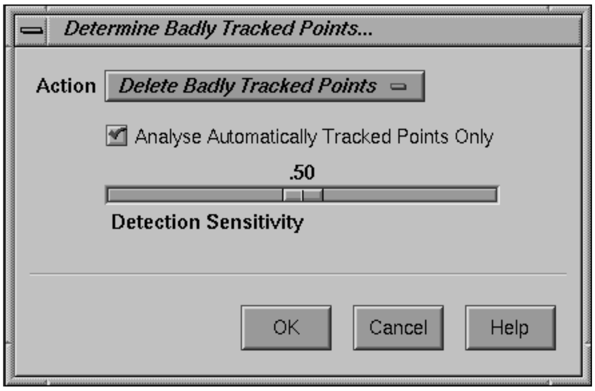
Displays the serial number. of the currently used licence. If <none> is displayed, no licence has been grabbed.

Close

The system tries to grab the currently selected licence. The requester is closed.

Requester: Determine Badly Tracked Points...

A requester to configure a function for determining bad tracking points.



Determine Badly Tracked Points...
Action
Delete Badly Tracked Points
☑ Analyse Automatically Tracked Points Only
.50
Detection Sensitivity
OK
Cancel
Help

Requester: Determine Badly Tracked Points...

Action

This option menu selects the mode of operation.

Delete Badly Tracked Points

Points that are identified to be badly tracked, become automatically deleted after the calculation procedure is completed.

Select Badly Tracked Points

Points that are identified to be badly tracked, only become selected after the calculation procedure is completed.

Analyse Automatically Tracked Points Only

If activated, 3DEqualizer does not analyse any points that were manually created by the user.

Detection Sensitivity

This slider defines how sensitive the calculation procedure is against badly tracked points. When set to 0.0 3DEqualizer will identify a small number of bad points, however when set to 1.0 it will identify a larger number of bad points.

Ok

Starts the calculation procedure for determining bad tracking points.

Cancel

Closes the requester without to apply any changes to the current project.

Zoom Window

This window provides the functionality for creating screen points and tracking them in the image sequences. Additionally, functions for measuring camera lens distortion are provided in this window. In the following, the various functions are explained.

In general, there are three different modes of operation, the motiontracking mode, the distortion grid edit mode and the autotracking mode. The current mode can be chosen from the Controls menu.

Miscellaneous

Display

The Zoom Window display represents a rectangular cut-out part of the frame currently displayed in the Overview Window. The location of this cut-out can be selected by either holding the middle mouse button and moving the mouse or by using the slider underneath the display for horizontal movement and the slider on the left hand side for vertical movement. The zoom-in-out functionality is accessed by holding the right mouse button and mouse movement or by using the slider on the right hand side of the display. The cut-out part is represented in the Overview Window display by a dotted red rectangle.

Keys

CURSOR-LEFT and CURSOR-RIGHT are used for jumping back and forth in the currenty selected sequence according to the setting of Frame Keyboard Grid. CTRL-CURSOR-LEFT and CTRL-CURSOR-RIGHT are used for going back and forth by one frame. CURSOR-UP and CURSOR-DOWN are used for selecting points. PAGE-UP/PAGE-DOWN jumps to the previous/next keyframe of the currently selected point. HOME/END jumps to the first/last sequence or reference frame.

Sliders

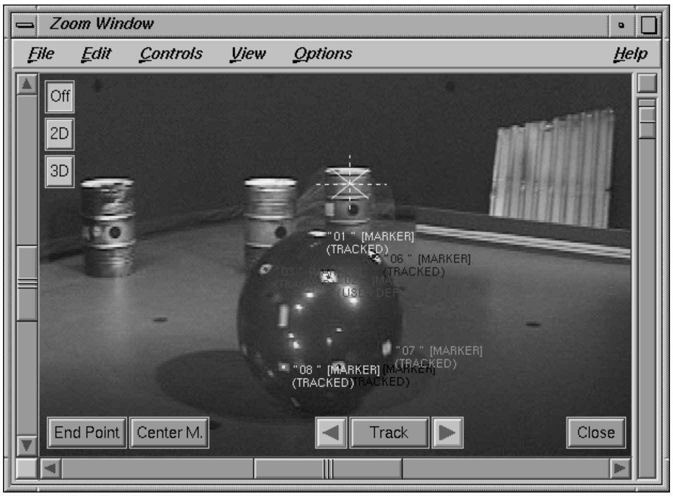
The slider on the right hand side of the display is used for zooming, the sliders on the left and underneath are used for moving the cut-out of the Main Window display represented in the Zoom Window display. Pressing the button above the zoom slider moves the knob to the upmost position, so that the entire image is displayed.

Field of View

The field of view can be adjusted as follows: hold CTRL, drag the field of view border with the left mouse button. The entire field of view can be moved by holding CTRL, clicking on the field of view’s middle cross and moving the mouse.

Motiontracking Mode

In the following section, the behaviour of the Zoom Window is described when motiontracking mode is active (see also Controls menu). This mode is primarily used for manual point tracking.



Zoom Window while in motiontracking mode

Screen Points

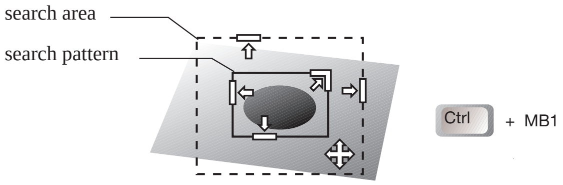
A screen point is a representation of a point (in three-dimensional space) on the screen for a specific frame. A screen point can be positioned in different ways. The most fundamental one is to navigate the Zoom Window's display to the location of interest within the frame and to place the screen point at the intended position using the left mouse button. The screen point is accompanied by text information. This screen point information text consists of the following components: identifier, selection-status, key-status.

Selecting Points

Holding down the ALT-key while clicking on a point with the left mouse button, selects the respective point. To select multiple points the ALT- and the SHIFT-key must be pressed, while clicking with the left mouse button. Holding down the ALT-key while click-dragging the left mouse button causes 3DE to draw a rubberbox. All points located within that rubberbox become selected. CURSOR-UP and CURSOR-DOWN are used for selecting single points.

Tracking area

If the currently selected point is going to be tracked, it is necessary to specify the search pattern size and the search area size. For this purpose, the boxes must be visible (see also Show Tracking Area). The borders of these areas can be moved by pressing CTRL and touching the respective border (hold left mouse button) and dragging them around. The entire search pattern can be moved by pressing CTRL, clicking the left mouse button within the borders and moving the mouse.



selection-status

If a point is currently selected, double dotted lines appear above and underneath the text information. If the point is not selected, the double dotted lines do not appear.

key-status

The key-status of a screen point in a given frame assumes the following values: *userdef, userdef endframe, userdef startframe, splined, forward, backward, tracked, obsolete, invalid*.

userdef means the screen point has been placed by the user, or the user has modified some parameters of the screen point e.g. size of the tracking area. In this way the current frame is made a keyframe for the tracking procedure of the currently selected point.

userdef endframe indicates that the motion curve of the point ends in the current frame.

userdef startframe indicates that the motion curve of the point begins in the current frame. The subsequent screen points are valid unless an endframe occurs.

splined means the screen point is valid, but has not been placed by the user. Its position results from one of the previous frames and, where applicable, from one of the following frames where the screen point has been userdefined or tracked.

forward means the screen point has undergone a tracking procedure based on the previous frame, and its position will be modified in a backward tracking procedure, if such a procedure is started.

backward means the screen point has undergone a tracking procedure based on the following frame, and its position will be modified in a forward tracking procedure, if such a procedure is started.

tracked means the screen point has undergone its complete tracking procedure as specified in the points trackingmode.

obsolete means the screen point has been tracked, but in some of the frames on which this tracking procedure was based, the screen points have changed. The current screen point may be placed well in the current frame, but if tracking is applied again its position will be calculated anew.

invalid means the screen point is not valid, i.e. does not exist in the current frame, hence does not contribute to any reconstruction procedure.

End Point

This button defines/undefines the current screen point as endpoint. The key-status “userdef endframe” indicates that the motion curve of the point ends in the current frame.

Center M.

This button can only be pressed as long as a single point is select. Depending on the trackingmode of the currently selected point, 3DEqualizer visualizes the tracking process of the current frame by means of white lines. If patterntracking is activated, the deformation of search area and search pattern are visualized. If markertracking is activated, the outline of the to be tracked marker is visualized. If cornertracking is activated, the edges that form the to be tracked corner are visualized.
In addition, if marker- or cornertracking is activated, the tracking point position is altered and becomes automatically centered over the marker (corner).

Track backward

The display jumps to the previous frame. The screen point(s) of the previous frame are tracked if necessary. ◀

Track forward

The display jumps to the following frame. The screen point(s) of the following frame are tracked if necessary. ▶

Track

Starts the tracking procedure according to the settings of the points’ tracking areas and tracking parameters. Tracking means that the search pattern of a point in a given frame is searched and localized in the neighbouring frame. When the patterns are being compared, the settings of the Image Control Window are taken into account and can be used for improving the contrast of the patterns.

Off

If activated, no automatic centering of tracking points takes place.

2D

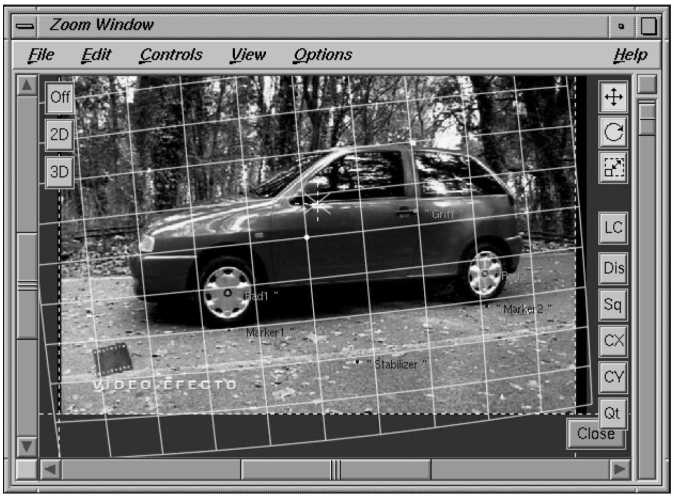
If activated, the currently selected tracking points are automatically centered in the Zoom Window's display area.

3D

If activated, the currently selected point's 3D positions are automatically centered in the Zoom Window's display area. The function Zoom Window::View::Show Reconstructed Points has to be activated (green crosses visible) in order to let this work properly.

Distortion Grid Edit Mode

In the following section, the behaviour of the Zoom Window is described when distortion grid edit mode is active (see also Controls menu). A virtual distortion grid is displayed, which can be edited by the use of the buttons located on the right-hand side.



Zoom Window while in distortion grid edit mode

Translate Distortion Grid

If activated, the user is able to move the distortion grid on the screen by holding down the left mouse button while dragging the mouse.

Rotate Distortion Grid

If activated, the user can rotate the distortion grid around its white center point by holding down the left mouse button and dragging the mouse.

Scale Distortion Grid

If activated, the user can scale the distortion grid by holding down the left mouse button and dragging the mouse.

LC

If activated, the user can modify the Lens Center Offset X & Y parameters of the camera object which is linked to the currently selected sequence or reference frame. Clicking on the screen with the left mouse button sets the new position of the lens center offset x & y values.

Dis

If activated, the user can modify the Lens Distortion parameter of the camera object which is linked to the currently selected sequence or reference frame. Holding down the left mouse button and dragging the mouse to the left decreases the lens distortion value, whereas dragging the mouse to the right increases the lens distortion value.

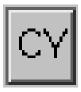
Sq

If activated, the user can modify the Anamorphic Squeeze parameter of the camera object which is linked to the currently selected sequence or reference frame. Holding down the left mouse button and dragging the mouse to the left decreases the anamorphic squeeze value, whereas dragging the mouse to the right increases the anamorphic squeeze value.

CX

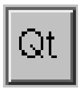
If activated, the user can modify the Curvature X parameter of the camera object which is linked to the currently selected sequence or reference frame. Holding down the left mouse button and dragging the mouse to the left decreases the curvature x value, whereas dragging the mouse to the right increases the curvature x value.

CY



If activated, the user can modify the Curvature Y parameter of the camera object which is linked to the currently selected sequence or reference frame. Holding down the left mouse button and dragging the mouse to the left decreases the curvature y value, whereas dragging the mouse to the right increases the curvature y value.

Qt



If activated, the user can modify the Quartic Distortion parameter of the camera object which is linked to the currently selected sequence or reference frame. Holding down the left mouse button and dragging the mouse to the left decreases the quartic distortion value, whereas dragging the mouse to the right increases the quartic distortion value.

Off

If activated, no automatic centering of tracking points takes place.

2D

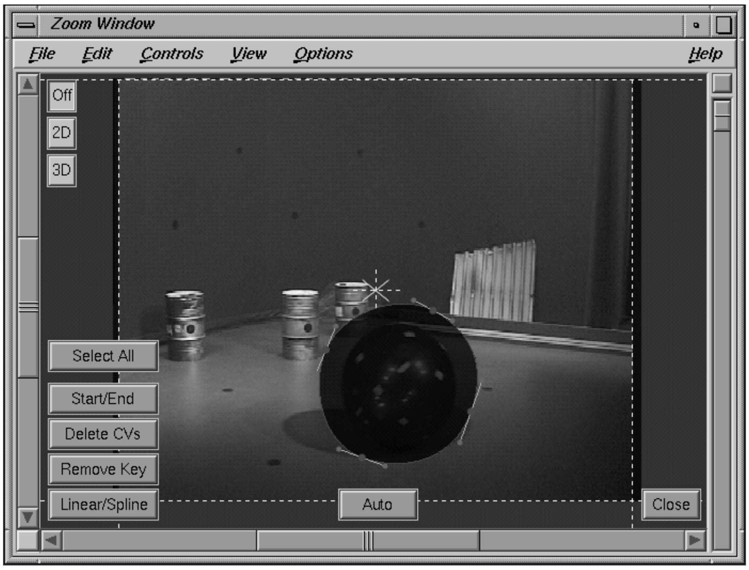
If activated, the currently selected tracking points are automatically centered in the Zoom Window’s display area.

3D

If activated, the currently selected point’s 3D positions are automatically centered in the Zoom Window’s display area. The function Zoom Window::View::Show Reconstructed Points has to be activated (green crosses visible) in order to let this work properly.

Autotracking Mode

In the following section, the behaviour of the Zoom Window is described when autotracking mode is active (see also Controls menu). This mode is primarily used to generate multiple points by means of 3DEqualizer’s autotracking engine.



Zoom Window while in autotracking mode

Spline Area Mattes

A spline area matte is always associated to exactly one autotracking region. It usually consists of a set of control vertices (“CVs”), which precisely define its outline. Each CV can be animated using simple keyframing techniques. During the automatic tracking procedure, points are only tracked inside or outside of a spline area matte, depending on the spline area’s configuration. In other words, by means of spline area mattes 3DEqualizer can be forced to track points only in certain areas of the FOV.

Only the spline area matte of the currently selected autotracking region is displayed and thus can be edited in the Zoom Window. If a spline area matte is keyframed in a certain frame, its CVs and outline are drawn in red, otherwise they are drawn in white. Currently selected CVs are drawn in yellow.

Creating and editing CVs

Holding down the CTRL-key while clicking with the left mouse button in the Zoom Window's display area will create a new CV. Already existing CVs can be edited by click-dragging them with the left mouse button. If a CV is edited, the spline area matte automatically becomes keyframed in the current frame.

Selecting CVs

A single CV can be selected by clicking on it with the left mouse button. Multiple CVs can be selected by holding down the SHIFT-key while clicking on them with the left mouse button. By holding down the ALT-key while click-dragging the left mouse button a rubberbox is created. All CVs located within the rubberbox become selected.

Select All

All CVs of the current autotracking region's spline area matte become selected.

Start/End

A spline area matte can be temporarily disabled (enabled), starting from the current frame, by clicking on the button Start/End. Disabled spline area mattes are indicated by a dotted outline.

Delete CVs

All selected CVs of the current autotracking region's spline area matte are deleted.

Remove Key

If a spline area matte is keyframed in the current frame, its CVs' keyframe positions can be removed by clicking on this button.

Linear/Spline

All selected CVs are toggled between linear and spline mode. This setting is not keyframed and thus is valid throughout the entire sequence. A spline CV has a tangent, whereas a linear CV only consists of a single point.

Auto

3DEqualizer's automatic tracking procedure is started. According to the settings of the set of autotracking regions, associated to the currently selected sequence, multiple points become created and tracked.

Off

If activated, no automatic centering of tracking points takes place.

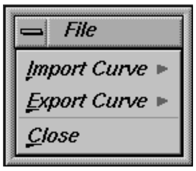
2D

If activated, the currently selected tracking points are automatically centered in the Zoom Window's display area.

3D

If activated, the currently selected point's 3D positions are automatically centered in the Zoom Window's display area. The function Zoom Window::View::Show Reconstructed Points has to be activated (green crosses visible) in order to let this work properly.

Menu: File



Contains functions for saving and loading 2D motion (tracking) curves.
shortcut: ALT-f

Import Curve

Functions for loading 2D motion curves.
shortcut: i

Raw

Opens a file requester. The motion curve of the currently selected point is replaced by the data found in the specified file. It must be in 3D-Equalizer raw motion curve format, a binary format which holds all tracking curve information including tracking areas and status information.
shortcut: r
see also: Export Curve

Flame

Opens a file requester. The motion curve of the currently selected point is replaced by the data found in the specified file. It must be in Discreet Logic Flame Raw (Track) format.
shortcut: f

Cineon

Opens a file requester. The motion curve of the currently selected point is replaced by the data found in the specified file. It must be in Kodak Cineon Raw format.
shortcut: c

Export Curve

Functions for saving 2D motion curves.
shortcut: e

Raw

Opens a file requester. The data of the motion curve of the currently selected point is saved into the specified file, using 3D-Equalizer raw motion curve format. The 3D-Equalizer raw motion curve format is a binary format which holds all tracking curve

information including tracking areas and status information.This function can be used to share motion curves between different projects.
shortcut: r
see also: Import Curve

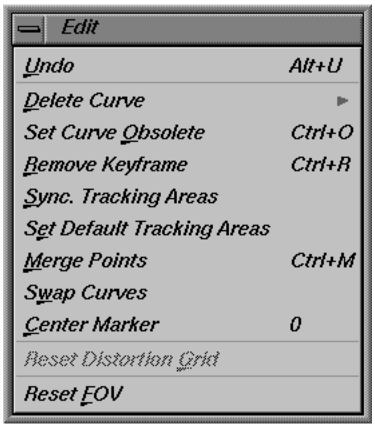
Ascii

Opens a file requester. The data of the motion curve of the currently selected point is saved into the specified textfile.
shortcut: a

Close

Closes the Zoom Window.
shortcut: c

Menu: Edit



Contains functions for editing motion curves and the field of view.
shortcut: ALT-e

Undo

Undo the previous point editing action.
shortcuts: u, ALT-u

Delete Curve

Contains functions for deleting parts of motion curves.
shortcut: d

All

Deletes the entire motion curve of the currently selected points.
shortcut: a

Until End

Deletes all screen points after the current frame until the end of the sequence and defines the current frame as the end-frame for this motion curve.
shortcut: e

Set Curve Obsolete

The tracking curves of the currently selected points are declared to be obsolete.
shortcuts: o, CTRL-o

Remove Keyframe

Removes the key-status userdef from a screen point. The screen point becomes invalid or splined depending on the status of the neighbouring points.
shortcuts: r, CTRL-r

Sync. Tracking Areas

The size and position of the currently selected point’s tracking areas are copied to all keyframes. So, the tracking areas of the entire tracking curve become the same.
shortcut: s

Set Default Tracking Areas

The size and position of the currently selected point’s tracking areas becomes the default tracking areas. The tracking areas of a newly created point object are of the same dimensions as the default tracking areas.
shortcut: e

Merge Points

The tracking curves of the currently selected points are merged, creating a single tracking curve. This tracking curve is copied into a new point object and all formerly selected points become deleted.
shortcuts: m, CTRL-m

Swap Curves

The tracking curves of two selected points are exchanged. This operation only takes place in the currently selected sequence.
shortcut: w

Center Marker

If marker- or cornertracking is activated, the tracking point’s position is altered and becomes automatically centered over the marker (corner).
shortcuts: c, NUM-0

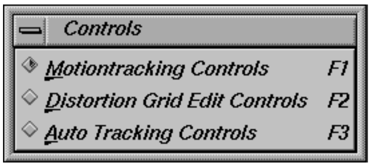
Reset Distortion Grid

Resets the translation, rotation and scale of the distortion grid of the currently selected sequence or reference frame.
shortcut: g

Reset FOV

Resets the field of view to the default settings. The field of view is located at the borders of the frame.
shortcut: f

Menu: Controls


Controls
◆ Motiontracking Controls F1
◇ Distortion Grid Edit Controls F2
◇ Auto Tracking Controls F3

Toggle buttons for selecting the motiontracking, lens distortion grid edit or autotracking mode of the Zoom Window.
shortcut: c

Motiontracking Controls

> If activated, the Zoom Window operates in motiontracking mode.
> *shortcuts*: m, F1

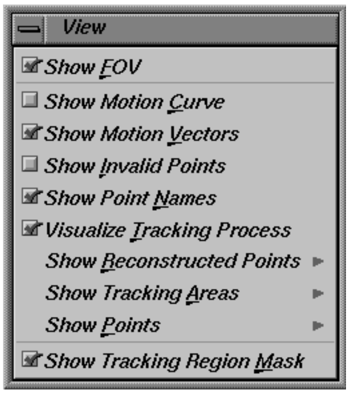
Distortion Grid Edit Controls

> If activated, the Zoom Window operates in lens distortion edit mode.
> *shortcuts*: d, F2

Auto Tracking Controls

> If activated, the Zoom Window operates in autotracking mode.
> *shortcuts*: a, F3

Menu: View



Contains toggle buttons for configuring the Zoom Window display.

Show FOV

> If activated, the field of view is displayed in the Zoom Window display.
> *shortcut*: f

Show Motion Curve

> If activated, the two-dimensional motion curve of the screen point for the entire sequence is displayed, depending on the setting of the menu item Show Invalid Points.
> *shortcut*: c

Show Motion Vectors

> If activated, a small motion vetor is displayed for each visible tracking point.
> *shortcut*: v

Show Invalid Points

If activated, the screen points which are declared invalid in the current frame are displayed.
shortcut: i

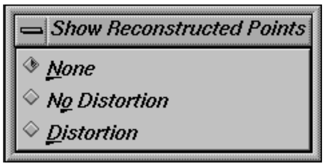
Show Point Names

If activated, the names of all visible points as well as their key-status information are displayed.
shortcut: n

Visualize Tracking Process

If activated, the tracking process will be visualized by means of white lines, depending on the trackingmode of the currently selected point. If patterntracking is activated, the deformation of search area and search pattern are visualized. If markertracking is activated, the outline of the to be tracked marker is visualized. If cornertracking is activated, the edges that form the to be tracked corner are visualized.

Show Reconstructed Points



Contains toggles for the display of reconstructed 3D points which are backprojected onto the image plane.
shortcut: r

None

If activated, no reconstructed points are displayed at all.
shortcut: n

No Distortion

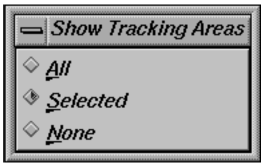
For each successfully calculated point, a green cross is displayed in the zoom window. The positions of these green crosses on the image plane are calculated according to a back-projection process. This means, the reconstructed 3D points are projected back on the image plane using a standard CG perspective projection routine. The mismatch between the original tracking point and its green cross in each frame is a measure for the quality of the reconstruction. The lens distortion settings of the camera of the currently selected sequence of reference frame ARE NOT taken into account, therefore the postions of the green crosses might be falsified.
shortcut: o

Distortion

For each successfully calculated point, a green cross is displayed in the zoom window. The positions of these green crosses on the image plane are calculated according to a back-projection process. This means, the reconstructed 3D points are projected back on the image plane using a standard CG perspective projection routine. The mismatch

between the original tracking point and its green cross in each frame is a measure for the quality of the reconstruction. The lens distortion settings of the camera of the currently selected sequence of reference frame ARE taken into account.
shortcut: d

Show Tracking Areas



Contains toggles for the display of the motiontracking areas of the points.
shortcut: a

All

If activated, the search patterns and the search areas of all visible points are displayed.
shortcut: a

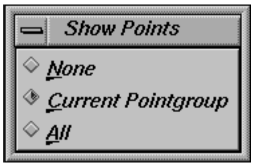
Selected

If activated, only the search pattern and the search area of the currently selected point are displayed.
shortcut: s

None

If activated, no motiontracking areas are displayed at all.
shortcut: n

Show Points



Contains toggles for hiding and showing points.
shortcut: p

Of All Pointgroups

If activated, all points of all pointgroups are displayed.
shortcut: a

Of Current Pointgroup

If activated, all points of the currently selected pointgroup are displayed.
shortcut: c

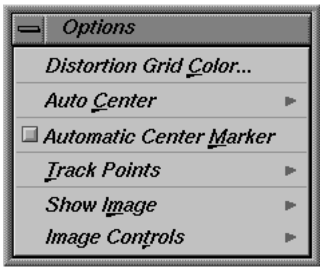
Only Selected Point

If activated, only the currently selected point of the currently selected pointgroup is displayed.
shortcut: s

Show Tracking Region Mask

If autotracking mode and this toggle button are activated, the areas of the FOV inside which no points are going to be tracked by 3DE's autotracking engine are displayed with reduced brithness.
shortcut: m

Menu: Options



Contains toggles for the Zoom Window display.
shortcut: ALT-o

Distortion Grid Color...

Opens a requester Edit Distortion Grid Color.
shortcut: c
see also: Main Window

Auto Center

Contains toggle buttons for centering tracking points on the Zoom Window's display.
shortcut: c

Off

If activated, no automatic centering of tracking points takes place.
shortcut: o

2D Tracking Point

If activated, the currently selected tracking points are automatically centered in the Zoom Window's display area.
shortcut: 2

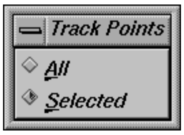
3D Reconstructed Point

If activated, the currently selected point's 3D positions are automatically centered in the Zoom Window's display area. The function Zoom Window::View::Show Reconstructed Points has to be activated (green crosses visible) in order to let this work properly.
shortcut: 3

Automatic Center Marker

If activated, the function Zoom Window::Edit::Center Marker is triggered, each time the user modifies the currently selected tracking point's position with the left mouse button.
shortcut: m

Track Points



Contains toggle buttons for configuring the motiontracking feature.
shortcut: t
see also: Track backwards, Track forwards, Track

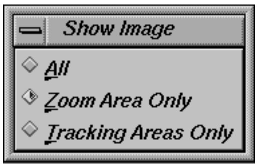
All

During either the automatic or manual single-step tracking procedure, all available points of the currently selected pointgroup are tracked.
shortcut: a

Selected

During either the automatic or manual single-step tracking procedure, only the currently selected point of the currently selected pointgroup is tracked.
shortcut: s

Show Image



Contains toggles for the image subload. The subload functions are especially useful when high resolution images are processed, since the working speed is increased considerably.
shortcut: m

All

If activated, the entire image is loaded into the memory.
shortcut: a

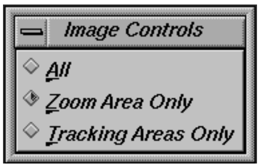
Zoom Area Only

If activated, only the part of the image that corresponds to the zoom area (the area displayed in the Zoom Window) is loaded into the memory. This function is only available for the film option of 3D-Equalizer.
shortcut: z

Tracking Area Only

If activated, only the tracking area (the dotted tracking box) is loaded in the memory. This increases the working speed. This function is only available for the film option of 3D-Equalizer.
shortcut: t

Image Controls



Contains toggles in order to choose whether the Image Control functions should be applied to the whole image or to a subregion only. These functions are useful when high resolution pictures are being processed. When the Image Control is applied to a subregion, this will speed up the workflow.
shortcut: t

All

Applies the Image Control functions to the entire image.
shortcut: a

Zoom Area only

Applies the Image Control functions to the zoom area only. The zoom area is the region of the image displayed in the Zoom Window. This function is only available for the film option of 3D-Equalizer.
shortcut: z

Tracking Area Only

Applies the Image Control functions to the tracking area only. The tracking area is the subregion of the image inside the dotted tracking box. This function is only available for the film option of 3D-Equalizer.
shortcut: t

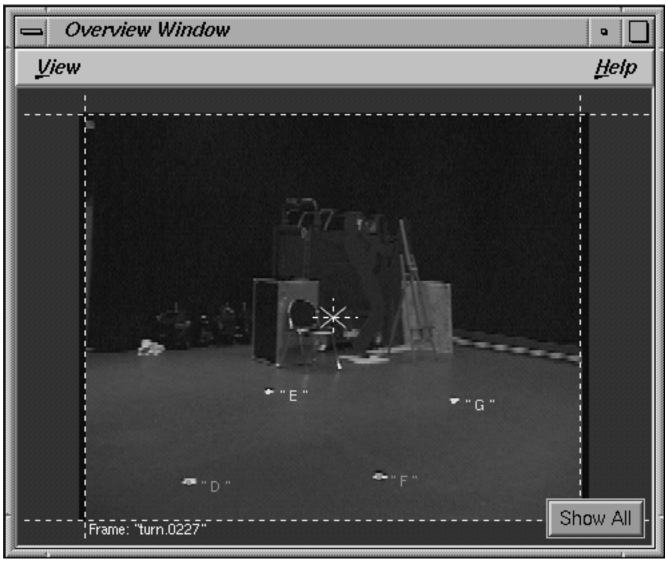
Overview Window

The Overview Window mainly consists of an image display area which shows the currently selected frame entirely. This window is also used for playing back preview and cache movies.

Miscellaneous

Display

The Overview Window display field displays the current frame of the currently selected sequence or reference frame. Field of view borders are displayed according to the setting of Show FOV. Screen points are also shown. The contents of the display can be configured by means of the menu View. When the Zoom Window is open, the cutout area displayed there can be localized by holding the middle mouse button and moving the mouse in the Overview Window display.



Overview Window

Keys

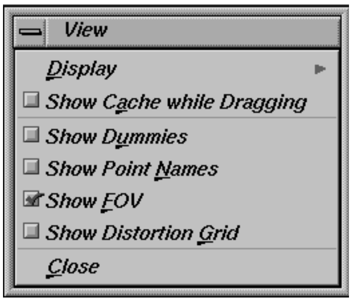
CURSOR-LEFT and CURSOR-RIGHT are used for jumping back and forth in the currently selected sequence according to the setting of Frame Keyboard Grid. CTRL-CURSOR-LEFT and CTRL-CURSOR-RIGHT are used for going back and forth by one frame. CURSOR-UP and CURSOR-DOWN are used for selecting points. PAGE-UP/PAGE-DOWN jumps to the previous/next keyframe of the currently selected point. HOME/END jumps to the first/last sequence or reference frame.

Show All

If the subload functions of the film version are activated, only a subarea of the image is loaded into memory (see also Zoom Window). By clicking on the Show All button, 3D-Equalizer is forced to load the currently selected frame *completely* into its image buffer and display it in the Overview Window.

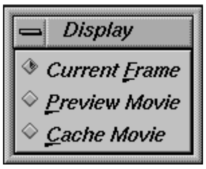
Menu: View

Contains several toggle buttons to configure the Overview Window's display.
shortcut: ALT-v



Display

Contains toggle buttons for choosing the display mode of the Overview Window. The setting is stored in the environments.
shortcut: d
see also: Main Window



Current Frame

If activated, the current frame of the currently selected sequence or reference frame is displayed like in the Zoom Window. Depending on settings of the other toggle buttons of the View menu, the dummy objects, point names, FOV and the distortion grid are displayed as well. Below the currently displayed image, its filename is indicated.
shortcut: f

Preview Movie

If activated, the preview movie frame associated to the currently selected frame is displayed in the Overview Window's display. If the respective preview frame has not yet been created, it remains black. Above the preview frame the label PREVIEW MOVIE is displayed. The remaining toggle buttons of the View menu have no effect on the display. The Overview Window must be at least as big as the preview movie frames, otherwise they cannot be displayed, and a warning message is shown.
shortcut: p
see also: Main Window, Preview Settings

Cache Movie

If activated, the cache movie frame associated to the currently selected frame is displayed in the Overview Window's display. If the respective cache frame has not yet been created, it remains black. Above the preview frame the label CACHE MOVIE is displayed. The remaining toggle buttons of the View menu have no effect on the display. The Overview Window's size must be above the size of the cache movie frames, otherwise they cannot be displayed and a warning message is shown.
shortcut: c
see also: Main Window, Preview Settings

Show Cache while Dragging

If activated, the content of the cache movie is displayed in the Overview Window while the user drags the frame slider of the Main Window, regardless of the settings in the Display menu. As soon as the user releases the left mouse button, the display switches back according to the original setting of the Display menu. The setting of this togglebutton is stored in the environments.
shortcut: a
see also: Main Window

Show Dummies

If activated, the dummies of all available pointgroups are displayed in the Overview Window display. The setting of this toggle button is stored in the environments.
shortcut: u
see also: Orientation Window

Show Point Names

If activated, the point names and the accompanying information text are displayed in the Overview Window display beside the screen points. The setting of this toggle button is stored in the environments.
shortcut: n

Show FOV

If activated, the field of view borders are represented in the Overview Window display as dotted lines. The setting of this toggle button is stored in the environments.
shortcut: f

Show Distortion Grid

If activated, the lens distortion setting of the camera linked to the currently selected sequence or reference frame is visualized in the Overview Window by the use of the distortion grid. The setting of this togglebutton is stored in the environments.
shortcut: g
see also: Zoom Window, Camera Adjustment Window

Image Controls Window

This window allows you to modify the color representation of the sequence and reference frames in the Overview- and the Zoom Window. It does not influence the preview movies or the representation of the dummies. The settings are taken into account when screen points are motiontracked. The features may be used for increasing contrast, to eliminate irrelevant or disturbing areas around the screen point being tracked. There are two different methods applied to the image, namely image control curves and color contrast. If both are employed, the software applies image control curves first and color contrast afterwards as shown in the figure below.

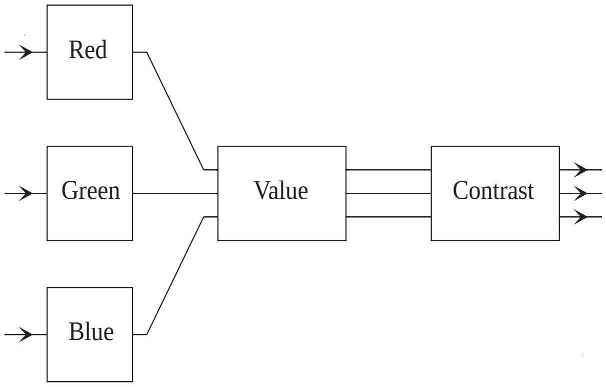


Image control diagram

Image Control Curves

There are 4 different types of curves available, namely Value (which means brightness), Red, Green and Blue. The image control curves field allows the user to modify each set of pixels of an image by specific brightness and color. The horizontal axis specifies the brightness of the pixel being modified, the vertical axis specifies the resulting luminosity. Shifting a point to the upper bound of the field causes the part of the image specified by the selection on the horizontal axis to appear in white, shifting it to the lower bound causes the part to appear in black. Intermediate settings conserve the color tone as far as possible while modifying the contrast. The curves Red, Green and Blue work as follows: the horizontal axis specifies the brightness of the respective color channels. The vertical axis specifies the resulting brightness of the color channel after applying the respective curve.

Active

If activated, the settings of the window are applied to the Overview Window display and to the Zoom Window display. The image control settings are then taken into account when motion- or markertracking is being performed.

Edit

To be able to modify one or more control curve(s), the respective edit toggle(s) must be turned on. All editable control curves are represented as a solid line.

Color Contrast

This slider ranges from full color contrast (default setting, right position) to a black-and-white representation (left position). In calculating the luminosity all color channels - red, green, blue - are weighted equally.

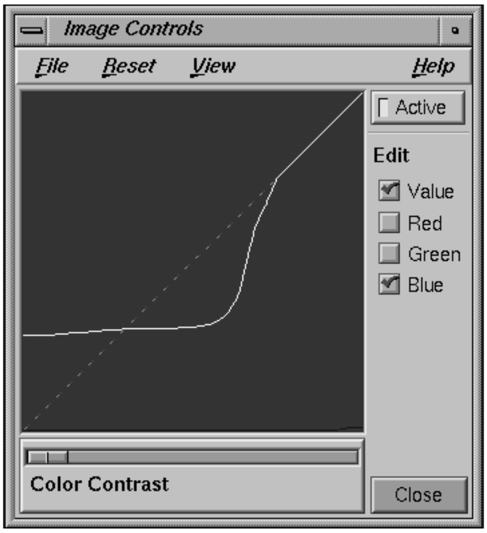
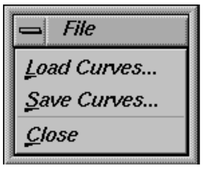


Image Controls Window

Menu: File



Contains functions for saving and loading image control curves.
shortcut: ALT-f

Load Curves...

Load an image control curve and a color contrast setting from a file. Selecting it opens a standard file requester Load Image Control Curve...
shortcut: l

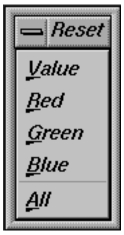
Save Curves...

Save an image control curve and a color contrast setting to a file. Selecting it opens a standard file requester Save Image Control Curve...
shortcut: s

Close

Closes the Image Control Window.
shortcut: c

Menu: Reset



Contains functions for resetting control curves to their default state.
shortcut: ALT-r

Value

Resets the Value control curve to its default state.
shortcut: v

Red

Resets the Red control curve to its default state.
shortcut: r

Green

Resets the Green control curve to its default state.
shortcut: g

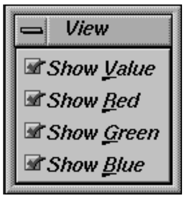
Blue

Resets the Blue control curve to its default state.
shortcut: b

All

All settings are reset to their default values, which leaves the image as it is, even if image control is active.
shortcut: a

Menu: View



View
☑ Show Value
☑ Show Red
☑ Show Green
☑ Show Blue

Contains toggles for showing/hiding control curves.
shortcut: ALT-v

Show Value

If activated, the Value control curve is displayed.
shortcut: v

Show Red

If activated, the Red control curve is displayed.
shortcut: r

Show Green

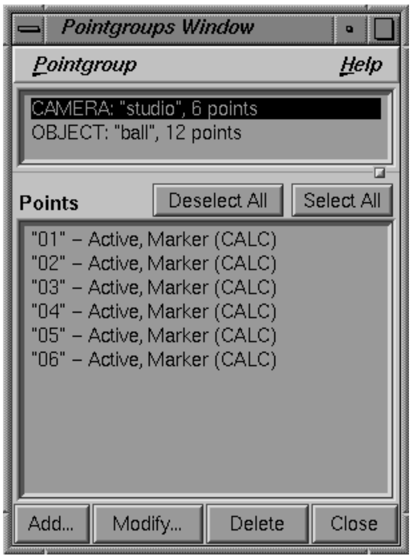
If activated, the Green control curve is displayed.
shortcut: g

Show Blue

If activated, the Blue control curve is displayed.
shortcut: b

Pointgroups Window

The Pointgroups Window provides functions for creating, modifying and deleting pointgroups and their associated points. It is mainly devided into two list widgets. The uppermost list widget contains a list of all available pointgroups of the current project. The pointgroup entry which is selected in this list is commonly called the “currently selected” pointgroup. It has a far-reaching influence on 3D-Equalizer’s windows and functions, because many functions are related to the “currently selected” pointgroup. The list widget below the pointgroup list widget contains a list of all points that are associated with the currently selected pointgroup. The list of points is sorted alphabetically. Double-clicking on a pointgroup entry opens the requester Modify Pointgroup for this pointgroup, whereas double-clicking on a point entry opens the requester Modify Point.



CAMERA: "studio", 6 points
OBJECT: "ball", 12 points

"01" – Active, Marker (CALC)
"02" – Active, Marker (CALC)
"03" – Active, Marker (CALC)
"04" – Active, Marker (CALC)
"05" – Active, Marker (CALC)
"06" – Active, Marker (CALC)

Pointgroups Window

Buttons

Add...

Opens a requester Create Point...

Modify...

Opens a requester Modify Point...

Delete...

Remove the selected points from the point list. The point's 2D tracking curve is deleted as well.

Select All

All points of the currently selected pointgroup become selected.

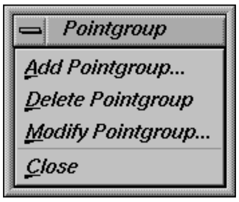
Deselect All

All points of the currently selected pointgroup become deselected.

Close

Closes the Pointgroups Window.

Menu: Pointgroup



Pointgroup
Add Pointgroup...
Delete Pointgroup
Modify Pointgroup...
Close

This menu provides functions for creating, modifying and deleting a pointgroup.
shortcut: ALT-P

Add Pointgroup...

Opens a requester Create Pointgroup.
shortcut: a
see also: Requester Modify Pointgroup

Delete Pointgroup

Deletes the currently selected pointgroup and all its points including their associated 2D tracking curves.
shortcut: d

Modify Pointgroup...

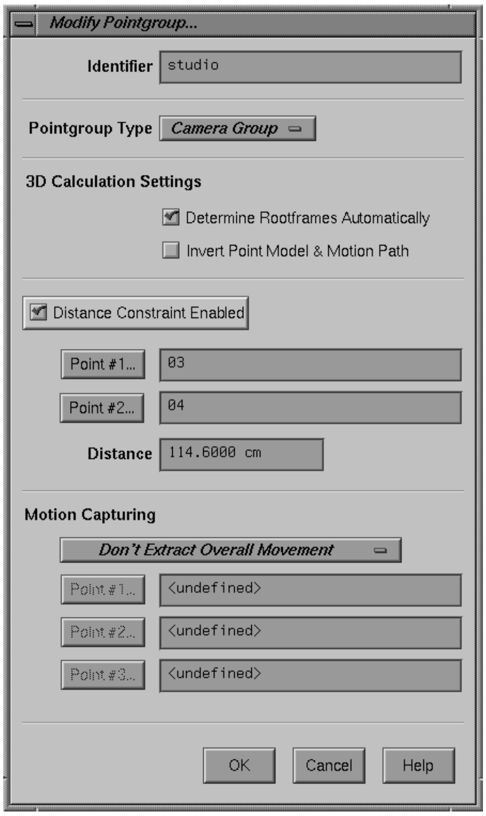
Opens a requester Modify Pointgroup.
shortcut: m
see also: Requester Create Pointgroup

Close

Closes the Pointgroups Window.
shortcut: c

Requester: Create Pointgroup...

Create a new pointgroup and enter its settings.
see also: Modify Pointgroup.



Modify Pointgroup...

Identifier: studio

Pointgroup Type: Camera Group

3D Calculation Settings

☑ Determine Rootframes Automatically

□ Invert Point Model & Motion Path

☑ Distance Constraint Enabled

Point #1...: 03

Point #2...: 04

Distance: 114.6000 cm

Motion Capturing

Don't Extract Overall Movement

Point #1...: <undefined>

Point #2...: <undefined>

Point #3...: <undefined>

Requester: Modify Pointgroup...

Identifier

The name of the pointgroup, by means of which it will be identified within 3D-Equalizer.

Pointgroup Type

This option menu defines the type of the pointgroup. It can either be a Camera Group, an Object Group, or a MoCap Group. A camera pointgroup is used to determine the movement of cameras in 3D space. There can either be one or no camera pointgroup in a 3D-Equalizer project. The points of a camera pointgroup most commonly tracked are situated in the static areas of a set, like the studio floor or walls. The camera motion of a given sequence is calculated in respect to the points of the camera pointgroup. If there is no camera pointgroup, all cameras remain fixed in 3D space.

An object pointgroup is used to determine the motion of a moving object, like the motion of a car or the movement of an actor's head. There is no limit to the number of object pointgroups within a 3D-Equalizer project. The 3D motion path of an object pointgroup of a given sequence is calculated in respect to the points of that object pointgroup, taking into account possible camera movements.

A mocap pointgroup is used to determine three dimensionally motion curves of individually moving points. Beside that it is similar to an object pointgroup.

Determine Rootframes Automatically

Each pointgroup has its own pair of rootframes. If this toggle is activated, 3D-Equalizer searches for two new rootframes, every time a new calculation is started. If this toggle is deactivated, the user can define a pair of rootframes manually using the respective controls in the Main Window.

see also: Main Window::Rootframes

Invert Point Model & Motion Path

If activated, 3D-Equalizer tries to invert the point group's point cloud and camera/object motion path, next time the calculation process is initiated.

see also: Main Window::Calc::Calc All Objects From Scratch

Distance Constraint Enabled

If activated, the scale of the respective pointgroup in 3D space is fixed and cannot be modified by the functions of the Orientation Window. A fixed distance between two points fo the pointgroup is used to define its scale in 3D space.

see also: Point #1, Point #2, Distance, Orientation Window

Point #1

Defines the first point which defines a distance constraint used in order to fix the pointgroup's 3D scale. Clicking on the button opens a requester with a list of all points associated to the pointgroup. By double-clicking on a point in this list, the respective point becomes "Point #1".

see also: Distance Constraint Enabled

Point #2

Defines the second point which defines a distance constraint used in order to fix the pointgroup's 3D scale. Clicking on the button opens a requester with a list of all points associated to the pointgroup. By double-clicking on a point in this list, the respective point becomes "Point #2".

see also: Distance Constraint Enabled

Distance

A textfield in which the user can define a distance between two different points of the pointgroup. This distance is used in order to fix the pointgroup's scale in 3D space.
see also: Distance Constraint Enabled

Motion Capturing

An option menu to define the global motion extraction mode of a mocap pointgroup.

Don't Extract Overall Movement

If activated, the points' motion curves will not be modified. The mocap pointgroup will remain static.

Extract Overall Movement Automatically

If activated, the motion curves of all points will be analysed. The motion that all points have in common will be extracted from them and transfered into the mocap pointgroup, instead. In contrast to the points, the pointgroup will become translated and rotated over time.

Extract Overall Movement Manually

If activated, the common motion of points #1 - #3 will be extracted from all points and transfered into the mocap pointgroup, instead. In contrast to the points, the pointgroup will become translated and rotated over time. To guarantee a good quality, points #1 - #3 should be static in respect to each other. In addition, they should not be located on a virtual line.
see also: Point #1, Point #2, Point #3

Point #1

Defines the first point which is used for manual extraction of overall movement. Clicking on the button opens a requester with a list of all points associated to the mocap pointgroup. By double-clicking on a point in this list, the selected point becomes "#1".
see also: Extract Overall Movement Manually

Point #2

Defines the second point which is used for manual extraction of overall movement. Clicking on the button opens a requester with a list of all points associated to the mocap pointgroup. By double-clicking on a point in this list, the selected point becomes "#2".
see also: Extract Overall Movement Manually

Point #3

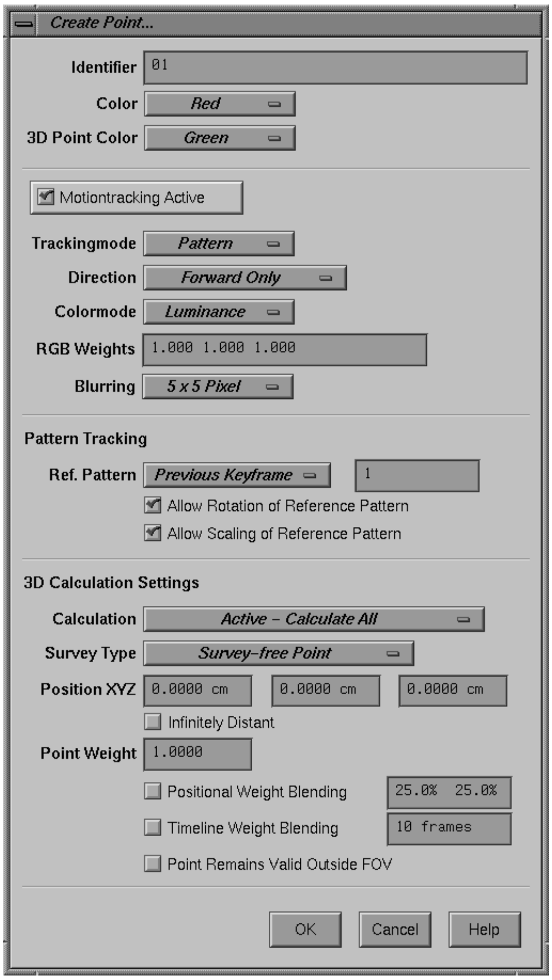
Defines the third point which is used for manual extraction of overall movement. Clicking on the button opens a requester with a list of all points associated to the mocap pointgroup. By double-clicking on a point in this list, the selected point becomes "#3".
see also: Extract Overall Movement Manually

Requester: Modify Pointgroup...

Allows the user to modify the settings of a pointgroup. The functions are the same as in the Create Pointgroup requester.

Requester: Create Point...

Create a new point associated to the currently selected pointgroup and enter its settings.
see also: Modify Point...



Create Point...

Identifier: 01

Color: Red

3D Point Color: Green

☑ Motiontracking Active

Trackingmode: Pattern

Direction: Forward Only

Colormode: Luminance

RGB Weights: 1.000 1.000 1.000

Blurring: 5 x 5 Pixel

Pattern Tracking

Ref. Pattern: Previous Keyframe 1

☑ Allow Rotation of Reference Pattern

☑ Allow Scaling of Reference Pattern

3D Calculation Settings

Calculation: Active – Calculate All

Survey Type: Survey-free Point

Position XYZ: 0.0000 cm 0.0000 cm 0.0000 cm

☐ Infinitely Distant

Point Weight: 1.0000

☐ Positional Weight Blending 25.0% 25.0%

☐ Timeline Weight Blending 10 frames

☐ Point Remains Valid Outside FOV

Requester: Create Point...

Identifier

The points must be given distinct names so that they can be identified uniquely. If the user does not specify a name, the program will create it automatically.

Color

Allows the user to select the color for the screen point representation in the Zoom Window and the Overview Window. Choose among *Red, Green, Blue, Yellow, Black, White, Gray*.

3D Point Color

Allows the user to select the reconstructed 3D point position's color in the Zoom- and Overview Window. Choose among *Red, Green, Blue, Yellow, Black, White, Gray*.

Motiontracking Active

If activated, the selected point is tracked next time the 3D-Equalizer motiontracking functions are applied. If a point should be excluded from motiontracking, this toggle should be deactivated.
see also: Zoom Window

Trackingmode

There are three different types of tracking, which are referred to as patterntracking, markertracking and cornertracking. Patterntracking can be used to track any type of structure found in the footage. Markertracking can be used in special situations, where a unicolored, point-symmetric spot is located on a flat, unicolored surface. Cornertracking can be used in special situations, where two straight edges of an unicolored object come together, forming a corner.

Direction

The direction of tracking. Choose among *Forward only, Backward only* and *Forward & Backward*.

Colormode

Defines the colormode for tracking. Choose either *Luminance* or *Chrominance.*

Luminance

Tracking is performed by recognizing the search pattern within the search area. When the colormode is set to *Luminance*, only the luminosity contrast is taken into account, but no color information. Usually, this mode is sufficient for tracking.

Chrominance

If colormode is set to *Chrominance*, all three color channels are employed for tracking.

RGB Weights

A textfield to configure the weight of each color channel used for tracking.

Blurring

An option menu to define a blurring filter that is applied to the tracking pattern. Choose among *Disabled*, *3x3 Pixel* and *5x5 Pixel.*

Ref. Pattern

An option menu to define how the reference pattern will be acquired, if patterntracking is activated. Choose among *Previous Frame*, *Previous Keyframe*, *First Keyframe* and *Expicit Frame.*

Allow Rotation of Reference Pattern

If activated, the reference pattern may be rotated by 3DE's tracking engine during the tracking process. This option is only taken into account if patterntracking mode is activated.

Allow Scaling of Reference Pattern

If activated, the reference pattern may be scaled separately in both dimensions by 3DE's tracking engine during the tracking process. This option is only taken into account if patterntracking mode is activated.

Calculation

Defines the handling of the point during the calculation process.

Off - Calculate Nothing

This causes, the point to be completely ignored during the calculation process.

Passive - Calculate Point Positions Only

If activated, the point is not used for calculating a 3D motion path at all. After the regular calculation process, only the point's 3D position is calculated and is shown in the Orientation Window.

Active - Calculate All

If activated, the point is used for the calculation of all 3D motion paths of its pointgroup.

Survey Type

Defines by which method 3D-Equalizer will compute the point's position in 3D space.

Survey-free Point

If activated, 3D-Equalizer will compute the selected point's position in 3D space completely from scratch.

Approximately Surveyed Point

If activated, 3D-Equalizer will compute the point's position in 3D space by means of user defined 3D coordinates. These coordinates can be specified in the textfields Position XYZ. During the calculation process 3D-Equalizer will modify the given 3D coordinates in order to further improve the overall calculation result.
see also: Position XYZ

Exactly Surveyed Point

If activated, 3D-Equalizer will compute the point's position in 3D space by means of user defined 3D coordinates. These coordinates can be specified in the textfields Position

XYZ. During the calculation process 3D-Equalizer will NOT modify the given 3D coordinate.
see also: Position XYZ

Position XYZ

Three textfields that contain a user definable coordinate in 3D space.
see also: Survey Type

Point Weight

A textfield that contains a parameter which makes a point more or less important in respect to other points. The higher the weight of a point the more important it will become. 3D-Equalizer will reduce the individual pixel deviations of points with a higher weight more, than for points with a lower weight.
see also: Status Window, Positional Weight Blending, Timeline Weight Blending

Infinitely Distant

If activated, 3D-Equalizer assumes the point to be very far away from the camera, like the moon or the horizon. During the calculation process an “infinitely distant” point is only used to compute the camera rotations, it doesn’t have any effect on the camera positions. “Infinitely distant” points only make sense when used in camera pointgroups.

Positional Weight Blending

If activated, the point’s weight will become dependent from its 2D tracking position. If the point is located at the border of the FOV, its weight will be reduced nearly to zero in that particular frame. However, if it is located in the middle of the FOV its weight will be equal to the point’s weight setting.
see also: Point Weight

Timeline Weight Blending

If activated, the point’s weight will be smoothly faded from zero to the full weight setting each time the 2D tracking point becomes visible. If the 2D tracking point becomes invalid (invisible) the point’s weight is smoothly faded down to zero.
see also: Point Weight

Point Remains Valid Outside FOV

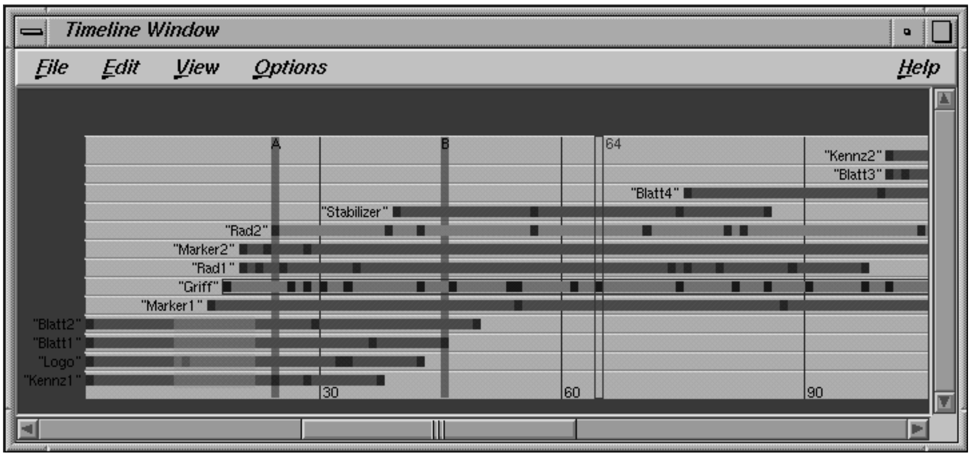
If activated, the point’s 2D tracking information is taken into account during the calculation process, even if it is located outside the field of view.

Requester: Modify Point...

Enables the user to modify the settings of the currently selected points. The functions are the same as in the requester Create Point...

Timeline Window

3D-Equalizer's Timeline Window provides a schematic overview of the 2D tracking curves of certain points. The window mainly consists of a graphical display area in which the user can examine and edit parts (or entire) 2D tracking curves of one or more points.



Timeline Window

Miscellaneous

Display

The Timeline Window gives an overview of tracking curves of the points of the currently selected pointgroup tracked in the currently selected sequence. The x-axis of the display area represents frames, whereas the y-axis represents the various points. Each tracking curve is represented by a horizontally oriented bar. In unselected state, a tracking curve is shown as various shades of gray. An entirely selected tracking curve is shown as shades of blue, and a selected curve segment is shown as shades of orange. Userdef frames within a tracking curve are shown as a dark, nearly black, shaded rectangle. Tracked frames within a tracking curve are shown as a medium shaded rectangle. Obsolete, splined, tracked-forward and tracked-backward frames are shown as a lighter shaded rectangle. Invalid frames are drawn in a dotted line. The name of a point belonging to its particular tracking curve is drawn left beside its first valid frame. The separate tracking curves are located one above another on a white workarea. They are sorted automatically. Tracking curves which start on low frame numbers are located below those tracking curves which start on higher frame numbers. The height of the white workarea represents the number of available points found in the currently selected pointgroup, whereas the width of the workarea represents the number of frames of the currently selected sequence. Selecting another pointgroup or sequence will cause the Timeline Window to update its display area immediately. Beside each 2D tracking curve, the currently selected frame and the rootframes of the selected pointgroup are displayed. The currently selected frame is displayed as a red outline, which covers the entire workarea from bottom to the top. The current frame number is displayed on the uppermost right hand side of the outline. The rootframes of the currenty selected pointgroup are represented by green rectangles that go from bottom to the top of the workarea. They are marked at the top with the letter "A" or "B", which either stands for Rootframe A or Rootframe B. The Syncframe of a synchronized sequence (mocap project) is represented by a yellow, vertical bar which is marked with the letter "S". Finally, the workarea uses red to indicate those frames that do not contain enough valid screen points for a proper calculation.

Selection and user interaction

The 2D tracking curves displayed in the Timeline Window can also be edited. For this purpose, various different selections can be applied. By clicking on a 2D tracking curve, the respective point and its tracking curve becomes selected. The tracking curve's colour is changed from grey to blue. All formerly selected tracking curves and curve segments become unselected. By left-clicking on a tracking curve, while holding down the SHIFT-key multiple 2D tracking curves become selected. All formerly selected tracking curves and curve segment stay selected. Left-clicking on a selected tracking curve while holding down the SHIFT-key, will deselect it. Some functions only apply to parts of tracking curves, which are called curve segments. In order to select a curve segment, the user must hold down the ALT-key and click-drag the left mouse button on that part of the tracking curve to be selected. All formerly selected tracking curves and curve segments become unselected. Selected curve segments are shown in an orange color. To select multiple curve segments, the user must hold down the SHIFT-key and the ALT-key while click-dragging the left mouse button on a tracking curve. Lastly, the current frame can be changed by holding down the CTRL-key while clicking into the workarea with the left mouse button. Holding down the middle mouse button while dragging the mouse transposes the workarea two-dimensionally within the Timeline Window's display. Holding down the right mouse button while dragging the mouse scales the workarea horizontally according to the mouse movement.

Keys

CURSOR-LEFT and CURSOR-RIGHT are used for jumping back and forth in the currenty selected sequence according to the setting of Frame Keyboard Grid. CTRL-CURSOR-LEFT and CTRL-CURSOR-RIGHT are used for going back and forth by one frame. CURSOR-UP and CURSOR-DOWN are used for selecting points. PAGE-

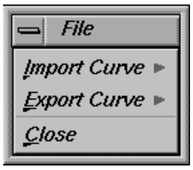
UP/PAGE-DOWN jumps to the previous/next keyframe of the currently selected point. HOME/END jumps to the first/last sequence or reference frame.

Sliders

The sliders below and on the right hand side area are used to transpose the workarea horizontally (or vertically) within the Timeline Window's diaplay area.

Menu: File

Contains functions for saving and loading 2D tracking curves.
shortcut: ALT-f



Import Curve

Loads 2D motion curves.
shortcut: i

Raw

Opens a file requester. The motion curve of the currently selected point is replaced by the data found in the specified file. It must be in 3D-Equalizer raw motion curve format, a binary format which holds all tracking curve information including tracking areas and status information.
shortcut: r
see also: Export Curve

Flame

Opens a file requester. The motion curve of the currently selected point is replaced by the data found in the specified file. It must be in Discreet Logic Flame Raw (Track) format.
shortcut: f

Cineon

Opens a file requester. The motion curve of the currently selected point is replaced by the data found in the specified file. It must be in Kodak Cineon Raw format.
shortcut: c

Export Curve

Saves 2D motion curves.
shortcut: e

Raw

Opens a file requester. The data of the motion curve for the currently selected point is saved into the specified file, using the 3D-Equalizer raw motion curve format. The 3D-Equalizer raw motion curve format is a binary format which holds all tracking curve

information including tracking areas and status information.This function can be used to share motion curves between different projects.
shortcut: r
see also: Import Curve

Ascii

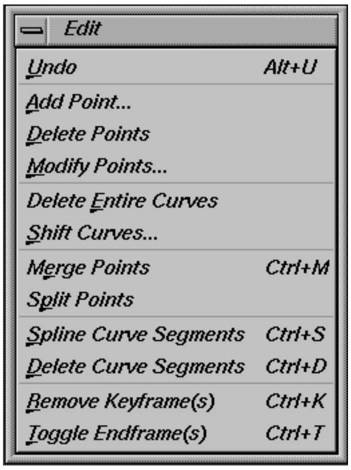
Opens a file requester. The data of the motion curve for the currently selected point is saved into the specified textfile.
shortcut: a

Close

Closes the Timeline Window.
shortcut: c

Menu: Edit

Functions for creating, deleting and modifing points, entire tracking curves and curve segments.
shortcut: ALT-e



Undo

Undo the previous action.
shortcut: u, ALT-u

Add Point...

Creates a new point in the currently selected pointgroup. For this purpose a requester Create Point is opened.
shortcut: a
see also: Pointgroups Window

Delete Points

Deletes all points whose tracking curves are currently selected in the Timeline Window.
shortcut: d

Modify Points...

The settings of the point currently selected in the Pointgroups Window can be modified. For this purpose a requester Modify Point opens.
shortcut: m

Delete Entire Curves

All selected tracking curves are deleted. The points belonging to these tracking curves are NOT deleted.
shortcut: e

Shift Curves...

Open a requester Shift Curves.
shortcut: s

Merge Points

The tracking curves of the currently selected points are merged, creating a single tracking curve. This tracking curve is copied into a new point object and all formerly selected points become deleted.
shortcuts: e, CTRL-m

Split Points

This function only affects the currently selected curve segments. All frames that are located within the selected curve segments are removed from their tracking curves. The removed curve segments are copied into newly created point objects.
shortcut: p

Spline Curve Segments

This function only affects the currently selected curve segments. All frames that are located within the selected curve segments and that are not userdef will be splined. All userdef frames remain unchanged.
shortcuts: s, CTRL-s

Delete Curve Segments

This function only affects the currently selected curve segments. All frames that are located within the selected curve segments are removed from the respective tracking curves. After invoking this function, the formerly selected areas of the tracking curves will be invalid.
shortcuts: d, CTRL-d

Remove Keyframe(s)

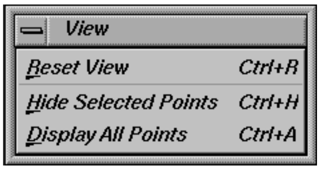
This function only affects the currently selected curve segments. All userdef frames, that are located within the selected curve segments will be converted to non-userdef frames.
shortcuts: r, CTRL-k

Toggle Endframe(s)

This function only affects the currently selected curve segments. All userdef frames that are located within the selected curve segments will be inverted. That means, userdef startframes become userdef endframes, and vice versa.
shortcuts: t, CTRL-t

Menu: View

Functions for changing the Timeline Window’s display area appearance.
shortcut: ALT-v



Reset View

Resets the Timeline Window’s workarea, so that it fills out the display area entirely.
shortcuts: r, CTRL-r

Hide Selected Points

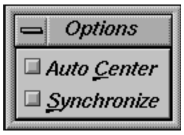
The currently selected tracking curves are hidden from the workarea. The workarea height is not adjusted, but the tracking curves are rearranged.
shortcuts: h, CTRL-h

Display All Points

Shows all points that have been formerly hidden using the Hide Selected Points function.
shortcuts: d, CTRL-A

Menu: Options

Toggle buttons to configure the Timeline Window’s behaviour.
shortcut: ALT-o



Auto Center

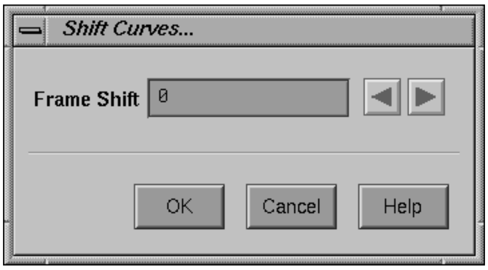
If activated, the Timeline Window’s workarea is automatically transposed so that the currently selected frame and the currently selected point are located at the center of the display area every time the current frame changes.
shortcut: c

Synchronize

If activated, 3D-Equalizer is forced to automatically synchronize the horizontal scale and horizontal position of the workareas of all windows that have the synchronize mode activated.
shortcut: s
see also: Status Window, Camera Zooming Window, Motion Path Window

Requester: Shift Curves...

A requester to move all selected tracking curves horizontally on the Timeline Window's workarea "back and forth through time".



Requester: Shift Curves...

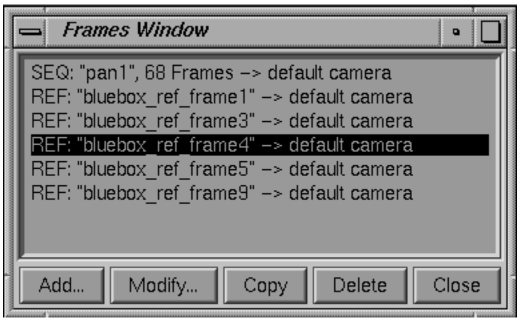
Frame Shift

A textfield that displays the current offset in frames. A value smaller than zero means moving the tracking curves on the workarea to the left ("back in time"). A value greater than zero means moving the tracking curves on the workarea to the right ("forward in time").

Reference Manual

Frames Window

A window for handling image sequences and single reference frames. The Frames Window mainly consists of a list widget. An entry in this list represents either an image sequence or a reference frame object. The entries in this list are sorted automatically, so that all sequences are listed above the reference frames. Clicking on an entry with the left mouse button selects it, and the respective sequence or reference frame becomes the “current” one. This has a far-reaching influence on 3D-Equalizer, because many functions relate to the “currently selected” sequence or reference frame. Sequence entries start with the label “SEQ:” followed by the sequence’s name, its number of frames and the name of the camera object, which is linked to it. Reference frame entries start with the label “REF:” followed by the reference frame’s name and the name of the camera which is linked to it.



SEQ: "pan1", 68 Frames -> default camera
REF: "bluebox_ref_frame1" -> default camera
REF: "bluebox_ref_frame3" -> default camera
REF: "bluebox_ref_frame4" -> default camera
REF: "bluebox_ref_frame5" -> default camera
REF: "bluebox_ref_frame9" -> default camera

Frames Window

Buttons

Add...

Creates a new sequence or reference frame. A requester opens asking the user to decide whether to create a sequence object or a reference frame object. If the user decides to create a sequence, the requester Create Sequence opens. If the user decides to create a reference frame, the requester Create Reference Frame opens.

Modify...

Edits the settings of the currently selected sequence or reference frame. The corresponding requester Modify Sequence or Modify Reference Frame then opens.

Copy

Creates a new reference frame and copies the contents of the currently selected frame into it.

Delete

The currently selected sequence or reference frame will be deleted from the current project. All associated tracking data will be lost.

Close

Closes the Frames Window.

Requester: Create Sequence...

To create a new sequence object and edit its initial settings.

Identifier

A textfield showing the name of the sequence.

Sequence Settings

Settings for defining the parameters of the images associated with the sequence object.

Browse...

Opens a standard file requester. By choosing the first image file of an image sequence, all necessary information such as start index, end index and filename pattern are automatically acquired, and the respective textfields are filled in.

Pattern

Enter here the file name pattern which matches the names of all sequence frames. For example, if the frame filenames are myimage.0100 ... myimage.0300 the pattern may be myimage.####.

Start, End, Step

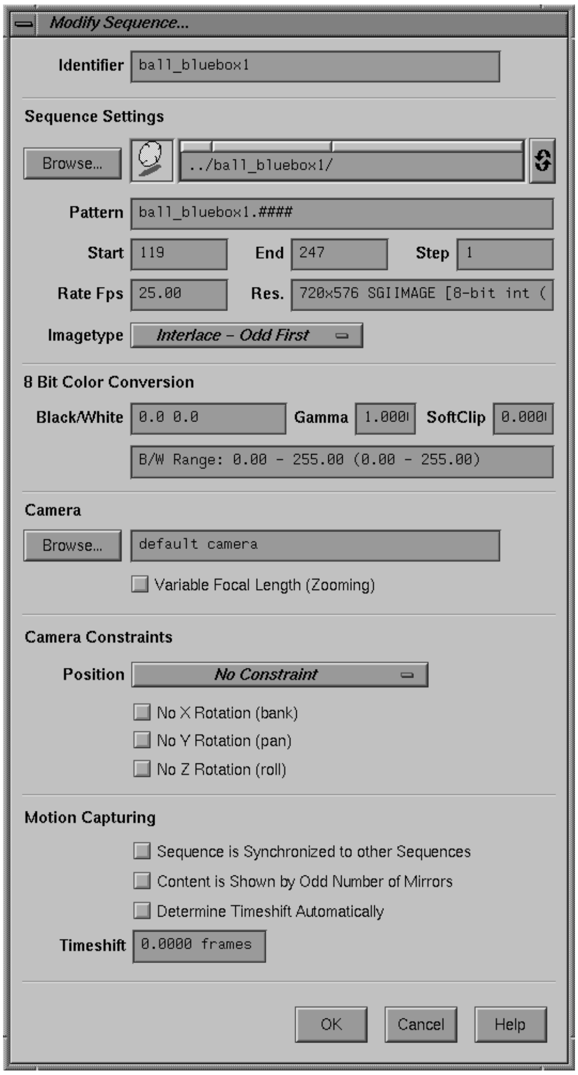
Specify the first and the last frame file and the step size between the frames. If the parameter is set to 1 each frame is used. Example: assume the frame filenames are myimage.0100 ... myimage.0300. If the parameter Step is set to 10, only the frames myimage.0100, my image.0110, myimage.0120, ... are used.

Rate Fps

In this textfield, the speed in frames per second should be entered. It is required by the export function and also for generating a preview movie.

Res.

Displays the resolution and image format of the sequence frames.



Modify Sequence...

Identifier: ball_bluebox1

Sequence Settings

Browse... ../ball_bluebox1/

Pattern: ball_bluebox1.####

Start: 119 End: 247 Step: 1

Rate Fps: 25.00 Res.: 720x576 SGIIMAGE [8-bit int (

Imagetype: Interlace – Odd First

8 Bit Color Conversion

Black/White: 0.0 0.0 Gamma: 1.000 SoftClip: 0.000

B/W Range: 0.00 - 255.00 (0.00 - 255.00)

Camera

Browse... default camera

☐ Variable Focal Length (Zooming)

Camera Constraints

Position: No Constraint

☐ No X Rotation (bank)

☐ No Y Rotation (pan)

☐ No Z Rotation (roll)

Motion Capturing

☐ Sequence is Synchronized to other Sequences

☐ Content is Shown by Odd Number of Mirrors

☐ Determine Timeshift Automatically

Timeshift: 0.0000 frames

Requester: Modify Sequence...

Image Type

Choose the image type used in the sequence: Noninterlace, Interlace-Even First, Interlace-Odd First, Field-Aspect 1:2, Anamorphic 2:1. Interlaced frames at a rate of 25 frames per second contain two fields displaced by 1/50 second. In the case of interlaced frames, this selector specifies which of the fields (even or odd) is used for representing the sequence frame within the project.

8 Bit Color Conversion

Settings to control 3D-Equalizer's behaviour when converting imported images into its internal imagebuffer format (8 bit per color channel).

Black/White

Defines the black- and white level of the imported image(s).

Gamma

Defines the gamma correction value for converting the imported image(s). Increasing this parameter will help you to enhance the image's dark areas.

SoftClip

Defines the softclip value for converting the imported image(s). Increasing this parameter will help you to enhance the image's bright areas.

Camera

Settings with further information about the camera in shooting the sequence frames.

Browse...

Opens a requester containing a list of all available camera objects for the current project. By double-clicking on a camera entry, the user links the respective camera object to the sequence. That means, 3D-Equalizer will use the settings of this camera for calculating the 3D data of the pointgroups to be tracked in this sequence. The name of the camera object is shown in the textfield beside the Browse button.
see also: Camera Adjustment Window

Variable Focal Length (Zooming)

If activated, the focal length of the camera linked to the sequence will assumed to be variable and can be edited and calculated in the Camera Zooming Window.
see also: Camera Zooming Window

Camera Constraints

Options that force 3D-Equalizer's calculation core to keep the camera animation within defined limits. This feature only has an influence, if a Camera pointgroup is available.

Position

Option menu to define constraints that have an effect on the camera path.

No Constraint

If activated, the camera path will not be limited in any dimension.

Fixed Camera Position Constraint

If activated, 3D-Equalizer assumes the real camera used to shoot the sequence did not change its position in 3D space during the shot. If a camera pointgroup exists, the reconstructed virtual camera will have a constant position in 3D space, and only its rotational attributes are animated. Unless no additional reference frames are introduced, the 3D point positions of this camera pointgroup will have no valuable z-depth information.

Plane Movement Constraint

If activated, the camera path will be forced to lie on a virtual plane. The orientation and position of this plane will be automatically computed by 3D-Equalizer.

Line Movement Constraint

If activated, the camera path will be forced to lie on a virtual line. The orientation and position of this line will be automatically computed by 3D-Equalizer.

Sphere Movement Constraint

If activated, the camera path will be forced to lie on the surface of a virtual sphere. The position and size of this sphere will be automatically computed by 3D-Equalizer.

Circle Movement Constraint

If activated, the camera path will be forced to lie on a virtual circle. The orientation, position and size of this circle will be automatically computed by 3D-Equalizer.

No X Rotation (bank)

If activated, the camera's rotational x components (rot x) will be constrained to zero. That means, the camera will not bank during the sequence.

No Y Rotation (pan)

If activated, the camera's rotational y components (rot y) will be constrained to zero. That means, the camera will not pan during the sequence.

No Z Rotation (roll)

If activated, the camera's rotational z components (rot z) will be constrained to zero. That means, the camera will not roll during the sequence.

Motion Capturing

Options to enable and configure motion capturing mode for the sequence.

Sequence is Synchronized to other Sequences

If activated, motion capturing mode is enabled for the sequence.

Content is Shown by Odd Number of Mirrors

If activated, points tracked in the sequence have to be reflected from an odd number of mirrors. This feature can be used to setup motion capturing projects just using a single camera and at least one mirror.

Determine Timeshift Automatically

If activated, 3DEqualizer will be allowed to compute the subframe timeshift of the sequence.
see also: Main Window::Calc::Calc MoCap Timeshift

Timeshift

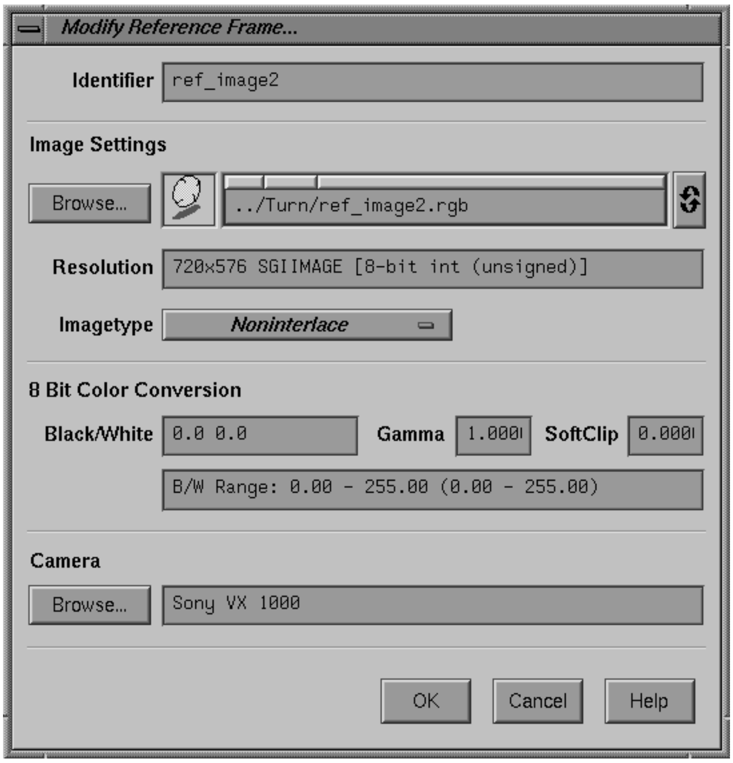
A textfield to define the subframe timeshift of the sequence.

Requester: Modify Sequence...

Allows the user to modify the settings of a sequence. The functions are the same as in the Create Sequence requester.

Requester: Create Reference Frame...

A requester to create and edit the initial settings of a reference frame.


Modify Reference Frame...

Identifier ref_image2

Image Settings

../Turn/ref_image2.rgb

Resolution 720x576 SGIIMAGE [8-bit int (unsigned)]

Imagetype Noninterlace

8 Bit Color Conversion

Black/White 0.0 0.0 Gamma 1.000 SoftClip 0.000

B/W Range: 0.00 - 255.00 (0.00 - 255.00)

Camera

Sony VX 1000

Requester: Modify Reference Frame...

Identifier

The name of the reference frame.

Image Settings

Settings defining the parameters of the image associated with the reference frame object.

Browse...

Opens a standard file requester which allows the user to choose the image file that should be associated to the reference frame object.

Resolution

Displays the resolution and image format of the reference frame. If the subload property is supported, the label SUB LOAD is issued as well.

Image Type

Choose the image type used for the reference frame: Noninterlace, Interlace-Even First, Interlace-Odd First, Field-Aspect 1:2, Anamorphic 2:1. Interlaced frames at a rate of 25 frames per second contain two fields displaced by 1/50 second. In the case of interlaced frames, this selector specifies which of the fields (even or odd) is used for representing the reference frame within the project.

8 Bit Color Conversion

Settings to control 3D-Equalizer’s behaviour when converting imported images into its internal imagebuffer format (8 bit per color channel).

Black/White

Defines the black- and white level of the imported image(s).

Gamma

Defines the gamma correction value for converting the imported image(s). Increasing this parameter will help you to enhance the image’s dark areas.

SoftClip

Defines the softclip value for converting the imported image(s). Increasing this parameter will help you to enhance the image’s bright areas.

Camera

Settings with further information about the camera used to shoot the reference frame.

Browse...

Opens a requester containing a list of all available camera objects in the current project. By double-clicking on a camera entry, the user links the respective camera object to the reference frame. That means 3D-Equalizer will use the settings of this camera for calculating the 3D data of the pointgroups, to be set in this reference frame. The name of camera object is shown in the textfield beside the Browse button.
see also: Camera Adjustment Window

Requester: Modify Reference Frame...

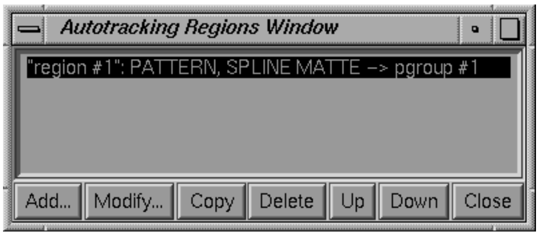
Allows the user to modify the settings of a reference frame. The functions are the same as in the Create Reference Frame requester.

Autotracking Regions Window

A window for handling autotracking region objects. The Autotracking Regions Window mainly consists of a list widget. Each entry in this list represents an autotracking region. An autotracking region usually defines a two dimensional area of the FOV, inside of which 3DEqualizer tracks points during the automatic tracking procedure. A two dimensional area can be defined by a matte image sequence or by a spline area matte or it can simply be the entire FOV. The list of autotracking regions, currently displayed in the Autotracking Regions Window, is associated to the currently selected sequence. That means, each sequence object has its own set of autotracking regions.

The entries in this list can be sorted manually by the user. Regions which are located on the top of the list have a higher priority than regions which are located at the bottom. Real world objects, located in the foreground of the footage, should be tracked by autotracking regions with a high priority, whereas real world objects, located in the background of the footage, should be tracked by autotracking regions with a low priority.

Clicking on an entry with the left mouse button selects it, and the respective autotracking region becomes the “current” one. An entry starts with the region’s name, followed by its tracking- and mattemode. Finally, the name of the pointgroup object into which points will be tracked is displayed.



"region #1": PATTERN, SPLINE MATTE -> pgroup #1

Autotracking Regions Window

Buttons

Add...

Creates a new autotracking region object. The requester Create Autotracking Region opens.

Modify...

The settings of the currently selected autotracking region object can be edited. For this purpose the requester Modify Autotracking Region opens.

Copy

A new autotracking region will be created and the settings of the currently selected autotracking region will be copied into it.

Delete

The currently selected autotracking region object will be deleted.

Up

The currently selected autotracking region object will be moved up by one position.

Down

The currently selected autotracking region object will be moved down by one position.

Close

Closes the Autotracking Regions Window.

Requester: Create Autotracking Region...

A requester to create a new autotracking region object and edit its initial settings.

Identifier

Textfield to edit the autotracking region's name.

Tracking Mode

An option menu to specify the autotracking engine which is going to be used for tracking the points of the region.

Don't Track Anything At All

If activated, 3DEqualizer will be forced to track no points in the area which is defined by the autotracking region.

Pattern Tracking

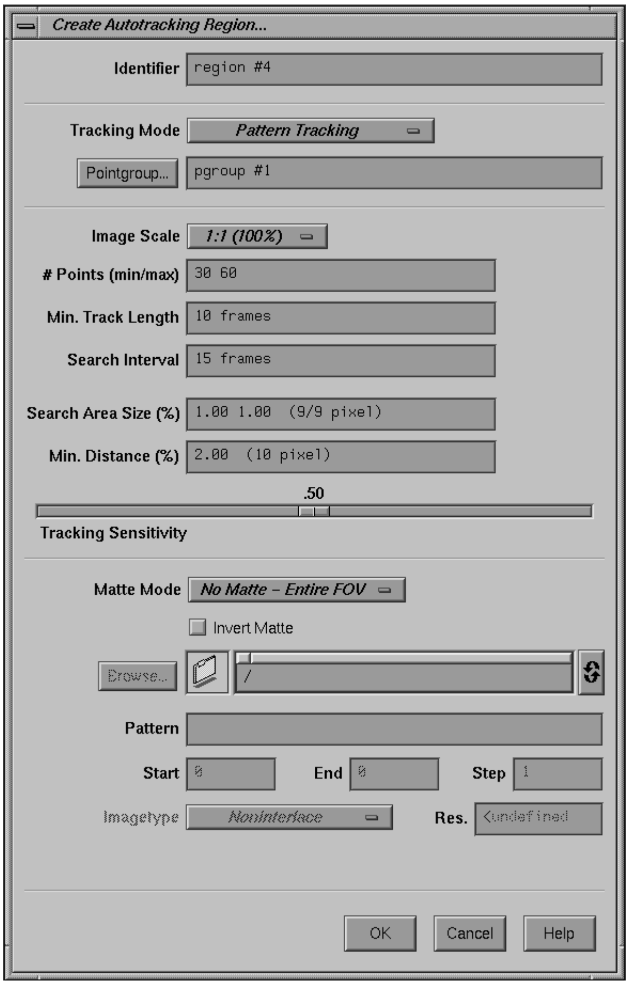
If activated, 3DEqualizer will track points using patterntracking techniques.

Marker Tracking

If activated, 3DEqualizer will track points using markertracking techniques.

Pointgroup...

Opens a requester containing a list of all available pointgroup objects for the current project. By double-clicking on a pointgroup entry, the respective pointgroup object is linked to the autotracking region. That means, during the automatic tracking procedure 3D-Equalizer will put the points, created by the autotracking region, into that pointgroup. *see also*: Pointgroups Window



Create Autotracking Region...

Identifier: region #4

Tracking Mode: *Pattern Tracking*

Pointgroup...: pgroup #1

Image Scale: *1:1 (100%)*

# Points (min/max): 30 60

Min. Track Length: 10 frames

Search Interval: 15 frames

Search Area Size (%): 1.00 1.00 (9/9 pixel)

Min. Distance (%): 2.00 (10 pixel)

Tracking Sensitivity: .50

Matte Mode: *No Matte – Entire FOV*

□ Invert Matte

Browse...: /

Pattern:

Start: 0 End: 0 Step: 1

Imagetype: *Noninterlace* Res.: <undefined>

OK Cancel Help

Requester: Create Autotracking Region...

Image Scale

An option menu to reduce image size for speeding up the tracking process. This setting is used for all autotracking regions. Choose among *1:1 (100%)*, *1:2 (50%)*, *1:3 (33.33%)*, *1:4 (25%)*, *1:5 (20%)*.

# Points (min/max)

A textfield to configure the autotracking engine's behaviour. If in the region's area less than "(min)" active points are available, 3DEqualizer is forced to search for new features and create new points respectively. If 3DEqualizer searches for new features, it is forced to create only so many points that "(max)" number of active points become available in the region's area. Already existing (manually tracked) points are taken into account.

Min. Track Length

A textfield to configure the autotracking engine's behaviour. Points created by the autotracking engine that are tracked in less than "Min. Track Length" frames are automatically being deleted, because 3DE usually cannot benefit from them.

Search Interval

A textfield to configure the autotracking engine's behaviour. After every "Search Interval" number of frames, 3DEqualizer is forced to search for new features and create new points respectively.

Search Area Size (%)

A textfield to define the search area size of the points which are created by 3DEqualizer's autotracking engine.

Min. Distance (%)

A textfield to define the smallest allowed distance between two tracking points. Already existing (manually tracked) points are taken into account.

Tracking Sensitivity

A slider to configure the autotracking engine's behaviour. It defines, how soon a feature is declared to be lost. A value of 1.0 will force 3DEqualizer to lose features very soon, thus generating points with relatively high quality, but rather short tracking curves (low number of frames). A value of 0.0 will result in points with relatively low quality, but rather long tracking curves (high number of frames).

Matte Mode

An option menu to define which matte technique is going to be used for the autotracking region.

No Matte - Entire FOV

The autotracking region's area is equal to the FOV. That means, points will be tracked within the entire FOV.

Image Sequence Matte

The autotracking region's area will be define by a matte image sequence. 3DEqualizer will be forced to track points only in bright areas of the matte image files.

Spline Area Matte

The autotracking region's area will be defined by a spline area matte. A spline area matte is a user-defined area which can be animated in 3DE's Zoom Window using regular keyframing techniques.

Invert Matte

If activated, the autotracking region's matte becomes inverted.

Browse...

Opens a standard file requester. By choosing the first image file of a matte image sequence, all necessary information such as start index, end index and filename pattern are automatically acquired, and the respective textfields are filled in.

Pattern

Enter here the file name pattern which matches the names of all matte sequence files. For example, if the frame filenames are matte.0100 ... matte.0300 the pattern may be matte.####.

Start, End, Step

Specify the first and the last frame file index and the step size between the frames. If the parameter is set to 1 each frame is used. Example: assume the frame filenames are matte.0100 ... matte.0300. If the parameter Step is set to 10, only the frames matte.0100, matte.0110, matte.0120, ... are used.

Res.

Displays the resolution and image format of the matte sequence frames.

Image Type

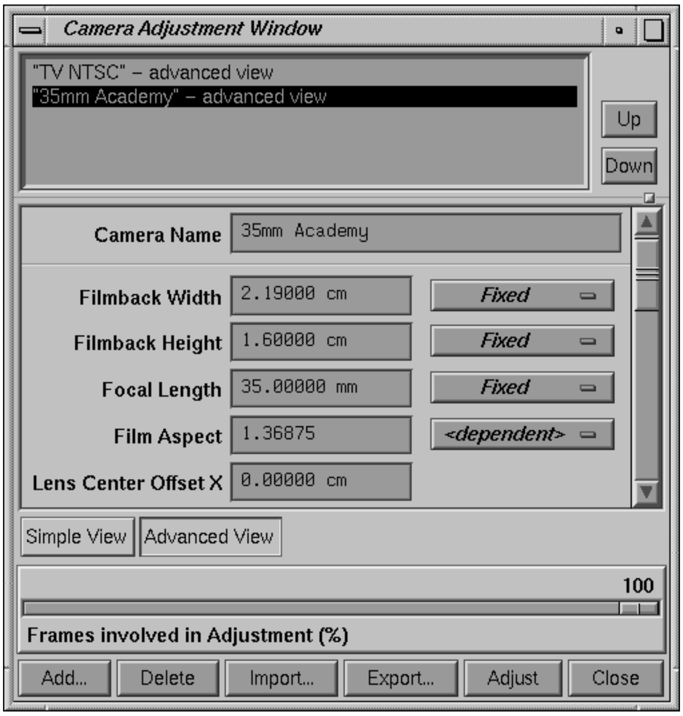
Choose the image type used in the matte image sequence: *Noninterlace*, *Interlace-Even First*, *Interlace-Odd First*. Interlaced frames at a rate of 25 frames per second contain two fields displaced by 1/50 second. In the case of interlaced frames, this selector specifies which of the fields (even or odd) is used for representing the matte sequence frame within the project.

Requester: Modify Autotracking Region...

Allows the user to modify the settings of an autotracking region object. The functions are the same as in the Create Autotracking Region requester.

Camera Adjustment Window

Provides the functionality for reconstructing and modelling the cameras used for recording the sequences and reference frames. The user can create, delete and edit the settings of multiple camera objects. In the upper part of the window there is a list widget which contains a list of all cameras which have been created within the current project. Below this list widget there are are several textfields which show the various parameters of the currently selected camera and the field of view of the sequences and reference frames linked to this camera. On the right hand side of some of the parameters there is a selector which can assume the values *Fixed*, *Fine Adjust*, *Unknown* and *<dependent>*. They specify whether the respective parameter is known, approximately known, unknown or if the parameter should be automatically derived from other parameters.



Camera Adjustment Window

"TV NTSC" – advanced view
"35mm Academy" – advanced view

Camera Name	35mm Academy	
Filmback Width	2.19000 cm	Fixed
Filmback Height	1.60000 cm	Fixed
Focal Length	35.00000 mm	Fixed
Film Aspect	1.36875	<dependent>
Lens Center Offset X	0.00000 cm	

Frames involved in Adjustment (%) 100

Camera Adjustment Window

If the selector of a parameter is set to *Fixed*, the text field color becomes pink and the user is able to enter a value. The parameter itself will not be modified when the system adjusts the camera geometry.

If a selector is set to *Fine Adjust*, the text field becomes pink as well and it is possible to enter a value. The corresponding parameter is modified in the adjustment procedure, where the system assumes that the entered value is approximately correct.

If a selector is set to *Unknown*, the text field color changes to pink-grey, a default value is automatically displayed with respect to the parameter type. The text field cannot be modifed by the user, and during the adjustment procedure a large parameter range is scanned.

All camera parameters (except Lens Distortion) are always kept consistent with regard to each other. There are always three parameters set to *Fixed*, *Fine Adjust* or *Unknown*. All other parameters are automatically set to <*dependent*>. The color of a <*dependent*> text field is set to grey, and it is not possible to enter a value. The value of a <*dependent*> parameter is immediately calculated by the system due to the non-<*dependent*> parameters.

Miscellaneous

Up, Down

These buttons are used to change the position of the currently selected camera in the camera list. Clicking on the Up button causes the currently selected camera to be moved up one position, whereas clicking on the Down button causes the currently selected camera to be moved down one position in the list of cameras. The order of the cameras in the list is important for the camera adjustment process. Important cameras should be placed at the top of the list, while less important cameras should be placed at the bottom of the list. A camera is important if there are many frames linked to it showing the points of a pointgroup from many different points of view. A camera is less important if only a few frames are linked to it, or if the frame show the points of a pointgroup from little different points of view.

Camera Name

The identifier of the currently selected camera.

Filmback Width, Filmback Height

Fields for displaying and modifying the current filmback width and height. The selector on the right of the fields is used to specify if or how precisely the values are known. The default length unit can be specified in the textfield Main Window::Options::3DE Preferences::Filmback Measure.

Focal Length

Field for displaying and modifying the current focal length. The selector on the right of the field is used to specify if or how precisely the value is known. The default length unit can be specified in the textfield Main Window::Options::3DE Preferences::Zoom Measure.

Film Aspect

Field for displaying and modifying the current film aspect. The selector on the right of the field is used to specify if or how precisely the value is known.

Lens Center Offset X, Lens Center Offset Y

Fields for displaying the Lens Center Offset X & Y parameters in respect to the settings of the Filmback textfields. Setting these values to (0.0, 0.0) will move the lens center axis exactly over the field of view's center.

Horizontal Angle, Vertical Angle

Fields for displaying the horizontal and vertical camera angles. The selector on the right of the field is used to specify if or how precisely the value is known.

Pixel Aspect

Field for displaying and modifying the pixel aspect. The selector on the right of the field is used to specify if or how precisely the value is known.

Fov Size

Field for displaying the field of view width and height. These values can only be modified by hand in the Zoom Window.

Lens Distortion

Field for displaying and modifying the parameter which describes the nonlinear distortion of the real camera. The selector on the right of the field is used to specify if or how precisely the value is known. Positive values compensate for barrel-type distortion, negative values compensate for pillow-type distortion.

Anamorphic Squeeze

Field for displaying the (horizontal) anamorphic squeeze factor of the camera lens system. A value of 1.0 means no anamorphic squeeze. A typical value for an anamorphic camera lens is 2.0.

Lens Curvature X, Lens Curvature Y

Fields for displaying special lens distortion parameters that only have an effect on a single screen dimension. Lens Curvature X models a separate lens distortion along the x axis of the screen, whereas Lens Curvature Y models a separate lens distortion along the y axis of the screen.

Quartic Distortion

Field for displaying and modifying the quartic lens distortion parameter, which has a simillar effect as lens distortion. The difference is that it deforms the outer parts of the image more than lens distortion does.

Lens D. Coefficients

Read-only textfield showing just another mathematical representation of the Lens Distortion, Anamorphic Squeeze and Curvature X / Y parameters.

Simple View

If activated, the Camera Adjustment Window will display only a reduced set of textfields for the currently selected camera.

Advanced View

If activated, all available textfields for the currently selected camera will be displayed.

Frames involved in Adjustment (%)

Enables the user to specify the number of frames being taken into account for adjusting the cameras. In order to reconstruct the camera geometry, many frames are required. Setting this slider to the right causes the program to take into account each frame, setting it to the left causes the program to use only 10% of the available frames for adjusting the camera.

Add...

Creates a new camera object and puts it at the bottom of the camera list. A newly created camera becomes the currently selected one.

Delete

Deletes the currently selected camera.

Import...

Opens a standard file requester for importing a formerly exported camera object. Exported cameras are normally stored in single files located in the directory “/usr/3dequalizer/user_data/camera_archive”. After a new camera has been imported it becomes the currently selected one and is added at the bottom of the camera list.

Export...

Opens a standard file requester. After the user has specified a filename, the settings of the currently selected camera are saved to disk. This feature can be used to build up a camera archive. Exported cameras are normally stored in the directory “/usr/3dequalizer/user_data/camera_archive”.

Adjust

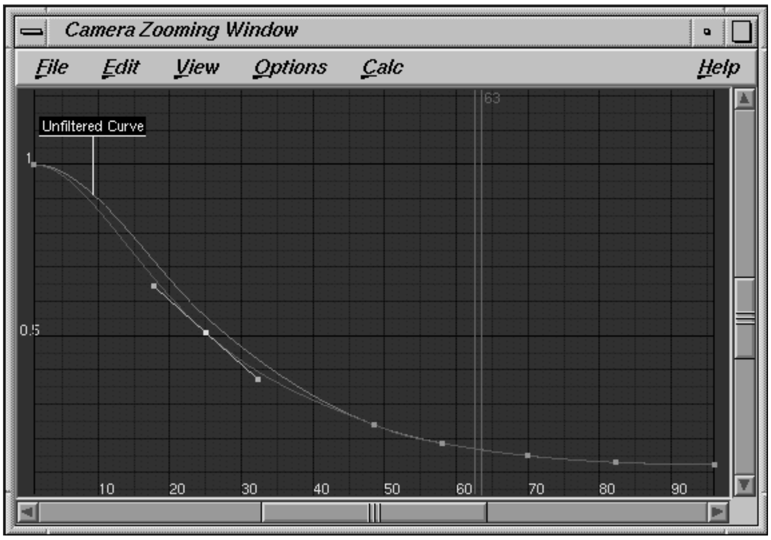
Starts the process of adjusting one or more cameras according to the settings of the selectors on the right of the fields containing the camera parameters. During the process, a requester containing a Stop button, a current and a best average pixel deviation value, are displayed. If the user hits the Stop button, the adjustment procedure is terminated, and the camera parameters of the best attempt are displayed in the text fields. If the user does not hit the Stop button, 3D-Equalizer calculates the best possible camera parameter values. This process may take from just a few minutes up to one or two hours. After the adjustment procedure, all non-<dependent> selectors are automatically set to Fixed.

Close

Closes the Camera Adjustment Window.

Camera Zooming Window

Provides functions for defining the change of focal length over time of a given sequence. The Camera Zooming Window mainly consists of a two-dimensional display area. If the currently selected sequence has the “Variable Focal Length (Zooming)” mode enabled (see requester Frames Window::Modify Sequence), the display area shows a 2D coordinate frame. The x-axis of this coordinate frame represents the number of frames of the currently selected sequence start at frame 1. The y-axis of the coordinate frame represents a zoom factor, starting from a value of 0. There are two curves, which represent the change of focal length over time. In order to determine the focal length in a particular frame, the zoom factor in this frame must be multiplied with the focal length setting of the camera that is linked to the currently selected sequence. The first, grey colored, curve is called the unfiltered curve. It cannot be edited by the user. The result of a zooming calculation is always stored as a reference in this curve. The second curve can either be red or blue colored. It can be edited by the user and therefore is called the edit curve. The edit curve represents the actual change of focal length over time, and is used for exporting the data and generating a preview movie. After a zoom calculation process, the unfiltered curve is automatically filtered using the current zoom filter settings (see requester Options::Zoom Filter Settings). Out of the resulting data of this filtering process, the edit curve is created with as few keyframes as possible. Right after that, the edit curve becomes blue. If the user edits the edit curve, it becomes red, indicating for the future that it has been modified manually. Finally, the currently selected frame is represented by an orange colored outline.



Camera Zooming Window

Selection and user interaction

The visible part of the Camera Zooming Window's coordinate frame can be changed in a way similarly to the Zoom Window. By holding down the middle mouse button while dragging the mouse, the coordinate frame can be moved around. By holding down the right mouse button while dragging the mouse, the coordinate frame is scaled separately in two dimensions. Pressing the CTRL-key, while holding down the right mouse button and dragging the mouse allows the user to draw a rubber box. After releasing the mouse button, the contents of the rubber box will be zoomed in, so that it fills out the entire display area. As in the Timeline Window, the user can change the currently selected frame by holding down the CTRL-key while left-clicking into the coordinate frame.

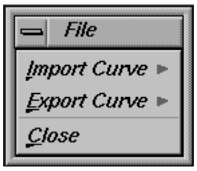
Editing the edit curve

The edit curve is a spline-based curve that consists of keyframes. Associated to each keyframe there are two tangents which define the behaviour of the curve at that keyframe. A keyframe is represented by a point, which is located directly on the edit curve. Unselected keyframes are drawn in white, whereas selected keyframes become yellow. The user is able to select a keyframe by clicking on it with the left mouse button. Keyframes that have been selected before, automatically become unselected. To keep or toggle the selection status of other keyframes, the SHIFT-key must be pressed while clicking with the left mouse button. Click-dragging the left mouse button will cause 3D-Equalizer to draw a rubber box, and all keyframes that are located within this rubber box become selected. All keyframes that have been previously selected become unselected, unless the SHIFT-key is pressed while dragging the rubberbox. In order to create a new keyframe, the ALT-key must be pressed while clicking with the left mouse button on the coordinate frame. Click-dragging a keyframe or the endpoint of a tangent will change its position.

Keys

CURSOR-LEFT and CURSOR-RIGHT are used for jumping back and forth in the currenty selected sequence according to the setting of Frame Keyboard Grid. CTRL-CURSOR-LEFT and CTRL-CURSOR-RIGHT are used for going back and forth by one frame. CURSOR-UP and CURSOR-DOWN are used for selecting points. PAGE-UP/PAGE-DOWN jumps to the previous/next keyframe of the currently selected point. HOME/END jumps to the first/last sequence or reference frame.

Menu: File



Functions for importing and exporting the edit curve.
shortcut: ALT-f

Import Curve::Raw

Opens a standard file requester. The selected file is interpreted as a 3D-Equalizer Raw Zoom Curve and replaces the current edit curve.
shortcut: i-r

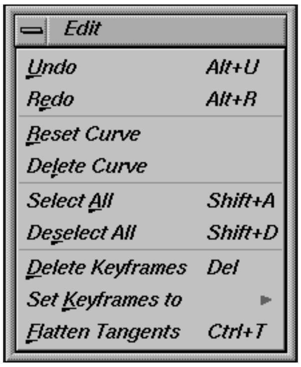
Export Curve::Raw

Opens a standard file requester. The current edit curve is written into the specified file in 3D-Equalizer’s Raw Zoom Curve format.
shortcut: e-r

Close

Closes the Camera Zooming Window.
shortcut: c

Menu: Edit



Functions for modifing the edit curve.
shortcut: ALT-e

Undo

Undo the last edit action.
shortcut: ALT-u

Redo

Redo the last undo action.
shortcut: ALT-r

Reset Curve

Resets the edit curve back to its last “blue state”. That means, the curve reverts to the status before being edited for the first time.
shortcut: r

Delete Curve

All keyframes of the edit curve are deleted and the curve is set to a default value of 1.
shortcut: l

Select All

All keyframes of the edit curve become selected.
shortcuts: a, SHIFT-a

Deselect All

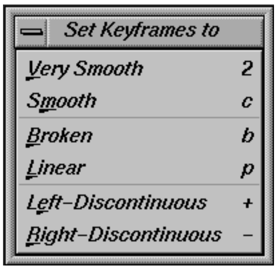
All keyframes of the edit curve become unselected.
shortcuts: s, SHIFT-d

Delete Keyframes

All selected keyframes will be deleted from the edit curve.
shortcuts: d, DELETE-key

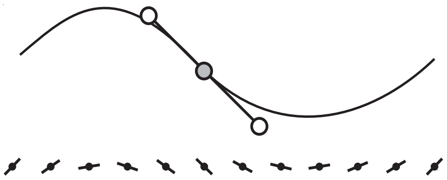
Set Keyframes to

Functions for modifying the behaviour of the selected keyframe’s tangents.
shortcut: k



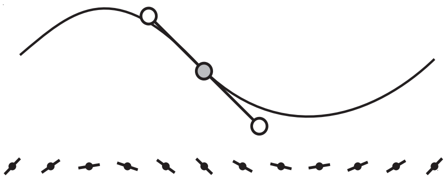
Very Smooth

If a keyframe is declared to be “Very Smooth”, the curve pieces left and right to the keyframe are glued together smoothly, and also the slopes of the curve pieces change smoothly in passing the keyframes. The curvature of the curve is well defined at the keyframe.
shortcuts: v, 2-key



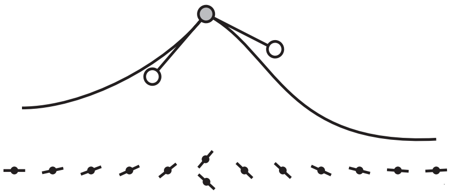
Smooth

If a keyframe is declared to be “Smooth”, the curve pieces left and right to the keyframe are glued together smoothly, and the slopes of the curve pieces change continously (however not smoothly) in passing the keyframe. This is the default behaviour for newly created keyframes.
shortcuts: m, c-key



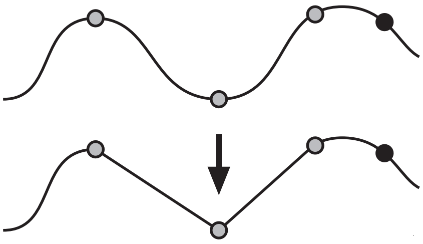
Broken

If a keyframe is declared to be “Broken”, the curve pieces are glued together continuosly (however not smoothly). The left and the right slope of the keyframe assume different values, that means the slope of the curve is not well defined at the keyframe.
shortcuts: b, b-key



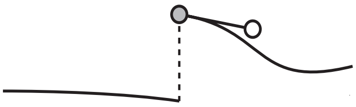
Linear

If a set of neighbouring keyframes is declared to be “Linear”, the curve pieces spanned by these keyframes are replaced by a piecewise linear curve.
shortcuts: l, p-key



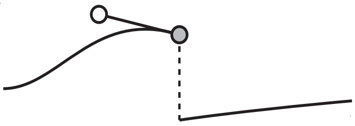
Left-Discontinuous

If a keyframe is declared to be “Left-Discontinuous”, the curve is not continuos in the keyframe and the keyframe belongs to the right curve segment.



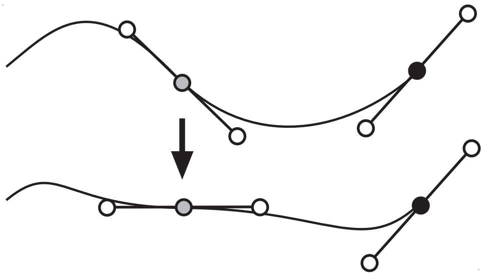
Right-Discontinuous

If a keyframe is declared to be “Right-Discontinuous”, the curve is not continuos in the keyframe and the keyframe belongs to the left curve segment.



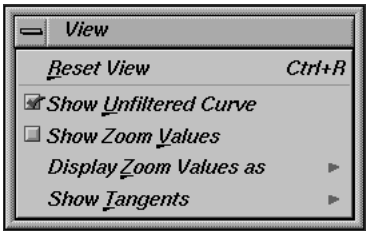
Flatten Tangents

The tangents of all selected keyframes will be made parallel to the x-axis of the coordinate frame.
shortcuts: f, CTRL-t



Menu: View

Functions to configure the Camera Zooming Window's display.
shortcut: ALT-v



Reset View

Scales and translates the coordinate frame, so that it fills out the entire display area of the Camera Zooming Window.
shortcuts: r, CTRL-r

Show Unfiltered Curve

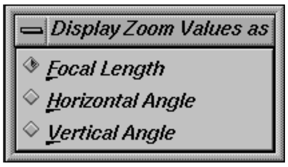
If activated, the unfiltered (grey) curve will be displayed.
shortcut: u

Show Zoom Values

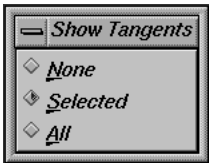
If activated, a label displayed for each keyframe is displayed, showing the particular zoom value.
shortcut: v
see also: Display Zoom Values as

Display Zoom Values as

Defines the unit type that will be used for displaying the zoom values. Choose among Focal Length, Horizontal Angle and Vertical Angle.
shortcuts: f, h, v
see also: Show Zoom Values

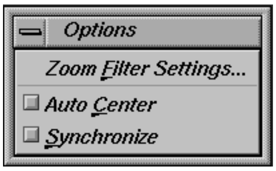


Show Tangents



Defines which of the keyframe's tangents should be displayed. Choose among: None, Selected and All. "None" means no tangents are displayed at all. "Selected" means only the tangents of the selected keyframes are displayed. "All" means the tangents of all keyframes are displayed.
shortcuts: n, s, a

Menu: Options



Functions for modifying the behaviour of the Camera Zooming Window.
shortcut: ALT-o

Zoom Filter Settings...

Opens a requester Edit Zoom Curve Filter Settings...
shortcut: f
see also: Main Window

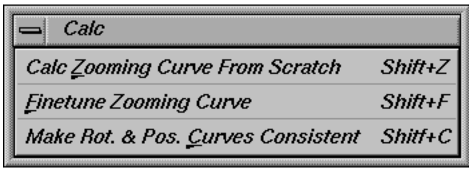
Auto Center

If activated, the Camera Zooming Window's coordinate frame is automatically translated in a way so that the currently selected frame is located in the center of the display area, every time the current frame changes.
shortcut: c

Synchronize

If activated, 3D-Equalizer is forced to automatically synchronize the horizontal scale and horizontal translation of the coordinate frames of all windows that have the synchronize mode activated.
shortcut: s
see also: Status Window, Timeline Window, Motion Path Window

Menu: Calc


Calc
Calc Zooming Curve From Scratch Shift+Z
Finetune Zooming Curve Shift+F
Make Rot. & Pos. Curves Consistent Shift+C

Functions for calculating the zooming curves.
shortcut: ALT-c

Calc Zooming Curve From Scratch

The unfiltered curve of the currently selected sequence is calculated entirely

from scratch. Afterwards, the unfiltered curve is automatically filtered using the current zoom filter settings. Out of the resulting data of this filtering process, the edit curve is created with as few keyframes as possible.
shortcuts: z, SHIFT-z
see also: Main Window, Zoom Filter Settings

Finetune Zooming Curve

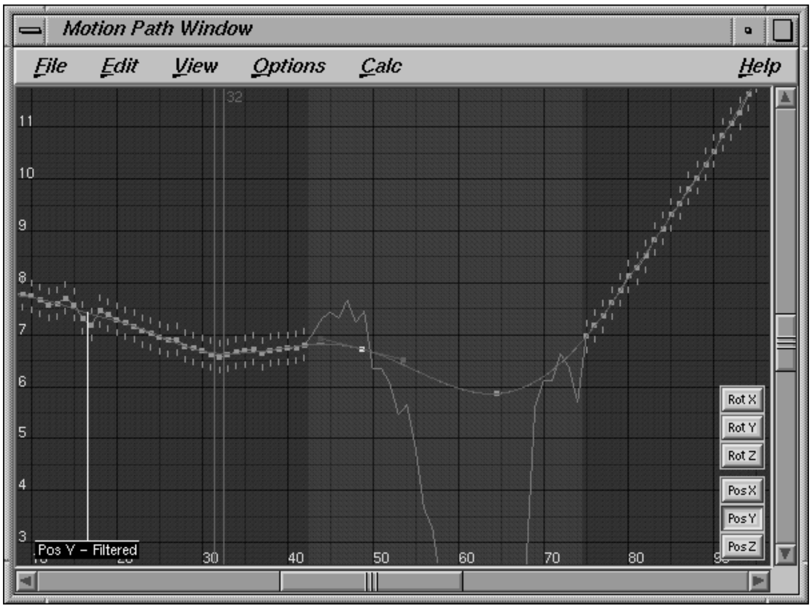
The edit curve of the currently selected sequence is taken as a starting point to compute a new unfiltered curve. Afterwards, the unfiltered curve is automatically filtered using the current zoom filter settings. Out of the resulting data of this filtering process, a new edit curve is created with as few keyframes as possible.
shortcuts: f, SHIFT-f
see also: Main Window, Zoom Filter Settings

Make Rot. & Pos. Curves Consistent

The rotational and positional three-dimensional motion paths of all pointgroups tracked in the current sequence are made consistent to the edit zooming curve. That means the edit zooming curve remains unchanged and the rotational and positional curves are modified, so that the deviation of the 3D motion paths becomes minimal for each frame.
shortcuts: c, SHIFT-c

Motion Path Window

Provides functions for editing the separate component curves of a 3D motion path. The Motion Path Window mainly consists of a two-dimensional display area. If there is a selected sequence and a selected pointgroup, the display area shows a 2D coordinate frame. The x-axis of this coordinate frame represents the number of frames of the currently selected sequence starting at frame 1. The y-axis of the coordinate frame represents positional or rotational values. The curves drawn on the coordinate frame represent the 3D motion path of the currently selected pointgroup tracked in the currently selected sequence. There are 6 component curves that build a 3D motion path: position x, position y, position z, rotation x, rotation y and rotation z. For each component curve there are 3 different versions available: the raw curve, the edit curve and the filtered curve. That means, all in all, the Motion Path Window may display 18 different curves at the same time.



Motion Path Window

The raw curves represent the data that have been calculated by 3D-Equalizer’s calculation core during the last calculation process. These curves are grey colored, and cannot be edited by the user. They are displayed for reference only.

The edit curves normally contain exactly the same data as the raw curves, but they are colored in blue, and the user is able to edit them. The last step of a calculation process is to copy the contents of the raw curves onto the edit curves, so that they are exactly the same. If the user wishes to fix a problem within the curves manually, he can do that by editing the edit curves. However, calculating the data will destroy the changes to the edit curves again. Therefore, editing the edit curves within the Motion Path Window should be avoided if possible. If it is necessary to modify the curves, it should always be the last step in a matchmoving process!

The filtered curves are red colored and cannot be edited by the user. They simply contain the result of filtering the edit curves, using the current postfilter settings (see requester Options::Postfilter Settings).

Frames in which at least one edit curve has been edited are indicated by a vertical grey colored rectangle, which is drawn behind all curves and the coordinate frame. Finally, the currently selected frame is represented by an orange colored outline. At the upper right side of this outline, the current frame number is indicated.

Selection and user interaction

The visible part of the Motion Path Window's coordinate frame can be changed similarly to the Zoom Window. By holding down the middle mouse button while dragging the mouse, the coordinate frame can be transposed. By holding down the right mouse button while dragging the mouse, the coordinate frame is scaled separately in two dimensions. Pressing the CTRL-key while holding down the right mouse button and dragging the mouse allows the user to draw a rubber box. After releasing the mouse button, the content of the rubber box will be zoomed in, so that it fills out the entire display area. As in the Timeline Window, the user can change the currently selected frame by holding down the CTRL-key while left-clicking into the coordinate frame. On the lower right hand side, there are six toggle buttons which allow the user to select the component of the 3D motion curve to be made editable. At most one toggle button can be activated. The editable component's edit curve is refered to as the currently selected edit curve.

Editing the edit curves

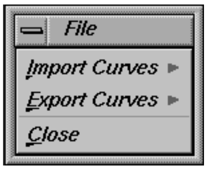
The edit curves are spline based curves consisting of keyframes. Normally there is a keyframe on each frame. Associated to each keyframe there are two tangents, which define the behaviour of the curve at that keyframe. A keyframe is represented by a point located directly on the edit curve. Unselected keyframes are drawn in white, whereas selected keyframes become yellow. The user is able to select a keyframe by clicking on it with the left mouse button. Keyframes that have been selected before automatically become unselected. To keep or toggle the selection status of other keyframes, the SHIFT-key must be pressed while clicking with the left mouse button. Click-dragging the left mouse button will cause 3D-Equalizer to draw a rubberbox and all keyframes that are located within this rubberbox become selected. All keyframes that have been formerly selected become unselected, unless the SHIFT-key is pressed while dragging the rubberbox. In order to create a new keyframe, the ALT-key must be pressed while clicking with the left mouse button on the coordinate frame. Click-dragging a keyframe or the endpoint of a tangent will change its position.

Keys

CURSOR-LEFT and CURSOR-RIGHT are used for jumping back and forth in the currenty selected sequence according to the setting of Main Window::Options::Frame Keyboard Grid. CTRL-CURSOR-LEFT and CTRL-CURSOR-RIGHT are used for going back and forth by one frame. CURSOR-UP and CURSOR-DOWN are used for selecting points. PAGE-UP/PAGE-DOWN jumps to the previous/next keyframe of the currently selected point. HOME/END jumps to the first/last sequence or reference frame.

Menu: File

Functions for importing and exporting the six edit curves associated to a 3D motion path.
shortcut: ALT-f



Import Curves::Raw

Opens a standard file requester. The selected file is interpreted as a 3D-Equalizer Raw 3D Edit Curves format, and replaces the edit curves of the currently selected sequence/pointgroup pair.
shortcut: i-r

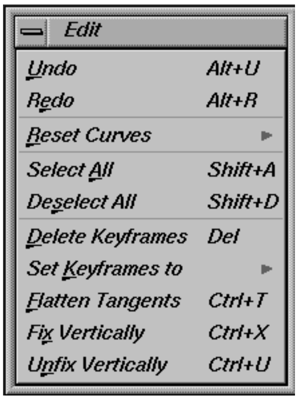
Export Curve::Raw

Opens a standard file requester. The edit curves of the currently selected sequence/pointgroup pair are written into the specified file in 3D-Equalizer's Raw 3D Edit Curves format.
shortcut: e-r

Close

Closes the Motion Path Window.
shortcut: c

Menu: Edit



Functions for modifing the currently selected edit curve.
shortcut: ALT-e

Undo

Undo the last edit action.
shortcut: ALT-u

Redo

Redo the last undo action.
shortcut: ALT-r

Reset Curves

Copies the contents of the raw curve(s) onto the edit curve(s).
shortcut: r

Selected

The contents of the currently selected edit curve are replaced by the contents of its raw curve.
shortcut: s

All

The contents of all edit curves are replaced by the contents of their raw curves.
shortcut: a

Select All

All keyframes of the currently selected edit curve become selected.
shortcuts: a, SHIFT-a

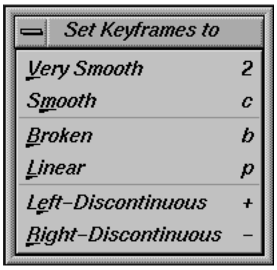
Deselect All

All keyframes of the currently selected edit curve become unselected.
shortcuts: s, SHIFT-d

Delete Keyframes

All selected keyframes will be deleted from the currently selected edit curve.
shortcuts: d, DELETE-key

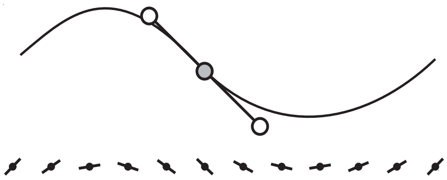
Set Keyframes to



Functions for modifying the behaviour of the selected keyframe’s tangents.
shortcut: k

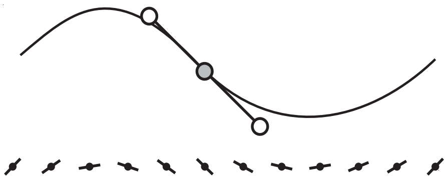
Very Smooth

If a keyframe is declared to be “Very Smooth”, the curve pieces left and right to the keyframe are glued together smoothly, and also the slopes of the curve pieces change smoothly in passing the keyframes. The curvature of the curve is well defined at the keyframe.
shortcuts: v, 2-key



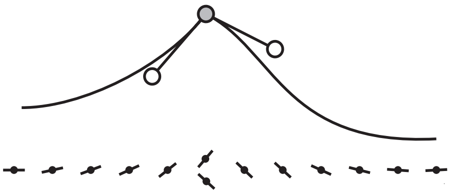
Smooth

If a keyframe is declared to be “Smooth”, the curve pieces left and right to the keyframe are glued together smoothly, and the slopes of the curve pieces change continously (however not smoothly) in passing the keyframe. This is the default behaviour for newly created keyframes.
shortcuts: m, c-key



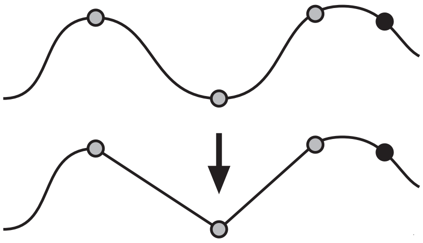
Broken

If a keyframe is declared to be “Broken”, the curve pieces are glued together continuosly (however not smoothly). The left and the right slope of the keyframe assume different values, that means the slope of the curve is not well defined at the keyframe.
shortcuts: b, b-key



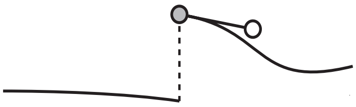
Linear

If a set of neighbouring keyframes is declared to be “Linear”, the curve pieces spanned by these keyframes are replaced by a piecewise linear curve.
shortcuts: l, p-key



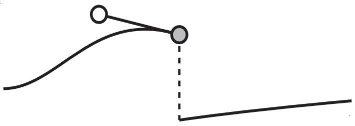
Left-Discontinuous

If a keyframe is declared to be “Left-Discontinuous”, the curve is not continuos in the keyframe and the keyframe belongs to the right curve segment.



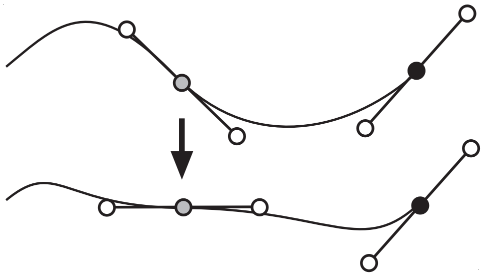
Right-Discontinuous

If a keyframe is declared to be “Right-Discontinuous”, the curve is not continuos in the keyframe and the keyframe belongs to the left curve segment.



Flatten Tangents

The tangents of all selected keyframes will be made parallel to the x-axis of the coordinate frame.
shortcuts: f, CTRL-t



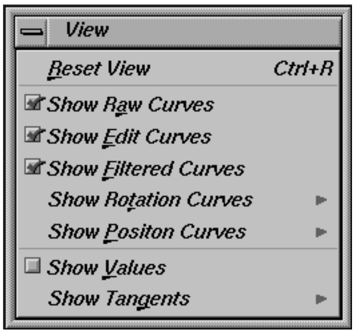
Fix Vertically

All selected keyframes of the currently selected edit curve will become vertically fixed. That means, they can no longer be moved horizontally. A keyframe which is vertically fixed, is indicated by two vertical lines drawn above and below the keyframe’s point.
shortcuts: x, CTRL-x

Unfix Vertically

All selected keyframes of the currently selected edit curve will no longer be vertically fixed.
shortcuts: n, CTRL-u

Menu: View



Functions to configure the Motion Path Window’s display.
shortcut: ALT-v

Reset View

Scales and translates the coordinate frame, so that it fills out the entire display area of the Motion Path Window.
shortcuts: r, CTRL-r

Show Raw Curves

If activated, the raw curves (grey) of all visible components of the 3D motion path will be displayed.
shortcut: a

Show Edit Curves

If activated, the edit curves (blue) of all visible components of the 3D motion path will be displayed.
shortcut: e

Show Filtered Curves

If activated, the filtered curves (red) of all visible components of the 3D motion path will be displayed.
shortcut: f

Show Rotation Curves

Togglebuttons that configure the visibility of the rotational curves of the 3D motion path. Choose among None, Selected and All. “None” means no rotational curves will be displayed at all. “Selected” means if the currently selected edit curve is a rotational one, it will be displayed. “All” means all rotational curves will be displayed.
shortcuts: n, s, a

Show Position Curves

Togglebuttons which configure the visibility of the positional curves of the 3D motion path. Choose among None, Selected and All. “None” means no positional curves will be displayed at all. “Selected” means if the currently selected edit curve is a positional one, it will be displayed. “All” means all positional curves will be displayed.
shortcuts: n, s, a

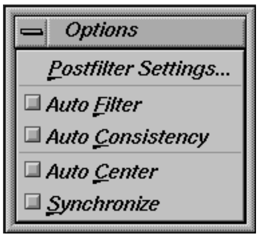
Show Values

If activated, a label for each keyframe of the currently selected edit curve is displayed, showing the particular function value.
shortcut: v

Show Tangents

Defines, which of the keyframe’s tangents should be displayed. Choose among: None, Selected and All. “None” means no tangents are displayed at all. “Selected” means only the tangents of the selected keyframes are displayed. “All” means the tangents of all keyframes are displayed.
shortcuts: n, s, a

Menu: Options



Functions for modifying the behaviour of the Motion Path Window.
shortcut: ALT-o

Postfilter Settings...

Opens a requester Edit Postfilter Settings...
shortcut: p
see also: Main Window

Auto Filter

If activated, all edit curves are automatically filtered using the current postfilter settings every time the user has edited the currently selected edit curve. The result of the filtering process is stored in the filtered curves (red), and the Motion Path Window's display is updated.
shortcut: f
see also: Postfilter Settings

Auto Consistency

If activated, the rotational edit curves are automatically made consistent to the positional edit curves, each time a positional edit curve has been edited (or vice versa). Making a set of curves consistent to another set of curves means modifying it, so that the deviation of the associated 3D motion path with respect to the selected pointgroup becomes minimal in each frame.
shortcut: c
see also: Status Window

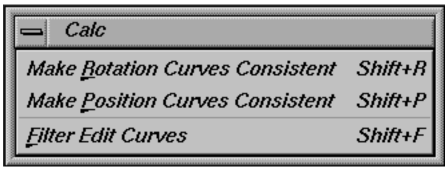
Auto Center

If activated, the Motion Path Window's coordinate frame is automatically transposed so that the currently selected frame is located in the center of the display area, every time the current frame changes.
shortcut: c

Synchronize

If activated, 3D-Equalizer is forced to automatically synchronize the horizontal scale and horizontal translation of the coordinate frames of all windows having synchronize mode activated.
shortcut: s
see also: Status Window, Timeline Window, Camera Zooming Window

Menu: Calc


Calc
Make Rotation Curves Consistent Shift+R
Make Position Curves Consistent Shift+P
Filter Edit Curves Shift+F

Functions for calculating certain curves.
shortcut: ALT-c

Make Rotation Curves Consistent

The rotational edit curves are made consistent to the positional curves. That means the positional curves remain unchanged and the rotational curves are modified, so that the deviation of the 3D motion path becomes minimal for each frame.
shortcuts: r, SHIFT-r
see also: Status Window, Options::Auto Consistency

Make Position Curves Consistent

The positional edit curves are made consistent to the rotational curves. That means, the rotational curves remain unchanged and the positional curves are modified, so that the deviation of the 3D motion path becomes minimal for each frame.
shortcuts: r, SHIFT-p
see also: Status Window, Options::Auto Consistency

Filter Edit Curves

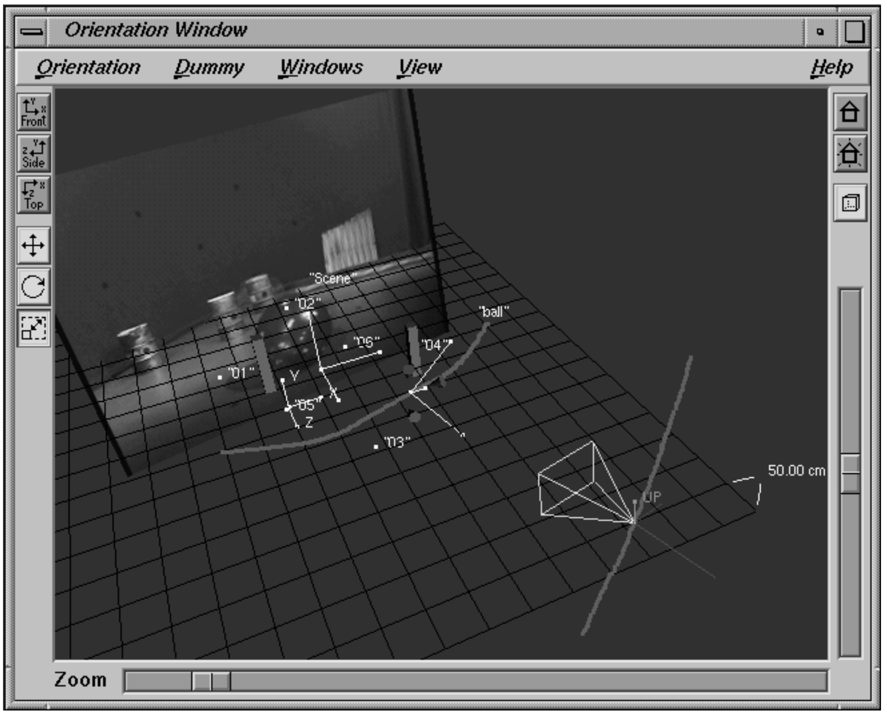
All edit curves of the currently selected sequence/pointgroup pair are filtered using the current postfilter settings. The result of the filtering process is stored in the filtered curves (red).
shortcuts: f, SHIFT-f

Orientation Window

Provides functions to browse and edit all types of three dimensional data which result in the course of the matchmoving process.

The Orientation Window mainly consists of a display area, which shows many different 3D objects such as cameras, 3D motion paths, points, 3D test (dummy) objects, coordinate frames, etc. The user is able to view and also edit many of the 3D objects shown in this window. The main purpose of this window is to define the position, rotation and scale of a matchmoving process's resulting 3D data (point clouds, cameras, etc.) within a “world-coordinate” frame. If the data are exported to some external animation package using a 3D-Equalizer exportfilter, the alignment in 3D space will be exactly the same as in the Orientation Window's display area.

Finally, simple 3D objects (dummies) can be created and edited. They are used to render preview movies.



Orientation Window

Miscellaneous

The 3D object hierarchy

The 3D objects displayed in the Orientation Window are organized in a certain hierarchy. That means, there are child objects grouped under parent objects, and so on. Furthermore, due to the nature of 3D-Equalizer's application, there are many constraints between different 3D objects. For instance, the relationship between a 3D camera and the respective 3D point cloud must not be destroyed due to a user action. This fact can result in strange behaviours. For instance, moving a point through 3D space, will move the entire hierarchy through 3D space.

```
Scene ¬
      Camera #1...Camera #n
      Point #1...Point #n
      Dummy #1...Dummy #n
      Pointgroup #1 ¬
            Point #1...Point #n
            Dummy #1...Dummy #n
      Pointgroup #2 ¬
            Point #1...Point #n
            Dummy #1...Dummy #n
      ...
```

Schematic 3D object hierarchy

There are 5 different types of 3D objects found in the hierarchy: scene nodes, cameras, points, dummies and pointgroups. The top of the hierarchy is build by the scene node. There is always only one scene node. It represents a classical “group” object, all other objects are children of it. Therefore, transforming the scene node will have an effect on all objects at once. The scene node is drawn in the Orientation Window as a “xyz-axis coordinate frame” with the label “Scene” attached to it.

The first set of objects grouped under the scene node are the camera objects. A camera object represents a sequence or a reference frame. There are always as many cameras as sequences and reference frames available. However, only the camera that relates to the currently selected sequence or reference frame is visible. Cameras that relate to a sequence are animated if a camera pointgroup exists, otherwise they are fixed. A camera in the Orientation Window is represented by a pyramid, the tip of which represents the camera position in 3D space.

The next set of objects grouped directly under the scene node are the points of the camera pointgroup. If there is no camera pointgroup, no points will be available at this place in the hierarchy. A point object is drawn in the Orientation Window by a small dot with the point’s name attached as a label to it.

The next set of objects grouped directly under the scene node are the dummy objects associated to the camera pointgroup. If there is no camera pointgroup, no dummy objects will be available at this place in the hierarchy. Dummies are usually simple geometrical objects like spheres, cubes or cones. They can be transformed separately, without “injuring” any constraints.

The final set of objects grouped directly under the scene node are the pointgroup objects. Each pointgroup object represents an “object” pointgroup defined in the Pointgroups Window. A pointgroup object is very similar to the scene node, mainly acting as a parent object to points and dummies that are grouped under it. A pointgroup object is always animated. It is drawn in the Orientation Window as a “xyz-axis coordinate frame” with the “object” pointgroup’s name attached as a label to it. Pointgroup objects can be scaled separately from the rest of the hierarchy without “injuring” any constraints!

There are always a set of point objects and a set of dummies grouped under each pointgroup object.

The display

The Orientation Window displays the 3D objects assembled in the 3D object hierarchy according to their respective 3D transformations. Furthermore, a white xyz coordinate frame representing the world-space and a black grid representing the xz-plane is displayed. The appearance of many objects can be configured using the togglebuttons of the View Menu. The entire scene is viewed by a camera which is referred to as the *internal camera* in the following.

Navigating the Internal Camera

The internal camera can be rotated around a virtual 3D center point by holding down the ALT-key and the left mouse button, while dragging the mouse. To translate the internal camera and its virtual 3D center point you must hold down the ALT-key and the middle mouse button, while dragging the mouse. Finally, the internal camera can be moved towards its 3D center point (dolly function), by holding down the ALT-key and the right mouse button, while dragging the mouse. In addition, on the right hand side of the Orientation Window there is a vertical slider, that can be used to “dolly” the internal camera, as well.

Zoom

A slider which allows the user to modify the focal length of the internal camera. If parallel perspective representation is chosen, this slider has no effect.

Selection

Single points and dummy objects can be selected by holding down the CTRL-key while clicking on it with the left mouse button. A previously selected object becomes unselected. A selected dummy is framed by a red bounding box, selected points are rendered in yellow.

Modifying a 3D object

On the left hand side of the Orientation Window there are buttons for modifying the currently selected object. It can be translated, rotated and scaled using the following functions:

Translate Object

If activated, the currently selected 3D object can be translated in 3D space. By click-dragging the left mouse button the object is translated along the x-axis of its own coordinate frame. Pressing the SHIFT-key while click-dragging the left mouse button will translate the object along the global x-axis. Using the middle mouse button translates in y, while the right mouse button translates in z-direction.

Rotate Object

If activated, this function enables the user to rotate the currently selected object around one of its own or the global coordinate frame axis. By pressing (and holding down) the left mouse button while moving the mouse, the object is rotated around its own x-axis. Pressing the SHIFT-key causes the currently selected object to be rotated around the global coordinate frame’s x-axis. Using the middle mouse button rotates the object around the y, while the right mouse button rotates it around the z-axis.

Scale Object

If activated, this function enables the user to scale the currently selected object. All objects, except for the dummies, can only be scaled uniformly. By pressing (and holding down) the left mouse button while moving the mouse, the currently selected dummy object is scaled in its x-dimension. Using the middle mouse button scales in y, while the right mouse button scales it in z-dimension. Pressing all three mouse buttons together causes the dummy to be scaled uniformly. Combinations of two mouse buttons are allowed as well.

Front View

Sets the internal camera of the Orientation Window to front view. That means, the internal camera looks along the negative z-axis of the global coordinate frame.

Side View

Sets the internal camera of the Orientation Window to side view. That means, the internal camera looks along the negative x-axis of the global coordinate frame.

Top View

Sets the internal camera of the Orientation Window to top view. That means, the internal camera looks along the negative y-axis of the global coordinate frame.

Home button

Resets the camera to its home position.
see also: Set Home button

Set Home button

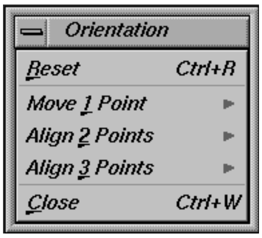
Defines the current camera position as the home position.
see also: Home button

Perspective/Orthogonal projection toggle

Switches the camera between perspective and orthogonal projection. While in orthogonal projection mode, the function View::Camera View cannot be activated.
see also: Menu: View

Menu: Orientation

Provides functions to align the entire 3D object hierarchy to the global coordinate frame. Example: assume you have three points A, B and C. You wish point A to coincide with the origin, points B and C to be parallel to the y-axis and points A, B and C to lie in the yz-plane. The simplest way to achieve this is to select points A, B and C in the 3D Object Browser Window and to align them within the yz-plane. Afterwards, points B and C are selected and aligned with respect to the y-axis. Finally, point A is selected and moved into the origin of the coordinate frame.
shortcut: ALT-o



Reset

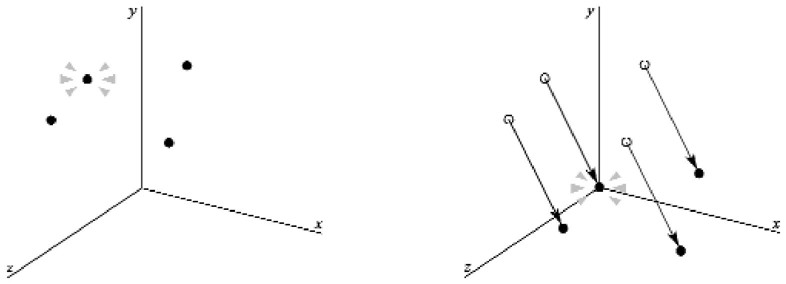
Resets scale, position and orientation of the 3D object hierarchy by resetting the scene node’s transformation attributes. Its position is set to (0,0,0), its scale is set to 1.0 and its rotation is set to (0,0,0).
*shortcut*s: r, CTRL-r

Move 1 Point

All functions found in this menu require one point to be selected in the 3D Object Browser Window for operation.
shortcut: 1
see also: 3D Object Browser Window

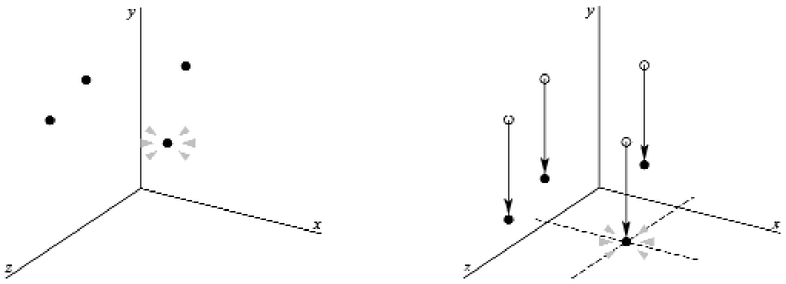
Origin

The scene node, and thus the entire 3D object hierarchy, is transformed so that the currently selected point is located in the origin of the global coordinate frame.
shortcut: o



XY-, XZ-, YZ-Plane

The scene node, and thus the entire 3D object hierarchy, is transformed so that the currently selected point is projected orthogonally into the xy-, xz- or yz-plane of the global coordinate frame.
shortcuts: x, z, y

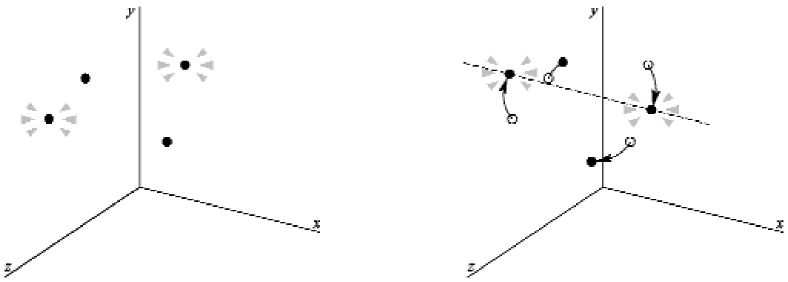


Align 2 Points

All functions found in this menu require two points to be selected in the 3D Object Browser Window for operation.
shortcut: 2
see also: 3D Object Browser Window

X-, Y-, Z-Axis

The scene node, and thus the entire 3D object hierarchy, is transformed so that the two currently selected points are aligned parallel to the x-, y- or z-axis of the global coordinate frame.
shortcuts: x, y, z

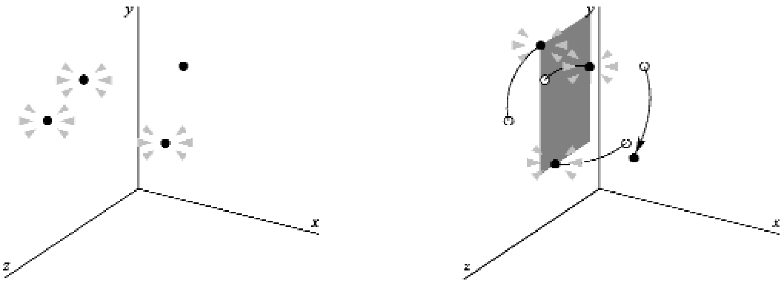


Align 3 Points

All functions found in this menu require three points to be selected in the 3D Object Browser Window for operation.
shortcut: 3
see also: 3D Object Browser Window

XY-, XZ-, YZ-Plane

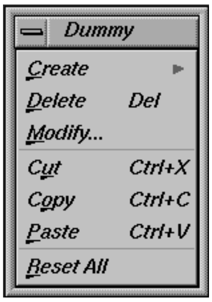
The scene node, and thus the entire 3D object hierarchy, is transformed so that the three currently selected points are aligned to the xy-, xz- or yz-plane of the global coordinate frame.
shortcuts: x, z, y



Close

Closes the Orientation Window, and all its sub-windows.
shortcuts: c, CTRL-w

Menu: Dummy



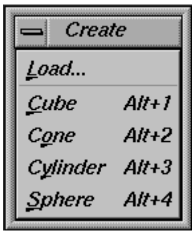
Menu providing elementary functions for creating and deleting dummies.
shortcut: ALT-d

Create

Functions for creating dummies. If more than one pointgroup exists, an additional requester opens, which allows the user to specify to which pointgroup the newly created dummy object should be assigned.
shortcut: c

Load...

Opens a standard file requester for loading a dummy object. 3D-Equalizer expects a dummy file to be in “OBJ” format.
shortcut: l



Cube

Creates a cube-shaped dummy.
shortcuts: c, ALT-1

Cone

Creates a cone-shaped dummy.
shortcuts: o, ALT-2

Cylinder

Creates a cylinder-shaped dummy.
shortcuts: y, ALT-3

Sphere

Creates a sphere-shaped dummy.
shortcuts: s, ALT-4

Delete

Removes the currently selected dummy from the object hierarchy.
shortcuts: d, DELETE

Cut

Copies the currently selected dummy object, including its size, orientation and material properties to the clipboard and removes it from the current object hierarchy.
shortcuts: u, CTRL-x

Copy

Copies the currently selected dummy object, including its size, orientation and material properties to the clipboard. Unlike the Cut function, the dummy remains in the current object hierarchy.
shortcuts: o, CTRL-c

Paste

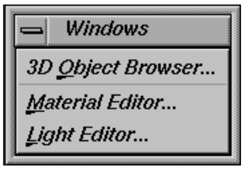
Creates a new copy of the dummy that is currently in the clipboard and places it at the origin of its parent object’s coordinate frame. If there is more than only one pointgroup available, a requester opens that lets the user specify to which pointgroup the newly created dummy object should be assigned.
shortcut: p, CTRL-v

Reset All

Resets the position, orientation and size of the currently selected dummy object.
shortcut: r

Menu: Windows

Provides menu items to open the 3D Object Browser, the Material Editor and the Light Editor Window.
shortcut: ALT-w



3D Object Browser...

Opens the 3D Object Browser Window.
shortcut: o

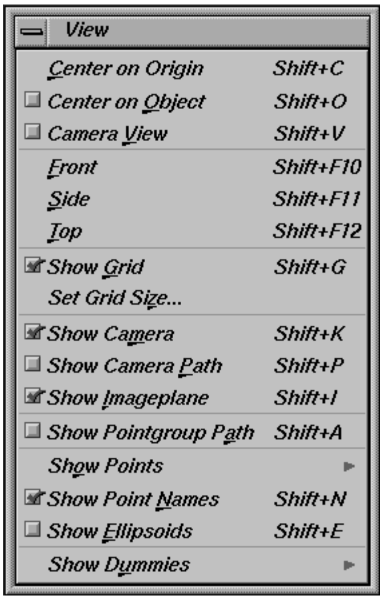
Material Editor

Opens the Material Editor Window.
shortcut: m

Light Editor

Opens the Light Editor Window.
shortcut: l

Menu: View



Functions for configuring the Orientation Window's display area.
shortcut: ALT-v

Center on Origin

Centers the internal camera of the Orientation Window on the origin of the global coordinate frame.
shortcuts: c, SHIFT-c

Center on Object

If activated, the internal camera is automatically centered on the currently selected object.
shortcuts: o, SHIFT-o

Camera View

If activated, the internal camera is automatically made equal to the camera object associated to the currently selected sequence or reference frame, each time the current frame changes. That means, the Orientation Window's display area shows the 3D object hierarchy from the point of view of the reconstructed camera, just like the Overview Window. This togglebutton can only be activated. while in perspective projection mode.
shortcuts: v, SHIFT-v

Front, Side, Top

Sets the internal camera of the Orientation Window to front, right side or top of the 3D object hierarchy, respective to the displayed global coordinate frame.
shortcuts: f, SHIFT-F10, s, SHIFT-F11, t, SHIFT-F12

Show Grid

If activated, a black grid with a white xyz coordinate frame is displayed at the origin of the global coordinate frame. The size of this grid can be specified in real length units using the Set Grid Size function.
shortcuts: g, SHIFT-g

Set Grid Size...

Opens a requester Change Grid Size.
shortcut: z

Show Camera

If activated, the camera object that is associated to the currently selected sequence or reference frame is displayed.
shortcuts: m, SHIFT-k

Show Camera Path

If activated and if the camera object is visible, the filtered 3D motion curve of the project's camera pointgroup tracked in the currently selected sequence is displayed as a three-dimensional red curve.
shortcuts: p, SHIFT-p

Show Imageplane

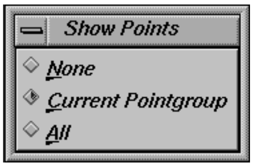
If activated and if the camera object is visible, the currently selected frame is displayed on a 3D plane which is directly attached to the camera object. The texture on the 3D plane is automatically updated each time the current frame changes.
shortcuts: i, SHIFT-i

Show PointgroupPath

If activated, the filtered 3D motion curve of all "object" pointgroups tracked in the currently selected sequence are displayed as separate three-dimensional red curves.
shortcuts: a, SHIFT-a

Show Points

Togglebuttons to configure the point object's visibility.
shortcut: o



None

If activated, no point objects will be displayed at all.
shortcut: n

Current Pointgroup

If activated, only the points of the currently selected pointgroup will be displayed as point objects within the Orientation Window's display area.
shortcut: c

All

If activated, the points of all pointgroups will be displayed as point objects within the Orientation Window's display area.
shortcut: a

Show Point Names

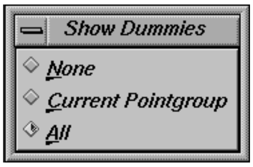
If this togglebutton is activated, there will be a label displayed for each visible point object. Each label carries the name of the individual point associated to the respective point object.
shortcuts: n, SHIFT-n

Show Ellipsoids

If activated, there will be small blue transparent spheres displayed at each visisble point object. The shape of a these spheres indicates how well the respective point's position has been reconstructed in 3D space. A flat sphere indicates that a point's position is quite good, whereas a sharp oblong sphere means that a point's reconstructed position is not very precise.
shortcuts: e, SHIFT-e

Show Dummies

A menu that contains togglebuttons to configure the visibility of dummy objects.
shortcut: u



Show Dummies

◇ None
◇ Current Pointgroup
◆ All

None

If activated, no dummy objects will be displayed at all.
shortcut: n

Current Pointgroup

If activated, only the dummy objects of the currently selected pointgroup will be displayed in the Orientation Window.
shortcut: c

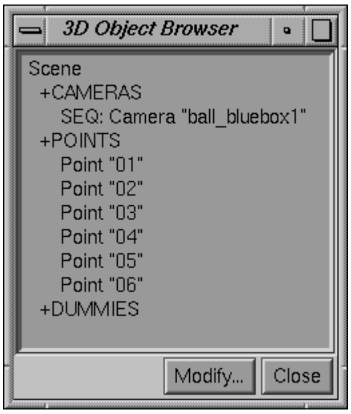
All

If activated, all available dummy objects will be displayed in the Orientation Window.
shortcut: a

Window: 3D Object Browser

The 3D Object Browser Window gives the user a schematic overview of the 3D object hierarchy currently displayed in the Orientation Window. In addition the user is able to select all kinds of 3D objects by the use of this window. Its position and size are stored in the environments.

The window mainly consists of a list widget. Each entry in this list widget either represents a 3D object or a so-called wrapper entry. A wrapper entry contains of a '+' or '-' sign, followed by a label written in uppercase letters. It represents a "virtual" parent object for a number of 3D objects of the same type like points or dummies. If a wrapper entry is in '+' mode, its child object entries are visible in the list widget, whereas if in '-' mode, its child object entries are hidden. To toggle between '+' and '-' mode, the user simply has to click on it. The wrapper entires are useful for maintaining an overview of the entire 3D object hierarchy.



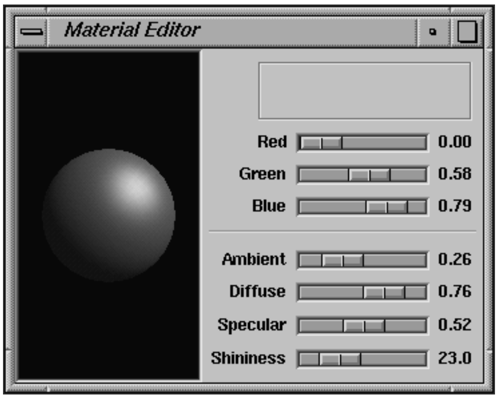
```
Scene
 +CAMERAS
   SEQ: Camera "ball_bluebox1"
 +POINTS
   Point "01"
   Point "02"
   Point "03"
   Point "04"
   Point "05"
   Point "06"
 +DUMMIES
```

Window: 3D Object Browser

A 3D object is selected in the list, and thus in the Orientation Window, by simply clicking on it with the left mouse button. Except for point objects, only one object can be selected at a time. Objects that are currently not visible in the Orientation Window, either because their visibility has been disabled or because they have not been calculated yet, are marked as “(invisible)”. Objects that are grouped under other objects are indented. Double-clicking on an object entry or clicking the Modify button will open a requester which enables the user to edit various object specific attributes (see also attribute requester).

Window: Material Editor

This window enables the user to modify the material/surface properties of the currently selected dummy object. Its position and size are stored in the environments.

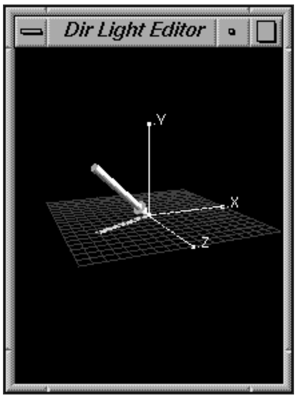


Red 0.00
Green 0.58
Blue 0.79
Ambient 0.26
Diffuse 0.76
Specular 0.52
Shininess 23.0

Window: Material Editor

Window: Dir Light Editor

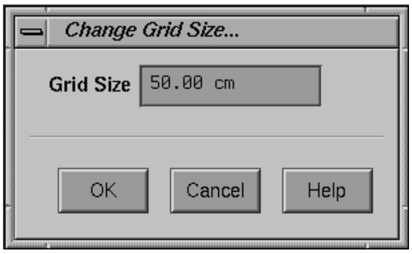
This window provides the functionality for editing a light source (direction) which illuminates the dummies. Its position and size are stored in the environments.



Window: Dir Light Editor

Requester: Change Grid Size...

A requester to set the size of the Orientation Window’s grid plane. The user can specify the length of one grid unit in real measurements. After the requester has been confirmed, the size of the entire grid is adjusted, and the new length of one grid unit is displayed as a small label at the corner of the grid.

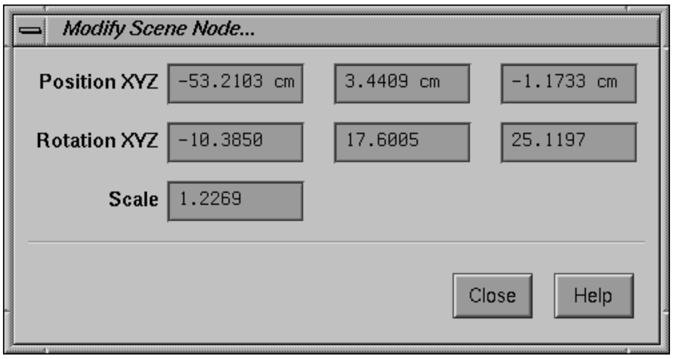


Grid Size: 50.00 cm

Requester: Change Grid Size...

Requester: Modify Scene Node...

A requester to edit a scene node object’s attributes. To open this requester the “Scene” entry in the 3D Object Browser must be double-clicked, or the Modify button must be pressed while the scene node object is selected. This requester, unlike most of 3D-Equalizer’s other requesters, does not block the application.



Position XYZ: -53.2103 cm | 3.4409 cm | -1.1733 cm

Rotation XYZ: -10.3850 | 17.6005 | 25.1197

Scale: 1.2269

Requester: Modify Scene Node...

Position XYZ

Textfields that contain the scene node's position regarding the global coordinate frame.

Rotation XYZ

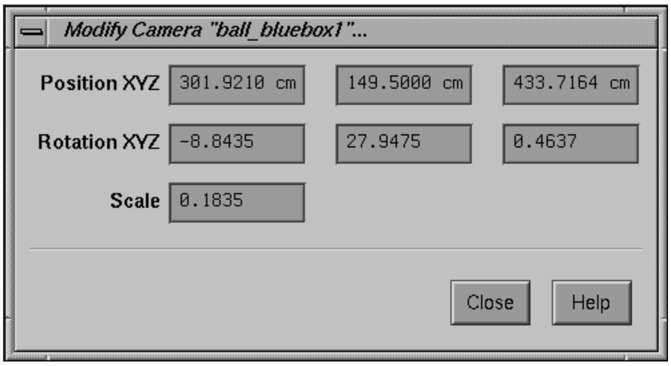
Textfields that contain the scene node's rotation regarding the global coordinate frame. The values are given out as angles, the rotation order is z-x-y.

Scale

Textfield that contains the scene node's scale.

Requester: Modify Camera...

A requester to edit a camera object's attributes. To open this requester, the respective entry in the 3D Object Browser must be double-clicked, or the Modify button must be pressed while the camera object is selected. This requester, unlike most of 3D-Equalizer's other requesters, does not block the application. Modifying most of the requester's values actually causes the scene node, and thus all objects, to be modified, because the complex relationships between the 3D objects must not be destroyed.



Modify Camera "ball_bluebox1"...

Position XYZ	301.9210 cm	149.5000 cm	433.7164 cm
Rotation XYZ	-8.8435	27.9475	0.4637
Scale	0.1835		

Requester: Modify Camera...

Position XYZ

Textfields that contain the camera object's position in the current frame with regard to the global coordinate frame. Modifying these values causes the scene node to be modified.

Rotation XYZ

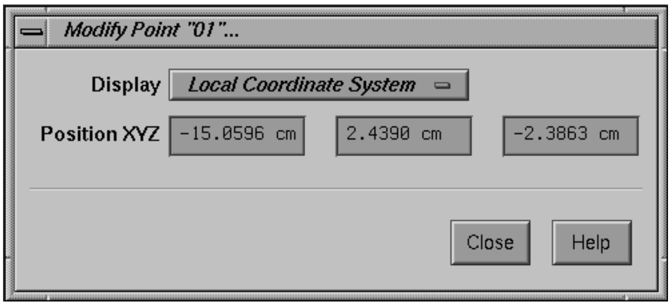
Textfields that contain the camera object's rotation with regard to the global coordinate frame. The values are given out as angles, the rotation order is z-x-y. Modifying them causes the scene node to be modified.

Scale

Textfield that contains the camera object’s scale. This parameter defines the size of the camera symbol in the Orientation Window’s display area, as well as size and distance of the imageplane. It has no effect on the scene node object.

Requester: Modify Point...

A requester to edit a point object’s attributes. To open this requester, the respective entry in the 3D Object Browser must be double-clicked, or the Modify button must be pressed while the point object is selected. This requester, unlike most of 3D-Equalizer’s other requesters, does not block the application. Modifying most of the requester’s values actually causes the scene node, and thus all objects, to be modified, because the complex relationships between the 3D objects must not be destroyed.



Modify Point "01"...

Display: Local Coordinate System

Position XYZ: -15.0596 cm | 2.4390 cm | -2.3863 cm

Requester: Modify Point...

Display

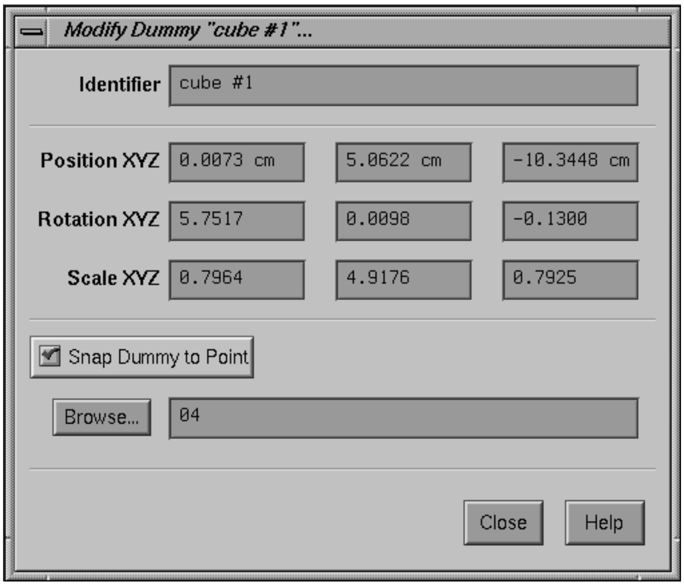
An option menu to choose the Position XYZ textfield’s mode of display. Choose among Local Coordinate System and Global Coordinate System. If set to Local Coordinate System, the Position XYZ textfields will display the point object’s coordinates with respect to its pointgroup’s local coordinate frame. In this case the textfields cannot be edited. If set to Global Coordinate System, the point object’s position are displayed with respect to the global coordinate frame.

Position XYZ

Textfields that contain the point object’s position in the current frame. The respective coordinate frame depends on the setting of the Display option menu. Modifying these values causes the scene node to be modified.
see also: Display

Requester: Modify Dummy...

A requester to edit a dummy object’s attributes. To open this requester, the respective entry in the 3D Object Browser must be double-clicked, or the Modify button must be pressed while the camera object is selected. This requester, unlike most of 3D-Equalizer’s other requesters, does not block the application.



Modify Dummy "cube #1"...

Identifier	cube #1		
Position XYZ	0.0073 cm	5.0622 cm	-10.3448 cm
Rotation XYZ	5.7517	0.0098	-0.1300
Scale XYZ	0.7964	4.9176	0.7925

☑ Snap Dummy to Point

Browse... 04

Requester: Modify Dummy...

Identifier

The name of the dummy object as it appears in the 3D Object Browser Window.

Position XYZ

Textfields that contain the dummy object’s position in respect to its associated pointgroup’s coordinate frame. If the Snap Dummy to Point mode is activated, these textfields cannot be edited.

Rotation XYZ

Textfields that contain the dummy object’s rotation in respect to its associated pointgroup’s coordinate frame. The values are given out as angles, the rotation order is z-x-y.

Scale XYZ

Textfield containing the dummy object’s scale.

Snap Dummy to Point

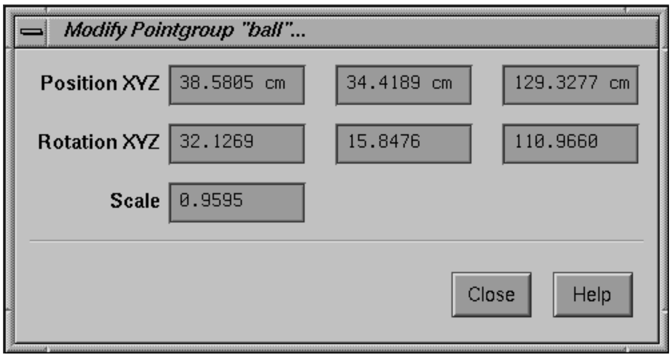
If activated, the dummy is automatically aligned to a given point object found in the same pointgroup.

Browse

Opens a requester which shows a list of the available points of the pointgroup to which the dummy is associated. Choosing a point from the list causes the dummy to be automatically aligned to the point in 3D space.

Requester: Modify Pointgroup...

A requester to edit a pointgroup object's attributes. To open this requester the respective entry in the 3D Object Browser must be double-clicked, or the Modify button must be pressed while the pointgroup object is selected. This requester, unlike most of 3D-Equalizer's other requesters, does not block the application. Modifying most of the requester's values actually causes the scene node, and thus all objects, to be modified, because the complex relationships between the 3D objects must not be destroyed.



Modify Pointgroup "ball"...

Position XYZ	38.5805 cm	34.4189 cm	129.3277 cm
Rotation XYZ	32.1269	15.8476	110.9660
Scale	0.9595		

Requester: Modify Pointgroup...

Position XYZ

Textfields that contain the pointgroup object's position in the current frame with regard to the global coordinate frame. Modifying these values causes the scene node to be modified.

Rotation XYZ

Textfields that contain the pointgroup object's rotation with respect to the global coordinate frame. The values are given out as angles, the rotation order is z-x-y. Modifying them causes the scene node to be modified.

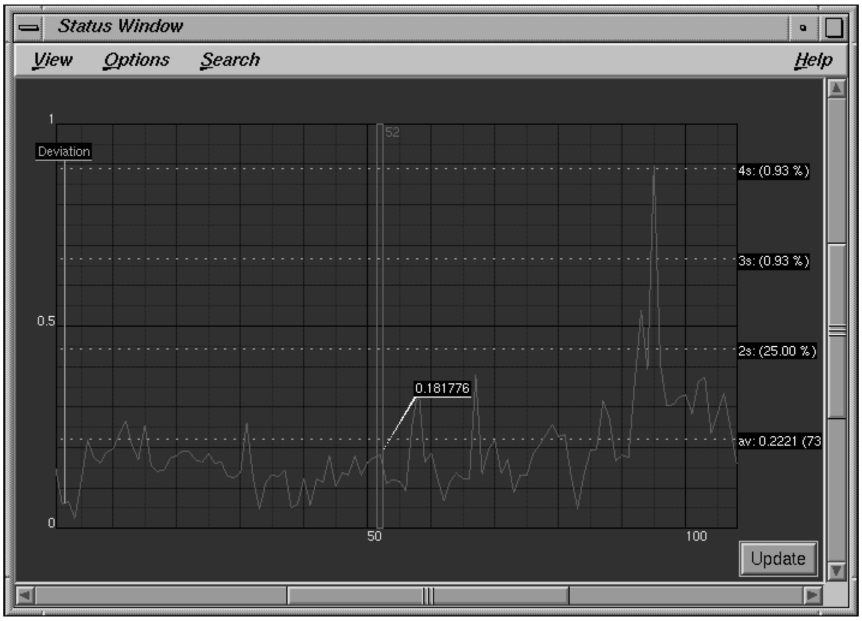
Scale

Textfield that contains the pointgroup object's scale. This parameter defines the size of the pointgroup object in the Orientation Window's display area, as well as the size of its children. It has no effect on the scene node object. Scaling a pointgroup object causes a change in the shape of the pointgroup's motion path within the Orientation Window, however it has no effect on the respective curves shown in the Motion Path Window.

Reference Manual

Status Window

Displays error and statistical information about the current project. The Status Window mainly consists of a two-dimensional display area. If there is a currently selected sequence or reference frame and a currently selected pointgroup, the display area shows a coordinate frame featuring a 2D coordinate frame. The x-axis of this coordinate frame represents the number of frames of the currently selected sequence starting with frame 1. If there is a reference frame selected, the x-axis represents all available reference frames, where the left-most frame relates to the uppermost reference frame object found in the Frame Window's list widget. The y-axis of the coordinate frame represents the so-called deviation in pixels, starting at a value of 0.



Status Window

A green curve, simply named deviation, displays the overall reconstruction quality in each single frame. Frames with a lower deviation value are reconstructed better than frames with a higher deviation value. If the deviation curve is visible, some additional statistical information is displayed by means of white colored, dotted lines to which black labels are attached. The lowest line marks the so-called average deviation, a single value which expresses the overall reconstruction quality of an entire sequence. Furthermore, there can be a separate curve for each calculated point found in the currently selected pointgroup. These curves are called point deviations and represent the reconstruction quality of a particular point in each frame. Point deviation curves are usually colored in blue, except for the curves of selected points, which are colored in orange. By analysing these curves, the user is able to find any and all kinds of tracking errors, and thus is able to “debug” a project. Finally, the currently selected frame is prepresented by an orange colored outline.

Selection and user interaction

The visible part of the Status Window’s coordinate frame can be changed similarly to the Zoom Window. By holding down the middle mouse button while dragging the mouse, the coordinate frame can be translated. By holding down the right mouse button while dragging the mouse, the coordinate frame is scaled separately in two dimensions. Pressing the CTRL-key, while holding down the right mouse button and dragging the mouse, allows the user to draw a rubberbox. After releasing the mouse button, the contents of the rubber box will be zoomed in, so that it fills out the entire display area. As in the Timeline Window, the user can change the currently selected frame by holding down the CTRL-key while left-clicking into the coordinate frame.

By holding down the ALT-key while click-dragging the left mouse button a rubberbox can be created. All point deviation curves that are located within the rubberbox become selected. This mechanism can only be used if the deviation curves of all points are visible.

Keys

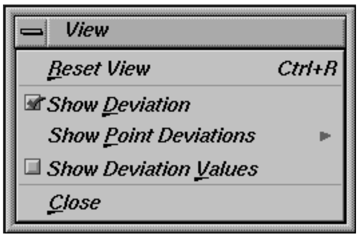
CURSOR-LEFT and CURSOR-RIGHT are used for jumping back and forth in the currenty selected sequence according to the setting of Frame Keyboard Grid. CTRL-CURSOR-LEFT and CTRL-CURSOR-RIGHT are used for going back and forth by one frame. CURSOR-UP and CURSOR-DOWN are used for selecting points. PAGE-UP/PAGE-DOWN jumps to the previous/next keyframe of the currently selected point. HOME/END jumps to the first/last sequence or reference frame.

Update

A function to quickly recalculate the curves displayed in the Status Window, after some changes have been applied to the project. This function typically operates at a very high speed and givens a good estimation of the resulting deviation.

Menu: View

Functions to configure the Status Window’s display area.
shortcut: ALT-v



Reset View

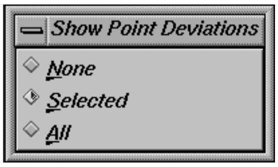
Resets the Status Window’s coordinate frame, so that it fills out the display area entirely.
shortcuts: r, CTRL-r

Show Deviation

If activated, the green deviation curve is displayed in the coordinate frame of the Status Window. It represents the reconstruction quality of each single frame.
shortcut: d

Show Point Deviations

Togglebuttons to show or hide point deviation curves. A point deviation curve represents the reconstruction quality of a particular point in each frame.
shortcut: p



None

If activated, no point curve is visible at all.
shortcut: n

Selected

If activated, an orange colored point deviation curve is drawn for the currently selected point in the coordinate frame of the Status Window. For each remaining point that is selected in the Timeline Window, its own blue-colored point deviation curve is drawn.
shortcut: s

All

If activated, for each point of the currently selected pointgroup, a point deviation curve is displayed in the coordinate frame of the Status Window. The curves of the currently selected points are colored in orange, whereas all other curves are colored in blue. Multiple points can be selected by holding down the ALT-key, while click-dragging the left mouse button.
shortcut: a

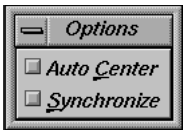
Show Deviation Values

If activated, each frame of the deviation curve is labeled, showing its deviation value. If the deviation curve is hidden, labels are displayed for the point deviation curve of the currently selected point.
shortcut: v

Close

Closes the Status Window.
shortcut: c

Menu: Options



Togglebuttons to configure the Status Window's behaviour.
shortcut: ALT-o

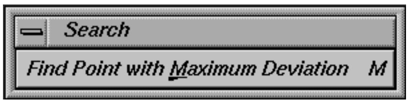
Auto Center

If activated, the Status Window's coordinate frame is automatically transposed so that the currently selected frame is located in the center of the display area, every time the current frame changes.
shortcut: c

Synchronize

If activated, 3D-Equalizer is forced to automatically synchronize the horizontal scale and horizontal translation of the coordinate frames of all windows that have the synchronize mode activated.
shortcut: s
see also: Timeline Window, Camera Zooming Window, Motion Path Window

Menu: Search



Menu containing miscellaneous search functions.
shortcut: ALT-s

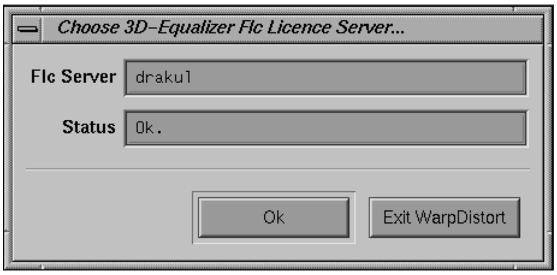
Find Point with Maximum Deviation

The point that has the highest individual deviation in the currently selected frame is identified and becomes the currently selected point.
shortcuts: m, M-key

Warpdistort

An external application to apply or remove lens distortion to a given sequence of images using 2D image warping techniques. The warpdistort program can be found in 3D-Equalizer's installation directory ("<3de install directory>/bin/"). It can be started out of a shell window, by typing in "warpdistort" and pressing the RETURN-key.

After it has been started, two windows and a requester open up. There are the Main Window, the Image View Window and the Flc Server Requester. Warpdistort needs to find a 3D-Equalizer Version 3 licence (Film option), in order to operate. This licence can already be in use by another 3D-Equalizer main program. In fact, an unlimited number of "warpdistorts" will run at the same time, as long as at least one valid film licence can be found.


Choose 3D-Equalizer Flc Licence Server...

Flc Server: drakul

Status: Ok.

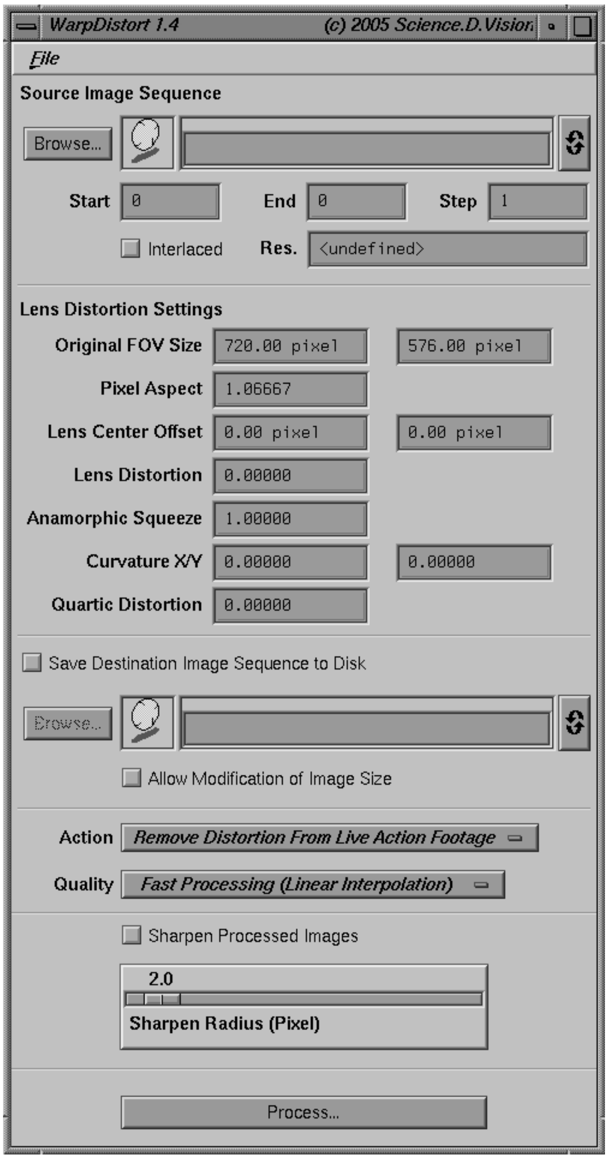
Requester: Flc Server...

Shortly after startup, the Flc Server Requester will ask the user to enter the name or IP address of a machine running a flc server deamon that handles a valid film licence, in the textfield Flc Server. If warpdistort is able to find a valid licence, the message "Ok." is displayed in the textfield Status. If no valid film licence can be found, or if warpdistort is not able to connect to a flc server deamon, the message "Error, no valid 3D-Equalizer film licence found." is displayed. Pressing the Ok button, will close the requester and enables the user to start working with warpdistort.

The basic workflow with warpdistort is very straight forward. First of all, an image sequence is specified, to be processed. After that, several parameters can be set to define the type and amount of lens distortion as well as the mode of operation. Furthermore, a destination image sequence can be specified into which the processed images are stored. Finally, the image processing procedure is started, and the results can be examined in the Image View Window.

Main Window

The Main Window contains all textfields, buttons and menus that are necessary to control WarpDistort's entire functionality.



Source Image Sequence

Start 0   End 0   Step 1

□ Interlaced   Res. <undefined>

Lens Distortion Settings

Original FOV Size 720.00 pixel   576.00 pixel

Pixel Aspect 1.06667

Lens Center Offset 0.00 pixel   0.00 pixel

Lens Distortion 0.00000

Anamorphic Squeeze 1.00000

Curvature X/Y 0.00000   0.00000

Quartic Distortion 0.00000

□ Save Destination Image Sequence to Disk

□ Allow Modification of Image Size

Action Remove Distortion From Live Action Footage

Quality Fast Processing (Linear Interpolation)

□ Sharpen Processed Images

2.0

Sharpen Radius (Pixel)

Main Window

Miscellaneous

Source Image Sequence

Defines the image sequence to be processed by warpdistort.

Browse...

Opens a standard file requester. By choosing the first image file of an image sequence, all necessary data (such as start index, end index and filename pattern) are automatically acquired, and the respective textfields are filled out.

Pattern

Enter a file name pattern which matches the names of the sequence frames. For example, if the frame filenames are myimage.0100 ... myimage.0300 the pattern may be myimage.####.

Start, End, Step

Specify the first and the last frame file and the step size between the frames. If the parameter is set to 1 each frame is used. Example: assume the frame filenames are myimage.0100 ... myimage.0300. If the parameter Step is set to 10, only the frames myimage.0100, myimage.0110, myimage.0120, ... are used.

Res.

Displays the image format, resolution and the color depth of the sequence frames.

Interlaced

If activated, it indicates that the images of the input sequence are interlaced. During the image processing process, warpdistort will split each frame into two single fields, process each field separately, and recombine them to one processed frame.

Lens Distortion Settings

A group of textfields that define the amount of distortion to be removed or applied and the type of lens distortion.

Original FOV Size

Textfields that display the size of FOV in pixels specified within the Zoom Window of 3D-Equalizer. The lens distortion parameters in 3D-Equalizer always relate to the current FOV setting. That means, modifying the FOV size does have an effect on the actual lens distortion. Therefore, the FOV size that relates to the current lens distorion settings must be specified.

Pixel Aspect

Textfield that displays the pixel aspect of the image sequence to be processed. It is similar to the Pixel Aspect textfield of 3D-Equalizer’s Camera Adjustment Window.

Lens Center Offset

Textfields that contain the offset of the center-of-projection axis to the center of the input images. The values found in these textfields represent 3D-Equalizer’s accumulated FOV center and lens center offset values, expressed in pixels.

Lens Distortion

Textfield that displays the lens distortion of the image sequence to be processed. It is equal to the Lens Distortion textfield of 3D-Equalizer’s Camera Adjustment Window.

Anamorphic Squeeze

Textfield that displays the anamorphic squeeze of the image sequence to be processed. It is equal to the Anamorphic Squeeze textfield of 3D-Equalizer’s Camera Adjustment Window.

Curvature X/Y

Textfields that display the curvature x/y of the image sequence to be processed. It is equal to the Curvature X/Y textfield of 3D-Equalizer’s Camera Adjustment Window.

Quartic Distortion

Textfield that displays the quartic distortion parameter of the image sequence to be processed. It is equal to the Quartic Distortion textfield of 3D-Equalizer’s Camera Adjustment Window.

Save Destination Image Sequence To Disk

If activated, warpdistort will save the processed images to disk. The output image format, resolution and color depth will be equal to the input image sequence.

Browse...

Opens a standard file requester, in which the path for the images to be saved must be specified. The frame number pattern is the same as for the input image sequence. That means, ‘#’ characters are replaced by the frame number. Furthermore, the numbering of the destination images is equal to the numbering of the input images; that means the start, end and step parameters are used here as well.

Allow Modification of Image Size

If activated, warpdistort will automatically modify the size of the processed image files so that no image content is lost. Either black borders will be cropped, or image size will be increased to avoid cropping of content at the corners.

Action

An option menu to define the mode of operation. If set to “Remove Distortion From Live Action Footage”, warpdistort will apply the “inverse” distortion to the input images, that is specified in the Lens Distortion Settings textfields. If set to “Apply Distortion To CGI Sequence”, warpdistort will apply the distortion to the input images, by the values specified in the Lens Distortion Settings textfields.

Quality

An option menu that defines the image processing quality. Choose among “Fast Processing (Linear Interpolation)” and “Slow Processing (Bicubic Interpolation)”.

Sharpen Processed Images

If activated, the warped images will be sharpened using an image processing filter.

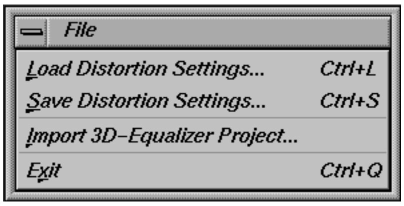
Sharpen Radius (Pixel)

Defines the amount of sharpness. A low value means less sharpness, whereas a high value means more sharpness.

Process...

Starts the image processing procedure, as specified in the Main Window’s textfields and option menus. The intermediate image processing results of each single step, can be examined in the Image View Window. The entire procedure can be interrupted at any time.

Menu: File


File
Load Distortion Settings... Ctrl+L
Save Distortion Settings... Ctrl+S
Import 3D–Equalizer Project...
Exit Ctrl+Q

Functions to store and retrieve lens distortion settings.
shortcut: ALT-f

Load Distortion Settings...

A function to retrieve warpdistort’s current lens distortion settings from a file. A standard file requester opens, which lets the user choose a file previously created with warpdistort.
see also: Save Distortion Settings...
shortcuts: l, CTRL-l

Save Distortion Settings...

A function to store warpdistort’s current lens distortion settings to disk using a binary format. A standard file requester opens allowing the user to specify a filename.
see also: Load Distortion Settings...
shortcuts: l, CTRL-l

Import 3D-Equalizer Project...

A function that enables the user to import data from a 3D-Equalizer project file. A standard file requester opens, which lets the user select a 3D-Equalizer project file. After the user has confirmed the file requester with a valid project file, another requester opens. This requester displays a list of all sequences and reference frames of the selected project.

If one entry is selected, the associated data are imported into the respective Source Image Sequence and Lens Distortion Settings fields. This function represents a very efficient way to transfer information from 3D-Equalizer to WarpDistort.
shortcut: i

Exit

Terminates WarpDistort.
shortcuts: x, CTRL-q

Image View Window

The Image View Window only consists of a 2D display area, and does not have any interaction components. The window displays the currently processed image. A white cross in the middle of the image indicates the center-of-projection axis.

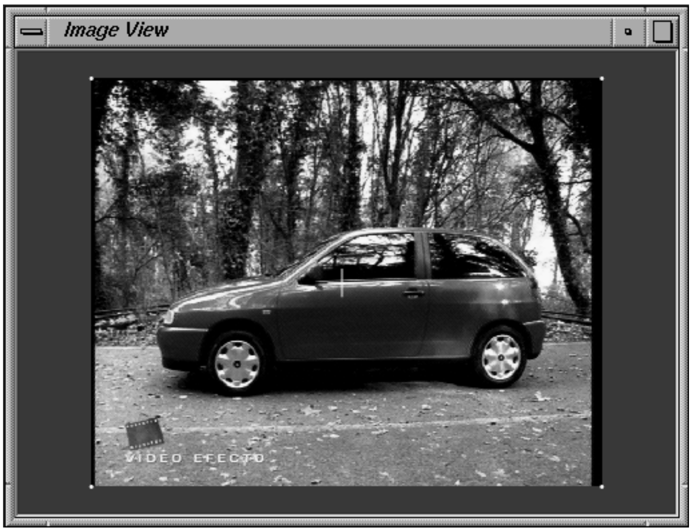


Image View Window

Appendix

Appendix A: Tcl interface

The following section describes 3D-Equalizer’s tcl interface. Tcl is a widespread, very powerful scripting language developed by John K. Ousterhout at University of California, Berkley.

The major application of 3D-Equalizer’s tcl interface is for customizing the graphical user interface and for exchanging data with other software tools. A good source of further information about tcl can be found at http://www.tcltk.com. A few 3DE example scripts can be found in the directory “<3de install directory>/user_data/tcl_archive/”.

Datatypes

<vector_2d>	= [list <x> <y>]
<vector_3d>	= [list <x> <y> <z>]
<matrix_3d>	= [list [list <m00> <m01> <m02>] [list <m10> <m11> <m12>] [list <m20> <m21> <m22>]]
<curve_2d>	= [list <vector_2d> <vector_2d> <vector_2d>...]

Command: get3DEInstallPath

Synopsis

<path> get3DEInstallPath

Parameters

<path> - an absolute filesystem path

Description

Executing this command causes 3D-Equalizer to return the path to its installation directory.

Example

```
# include “veclib.tcl”
set path [get3DEInstallPath]
source $path/user_data/tcl_archive/veclib.tcl
```

Command: get3DEVersion

Synopsis

<string> get3DEVersion

Parameters

<string>   - text string

Description

Executing this command causes 3D-Equalizer to return its version.

Example

```
# print 3DE’s version
set version_string [get3DEVersion]
open3DEConsole
print3DEConsole “Current version: “ $version_string
```

Command: open3DEConsole

Synopsis

open3DEConsole

Description

Executing this command causes 3D-Equalizer to open the tcl console window.

Example

```
# open console and print error message
open3DEConsole
print3DEConsole “ERROR!!!\n”
```

See also

close3DEConsole, print3DEConsole, flush3DEConsole

Command: close3DEConsole

Synopsis

close3DEConsole

Description

Executing this command causes 3D-Equalizer to close the tcl console window.

Example

```
close3DEConsole
```

See also

open3DEConsole, print3DEConsole, flush3DEConsole

Command: print3DEConsole

Synopsis

print3DEConsole <string1> <string2>...

Parameters

<string> - Argument that is interpreted as a string

Description

Executing this command causes 3D-Equalizer to print out the supplied parameters in the tcl console window. The actual output will take place by the time the tcl script is finished or if the command “flush3DEConsole” is executed.

Example

```
set a 1
print3DEConsole “a = “ $a “\n”
```

See also

close3DEConsole, open3DEConsole, flush3DEConsole

Command: flush3DEConsole

Synopsis

flush3DEConsole

Description

Executing this command causes 3D-Equalizer to flush the tcl console window output.

Example

```
# display the numbers 1..100
for {set i 1} {$i<=100} {set i [expr $i+1]} \
  {
  print3DEConsole “i = “ $i “\n”
  flush3DEConsole
  }
```

See also

close3DEConsole, open3DEConsole, print3DEConsole

Command: post3DEInfoRequester

Synopsis

post3DEInfoRequester <message>

Parameters

<message> - text message string

Description

On execution, a requester window opens on the screen, displaying the specified text message. The execution of the TCL script stops until the user clicks on the Ok button.

Example

```
# post info requester
post3DEInfoRequester “Selected pointgroup deleted!”
```

See also

post3DEQuestionRequester, post3DEPromptRequester, post3DEFileRequester

Command: post3DEQuestionRequester

Synopsis

<bool> post3DEQuestionRequester <message> <ok_label> <cancel_label>

Parameters

<bool>	- boolean value: 0, 1
<message>	- text message string
<ok_label>	- text string of Ok button
<cancel_label>	- text string of Cancel button

Description

On execution, a requester window opens on the screen, displaying the specified text message. The execution of the TCL script stops until the user clicks on a button. If the user clicks on the left button (Ok button), the command returns 1. Otherwise, 0 is returned.

Example

```
# post question requester asking to delete a pointgroup
set ok [post3DEQuestionRequester “Delete?” “Yes” “No”]
if {$ok}\
  {
  set pg [getSelectedPGroup]
  deletePGroup $pg
  }
```

See also

post3DEInfoRequester, post3DEPromptRequester, post3DEFileRequester

Command: post3DEPromptRequester

Synopsis

<string> post3DEPromptRequester <message>

Parameters

<string>	- text string
<message>	- text message string

Description

On execution, a requester window opens on the screen, displaying the specified text message. Furthermore, the requester contains a textfield, an Ok button and a Cancel button. The user can enter some text into the textfield. The execution of the TCL script stops until the user clicks on a button. If the user clicks on the Ok button, the command returns the string that has been entered into the textfield. Otherwise, an empty string (“”) is returned.

Example

```
# post prompt requester to ask for new pointgroup name
set name [post3DEPromptRequester “Enter name:”]
if {$name!=””}\
  {
  set pg [getSelectedPGroup]
  setPGroupName $pg $name
  }
```

See also

post3DEInfoRequester, post3DEQuestionRequester, post3DEFileRequester

Command: post3DEFileRequester

Synopsis

<path> post3DEFileRequester <message> <filter_pattern>

Parameters

<path>	- text string containing an absolute path
<message>	- text message string
<filter_pattern>	- string that defines a file pattern

Description

On execution, a standard 3D-Equalizer file requester window opens on the screen. The execution of the TCL script stops until the user closes this file requester. If the user has specified a filename, the absolute path of this file is returned. If the user clicks on the Cancel button to close the requester, an empty string (“”) is returned.

Example

```
# post file requester to ask for a file
set path [post3DEFileRequester “Save As...” “*”]
if {$path!=””}\
  {
  open3DEConsole
  print3DEConsole “Saving data to file: “ $name “...\n”
  flush3DEConsole

  ...
  }
```

See also

post3DEInfoRequester, post3DEQuestionRequester, post3DEPromptRequester

Command: getSelectedPGroup

Synopsis

<pgroup_id> getSelectedPGroup

Parameters

<pgroup_id> - identifier of a pointgroup object

Description

On execution this command returns the pointgroup id of the pointgroup object that is currently selected in the Pointgroups Window. If no pointgroup object is currently selected, a warning message is displayed in the tcl console window.

Example

```
set pg [getSelectedPGroup]
```

See also

deletePGroup, getNextPGroup, getPGroupName, getPGroupType

Command: getSelectedFobj

Synopsis

<fobj_id> getSelectedFobj

Parameters

<fobj_id> - identifier of a frame object

Description

On execution this command returns the frame object id of the sequence or reference frame object that is currently selected in the Frames Window. If no frame object is currently selected, a warning message is displayed in the tcl console window.

Example

```
set fobj [getSelectedFobj]
```

See also

deleteFobj, getNextFobj, getFobjName

Command: getSelectedPoint

Synopsis

<point_id> getSelectedPoint

Parameters

<point_id> - identifier of a point object

Description

On execution this command returns the point id of the point object that is currently selected in the Pointgroups Window. If no point object is currently selected, a warning message is displayed in the tcl console window.
This command is obsolete. Use *getPointSelectionFlag* instead!

Example

```
set point [getSelectedPoint]
```

See also

deletePoint, getNextPoint, getPointName, getPointSelectionFlag

Command: getSelectedCamera

Synopsis

<camera_id> getSelectedCamera

Parameters

<camera_id>	- identifier of a camera object

Description

On execution this command returns the camera id of the camera object that is currently selected in the Camera Adjustment Window. If no camera object is currently selected, a warning message is displayed in the tcl console window.

Example

```
set camera [getSelectedCamera]
```

See also

getNextCamera, getCameraName

Command: getSelectedFrame

Synopsis

<frame> getSelectedFrame

Parameters

<frame> - frame number

Description

On execution this command returns the number of the currently selected frame specified with the Main Window’s frame slider. If no frame object is currently selected, a warning message is displayed in the tcl console window.

Example

```
set frame [getSelectedFrame]
```

Command: createPGroup

Synopsis

<pgroup_id> createPGroup <pgroup_type>

Parameters

<pgroup_id>	- identifier of a pointgroup object
<pgroup_type>	- constant that defines the pointgroup type: CAMERA, OBJECT, MOCAP

Description

On execution this command creates a new pointgroup object and returns its pointgroup id.

Example

```
# create a new object pointgroup
set pg [createPGroup OBJECT]
```

See also

deletePGroup, getNextPGroup, getPGroupName

Command: deletePGroup

Synopsis

deletePGroup <pgroup_id>

Parameters

<pgroup_id>   - identifier of a pointgroup object

Description

On execution this command deletes the specified pointgroup object.

Example

```
# delete the currently selected pointgroup
set pg [getSelectedPGroup]
deletePGroup $pg
```

See also

createPGroup, getNextPGroup, getPGroupName

Command: getPGroupType

Synopsis

<pgroup_type> getPGroupType <pgroup_id>

Parameters

<pgroup_type> - constant that defines the pointgroup type: CAMERA, OBJECT, MOCAP

<pgroup_id> - identifier of a pointgroup object

Description

On execution this command returns the type of the specified pointgroup object.

Example

```
# display type of the currently selected pointgroup
set pg [getSelectedPGroup]
set pg_type [getPGroupType $pg]
print3DEConsole “Pointgroup type: “ $pg_type “\n”
```

See also

createPGroup, deletePGroup, getNextPGroup, getPGroupName

Command: getFirstPGroup

Synopsis

<pgroup_id> getFirstPGroup

Parameters

<pgroup_id>	- identifier of a pointgroup object

Description

On execution this command returns the pointgroup id of the first pointgroup object. If there are no pointgroup objects available, the string “0” is returned.

Example

```
set pg [getFirstPGroup]
```

See also

createPGroup, deletePGroup, getNextPGroup, getPGroupName

Command: getNextPGroup

Synopsis

<pgroup_id> getNextPGroup <pgroup_id>

Parameters

<pgroup_id> - identifier of a pointgroup object

Description

On execution this command returns the pointgroup id of the pointgroup object that comes next after the specified pointgroup object. If the specified pointgroup object is the last one, the string “0” is returned.

Example

```
# display the names of all pointgroups
for {set pg [getFirstPGroup]} \
    {$pg != 0} \
    {set pg [getNextPGroup $pg]} \
  {
  set name [getPGroupName $pg]
  print3DEConsole “Pointgroup name: “ $name “\n”
  }
```

See also

createPGroup, deletePGroup, getPGroupName

Command: getNoPGroups

Synopsis

<number_of_pgroups> getNoPGroups

Parameters

<number_of_pgroups>	- integer that specifies the number of pointgroup objects

Description

On execution this command returns the number of available pointgroup objects.

Example

```
set no_pgroups [getNoPGroups]
```

See also

getIndexPGroup

Command: getIndexPGroup

Synopsis

<pgroup_id> getIndexPGroup <pgroup_index>

Parameters

<pgroup_id>	- identifier of a pointgroup object
<pgroup_index>	- integer that specifies a pointgroup object

Description

On execution this command returns the pointgroup id of a pointgroup object specified by the pointgroup index parameter. The first pointgroup object is defined by an index of “0”.

Example

```
# display the names of all pointgroups
for {set index 0} \
    {$index < [getNoPGroups]} \
    {set index [expr $index+1]} \
  {
  set pg [getIndexPGroup $index]
  set name [getPGroupName $pg]
  print3DEConsole “Pointgroup name: “ $name “\n”
  }
```

See also

getNoPGroups

Command: getPGroupName

Synopsis

<name> getPGroupName <pgroup_id>

Parameters

<name>	- name string
<pgroup_id>	- identifier of a pointgroup object

Description

On execution this command returns the name of the pointgroup object specified by the supplied pointgroup id.

Example

```
# display name of the currently selected pointgroup
set pg [getSelectedPGroup]
set name [getPGroupName $pg]
print3DEConsole “Pointgroup name: “ $name “\n”
```

See also

setPGroupName

Command: setPGroupName

Synopsis

setPGroupName <pgroup_id> <name>

Parameters

<pgroup_id>	- identifier of a pointgroup object
<name>	- name string

Description

On execution this command sets the name of the given pointgroup object to the supplied name string.

Example

```
# set name of the currently selected pointgroup to “wurst”
set pg [getSelectedPGroup]
setPGroupName $pg “wurst”
```

See also

getPGroupName

Command: getPGroupDeviation

Synopsis

<double> getPGroupDeviation <pgroup_id> <fobj_id> <frame>

Parameters

<double>	- floating point value
<pgroup_id>	- identifier of a pointgroup object
<fobj_id>	- identifier of a frame object
<frame>	- integer value that defines a frame number

Description

On execution this command returns the frame-deviation of the given pointgroup object, tracked in the specified sequence.

Example

```
# print deviation of pgroup „pg“
set pg [getSelectedPGroup]
set fobj [getSelectedFobj]
set no_frames [getFobjNoFrames $fobj]
for {set f 1} {$f <= $no_frames} {set f [expr $f +1]}\
  {
  set dev [getPGroupDeviation $pg $fobj $f]
  print3DEConsole $dev
  }
```

See also

getPGroupAverageDeviation

Command: getPGroupAverageDeviation

Synopsis

<double> getPGroupAverageDeviation <pgroup_id> <fobj_id>

Parameters

<double>	- floating point value
<pgroup_id>	- identifier of a pointgroup object
<fobj_id>	- identifier of a frame object

Description

On execution this command returns the average deviation of the given pointgroup object, tracked in the specified sequence.

Example

```
# print av. deviation of pgroup „pg“
set pg [getSelectedPGroup]
set fobj [getSelectedFobj]
print3DEConsole [getPGroupAverageDeviation $pg $fobj]
```

See also

getPGroupDeviation

Command: createFobj

Synopsis

<fobj_id> createFobj <fobj_type>

Parameters

<fobj_id>	- identifier of a frame object
<fobj_type>	- constant that defines the frame object type: SEQUENCE, REF_FRAME

Description

On execution this command creates a new frame object and returns its id.

Example

```
# create a new sequence
set fobj [createFobj SEQUENCE]
```

See also

deleteFobj, getNextFobj, getFobjName

Command: deleteFobj

Synopsis

deleteFobj <fobj_id>

Parameters

<fobj_id> - identifier of a frame object

Description

On execution this command deletes the specified frame object.

Example

```
# delete the currently selected frame object
set fobj [getSelectedFobj]
deleteFobj $fobj
```

See also

createFobj, getNextFobj, getFobjName

Command: getFobjType

Synopsis

<fobj_type> getFobjType <fobj_id>

Parameters

<fobj_type> - constant that defines the frame object type: SEQUENCE, REF_FRAME

<fobj_id> - identifier of a pointgroup object

Description

On execution this command returns the type of the specified frame object.

Example

```
# display type of the currently selected frame object
set $fobj [getSelectedFobj]
set $fobj_type [getFobjType $fobj]
if {$fobj_type==”SEQUENCE”} then \
  {
  print3DEConsole “This is an image sequence!\n”
  } \
else \
  {
  print3DEConsole “This is a reference frame!\n”
  }
```

See also

createFobj, deleteFobj, getNextFobj, getFobjName

Command: getFirstFobj

Synopsis

<fobj_id> getFirstFobj

Parameters

<fobj_id> - identifier of a frame object

Description

On execution this command returns the frame object id of the first frame object. If there are no frame objects available, the string “0” is returned.

Example

```
set fobj [getFirstFobj]
```

See also

createFobj, deleteFobj, getNextFobj, getFobjName

Command: getNextFobj

Synopsis

<fobj_id> getNextFobj <fobj_id>

Parameters

<fobj_id> - identifier of a frame object

Description

On execution this command returns the frame object id of the frame object that comes next after the specified frame object. If the specified frame object is the last one, the string “0” is returned.

Example

```
# display the names of all frame objects
for {set fobj [getFirstFobj]} \
    {$fobj != 0} \
    {set fobj [getNextFobj $fobj]} \
  {
  set name [getFobjName $fobj]
  print3DEConsole “Frame object name: “ $name “\n”
  }
```

See also

createFobj, deleteFobj, getFobjName

Command: getNoFobjs

Synopsis

<number_of_fobjs> getNoFobjs

Parameters

<number_of_fobjs> - integer that specifies the number of frame objects

Description

On execution this command returns the number of available frame objects.

Example

```
set no_fobjs [getNoFobjs]
```

See also

getIndexFobj

Command: getIndexFobj

Synopsis

<fobj_id> getIndexFobj <fobj_index>

Parameters

<fobj_id>	- identifier of a frame object
<fobj_index>	- integer that specifies a frame object

Description

On execution this command returns the frame object id of a frame object specified by the frame object index parameter. The first frame object is defined by an index of “0”.

Example

```
# display the names of all frame objects
for {set index 0} \
    {$index < [getNoFobjs]} \
    {set index [expr $index+1]} \
  {
  set fobj [getIndexFobj $index]
  set name [getFobjName $fobj]
  print3DEConsole “Frame object name: “ $name “\n”
  }
```

See also

getNoFobjs

Command: getFobjName

Synopsis

<name> getFobjName <fobj_id>

Parameters

<name>	- name string
<fobj_id>	- identifier of a frame object

Description

On execution this command returns the name of the frame object specified by the supplied frame object id.

Example

```
# display name of the currently selected frame object
set fobj [getSelectedFobj]
set name [getFobjName $fobj]
print3DEConsole “Frame object name: “ $name “\n”
```

See also

setFobjName

Command: setFobjName

Synopsis

setFobjName <fobj_id> <name>

Parameters

<fobj_id>	- identifier of a frame object
<name>	- name string

Description

On execution this command sets the name of the given frame object to the supplied name string.

Example

```
# set name of the currently selected frame object to “willi”
set fobj [getSelectedFobj]
setFobjName $fobj “willi”
```

See also

getFobjName

Command: getFobjNoFrames

Synopsis

<no_frames> getFobjNoFrames <fobj_id>

Parameters

<no_frames>	- integer value
<fobj_id>	- identifier of a frame object

Description

On execution this command returns the number of frames of the frame object specified by the supplied frame object id. If the frame object is a reference frame then “1” is returned.

Example

```
# display number of frames of the
# currently selected frame object
set fobj [getSelectedFobj]
set no_frames [getFobjNoFrames $fobj]
print3DEConsole “Number of frames: “ $no_frames “\n”
```

See also

setFobjSequenceAttr, getFobjSequenceAttr

Command: getFobjPath

Synopsis

<path> getFobjPath <fobj_id>

Parameters

<path>	- absolute filename path starting from “/”
<fobj_id>	- identifier of a frame object

Description

On execution this command returns the absolute image filename path of the frame object specified by the supplied frame object id. All indexing numbers in the path are replaced by ‘#’ characters, similar to the “Sequence Settings” requester.

Example

```
# display the path of the currently selected frame object
set fobj [getSelectedFobj]
set path [getFobjPath $fobj]
print3DEConsole “Image path: “ $path “\n”
```

See also

setFobjPath

Command: setFobjPath

Synopsis

setFobjPath <fobj_id> <path>

Parameters

<fobj_id>	- identifier of a frame object
<path>	- absolute filename path starting from “/”

Description

On execution this command redefines the absolute image filename path of the frame object specified by the supplied frame object id. All indexing numbers in the path must be replaced by ‘#’ characters, similar to the “Sequence Settings” requester.

Example

```
# set image filename path of the currently selected
# frame object to ‘/data/images/dog.####.jpg’
set fobj [getSelectedFobj]
setFobjPath $fobj “/data/images/dog.####.jpg”
```

See also

getFobjPath

Command: getFobjSequenceAttr

Synopsis

[list <start> <end> <step>] getFobjSequenceAttr <fobj_id>

Parameters

<start> - integer value defining the start index of an image sequence
<end> - integer value defining the end index of an image sequence
<step> - integer value defining the index increment of an image sequence
<fobj_id> - identifier of a frame object

Description

On execution this command returns indexing information of the frame object specified by the supplied frame object id. The return value consists of a list containing three integer numbers, representing the content of the Start, End, Step textfields found in the “Sequence Settings” requester.

Example

```
# display the sequence attributes of the
# currently selected frame object
set fobj [getSelectedFobj]
set attr_list [getFobjSequenceAttr $fobj]
print3DEConsole “Start index: “ [lindex $attr_list 0] “\n”
print3DEConsole “End index: “ [lindex $attr_list 1] “\n”
print3DEConsole “Stepsize: “ [lindex $attr_list 2] “\n”
```

See also

setFobjSequenceAttr

Command: setFobjSequenceAttr

Synopsis

setFobjSequenceAttr <fobj_id> <start> <end> <step>

Parameters

<fobj_id>	- identifier of a frame object
<start>	- integer value defining the start index of an image sequence
<end>	- integer value defining the end index of an image sequence
<step>	- integer value defining the index increment of an image sequence

Description

On execution this command redefines the indexing attributes of the frame object specified by the supplied frame object id. Executing this command on a reference frame has no effect.

Example

```
# set the indexing attributes of the currently selected
# frame object to 1 - 55, stepsize 2
set fobj [getSelectedFobj]
setFobjSequenceAttr $fobj 1 55 2
```

See also

getFobjSequenceAttr

Command: getFobjImageType

Synopsis

<image_type> getFobjImageType <fobj_id>

Parameters

<image_type> - constant that defines the image type of a frame object: NON_LACE, LACE_EVEN, LACE_ODD, FIELD, ANAMORPHIC

<fobj_id> - identifier of a frame object

Description

On execution this command returns the image type of the frame object specified by the supplied frame object id.

Example

```
# display the image type of the currently
# selected frame object
set fobj [getSelectedFobj]
set imgtype [getFobjImageType $fobj]
print3DEConsole “Image type: “ $imgtype “\n”
```

See also

setFobjImageType

Command: setFobjImageType

Synopsis

setFobjImageType <fobj_id> <image_type>

Parameters

<fobj_id>	- identifier of a frame object
<image_type>	- constant that defines the image type of a frame object: NON_LACE, LACE_EVEN, LACE_ODD, FIELD, ANAMORPHIC

Description

On execution this command redefines the image type of the frame object specified by the supplied frame object id.

Example

```
# set the image type of the currently selected
# frame object to “Noninterlace”
set fobj [getSelectedFobj]
setFobjImageType $fobj NON_LACE
```

See also

getFobjImageType

Command: getFobjCamera

Synopsis

<camera_id> getFobjCamera <fobj_id>

Parameters

<camera_id>	- identifier of a camera object
<fobj_id>	- identifier of a frame object

Description

On execution this command returns the identifier of the camera object that is linked to the frame object specified by the supplied frame object id.

Example

```
# display the name of the camera object linked to the
# currently selected frame object
set fobj [getSelectedFobj]
set camera [getFobjCamera $fobj]
set name [getCameraName $camera]
print3DEConsole “Camera name: “ $name “\n”
```

See also

setFobjCamera

Command: setFobjCamera

Synopsis

setFobjCamera <fobj_id> <camera_id>

Parameters

<camera_id>	- identifier of a camera object
<fobj_id>	- identifier of a frame object

Description

On execution this command links the specified camera object to the frame object specified by the supplied frame object id.

Example

```
# link selected camera to selected sequence
set fobj [getSelectedFobj]
set camera [getSelectedCamera]
setFobjCamera $fobj $camera
```

See also

getFobjCamera

Command: getFobjZoomingFlag

Synopsis

<bool> getFobjZoomingFlag <fobj_id>

Parameters

<bool>	- boolean value: O, 1
<fobj_id>	- identifier of a frame object

Description

On execution this command returns the status of the “Variable Focal Length” flag of the frame object specified by the supplied frame object id.

Example

```
# display whether zooming is enabled or disabled with the
# currently selected frame object
set fobj [getSelectedFobj]
set zooming [getFobjZoomingFlag $fobj]
if {$zooming} then \
  {
  print3DEConsole “Zooming enabled!\n”
  } \
else \
  {
  print3DEConsole “Zooming disabled!\n”
  }
```

See also

getFobjZoomFactor

Command: getFobjZoomFactor

Synopsis

<float> getFobjZoomFactor <fobj_id> <frame>

Parameters

<float>	- floating point value
<fobj_id>	- identifier of a frame object
<frame>	- integer value that defines a frame number

Description

On execution this command returns the zoom factor of the specified frame object in the specified frame. Zoom factor values must be multiplied with the focal length value of a camera object to obtain the “real” zoomed focal length.

Example

```
# display all zoom factors of the
# currently selected sequence
set fobj [getSelectedFobj]
for {set frame 1} \
    {$frame<=[getNoFrames $fobj]} \
    {set frame [expr $frame+1]} \
  {
  set zoom_factor [getFobjZoomFactor $fobj $frame]
  print3DEConsole “Frame: “ $frame
  print3DEConsole “, zoom factor: “ $zoom_factor “\n”
  }
```

See also

getFobjZoomingFlag

Command: getFobjImageWidth

Synopsis

<image_width> getFobjImageWidth <fobj_id>

Parameters

<image_width>	- integer value representing a horizontal image resolution in pixel
<fobj_id>	- identifier of a frame object

Description

On execution this command returns the horizontal image resolution of the specified frame object.

Example

```
# display the image width of the selected frame object
set fobj [getSelectedFobj]
set width [getFobjImageWidth $fobj]
print3DEConsole “Image width: “ $width “\n”
```

See also

getFobjImageHeight

Command: getFobjImageHeight

Synopsis

<image_height> getFobjImageHeight <fobj_id>

Parameters

<image_height>	- integer value representing a vertical image resolution in pixel
<fobj_id>	- identifier of a frame object

Description

On execution this command returns the vertical image resolution of the specified frame object.

Example

```
# display the image height of the selected frame object
set fobj [getSelectedFobj]
set height [getFobjImageHeight $fobj]
print3DEConsole “Image height: “ $height “\n”
```

See also

getFobjImageWidth

Command: getFobjImageWidthFOV

Synopsis

<image_width_fov> getFobjImageWidthFOV <fobj_id>

Parameters

<image_width_fov>	- floating point value representing a horizontal FOV size in pixel
<fobj_id>	- identifier of a frame object

Description

On execution this command returns the horizontal FOV size of the specified frame object.

Example

```
# display the fov width of the selected frame object
set fobj [getSelectedFobj]
set width [getFobjImageWidthFOV $fobj]
print3DEConsole “FOV width: “ $width “\n”
```

See also

getFobjImageHeightFOV

Command: getFobjImageHeightFOV

Synopsis

<image_height_fov> getFobjImageHeightFOV <fobj_id>

Parameters

<image_height_fov>	- floating point value representing a vertical FOV size in pixel
<fobj_id>	- identifier of a frame object

Description

On execution this command returns the vertical FOV size of the specified frame object.

Example

```
# display the fov height of the selected frame object
set fobj [getSelectedFobj]
set height [getFobjImageHeightFOV $fobj]
print3DEConsole “FOV height: “ $height “\n”
```

See also

getFobjImageWidthFOV

Command: getFobjTimeshift

Synopsis

<time_shift> getFobjTimeshift <fobj_id>

Parameters

<time_shift>	- floating point value representing a subframe timeshift value
<fobj_id>	- identifier of a frame object

Description

On execution this command returns the subframe timeshift value of the specified frame object.

Example

```
# display the timeshift of the selected frame object
set fobj [getSelectedFobj]
print3DEConsole [getFobjTimeshift $fobj]
```

Command: createPoint

Synopsis

<point_id> createPGroup <pgroup_id>

Parameters

<point_id>	- identifier of a point object
<pgroup_id>	- identifier of a pointgroup object

Description

On execution this command creates a new point object in the specified pointgroup and returns its point object id.

Example

```
# create a new point in the currently selected pointgroup
set pg [getSelectedPGroup]
set point [createPoint $pg]
```

See also

deletePoint, getNextPoint, getPointName

Command: deletePoint

Synopsis

deletePoint <pgroup_id> <point_id>

Parameters

<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object

Description

On execution this command deletes the specified point object.

Example

```
# delete the currently selected point
set pg [getSelectedPGroup]
set point [getSelectedPoint]
deletePGroup $pg $point
```

See also

createPoint, getNextPoint, getPointName

Command: getFirstPoint

Synopsis

<point_id> getFirstPoint <pgroup_id>

Parameters

<point_id>	- identifier of a point object
<pgroup_id>	- identifier of a pointgroup object

Description

On execution this command returns the point object id of the first point in the specified pointgroup. If there are no point objects available, the string “0” is returned.

Example

```
set point [getFirstPoint $pg]
```

See also

createPoint, deletePoint, getNextPoint, getPointName

Command: getNextPoint

Synopsis

<point_id> getNextPoint <pgroup_id> <point_id>

Parameters

<point_id>	- identifier of a point object
<pgroup_id>	- identifier of a pointgroup object

Description

On execution this command returns the point id of the point that comes next after the specified point in the specified pointgroup. If the specified point is the last one, the string “0” is returned.

Example

```
# display the names of all points of the selected pointgroup
set pg [getSelectedPGroup]
for {set point [getFirstPoint $pg]} \
    {$point != 0} \
    {set point [getNextPoint $pg $point]} \
  {
  set name [getPointName $pg $point]
  print3DEConsole “Point name: “ $name “\n”
  }
```

See also

createPoint, deletePoint, getPointName

Command: getNoPoints

Synopsis

<number_of_points> getNoPoints <pgroup_id>

Parameters

<number_of_points>	- integer that specifies the number of point objects
<pgroup_id>	- identifier of a pointgroup object

Description

On execution this command returns the number of available points in the specified pointgroup.

Example

```
set no_points [getNoPoints $pg]
```

See also

getIndexPoint

Command: getIndexPoint

Synopsis

<point_id> getIndexPoint <pgroup_id> <point_index>

Parameters

<point_id>	- identifier of a point object
<pgroup_id>	- identifier of a pointgroup object
<point_index>	- integer that specifies a point object

Description

On execution this command returns the id of the point specified by the point index parameter in the specified pointgroup. The first point object is defined by an index of “0”.

Example

```
# display the names of all points in the selected pointgroup
set pg [getSelectedPGroup]
for {set index 0} \
    {$index < [getNoPoints $pg]} \
    {set index [expr $index+1]} \
  {
  set point [getIndexPGroup $pg $index]
  set name [getPointName $pg $point]
  print3DEConsole “Point name: “ $name “\n”
  }
```

See also

getNoPoints

Command: findPointByName

Synopsis

<point_id> findPointByName <pgroup_id> <name>

Parameters

<point_id>	- identifier of a point object
<pgroup_id>	- identifier of a pointgroup object
<name>	- name string

Description

On execution this command returns the id of the point in the specified pointgroup that matches the specified name. If no point object could be found, a warning message is displayed and the string “0” is returned.

Example

```
# search point “wurst” in the selected pointgroup
set pg [getSelectedPGroup]
set point [findPointByName $pg “wurst”]
if {$point!=0}\
  {
  print3DEConsole “Point wurst has been found!\n”
  } \
else \
  {
  print3DEConsole “Point wurst has not been found!\n”
  }
```

See also

setPointName, getPointName

Command: getPointName

Synopsis

<name> getPointName <pgroup_id> <point_id>

Parameters

<name>	- name string
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object

Description

On execution this command returns the name of the specified point in the specified pointgroup.

Example

```
# display name of the currently selected point
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set name [getPointName $pg $point]
print3DEConsole “Point name: “ $name “\n”
```

See also

setPointName

Command: setPointName

Synopsis

setPointName <pgroup_id> <point_id> <name>

Parameters

<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<name>	- name string

Description

On execution this command sets the name of the specified point in the specified pointgroup to the supplied name string.

Example

```
# set name of the currently selected point to “honsel”
set pg [getSelectedPGroup]
set point [getSelectedPoint]
setPointName $pg $point “honsel”
```

See also

getPointName

Command: getPointSelectionFlag

Synopsis

<bool> getPointSelectionFlag <pgroup_id> <point_id>

Parameters

<bool>	- boolean value: O, 1
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object

Description

On execution this command returns the string “1”, if the specified point is currently selected. Otherwise string “0” is returned.

Example

```
# display the names of all selected points in the
# currently selected pointgroup
set pg [getSelectedPGroup]
for {set point [getFirstPoint $pg]} \
     {$point!=0} \
     {set point [getNextPoint $pg $point]} \
  {
  if {[getPointSelectionFlag $pg $point]} \
      {
      set name [getPointName $pg $point]
      print3DEConsole “Point “ $name “ is selected.\n”
      }
  }
```

See also

getSelectedPoint

Command: getPointPosition3D

Synopsis

<vector_3d> getPointPosition3D <pgroup_id> <point_id>

Parameters

<vector_3d>	- three-dimensional floating point vector
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object

Description

On execution this command returns the user-defined 3D point position of the specified point in the specified pointgroup.

Example

```
# display the user-defined 3D position of the currently
# selected point
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set vec3d [getPointPosition3D $pg $point]
print3DEConsole “3D point position: “ $vec3d “\n”
```

See also

setPointPosition3D

Command: setPointPosition3D

Synopsis

setPointPosition3D <pgroup_id> <point_id> <vector_3d>

Parameters

<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<vector_3d>	- three-dimensional floating point vector

Description

On execution this command redefines the user-defined 3D point position of the specified point in the specified pointgroup.

Example

```
# set the user-defined 3D position of the currently
# selected point to (0,0,0)
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set vec3d [list 0 0 0]
setPointPosition3D $pg $point $vec3d
```

See also

getPointPosition3D

Command: getPointCalcPosition3D

Synopsis

<vector_3d> getPointCalcPosition3D <pgroup_id> <point_id>

Parameters

<vector_3d>	- three-dimensional floating point vector
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object

Description

On execution this command returns the calculated 3D point position of the specified point in the specified pointgroup.

Example

```
# display the calculated 3D position of the currently
# selected point
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set vec3d [getPointCalcPosition3D $pg $point]
print3DEConsole “3D point position: “ $vec3d “\n”
```

See also

setPointCalcPosition3D

Command: setPointCalcPosition3D

Synopsis

setPointPosition3D <pgroup_id> <point_id> <vector_3d>

Parameters

<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<vector_3d>	- three-dimensional floating point vector

Description

On execution this command redefines the calculated 3D point position of the specified point in the specified pointgroup.

Example

```
# set the calculated 3D position of the currently
# selected point to (0,0,0)
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set vec3d [list 0 0 0]
setPointCalcPosition3D $pg $point $vec3d
```

See also

getPointCalcPosition3D

Command: getPointMoCapCalcPosition3D

Synopsis

<vector_3d> getPointMoCapCalcPosition3D <pgroup_id> <point_id> <fobj_id> <f>

Parameters

<vector_3d>	- three-dimensional floating point vector
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<fobj_id>	- identifier of a frame object
<f>	- integer value that defines a frame number

Description

On execution this command returns the calculated 3D point position of the specified point in the specified mocap pointgroup, tracked in the specified frame of the specified sequence.

Example

```
# display the calculated 3D position of the currently
# selected mocap point, in frame 1 of the first sequence
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set fobj [getFirstFobj]
set vec3d [getPointMoCapCalcPosition3D $pg $point $fobj 1]
print3DEConsole “3D point position: “ $vec3d “\n”
```

See also

getPGroupType

Command: setPointMoCapCalcPosition3D

Synopsis

setPointMoCapCalcPosition3D <pgroup_id> <point_id> <fobj_id> <f> <vector_3d>

Parameters

<vector_3d>	- three-dimensional floating point vector
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<fobj_id>	- identifier of a frame object
<f>	- integer value that defines a frame number

Description

On execution this command sets the calculated 3D point position of the specified point in the specified mocap pointgroup, tracked in the specified frame of the specified sequence to the specified 3D position.

Example

```
# set the calculated 3D position of the currently
# selected mocap point, in frame 1 of the first sequence
# to (0.0, 0.0, 0.0)
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set fobj [getFirstFobj]
setPointMoCapCalcPosition3D $pg $point $fobj 1 [list 0 0 0]
```

See also

getPGroupType

Command: getPointSurveyType

Synopsis

<survey_type> getPointSurveyType <pgroup_id> <point_id>

Parameters

<survey_type>	- constant that defines a point’s survey type: SURVEY_FREE, SURVEY_APPROX, SURVEY_EXACT
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object

Description

On execution this command returns the survey type of the specified point in the specified pointgroup.

Example

```
# display the survey type of the currently selected point
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set survey_type [getPointSurveyType $pg $point]
print3DEConsole “Survey type: “ $survey_type “\n”
```

See also

setPointSurveyType

Command: setPointSurveyType

Synopsis

setPointSurveyType <pgroup_id> <point_id> <survey_type>

Parameters

<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<survey_type>	- constant that defines a point's survey type: SURVEY_FREE, SURVEY_APPROX, SURVEY_EXACT

Description

On execution this command redefines the survey type of the specified point in the specified pointgroup.

Example

```
# set the survey type of the currently selected point to
# ‘exactly surveyed point’
set pg [getSelectedPGroup]
set point [getSelectedPoint]
setPointSurveyType $pg $point SURVEY_EXACT
```

See also

getPointSurveyType

Command: getPointPosition2D

Synopsis

<vector_2d> getPointPosition2D <pgroup_id> <point_id> <fobj_id> <frame>

Parameters

<vector_2d>	- two-dimensional floating point vector; (0,0) means the lower left image corner; (1,1) means the upper right image corner
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<fobj_id>	- identifier of a frame object
<frame>	- integer value that defines a frame number

Description

On execution this command returns the 2D tracking point position of the specified point in the specified pointgroup, tracked in the specified frame of the specified frame object. This command is not very efficient. For handling large amounts of 2D tracking data use “get/setPointPosition2DBlock” instead.

Example

```
# display the 2D tracking curve of the selected point
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set fobj [getSelectedFobj]
for {set frame 1} {$frame<=[getNoFrames $fobj]} \
    {set frame [expr $frame+1]} \
  {
  set vec2d [getPointPosition2D $pg $point $fobj $frame]
  print3DEConsole “Frame: “ $frame “,” vec2d “\n”
  }
```

See also

setPointPosition2D, setPointPosition2DBlock, getPointPosition2DBlock

Command: setPointPosition2D

Synopsis

setPointPosition2D <pgroup_id> <point_id> <fobj_id> <frame> <vector_2d>

Parameters

<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<fobj_id>	- identifier of a frame object
<frame>	- integer value that defines a frame number
<vector_2d>	- two-dimensional floating point vector; (0,0) means the lower left image corner; (1,1) means the upper right image corner

Description

On execution this command sets the 2D tracking position of the specified point in the specified pointgroup, tracked in the specified frame of the specified frame object, to the specified 2D vector. This command is not very efficient. For handling large amounts of 2D tracking data use “get/setPointPosition2DBlock” instead.

Example

```
# set the 2D tracking point position of the selected point
# in frame 13 of the selected frame object to (0.45,0.74)
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set fobj [getSelectedFobj]
setPointPosition2D $pg $point $fobj 13 [list 0.45 0.74]
```

See also

getPointPosition2D, setPointPosition2DBlock, getPointPosition2DBlock

Command: getPointPosition2DBlock

Synopsis

<curve_2d> getPointPosition2DBlock <pgroup_id> <point_id> <fobj_id> <start> <end>

Parameters

<curve_2d>	- list of two-dimensional floating point vectors; a (-1,-1) 2D vector element represents an invalid tracking point
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<fobj_id>	- identifier of a frame object
<start>	- integer value that defines a frame number
<end>	- integer value that defines a frame number

Description

On execution this command returns a list of 2D tracking point positions of the specified point in the specified pointgroup, tracked in the specified frame object. The first 2D vector in the list represents the tracking position of the specified start frame number.

Example

```
# display the 2D tracking curve of the selected point
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set fobj [getSelectedFobj]
set n [getNoFrames $fobj]
set curve [getPointPosition2DBlock $pg $point $fobj 1 $n]
print3DEConsole “Tracking curve: “ $curve “\n”
```

See also

setPointPosition2D, getPointPosition2D, setPointPosition2DBlock

Command: setPointPosition2DBlock

Synopsis

setPointPosition2DBlock <pgroup_id> <point_id> <fobj_id> <start> <curve_2d>

Parameters

<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<fobj_id>	- identifier of a frame object
<start>	- integer value that defines a frame number
<curve_2d>	- list of two-dimensional floating point vectors; a (-1,-1) 2D vector element represents an invalid tracking point

Description

On execution this command copies the 2D tracking point positions contained in the specified curve to the tracking curve of the specified point in the specified pointgroup, of the specified frame object. The 2D tracking curve is copied to the point's curve starting at the specified start frame number.

Example

```
# copy the 2D tracking curve in frame object fobj from
# point p1 to point p2 (pointgroup pg)
set n [getNoFrames $fobj]
set curve [getPointPosition2DBlock $pg $p1 $fobj 1 $n]
setPointPosition2DBlock $pg $p2 $fobj 1 $curve
```

See also

setPointPosition2D, getPointPosition2D, getPointPosition2DBlock

Command: isPointPos2DValid

Synopsis

<bool> isPointPos2DValid <pgroup_id> <point_id> <fobj_id> <frame>

Parameters

<bool>	- boolean value: O, 1
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<fobj_id>	- identifier of a frame object
<frame>	- integer value that defines a frame number

Description

On execution this command returns the visibility status of the 2D tracking curve of the specified point in the specified pointgroup, tracked in the specified frame object. If the tracking point in that frame is visible, the string “1” is returned. Otherwise string “0” is returned.

Example

```
# display the visibility status of the selected point’s
# 2D tracking curve
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set fobj [getSelectedFobj]
for {set frame 1} {$frame<=[getNoFrames $fobj]} \
    {set frame [expr $frame+1]} \
  {
  if {[isPointPos2DValid $pg $point $fobj $frame]} then\
     print3DEConsole “Frame: “ $frame “ is valid.\n”
  }
```

See also

setPointPosition2D, getPointPosition2D

Command: deletePointCurve2D

Synopsis

deletePointCurve2D <pgroup_id> <point_id> <fobj_id>

Parameters

<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<fobj_id>	- identifier of a frame object

Description

On execution this command deletes the entire 2D tracking curve of the specified point in the specified pointgroup, tracked in the specified frame object.

Example

```
# remove the selected point’s 2D tracking curve
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set fobj [getSelectedFobj]
deletePointCurve2D $pg $point $fobj
```

See also

setPointPosition2DBlock, getPointPosition2DBlock

Command: isPointCalculated3D

Synopsis

<bool> isPointCalculated3D <pgroup_id> <point_id>

Parameters

<bool>	- boolean value: O, 1
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object

Description

On execution this command returns the calculation status of the specified point in the specified pointgroup. If the point's 3D position has been calculated, the string "1" is returned. Otherwise string "0" is returned.

Example

```
# display the calculatation status of all points in the
# currently selected pointgroup
set pg [getSelectedPGroup]
for {set point [getFirstPoint $pg]} \
    {$point!=0} \
    {set point [getNextPoint $pg $point]} \
  {
  if {[isPointCalculated3D $pg $point]} \
     {
     set name [getPointName $pg $point]
     print3DEConsole “Point “ $name “ is calculated.\n”
     }
  }
```

See also

setPointCalcPosition3D, getPointCalcPosition3D

Command: getPointDeviation

Synopsis

<double> getPointDeviation <pgroup_id> <point_id> <fobj_id> <frame>

Parameters

<double>	- floating point value
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object
<fobj_id>	- identifier of a frame object
<frame>	- integer value that defines a frame number

Description

On execution this command returns the deviation of the specified point in the specified pointgroup, tracked in the specified frame object.

Example

```
# display the status of the selected point’s
# 2D tracking curve
set pg [getSelectedPGroup]
set point [getSelectedPoint]
set fobj [getSelectedFobj]
for {set frame 1} {$frame<=[getNoFrames $fobj]} \
    {set frame [expr $frame+1]} \
  {
  set dev [getPointDeviation $pg $point $fobj $frame]
  print3DEConsole “Frame: “ $frame “: “ $dev “\n”
  }
```

Command: getPointColor2D

Synopsis

<color_index> getPointColor2D <pgroup_id> <point_id>

Parameters

<color_index>	- integer value: O,...,6
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object

Description

On execution this command returns the tracking points’ color index of the specified point in the specified pointgroup.

Example

```
# display the 2D color index of all points
set pg [getSelectedPGroup]
for {set point [getFirstPoint $pg]} \
     {$point!=0} \
     {set point [getNextPoint $pg $point]} \
  {
  set col [getPointColor2D $pg $point]
  set name [getPointName $pg $point]
  print3DEConsole “Point “ $name “: “ $col “\n”
  }
```

See also

setPointColor2D

Command: setPointColor2D

Synopsis

<color_index> setPointColor2D <pgroup_id> <point_id>

Parameters

<color_index>	- integer value: O,...,6
<pgroup_id>	- identifier of a pointgroup object
<point_id>	- identifier of a point object

Description

On execution this command sets the tracking points’ color index of the specified point in the specified pointgroup to the specified value.

Example

```
# set all selected points to green
set pg [getSelectedPGroup]
for {set point [getFirstPoint $pg]} \
     {$point!=0} \
     {set point [getNextPoint $pg $point]} \
  {
  if {[getPointSelectionFlag $pg $point]} \
      {
      setPointColor2D $pg $point 1
      }
  }
```

See also

getPointColor2D

Command: getScenePosition3D

Synopsis

<vector_3d> getScenePosition3D

Parameters

<vector_3d>	- three-dimensional floating point vector

Description

On execution this command returns the 3D position of the scene node object displayed in the Orientation Window.

Example

```
set vec3d [getScenePosition3D]
```

See also

getSceneRotation3D, getSceneScale3D

Command: getSceneRotation3D

Synopsis

<matrix_3d> getSceneRotation3D

Parameters

<matrix_3d>	- three-dimensional floating point matrix

Description

On execution this command returns the 3D orientation of the scene node object displayed in the Orientation Window.

Example

```
set mtx3d [getSceneOrientation3D]
```

See also

getScenePosition3D, getSceneScale3D

Command: getSceneScale3D

Synopsis

<float> getSceneScale3D

Parameters

<float> - floating point value

Description

On execution this command returns the scale value of the scene node object displayed in the Orientation Window.

Example

```
set scale [getSceneScale3D]
```

See also

getScenePosition3D, getSceneRotation3D

Command: getPGroupPosition3D

Synopsis

<vector_3d> getPGroupPosition3D <pgroup_id> <fobj_id> <frame>

Parameters

<vector_3d>	- three-dimensional floating point vector
<pgroup_id>	- identifier of a pointgroup object
<fobj_id>	- identifier of a frame object
<frame>	- integer value that defines a frame number

Description

On execution this command returns the reconstructed 3D position of the specified pointgroup tracked in the specified frame of the specified frame object. If the pointgroup is of type “camera group” the return value represents the camera’s position in 3D space.

Example

```
# display the 3D position of the selected pointgroup in the
# selected frame object in the current frame
set pg [getSelectedPGroup]
set fobj [getSelectedFobj]
set frame [getSelectedFrame]
set vec3d [getPGroupPosition3D $pg $fobj $frame]
print3DEConsole “3D Position: “ $vec3d “\n”
```

See also

getPGroupRotation3D, getPGroupScale3D

Command: getPGroupRotation3D

Synopsis

<matrix_3d> getPGroupRotation3D <pgroup_id> <fobj_id> <frame>

Parameters

<matrix_3d>	- three-dimensional floating point matrix
<pgroup_id>	- identifier of a pointgroup object
<fobj_id>	- identifier of a frame object
<frame>	- integer value that defines a frame number

Description

On execution this command returns the reconstructed 3D orientation of the specified pointgroup tracked in the specified frame of the specified frame object. If the pointgroup is of type “camera group” the return value represents the camera’s orientation in 3D space.

Example

```
# display the 3D rotation of the selected pointgroup in the
# selected frame object in the current frame
set pg [getSelectedPGroup]
set fobj [getSelectedFobj]
set frame [getSelectedFrame]
set mtx3d [getPGroupRotation3D $pg $fobj $frame]
print3DEConsole “3D Rotation: “ $mtx3d “\n”
```

See also

getPGroupPosition3D, getPGroupScale3D

Command: getPGroupScale3D

Synopsis

<float> getPGroupScale3D <pgroup_id>

Parameters

<float>	- three-dimensional floating point value
<pgroup_id>	- identifier of a pointgroup object

Description

On execution this command returns the overall scale value of the specified pointgroup.

Example

```
# display the 3D scale of the selected pointgroup
set pg [getSelectedPGroup]
set scale [getPGroupScale3D $pg]
print3DEConsole “Scale value: “ $scale “\n”
```

See also

getPGroupPosition3D, getPGroupRotation3D

Command: getFirstCamera

Synopsis

<camera_id> getFirstCamera

Parameters

<camera_id>	- identifier of a camera object

Description

On execution this command returns the camera id of the first camera object. If there are no camera objects available, the string “0” is returned.

Example

```
set camera [getFirstCamera]
```

See also

getNextCamera, getCameraName

Command: getNextCamera

Synopsis

<camera_id> getNextCamera <camera_id>

Parameters

<camera_id>	- identifier of a camera object

Description

On execution this command returns the camera id of the camera object that comes next after the specified camera object. If the specified camera object is the last one, the string “0” is returned.

Example

```
# display the names of all cameras
for {set camera [getFirstCamera]} \
    {$camera != 0} \
    {set camera [getNextCamera $camera]} \
  {
  set name [getCameraName $pg]
  print3DEConsole “Camera name: “ $name “\n”
  }
```

See also

getFirstCamera, getCameraName

Command: getNoCameras

Synopsis

<number_of_cameras> getNoCameras

Parameters

<number_of_cameras>	- integer that specifies the number of camera objects

Description

On execution this command returns the number of available camera objects.

Example

```
set no_cameras [getNoCameras]
```

See also

getIndexCamera, getCameraName

Command: getIndexCamera

Synopsis

<camera_id> getIndexCamera <camera_index>

Parameters

<camera_id>	- identifier of a camera object
<camera_index>	- integer that specifies a camera object

Description

On execution this command returns the camera id of a camera object specified by the camera index parameter. The first camera object is defined by an index of “0”.

Example

```
# display the names of all cameras
for {set index 0} \
    {$index < [getNoCameras]} \
    {set index [expr $index+1]} \
  {
  set camera [getIndexCamera $index]
  set name [getCameraName $camera]
  print3DEConsole “Camera name: “ $name “\n”
  }
```

See also

getNoCameras, getCameraName

Command: getCameraName

Synopsis

<name> getCameraName <camera_id>

Parameters

<name>	- name string
<camera_id>	- identifier of a camera object

Description

On execution this command returns the name of the specified camera.

Example

```
# display name of the currently selected camera
set camera [getSelectedCamera]
set name [getCameraName $camera]
print3DEConsole “Camera name: “ $name “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraName

Synopsis

setCameraName <camera_id> <name>

Parameters

<name>	- name string
<camera_id>	- identifier of a camera object

Description

On execution this command sets the name of the specified camera.

Example

```
# set name of the currently selected camera to „wumpratte“
set camera [getSelectedCamera]
setCameraName $camera „wumpratte“
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraFBackWidth

Synopsis

<float> getCameraFBackWidth <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the filmback width value of the specified camera.

Example

```
# display filmback width of the currently selected camera
set camera [getSelectedCamera]
set fback_w [getCameraFBackWidth $camera]
print3DEConsole $fback_w “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraFBackWidth

Synopsis

setCameraFBackWidth <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the filmback width value of the specified camera.

Example

```
# set filmback width of the currently selected camera
# to 12.0 mm
set camera [getSelectedCamera]
setCameraFBackWidth $camera 1.2
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraFBackHeight

Synopsis

<float> getCameraFBackHeight <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the filmback height value of the specified camera.

Example

```
# display filmback height of the currently selected camera
set camera [getSelectedCamera]
set fback_h [getCameraFBackHeight $camera]
print3DEConsole $fback_h “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraFBackHeight

Synopsis

setCameraFBackHeight <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the filmback height value of the specified camera.

Example

```
# set filmback height of the currently selected camera
# to 9 mm
set camera [getSelectedCamera]
setCameraFBackHeight $camera 0.9
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraFocalLength

Synopsis

<float> getCameraFocalLength <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the focal length value of the specified camera.

Example

```
# display focal length of the currently selected camera
set camera [getSelectedCamera]
set focal [getCameraFocalLength $camera]
print3DEConsole $focal “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraFocalLength

Synopsis

setCameraFocalLength <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the focal length value of the specified camera.

Example

```
# set focal length of the currently selected camera
# to 20 mm
set camera [getSelectedCamera]
setCameraFocalLength $camera 2.0
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraFilmAspect

Synopsis

<float> getCameraFilmAspect <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the film aspect value of the specified camera.

Example

```
# display film aspect of the currently selected camera
set camera [getSelectedCamera]
set film_aspect [getCameraFilmAspect $camera]
print3DEConsole $film_aspect “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraFilmAspect

Synopsis

setCameraFilmAspect <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the film aspect value of the specified camera.

Example

```
# set film aspect of the currently selected camera
# to 1.333
set camera [getSelectedCamera]
setCameraFilmAspect $camera 1.333
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraLensCenterX

Synopsis

<float> getCameraLensCenterX <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the lens center x offset value of the specified camera.

Example

```
# display lens center x of the currently selected camera
set camera [getSelectedCamera]
set lcenter_x [getCameraLensCenterX $camera]
print3DEConsole $lcenter_x “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraLensCenterX

Synopsis

setCameraLensCenterX <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the lens center x offset value of the specified camera.

Example

```
# set lens center x of the currently selected camera
# to 3 mm
set camera [getSelectedCamera]
setCameraLensCenterX $camera 0.3
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraLensCenterY

Synopsis

<float> getCameraLensCenterY <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the lens center y offset value of the specified camera.

Example

```
# display lens center y of the currently selected camera
set camera [getSelectedCamera]
set lcenter_y [getCameraLensCenterY $camera]
print3DEConsole $lcenter_y “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraLensCenterY

Synopsis

setCameraLensCenterY <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the lens center y offset value of the specified camera.

Example

```
# set lens center y of the currently selected camera
# to 1.5 mm
set camera [getSelectedCamera]
setCameraLensCenterY $camera 0.15
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraHorizAngle

Synopsis

<float> getCameraHorizAngle <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the horizontal angle value of the specified camera.

Example

```
# display horizontal angle of the currently selected camera
set camera [getSelectedCamera]
set h_angle [getCameraHorizAngle $camera]
print3DEConsole $h_angle “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraHorizAngle

Synopsis

setCameraHorizAngle <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the horizontal angle value of the specified camera.

Example

```
# set horizontal angle of the currently selected camera
# to 45.0 degree
set camera [getSelectedCamera]
setCameraHorizAngle $camera 0.785398
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraVertiAngle

Synopsis

<float> getCameraVertiAngle <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the vertical angle value of the specified camera.

Example

```
# display vertical angle of the currently selected camera
set camera [getSelectedCamera]
set v_angle [getCameraVertiAngle $camera]
print3DEConsole $v_angle “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraVertiAngle

Synopsis

setCameraVertiAngle <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the vertical angle value of the specified camera.

Example

```
# set vertical angle of the currently selected camera
# to 30.0 degree
set camera [getSelectedCamera]
setCameraVertiAngle $camera 0.5235988
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraPixelAspect

Synopsis

<float> getCameraPixelAspect <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the pixel aspect value of the specified camera.

Example

```
# display pixel aspect of the currently selected camera
set camera [getSelectedCamera]
set pixel_aspect [getCameraPixelAspect $camera]
print3DEConsole $pixel_aspect “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraPixelAspect

Synopsis

setCameraPixelAspect <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the pixel aspect value of the specified camera.

Example

```
# set pixel aspect of the currently selected camera
# to 1.06666
set camera [getSelectedCamera]
setCameraPixelAspect $camera 1.06666
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraLensDistortion

Synopsis

<float> getCameraLensDistortion <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the lens distortion value of the specified camera.

Example

```
# display lens distortion of the currently selected camera
set camera [getSelectedCamera]
set distortion [getCameraLensDistortion $camera]
print3DEConsole $distortion “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraLensDistortion

Synopsis

setCameraLensDistortion <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the lens distortion value of the specified camera.

Example

```
# set lens distortion of the currently selected camera
# to 0.04
set camera [getSelectedCamera]
setCameraLensDistortion $camera 0.04
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraAnamorphicSqueeze

Synopsis

<float> getCameraAnamorphicSqueeze <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the anamorphic squeeze value of the specified camera.

Example

```
# display anamorphic squeeze of the
# currently selected camera
set camera [getSelectedCamera]
set anamorph [getCameraAnamorphicSqueeze $camera]
print3DEConsole $anamorph “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraAnamorphicSqueeze

Synopsis

setCameraAnamorphicSqueeze <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the anamorphic squeeze value of the specified camera.

Example

```
# set anamorphic squeeze of the currently selected camera
# to 0.01
set camera [getSelectedCamera]
setCameraAnamorphicSqueeze $camera 0.01
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraCurvatureX

Synopsis

<float> getCameraCurvatureX <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the curvature x value of the specified camera.

Example

```
# display curvature x of the currently selected camera
set camera [getSelectedCamera]
set curve_x [getCameraCurvatureX $camera]
print3DEConsole $curve_x “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraCurvatureX

Synopsis

setCameraCurvatureX <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the curvature x value of the specified camera.

Example

```
# set curvature x of the currently selected camera
# to 0.02
set camera [getSelectedCamera]
setCameraCurvatureX $camera 0.02
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraCurvatureY

Synopsis

<float> getCameraCurvatureY <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the curvature y value of the specified camera.

Example

```
# display curvature y of the currently selected camera
set camera [getSelectedCamera]
set curve_y [getCameraCurvatureY $camera]
print3DEConsole $curve_y “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraCurvatureY

Synopsis

setCameraCurvatureY <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the curvature y value of the specified camera.

Example

```
# set curvature y of the currently selected camera
# to -0.03
set camera [getSelectedCamera]
setCameraCurvatureY $camera -0.03
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: getCameraQuarticDistortion

Synopsis

<float> getCameraQuarticDistortion <camera_id>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command returns the quartic distortion value of the specified camera.

Example

```
# display quartic distortion of the current camera
set camera [getSelectedCamera]
set q_distort [getCameraQuarticDistortion $camera]
print3DEConsole $q_distort “\n”
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Command: setCameraQuarticDistortion

Synopsis

setCameraQuarticDistortion <camera_id> <float>

Parameters

<float>	- floating point value
<camera_id>	- identifier of a camera object

Description

On execution this command sets the quartic distortion value of the specified camera.

Example

```
# set quartic distortion of the current camera
# to 0.0
set camera [getSelectedCamera]
setCameraQuarticDistortion $camera 0.0
```

See also

getFirstCamera, getNextCamera, getNoCameras, getIndexCamera

Appendix B: A collection of vector, matrix and quaternion operations in Tcl

The file “veclib.tcl” in the 3DE distribution (directory “<3de install dir>/sys_data/tcl_archive”) is a collection of Tcl procedures, which are based on the vector, matrix and quaternion types used for 3DE’s Tcl interface. You will find a more detailed description and examples in the file “veclib_doc.pdf”. In the list below we used the following conventions:

- The entries for each procedure in the list consist of the return type of the procedure (in <angle> brackets), the procedure name and the list of parameters.

- The possible return types are:
    - <real> for real numbers
    - <vec2> for two-dimensional vectors
    - <vec3> for two-dimensional vectors
    - <quat> for quaternions
    - <mat2> for 2x2-matrices
    - <mat3> for 3x3-matrices
    - <mat4> for 4x4-matrices
    - <srt2> for two-dimensional general transformations
    - <srt3> for three-dimensional general transformations
    - <boole> boolean return type: “0” or “1”

    - <vec> stands for all types of vectors (i.e. vec2, vec3 and the unsupported type vec4)
    - <mat> stands for all types of matrices (i.e. mat2, mat3 and the unsupported type mat4)
    - <srt> stands for all types of general transformations (i.e. srt2 and srt3)

- The command names are constructed as follows
    - Binary operators match the pattern <type1><op><type2>, e.g. vec2+vec2, mat3*vec3
    - Unary operators match the pattern <op><type>, e.g. -quat
    - Functions mapping from an object of type <type> and possible parameters to anywhere match the pattern <type><func-name>, e.g. mat2invert, mat3subcyc
    - Functions testing the type match the pattern is<type>, e.g. ismat3
    - Type conversions / constructors match the pattern <type1>=<type2>, e.g. quat=vec3,
    - The nomenclatura of get/set functions is not necessarily regular, e.g. mgetrow, qgetvec...

- The list of parameters obey the following convention:
    - vectors are denoted by v, v1, v2, v3...
    - matrices are denoted by m, m1, m2, m3...
    - integers are denoted by i, j, k...
    - angles (radiant!) are denoted by w, w*
    - numbers (scalars) are denoted by s, s1, s2, s3...
    - general transformations (srt) are denoted by l, l1, l2, l3...
    - parameters are written as if they were given to the procedure by variables (with or without '$')

In Tcl there are no elementary data types except for strings. We use strings and Tcl lists in order to construct vectors, matrices, quaternions and transformations. Examples:

```
wish>set va {1 2 3}
wish>set vb {4 5 6}
wish>puts [vec3+vec3 $va $vb]
5 7 9

wish>set ma {{1 2} {3 4}}
wish>set mb {{5 6} {7 8}}
wish>puts [mat2*mat2 $ma $mb]
{19 22} {43 50}

wish>set q {1 {2 3 4}}
wish>puts $q
1 {2 3 4}

wish>set m {{11 12 13} {21 22 23} {31 32 33}}
wish>puts $m
{11 12 13} {21 22 23} {31 32 33}

wish>set l [list $v $m]; # or: set l [srt=vecmat $v $m]
wish>puts $l
{ 5 7 9 } {{11 12 13} {21 22 23} {31 32 33}}
```

Commands

get/set functions

<real>	**vget** $v $i	Component i of vector v.
<vec>	**vset** v $i $s	Set component i of vector v to s. Returns v.
<vec>	**mgetrow** $m $i	Row-vector i of matrix m.
<mat>	**msetrow** m $i $v	Set row-vector i of matrix m to v. Returns m.
<real>	**mget** $m $i $j	Component (i,j) of matrix m.
<mat>	**mset** m $i $j $s	Set component (i,j) of matrix m to s.
<real>	**qget** $q $i	Component i of quaternion q.
<quat>	**qset** q $i $s	Set component i of quaternion q to s.
<real>	**qgets** $q	Scalar part of quaternion q (component 0).
<quat>	**qsets** q $s	Set scalar part of quaternion q (component 0) to s.
<vec3>	**qgetvec** $q	Vector part of quaternion q.
<quat>	**qsetvec** q $v	Set vector part of quaternion q to v.
<vec>	**srtgetvec** $l	Vector part of transformation l.
<srt>	**srtsetvec** l $v	Set vector part of transformation l to v. Returns l.
<mat>	**srtgetmat** $l	Matrix part of transformation l.
<srt>	**srtsetmat** l $m	Set matrix part of transformation l to m. Returns l.

vec2

<vec2>	**vec2+vec2** $v1 $v2	Sum of vectors v1 and v2.
<vec2>	**vec2-vec2** $v1 $v2	Difference of vectors v1 and v2.
<vec2>	**-vec2** $v	Negative of vector v.
<real>	**vec2*vec2** $v1 $v2	Scalar product of vectors v1 and v2.
<vec2>	**vec2*s** $v $s	Component-wise product of vector v and scalar s.
<vec2>	**s*vec2** $s $v	Component-wise product of vector v and scalar s.
<vec2>	**vec2/s** $v $s	Component-wise quotient of vector v and scalar s.
<real>	**vec2^vec2** $v1 $v2	Skew-symmetric spinor product of vectors v1 and v2.
<mat2>	**vec2&vec2** $v1 $v2	Tensor product of vectors v1 and v2.
<real>	**vec2normquad** $v	Square of the euclidian norm of vector v.
<real>	**vec2norm** $v	Euclidian norm of vector v.
<vec2>	**vec2unit** $v	Vector v normalized.
<boole>	**isvec2** $a	a is a list that consists of two components? 1:0.

vec3

<vec3>	**vec3+vec3** $v1 $v2	Sum of vectors v1 and v2.
<vec3>	**vec3-vec3** $v1 $v2	Difference of vectors v1 and v2.
<vec3>	**-vec3** $v	Negative of vector v.
<real>	**vec3*vec3** $v1 $v2	Scalar product of vectors v1 and v2.
<vec3>	**vec3*s** $v $s	Component-wise product of vector v and scalar s.
<vec3>	**s*vec3** $s $v	Component-wise product of vector v and scalar s.
<vec3>	**vec3/s** $v $s	Component-wise quotient of vector v and scalar s.
<vec3>	**vec3^vec3** $v1 $v2	Skew-symmetric vector product of vectors v1 and v2.
<mat3>	**vec3&vec3** $v1 $v2	Tensor product of vectors v1 and v2.
<real>	**vec3normquad** $v	Square of the euclidian norm of vector v.
<real>	**vec3norm** $v	Euclidian norm of vector v.
<vec3>	**vec3unit** $v	Vector v normalized.
<mat3>	**vec3dual** $v	Dual matrix of vector v.
<boole>	**isvec3** $a	a is a list that consists of three components? 1:0.

quat

<quat>	**quat+quat** $q1 $q2	Sum of quaternions q1 and q2.
<quat>	**quat-quat** $q1 $q2	Difference of quaternions q1 and q2.
<quat>	**-quat** $q	Negative of quaternions.
<quat>	**quat^quat** $q1 $q2	Non-commutative product of quaternions q1 and q2.
<real>	quat*quat $q1 $q2	Real-valued scalar product of quaternions q1 and q2.
<quat>	**quat*s** $q $s	Product of quaternion q and scalar s.
<quat>	**s*quat** $s $q	Product of quaternion q and scalar s.
<quat>	**quat/s** $q $s	Quotient of quaternion q and scalar s.
<quat>	**quat=s** $s	Generates a quaternion out of scalar s.
<quat>	**quat=vec3** $v	Generates a quaternion out of vector v.
<real>	**quatnormquad** $q	Square of the euclidian norm of quaternion q.
<real>	**quatnorm** $q	Euclidian norm of quaternion q.
<quat>	**quatunit** $q	Quaternion q normalized.
<quat>	**quatconj** $q	Conjugate of quaternion q.
<quat>	**quatinvert** $q	Inverse quaternion of q.
<mat3>	**quatmaptomat3** $q	Maps quaternion q into a 3x3-matrix. See explanation in file “veclib_doc.pdf”.

mat2

<mat2>	**mat2+mat2** $m1 $m2	Sum of matrices m1 and m2.
<mat2>	**mat2-mat2** $m1 $m2	Difference of matrices m1 and m2.
<mat2>	**-mat2** $m	Negative of matrix m.
<mat2>	**mat2*mat2** $m1 $m2	Product of matrices m1 and m2.
<mat2>	**mat2*s** $m $s	Component-wise product of matrix m and scalar s.
<mat2>	**s*mat2** $s $m	Component-wise product of matrix m and scalar s.
<vec2>	**mat2*vec2** $m $v	Product of matrix m and vector v.
<mat2>	**mat2/s** $m $s	Component-wise quotient of matrix m and scalar s.
<mat2>	**mat2=s** $s	Generates a matrix out of scalar s.
<mat2>	**mat2=diag** $s1 $s2	Generates a diagonal matrix out of scalars s1 and s2.
<real>	**mat2trace** $m	Trace of matrix m.
<real>	**mat2det** $m	Determinant of matrix m.
<mat2>	**mat2trans** $m	Matrix m transposed.
<real>	**mat2subcyc** $m $i $j	Adjoint element of matrix m with respect to component (i,j).
<mat2>	**mat2invert** $m	Inverse of matrix m.
<real>	**mat2rotangle** $m	Calculate rotation angle of a rotation matrix m. See explanation in file “veclib_doc.pdf”.
<mat2>	**mat2rotmatrix** $w	Calculate rotation matrix out of angle w. See explanation in file “veclib_doc.pdf”.
<boole>	**ismat2** $a	a is a list of two vec2 objects? 1:0.

mat3

<mat3>	**mat3+mat3** $m1 $m2	Sum of matrices m1 and m2.
<mat3>	**mat3-mat3** $m1 $m2	Difference of matrices m1 and m2.
<mat3>	**-mat3** $m	Negative of matrix m.
<mat3>	**mat3*mat3** $m1 $m2	Product of matrices m1 and m2.
<mat3>	**mat3*s** $m $s	Component-wise product of matrix m and scalar s.
<mat3>	**s*mat3** $s $m	Component-wise product of matrix m and scalar s.
<vec3>	**mat3*vec3** $m $v	Product of matrix m and vector v.
<mat3>	**mat3/s** $m $s	Component-wise quotient of matrix m and scalar s.
<mat3>	**mat3=s** $s	Generates a matrix out of scalar s.
<mat3>	**mat3=diag** $s1 $s2 $s3	Generates a diagonal matrix out of scalars s1, s2 and s3.
<real>	**mat3trace** $m	Trace of matrix m.
<real>	**mat3det** $m	Determinant of matrix m.
<mat3>	**mat3trans** $m	Matrix m transposed.
<mat2>	**mat3subcyc** $m $i $j	Adjoint cyclic submatrix of matrix m with respect to

		component (i,j).
<mat3>	**mat3invert** $m	Inverse of matrix m.
<vec3>	**mat3dual** $m	Dual vector of matrix m.
<quat>	**mat3maptoquat** $m	Maps matrix m into a quaternion. See explanation in file “veclib_doc.pdf”.
<vec3>	**mat3rotangles** $m $i $j $k	Calculate rotation angles of a rotation matrix m. See explanation in file “veclib_doc.pdf”.
<mat3>	**mat3rotmatrix** $i $j $k $wi $wj $wk	Calculate rotation matrix out of angles wi, wj, wk. See explanation in file “veclib_doc.pdf”.
<boole>	**ismat3** $a	a is a list of three vec3 objects? 1:0.

mat4

<mat4>	**mat4*mat4** $m1 $m2	Product of matrices m1 and m2.

srt2

<vec2>	**srt2*vec2** $l $v	Applies transformation l to vector v.
<mat2>	**srt2*mat2** $l $m	Applies transformation l to matrix m.
<srt2>	**srt2*srt2** $l1 $l2	Product of transformations l1 and l2.
<srt2>	**srt2invert** $l	Inverse of transformation l.
<mat3>	**mat3=srt2** $l	Generate a 3x3-matrix representation out of transformation l.

srt3

<vec3>	**srt3*vec3** $l $v	Applies transformation l to vector v.
<mat3>	**srt3*mat3** $l $m	Applies transformation l to matrix m.
<srt3>	**srt3*srt3** $l1 $l2	Product of transformations l1 and l2.
<srt3>	**srt3invert** $l	Inverse of transformation l.
<mat4>	**mat4=srt3** $l	Generate a 4x4-matrix representation out of transformation l.