StubeRenA Calibration Software
User Manual

G. Eibner

June 2003

Contents

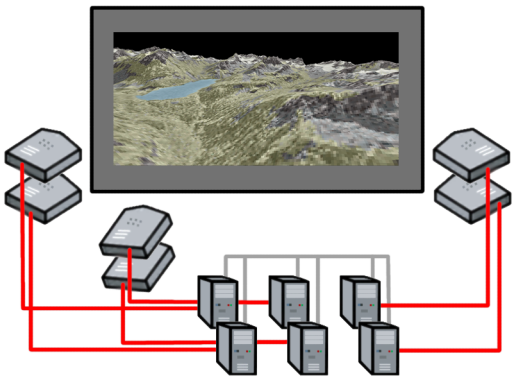

Figure 1: A tiled display composed of various displays connected to a computer cluster.

1 Introduction

The Overlap software package is a calibration software designed to run on a cluster of workstations to meet the need for a large tiled display. Its intended use is to calibrate a tiled display composed of various displays connected to a computer cluster. A computer cluster is a set of networked computers which share data of the different overlap software parts. Each computer within the cluster can drive one or more displays. All displays together form a huge display termed a tiled display. And each display will be referred to as a tile of the tiled display. The output generated by the calibration software is used to run a tiled display (examples for outputs used by other applications are blending masks, projection and collineation matrices). See figure 1.

The display assembly can be homogeneous (using the same or a similar display device for each tile of the tiled display), or heterogenous (using different devices to form a multiscreen display). In the homogenous case using projectors one can build a seamless tiled display with as many devices as needed to form a megapixel screen wall. Alternatively one can use regular monitors for displaying a large scenery with high detail and high resolution. The drawback are the seams between adjoining displays. In the heterogenous case any kind of displays may be used building a mixture of seamless displays and information sideboards. Figure 2 depicts these three configurations.

In our implementation we used a 3×2 projector array to build a display wall with a total resolution of 4.7 Megapixels.

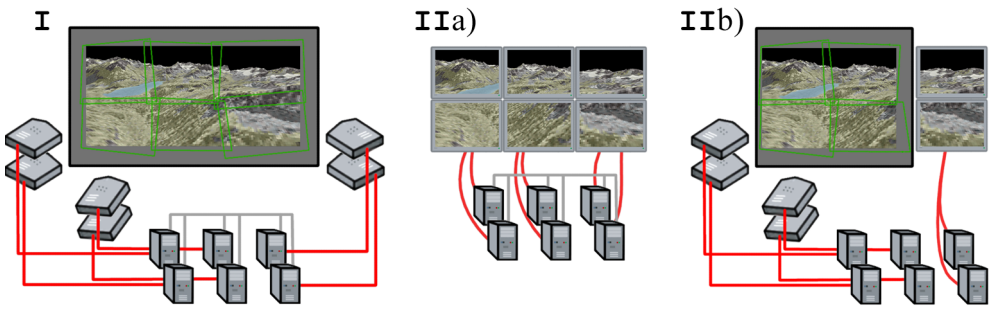

Figure 2: The three different display configurations. I) homogeneous display assembly: a seamless tiled display using projectors. II) heterogeneous display assembly: a) a tiled display using multiple monitors, b) a tiled display using a seamless display part and an information part with seams.

2 Application usage

The Overlap software package contains four applications to calibrate each display in the tiled display, namely a server application, a camera client, projector client, and user client application. The server application manages the communication between the three other applications. The server must be the first application to run. All other applications can be started arbitrarily. The camera client application takes care of calibration. A projector client application will be launched for each display within the render cluster that contributes to the tiled display. The user client application is the interface for an user to start the calibration and change various calibration parameters. Each application will be started with a configuration file holding a brief description, see the following paragraphs for detailed information.

2.1 Server application

The server application will be launched just with a single parameter telling it on which port to listen to. The server is only responsible for delivering the concurrent clients with the right information. Thus every client application is able to communicate with each other.
Usage: `server [port]`, if no port is supported the a default port number is used.

`server:` *string*`:`*integer*	`server: 169.128.128.1:23451`
`size:` *integer*`,`*integer*	`size: 640,480`
`name:` *string*	`name: 1394`
`param file:` *string*	`param file: camparams.txt`

Table 1: Camera configuration file specification (left), example (right).

2.2 Camera client application

The camera application should be started with a configuration file telling the application which camera to use and at which resolution. The application will run a preview window showing the current area seen by the camera. The camera should be installed in a way keeping its image plane roughly parallel to the screen surface with the camera's field of view covering the whole area of the projection screen. In the current implementation of the Overlap software only one camera is used to calibrate the tiled display. So, all displays must be seen by the selected camera. Once running the calibration step the camera will take a snapshot of each display finding its relative position and orientation to all other displays and in respect to the display screen.

2.2.1 Configuration file

The configuration file contains information about the server and the camera. There are four values to set in the configuration file, see also table 1.

`server`: Specifies the host name or the IP-address where the calibration server is running. The port can be additionally set. Write the port after the name with a colon separated

`size`: Descibes the resolution of the used camera in pixels. Horizontal resolution first, vertical second.

`name`: The camera's name, or a part of the name (like 1394, Quickcam, Sony, ...).

`param file`: Sepcifies a parameter file that contains the distortion parameter of the camera. See next section for a brief description on parameter files.

float float float float	`-0.417830 0.343488 0.002087 0.001606`
float float float	`719.8320313 0.0000000 346.2613831`
float float float	`0.0000000 717.3533325 226.6094513`
float float float	`0.0000000 0.0000000 1.0000000`

Table 2: Camera parameter file specification (left), example (right).

2.2.2 Parameter file

The parameter file contains the distortion coefficients used by OpenCV to undistort images taken by a calibrated camera. Within the file the parameters are written as floats and separated by spaces. The four floats in the first line representing the four tangential and radial distortion parameters. Where the next three lines containing three floats each line represent the camera instrinsic matrix. Table 2 shows the file specification and an example.

2.2.3 Usage

`camera [config-file]`, if no configuration file is supported default values are used for the camera without a guarantee to run the application.

`camera -calib images fieldX fieldY fieldSize time`, use camera client as stand-alone application to find distortion parameters. *Note:* a calibration pattern (chessboard) is needed to compute the parameters. Move and rotate the calibration pattern in front of the camera to find good results for the distortion parameters. After all images are taken the undistorted camera view is shown. The computed result can be accepted and the application quits. If the result is not accepted by the user the camera calibration proccess starts again. The result can be copied to a parameter file. Set `images` to the number of images that should be taken to find the distortion parameters. Set `fieldX` and `fieldY` to the number of fields on the chessboard in horizontal and vertical direction. `fieldSize` is the real size of a chessboard field in centimetres. `time` is the elapsed time in milliseconds between consecutive snapshots.

`camera -preview parameter-file`, use camera client as stand-alone application to manually tune distortion parameters. The output can be copied to a parameter file. Set `parameter-file` to a distorion parameter file. Its content will be read and used for fine adjustment.

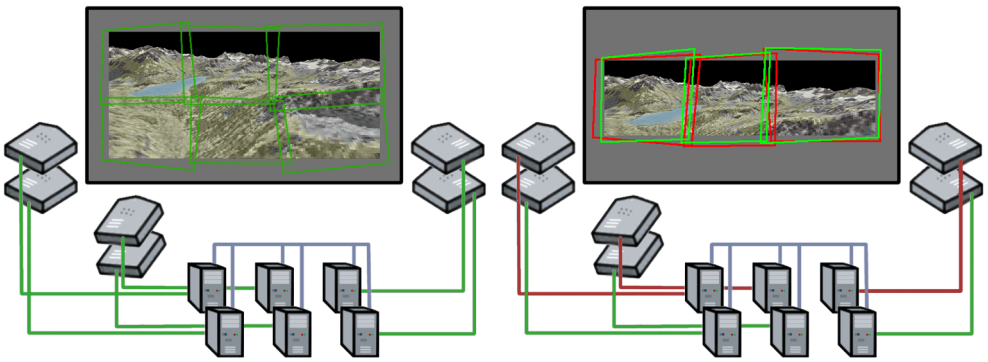

Figure 3: A mono tiled display on the right and a stereo tiled display on the left. Note the overlap of two projectors at a time from the left eye group (red) and the right eye group (green).

2.3 Projector client application

The projector application will be started for each display in the render cluster respectively in the tiled display. With the information taken from the configuration file one can add as many displays as needed to form the tiled display. The configuration file contains a description of the display like resolution, gamma correction value, and its brightness. But it delivers also the opportunity to build a mono tiled display, or a stereo tiled display. In the latter case all displays have to be either a stereo left, or stereo right display. If a mixed usage of mono and stereo displays are used, the Overlap software will change to mono mode. All displays with a stereo left attribute forming the tiled display for the left eye, and all displays with a stereo right attribute forming the tiled display for the right eye. Each tiled display will be calibrated like a mono tiled display would do, but to achieve a stereo tiled display, both should overlap. Figure 3 depicts the situation for a mono display and a stereo display.

2.3.1 Configuration file

The configuration file contains information about the server and the projector or display used. There are six values to set in the configuration file, see also table 3.

`server`: Specifies the host name or the IP-address where the calibration server is running. The port can be additionally set. Write the port after the name with a colon separated

`server:` *string*`:`*integer*	`server: 169.128.128.1:23451`
`origin:` *integer*`,`*integer*	`origin: 0,0`
`size:` *integer*`,`*integer*	`size: 1024,768`
`mode:` *string*	`mode: mono`
`gamma:` *float*	`gamma: 1.9`
`brightness:` *float*	`brightness: 1.0`

Table 3: Projector configuration file specification (left), example (right).

`origin` and `size`: Descibes the origin and the size of the used display in pixels. The origin is measured with respect to the computer's desktop or framebuffer resolution (e.g. using a 2048×768 pixels desktop resolution with two displays at a resolution of 1024×768 pixels leads to 0,0 for the first and 1024,0 for the second display).

`mode`: This value is one of `mono`, `stereo_left`, or `stereo_right` to describe the intended use of the display. If `mono` is used a mono display wall will be built. If `stereo_left` or `stereo_right` is used for the used displays, a passive stereo display will be built. If a mixture of `mono` and `stereo_*` is used, a mono display wall is constructed.

`gamma` and `brightness`: Sepcifies an exponential gamma correction value within the alpha mask generation parameter, and a linear brightness factor with respect to all other used displays (range between 0 and 1)

2.3.2 Usage

`projector [config-file]`, if no configuration file is supported default values are used for the display without a guarantee to run the application or representing the display correctly.

2.4 User client application

The user application represents the interface for the user to start and change the calibration steps. It runs in a console window and the user controls the application with the keyboard. See table 4 for a description of the keys' intended use.

Calibration control keys	
CTRL-X	Stops all overlap applications connected with the server (including all clients and the server)
X	Stops the user application
G,.	Terminates a client group (P-projector, C-camera, A-both projector and camera clients)
H	Shows help (command keys)
S	Starts calibration
ESCAPE	Stops calibration
R	Runs Studierstube on display wall
T	Runs Studierstube on display wall and terminates all calibration programs
Projector control keys	
N	Sets new resolution of chessboard
P	Shows/hides pattern on each projector client's display area
B	Shows/hides blank screen on each projector client's display area
RIGHT/LEFT	Increase/decrease pattern brightness of each projector client
Camera control keys	
V	Switches between camera preview modes
C	Shows/hides camera client's preview window
SPACE	Resets calibration values (alpha masks and display wall area)
UP/DOWN	Scrolls through detected pictures or computed alpha masks

Table 4: Keys and their meanings.

`server:` *string*`:`*integer*
`render sync server address:` *string*
`render sync multicast address:` *string*
`shared directory:` *string*
`server: 169.128.128.1:23451`
`render sync server address: 169.128.128.1`
`render sync multicast address: 224.0.0.1`
`shared directory: //Blue/shared/overlap`

Table 5: User configuration file specification (top), example (bottom).

2.4.1 Configuration file

The configuration file contains information about the Overlap server and contains information neccessary to build Open Inventor files for the Studierstube environment. There are four values to set in the configuration file, see also table 5.

`server`: Specifies the host name or the IP-address where the calibration server is running. The port can be additionally set. Write the port after the name with a colon separated

`render sync server address`: This specifies the host name or IP address where the synchronization server runs. This server is used in the StubeRenA environment to synchronize framebuffer switches between the all render synchronization clients. Clients are instantiate by the subclass of SoRenderSyncUDP and join a multicast group to listen to the server's synchronization packet.

`render sync multicast address`: The multicast IP address where the render synchronization clients can join. This multicast group is used by the render synchronization server. A packet is sent to all members of this group to force a framebuffer switch. This guarantuees synchronized output on all tiles of the tiled display.

`shared directory`: A shared directory destination, where generated files are stored. Files that are generated during the calibration step are store to this directory. This directory contains all files needed for the StubeRenA to run (e.g. alpha masks, correcting projection matrices, startup informations).

2.4.2 Usage

`user [config-file]`, if no configuration file is supported default values are used for the user application without a guarantee to run the Studierstube software correctly afterwards.

3 Installation and Compilation

Overlap consists of four program parts. The files overlap.exe, camera.exe, projector.exe, and user.exe. Additional components are the DLLs from Intel's Image Processing Library (IPL) and the DLL from the DsDxVideoWrapper which must also be installed (extend PATH variable with './overlap/bin/DLLs') or in the same directory as the camera.exe file.

The original Overlap directory structure looks as follows (unzipping the overlap software package)

Executables and DLLs
- `./overlap/bin`
- `./overlap/bin/DLLs`

Source code
- `./overlap/src`
- `./overlap/src/all`
- `./overlap/src/lib`
- `./overlap/src/renaCameraClient`
- `./overlap/src/renaClient`
- `./overlap/src/renaProjectorClient`
- `./overlap/src/renaProtocols`
- `./overlap/src/renaServer`
- `./overlap/src/renaUserClient`

3.1 Before compiling the Overlap package

- Install DirectX8.1 SDK (for the DsDxVideoWrapper)
- Install OpenCV (for camera calibration routines)
- Set an environment variable OPENCV to full path of `./Opencv/cv` directory
- Compile and build the `CVl.lib` (for release code) and/or `CVld.lib` (for debug code)

  - Open `./OpenCV/_dsw/cv.dsw`
  - Switch to 'Win32 Release Static' or 'Win32 Debug Static' within the Build bar.
  - Start compile

- Compile Overlap package
  - Switch to project 'all files' in the workspace window to compile the whole package
  - or switch to any project of 'rena*Client', 'renaServer' to compile individual parts of the package

3.2 Before running any part of the Overlap software

- Add to PATH variable the full path to `./Overlap/bin/DLLs`
- Install DirectX8.1 SDK (for the DsDxVideoWrapper)

4 Running the applications

- See section 3.2 before proceeding.
- Run the server application first – `server.exe` file (usage: `server [port]` see section 2.1).
- Run any of the client applications on any node within the network.
- You have to run at least one camera and two projector clients to be able to start the calibration.
- The server does not allow more than one user client. The first user client that register at the server will be served.
- In the server and user console window you will see which clients are connected to the server.
- Hit the key 'H' in the user console to get the list of control commands to use the calibration software (see also table 4.
- In the camera *preview window* you will see the actual process of the calibration and the camera's view.
- The camera *console window* shows information about the calibration progress.

5 Known problems

- Run the server and user application on the same machine. Since very small buffers are used for communication and no care about message fragmentation is done, this should avoid any problem. If the user client fails start it again (this is because of fragmented messages, maybe we will fix this problem in future versions of Overlap).