# RAVEPC: Remotely Accessible Visualizer & Explorer of Point Cloud

An Interactive Visualization Application for LiDAR Data : Part I

**Authors:** Beena Kumari, Avijit Ashe and Jaya Sreevalsan Nair

Graphics Visualization Computing Lab,
International Institute of Information Technology Bangalore,
26/C Electronics City, Hosur Road, Bangalore 560100, India.
http://cds.iiitb.ac.in/gvcl

*Software Manual Ver-0.0, August, 2015*

*This document is a Part I of 3-part series of the user manual of our project, "LAN-based Interactive Visualization of Three-dimensional LiDAR Point Cloud", submitted to Department of Science and Technology (DST), Government of India. Part I comprises the instruction for installation and point reduction. Part II contains instruction for feature tracking and Part III includes procedure for remote visualization, and appendix section.*

# Software Description

RAVEPC is an open source desktop application for remote visualization of 3D range data. It can also be used to manually track the features in time-varying 3D range data. Currently, it supports las, ply and pcd data file format and is developed for Ubuntu operating system. It has server-client architecture where server should have high-end graphics card with all computational capability and clients can be used as thin devices for display purposes. User interface has been written using C++, fltk library and OpenGL. Algorithm is implemented using C++, PCL library and CUDA. Remote visualization is done using an open source product ThinLinc developed by Cendino AB. RAVEPC supports following features:

- RAVEPC is a stand alone desktop application for visualization and exploration of 3D Range data.
- It supports las, ply and pcd data file format.
- Points Classification into Geometrical classes
- Manual Feature Tracking up to 3 time-stamp data
- Remote visualization and analysis of the 3D range data
- Mesh Construction from unstructured 3D point cloud

# Disclaimer

# Contents

## Part I

# Part II



# Part III

# List of Figures

# Chapter 1

# RAVEPC

## 1.1 Introduction

RAVEPC, Remotely accessible visualizer and explorer for points clouds, is a software for LAN-based interactive visualization of 3D range data. It is mainly designed for visualization and exploration of 3D airborne lidar data. It supports only las, pcd and ply data format. It can be used for web-based interactive visualization of 3D range data. RAVEPC is a standalone application for exploration and visualization of 3D point cloud. RAVEPC has to installed on a server with high-end graphics card and computational capability. Remote visualization is achieved by using a product called ThinLinc developed by Cendio AB. Thinlinc has to be installed on server as well as client for remote visualization. Installation procedure has been discussed in this chapter. There are two softwares which has to be installed:

- RAVEPC
- ThinLinc

## 1.2 Installation of RAVEPC

It stands for Remotely Accessible Visualizer and Explorer of Point Clouds and as it is a standalone product with a native user interface, it does not need any other additional package to run and thus, there is no extension or suffix to the name. RAVEPC has some dependencies which need to be installed before installing RAVEPC.

### 1.2.1 Prerequisites

- *sudo apt-get update*
- *sudo apt-get install build-essential*

- *sudo apt-get install freeglut3 freeglut3-dev*
- *sudo apt-get install fluid*
- *sudo apt-get install cmake cmake-curses-gui*
- *sudo apt-get install libboost-all-dev libeigen3-dev libflann-dev libvtk5-dev libvtk5-qt4-dev*
- *sudo apt-get install libglew-dev libxmu-dev libsuitesparse-dev libqhull-dev libpcap-dev libxmu-dev libxi-dev -*
- Install OpenNI
  - *sudo apt-get install libopenni-dev libopenni-sensor-primesense-dev -y*
- Install CUDA (We have tested with cuda 5.5 on ubuntu 12.04)
  - *sudo apt-get install nvidia-cuda-dev nvidia-cuda-toolkit -y*
  - *sudo dpkg -i ⟨package.deb⟩*
  - *sudo apt-get update*
  - *sudo apt-get install cuda -y*

  Set the environment variables for CUDA libraries and binaries. Add the cuda's bin directory to your path.
- Install PCL-1.7. Do not install PCL from terminal or synaptic packet manager. Download PCL source code and compile it using cmake with cuda support. Build the compiled code and install it. Getting the Source code:
  - *sudo apt-get install git -y*
  - *git clone https://github.com/PointCloudLibrary/pcl PCL-trunk-Source*

  Compiling the PCL source code. Go the the PCL source directory (pcl-pcl-1.7.1/ or PCL-trunk-Source/), and create a new subdirectory to keep the build files in:
  - *mkdir build*
  - *cd build*

  Build the PCL code in Release (fully optimized, no debug capabilities) mode now as shown, and customize the rest of the options later. CMake should be able to find every dependency, thus being able to build every subsystem except for the ones marked as "Disabled by default".
  - *cmake -DCMAKE_BUILD_TYPE=Release ..*
  - *ccmake .*

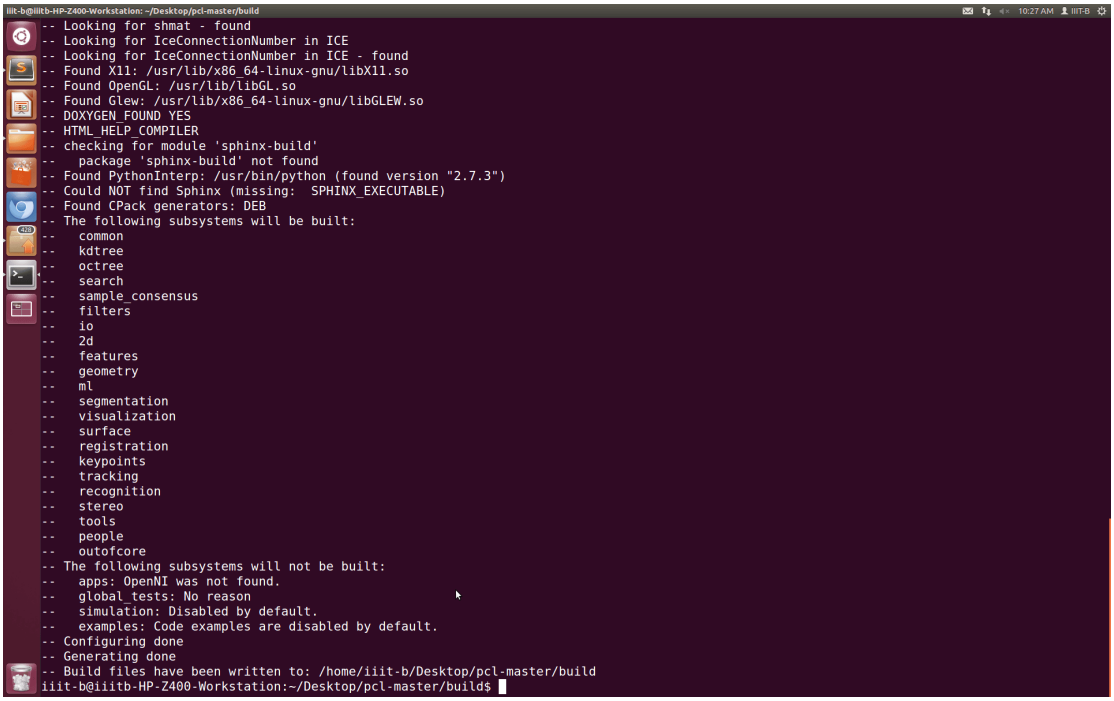


FIGURE 1.1: Dependecies to build the PCL Source code

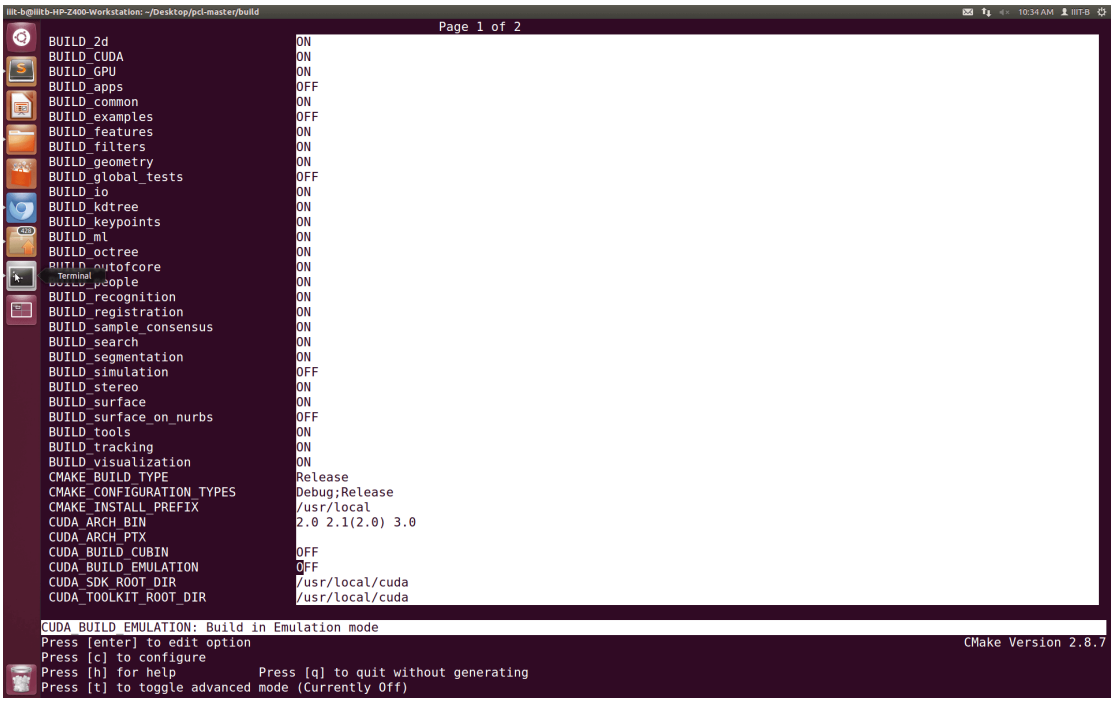


FIGURE 1.2: Interface of CMake to build PCL source code

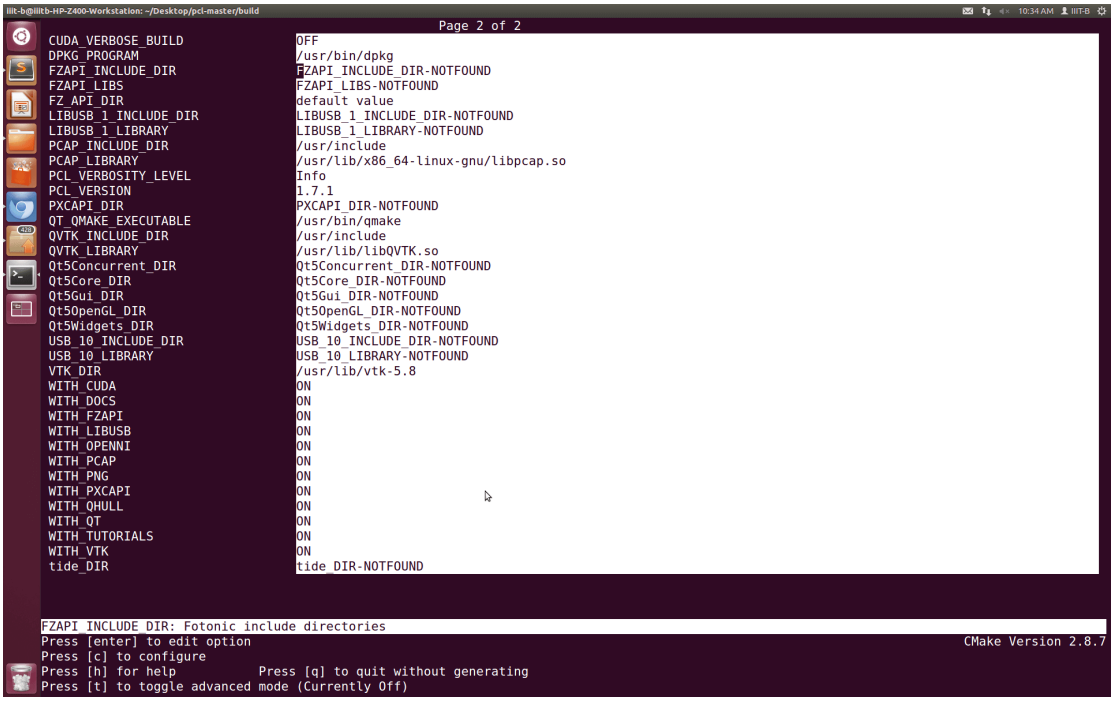


FIGURE 1.3: Interface of CMake to build PCL source code

The interface of CMake to build pcl source code has been shown in Figure 1.2 and Figure 1.3. Try turning all functionality "ON" except those whose dependencies are not installed. Enable the CUDA options and give CMake the path to your SDK. Go to the "CUDA_SDK_ROOT_DIR" option and enter the correct path. When you are done, press C to configure and G to generate and exit the tool. Now build the code.

– *make*

Install the PCL:

– *sudo make install*

### 1.2.2 RAVEPC

Every package of RAVEPC has the following directories and files. There are directories named bin, Data, obj, output-files and src. RAVEPC also contains makefile and readme file.

- The *bin* directory contains RAVEPC executable.
- The *Data* directory contains and must contain the point cloud LiDAR dataset you want to test with RAVEPC-PV, that support .pcd, .ply and .las extensions, as of now.
- The *obj* directory contains obj files for RAVEPC.
- The *outputfiles* directory contains subdirectories: $line\_feature$, $Original\_Point\_Cloud$, $Reduced\_Point\_Cloud$ and $surface\_feature$. These subdirectories contain files for freature tracking.

- The *src* directory contains source files for RAVEPC.
- There is a *makefile* text file which compile and build the RAVEPC source code.
- There is a *readme* text file which guides you through the proper installation procedure for all the related libraries and dependencies to run the application properly.

- Install the RAVEPC

  Go to RAVEPC folder and run the make file. Make sure that cuda, eigen and pcl compiler path in make file is same as you set in environment variables. For example, By default, cuda compiler path is /usr/local/cuda-5.5/bin/nvcc. Change the path to your cuda installation folder. Similarly for pcl and eigen library. Also make sure the path of cuda's, eigen's and pcl's includes and libraries files in make file are also set to their installation folder. By default, It is set to /usr/local. Now run the make file which creates the *./app* executable in bin folder. To run the RAVEPC, go to bin folder and run the *./app*. Go to RAVEPC folder, run the following command on the terminal:

  - *cd RAVEPC*
  - *make clean*
  - *make*
  - *cd bin*
  - *./app*

## 1.3 ThinLinc

The detailed information and documentation about the set-up, the requirements and prerequisites, optional and mandatory configurations, administrative tasks, different user interfaces available, installation guides for various platforms et cetera can be found at the official website of ThinLinc at the following link http://www.cendio.com/resources/docs/tag/index.html. This is the link to the latest version of ThinLinc at the time of writing and what was tested in our set-up(s), which is 4.1.1. There may arise certain disambiguation while following this experimentation guidelines, specifically, for any other version or platform.

We have tested the following Linux distributions for server set-up successfully. Ubuntu 11.04 Natty Narwhal (Intel HD Graphics) Ubuntu 12.04 LTS Precise Pangolin (Nvidia Graphics) Ubuntu 10.04 LTS Lucid Lynx (ATI Radeon 5730) As of the ThinLinc official documentation any LSB and Solaris machine can be configured to become the Master ThinLinc Server with the server set-up executable file (install-server) but we have tested in only on Linux base as it was our requirement here.

### 1.3.1 System Requirements

The system requirements can be viewed in detail at their official website documentation link. It is divided into two parts: server requirements and client requirements. The server requirements

can be found at http://www.cendio.com/resources/docs/tag/requirements_server.html. The client requirements are practically any device with an OS running on it or even if it is a dumb terminal ThinLinc provides TLCOS (ThinLinc Client OS) to enable it to perform as a client.

### 1.3.2 Download Requirements

As ThinLinc offers a server-client model for remote hardware-intensive computation, it ships its required executables in two separate modules; one for setting up the server machine called the VSM Server and other for setting up the client machine called simply 'tl-client'. Depending upon the specifications of oner machine one would need to install specific server-client package on them. They can be downloaded from their official website following the link http://www.cendio.com/downloads/. This link allows one to download the server set-up executable upon creation of an account which is merely for verification purpose and may be used for other notifications only. About the client one can follow the same link and click upon download clients and choose from among them what one need for oner installation. There is also an option for one to create an account on their demo servers and test their product before one actually implement it on oner site. So, one might just download a client for oner machine; say one work on windows then download windows client; and test their services provided by their terminal and application servers. The terminal server is the one that has been configured by installing the server set-up file, just mentioned above to download from that link. The application servers are extra servers linked to this terminal server (VSM Server or ThinLinc Master Server) and are called as VSM Agents. They do not need any installations but just their addresses to get connected and their resources can be used as per the conditions set at the VSM Server (the one one will be the admin of once one install the server set-up file on it, after downloading it from the website using the provided link).

### 1.3.3 Installation

A detailed guide to start installing the ThinLinc terminal server (creation of VSM Server or Master Server) can be found at their official website by pointing the browser at https://www.cendio.com/resources/docs/tag/ch03s04.html. Using the graphical way is straight and simple. The downloaded file tl-4.1.1-server has the following contents which has the executable file named 'install-server'. The figure provided below were screen shots obtained from a Ubuntu 12.04 Desktop Version running terminal. This is shown in the following figure 1.4

On double clicking the executable file named 'install-server' it guides through all the supplementary installation of packages missing from the system, sets up default configurations, installs the dependencies of the newly installed packages and finally lets the user select a username and a password for authenticating the server as its administrator. Following Figures 1.5, 1.6, 1.6 and 1.7 list out the initial steps.

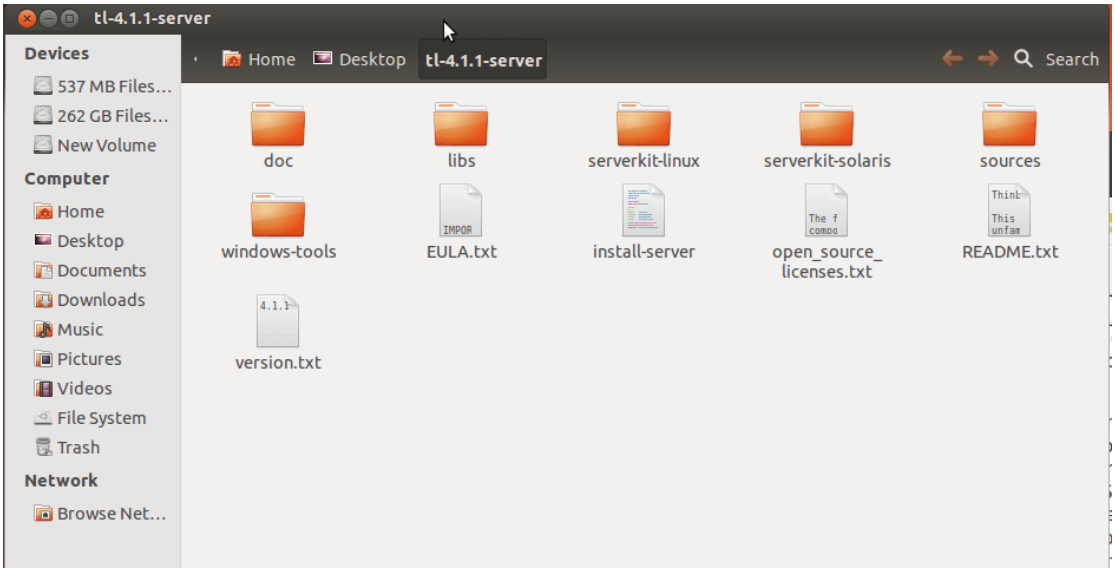


FIGURE 1.4: ThinLinc file directory

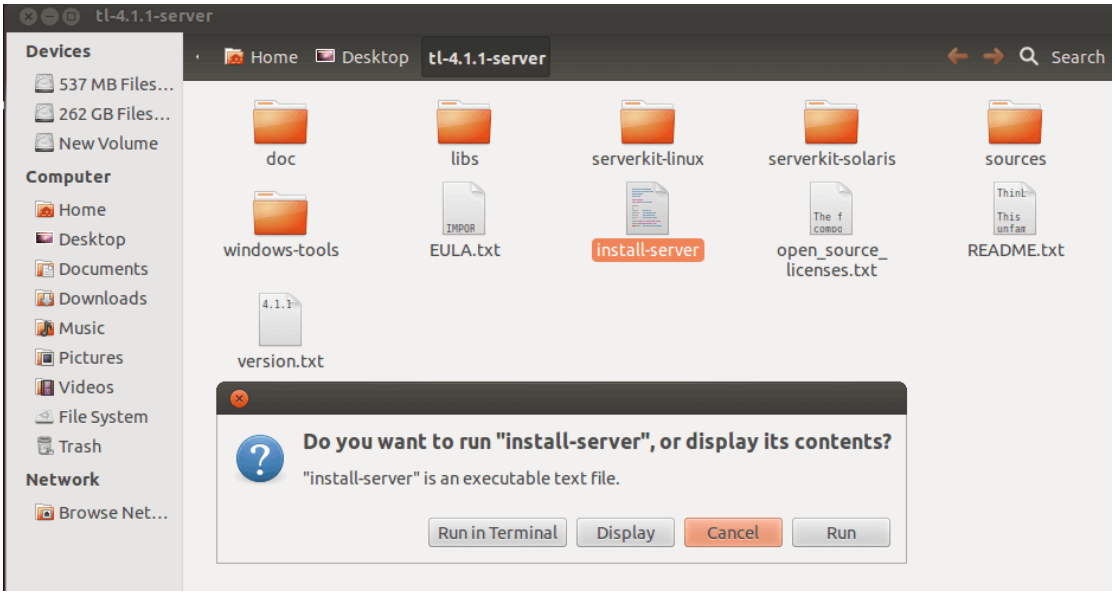


FIGURE 1.5: ThinLinc Installation step 1

### 1.3.4 Creating Sessions

Creating sessions from a client machine is very simple and there are three ways to do so. First one using the native client application installed on the system, second one using a web interface which again provides two options: to either launch the native client or directly connect to the server via the browser. In case of making a connection via the browser there is no need of any client set-up to be installed on any system. One just need to have a browser with Java installed on it and even some later releases of ThinLinc provide a HTML 5 based version which will allow one to connect to the servers by just pointing oner browser to the IP address of the terminal server thereby eliminating the device dependency completely altogether. This will make it super flexible and super device independent.

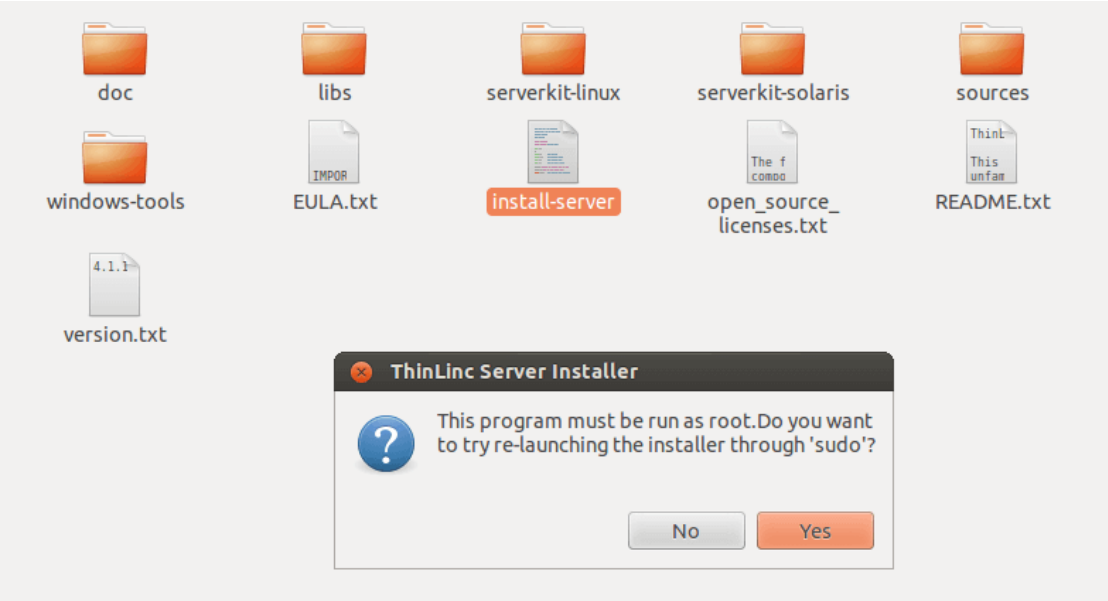


FIGURE 1.6: ThinLinc Installation step 2

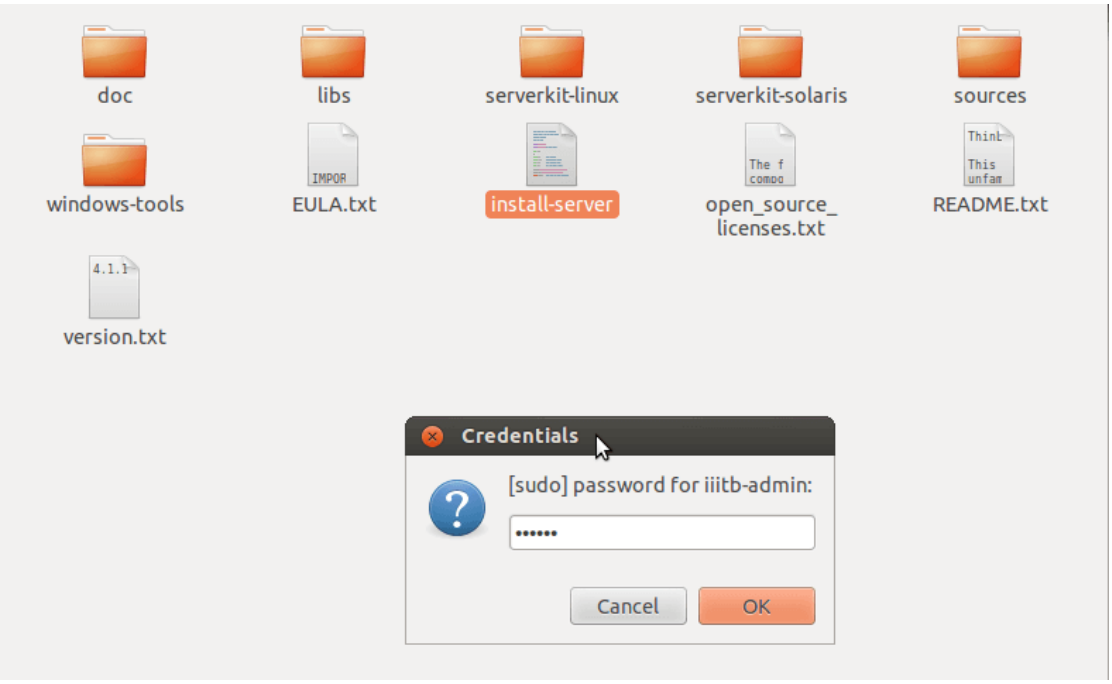


FIGURE 1.7: ThinLinc Installation step 3

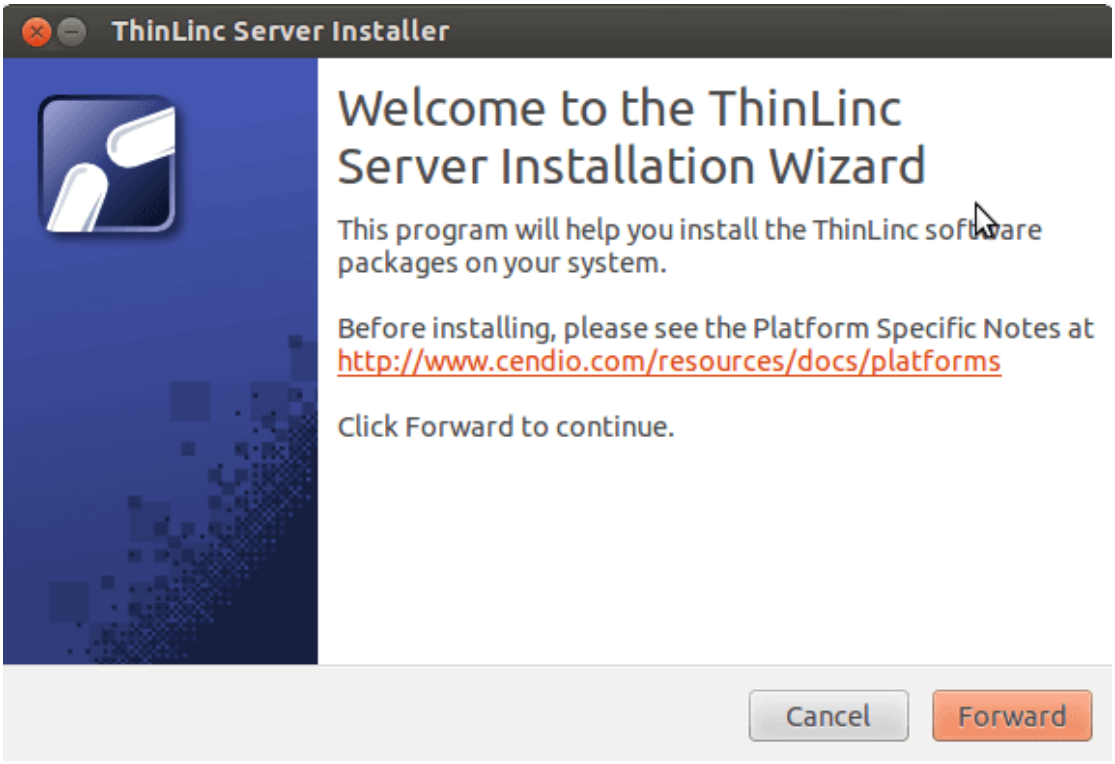


FIGURE 1.8: ThinLinc Installation step 4

The session starts with making a SSH tunneling to the terminal server, which in case if busy provides load balancing feature and connects the client to a less loaded server. one need three piece of information to be able to connect to the terminal server: its address or the domain name, the client specific username and an appropriate authentication. During the first time installation of the server set-up, use the server name and login password from any other machine (using any of the three client setups) to connect to the terminal server just created. By default the server is not administered to provide client specific username and password to remote users and this has to be enabled by the admin. Every client has a unique username and password provided by the terminal server which is important for creating unique sessions for that particular client. The sessions allow keep-alive feature along with mandatory saved sessions. A remote user (client) can access its customized desktop from anywhere in the world and get the benefit of working on a computationally efficient machine without having to confine within the premises of the laboratory. This greatly reduces the expenditure. Until now data visualization needed high configuration machines to do the rendering but with this alternative a single server at an organization can be used concurrently by many remote users (clients) each with their own customized desktop and set of applications, share resources and much more.

### 1.3.5 Troubleshooting

This particular topic can be dealt in many ways. In the official documentation one would find some information under this heading. Actually Cendio AB provides an email list for ThinLinc users where one can post problems encountered and discuss with other users anywhere in

the world and even with those who created this product. This is an official forum for discussion. For simple issues one may follow the link http://www.cendio.com/resources/docs/tag/troubleshoot.html. However, the email list is very useful from our personal experience and we recommend others to actively use it.

## 1.4 Browser Access

The flexibility that ThinLinc offers to the client devices for connecting to their respective terminal servers servers is unparalleled. It is not just a cross-platform application but allows using all features via a web browser and on mobile devices like smartphone and tablet. There might be constraints of installing any program at some places and this is where the browser based clients come to the rescue.

### 1.4.1 HTML 5 Client

The browser based client uses HTML5 features like Websockets and Canvas to establish connections. All HTML5 enabled browsers that support Websockets and Canvas can be used and hence serve as suitable browser based clients. At the time of testing, the latest release was ThinLinc 4.1 and back then stock Android browser did not support Websockets and therefore could not be used as a client for the same reason. Official documentations show that Apple iPads running Safari browsers are fully compatible. They also mention that all following four major browsers can be used as ThinLinc client.

- Internet Explorer
- Mozilla Firefox
- Google Chrome
- Apple Safari

At the time of testing, these were compatible back then and so are supposed to work with current releases too. We have tested it on Chromium and Firefox and it seem to work pretty well. The VNC used for this purpose is noVNC instead of TigerVNC, as we had mentioned before, simply because it is lightweight and suitable for mobile clients.

### 1.4.2 Touch Devices

The browser based clients also support touch based navigation on mobile devices and when this feature is enabled, there would be a couple of extra options on the top-left corner of your desktop window, after you have logged in.


FIGURE 1.9: ThinLinc Web Browser Client Login

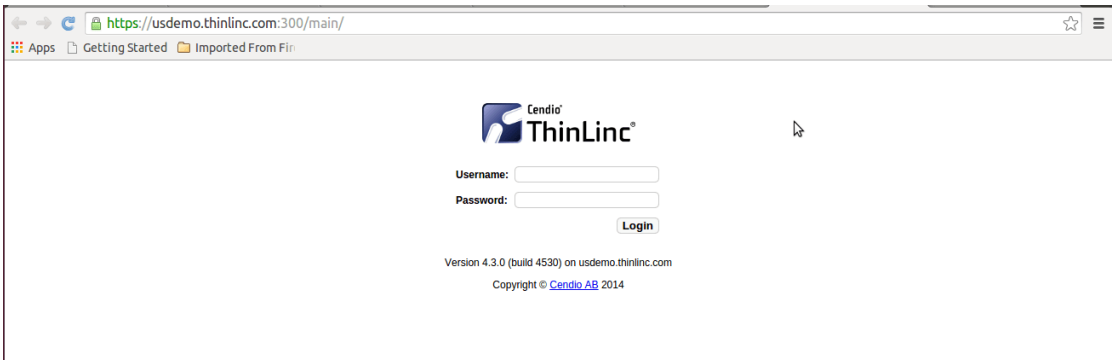


FIGURE 1.10: ThinLinc HTML5 Client

### 1.4.3 Establishing Connection

- Create a DNS for the ThinLinc Server or assign a Public IP Address
- Use this DNS or Public IP Address in the address bar of your browser followed by the port address, which is 300.
- It must look like https://server-ip-address.com:300 or https://yourserver.com:300 (See Fig 1)
- The ThinLinc login window opens up. Fill in the username and password to login. (See Fig 2)

The terminal server, however, does not change in any manner. We tested it on various versions of Ubuntu, from 10.04 LTS to the recent version of 12.04.03 LTS, and have satisfactorily able to deploy it. The version of Thinlinc installed is 4.1. In our setup, Thinlinc was installed on a hp Z400 workstation, which has a high-end graphics card onboard, being used as the server and remotely takes care of computations leveraging GPGPU.

# Chapter 2

# Visualization and Exploration of 3D Range Data

## 2.1 Visualization and Exploration of 3D Range Data

On starting the application, RAVEPC will appear as shown in Figure 2.1. The user interface contains Menu Bar, Display Area, Reduction Parameters Panel, Segmentation Parameters Panel, Mesh Parameters Panel and Tool Bar.

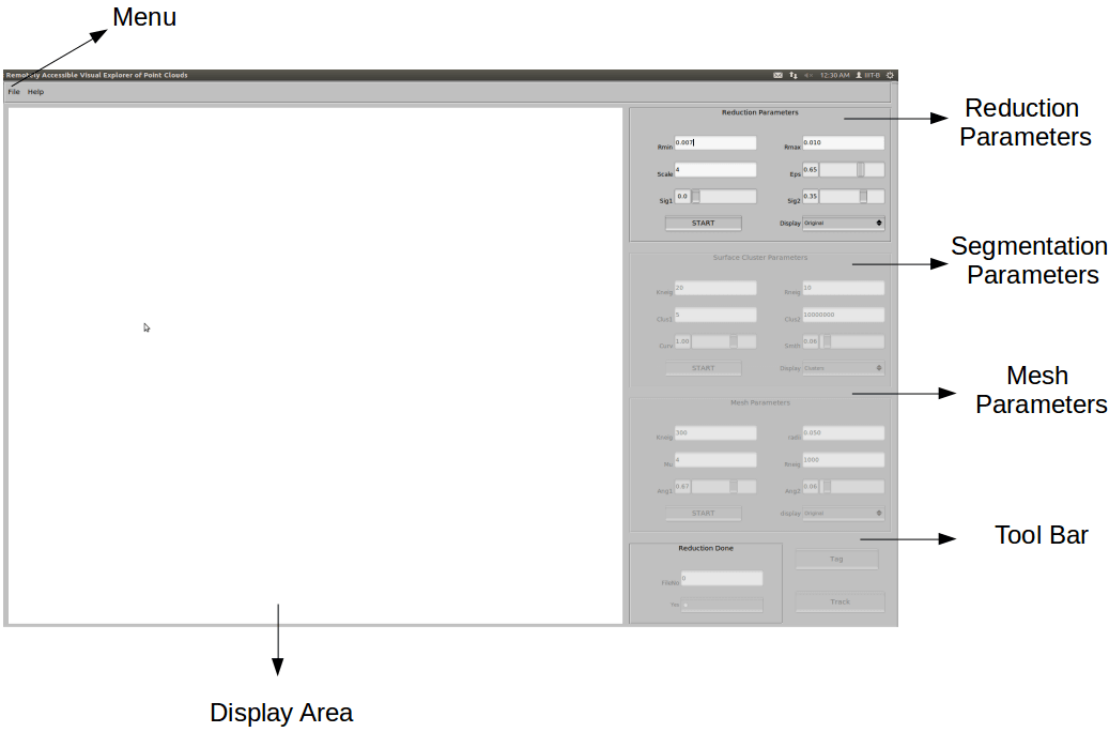


FIGURE 2.1: RAVEPC Interface

### 2.1.1 Menu Bar and Display Area

It Provides two button, one for File and other for Help Menu. File Menu can be used to load the las, ply or pcd file. Only one file can be selected at a time as shown in Figure 2.2. The output will be shown in the display area as shown in Figure 2.7.

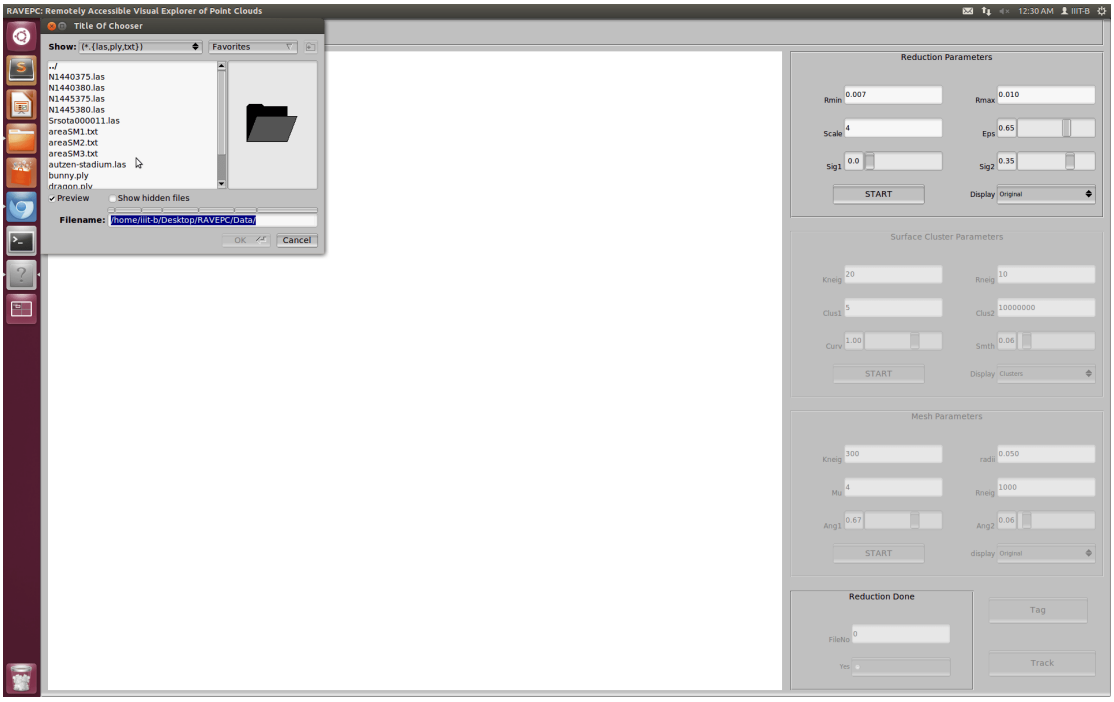


FIGURE 2.2: RAVEPC User Interface

User can analyze the output with the help of various other options like rotation, scaling, etc. Following options are provided:

- ctrl and R or mouse left click for rotation
- mouse right click for translation
- ctrl and + for zoom-in
- ctrl and - for zoom-out
- ctrl+Esc to close the application

## 2.2 Point Reduction

There are several parameters which user has to defined to reduce the point cloud. Figure 2.1 shows the panel where user can set the parameters. There are total six user defined parameters for point reduction.

- Scale Parameter: Here, scale parameter is the radius value for $r-$nearest neighbor search. The is a main crucial parameter which user has to tune to get correct value for a particular

data-set. *Rmin* defines the lower bound for radius and *Rmax* defines the upper bound for the radius. *Scale* defines the number of step between *Rmin* and *Rmax*. On decreasing the radius, more points may be categorized in spherical class while on increasing , more points may be categorized in disc and curve class

- *Eps*: This parameter is used to classify the points into critical, curve and surface classes. It is not a critical parameter. We have used 0.65 for most of data-sets.
- $Sig1$ and $Sig2$: $Sig1$ and $Sig2$ defines $\sigma_{min}$ and $\sigma_{max}$ respectively.$\sigma_{max}$ controls the transition between features like creases and corners. On the other hand, $\sigma_{min}$ differentiate between the creases and non-creases type features on surface.

Once parameters are set for point reduction, press the start button on point reduction panel. It will process the data and display the reduced point cloud. Figure 2.7 and Figure 2.8 show the original and reduced point cloud of the same data-set respectively.

There are various display options are available in the reduced point cloud panel which shows other derivatives of the data. User can view the curvature map, saliency map, seed points. If user does not satisfy with the output of any of these maps, user can change the parameters and re-run the program to get the correct output. Followings display options are provided in the reduced point cloud panel:

- Curvature Map: Curvature value of data as given in the Figure 2.3.
- Curve Seed Points: Shows the curve-type points as given in the Figure 2.5.
- Surface Seed Points: Shows the surface-type points as given in the Figure 2.6.
- Original Point Cloud: Shown in the Figure 2.7
- Reduced Point Cloud: Shown in the Figure 2.8

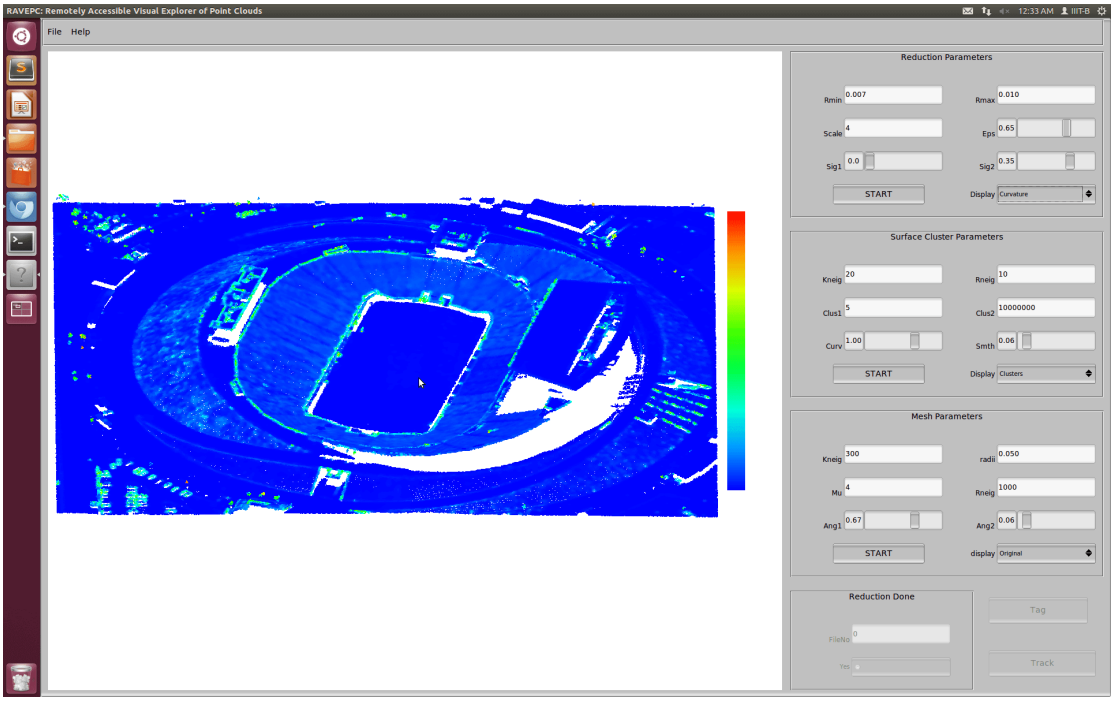


FIGURE 2.3: Curvature Map, Blue indicates low value and red indicates high value

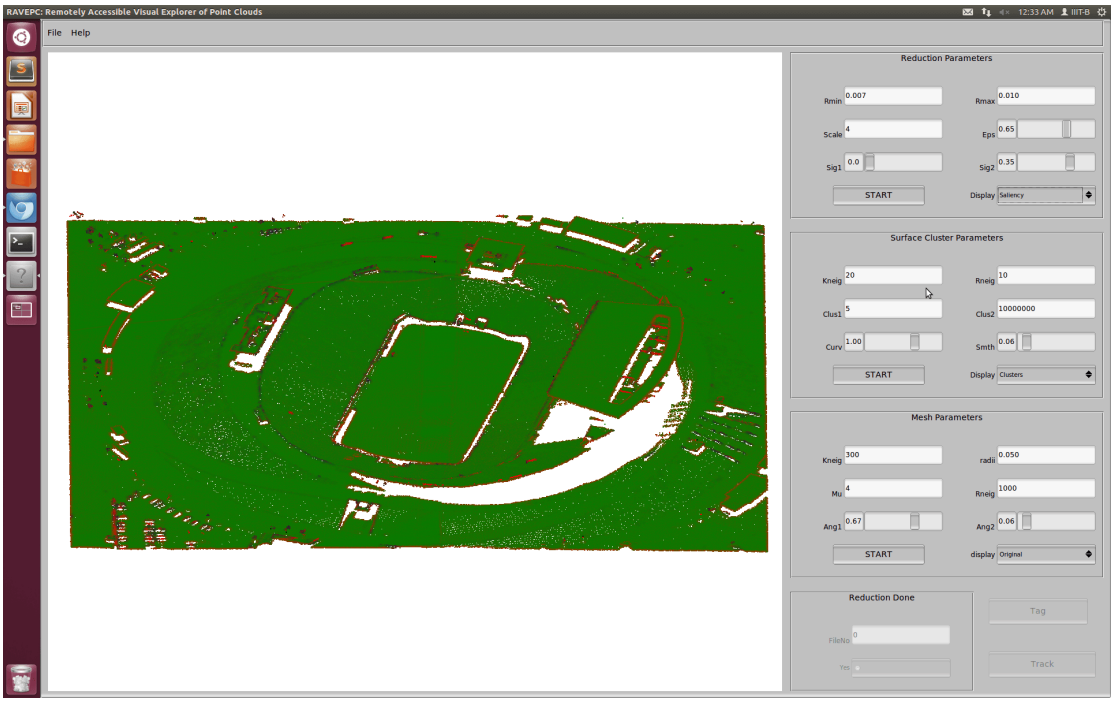


FIGURE 2.4: Saliency Map, red shows curve-type points, green shoes surface-type points and blue shows critical-type points

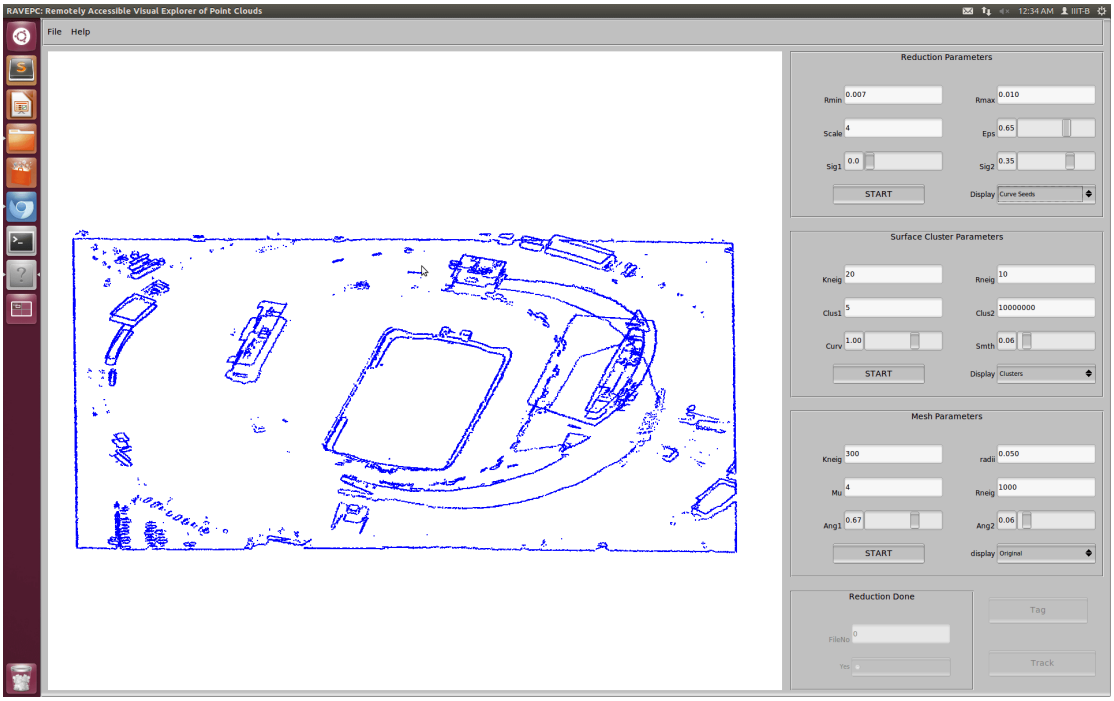


FIGURE 2.5: Curve-type points

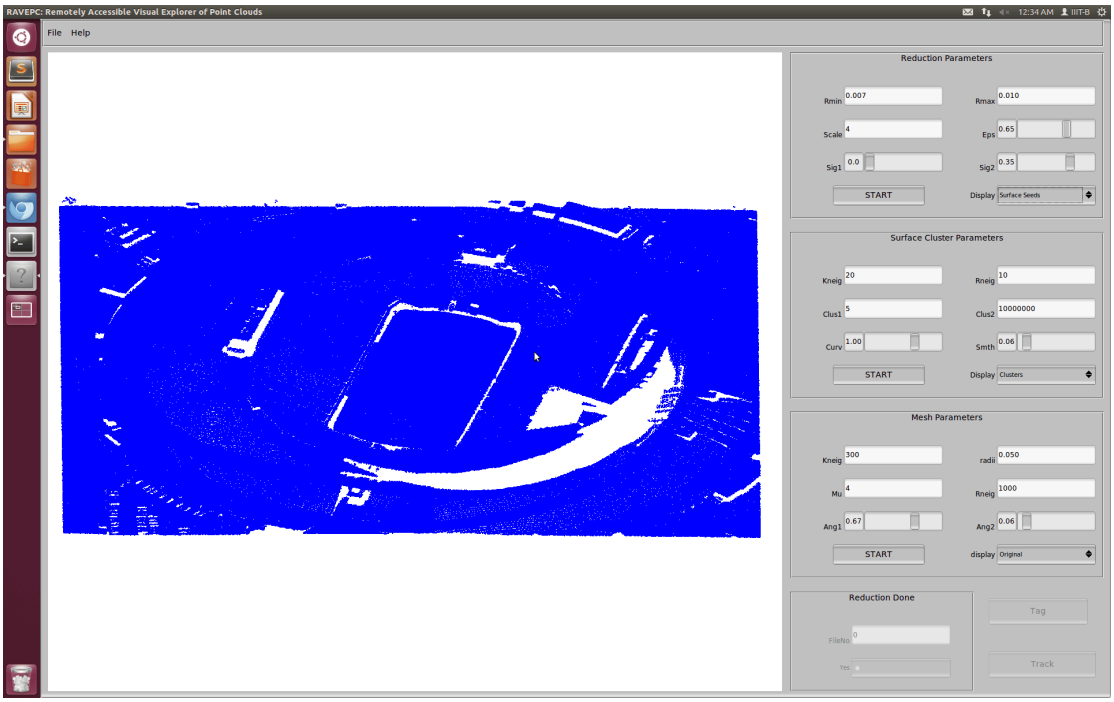


FIGURE 2.6: Surface-type Points

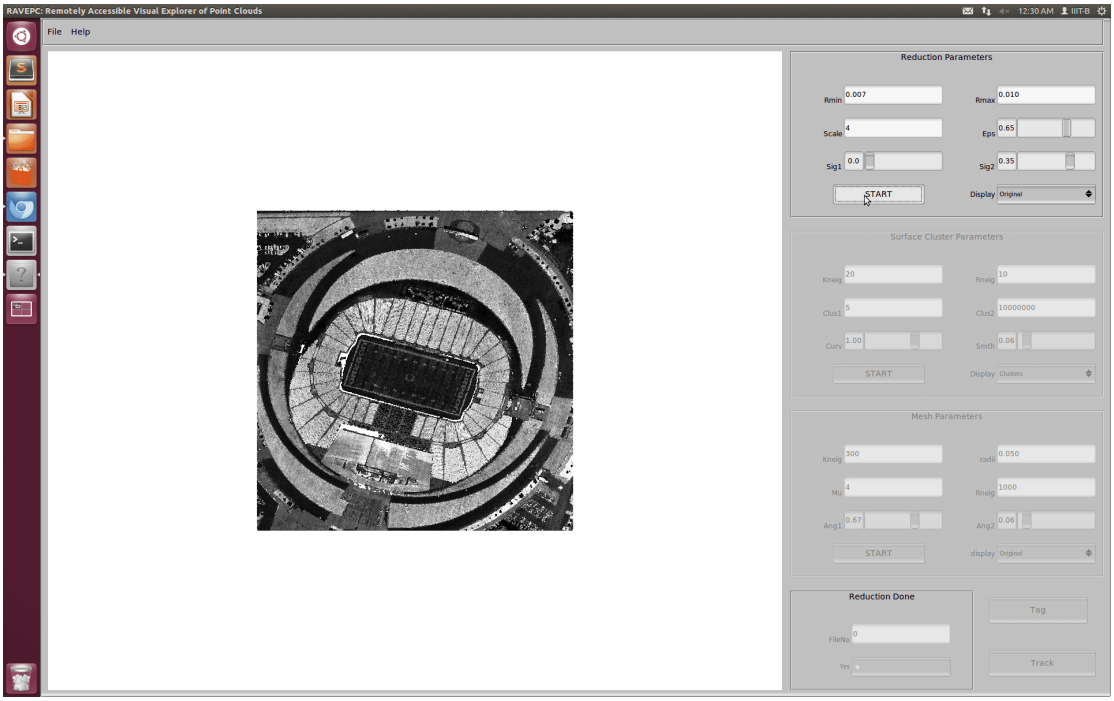


FIGURE 2.7: Original Point Cloud of Autzen Stadium

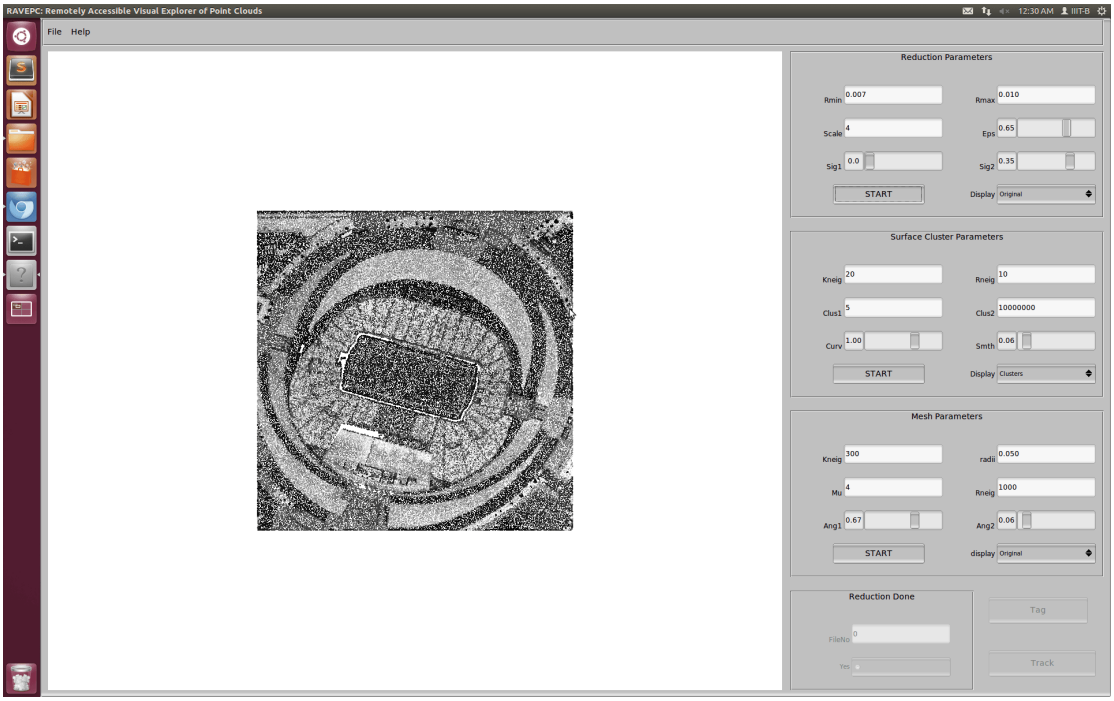


FIGURE 2.8: Reduced Point Cloud of Autzen Stadium

## 2.3 Segmentation

There is a panel for segmentation of reduced point cloud as given in the Figure 2.1. Region growing algorithm has been used for segmentation of the reduced point cloud. *curvth* and *smth*

are critical parameters which user has to tune to get proper segmentation of the point cloud. Segmentation may take some time to process based on tha data size. Colors are randomly assigned to the clusters as shown in the Figure 2.9.

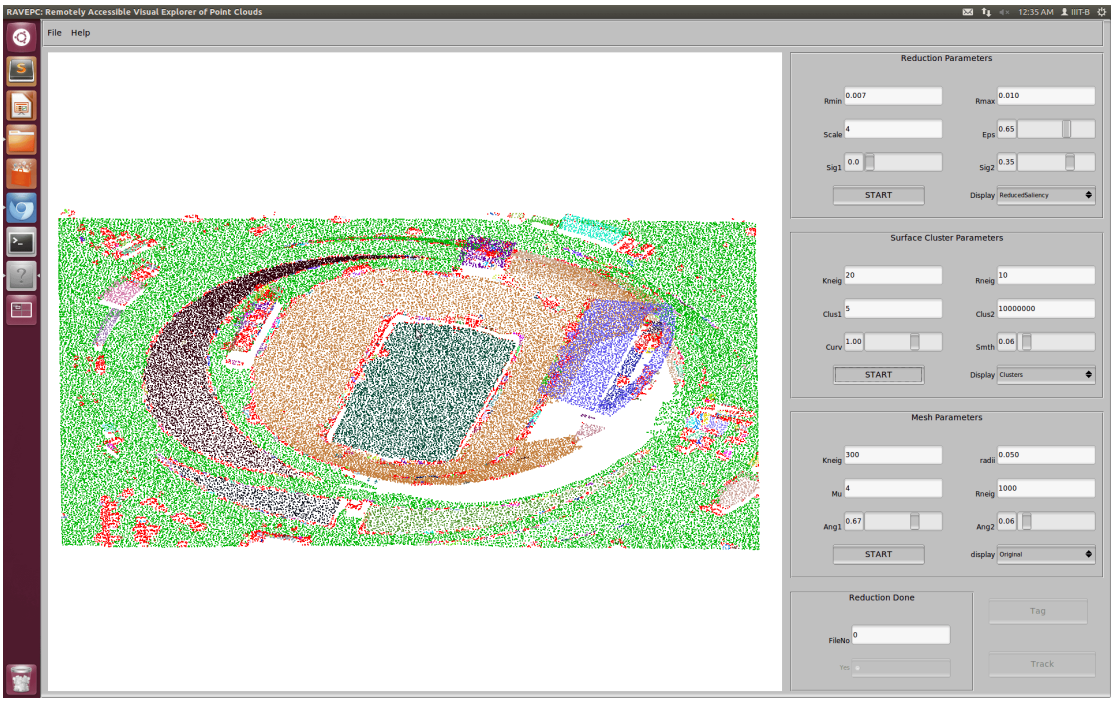


FIGURE 2.9: Segmentation of Autzen Stadium. Colors are randomly assigned to different clusters

## 2.4 Mesh

Third Panel in the Right area of the application is for mesh parameters as given in the Figure 2.1. Triangle primitive is used to construct the mesh. There are several user-defined parameters which user can fine tune to get the correct output if he/she does not satisfy with output obtained with default parameter settings. Following parameters are used for construction of mesh:

- kenig : Number of neighbors to be considered for normal estimation
- radii : Maximum edge length of any triangle in the mesh
- Mu : Neighbors can be searched within this range only
- Rneig :Maximum number of neighbors to be considered to construct the mesh
- Ang1 : Minimum angle between any two side of the triangle
- Ang2: Maximum angle between any two side of the triangle

In the Mesh parameter panel, there is an option to view meshes constructed by original and reduced point cloud as shown in the Figure 2.10 and Figure 2.11

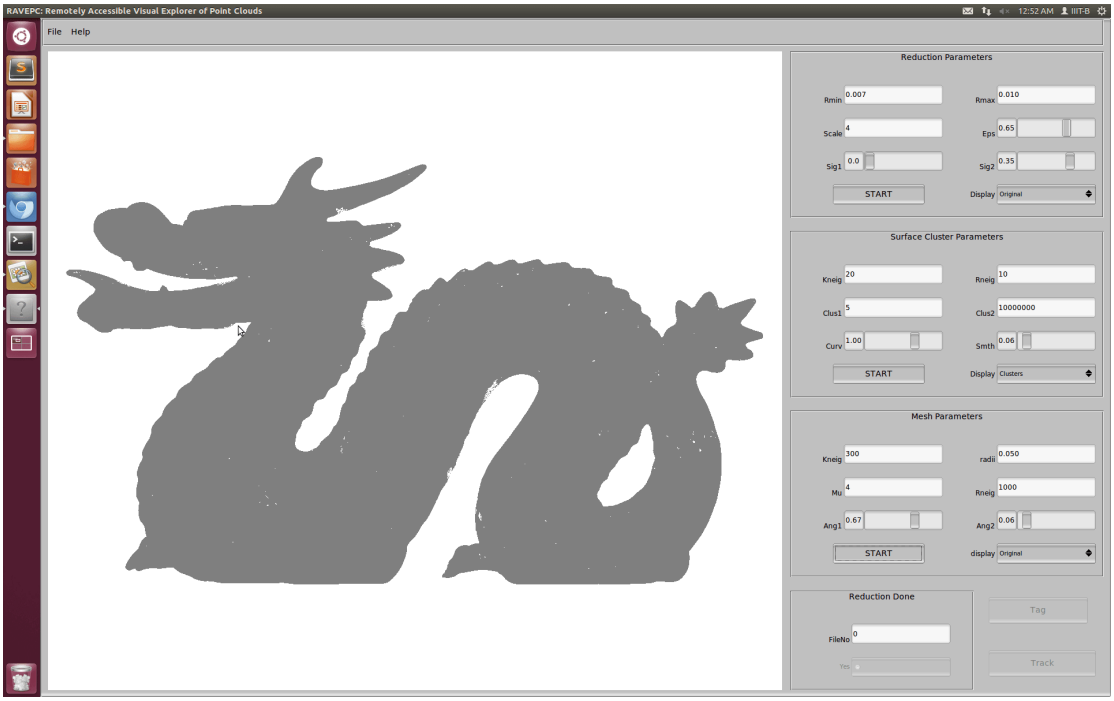


FIGURE 2.10: Mesh constructed using original point cloud

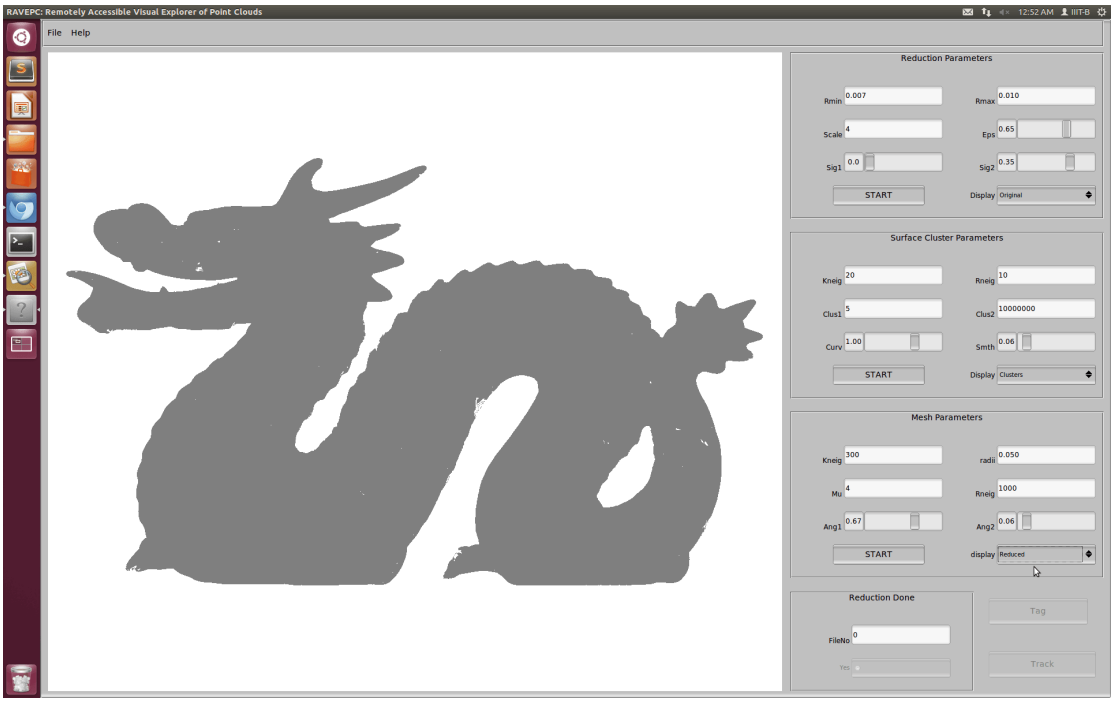


FIGURE 2.11: Mesh constructed using reduced point cloud

Tool panel in the Figure 2.1 provides the options for feature tracking in the time-varying data which is discussed in details in the chapter 3