Technical report
Heightmap Analyser

Benjamin V. Hugo **Brandon J. Talbot**
Supervisor: Professor Michelle Kuttle
Department of Computer Science, University of Cape Town

Abstract

Data visualization is one of the corner stones in the analysis of scientific experiments. Analyzing the differences between two datasets may not be very complicated for one dimensional problems. However, when these problems increase in number of dimensions, analysis becomes much harder. The task at hand was to plan and implement a tool, to aid the visualization of two dimensional datasets, as well as an intuitive way to compare two or more of such datasets. Amongst other techniques, both path analysis and side-by-side methods (such as comparing heat maps, difference maps, contour plots and a 3D fly through) were examined. The proposed design, as well as the implementation of such a tool will be discussed in this report.

1 Introduction

The visualization of data drawn form higher dimensional problems for the purposes of comparison is fundamental to the analysis of large datasets in science and engineering. Consider, as example, an experiment is run to determine how the free energy of an atom varies, if the dihedral angles between the atoms in a molecule are changed. The experiment is repeated several times under different conditions, each time yielding different results. The output of each trial takes the form of a two dimensional grid of values, which is infeasible to analyse as raw data.

The problem now can be summarized as constructing a visualization of the data that can fulfill the following needs (both qualitative and quantitative):

- Provide a high level overview between the differences in the data.
- Place an emphasis on the areas where the differences in the data is maximal, as well as minimal.
- Provide an easy way to extract exact quantitative information from the data.

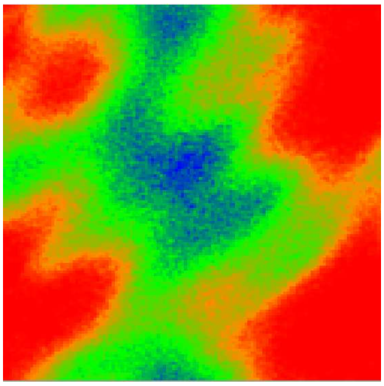
2 Design Methodology

The basic principle behind visualizing two dimensional data is to consider the data points as heights of the form $(x, y) \rightarrow h$ where $x, y \in \mathbb{R}$ are equally spaced discrete points constituting the two axis of the dataset. $h \in \mathbb{R}$ is the height of the curve at a point x, y.

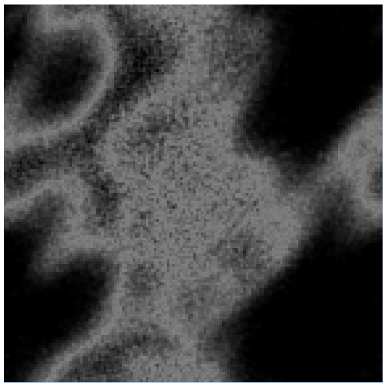
To cover the aspects mentioned in the introduction several different visualization approaches were considered. Each has its strengths and weaknesses as can be outlined as follows:

- Heat maps. Are useful to provide a qualitative overview of the dataset. It is essentially a projected image of the heights onto a two dimensional surface, where a color gradient indicates the differences in height (interpolated from deep blue for the lowest points to bright red for the highest peeks). However, they are not that useful for extracting qualitative information.
- Contour plots. Provides some form of quantitative information as well as qualitative, but it still cannot be used to obtain exact quantitative data.
- Difference maps. Useful only to highlight areas where the differences between the heightmaps are greatest.
- Three dimensional rendering. Coupled with rotation and zoom this approach provides a very effective overview of the differences between the heightmaps, but, as standalone this method does not provide a very attractive way of finding quantitative information. It is, however limited to displaying only one or 2 heightmaps at a time (otherwise it becomes increasingly difficult for the user to distinguish between the different heightmaps.
- Two dimensional line graph. Can be used to gain quantitative information along, for example some line on the dataset. It, by itself cannot provide any qualitative information.

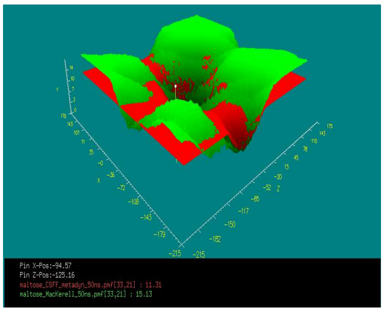
The challenge was to combine these methods in a useful way as to provide the end user with a system that



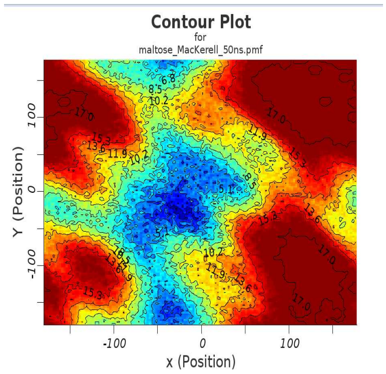
(a) Example of a heat map



(b) Example of a difference map



(c) Example of a 3D visualisation



(d) Example of a contour plot

Figure 1: Visualisation Techniques

could provide both an overview and exact quantitative data. See 1.

3 Planning & Design

3.1 Initial planning

The initial software engineering (done in accordance to the approaches outlined by the Software Development Life Cycle methodology), including requirements capturing, analysis, and design has been outlined in the system specification which supplements this report.

3.2 Overview of the system design

All the visualisation techniques outlined above were incorporated into the requirements for the tool. The following design decisions were made:

- In order for the user to extract qualitative information about the datasets a list of heatmaps is kept for side-by-side comparison. Furthermore, the user can select one of the heatmaps to perform path analysis. Several tools were developed for this purpose: a horisontal, vertical, left diagonal, right diagonal and free-hand drawing tool. Tbe heights along the selected path.

- It was also decided to include a way for the user to examine two datasets more closely, as to draw a comparison between them. To achieve this the user can select, from the list of heatmaps the two datasets to be compared. A difference map, contour plot and a 3D rendering of the two datasets are then constructed. The difference map indicates the areas of greatest difference, while the contour plot provides a side-by-side view of the two datasets. The 3D rendering makes it simple to examine the differences between the two datasets, by providing the means to manipulate (rotation, zoom and scaling) them. It was furthermore decided to add labeled axis, as well as using the cursor (the height is computed at the point the cursor intersects the X-Z axis) in order to retrieve quantitative data from the visualisation.

- Statistics on the datasets are computed and is readily avaiable. This includes computing the root-mean-square-deviation between the datasets, displaying the width, height, minima and maxima of the datasets, displaying the difference at the cursor position as numerical data when viewing the difference map. The data can be exported to comma seperated text and the visualisations can be saved as images.

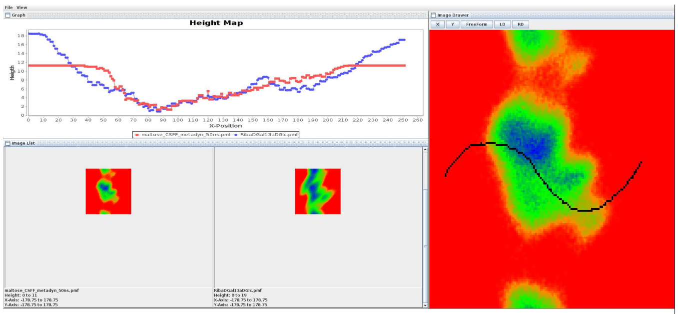


Figure 2: Tool in action

3.3 Execution of planning

It was decided to split the project work into two roles (since implementation and software engineering count 50% each):

1. Brandon Talbot was the lead developer. He was tasked with writing the path analysis algorithm, generating statistics, integrating the difference maps and contour plots, and the Graphical User Interface layout.

2. Benjamin Hugo was the lead systems analyst. His primary role was to analyse requirements, designing the overall layout for the system and writing both the systems specification and this report. He furthermore implemented the 3D fly-through using $OpenGL^{TM}$ as his contribution to the development of the system.

4 Findings and post-implementation review

The tool, as described in the report and its constraints, as documented in the specification document has been fully implemented and tested (details of this can be found in the specification document). It was found to meet the original requirements, as well as the requirements (and changes) that were proposed after the prototype demonstration. The tool succeeds in its primary objective: to analyse the differences between datasets, both in qualitative and quantitative terms (as required by scientific experiments). See 2. The 3D rendering, contour plot and difference maps seen in 1 are also generated by the tool.

Some changes were made to the initial design, of which the most significant was to move the tool to an Open Source, cross-platform environment using $Java^{TM}$ as development platform. Furthermore several updates to the overall architecture of the tool has been made. These changes are described in detail in the updated specification document.

Future extensions to the tool can include providing more sophisticated export functionality (including exporting to spreadsheets and pdf format), and to add functionality to ensure that the program will work for heightmaps of different sizes.

Department of Computer Science
CSC3003S - 2012

USER REQUIREMENTS & SYSTEMS SPECIFICATION (POST-IMPLEMENTATION REVISION)

HEIGHTMAP ANALYSER

TEAM MEMBERS

Student number: HGXBEN001	Benjamin V. Hugo
Student number: TLBBRA002	Brandon J. Talbot

Declaration

1. We know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. This Specification Document is our own work.

3. We have not allowed, and will not allow, anyone to copy our work with the intention of passing it off as their own work.

Team Member Name...........................Signature:.........................Date ……./...../...........

Team Member Name...........................Signature:.........................Date ……./...../...........

Table of Contents

1. Introduction

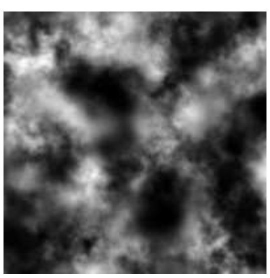
1.1 Overview of the problem

The visualisation of data, gathered during scientific experiments, is of great importance when comparisons are made between different sets of data. This task, however, is seldom straightforward due to the amount of input data that has to be visualised, and the fact that data gathered in higher dimensions has to projected down to a two dimensional representation which can be rendered by a computer.

The aim of the project is to construct a tool that will enable the client, Dr. Michelle Kuttle, to analyse raw data from her chemical experiments with ease. It should facilitate the extraction of quantitative data to aid further experimentation. This will be achieved by visualising the data in an effective manner, as well as highlighting the areas of greatest and least difference between the datasets. A simple side-by-side comparison of the raw data is not adequate, because of the amount of input data received from these experiments, as well as the constraint that both quantitative and qualitative information has to be extracted simultaneously.

The input data described above takes the form of a two dimensional grid of values (i.e. a collection of points in three dimensional space). We will make use of a well known visualisation technique, known as a heightmap (although other representations exist, for example contour plots, they will not be considered). These heightmaps can be represented in both two dimensional space, as well as three dimensional space.

In two space an image can be composed, using different tones of color, to indicate the differences in height along the dataset. In three space the data can be visualised using a curved surface, which in turn can be rendered using appropriate, well-established, software libraries.



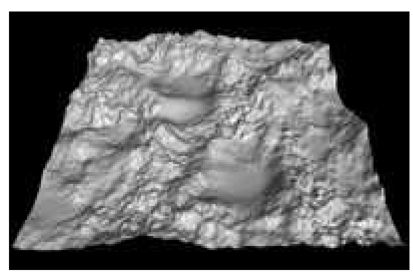


Figure 1*: On the left: a 2D heightmap using different shades of gray as a visual queue. On the right: a 3D rendering of the same heightmap as the one seen on the left [1].*

1.2 Statement of scope

Our main objective is to construct an interactive way of displaying and analising the differences between two or more of the heightmaps, as described above. The client has requested the following functionality be included:

- Given two or more height map inputs. Generate a visual image representation for each of the heightmaps, in order to facilitate qualitative information.
- Highlight areas of greatest and least difference to aid identification of qualitative information. Identify movement of minima and maxima on the heightmaps.

- The ability to analise the two height maps along an arbitrary path (an arbitrary subset of the data). This path will determined using cursor data from the mouse. The differences between the height maps will be best visualised using graphs (in order to extract quantitative data using motion). Other, more sophisticated techniques, may be included at the discretion of the software design team.
- The ability to export output from the program to various other formats. The possible formats may include images, html, pdf or spreadsheets.

1.3 Risk model

There are very few risks involved with this project as it is set in a non-production environment (we do not have to consider outsourced components, end-user acceptance, financing, training in this regard and legal matters). The other risks are indicated in the table below.

Risk to project	Impact	Likelihood	Mitigation
Personal equipment failure	High	Low	Daily backups on local work. Repository system (SVN provided by Google Code)
Time constraints due to other courses: CSC3020H, CSC3022H	High	Medium	Restructure and drop components with lower importance. There may or may not be time to do the 3D visualization. Abandoning this component will not be considered catastrophic project failure.
Team organization problems including unforeseen leaves of absence.	High	Low	Renegotiate project terms with academic supervisors.
Lack of skill or know how on certain aspects	Medium	Low	Consult academic support and peers.

2. Requirements capture

2.1 Functional Requirements

During the discussion with Dr. Michelle several functional requirements arose. These requirements will be detailed in narrations first and then condensed as a use case diagram.

Use case	**ID**	**Importance level**
Add experiment data	1	High
Brief description	In order to compare several different data sets we need to add a simple process for adding two or more data sets.	
Actors	Client	
Stakeholders	N/A	
Related Use Cases	None	
Preconditions	Valid data files must already exist. These files will take the form of space	

	separated text.		
Postconditions	Data is in a parsed state, heightmap dimensions and extremes are calculated and ready to be visualized and / or analysed.		
Guarantees	None		
Flow of Events	**#**	**Action**	**System Response**
	1	Initiates add wizard.	Brings up a filtered browser.
	2	Finds and selects file.	i. Loads file into a heightmap data structure. ii. Adds the dataset to the list of available datasets.
Alternate flows	1. If the input data contains anything but a two dimensional grid of values we will discard it as a corrupt file. The user may specify the number of header lines to skip. Empty lines will be ignored.		
Business Rules	The input data is immutable. No changes shall be made to any part of it.		
Frequency of Occurrence	Very frequent, at least each time the client uses the tool.		

Use case	**ID**	**Importance level**	
View multiple datasets simultaneously	2	High	
Brief description	Viewing different datasets side by side.		
Actors	Client		
Stakeholders	N/A		
Related Use Cases	Obtains the dataset size of the biggest loaded heightmap. Obtains the maximum and minimum value of the current and collection of data. Exportation of generated images.		
Preconditions	One or more datasets has already been loaded by this stage.		
Postconditions	An image of the heightmap data is generated and displayed in grid format.		
Guarantees	An image will be visible after completion of event flow.		
Flow of Events	**#**	**Action**	**System Response**
	1	Views dataset collection	i. Opens a grid based display, scaled to fit multiple heightmaps into viewport. ii. Generates an image for each dataset, using the minimum and maximum values of all loaded dataset points to calculate visualisation colours. iii. Adds each visualisation to the grid display. iv. <<extends>> User can choose to save visualisations.
Alternate flows	1. If the input datasets are not of the same size then the visualisations will be scaled to the same size as the visualisation of the biggest dataset.		
Business Rules	None.		
Frequency of	Very frequent. Entire tool is based around these visualisations.		

Occurrence

Use case	**ID**	**Importance level**
Path height evaluation	3	High

Brief description Visualise the difference in height along a path on several of the datasets simultaneously.

Actors Client

Stakeholders N/A

Related Use Cases Obtains the dataset size of the biggest loaded heightmap.
Obtains the maximum and minimum value of all the datasets.
Export graphs to other formats.

Preconditions One or more datasets has already been loaded by this stage.

Postconditions A graph of the change in height along the selected path will be created.
A rate of change in height graph will be created.

Guarantees Both graphs will be created by the end of event flow.

Flow of Events

#	**Action**	**System Response**
1	Selects one or more datasets to compare	
2	Chooses a tool to use for drawing a path (hand drawn line, straight line or circle).	Draws the path on the visualisation according to mouse input.
3.	(As the user moves the mouse along one of the visualisations)	i. Renders a graph of the height along the path on each of the datasets. ii. A graph for the tempo of change will be included (per dataset path). iii. The different dataset paths will be rendered using different colours. iv. The graphs will be scaled according to the minimum and maximum height values of the collection of datasets. v. <<extends>> User can choose to export graph to several other formats.

Alternate flows 1. If the input datasets are not of the same size then plotting is only done with respect to the size of each of the individual datasets. In other words if the path along dataset B is beyond the bounds of dataset A then the graph for dataset A will not be plotted (or only a subset of the path along dataset A will be plotted at most).

Business Rules The generated heightmap data and properties are immutable and will
not

be permanently modified by this process

Frequency of Very frequent. Main use of this tool.

Occurrence

Use case	**ID**	**Importance level**	
Analise difference Map and contour plots (side-by-side analysis).	4	Medium	
Brief description	Calculates a difference map for two datasets		
Actors	Client		
Stakeholders	N/A		
Related Use Cases	Obtains the dataset size of the larger of the two datasets Obtains the maximum and minimum value of both datasets		
Preconditions	Two or more datasets has already been loaded by this stage.		
Postconditions	An image of the difference map has been calculated		
Guarantees	An image will be visible after completion of event flow.		
Flow of Events	**#**	**Action**	**System Response**
	1	Selects two datasets	
	2	Initiates difference map generation	i. A difference map is generated ii. Image is rendered to screen. iii. <<extends>> User can export difference map visualisations to image file format.
Alternate flows	1. If the input datasets are not of the same size then the visualisation will be the size of the larger dataset. The difference map will be visualised for the overlapping area, while the non-overlapping sections will be left as undefined whitespace.		
Business Rules	None.		
Frequency of Occurrence	Less frequent.		

Use case	**ID**	**Importance level**	
Obtain size of dataset.	5	High	
Brief description	Obtaining dataset size		
Actors	Client		
Stakeholders	N/A		
Related Use Cases	This use case is included in multiple other use cases.		
Preconditions	Dataset must be fully loaded		
Postconditions	No visible state change.		
Guarantees	The size of the dataset (in x,y space) will be returned		
Flow of Events	**#**	**Action**	**System Response**
	1	(another use case provides a loaded dataset as input)	Computes the size of the dataset.
Alternate flows	None		
Business Rules	The input data is immutable. No changes shall be made to any part of it.		
Frequency of Occurrence	Very frequent. Most actions require this.		

Use case	ID	Importance level
Obtain maxima and minima for a dataset.	6	High

Brief description	Obtaining all maxima and minima points on the dataset
Actors	Client
Stakeholders	N/A
Related Use Cases	None
Preconditions	Dataset must be fully loaded
Postconditions	No visible state change.
Guarantees	All maxima and minima points will be returned

Flow of Events	**#**	**Action**	**System Response**
	1	(another use case)	Lists maxima and minima points

Alternate flows	None
Business Rules	The input data is immutable. No changes shall be made to any part of it.
Frequency of Occurrence	Very frequent. Most actions require this.

Use case	ID	Importance level
Export generated output	7	Low

Brief description	Saving heightmap, difference map, graphs and 3D rendering to image or spreadsheet format.
Actors	Client
Stakeholders	N/A
Related Use Cases	This use case extends the following use cases: 2, 3, 4 and 8.
Preconditions	Datasets must be fully loaded and depending on the image that has to be saved either one of use cases 2, 3 4 or 8 completed.
Postconditions	Output files will have been generated.
Guarantees	Output will contain exact copies of visible results.

Flow of Events	**#**	**Action**	**System Response**
	1	User selects output that has to be saved	Renders output to image format
	2	User specifies output format	Saves rendered image to requested format.

Alternate flows	None
Business Rules	No part of the displayed contents shall be modified during this action.
Frequency of Occurrence	Moderate. User may or may not export output per session.

Use case	ID	Importance level
Generate 3D rendering of heightmaps	8	Lowest

Brief description	Renders the heightmap in as a 3D image with a movable view frustum.
Actors	Client
Stakeholders	N/A
Related Use Cases	Rendered image can be exported (use case 7).

Preconditions	One or more datasets must be loaded by this stage.
Postconditions	A 3D view of the heightmap is visible
Guarantees	A subset of the heightmap(s) are visible (depending on the size of the dataset(s)).

Flow of Events

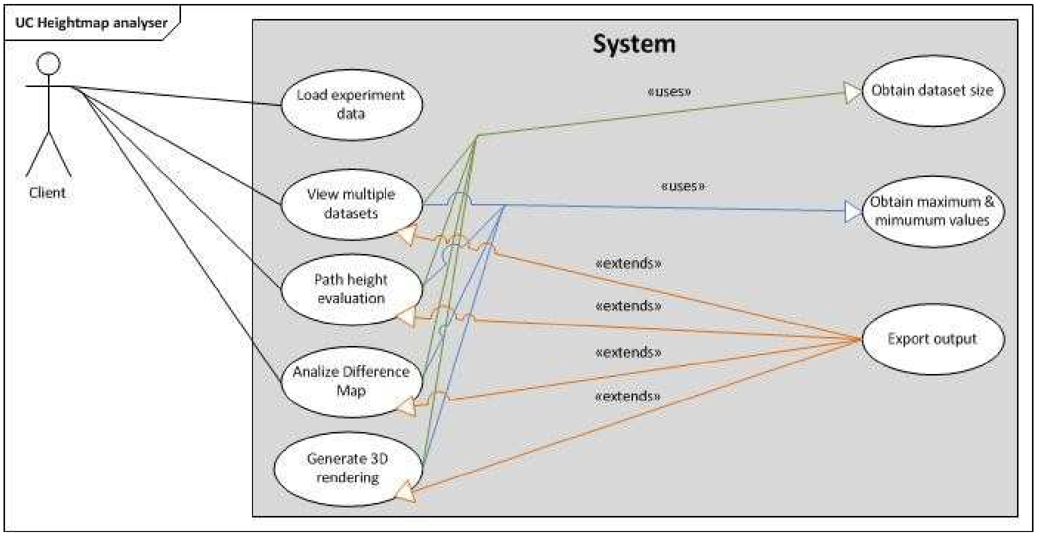
#	Action	System Response
1	User selects a maximum of 2 datasets.	
2	Initiates view	i. Sets up a 3D environment ii. Renders first and second heightmaps, each using shades of a single colour according to the height of the vertices under consideration. iii. Highlights minima and maxima
3	User drags mouse	i. Rotates world by adjusting the view frustum. ii. Renders using step 2.
4	User scrolls using mouse	i. Zooms in/out by adjusting the view frustum. ii. Renders using step 2.
5	Sets the level of detail	i. Increases the number of steps used during interpolation to draw the world. ii. Renders using step 2

Alternate flows	None
Business Rules	No part of the parsed data shall be modified during this process.
Frequency of Occurrence	Moderate. User may or may not use this technique for visualising the data.

2.2 Non-functional requirements

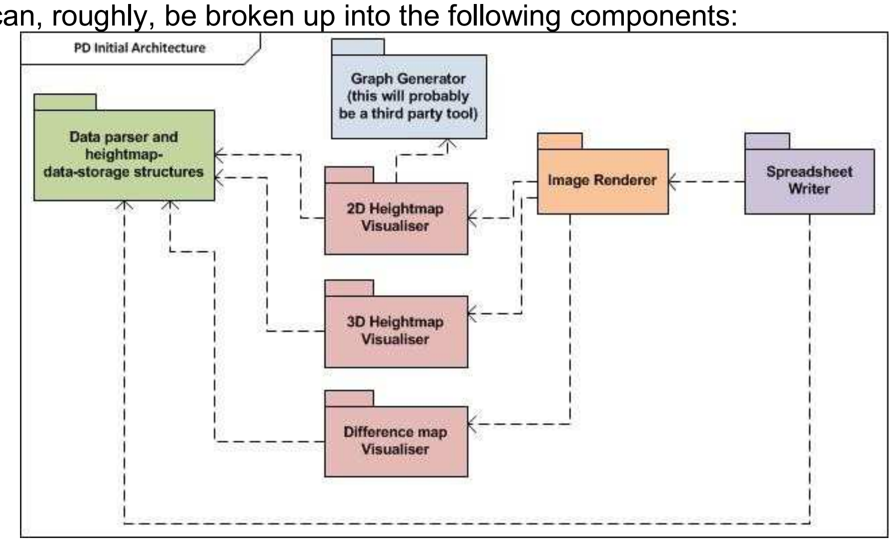
The only non-functional requirements are that the solution must be scalable to accommodate large datasets, as well as efficient to ensure that visualisation of the data can be done in real time.

2.3 Use case diagram



2.4 Initial architecture

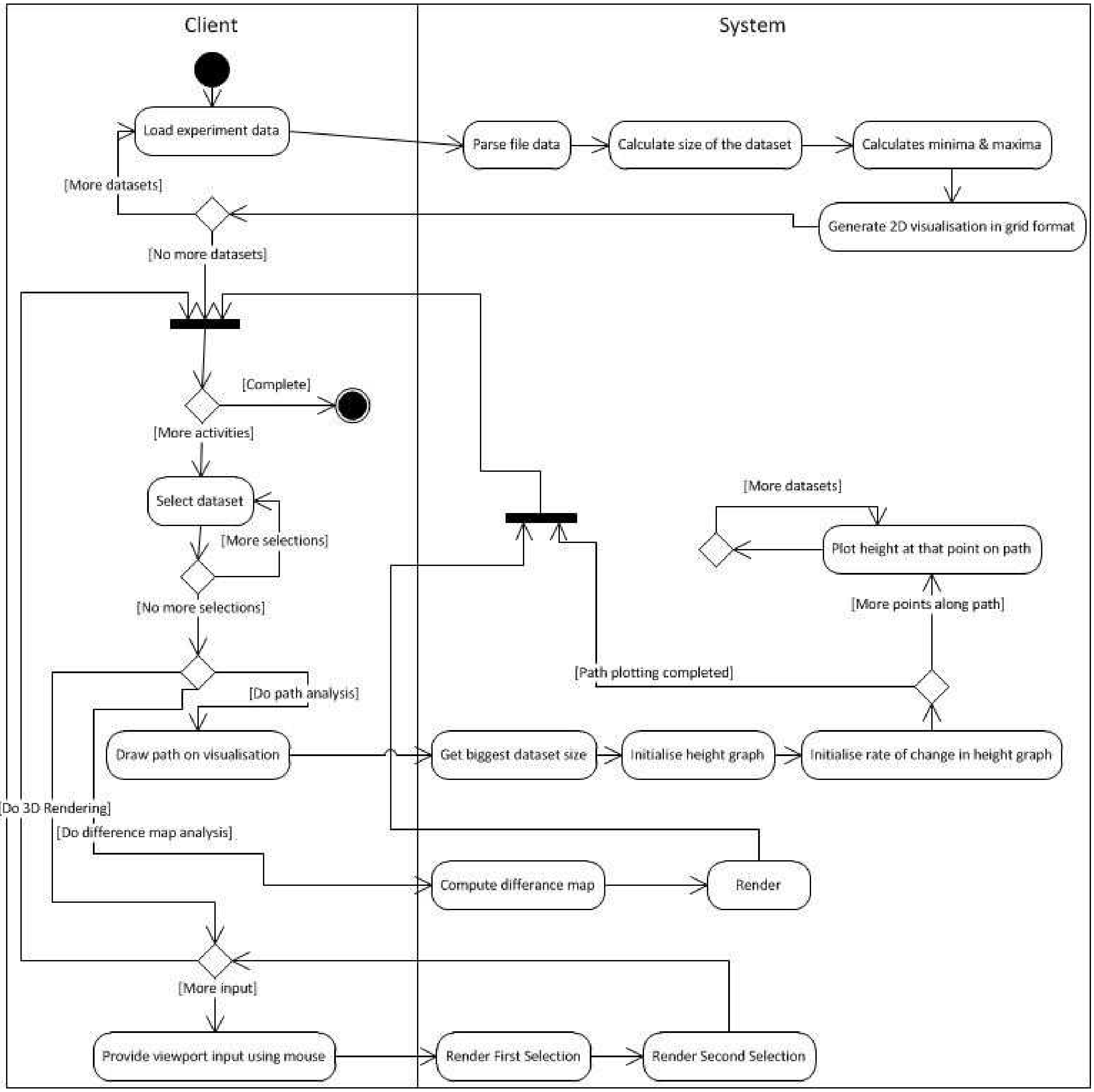
The system can, roughly, be broken up into the following components:



3. Analysis Model

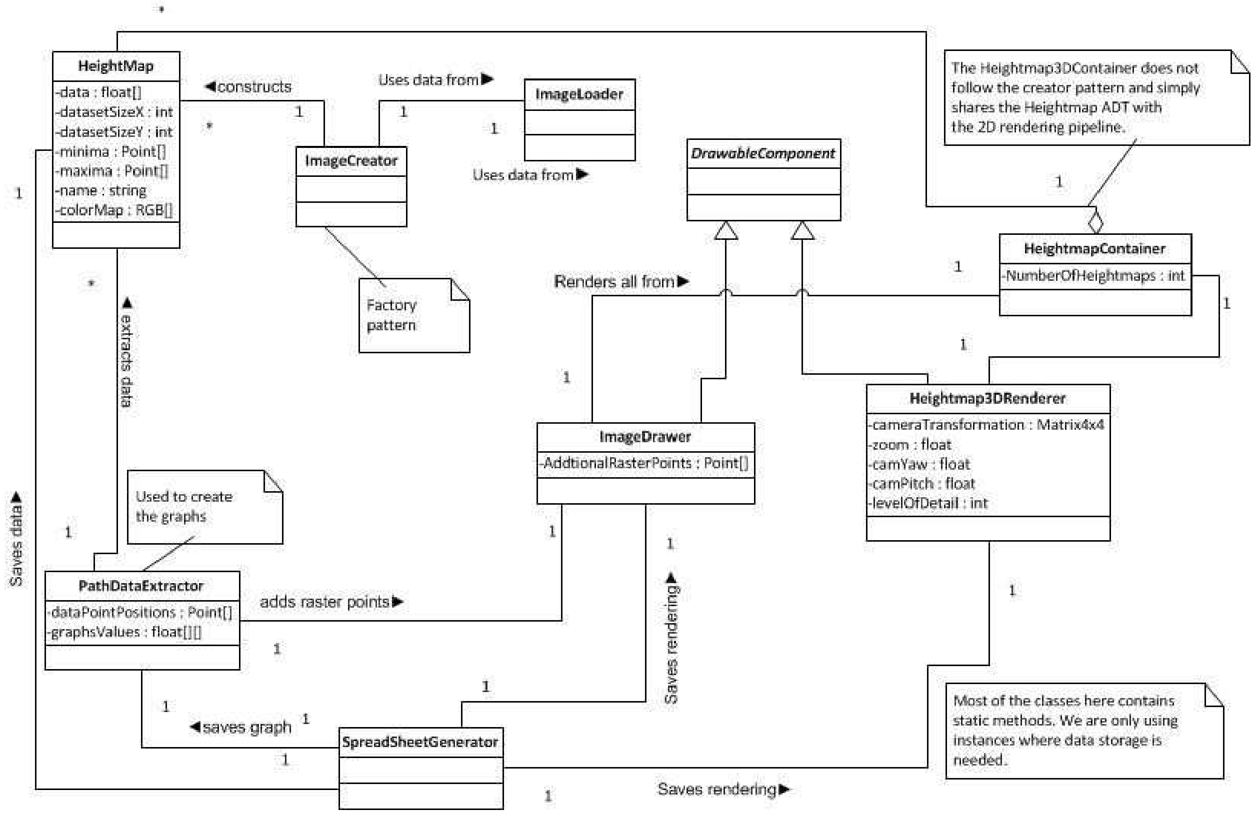
3.1 Activity Diagram

Although activity diagrams are not normally used to model user interaction and activities at a system level (a physical environment such as a business activity is normally modeled with this diagram), we decided to use it to show you the basic communication between the user and the system at a very high level of abstraction.



3.2 Analysis Class Diagram

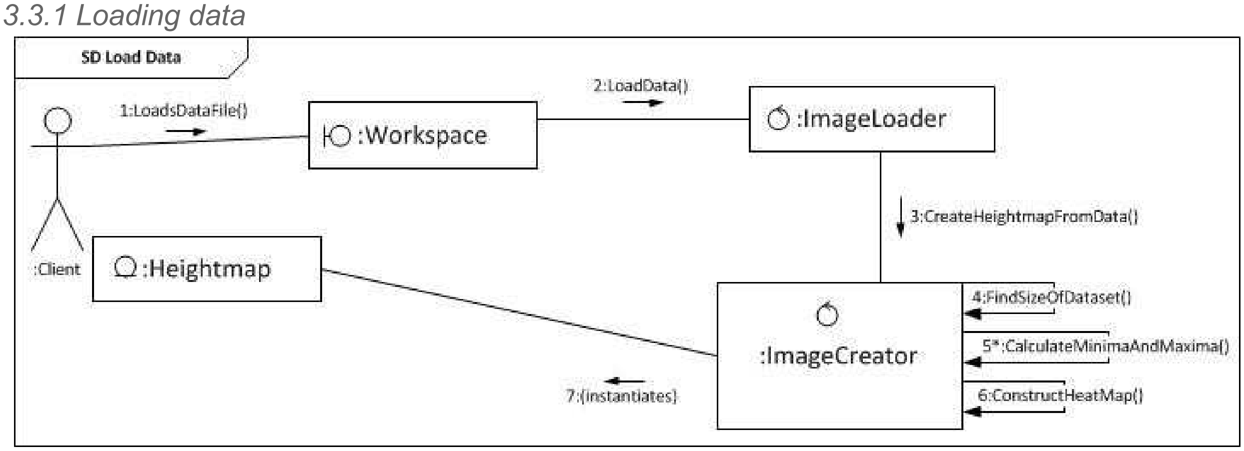
The initial planning for the association between classes are as follows. The methods will be fleshed out during the design phase.



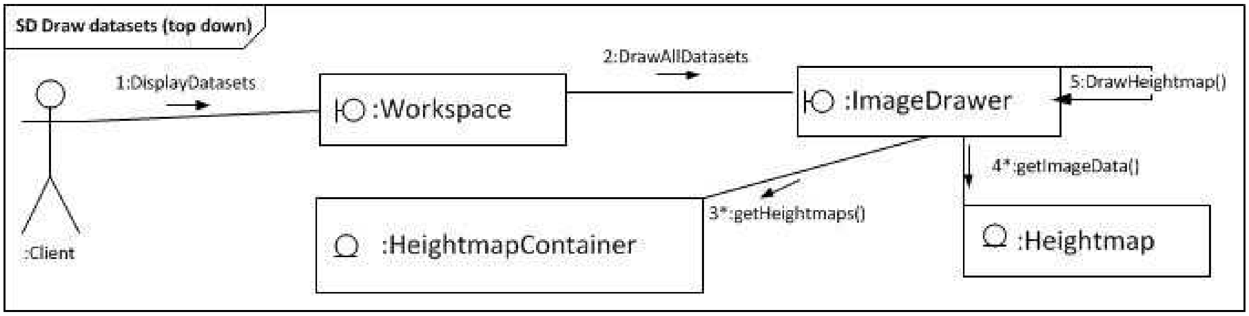
3.3 Communication Diagrams

We will now consider the communication between objects for each of the use cases.

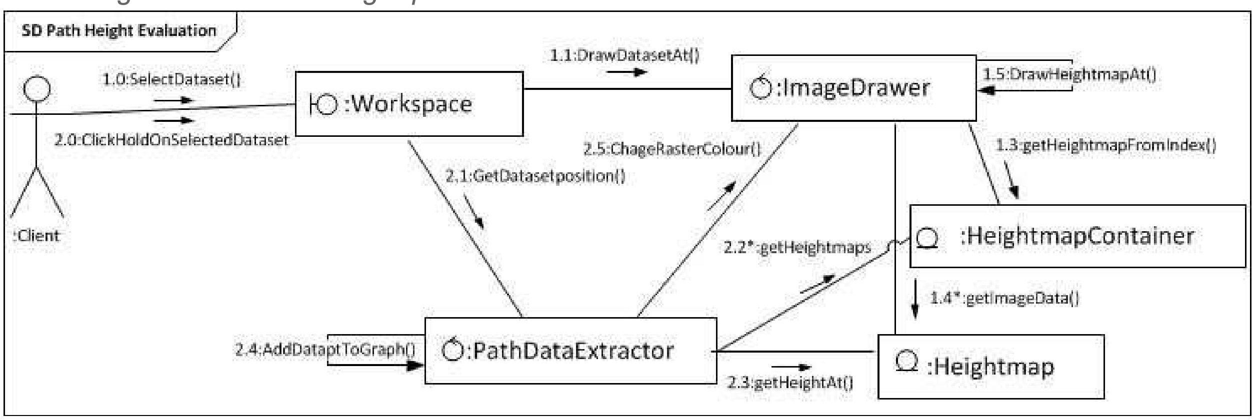
3.3.1 Loading data



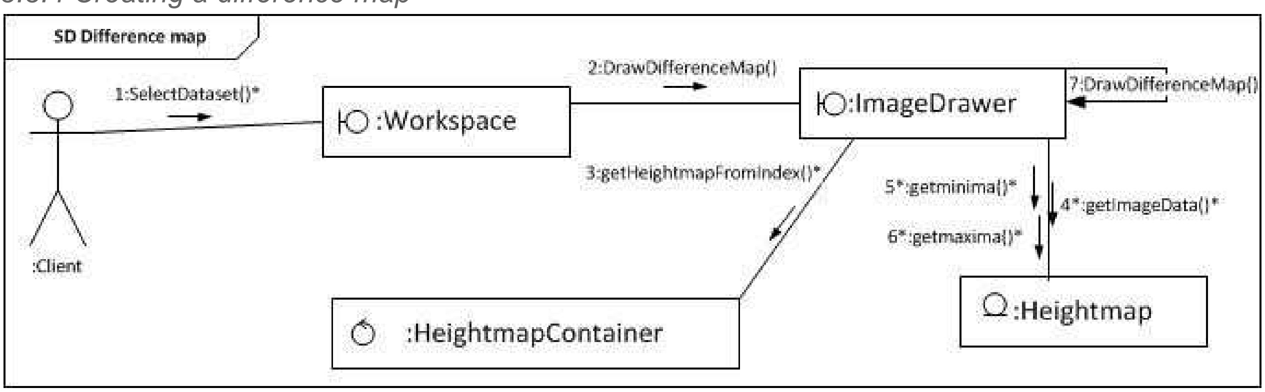
3.3.2 Drawing datasets as heatmaps



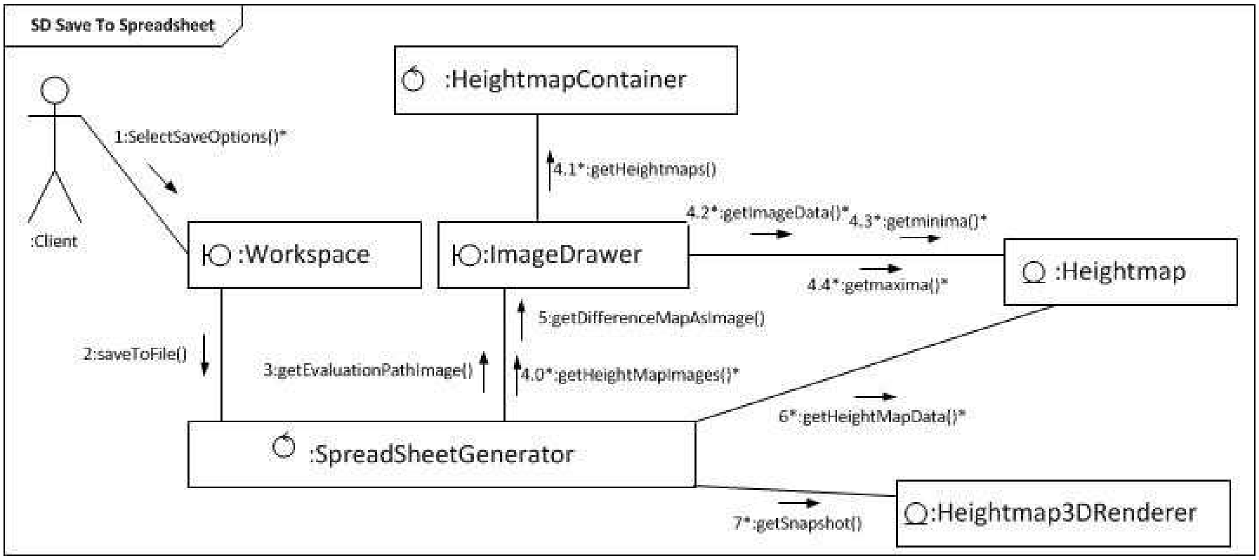
3.3.3 Height evaluation along a path



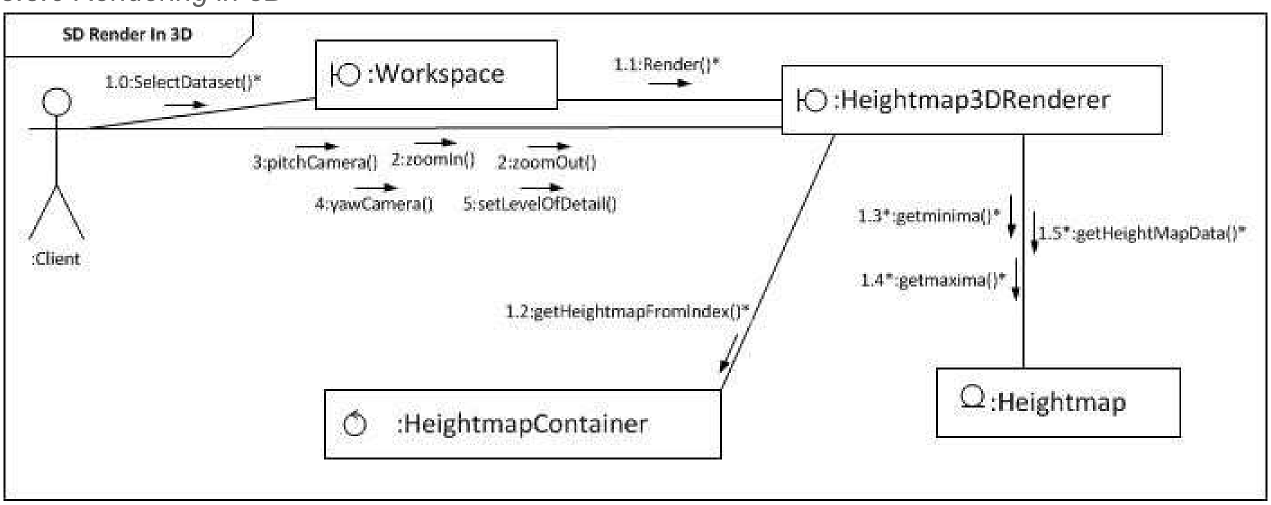
3.3.4 Creating a difference map



3.3.5 Exporting to a spreadsheet

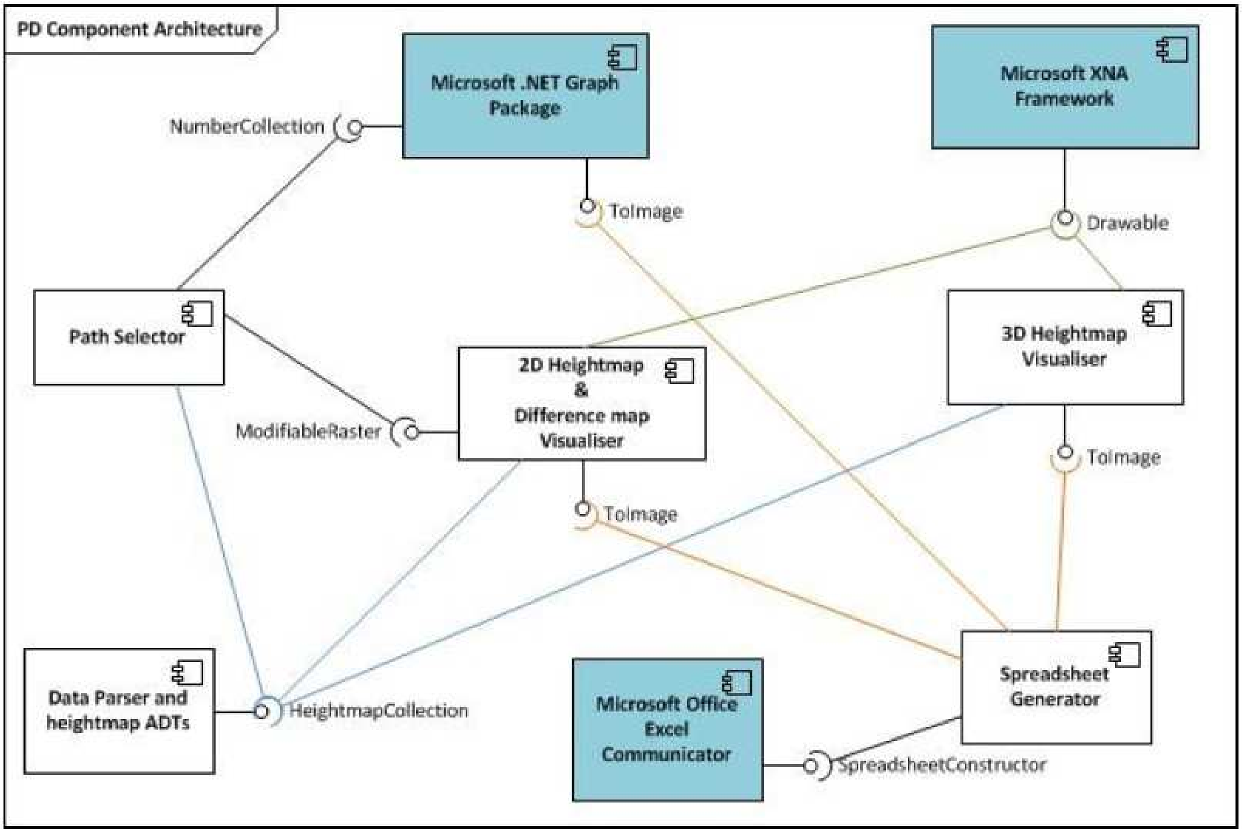


3.3.6 Rendering in 3D



4. Design Model

4.1 System Architecture (Component Model)

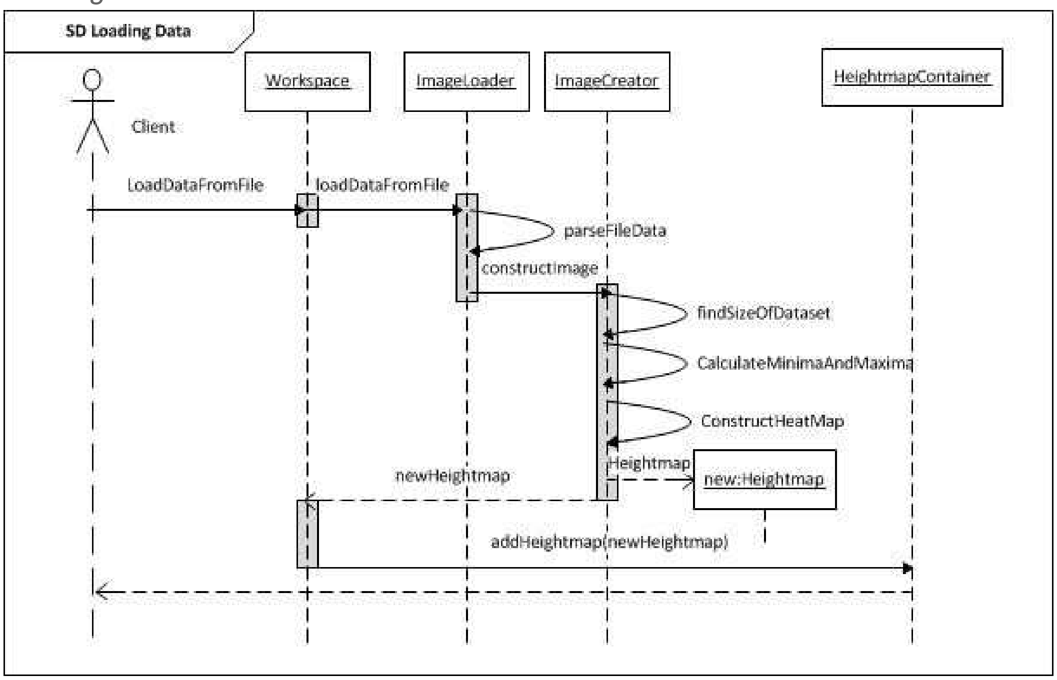


4.2 Interaction Design

4.2.1 Sequence Diagrams

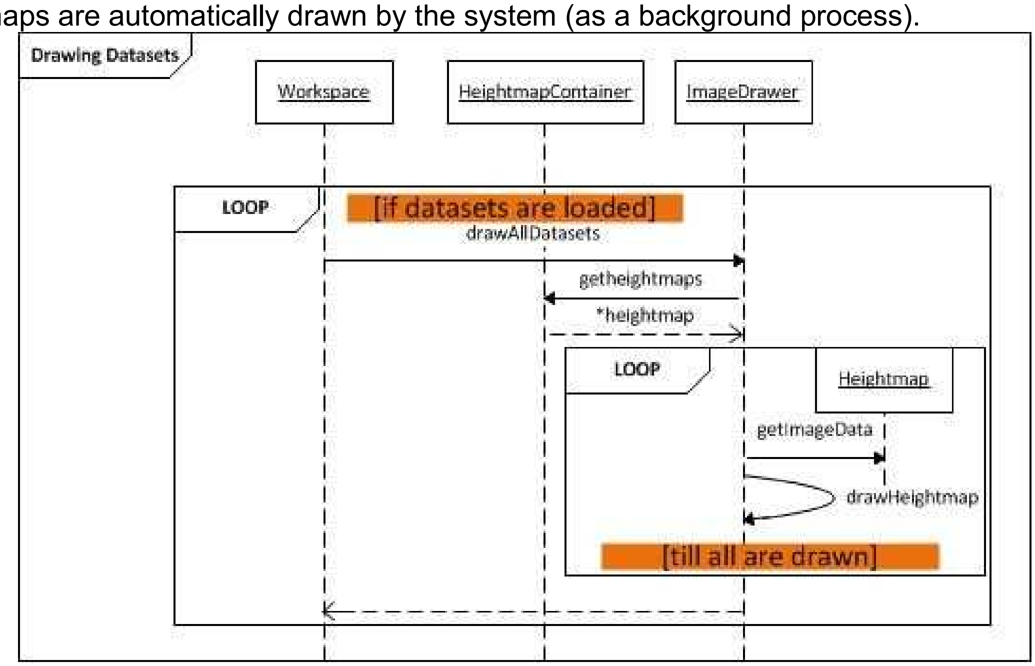
The following state machine diagrams are based on the communication diagrams depicted in the analysis section

4.2.1.1 Loading data

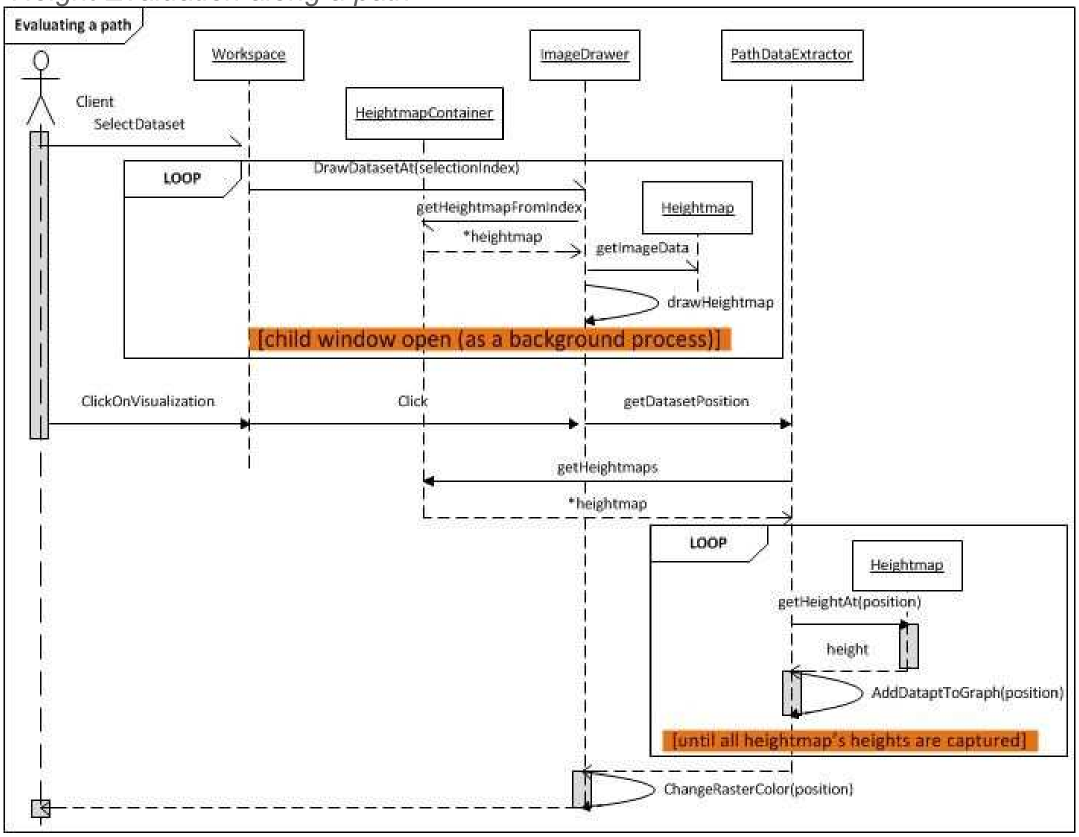


4.2.1.2 Visualising datasets as heatmaps

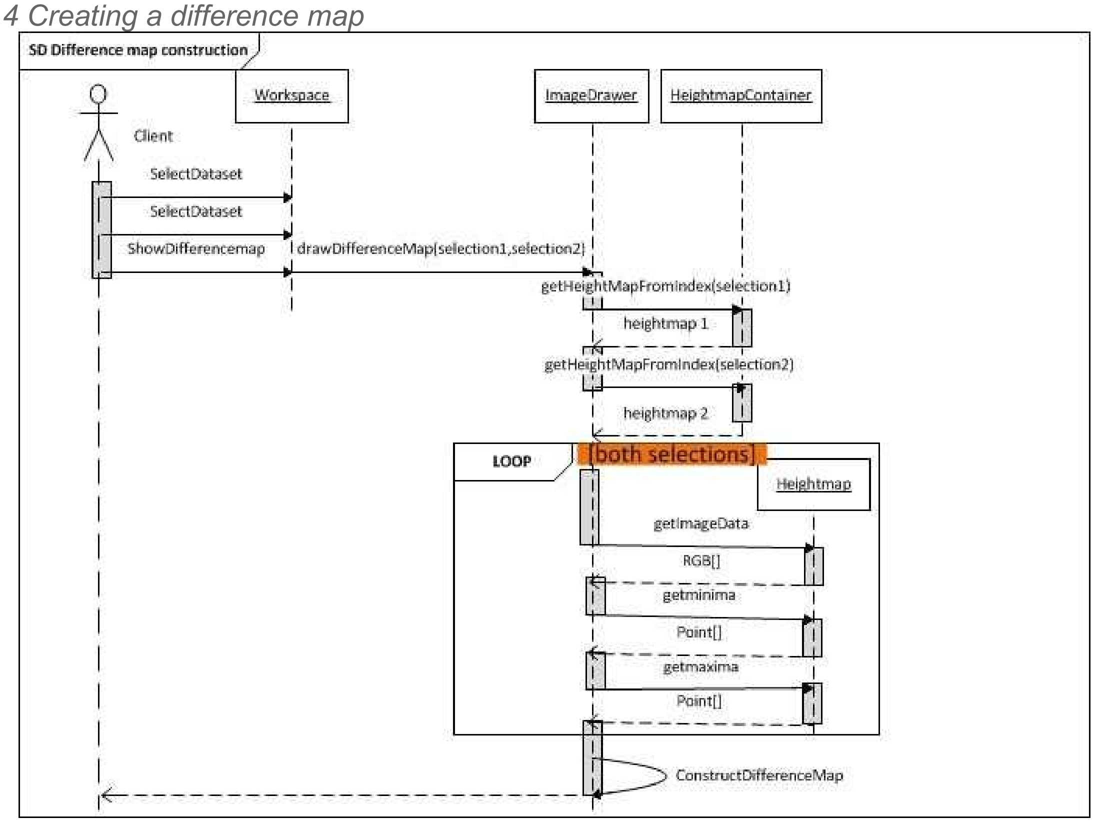
The following sequence of events is entirely automated. Once a user has loaded data the heightmaps are automatically drawn by the system (as a background process).



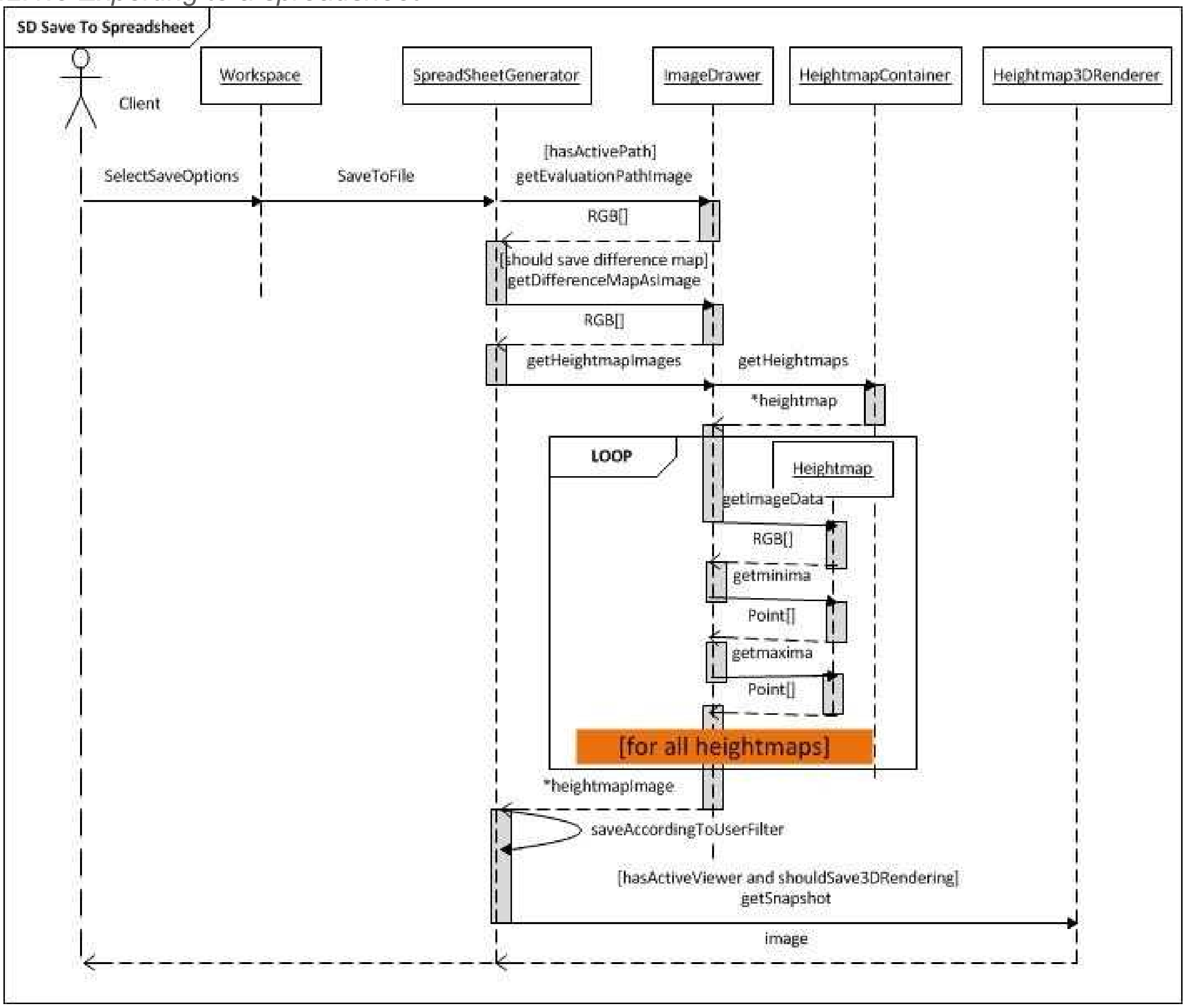
4.2.1.3 Height Evaluation along a path



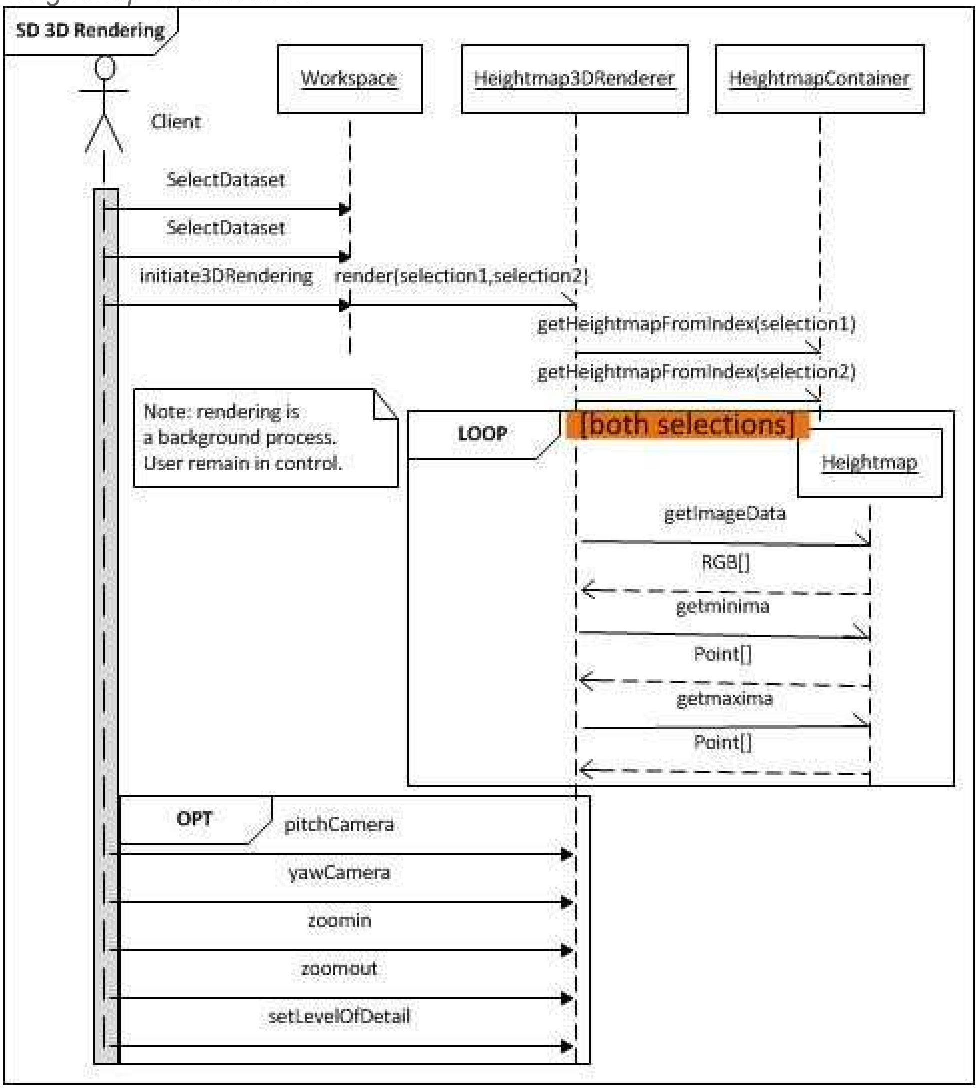
4.2.1.4 Creating a difference map



4.2.1.5 Exporting to a spreadsheet

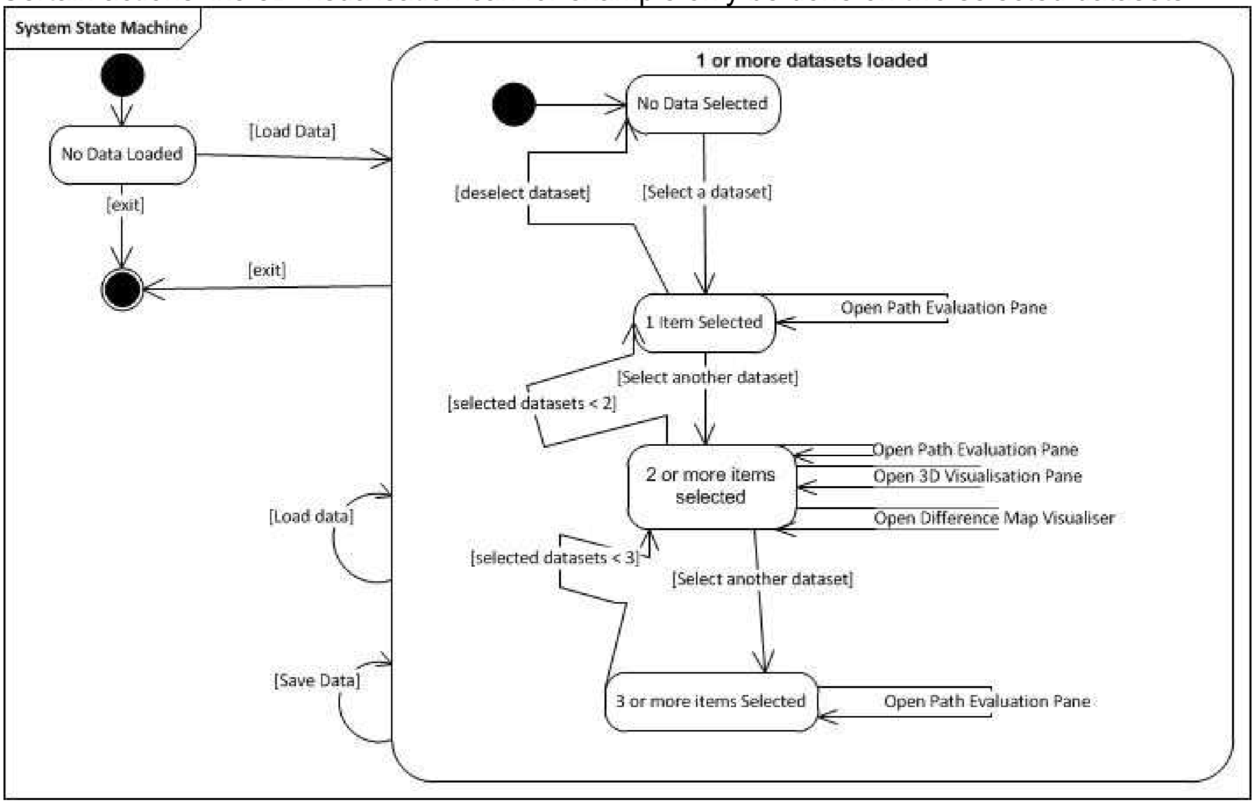


4.2.1.6 3D heightmap visualisation

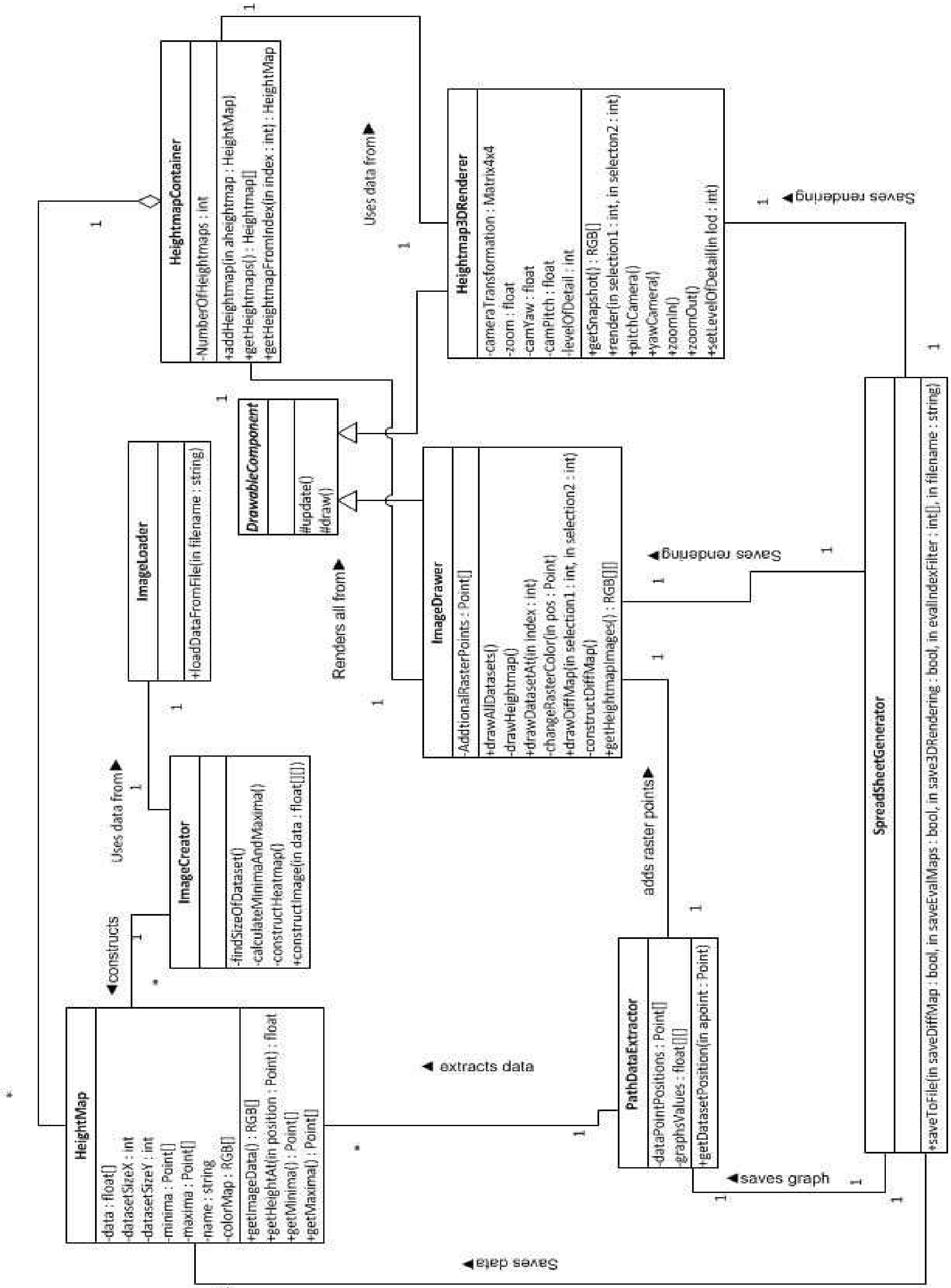


4.2.2 Finite State Machine

The following state machine diagram describes the valid user actions during usage of the tool. Certain actions like 3D visualisation can for example only be done on two selected datasets.

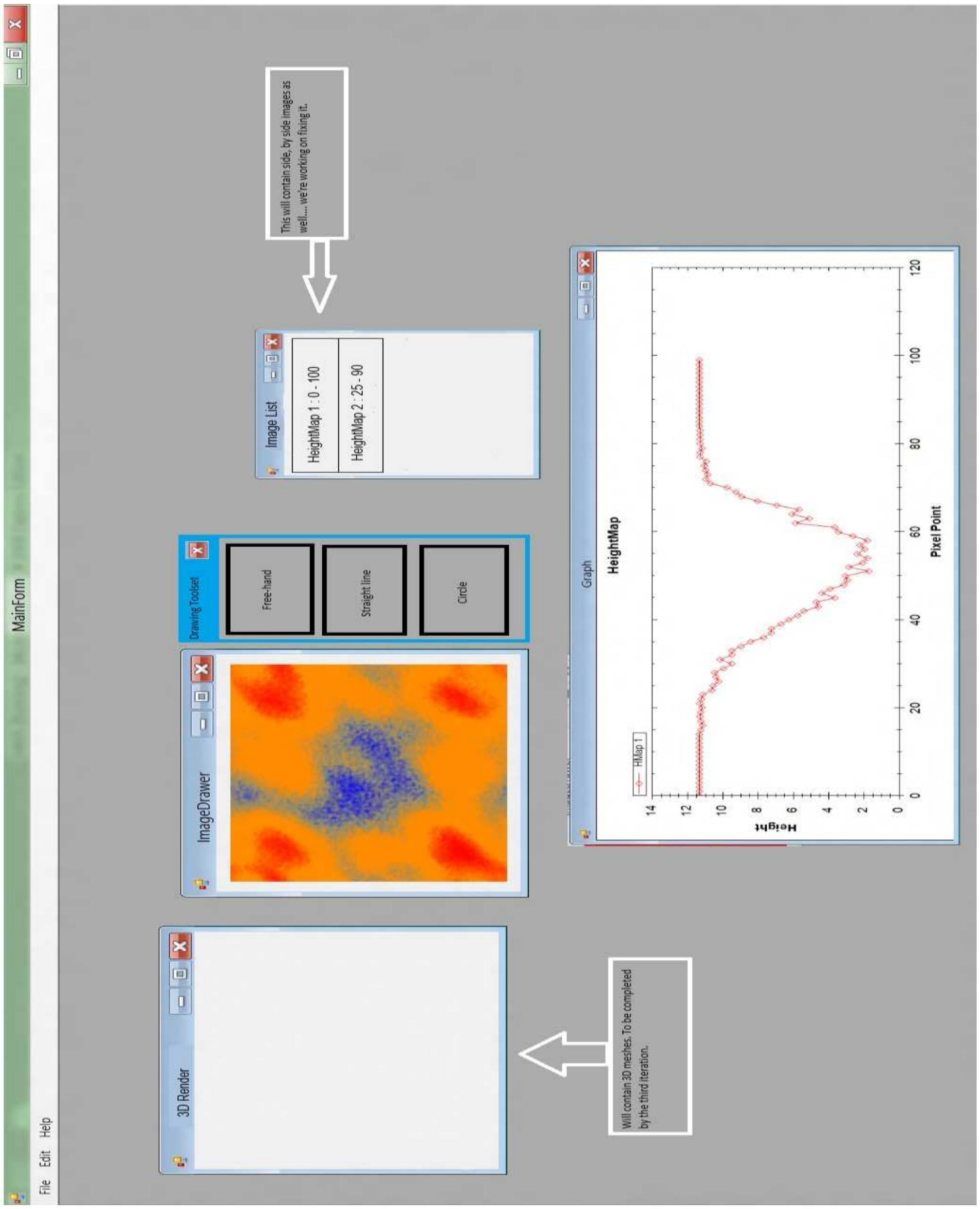


4.3 Class Design (Design Class Diagram)



4.4 Interface Design

The following shows a prototype design for our system using the Microsoft .Net framework.



5. Project construction plan

5.1 Platform

We have decided to use the following implementation platform:

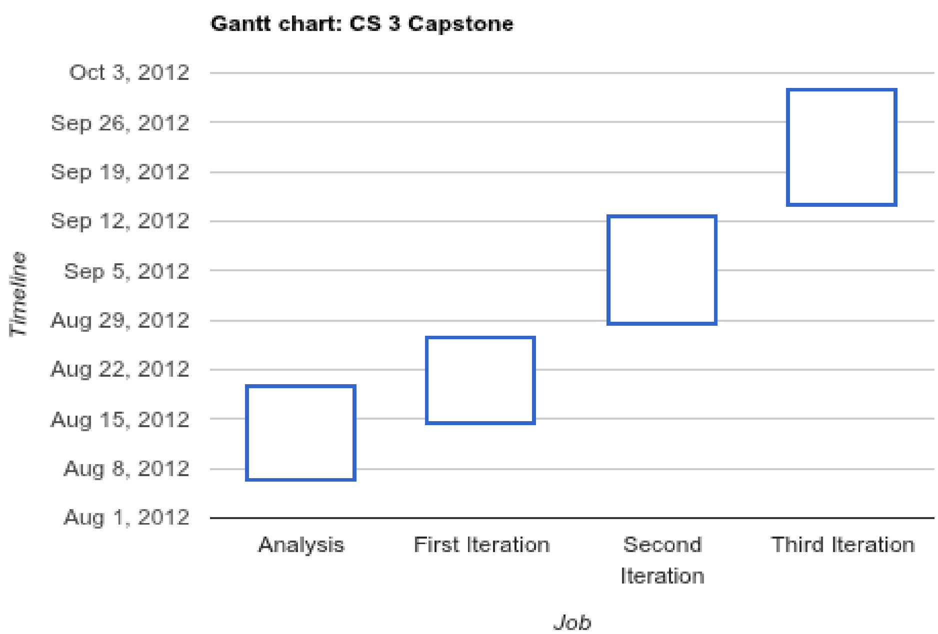
- Microsoft Visual C# 2008
- Microsoft .Net Framework v4.0

- Microsoft XNA Framework v3.1
- OS: Microsoft Windows 7

5.2 Work allocation

We have decided that the project work is to be split up in the following way: Benjamin will play the role of systems analyst, while Brandon implements the tool. Benjamin will help out during the later stages of the implementation, specifically the for the implementation of the 3D visualisation.

5.3 Schedule (Gantt chart)



We have decided to start the prototype as soon as possible. This is why it is running parallel to the analysis phase (some of the features will only be implemented later, so this was possible).

During the first iteration Brandon will construct the following:

- The Dataset loading mechanisms (use case 1)
- Internal storage mechanisms (use case 1)
- 2D visualisation (excluding multiple views) (part of use case 2)
- Path drawing, height extraction and plotting (use case 3)

This will conclude the first prototype.

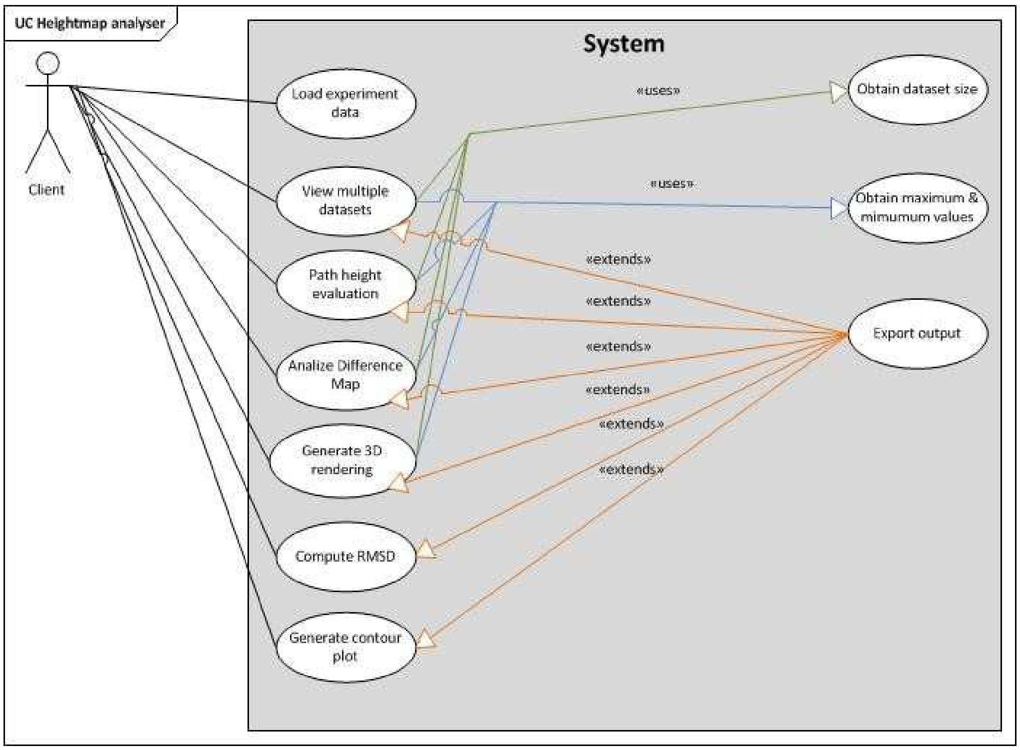
During the second iteration Brandon will construct the grid layout, difference map generation, contour plot and thus completed the core functionality of the program.

During the third iteration Benjamin and Brandon will work on the 3D visualisation together.

6. Change requests (post planning phase)

The client requested that the following small changes to the functionality be added:

- Contour plots
- Root-mean-square deviation calculation
- Spreadsheet generation is to be removed. Only the difference map has to be saved to CSV format.



It was also stated that we could assume that the height maps are equal in size and starts and ends at the same minima and maxima.

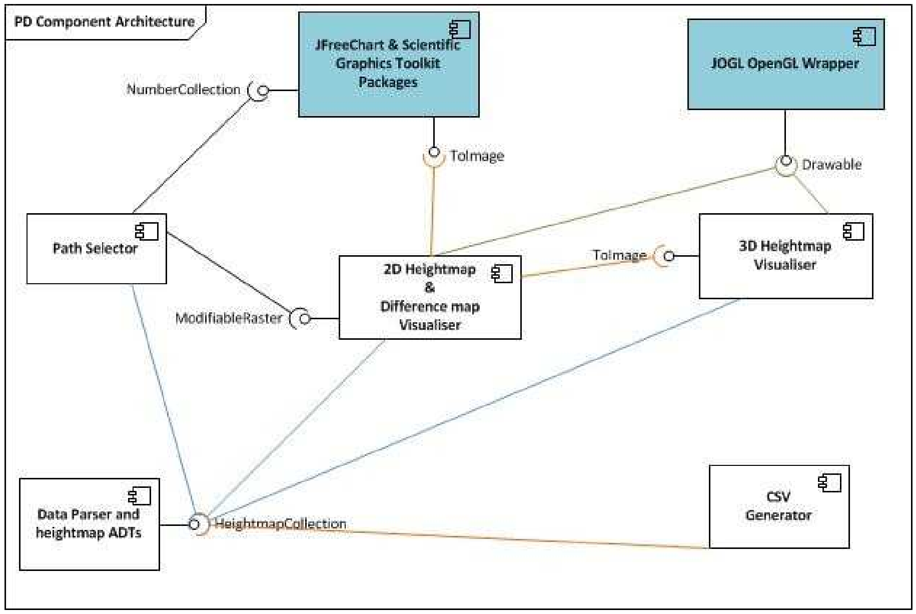
After deliberation we decided to implement the product using Open Source platforms in order to provide a cross-compatible solution. The following platforms will now be used:

- Java Swing framework.
- OpenGL rendering bindings for 3D visualization.
- Ubuntu AMD64 & Microsoft Windows 7 AMD64

7. Implementation

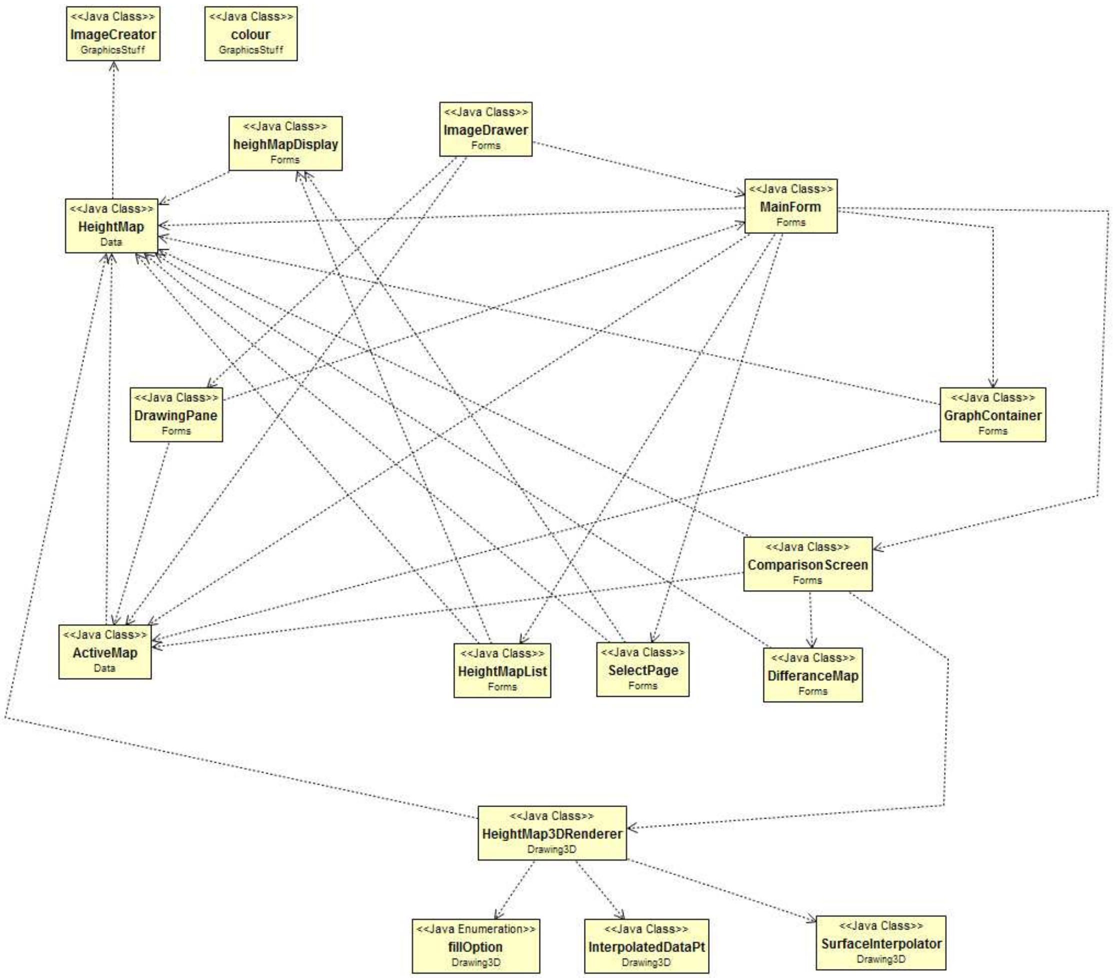
7.1 Changes to the architecture

The change in platforms necessitated that the architecture be slightly modified, as shown below:



7.2 Changes to the class layout

Although the tool was implemented according to the design class diagram initially proposed, the move to other platforms and the change in requirements necessitated additional classes to be added. The following diagram shows the class hierarchy, with most of the dependencies between the classes. It was not possible to add further relationships as the diagram would become too complex.



7.3 Description of algorithms & data structures used

7.3.1 A note on datastructures

No advanced data structures were needed for the purposes of this tool (each dataset can be stored as a two dimensional array).

7.3.2 Threading

Threading was used extensively (when for example the comparison panels were computed, or the linear interpolation was computed for the 3D fly-through tool).

7.3.3 I/O

The datasets are specified as a list of x,y,z coordinates (separated by spaces), from which a grid of values can be computed. The z value can be any real number. Headers and whitespace is ignored.

It is therefore straightforward to read in these values using brute force iteration, and to obtain the minimums, maximums and axis step sizes.

7.3.4 Heat map color interpolation

The algorithm interpolates between red (highest), through orange to blue (lowest) points. These colors can be thought of as triples of values, and therefore we can compute the distances between these points using basic vector mathematics.

The first step is to get the minimum and maximum heights in order to obtain the range of possible heights between them. After this the distance from blue to orange and orange to red is computed. It is then possible to calculate the color of a data point, by using its height as a percentage of the range we computed earlier. When this is done it is simple to interpolate between the colors using this percentage.

7.3.5 Difference map

The difference map is fairly similar to the heat map construction, except that the colors are computed between fully black and fully white. The difference between the heights of two points is computed as a percentage of the maximum height difference, after which the color for the difference can be computed.

7.3.6 Algorithms for the 3D fly through

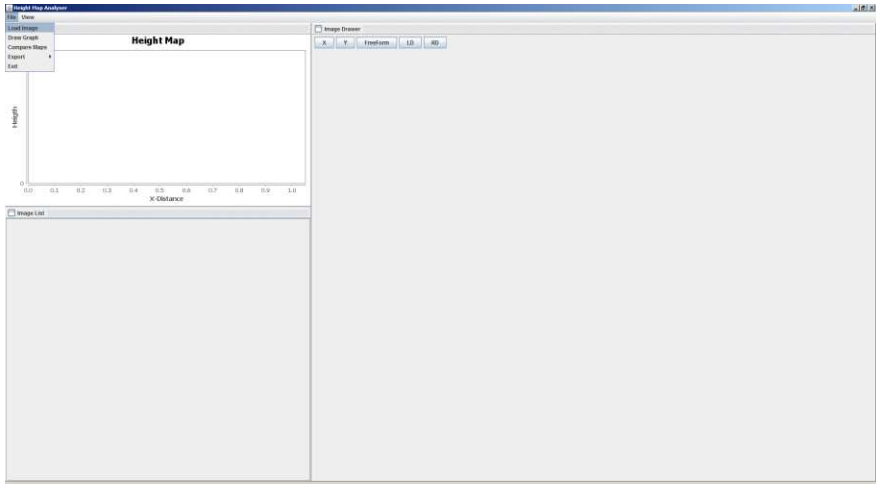
The height pin, as seen in the fly-through tool is created by simply applying the inverse of the projection matrix to convert screen coordinates to world coordinates. When two such coordinates are obtained (one for the near plane and one for the far plane), it is possible to compute a ray in 3 space. This ray is then simply intersected with the Z = 0 plane to obtain an X,Y point, for which heights can be computed.

An interpolated surface is computed by simply doing a linear interpolation between four data (top-left to top-right, top-left to bottom-left, bottom-left to bottom-right, top-right to bottom-right) points and then iteratively computing the inner (stepped) values, each of which constitute a new subsurface, by averaging the intermediate, pre-computed, heights.

8. User guide

8.1 Loading a data set

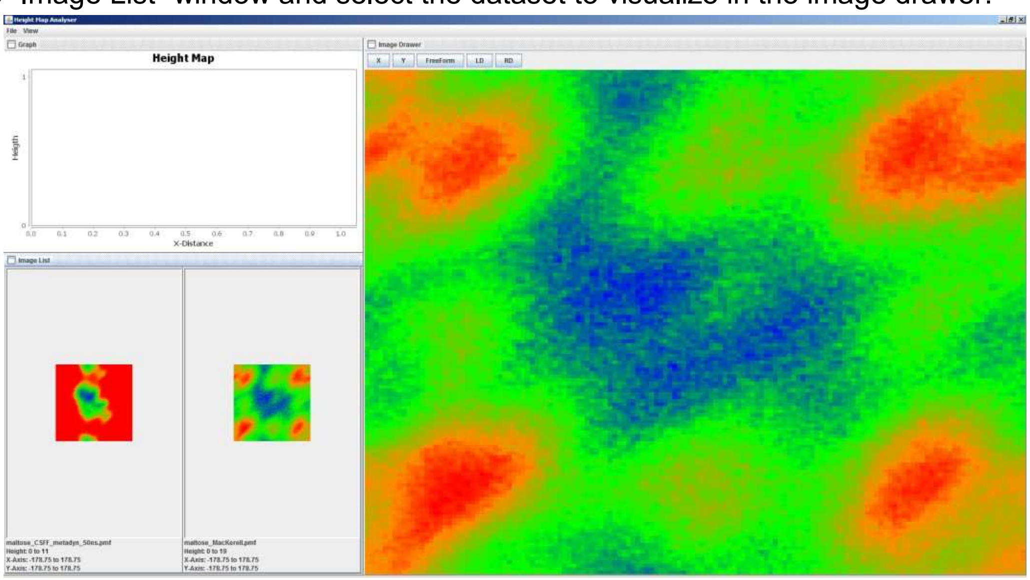
Goto File → Load Image and browse your local directory structure for a suitable dataset.



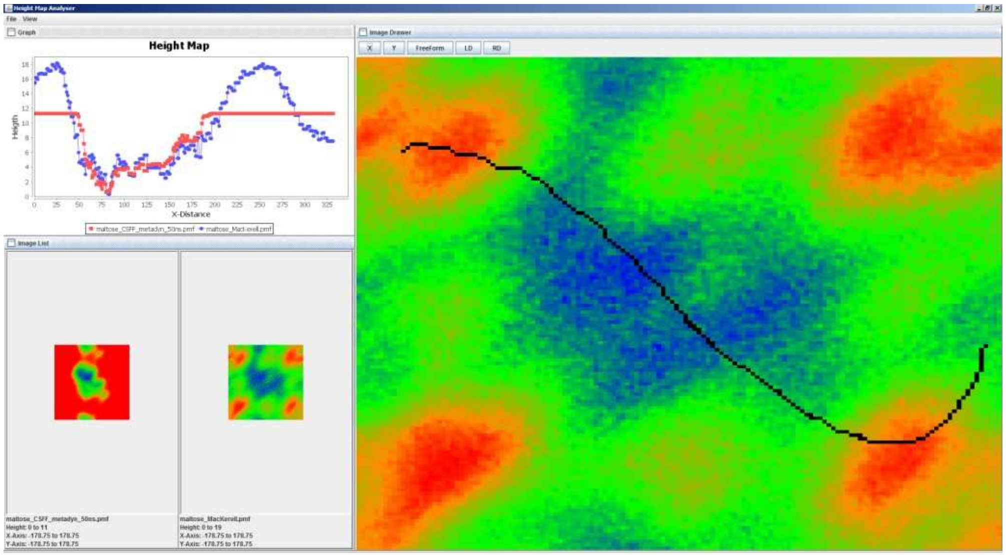
Possible errors:
** An input error can occur if the dataset the user selected is either corrupt or is not a dataset at all. In this case the user will have to regenerate / re-obtain the input file and try again. See 7.3.3 for notes on the format of the input.*
** The user loads a dataset that was previously loaded. The system simply stops the user from performing the illegal action. No error recovery is necessary.*

8.2 Comparing heights along a path

Goto the “Image List” window and select the dataset to visualize in the image drawer.



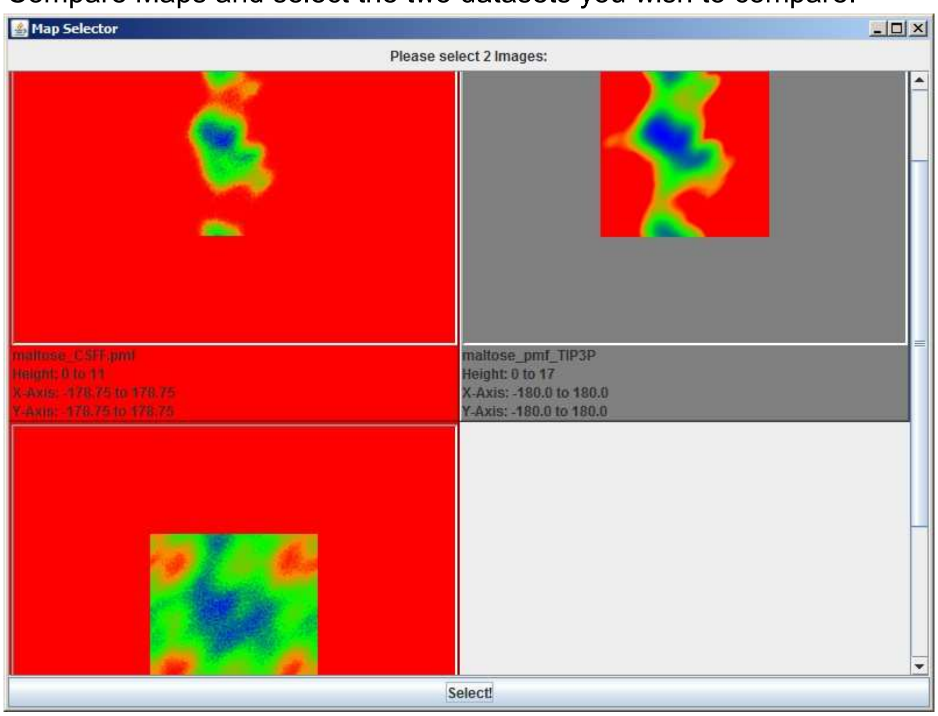
The user may then draw a path by selecting to use one of the drawing tools (indicated at the top of the image drawer (X,Y,FreeForm,LD,RD).



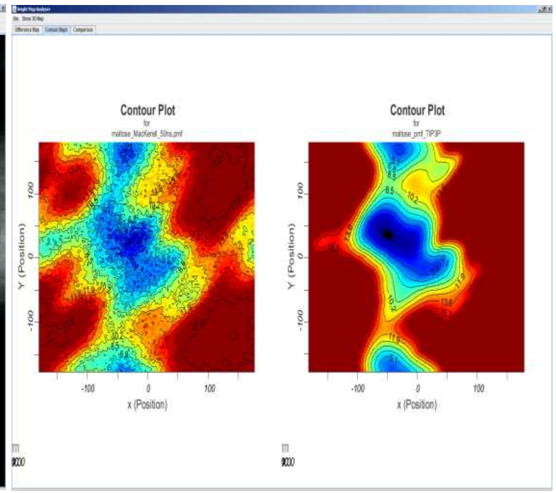
The data, heatmaps and graph can be exported to CSV format by choosing export from the file menu.
Possible error messages: Only exportation can trigger an error if the destination folder is write-protected, in which case the user will have to choose another folder.

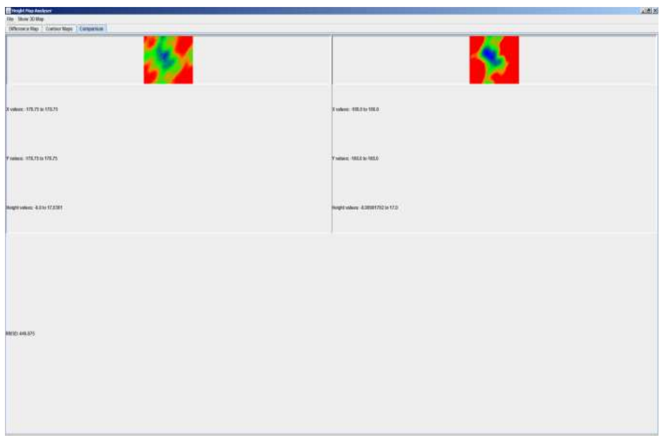
8.3 Comparing datasets

Goto File → Compare Maps and select the two datasets you wish to compare.



The comparison screen will open, displaying the difference map, contour plots, statistics (including rsmd calculation) and a graph of the selected path for only the two datasets that are being compared.

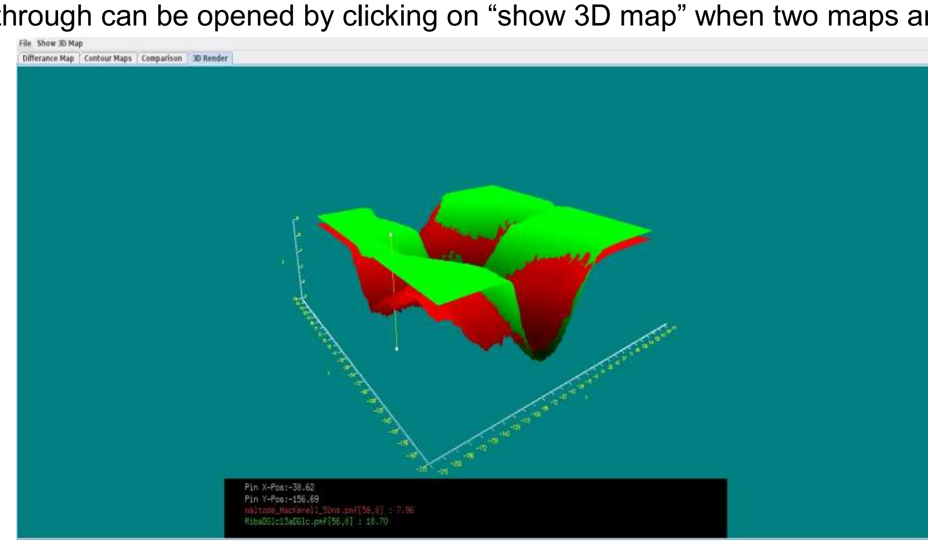


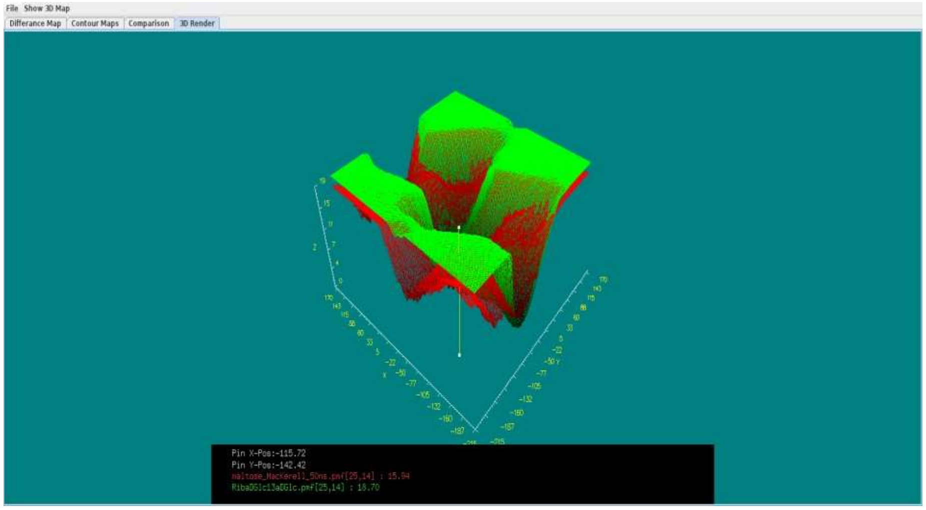


The comparison data can all be exported by choosing Export → CSV from the File menu.
Possible error messages: Only exportation can trigger an error if the destination folder is write-protected, in which case the user will have to choose another folder.

8.4 3D renderer

The 3D fly-through can be opened by clicking on “show 3D map” when two maps are compared.





Rendering parameters can be set by right clicking and choosing one of the parameters (such as number of major, minor axis ticks, font size, wireframe rendering, interpolation detail, z-scale, etc. The rendering can be exported to png image format by using the same pop-up.
The image above shows the effects of setting these parameters.

The camera is rotated by left clicking and dragging the mouse. The mouse wheel can be used to zoom in and out. Simply move the cursor to move the pin indicating where heights are to be measured.
Possible errors: Input validation is done automatically. No errors are produced.

9. Test plan (with change requests considered)

9.1 Overview

We have decided to use a standard framework, called the v-model for the purposes of testing our tool (illustrated to the right).

We will, hence, perform tests on all the units of code we write, as well as keeping in mind the original client requirements and all constraints placed on the overall objectives of the project.

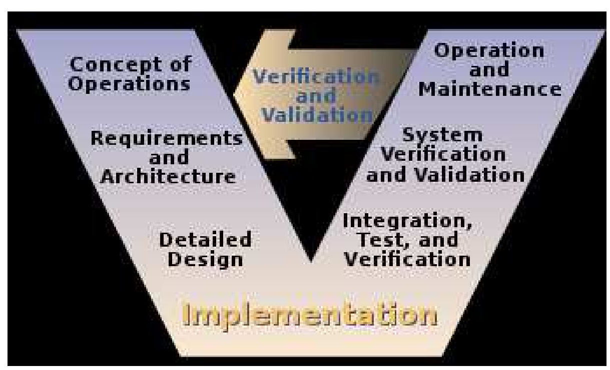


***Figure 2**: V-Model of testing* [2]

As previously specified our tool is primarily a visualizing program. For this reason alone, not much input validation was necessary, because the tool simply loads data and displays it. The primary Human Input Device (HID) is in this case the mouse – and it is primarily used to get coordinates as input.

9.2 Test plan per module (an overview on requirements)

9.2.1 Data parser

Test case	Type	Expected output
Load valid data file	Normal	Loads all lines of data correctly. All heights are correctly loaded into data structure.
Load file with empty lines	Boundry	Skips empty lines, continue to read.
Load file with header	Boundry	Skips header lines, continue to read.
Load file with bogus data (non-floats, missing data points)	Error	Tell the user the data file is corrupt.

9.2.2 2D Visualiser

Test case	Type	Expected output
Multiple data files	Normal	Displays all datasets in a grid layout. User can scroll through grid.
Dataset with positive points only	Boundary	Heat map correctly adjusts to 0 = deep blue, max = bright red.
Dataset with negative points only	Boundary	Heat map correctly adjusts to 0 = bright red, min = deep blue.
Difference map calculation: select 2 heightmaps and initiate visualisation.	Normal	Difference map is correctly calculated. Includes all values of the overlapped area. The non-overlapping area is left as whitespace. Minima, maxima and movement of these points is correctly indicated. Colour ranges from black (least difference to white most difference).
Check that user can only initiate difference map, contour plot & 3D visualisation with 2 datasets.	Error	Disallow users to continue.
Contour plot generation	Normal	Uses the SGT to generate two side-by-side contour plots. Only two datasets can be selected.
Contour plot has correct axis	Normal	Axis (horizontal & vertical is within range minimum to maximum). Minima is at the top-left corner.
Rendering of dataset is the correct way around	Normal	Generate sample input file with 0's around top left position. This area should display in the top-left corner for heatmap, contour plot and difference map.

9.2.3 Path Selector

Test case	Type	Expected output
Draw a path on the input heightmap.	Normal	Draws a visual representation of the cursor movement. Updates the graph correctly (values are the expected values for that area of the dataset).
Test if the user can clear the current path.	Normal	Clears the path and both the graphs.
User can switch between drawing tools: horizontal, vertical, left diagonal, right diagonal and free form line.	Normal	Drawing tool is switched successfully. If the user selects another drawing tool the current path is cleared automatically.

9.2.4 CSV Generator & Export

Test case	Type	Expected output
Have multiple datasets open. User chooses to save the 2D visualisations.	Normal	All the visualizations are saved to png format.
Take a snapshot of the 3D visualisation	Normal	Visualisation is correctly saved to a png image correctly.
User chooses to save difference map.	Normal	If the difference map has been generated already then save it to png format. Does nothing of the difference map has not been generated yet.
User chooses to save contour plots.	Normal	Saves contour plot to png image correctly.
Save CSV	Normal	Saves each data set to csv format. Saves heat maps, rsmd, contour plot (if generated), difference map (if generated) and graphs.
Chooses to save if no dataset is loaded	Error	Nothing should be done in this case.

9.2.5 3D Visualiser

Test case	Type	Expected output
Select two datasets and initiate visualization.	Normal	Renders the 2 datasets each using a different colour-scale. World is centered at the origin. Camera is pointing towards origin.
Selection not equal to 2 datasets.	Error	If more than two datasets are selected then the visualization is not started (too cluttered). Comparison cannot be done for a single dataset (obviously)
Rotate camera	Normal	Camera stays focused at the origin world is rotated correctly when yaw and pitch are adjusted. Full 360 degree rotation is possible (with left mouse button).

Zoom	Boundary	Zoom is only bounded to positive numbers
Level of detail	Boundary	Higher level of detail generates more mesh triangles. The converse holds as well. Level of detail is always >= 0.
Axis font size (slider input)	Boundary	Text grows/shrinks depending on slider input. Text must always be visible.
Sets the number of ticks on the axis.	Normal	Divides the number of ticks correctly for X,Y and Z axis (as well as the number of minor ticks.
Axis ticks is an integer >= 1	Boundary	Both major and minor axis does not allow user to specify a number < 1.
Input for axis ticks is a non-integer	Error	Reject change.
Render mode (limited to a list of states)	Normal	Switches between filled and wireframe correctly.
Height pin is moved according to cursor input.	Normal	Heights are computed correctly. Pin is at the correct position.
Height pin is moved off one or both of the datasets	Error	Out of Bounds is displayed for all 4 boundaries.
Small step sizes	Boundary	Should display points very close together.
Large step sizes	Boundary	Should display points very sparsely spread out.

9.3 Recorded outcomes of testing

Test cases from 9.2.1:
The test cases succeeded. The sample input files from the client are all read in properly (even if some contains blank lines, extra whitespace and header information). If a corrupted file is used as input the program displays an error message.
Test cases from 9.2.2:
Tests succeeded with all the datasets provided by the client. When we loaded our own datasets, as specified in the test cases the heat maps computed correctly. The comparison screen can only be initiated with two data sets. The contour plots correspond to our own heat map construction.
Test cases from 9.2.3:
We've tested the path-height evaluation extensively. The heights are calculated correctly for all the datasets. The user can successfully switch between tools, and the path and graph clears when the user starts drawing a new path.
Test cases from 9.2.4:
Exporting works correctly. When the user saves to CSV all the data, including heat maps, contour plots, rmsd, difference maps and graphs saves to the directory the user selected. When export is chosen from the comparison screen only the selected datasets are exported, along with the path height evaluation for those two datasets.

The 3D fly through is correctly saved when the user chooses the “Take snapshot” option from the pop-up list on that display.
Test cases from 9.2.5:
The test cases succeeded. The rendering changes between wireframe and solid modes. The axis’ major and minor ticks can be set and only accepts numerical input >1. The axis are computed correctly (when compared to the x,y values produced when moving the pin. The font is displayed according to font size. Heights are displayed correctly. Exporting works correctly. Interpolation smoothens by increasing the number of triangles.

9.4 Deployment testing

We have tested the system to work on both Ubuntu AMD64 and Microsoft Windows 7 platforms.

9. References

[1] Images courtesy of an article on heightmaps located at
http://en.wikipedia.org/wiki/Heightmap
[2] Image courtesy of an article on the V-Model located at
http://en.wikipedia.org/wiki/V-Model_%28software_development%29