RAIN TABLE

TUTORIAL

SECTION PAGE

1 Compiling..... 2

2 Preparation of Data..... 5

2.1 High-Resolution Images.....6

2.2 Digital Elevation Models..... 10

2.3 Final Data Folder.....12

3 Running.....13

3.1 Data Configuration.....13

3.2 Tile Configuration.....15

3.3 Lambda Table Configuration..... 19

3.4 Run Time Configuration..... 20

3.5 Start Up Scripts.....21

3.6 Default User Interface.....22

3.7 Interaction Modes..... 22

3.8 Alternative User Interfaces.....22

1 Compiling

There are five executables that will be created after successful compilation:

- pyramidmaker ... used to process image data
- machinefilegen ... used to create a machine file to start up rain table
- sinkfill ... used to process elevation data
- raintable ... rain table application
- parallel ... a server program

The libraries required to compile Rain Table are not included with the standard download since Rain Table does not require a specific linux distribution. The libraries can be installed via installers that come with your linux distribution or they can be obtained from other sources and compiled individually. Here is a list of required libraries:

- Graphics
  - OpenGL (opengl.org)
  - GLEW (glew.sourceforge.net)
- Windowing
  - FLTK 1.17 (www.fltk.org)
  - SDL (www.libsdl.org)
  - SDL_image(www.libsdl.org/projects/SDL_image)
- Fonts
  - FreeType2 (freetype.org)
- Data
  - TIFF (www.libtiff.org)
  - JPEG (www.ijg.org)
  - PNG (www.libpng.org)
  - SQUISH (http://www.sjbrown.co.uk)
- Cluster
  - MPI (mcs.anl.gov/mpi)
- Networking
  - QUANTA (www.evl.uic.edu/cavern/quanta)

- Sound
    Ogg Vorbis (www.vorbis.com)
    OpenAL (www.openal.org)

Packages available under Suse:

Basis Devel
C/C++ Devel
Kernel Devel

Individual Package Selections:
libtiff-devel
libpng-devel
libjpeg-devel
fltk
fltk-devel
SDL-devel
SDL-image-devel
openal-devel
freealut-devel
libogg-devel
libvorbis-devel

Compiling from scratch on LINUX:

1. Check out source code

    svn checkout svn://cube.evl.uic.edu/waterplanet/raintable_lt

3. Type

    sh build.sh

    If there is a problem with .h files not being found:

    Open makefile.in in the root directory and put include and library paths in appro-

    priate fields. For example:

```
FLTK_INCLUDE=/usr/local/include
GLEW_INCLUDE=/usr/local/include
GL_INCLUDE=/usr/X11R6/include
FREETYPE_INCLUDE=/sw/include
SDL_INCLUDE=/sw/include/SDL
MPI_INCLUDE=/
JPEG_INCLUDE=/sw/include
PNG_INCLUDE=/sw/include
TIFF_INCLUDE=/sw/include
SQUISH_INCLUDE=/Users/dmitri/dev
QUANTA_INCLUDE=/usr/local/include
OGG_INCLUDE=/sw/include
OPENAL_INCLUDE=/sw/include

FLTK_LIB=/usr/local/lib
GLEW_LIB=/usr/local/lib
GL_LIB=/usr/X11R6/lib
FREETYPE_LIB=/sw/lib
SDL_LIB=/sw/lib
MPI_LIB=/
JPEG_LIB=/sw/lib
PNG_LIB=/sw/lib
TIFF_LIB=/sw/lib
SQUISH_LIB=/Users/dmitri/dev/squish
QUANTA_LIB=/usr/local/lib
OGG_LIB=/sw/lib
OPENAL_LIB=/sw/lib
```

2 Preparation of Data

There are two types of data that Rain Table needs: image data and digital elevation (DEM) data (Figure 1). The image data should come in either JPG or TIFF formats. The elevation data should be in Binary Terrain (BT) format (http://www.vterrain.org/Implementation/Formats/BT.html). They do not have to be of the same dimensions, but they should have the same aspect ratio. Several GIS programs are able to convert elevation data to BT. One of these is Global Mapper (http://www.globalmapper.com). Another one that is also freely available is LandSerf (http://www.soi.city.ac.uk/~jwo/landserf). Instructions for LandSerf:

1. Open your elevation data LandSerf. Acceptable formats: ArcGIS, USGS DEM
2. Save it as a Binary Terrain file (in LandSerf it is called Virtual Terrain Project raster).

Compiled data utilities for Windows XP are located here:

http://www.evl.uic.edu/cavern/mc/raintable/datautilities-win32.zip

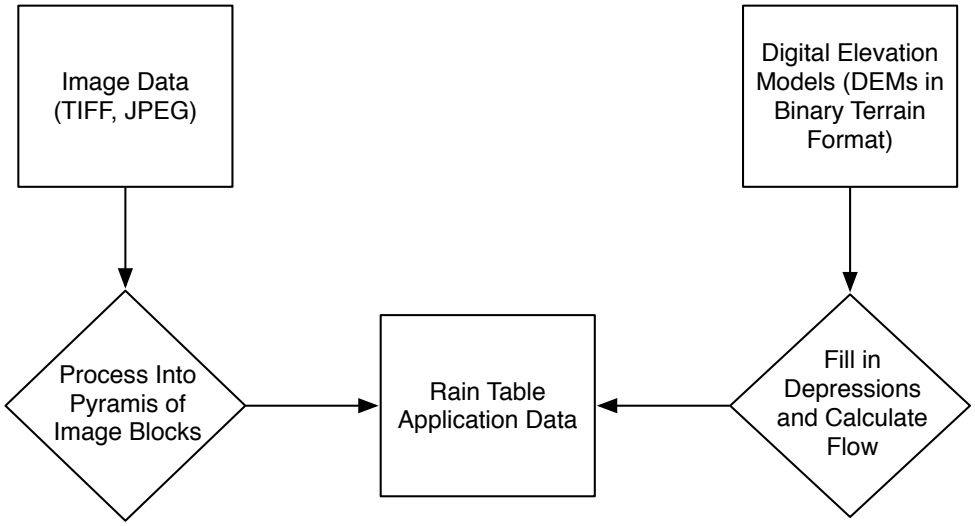


Figure 1. Data preparation chart.

A sample dataset can be obtained at:

http://www.evl.uic.edu/cavern/mc/raintable/bigisland.zip (27 mb)

2.1 High-Resolution Images

This step requires 2 configuration files. The first file represents the layout of the entire image. The second file is a configuration file that contains some information about the image and a pointer to the layout file.

To prepare image data:

1. Create a layout file, follow the format outlined in Figure 2.
2. Create a configuration file, follow the format outlined in Figure 3.

12523 12894	Dimensions of the entire image.
1 1	Number of columns by rows (if the image comes in broken up into tiles). This should be 1 by 1 for a single non-tiled image.
12523 12894	Dimensions of the image block. This should be the same as the dimensions of the entire image for non-tiled images.
d:/data/europe.tif	A list of the tiles goes here. This list would have only one item if the image is non-tiled. For tiled images, the files should be listed column by column from first to last, for example if your image consists of 3 x 4 tiles, and say, each tile has a name FILE_COLUMN#_ROW#, then the list would look like: FILE_0_0 FILE_0_1 FILE_0_2 FILE_0_3 FILE_1_0 FILE_1_1 FILE_1_2 FILE_1_3 FILE_2_0 FILE_2_1 FILE_2_2 FILE_2_3

Figure 2. Layout file d:/data/layout.conf.

europe	Name of image data as it should appear on disk (as a data folder).
12523 12894 3	Dimensions of the image data.
1	Number of frames for the image data. This should always be 1 unless the image is a time-dependent dataset. This is followed by a list of image layout files representing time steps.
d:/data/layout.conf	Pointer to layout file.

Figure 3. Configuration file.

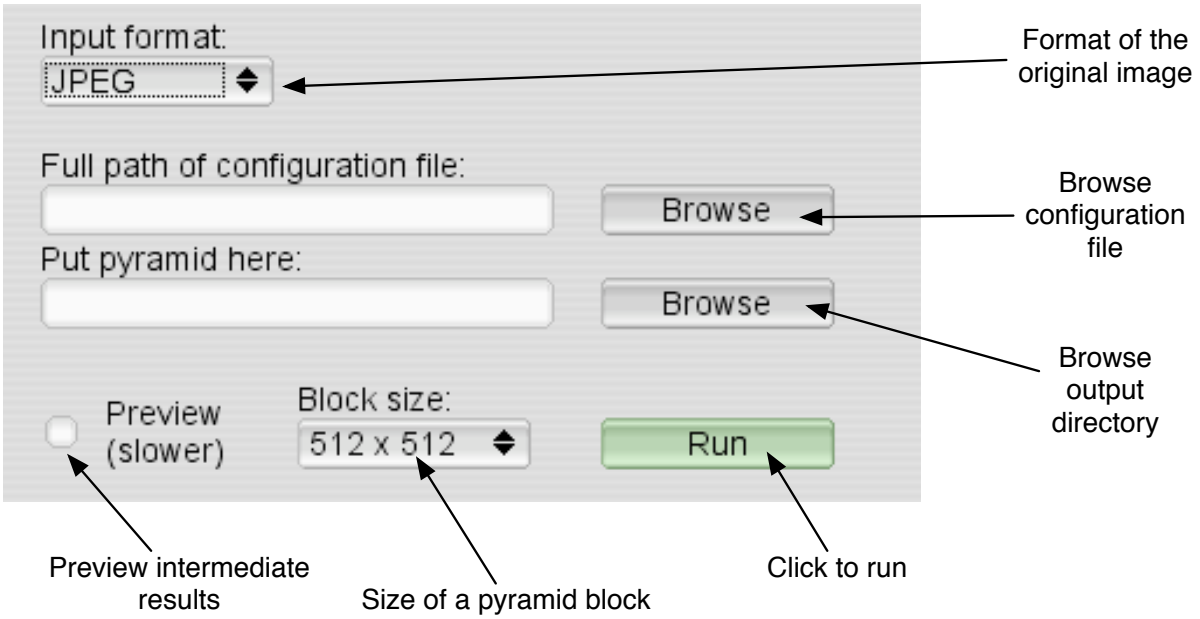


Figure 4. User interface for image data processor.

After creating the two files:

1. Run pyramidmaker by typing:

   ./pyramidmaker

   Or double clicking under Windows.

2. You should see the user interface on the screen (shown in Figure 4).

3. Choose format of the original image.

4. Browse to the configuration file.

5. Choose output directory. This directory is where the data folder will end up.

6. Choose preview if not sure that the configuration files are correct. This option processes the image while displaying intermediate image blocks.

7. Leave the block size in its default setting.

8. Click Run.

After running pyramidmaker, you should see a new folder appear in the directory specified under “Put pyramid here”. This is the image data folder.

2.2 Digital Elevation Models

To create elevation data for use in Rain Table:

1. Run sinkfill by typing:

    ./sinkfill

    Or double clicking under Windows.

2. The user interface shown in Figure 5 should show up on the screen.

3. Browse the original elevation file.

4. Set data shrinking factor (default at 1). This integer represents the factor by which to reduce the dimensions of elevation data. For example, if the original data is 9,000 by 9,000 then a factor of 3 will reduce the size to 3,000 by 3,000. This can potentially improve the time it takes to process the data.

5. Set maximum size of a depression to fill (default at 200). This number represents the maximum size of a depression to fill in. This will depend on the resolution and the dimensions of data, but should be at about 1% of the number of data cells in the grid:

    Threshold = (Horizontal data dimension * Vertical data dimension) * 0.001

6. Set elevation exaggeration. This will increase any values in the original elevation data by the specified factor. This can be used to create steeper slopes.

7. Set empty data value (default at -32768 in Binary Terrain data format).

8. Set ground elevation value. This value is usually 0, but may be a negative number for various datasets where sea level is not assumed to be at 0.

9. Preview may be selected to look at the intermediate data. The shrinking factor should be set to a high number for large data when the preview is selected.

10. Click Run.

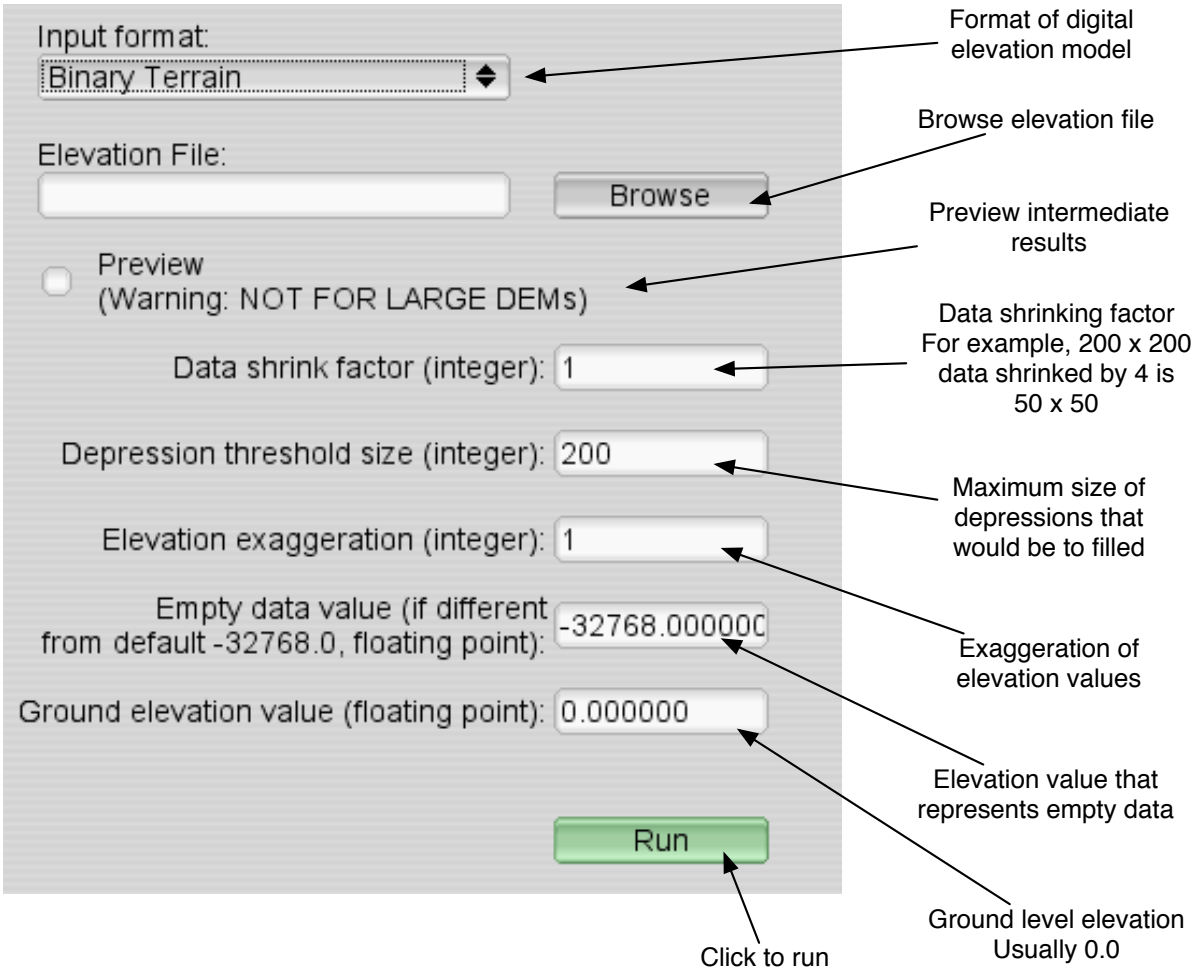


Figure 5. User interface for elevation data processor.

Assume that the original elevation data file name was sample.bt. After running sinkfill, there should be 4 extra files in the directory where sample.bt is located. They are:

sample.bt.x

sample.bt.y

sample.bt.accum

sample.bt.elev

2.2 Final Data Folder

The final step is to move the data created by sinkfill to the data folder created by pyramid-maker:

1. Figure out what the name of the image data folder is. Let's assume it is "europe".
2. Rename the 4 files created by sinkfill to:

sample.bt.x	to	europe.dem.x
sample.bt.y	to	europe.dem.y
sample.bt.accum	to	europe.dem.accum
sample.bt.elev	to	europe.dem.elev

3. Drop these renamed files into the image data folder.
4. This folder is now a usable data folder for Rain Table.

3 Running

Rain Table uses a few configuration files. Some of the files have to be created from scratch (or from an example file), others are located in bin/appdata/raintable/conf folder.

3.1 Data Configuration

The following table shows an example of a data configuration file. This file contains information about all of the datasets that should be loaded by Rain Table. This file resides in bin/appdata/raintable/conf/datasets/conf.

Datasets 2	Keyword "Datasets" followed by the number of datasets. The information after this line is the list of datasets.
Data	First dataset.
NameOnDisk europe	The name of the data folder on disk.
PathOnDisk d:/data	Relative path to the data folder. In this case, the full path is d:/data/europe
Scale 1.0	Scale of the dataset as it will appear initially in Rain Table application. 1.0 represents 1:1 scale, larger number scales up the data and smaller scales it down.
Translate 0.0 0.0	Translation of the dataset as it will appear initially.
Parent none	This is for nesting datasets inside of another datasets to create multi-resolution data. Keyword "Parent" should be followed by the "NameOnDisk" of the parent dataset.
PolyAttachments 0	This is for attaching polygonal data on top of a dataset. The keyword "PolyAttachments" should be followed by the list of geometry files in OBJ format.
TextAttachments 1 0.0 0.0 this_is_europe	This is for attaching text labels on top of datasets. The keyword "TextAttachments" should be followed by a list of labels. In this case, the label will appear at (0.0, 0.0) relative to the dataset, displaying "this is europe". The underscores represent spaces.
Data NameOnDisk bigisland PathOnDisk d:/data Scale 0.5 Translate -2000.0 0.0 Parent none PolyAttachments 0 TextAttachments 0	Second dataset.

Figure ?. Example of a data configuration file.

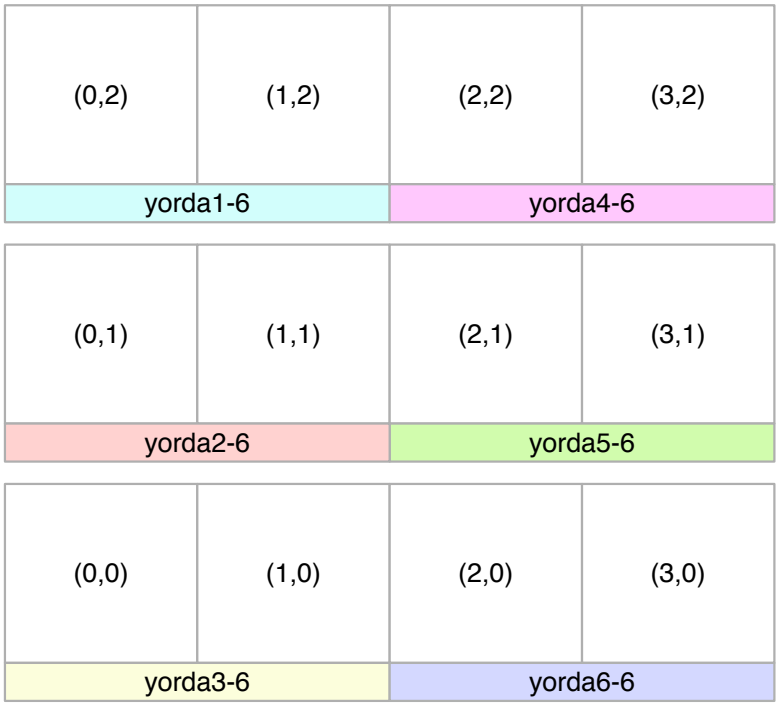
3.2 Tile Configuration

This file in the following table represents the layout of screens in your tiled display.

TileDisplay Dimensions 1 1 Mullions 0.625 0.625 Resolution 1280 1024 PPI 90 Machines 1	The keyword TiledDisplay must appear at the top of this section. Dimensions: This is the number of columns and rows of tiled display. Here, there is only a single screen, so the number of columns and rows is 1. Please note, the order of dimensions is columns followed by rows. Mullions: The value for this parameter should indicate the width of the horizontal and vertical mullions in units of inches. Here, the display has an equal border on each side of the display of 0.625 inches (or for fans of fractions, 5/8"…but use decimals in the configuration file). Resolution: This is the screen resolution of each tile. Here, the resolution is 1280x1024. PPI: This value is the pixels per inch of each tile. Here, there are 90 pixels per inch. Machines: This represents the number of display nodes which drive tiled display for each DisplayNode. For each "machine", make a block of text whose title has the keyword DisplayNode.
DisplayNode Name chewbacca IP 99.6.30.2 Monitors 1 (0,0)	Name: This is the host name of each display node. Here, the name is chewbacca. IP: This is the IP address of each display node. Here the IP address is 99.6.30.2. Monitors (tileColumn, tileRow): This is the number of tiles which is driven by each node. The coordinates provide the column and row position of the display in the tiled display. In this case, there is only 1 display, so the tile location is (0,0).

Figure ?. Example of a tile configuration file.

Here is an example of creating a tile configuration file for a specific tiled display called "yorda". The screens in this tiled display are arranged in the following manner:



The corresponding tile configuration file will be:

```
TileDisplay
    Dimensions 4 3
    Mullions 0.5 0.5
    Resolution 1280 1024
    PPI 72
    Machines 6


DisplayNode
    Name yoda1-6
    IP 60.60.7.128
    Monitors 2 (0,2) (1,2)


DisplayNode
    Name yoda2-6
    IP 60.60.7.129
    Monitors 2 (0,1) (1,1)


DisplayNode
    Name yoda3-6
    IP 60.60.7.130
    Monitors 2 (0,0) (1,0)


DisplayNode
    Name yoda4-6
    IP 60.60.7.131
    Monitors 2 (2,2) (3,2)

DisplayNode
    Name yoda5-6
    IP 60.60.7.132
    Monitors 2 (2,1) (3,1)

DisplayNode
    Name yoda6-6
    IP 60.60.7.133
    Monitors 2 (2,0) (3,0)
```

TileDisplay Dimensions 4 3 Mullions 0.5 0.5 Resolution 1280 1024 PPI 72 Machines 6	Remember to begin the file with the keyword TiledDisplay. Since there are 4 columns and 3 rows, the dimensions parameter should have the values of 4 3. The Resolution parameter should be set to 1280 1024. PPI should have the value 72. Finally, since there are 6 machines in the cluster, we need to provide the value of 6 for the parameter Machines.
DisplayNode Name yoda1-6 IP 60.60.7.128 Monitors 2 (0,2) (1,2)	Remember to begin each display node section with the keyword DisplayNode. In general, it should not matter how display nodes are specified in the configuration file, as long as the tile arrangement matches your plans. Again, the name of the node is optional, but it is much easier to remember names of things rather than cryptic IP addresses. The IP address of the display node yoda1-6 is 60.60.7.128. It runs 2 displays in the tile positions (0,2) and (1,2). Please note that if the display node controls more than 2 displays, each tile's coordinate would have to be listed here as a value for the monitors' parameter.
DisplayNode Name yoda2-6 IP 60.60.7.129 Monitors 2 (0,1) (1,1)	The IP address of the display node yoda1-6 is 60.60.7.129. It runs 2 displays in the tile positions (0,2) and (1,2).
DisplayNode Name yoda3-6 IP 60.60.7.130 Monitors 2 (0,0) (1,0)	The IP address of the display node yoda1-6 is 60.60.7.130. It runs 2 displays in the tile positions (0,0) and (1,2).
DisplayNode Name yoda4-6 IP 60.60.7.131 Monitors 2 (2,2) (3,2)	The IP address of the display node yoda1-6 is 60.60.7.132. It runs 2 displays in the tile positions (2,1) and (3,1).
DisplayNode Name yoda6-6 IP 60.60.7.133 Monitors 2 (2,0) (3,0)	The IP address of the display node yoda1-6 is 60.60.7.133. It runs 2 displays in the tile positions (2,0) and (3,0).

3.3 Lambda Table Configuration

The file located in bin/appdata/raintable/pucks.conf maps actions to puck ids. The following table describes this file.

Rotate 2	Rotation mapped to puck 2.
Pan 1	Panning mapped to puck 1.
Zoom 3	Zooming mapped to puck 3.
	The following keywords represent lists of pucks. A keyword is followed by the number of objects, which is followed by a list of puck IDs.
Magnifiers 1 5	One magnifier mapped to puck 4.
Inputs 4 1 2 3 4	Four inputs (rain spots) mapped to pucks 1, 2, 3, and 4.
Modifiers 0	No modifiers (this is not currently in Rain Table).
Triggers 1 5	One trigger mapped to puck 5. This is used for triggering volcanos over an area.
	Note: a single puck can be mapped to multiple actions...in this case puck 3 is mapped to be one of the inputs and a zooming tool.

3.4 Run Time Configuration

The configuration file bin/appdata/conf/mconfig.conf represents some run time configuration. However, most of it is internal to Rain Table and should be left unchanged. The following table is an example of this file.

DataFetchStyle nfs	Leave unchanged.
ClusterSync on	Leave unchanged.
MasterWindow on	Leave unchanged.
MainCacheMax 400	Leave unchanged.
TexCacheMax 200	Leave unchanged.
Magnifiers 1 512	The keyword "Magnifiers" is followed by the number of magnifiers and their radius. This should be consistent with the number listed in pucks.conf file.
MasterIP 10.0.8.10	IMPORTANT: IP address of the master node. Rain Table will not start properly if this IP address is wrong.

3.5 Start Up Scripts

Example start up scripts are located in bin folder (start_parellel_teiburu.sh and run_raintable_teiburu.sh).

The first script will be used to start up a program required to run Rain Table. The name of the program “parallel” is followed by the number of nodes in your cluster (start_parellel_teiburu.sh).

```
parallel 3
```

The main script is (run_raintable_teiburu.sh):

```
machinefilegen teiburu-3.conf
start_parallel_teiburu.sh &
mpirun -np 4 -machinefile machinefile raintable teiburu-3.conf
```

machinefilegen takes a tile display configuration file; teiburu-3.conf is the tile configuration file used in this setup.

start_parallel_teiburu.sh executes the script described earlier.

mpirun argument -np is followed by the number of machines in the tiled display cluster (including the mater node); argument -machinefile is followed by the file file that contains IP addresses of all machines in the cluster - this file is generated automatically by machinefilegen; next is the Rain Table executable (raintable) followed by the tile configuration file.

This script is all that one needs to start up Rain Table.

3.6 Default User Interface

3.7 Interaction Modes

3.8 Alternative User Interfaces

TO DO