Brown University YURT Operations Manual

Tom Sgouros

June 25, 2018

Contents

1 Access and safety

1.1 Access to catwalk

Access to the catwalk is via the ladder leaning on the wall.

When doing work on the catwalk, there is a wooden grating that is to be placed between the catwalk supports to prevent falls and dropping items onto the mirrors below. There are two gratings stored under the floor, just inside the eastern access door. Projectors shine through those gaps so the gratings cannot be left in place.

1.2 Access to upper mirrors and rear top of wall

Inside the eastern access door to the under-floor area, there is a platform with two long legs and two short legs. It is meant to sit behind the front wall of the YURT, with two short legs resting on the back of the floor structure and the two long legs reaching down to the actual room floor. From this platform, you can reach the bottom of the catwalk, or the top of the upper mirrors.

1.3 Removing upper projectors

To remove an upper wall projector or a ceiling projector, please follow these steps:

1. There is a small dolly that will fit on the catwalk, stored in the work area behind the elevator.
2. Place safety grating over the nearest gap to the projector. This is stored inside the eastern access door under the floor.
3. Loosen set screws on vertical shaft. The projector mount, including the flat plate, and the shaft, must be removed with the projector.
4. Slide projector and mount down and out of collar, place onto dolly.

2 Care of Nodes

The computers that run the YURT are 20 server-class machines, named cave001 through cave020. They each have 128G memory, and two network interfaces, so are addressable as 172.20.160.X and 192.168.160.X, where X is the number of the node, between 1 and 20. The projector numbers correspond to the following cave nodes:

Node	Projectors
cave001	34,35,36,37
cave002	30,31,32,33
cave003	26,27,28,29
cave004	22,23,24,25
cave005	18,19,20,21
cave006	14,15,16,17
cave007	10,11,12,13
cave008	7,8,9
cave009	3,4,5,6
cave010	0,1,2
cave011	38,39,40,41
cave012	44,45
cave013	46,47,48,49
cave014	50,51,52
cave015	53,54,57,58
cave016	55,56,59,60
cave017	61,62,63,66
cave018	64,65,67,68
cave019	42,43
cave020	test

Cave node 20 is a spare computer, and is also used to power projectors in the testing area. The test projector port can be addressed as projector 70.

Each cave node carries four NVidia GPUs, one for each projector. The GPUs are each mapped to a different X windows display, so cave001 has cave001:0.0, cave001:0.1, cave001:0.2, and cave001:0.3.

2.1 X11

The X server is invoked in a locally written script called `cavedm.conf`. This is where different command-line options can be specified.

Do not modify the installed version of this file. During the installation of the cave image, the init file found at `/gpfs/runtime/nvidia/cave-X11/cavedm.conf` is copied to `/etc/xinit`, where it is executed at boot time.

To restart the X server on a node, log into that node as cavedemo, and run `killall unclutter`.

Find the relevant startx script here: `/gpfs/runtime/nvidia/cave-X11/cave-startx`.

2.2 NVidia driver

To see details about the NVidia GPUs on a cave node, use the `nvidia-smi` command. (Only works for the cavedemo user.)

You'll find all the installed drivers in `/gpfs/runtime/nvidia`. There is a link to "latest" that will indicate which driver has actually been installed on the cave nodes.

Upgrades to the NVidia drivers should be done with care. Upgrading from v313 to v361 caused boot failures on two nodes.

3 Projectors

There are two documents we have from the projector manufacturer.

Brown 3D Spec Rev1 2011-11-21.pdf The specification for the projectors we have.

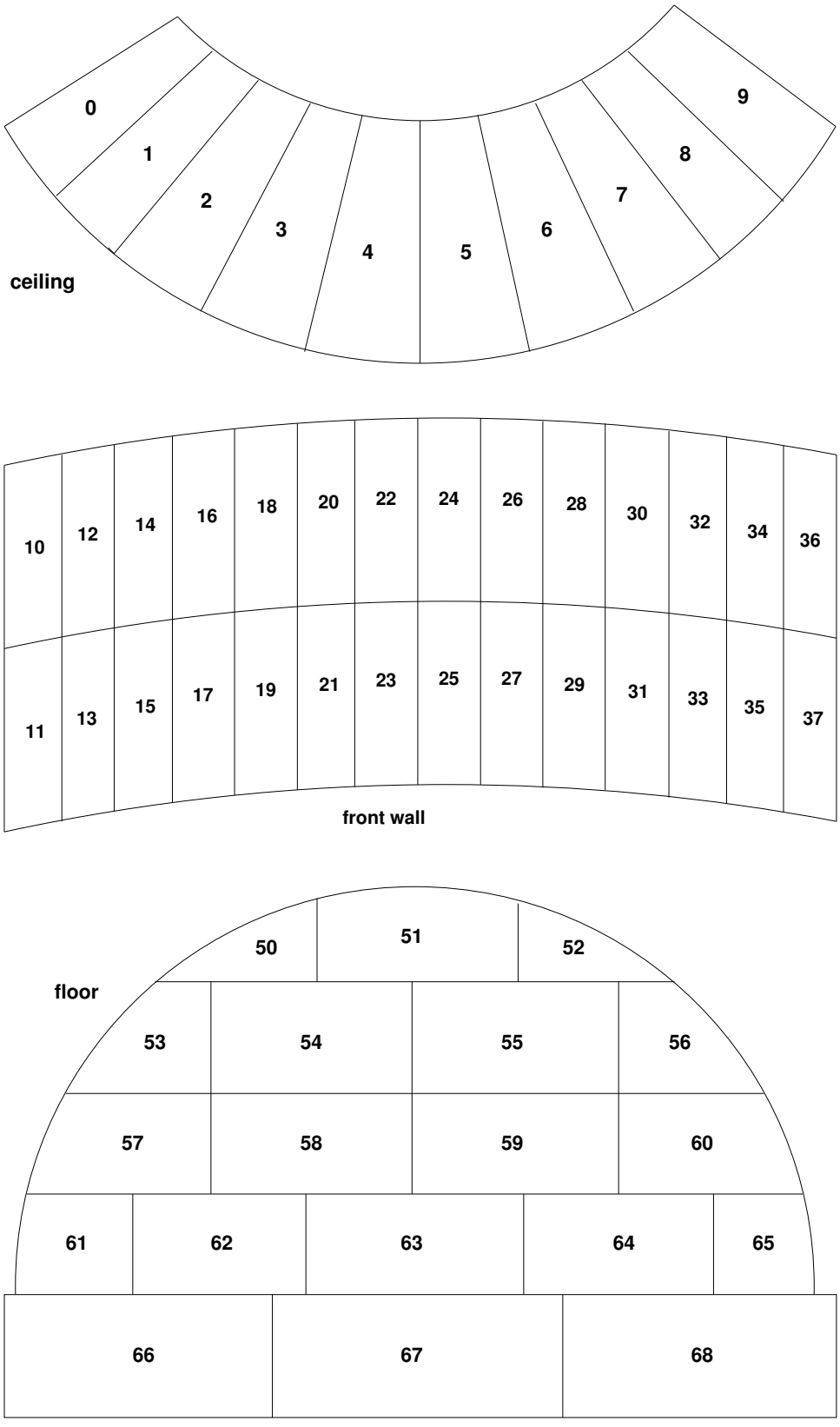
DP HTD-8650_Service Manual Guide.2015-4-30 Rev 01.pdf A repair manual for the projector breed that ours are hacked from.

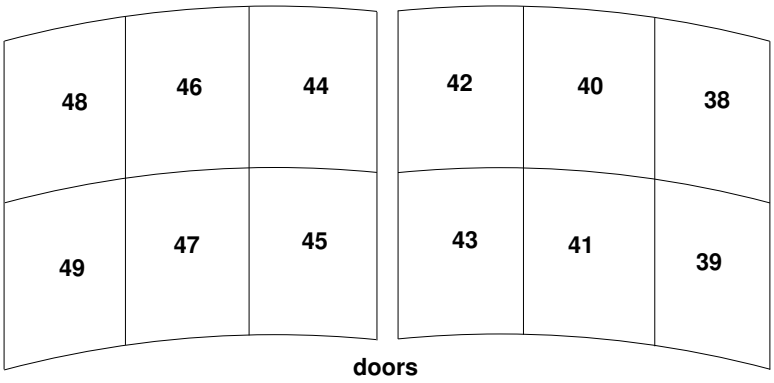
Both documents can be found on the OSCAR system at `/home/cavedemo/cave-documents`.

3.1 Location

The projectors are numbered 0 through 68, plus number 70, which is used for machine control of projectors in the repair station.

0–9	ceiling
10–37	front wall
38–43	west door
44–49	east door
50–68	floor
70	repair station


0
1
2
3
4
5
6
7
8
9
ceiling
10
12
14
16
18
20
22
24
26
28
30
32
34
36
11
13
15
17
19
21
23
25
27
29
31
33
35
37
front wall
floor
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68



3.2 Operation

Turn projectors on and off with the pjcontrol command. The pjcontrol command also is used to configure the projector color and brightness, and to log repairs and track inventory. The pjcontrol script is part of the `cave-utils` module. Load this with:

```
$ module load cave-utils
```

You can see a help message with `pjcontrol --help`.

3.2.1 pjcontrol

Synopsis:

```
pjcontrol.py [-h] [-s [SERIALNO]] [--clearErrs] [-d [MFGDATE]] [-a]
             [-R] [-G]
             [-r [{none,bulb,ballast,lens,board,install,uninstall,ship}]]
             [-l [{none,long,short}]]
             [-p [{none,spare,broken,installed,returned}]]
             [-c [COMMENT]]
             [projs] ...
```

The pjcontrol script has two forms of usage, reference by projector number, and reference by serial number. Reference by projector number looks like the following.

```
$ pjcontrol projs command
```

You can identify a projector by number individually, or in a list or range:

```
$ pjcontrol 53 on
$ pjcontrol 33,34,35 on
$ pjcontrol 33-35 off
```

The commands available are in the list below. Unique abbreviations are allowed. Some of these arguments require further args. For example 'install' requires a serial number, switch name and port, and location. And 'repair' needs a serial number.

on Turns projector on. No arguments.

off Turns projector off. No arguments.

power Returns the power status (0 = standby, 1 = warm up, 2 = imaging, 3 = cooling, 4 = error)

version Returns a software version string for the projector firmware.

mode Return the operation mode (0 = mono, 2 = stereo)

mono Set the operation mode to mono (s3d.mode = 0).

stereo Set the operation mode to stereo (s3d.mode = 2).

lamp Return the lamp mode (0 = standard, 1 = economy).

eco Set the lamp mode to economy.

std Set the lamp mode to standard.

hour Return the lamp hour counter value.

error Print the error log.

raw Send a raw command to the projector. See the projector manuals for a list of the commands. A command like the following sets the red gain to 125.

```
$ pjcontrol 13 raw red.gain = 125
```

repair Log a repair. (Please just use a single projector number for this command.) This command needs a repair type and a double-quoted comment.

install Install a projector at a given switch location. Use it like this when you're replacing a projector:

```
$ pjcontrol 42 install WACY00041
```

Use it like this if you have to replace the switch names as well:

```
$ pjcontrol 42 install WACY00041 switch03 1014 wall
```

Where the three additional arguments represent the name of the serial server switch, the number on the serial server that connects to the right projector, and an arbitrary screen name.

uninstall Takes a serial number.

report Provide a report (from the projector database.

gather Gather the variables that will be sued in a report.

Reference by serial number looks like this:

```
$ pjcontrol -s WACY00060 -r "bulb" -c "replaced bulb with s/n WYL889214"
```

Note that you cannot reference multiple projectors by serial number wild cards. The serial numbers may be parial. The projector referenced with a partial serial number is the first one in the list that matches the given fraction of a number.

-h, --help Show a help message and exit.

-s [SERIALNO], --serial [SERIALNO] A projector serial number (or fraction thereof).

--clearErrs Clear the error log for a projector.

-d [MFGDATE], --date [MFGDATE] The projector serial number stickers have a manufacturing date on them. Record it here, in the format 2012-04-17.

-a, --add Add a projector to the database. Requires that you specify a serial number and date and optional lens type, and nothing else.

-R, --report Produce a summary report about a projector. Without a serial number specified, produce a summary report about all projectors and all projector controls. Ignores all other arguments.

-G, --gather Run through all the projectors gathering all their data. Ignores all other arguments.

-r [REPAIRTYPE], --repairType [REPAIRTYPE] Records a repair. Use this to specify the type of repair, and do not forget to include a comment. The possible repair types are bulb, ballast, lens, board, install, uninstall, ship. You can also use `none` to insert a comment into the repair log.

-l [LENSTYPE], --lens [LENSTYPE] Sets type of lens (long, short, none) for the given serial number.

-p [PURPOSE], --purpose [PURPOSE] Record the current purpose of the projector (spare, broken, installed, returned).

-c [COMMENT], --comment [COMMENT] Commentary about the repair. Must be quoted if it contains more than one word.

3.2.2 pjlog

The pjcontrol command logs its usage at cave-utils/yurt/log. You can see the last few log entries with the pjlog command.

3.2.3 pjreport

Creates and stores a full report of the state of the projector database. This should be run at least weekly.

3.2.4 projd

In order to address deficiencies in the Scalable Display Manager software,[1] there is a system by which projector commands can be sent to an IP address, and forwarded to a specific projector. The projd command listens on a set of ports, and forwards the connections to the projector correponding to that port.

The projd script calls the projd.py Python script, which defines a projChats object that takes two numbers as an argument:

```
projChats(address, projector)
```

Where `address` is the last segment of an IP address that begins with 192.168.160. and projector is the number of the projector to be controlled at this address. The projd.py command uses the pjcontrol-raw script, which is not to be used otherwise.

3.2.5 pjexpect

This is part of the pjcontrol script operation and is not meant to be used independently.

3.3 Problem on power up

About one time in 200, a projector fails to power up properly and displays a message about not being able to find a signal. Since there are 69 projectors in the system, this happens fairly frequently. Toggling the stereo on the problem projector seems to address the issue:

```
$ pjcontrol 15 mono; sleep 15; pjcontrol 15 stereo
```

[1] It expects all projectors to be connected via ethernet, not serial control. This option was not available on our projectors at the time of purchase.

This is also a generally useful way to restart a projector.

3.4 Inventory and Logs

The projector database is currently held in the etc directory of the cavedemo account. It is an anydbm file, readable with the pjcontrol python script stored in the cavedemo bin directory.

```
$ /users/cavedemo/bin/pjcontrol -R
```

This produces a report of the condition of all the projectors, and a list of all the installed projectors.

The following commands will produce a report about a single projector. The first will only work on an installed projector, while the second will work on any projector in the database.

```
$ pjcontrol 53 report
```

```
$ pjcontrol -s WOCY00022 -R
```

3.4.1 Periodic logs

The pjcontrol system should have all the error and color calibration records updated periodically. Run:

```
$ /users/cavedemo/bin/pjcontrol -g
```

This steps through all the projectors, downloading their error record, color calibration, and usage time data into the master database.

> **IMPORTANT**: This is only effective when all the projectors are turned on.

This should be run weekly, with a report run and recorded at least as often. Use `pjreport` (with no arguments) to create and store a report for the current date and time.

3.5 Maintenance and repairs

> **Note**: All changes to the projectors, whether it be repairs or maintenance, bulb replacement or guts, should be logged on the repair log clipboard hanging on the tool shelf.

So far as we currently understand it, the only maintenance required is the replacement of bulbs. Policy is to replace them when they blow out, not to anticipate their demise.

All maintenance and repair actions should be recorded on the paper maintenance log. The actions relevant to a specific projector will be entered into the projector database from that record, by Tom. (You are welcome to enter them yourself with the pjcontrol script. Please record on the paper log that it was successfully entered. See section 3.4.)

3.5.1 Bulbs

Bulbs should be replaced when they blow out. We are not currently replacing them proactively. Once the color calibration is functional, we will be keeping data about color change over the bulb lifetime and may change policy after analyzing that data.

We are currently procuring bulbs from the manufacturer, but we have identified two alternate bulb suppliers at present:

Relampit Contact: Tom Sullivan, Relampit, 120 Wilbur Place, Suite C, Bohemia, NY 11716, 631-244-0051x122, tsullivan@relampit.com, $558 new, $495 relamp. Warranty runs 90 days from installation, free shipping.

Greenleigh and Wong Contact: Tom Greenleigh, Greenleigh Wong Lamps, 1680 Navajo Road, Ground Floor, Ogden UT, 84403, 801-475-7800, tom@gwlamps.com. $535 new, $361 relamp. Warranty runs 180 days from shipping.

3.5.2 Shipping for repair

Units should be shipped for repair in one of the original boxes. There are a few still around.

Before sending off, please ensure that the error record of the projector is recorded in the projector database. If the projector is not currently installed, do this by installing it at port 70 (the repair port), and issuing the following:

```
$ pjcontrol 70 gather
```

3.5.3 Reinstalling after repair

The projectors come back from repair almost certainly have the wrong firmware. To install the correct firmware, use the instructions located in the yurt room at the repair station.

Most of the time, you can get away with updating the App01 and FPGA software, and this can be done after the projector is installed, so long as the back of the projector is accessible. To do this, insert the master thumb drive into the projector, and issue the command:

```
$ pjcontrol 60 raw bootop = 4
```

(But use the projector number instead of 60.) Then reboot the projector by removing and reinstalling the power cord, with a 10 second wait in between. Then do the same thing with this command:

```
$ pjcontrol 60 raw bootop = 9
```

Then

```
$ pjcontrol 60 raw factory.reset
```

3.6 Ventilation

The wall projectors are vented directly into the AC return duct at the back of the room. It is important to make sure the lowest extension of the duct is blocked.

Where is the system controlled? How? Does it change with the heating/cooling season?

1) how can the cooling system be monitored so that problems with it dont lead to projector damage

2) is the return booster operating? How can we tell, and how do we turn it on if it isnt on

3) the makeup air intake through the roof will potentially mix in outside air, potentially reducing the return flow. How do we ensure that doesnt cause problems with this system that is cooling both people space and equipment?

3.7 Frame Lock

Sometimes the projectors get out of sync with each other, and the stereo is different every way you look.

The projectors are synchronized electronically with daisy-chain connections among the cave nodes which serve to synchronize the graphics cards. The settings do not persist through a restart of the cave X servers (which includes power interruptions or reboots). Use this procedure to reset the frame lock.

1. Log in as cavedemo. Use one of the linux terminals in the lab, or ssh -X to one of the cave nodes.
2. Run `nvidia-settings -c cave001:0`. This should start up a GUI for you.
3. In the left pane of the window, look to the bottom for “Framelock” and click that. You should see a long list of projectors appear in the main window.

4. Look in the bottom right of the window for a Disable frame lock button. Click it, wait, then click Enable frame lock. You'll see lots of green if it worked.

The nvidia-settings program is initialized with a startup file called `.nvidia-settings-rc`, currently in the cavedemo home directory. We have not had good results editing this file directly; editing configuration should be done with the GUI.

3.8 Color calibration

There is a manual color calibration procedure, using a Python program to select a projector to adjust.

Go to yurtcol directory inside cave-utils.

python yurtcol.py gives a curses window, reloads config file. This takes quite some time, so let it run before beginning to adjust projectors.

j,k to go up and down, h,l to go right to left. Mostly. The mapping up and down seems troublesome.

+,- for brightness R,r G,g B,b for specific colors

q to exit

4 Tracking, Heads and Pointers

4.1 Opti-Track tracking system

> **Note**: The Scalable calibration display clients occasionally seem to affect the sync signal received by the opti-track software. You may have to restart the framelock after calibrating the projectors. See section 3.7.

The Opti-Track system consists of a bunch of cameras aimed at a bunch of reflective balls. When three or more cameras see a particular ball, they can pinpoint its position to within a millimeter. When the cameras see three out of some random constellation of balls, such as is on the tracked glasses, or the wand, the system can pinpoint that constellation and thus the gaze or the hand of the user.

> **Note**: The Opti-Track license dongle seems to interfere with the booting of the kiosk machine. Remove the dongle to reboot and reinsert it when the computer is safely up and running.

4.1.1 Starting the Opti-Track

Double click on the "Motive" software icon. When it comes up it will ask whether you want to open an existing project. Say yes, and then choose the "Cave" project, usually at the bottom of the list.

4.1.2 Calibrating the Opti-Track

To calibrate the Opti-Track system, start it up as above, then click "Layout" and choose the calibration option from the drop-down menu.

1. Mask the visible light sources.
2. Choose calibration type ("full") and 'optiwand' option ("large").
3. Make sure that all the cameras are accounted for and reporting properly in the list on the left of the window.
4. Click start wanding and start waving the wand around slowly up and down. The cameras have some feedback capacity built into their lights. You'll see the cameras near you turn green as they are satisfied, then move over nearer another not-yet-green camera.
5. Eventually the software will tell you it has enough data for a good (or very good) calibration, at which point you can stop.
6. Click calculate.
7. Set the ground plane thing on the floor. The origin of the cave is on the seam of the true floor (not the sacrificial floor), halfway across. Put the ground corner of the ground plane over that point, and aim the long leg at the center of the screen. Level it if necessary.
8. Click ground plane.
9. The actual zero of the YURT is four feet above the ground plane, so you'll have to adjust the Y dimension of the calibration to equal -1219 (millimeters). You should see the lighting icons drop down in the display pane.
10. Click on the file menu, and save project.

4.1.3 Adding an object to the calibration

The glasses and wand are constellations of reflectors whose pattern is stored and recognized by the Opti-Track software. Sometimes you'll want to replace these items, or the metal may become bent and the system will not recognize it any more.

To replace an object in the calibration, click on the project description, then find the object in question and delete it.

To add an object, use the "View/Rigid Bodies Properties" tool. You will see two choices, Create From Visible and Create From Selected. The easiest thing to do is often to put the object to be tracked alone in the middle of the floor, leave the YURT, use the mouse cursor to select the area containing the reflectors on the object, then use the Create From Selected option. This will create a new object out of the markers visible to the system at the time. Once the object is created, you should change its name to whatever is appropriate.

> **Note**: Note The "Orientation" of the wand usually winds up pointing in exactly the opposite direction you expect. Click the "Orientation" tab and set the yaw or pitch to 180 degrees to make the pointer point the way you expect.

4.1.4 3D printer patterns

There are a variety of 3D print patterns for various YURT accessories found in the cavedemo data directory, in the `yurt-tools` directory.

4.2 Pointer

The location of the wand is tracked and reported by the Opti-track software, but the button presses and joystick operation take a different route.

The wand is a PS/2 game controller that communicates with the system via a Bluetooth dongle on the kiosk computer. If the wand's VRPN server is running, the button presses are reported as VRPN events, using the key below.

If the dongle is unplugged, you must restart the wand's VRPN server. This is on the kiosk Desktop.

To plug in the dongle, look for a small button on its end. Press the button while inserting the dongle into the kiosk computer's USB port.

Wand must sometimes be "paired" with the dongle. Sometimes this happens on battery changes, and it definitely happens when the wand is swapped for repair. The wand will complain that it is not paired by flashing its blue power light. To pair, press the button on the dongle and within eight seconds, press both the power button and the select button (the button opposite the power button on the wand). This is an undocumented feature of the wand, so these instructions are merely provisional, but this appears to work.

4.3 Polhemus tracking

The Polhemus (magnetic) tracking system is not currently installed.

5 Glasses, Stereo

The Volfoni glasses are operated by an RF signal generated by a small transmitter plugged into the sync output of the Opti-Track synch box. It is powered by a controllable plug strip, so that it can be turned off then the YURT is not in use. This allows the glasses to time out and power down.

5.1 Stereo backwards

There is currently no positive identification of the left and right eye in the stereo signal. Sometimes when the system is started up, the signals will be swapped and the stereo will appear wrong, with the viewer's left eye shown the image that belongs on the right. Worse, the stereo polarity is known to spontaneously reverse from time to time. This is irritating and we are working on fixing this problem. In the meantime, to fix the problem, you can use the controls for the Opti-Track software. (See section 4.1.)

Over on the right hand side of the Motive software window, you'll see an "Output 1" list of attributes. Beneath it, there is a Polarity setting, with a dropdown menu. Click on that option, change the polarity from inverse to normal or the other way around, and click Apply at the top of that column of the screen.

The kiosk has a program on it for checking the stereo configuration (on the System tab). It alternates red and green fields for the left and right eyes. When you hold the glasses up against any part of the YURT screen, one lens will appear red and the other green. When the glasses look like the image of the glasses in the icon, the stereo is configured correctly.

When the framelock is not working, there will frequently be areas of the screen for which the stereo is not working. If you see inconsistencies across the field of view, try resetting the framelock. See section 3.7 on Framelock.

5.2 Operating the glasses

To turn the Volfoni glasses on, press the button (not the switch) on the left temple. You'll see the lenses blink left, then right, then wait a beat, and both lenses will go dark together. When you put the glasses on you should see the stereo images.

The switch on the glasses does not do anything. (Except on one pair somewhere, where it reverses the stereo. That pair is marked with an 'M' on the temple.)

If the glasses are on, pressing the button will cause the LED on the bridge of the nose to blink. Three blinks mean fully charged, two means sort of charged, and one blink means charge me.

The glasses will not turn themselves off as long as the RF transmitter is within range. Since it covers the room, assume the glasses do not turn themselves off. To power off the glasses, hold the button

down until the lenses go clear.

6 Kiosk and Demos

The Kiosk is a Windows machine. For the YURT to operate, it needs several pieces of software to be running:

1. Opti-Trak tracking software to monitor the location of the tracked glasses and the wands. Use the Motive application, on the kiosk machine Desktop. See section 4.1 about starting it.
2. The VRPN server to monitor button presses and joystick use on the wands. This is started from a script on the Desktop called `start_joysticks.bat`.
3. A sound server to run the sound effects for CaveWriting. This is started from a script on the Desktop called `start_cavewriting_soundserver.bat`.

One thing to note is that for now the script titled `start_tracker.bat` is not used. Additionally, the scripts are currently most up to date on the Desktop, not in the Scripts folder.

In addition, the Kiosk itself is a web page, hosted on dev09. Running the kiosk involves making sure there is a browser open, and pointing to the kiosk host.

6.1 Rebooting the Kiosk

When the kiosk misbehaves and needs to be rebooted, use this procedure.

1. Remove the Opti-Track software dongle from the USB ports on the back. Remember which port it came from. When inserted, this dongle appears to prevent the machine from booting properly.
2. Reboot.
3. Replace Opti-Track dongle.
4. Restart tracking, pointer, and sound server software.
5. Start Internet Explorer and make sure it's pointing at the Kiosk web page.

6.2 Web interface

The Kiosk interface is a CGI Perl script that looks at the *yurt-kiosk* directory of the `cavedemo` account home directory and creates a web page according to the structure it finds there.

Each sub-directory of *yurt-kiosk* becomes a tab that holds a collection of demos. Each sub-directory of that contains an image and a run script, and is transformed into a button with the given image that executes the run script when the user clicks on it. So there might be a *Demos* sub-directory in *yurt-kiosk*, and inside it an *Ode* directory, containing a PNG and a run script, called `run`. This will result in a *Demos* tab on the front page of the kiosk, which will contain a button with the Ode PNG on it, and which will execute the `run` script when clicked.

To add items to the demo page, create a `run` script and include it (with that name) and a PNG file called `folder.png` in the same directory. The source for any program should be stored in compilable form in the cavedemo home directory, at *cavedemo/project-archive*, and any data it uses should be in *cavedemo/data/projects*. Please leave lots of README files around and a working make or cmake file.

6.2.1 Authorization

The kiosk has a set of accounts with which to log activity. The account authorization list is at `/users/cavedemo/etc/psd.txt`. It is a simple comma-separated list with a username and a password on each line. Use "12345" for an initial password, or to restore access, and the script will demand a user change it, and store a hashed password string when they sign in. (Yes, it is not a high-security system, merely meant to log activity in a non-anonymous way.)

6.3 Tablet

The YURT is equipped with a Samsung Galaxy S2 tablet with which to operate the kiosk web interface.

There is a "yurt-wifi" user that has a Brown Internal ID that causes it to acquire one of a specific set of IP addresses when the tablet connects to the Brown network using that ID. Those IP addresses (10.12.128.0/18) are permitted to make connections through the ccv firewall to the caveweb1 server. Maintenance of the yurt-wifi user and the hole in the firewall is the responsibility of the CIS network services group.

6.4 Demos

Many of the demo programs use data assumed to be in the */tmp/yurt-data* directories of the cave nodes. There is a model directory in the cavedemo account, see */users/cavedemo/data/model_tmp*. You can copy the *model_tmp* directory contents to the node */tmp/yurt-data* directories with the `copy_data_to_tmp` script in the cavedemo *bin* directory.

If you are writing a demo to be included in the kiosk that expects to find data in */tmp*, please use the */tmp/yurt-data* and *model_tmp* directories to load your data.

7 Scalable software, Warping and Blending

Software from Scalable Display manages the blending and warping of the projector images so that several projectors can work as one. This is commercial software, with a license that must be kept up to date. Also, the system must be recalibrated periodically, as projectors are jostled or vibrate to a different position.

The Scalable software consists of two components. There is an SDK that is linked into the YURT software, and that does blending and warping according to the current calibration. The blending and warping instructions for each projector are contained in a POL file (see section 7.3). The SDK reads the file and conforms the images produced accordingly.

The other part of the Scalable software consists of the Scalable Display Manager, which is used to manage the process of generating the POL files, also known as calibrating the screens. This software consists of two parts, the SDM itself, and the Display Clients, to be run on each of the computers involved in the screen being calibrated.

7.1 License

The scalable license file should be kept in the cavedemo z: folder, under the `scalable` directory. Look for a .licx file.

7.2 Calibrating the software

Calibrating a wall of the YURT involves pointing a camera at the wall and running the calibration software. The software illuminates the projectors one at a time, and notes its position, including the edges. This is used to calculate the zones in which the projector images overlap.

7.2.1 Running calibration

These are the steps for running a calibration. The projectors should be calibrated periodically, as well as after any work that might involve jostling or moving them. The calibration is done per wall. There are five walls in the system: front, ceiling, floor, left and right doors.

1. Turn on relevant projectors, plug in camera. Make sure the camera appears as a USB device on the console computer. Set up the camera so it is looking at the wall to be calibrated. The camera should be set on Manual exposure.[2]

[2]The camera occasionally loses the connection with the Windows machine. You'll know when the Scalable Display Manager fails to operate the camera shutter. At the moment, there is no known cure besides rebooting the kiosk machine.

2. Start Scalable display clients on each of the relevant nodes. This is done by executing RunDisplayClient.sh on each node. For convenience, there is a Calibrate icon in the System window of the kiosk that will start the display clients on all the nodes.

3. The configurations are kept on the kiosk machine, in zip files within the directory `C:\cave\scalable_configs`. You will find several zips in there, called `DefaultSystem_Ceiling.zip` and so on. These need to be unpacked into `C:\Program Files\ScalableDisplay\DEI\SystemData`. On the kiosk Desktop, you'll find a `Scripts` directory with clickable Windows scripts you can use that will clear out the previous calibration and unpack the right zip file for you. Click on the appropriate one of those scripts to copy all the calibration files into the right place and start up the Scalable Display Manager.

   If the zip files unpacked by each script are working properly, the Display Clients and Projectors should be set up correctly. But check them as follows.

4. With the SDM running, use the "Display Clients" tab to choose which display clients you are talking to. Check that the clients used by the screen you are calibrating are green and on the right.

Front wall	cave001, cave002, cave003, cave004, cave005, cave006, cave007
Ceiling	cave008, cave009, cave010
Left door	cave011, cave019
Right door	cave012, cave013
Floor	cave014, cave015, cave016, cave017, cave018

   The display client window does not scroll. If you don't see the right client on the left-hand list, you can specify the IP address directly into a box in the right column. The Display Client nodes you want are numbered 172.20.160.X, where X is the number of the cave node.

5. "Projectors" display. Your projectors should be in a 1xn row in order to make the calibration cursors appear in a later step. All the projectors you see should have the little box checked. Once the projectors are in a row, click "Next" in the lower right corner, and move on to the camera display.

6. "Cameras" display. Set the exposure for the camera down to about .25s, then click on Begin Data Collection and watch the (not very interesting) light show. The computer will turn on each projector full white, then go back and do them all with little blocks, and with a central image that defines the orientation of the screen. Set the exposure to something higher, like 5s, and click "Next" in the bottom right corner to advance.

> **Note**: When calibrating the ceiling, you will need to do it twice. Calibrate the ceiling up to this point (including the Update Calibration button on the next page), and then run the `Create_Ceiling_Masks` script in the Desktop *Scripts* directory and do the "Cameras" step again.

7. Screens tab. Use this to confirm the shape of the screen looks more or less the way you anticipate it will look, and then click Next and move on.

8. "Image Boundary" tab. Use this to set the control point positions for the calibration. There are two methods, the first involving moving cursors pprojected onto the screen itself, and the second looking at a camera image of the screen and using the mouse to move control points around on that. To use the camera method, make sure the exposure is set to a large value (about 5 seconds works well) then select "Pick boundary points within camera images." Use the mouse to drag the control point positions around the screen image.

   For the ceiling calibration, this is probably the preferred method, since the outside control points at the top of the screen (numbers 1 and 3) seem best when located a few inches outside the physical screen. (The outside edges of the screen are slightly convex, so if you put the points exactly on the screen corners, the outside edges have no image on them halfway between the corners.)

9. To use the cursor-on-wall method, click "Position points along screen boundary." Click on each of the points, and maneuver it into place by either dragging with the mouse or using the arrow keys.

   Front wall Six control points, four on the outer corners, and two at the center, at the top and bottom edges. The center isn't spiked just now, but should be.

   Floor Four control points, two are placed where the floor seams meet the front wall, and the other two are placed at the floor corners, just past the wall edge.

   Doors Place the control points at the four corners.

   Ceiling Six control points, four on the outer corners, and two central. The center isn't spiked, but should be.

   If the control points are placed in some way that the software deems impossible (maybe you have the order incorrect, for example), the software will respond with an error 79, and an incomprehensible message about not being able to write the `ScalableDisplay.ol0` file.

10. Perspective display. All the walls are defined with coordinates that place them in the front of the room. After calibration, these shapes must be rotated into the correct location. The "Eye Point" menu in the perspective display is where this rotation happens. This has been set already, and if you are using the proper *DefaultSystem* directory, you should not have to reset this.

11. After the control points have been placed, you can click Update Calibration and if it runs successfully you're done with the hard part, congratulate yourself.

12. After successful calibration, the output POL files must be renamed and distributed to their final locations. There are a set of scripts in the cavedemo directory `winders/cave/scripts` called `copy_front.sh` and the like. These are to be executed in a cygwin window on the kiosk (Windows) machine, where they can be found at `/cygdrive/z/winders/cave/scripts`.

13. Backup a successful calibration by running the script (in the same directory) called `backup_front_wall.sh` and its friends. Run in the same manner as the copy script above.

Note: The Scalable display clients seem to affect the sync signal received by the optitrack software, sometimes. You may have to restart the framelock after calibrating the projectors. See section 3.7.

These are some of the settings to the Scalable Display Manager that we have learned about. Not all of them are documented in the SDM help. Most of the time you will not need to adjust any of these settings. They are recorded here in case something goes wrong.

1. There is an OBJ file modeling the shape of the screen in *DEI/SystemData/DefaultSystem/Config*. Look for `right.obj` and `front.obj` and the like. Set this value with the "Advanced" options page of the Display Manager.
2. *ControlPoint3DLocationOverride* contains the spatial coordinates of the control points.
3. There are configuration options for the output files: *RemoteOutputDirectoryName* (directory only), also Perspective SDK Mesh File Output Path and Orthographic File Output Path. These two are complete path and file names. These are redundant with the remote output directory name, but are designed to be set redundantly. Experimentation has shown that setting these is not necessary. These settings are controlled on the "Advanced" options page of the Display Manager.
4. Camera settings: ISO 200, .125 sec, f/11.
5. `C:\ScalableDisplay\ScalableServiceInformation` has JPEG images in it for each projector. These can be used to generate masks for calibration.

7.3 Location of POL files

The installed POL files (the output of the Scalable calibration step) are located in the cavedemo account home directory, in */users/cavedemo/scalable/cave*.

7.4 Scalable libraries

Pointer to Scalable documentation?

mgt.oscar.ccv.brown.edu

Linking them in.

7.5 Making a model

What constraints on the model?

via blender, via kinect?

On Win machine, logged in as last mohican (hawkeye), there is said to be a file with a modest amount of documentation about how to use and install obj files. Lies, all lies!

7.5.1 Importing the model

How to get it read by the Scalable Display manager thing.

John's zipfile hack.

ScreenTypeOverride

8 Software

This section describes where some of the software used in the YURT can be found and how it is used.

8.1 vrg3d

To use vrg3d, use the `module load cave`.

8.2 VRUI

Libraries, APIs,

9 Construction

Knot for tying monofilament, "Nanofil"

Mirror suppliers

10 Granoff Cave

This section of this document is about the old VR installation, herein called the "cave"

10.1 Startup

Start the various servers from the start menu, not from the desktop.

The VRPN server will start up, but might not start sending frames if an OpenGL program is not running somewhere. Try starting it while the "Align Screen" (a simple OpenGL program that uses no input) application is running.

10.1.1 Frame lock

On a restart, you should check to see if the frame lock is still set. Log in to the caveserver as cavedemo, start nvidia-settings -c cave5:0, and then add cave5 in the framelock panel. See section 3.7 for more.