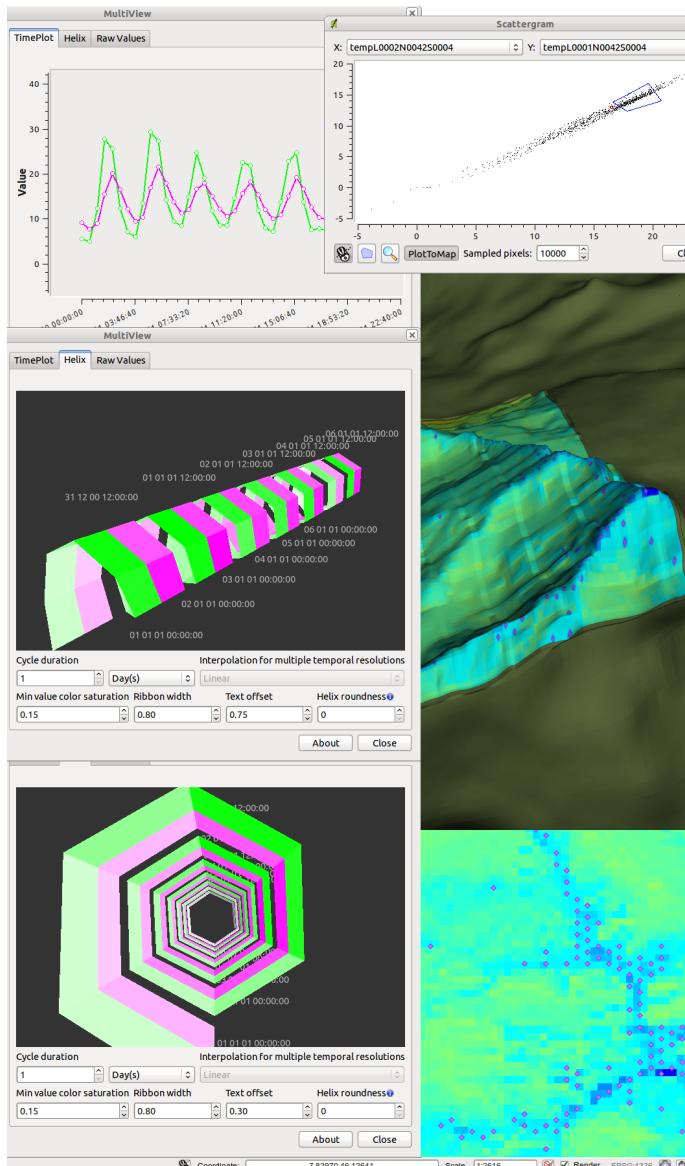




Visualising multivariate spatio-temporal data

Design and evaluation of an open source software environment based on 3D DEMs and multiple linked views



GIScience Center:
Geographic Information
Visualization and Analysis
(GIVA)

Master of Science Thesis [GEO511]
Delivery: 30.04.2011

Marco Bernasocchi
of Sagogn GR, Switzerland

Student-ID: 02-730-620
marco@bernawebdesign.ch

Advisor: Dr. Arzu Çöltekin
Co-Advisor: Dr. Stephan Gruber

Faculty Member:
Prof. Sara Fabrikant, PhD
GIScience Center
University of Zurich
<http://www.geo.uzh.ch>

Acknowledgements

I want to thank my advisor Dr. Arzu Çöltekin for sharing her outstanding technical expertise, continuous support and motivation during this last year. Foremost I want to thank her for her support and encouragement in pursuing my goal. Thanks Arzu!

Another big thanks goes to my co-advisor Stephan Gruber for always being excited and curious about what visualisations could offer to his research. This thesis would not have happened without your problems!

My next thought goes to all the focus-group, expert testing participants and GIVA staff led by Prof. Sara Fabrikant for having helped me design and test my project. Sara, Jeannette, Joel, Russel, Stefanie, Stefano, Stephan and Tumasch thank you very much for your invaluable inputs and patience.

Pirmin Kalberer from sourcepole.ch deserves a very special thanks for having developed the initial version of the globe plugin and for having sorted out its most challenging C++ parts.

To Dr. Ing. Christian Tominski from Universität Rostock, I owe a big thanks for giving the original source code of the helix visualisation and for publishing many inspirational tools and papers.

To all the peoples working on an open source project, helping on forums, mailing lists, IRC, writing how-to, etc. I am grateful for the great effort you are putting in. In this sense I would like extend a special thanks to the QGis developers for developing such an amazing piece of code and for being so helpful on the mailing list and on IRC. I'm looking forward for more work with you all.

To my friends Sarah, Bruno, Pädi, Rahel, Jaqueline, Rolf, Züghi and all the others: thanks for proofreading, lending me a couch in Zurich, skiing, climbing,

diving, listening to music and all the cool things that we do together.

Per finire, vorrei ringraziare la mia famiglia, per sempre avermi supportato e creduto in me anche quando ero ad un oceano di chilometri da casa, eravate sempre qui per me. Grazie di cuore¹.

*"O frati," dissi, "che per cento milia
perigli siete giunti a l'occidente,
a questa tanto picciola vigilia

d'i nostri sensi ch'è del rimanente
non vogliate negar l'esperienza,
di retro al sol, del mondo senza gente.*

*Considerate la vostra semenza:
fatti non foste a viver come bruti,
ma per seguir virtute e canoscenza".*

Dante Alighieri, Divina Commedia - Inferno XXVI

*"If you wanna find out what's behind these cold eyes You'll just have to claw your way
through this disguise"*

Roger Waters - In the flesh?

¹Finally, I would like to thank my family for always supporting and believing in me, even when I was an ocean away, you were always there for me. Heartfelt thanks

Abstract

The aim of the thesis is to conceptualise, implement and evaluate a novel ad-hoc geovisualization tool. The tool combines a 3D terrain model with multiple 2D visualization displays to visualise multivariate spatio-temporal data. It has been given special attention to the used methodology. In fact, the design and the development of such a geovisualization tool can be a very daunting task due to its conceptual and methodological complexity. The design process is based on the combination of users centered design and agile development. The actual conceptualization of the tool is based on concrete users' needs that have been identified and collected during informal brainstorming sessions and in a focus group session. Furthermore, a final evaluation based on expert interviews was carried out to give insight into the usability of the developed tool and to highlight further possible improvements and challenges.

Keywords: 3D digital terrain model, multivariate data, temporal data, multivariate spatio-temporal data, Open Source, Quantum GIS, user centered design

Table of Contents

Table of Contents	vii
List of Figures	xi
List of Tables	xiii
List of Listings	xv
1 Introduction	1
1.1 Steep landscapes of the Matter valley, VS, Switzerland	1
1.2 Motivation	2
1.3 Research questions	4
1.4 Why open source?	5
1.4.1 Quantum GIS	6
1.5 Thesis structure	7
2 State of the art	9
2.1 Multivariate data visualisation	9
2.1.1 Scatterplots	10
2.1.2 Line graphs and multiple line graphs	11
2.1.3 Parallel Coordinates Plots	11
2.2 Multivariate temporal data visualisations	12
2.2.1 Time extended parallel coordinates plot	13
2.2.2 Spiral visualisations	13
2.2.3 Axes-Based Visualizations with Radial Layouts	14

2.3	Spatio-temporal data visualisations	15
2.4	3D digital terrain model visualisations	16
2.5	Linked views	18
2.6	Available tools	20
3	Design	23
3.1	Requirements from user tasks analysis	24
3.2	Focus Group	25
3.3	Formal requirements	26
3.3.1	Must-have	27
3.3.2	Nice-to-have	27
3.4	Evaluation	28
4	Implementation	29
4.1	Developement strategy	29
4.1.1	Software used	29
4.1.2	Agile software developement	30
4.1.3	Code management with GIT	31
4.2	System architecture	33
4.2.1	QT framework	34
4.2.2	PyQt framework	36
4.2.3	QGis API	36
4.2.4	QGis plugin architecture	36
4.3	Globe C++ plugin	37
4.3.1	Architecture	37
4.4	MultiView Python plugin	39
4.4.1	Architecture	40
4.4.2	Rawvalue widget	42
4.4.3	TimePlot widget	42
4.4.4	HelixView widget	43
4.5	Scattergram Python plugin extension	44
4.6	Plugins starter Python plugin	45

5 Results	47
5.1 Expert interview setup	47
5.1.1 Test setup	48
5.1.2 Test tasks	48
5.1.3 Participants structure	49
5.2 Results evaluation	50
5.2.1 Data loader	50
5.2.2 QGis layer visibility	52
5.2.3 TimePlot visualisation	53
5.2.4 HelixView visualisation	56
5.2.5 Comparison of temporal visualisations	58
5.2.6 Scattergram	60
5.2.7 Globe	61
5.2.8 General remarks	62
6 Discussion	63
6.1 Formal requirements analysis	63
6.2 Answering the research questions	65
7 Conclusions	69
7.1 Known limits	70
8 Future work	71
8.1 Globe plugin	71
8.1.1 Google Summer Of Code Program [GSOC]	72
8.1.2 Globe on geowall	73
8.2 MultiView plugin	73
8.2.1 Data loader	74
8.2.2 TimePlot visualisation	74
8.2.3 Helix visualisation	75
8.3 Scattergram extention	76
8.4 Plugins starter plugin	77
8.5 Time manager plugin raster support	77
8.6 Concluding personal note	78

Bibliography	79
A Appendix	89
A.1 QGis API UML	89
A.2 Globe plugin	90
A.3 Multiview	91
A.3.1 Dataloader	91
A.3.2 Raw value widget	92
A.3.3 TimePlot widget	92
A.3.4 HelixView widget	93
A.4 Plugins starter plugin	93
A.5 Scattergram plugin	94
A.6 Focus group participant's mockups	95
A.7 Expert Interview Script	100

List of Figures

1.1	Envy, the main visualisation tool currently in use at GIUZ	3
2.1	Scatter plot matrix and Time plot [sources: Grinstein et al. [2001]; http://www.xjtek.com]	11
2.2	Parallel coordinates plot and Time extended parallel coordinates plot [source: Hautle [2010]]	12
2.3	Enhanced Interactive Spiral Display and 3D Helix [sources: Tominski et al. [2004]; Weber et al. [2001]]	13
2.4	TimeWheel and MultiComb [source: Tominski et al. [2004]]	14
2.5	3D TimeWheel and Stardiagramm [source: Aigner et al. [2008]; Noirhomme-Fraiture [2002]]	15
2.6	Pencil and helix glyphs places on a 3D map [source: Tominski et al. [2005]]	16
2.7	Linked views example [source: OEC [2011]]	18
3.1	User centred design methodology after Torres [2002]	23
4.1	Agile iteration workflow [source: http://extremeprogramming.org]	31
4.2	QGis high level architecture [source: http://www.qgis.org] .	33
4.3	Qt framework [source: http://qt.nokia.com]	34
4.4	Qt signal/slot framework [source: http://qt.nokia.com] . . .	35
4.5	Evolution of the Globe plugin. Basic globe, globe with GUI, globe with GUI and stereo 3D	38

4.6	Globe plugin dependencies and high level architecture [Own illustration]	39
4.7	Globe plugin's stereo 3D and DEM settings dialogs	40
4.8	MultiView architecture overview [Own illustration]	40
4.9	High level architecture of Scattergram's plotToMap functionality [Own illustration]	44
5.1	Temporal Data loader interface pre expert testing.	51
5.2	Dialog presented to users when an header file is found in the data import folder	52
5.3	MultiView plugin, old dataloader and new dataloader icons	52
5.4	TimePlot visualisation as used during the expert testing with added pre and post expert testing warning box	54
5.5	TimePlot visualisation sketch with added last selected values lines	55
5.6	HelixView visualisation as used during the expert testing with added pre and post expert testing warning box	59
5.7	Scattergram visualisation used during the expert testing	61
5.8	Globe tilt and rotate controls Pre- and Post- expert interview	62
A.1	QGis API UML diagram [Source: http://www.qgis.org]	89
A.2	Globe plugin UML class diagram [Own illustration]	90
A.3	MultiView plugin high level architecture UML class diagram [Own illustration]	91
A.4	Dataloader UML class diagram [Own illustration]	91
A.5	Raw value UML class diagram [Own illustration]	92
A.6	TimePlot UML class diagram [Own illustration]	92
A.7	HelixView UML class diagram [Own illustration]	93
A.8	Plugins starter UML class diagram [Own illustration]	93
A.9	Scattergram UML class diagram [Own illustration]	94
A.10	Participant 1 GUI mock-up	95
A.11	Participant 2 GUI mock-up	96
A.12	Participant 3 GUI mock-up	97
A.13	Participant 4 GUI mock-up	98
A.14	Participant 5 GUI mock-up	99

List of Tables

2.1	Summary of rules and areas of major impact on utility [source: Wang Baldonado et al. [2000]]	19
5.1	Experts previous knowledge (PK) in years of experience and daily usage (DU) of GIS, QGis, environmental data (ED) and temporal data (TD)	50
6.1	Summary of the implemented features from the formal require- ments. * Means low and *** high.	63

List of Listings

4.1	MultiView timestep duration example array	41
4.2	MultiView multidimensional example result array	41
4.3	MultiView multidimentional example array before and after pre-processing in setData	43
4.4	HelixView's DrawWithNames algorithm in pseudocode	44
4.5	Scattergram's coordinates mapping algorithm in pseudocode . . .	45

1

Introduction

In this chapter I will first introduce the reader to the problems faced on a daily basis by researchers dealing with multivariate spatio-temporal in mountainous terrain. An example from a steep region in the canton Valais of Switzerland will help illustrate the real extent of the issue. The example will also serve as starting point for the definition of this thesis' research questions presented in the second part of this chapter. In section 1.4 I will explain why I wanted the whole project code to be available to the public domain by publishing it under the GNU Public License. The last section of this chapter will then explain the structure of the thesis.

1.1 Steep landscapes of the Matter valley, VS, Switzerland

The slopes on the orographic right side of the Matter valley¹ exhibit a number of differing slope movement, many of which are hypothesized to be controlled by subsurface ice and the impact of environmental change on it. In a large

¹The Matter valley is located in southwestern Switzerland, south of the Rhone valley in the canton of Valais. The village of Zermatt is the most important settlement of the valley, which is surrounded by many four-thousanders, including the Matterhorn [Source: <http://en.wikipedia.org/wiki/Mattertal>].

multi-disciplinary project, these movements are currently investigated to elucidate the underlying processes. Movement is detected in spatial domain by differential SAR Interferometry and temporally with a high number of continuous GPS recorders. For the fast testing of initial hypothesis (e.g., «the recorded acceleration in spring is concurrent with the pulse of snowmelt water infiltration») a physics-based cryosphere model is developed and tested at Geographic Institute of the University of Zurich [GIUZ]. This model can be run in 1D mode, i.e. simulation of the temporal evolution of one point in a landscape or in 2D mode, simulating a large grid of points. The testing and interpretation of this model can greatly benefit from an intuitive display and interactive analysis of several time-dependent variables in their topographic context [Gruber, 2011].

1.2 Motivation

The example of the previous section is just one of many examples of datasets that become almost too big to be efficiently analysed without dedicated visualisations. In the last decades, the amount of data generated by sensors and models has become almost overwhelming: datasets easily reaching the multi-Terabyte dimension and in some cases going far beyond it like at the large Hadron Collider at CERN where roughly 15 Petabytes of data are produced annually².

Big geographic multivariate time series datasets are extremely difficult to analyse and gaining insight from them is even more challenging. Many established methods such as PCP (parallel coordinates plots), space-time cubes, scatter-plots and multiple visualisations can help gaining more information but in very steep terrain the addition of a 3D terrain based visualisation could help improve the readability and thus the information gain from those datasets. The main visualisation tool currently in use at GIUZ 3G³ department for geoTOP data analysis is Envy 1.1. To analyse the data, the operator visualises a scatter-plot, a pixel

²<http://public.web.cern.ch/public/en/lhc/Computing-en.html>

³The research of the unit 3G of the Department of Geography at the University of Zurich focuses on cold mountain ranges and processes that result from the interaction between climate and the surface. The related research topics are on glaciers, permafrost, geochronology, and natural hazards [Source: <http://www.geo.uzh.ch/en/units/physical-geography-3g>].

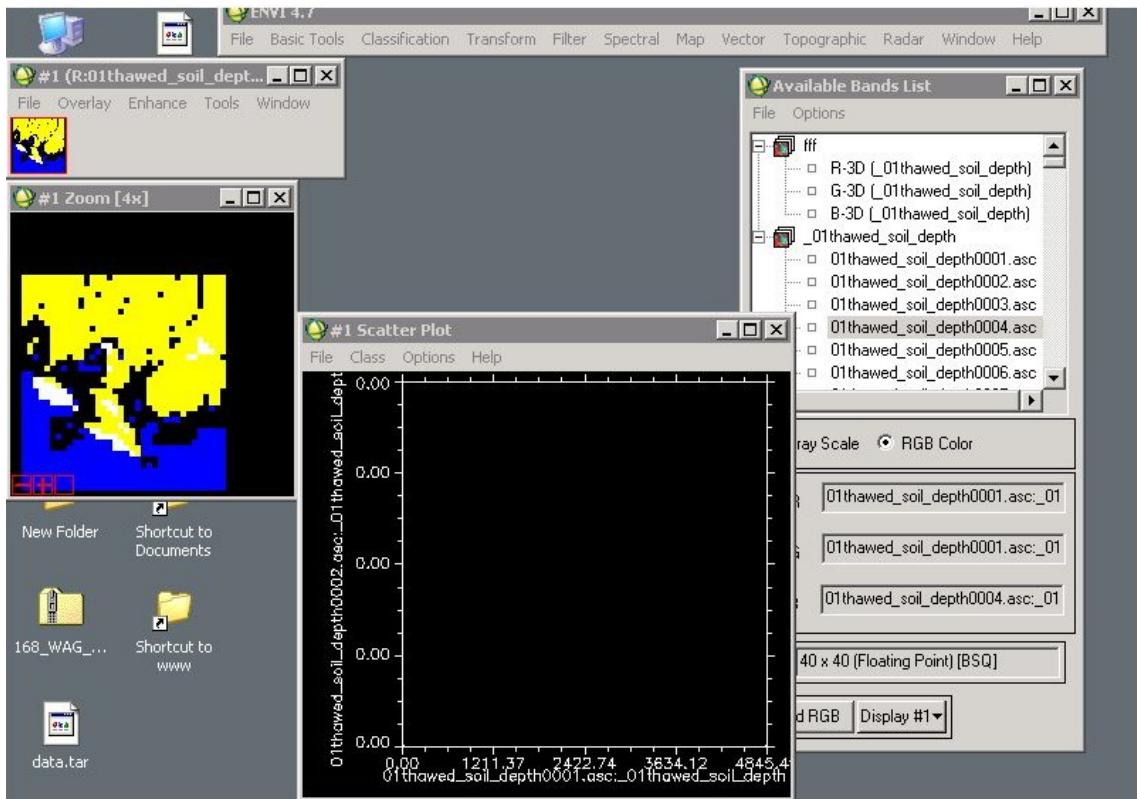


Figure 1.1: Envy, the main visualisation tool currently in use at GIUZ

map and a time-slider linked together. The analysis is then performed by moving the time cursor around different periods. Direct comparison is only available by opening two concurrent sessions of envy. The necessity of a better visualisation system to allow a thorough analysis of the data has risen with the growing of the dataset produced with the geoTOP model and is one of the main concerns addressed in this thesis.

The visualisation of multivariate spatio-temporal data is currently hot topic in visualisations research and different groups have published taxonomy works on how to categorise such visualisations based on the data that need to be analysed. A very good overview of all available techniques is given in Aigner et al. [2007, 2008], Andrienko et al. [2003] and Daassi et al. [2006].

Innovative visualisations such as Space-Time cube, Multi-variable-Time cube (MTC) and PCP-Time-Cube (PTC) [Li and Kraak, 2005], helix and pencil glyphs [Tominski et al., 2005], timewheel [Tominski et al., 2004], enhanced spiral display

[Tominski and Schumann, 2008] are only a few examples of how researchers have been trying to visualise spatio-temporal multivariate data to allow data exploration and analysis in an intuitive and usable way. Intuitivity and usability have become even more important in the modern society, since spatio-temporal analysis is not solely the business of professional analysts any-more. Many non expert users are interested in the possibility offered from such analyses and therefore researchers should try to develop usable and intuitive tools [Andrienko et al., 2010].

1.3 Research questions

The main aim of this work is to research the possibility for the creation of a modular, open source framework for multivariate spatio-temporal data visualisation and analysis directly implemented in a mature open source GIS software. The tool should allow the user to analyse and visualise data on various visualisations using a three dimensional terrain module with surface topography as input controller. The development of the tool and the therefore necessary research should answer the research questions listed below. A systematic evaluation of the developed tools is not part of this thesis as it would go beyond the scope of a master thesis, but it would be an ideal subject for follow-up work.

The first research question is dedicated to a deeper understanding of linking techniques between multiple visualisations and how to successfully use them with a three dimensional globe display.

Research question 1:

How can a 3D terrain be successfully linked to multivariate spatio-temporal data visualisations?

The second research question is about discovering which visualisations are more suited for visualising temporal and cyclic data generated by sensors or models used in environmental studies.

Research question 2:

Which temporal data visualisations would make our system intuitive, effi-

cient and easy to use with the specific type of data (i.e. temporal and cyclic data) generated by environmental monitoring projects?

The third question deals with a recurrent problem in data analysis and model validation: outlier detection. Instead of an algorithmic solution, a visual tool is required to allow the user to decide if the data anomaly is due to a model or sensor malfunction or if it is a correct value shifted from the average because of its location.

Research question 3:

How can data anomalies and outliers be visually correlated to the topography?

The last research question is dedicated to an evaluation of the effectiveness of adding a topographic context to a visualisation suite compared to bare “traditional” two dimensional and three dimensional plots.

Research question 4:

Does adding the surface topography (on a DEM), in comparison to more abstract visualisations (various 2D or 3D plots) make a (positive) difference in inference making in environmental (and similar) monitoring projects?

1.4 Why open source?

Every program has source-code, some companies and individuals decide to hide it and to deliver the customers only installable binaries, other individuals and companies decide on the other hand to allow everybody to have the source of the program. At first it might seem that this is a bare technical caveat, but the availability or not of the source code means a lot in terms of business models, customisability of the program, technology lock-in and finally community benefit.

This section does not want to be a debate about advantages and disadvantages of open source software, nor a debate between closed source software, open

source software and free (as in speech) software⁴ but a simple motivation of the choice to prefer open source⁵ over closed source development.

The main point of choosing open source for me was the possibility of freely sharing my work with others to allow a real knowledge passing to whoever is interested in it. Privately available knowledge is to me worth almost as little as no knowledge and thus I absolutely wanted my work to be publicly available. If all past knowledge was to be available only against payment, the whole scientific world would suffer from it. The second key argument was the quality of the code. Open source codes get much more reviews and testing than closed source codes due to their intrinsic open and communitarian nature and are therefore fixed and enhanced more often. A third argument is the ability of taking a software and being able to adapt it to the exact necessities of this project. This was for example the case of one of the plugins programmed which needed an enhancement in the API⁶ of the main program. In a closed development environment I would have had to file a feature request ticket and wait for the company to implement it and to release a new program version. On the other hand, in the open source environment, I created a patch to the main program, I submitted it and an hour later one of the main developers⁷ had accepted and applied it. The last reason that drove me toward open source is that I daily use only open source software and that I wanted to use this opportunity to give something back to the community.

1.4.1 Quantum GIS

Once faced with the decision on which platform to use for the development of the project, multiple reasons have been considered. Beside the main requirement of being open source, the software had to be user friendly, mature, cross-platform,

⁴for this I suggest the reader Feller and Fitzgerald [2002], fre [2011] and Lakhani and von Hippel [2003] or for a more socio-political approach Torvalds and Diamond [2001] and Williams [2002]

⁵The software used in this thesis is mostly always free software but to be correct, I will use the expression open source which is a bit more lax and covers all the software used.

⁶Application programming interface

⁷The availability of the source means that anybody can have the source and modify it the way they want. This does not mean that every change has to flow back to the main source code and that there is no quality assurance. Decision on how to manage this are usually taken by a project steering committee.

have a plugin infrastructure and be under active development. The programming languages used in the software were only a secondary concern since learning a new language was seen as an opportunity instead of a hurdle. The ease of implementing a globe into it was evaluated as well. Among the considered candidates (gvSIG⁸, OpenJump⁹, QGis¹⁰, uDig¹¹) QGis was chosen since it best matched the required criteria.

Quantum GIS [QGis] is an Open Source GIS that runs on Linux, Unix, Mac OSX, and Windows. QGis supports vector, raster, and database formats and it is licensed under the GNU Public License. The Quantum GIS project was officially born in May of 2002 when coding began. The idea was conceived in February 2002 when Gary Sherman began looking for a GIS viewer for Linux that was fast and supported a wide range of data stores. That, coupled with an interest in coding a GIS application led to the creation of the project. In the beginning Quantum GIS was established as a project on SourceForge in June 2002. The first code was checked into CVS on SourceForge on Saturday July 6, 2002, and the first, mostly non-functioning release came on July 19, 2002. The first release supported only PostGIS layers.¹²

1.5 Thesis structure

The structure of this thesis represent the development process used during the whole project. After this introductory chapter, chapter 2 will describe the current state of the art in multivariate spatio-temporal data visualisation by reviewing the main software packages available today and the techniques they use.

In chapter 3 I will explain what design decision were made and how it came to such decisions. The chapter will also list the formal requirements of the code to be developed for this project.

Chapter 4 will be dedicated to the explanation of the development strategy

⁸<http://www.gvsig.org>

⁹<http://www.openjump.org>

¹⁰<http://www.qgis.org>

¹¹<http://udig.refractions.net>

¹²[source: <http://www.qgis.org/en/about-qgis.html>]

used and to a detailed explanation of the system architecture of each of the developed tools.

In chapter 5, the results of the expert interviews will be assessed and will introduce the reader to the discussion of chapter 6. Lastly, chapters 7 and 8 will close the thesis and give directions for future developments of the tools developed during this project.

2

State of the art

Many different visualisations for spatio-temporal datasets exist nowadays [Aigner et al., 2007; Andrienko et al., 2003]. Each visualisation has pros and cons depending on the kind of data that is analysed. The main difficulty lies in choosing an appropriate visualisation for the required task. In the following sections I will give a short overview of the currently available visualisations techniques and tools, separated by the type of data they aim to visualise. Based on this review, the following chapter will illustrate how the tools to be developed are conceptualized.

2.1 Multivariate data visualisation

Multivariate datasets are defined by the presence of multiple dependent variables for each point in the data. Due to the intrinsic nature of the dataset, all variables convey part of the information contained in the whole set. In explorative data analysis, the ability of analysing different combinations of variables enables the researcher to find new dependencies among variables in the dataset [Hautle, 2010].

The main challenge in displaying high dimensional data is trying to help to overcome the fact that our perceptual system is well functional with lower dimensional data, but cannot efficiently cope with data beyond the second or, if properly presented, third dimension. The aim of multivariate data visualisations

is to try to overcome this "impassable barrier" [Bertin, 1983] by employing various techniques [Siirtola, 2007].

Grinstein et al. [2001, p:28] defines more than twenty visualisation types that deal with high dimensional data and concludes:

"There is no one visualization that is best for high dimensional data exploration or data mining. Some visualizations are better for showing clusters or outliers, other visualizations can show two and three way correlations. Some visualizations can handle more data records than others. Most high dimensional visualizations require some training and experience by a user. We believe a data mining package should contain an arsenal of visualizations with appropriate tutorials for various datasets. "

According to Grinstein et al. [2001] the most relevant visualisations for this project have been selected and will be further analysed in the following sections.

2.1.1 Scatterplots

Grinstein et al. [2001, p:4] describe the scatter plot as:

"probably the most popular data mining visualization tool. It helps find clusters, outliers, trends and correlations."

Adding functionalities like brushing coloured points, zooming, panning, and jittering to the scatter plots can make them capable of efficiently visualising very large datasets. High dimensional datasets can be displayed in scatter plots using two main techniques. On the one hand, the points' size, shape, colour, saturation and glyph can be modified to include multiple variables. On the other hand, a two-dimensional scatter plot matrix can be used to represent the correlation among the variables by plotting each variable against each other. The first approach rarely yields as good results as the standard two-dimensional plot because the effect of every added variable has less impact on the whole visualisation than the X and Y axes. The second approach on the other hand is limited by the quantity and size of the multiple plots produced since the display quickly gets too

full. User controls for panning and zooming become in such cases indispensable [Grinstein et al., 2001].

2.1.2 Line graphs and multiple line graphs

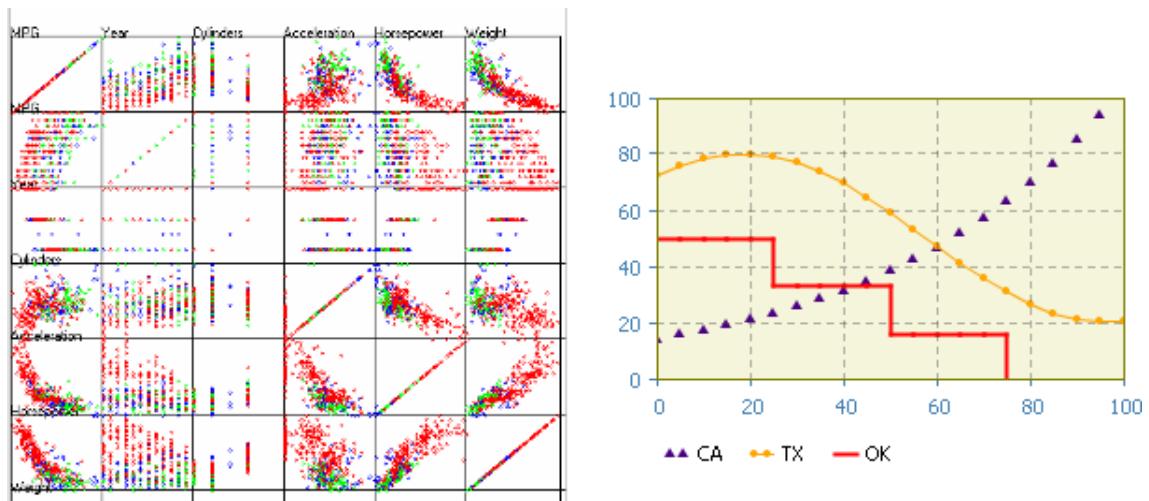


Figure 2.1: Scatter plot matrix and Time plot [sources: Grinstein et al. [2001]; <http://www.xjtek.com>]

According to Grinstein et al. [2001], multiple line graphs are used in multivariate analysis to display data that is dependent on one variable. Usually, the values of this variable are unique like (ID values or time-stamps). Despite the immediate readability of such visualisations, three main problems have been identified when working with many dimensions and need considerations when designing such tools. First, each variable needs to be displayed using a line with a different style. Second, the value scale of each variable needs to be present which might lead to confusion depending on the scale and whether or not an offset is used. If the absolute values are not relevant, this problem can easily be overcome by using data normalisation.

2.1.3 Parallel Coordinates Plots

Inselberg [1985] first described the plane with parallel coordinates, in his work he

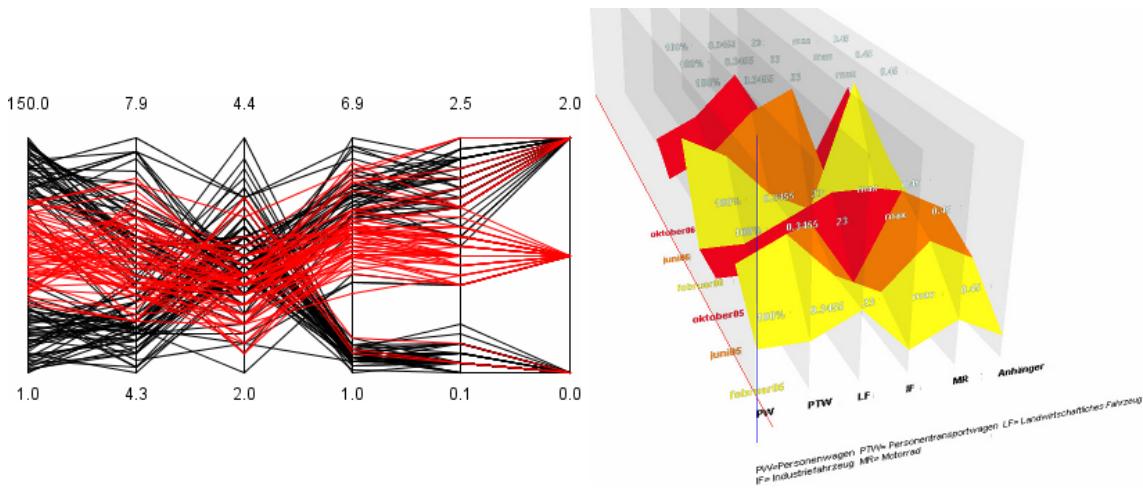


Figure 2.2: Parallel coordinates plot and Time extended parallel coordinates plot [source: Hautle [2010]]

presented a novel mathematical visualisations where high dimensional data can be plotted. Each N-dimensional point is no longer represented by a point but by a line. Each of the vertical parallel axes is the scale of a variable and where a line reaches it, it indicates the value of the sample for the specific variable. Twenty-five years after their formal introduction, PCPs have reached a well respected position among the multivariate visualisation and keep being a reference a base for novel visualisations [Hautle, 2010; Inselberg, 2009].

2.2 Multivariate temporal data visualisations

Temporal data visualisation is a strongly investigated topic in visualisation design. Many authors continuously develop and publish new approaches on the subject. So many different approaches have been published that different taxonomies, to help classify temporal visualisation methods, had to be created [Muller and Schumann, 2003; Daassi et al., 2006]. Not all of these methods are suitable for the analysis of multivariate temporal data or of interest for this project and therefore the selection of methods presented here is much more limited.

In this thesis temporal data is subdivided into two main groups, *linear* and *cyclical* data. We will consider linear data, datasets where cyclicity might be

present but is of no interest to the observer like in historic data where the sequence of the events is more important than the periodicity¹. Cyclical data represents for example most of the data produced by an environmental monitoring and modelling and are deeply connected to the cyclic nature of natural events.

2.2.1 Time extended parallel coordinates plot

A further interesting technique is the use of a time extended parallel coordinates plot [Hautle, 2010]. This technique extends the original parallel coordinate plot by implementing time in an extruded parallel coordinates plot [Wegenkittl et al., 1997]. Such a plot works by displaying the variables on a two dimensional plane instead of the one dimensional line used in normal PCPs. This approach facilitates the representation of multiple variables and their temporal component on a three dimensional surface. Hautle [2010] concludes her work stating that the system proved itself as reliable and effective. All the information contained in the data that usually are available in a PCP are conserved and displayed so that the user can look at his/her datasets from a novel perspective that might reveal new correlations.

2.2.2 Spiral visualisations

Weber et al. [2001] argue, that displaying temporal data on a linear scale is a poor use of the two-dimensional screen space. Instead they propose to display the data in a cyclic manner using spirals. Tominski and Schumann [2008] take the concept of visualising cyclical data in a cyclical manner further by applying an expressive two-tone pseudo colouring to allow users to read data values more precisely. Both Tominski and Schumann [2008] and Weber et al. [2001] add a degree of interactivity to their spirals arguing that due to the fact that periodicity is dependent on the dataset's temporal resolution and on the data itself, a user customizable spiral will allow a better pattern identifications in unknown datasets. A further development of the spiral visualisations is its implementation as a three

¹a typical example of such data is Minard's classical visualization of the fate of Napoleon's army on the campaign against Russia

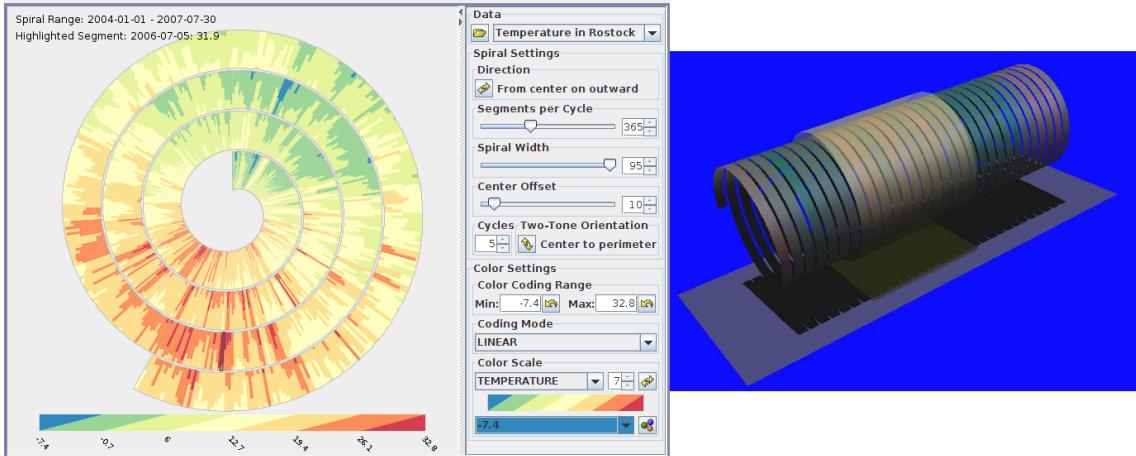


Figure 2.3: Enhanced Interactive Spiral Display and 3D Helix [sources: Tominski et al. [2004]; Weber et al. [2001]]

dimensional helix Weber et al. [2001]; Tominski et al. [2005] that allows direct interaction with the data. Weber et al. [2001] note, that the helix visualisation can be difficult to interpret when the helix is seen from the side since not all the values are directly visible.

2.2.3 Axes-Based Visualizations with Radial Layouts

Two further methods basing on radial axes are proposed in Tominski et al. [2004], the TimeWheel and the MultiComb [see figure 2.4]. The idea behind the TimeWheel is to present the time axis at the centre of the visualisation and to arrange the other variables' axes around it and to assign each axis a unique colour. The points on each of the axes are then connected to the time axes to show the relation between the axes similarly to PCP. The two parallel axes are the easiest ones to interpret and the two perpendicular ones the hardest, thus the TimeWheel allows an interactive rotating of the axes. Further techniques such as colour fading and axes length adjustment are used to avoid over-clogging the visualisation.

The idea behind the MultiComb is to align a series of plots in a circular form. Each of the plots (i.e. timeplot) are so comparable to each other by analysing the expressiveness of each plot. Both approaches have been extended to use the three dimensional space [see figure 2.5] in form of the 3D TimeWheel Aigner et al.

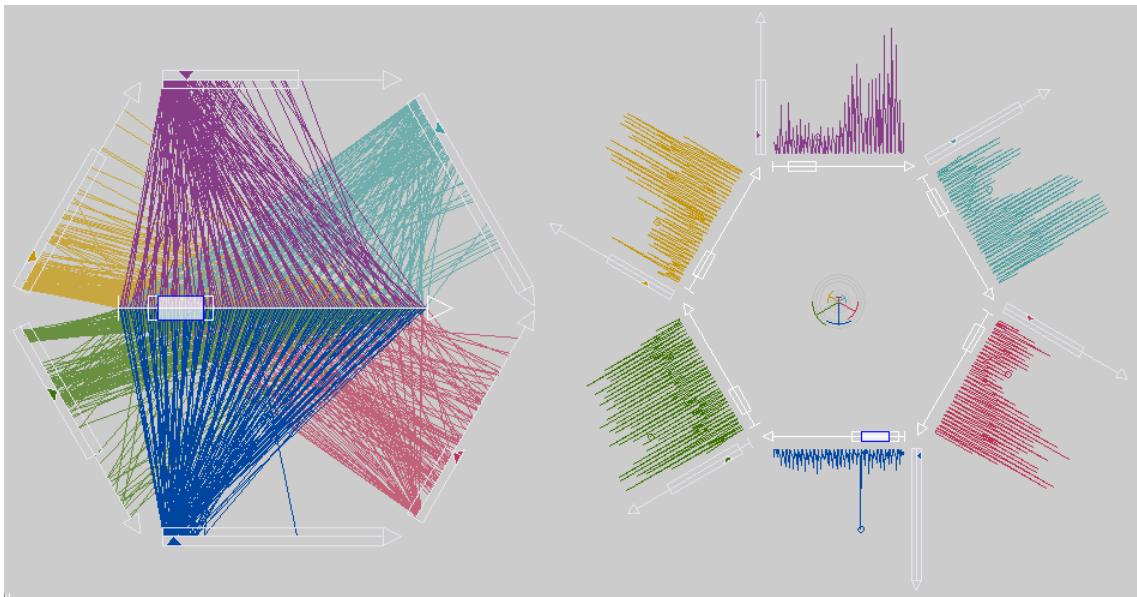


Figure 2.4: TimeWheel and MultiComb [source: Tominski et al. [2004]]

[2008] where the axes are all placed parallel to the time axis and the Stardiagramm Noirhomme-Fraiture [2002] where each plot is perpendicular to the time axis.

2.3 Spatio-temporal data visualisations

Spatio-temporal data can be separated in two main categories, movement data and static data. While in movement data the sensors move over time (such as in wildlife tracking), in static data the sensor do not move. This type of data is the typical outcome of environmental monitoring projects.

Several visualisations exist for movement data [Kjellin et al., 2008; Andrienko et al., 2003] including: maps with timestamped objects, Space-Time cube [Kraak, 2003], Multi-variable- and PCP-Time-Cube [Li and Kraak, 2005] as well as animation. All these techniques but animation are, due to their intrinsic connection to point data, useful only for movement visualisation and are therefore irrelevant to this project. Animation is an exception since it can be used to visualise both movement and static temporal data [Andrienko et al., 2003].

More relevant to this thesis are visualisations like pencil and helix glyphs

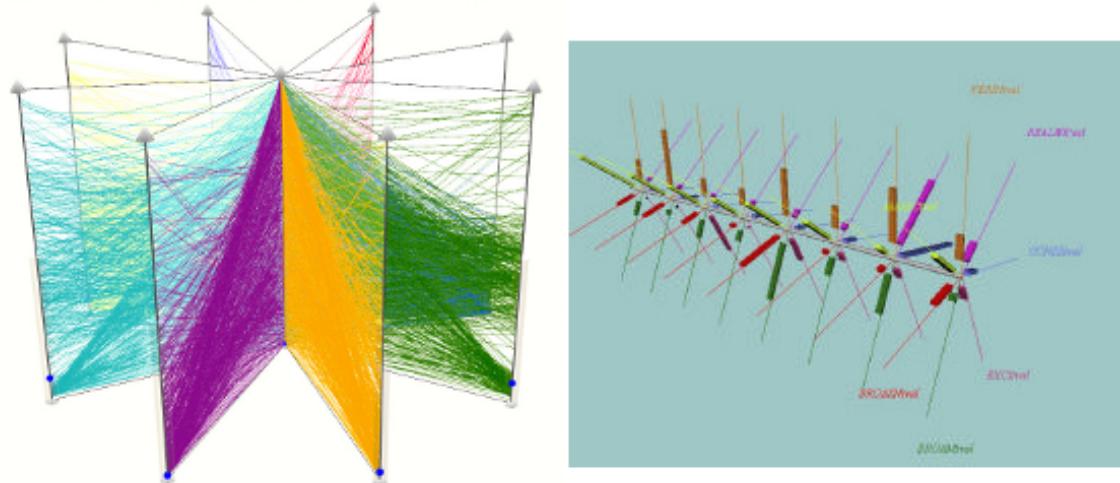


Figure 2.5: 3D TimeWheel and Stardiagramm [source: Aigner et al. [2008]; Noirhomme-Fraiture [2002]]

[Tominski et al., 2005] placed as objects on a map, change maps [Andrienko et al., 2003] or all the multivariate temporal data visualisations discussed in section 2.2 if used in a linked views context as explained in section 2.5. All these visualisations allow displaying changes over time for specific locations. On the one hand, multiple glyphs placed directly on a map as in Tominski et al. [2005], allow direct comparison of temporal data at different locations. On the other hand, the use of the tools described in section 2.2, in combination with a location input view, allows the exploration of the dataset of one location at a time. Further, even more complex visualisation tools that integrate data mining capabilities are being studied and developed [Compieta et al., 2007].

2.4 3D digital terrain model visualisations

The evolution of digital globes since the first released tools [Riedl, 2005] has been exponential. The release of tools such as Google Earth and NASA World Wind have made digital globes become increasingly popular. Nowadays, it has become a routine to check vacations destinations and personal whereabouts in such tools. When ten years ago explaining what GIS was to a layman involved long expla-

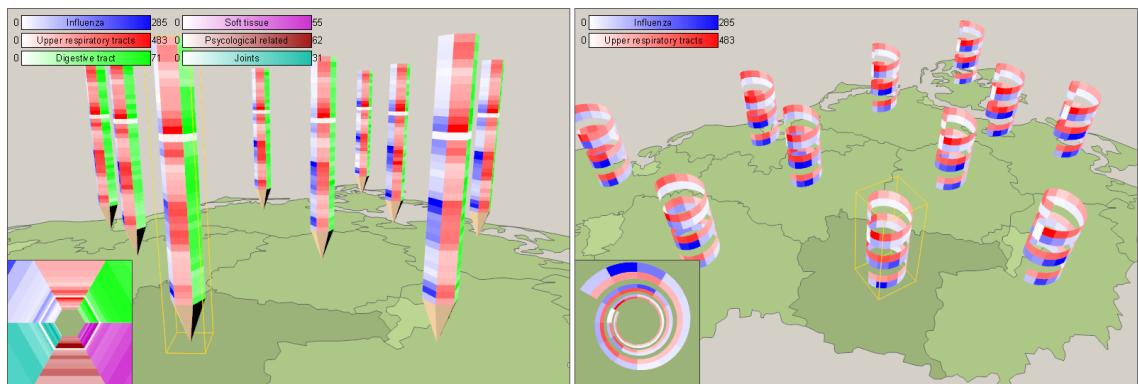


Figure 2.6: Pencil and helix glyphs places on a 3D map [source: Tominski et al. [2005]]

nations, it is now possible to give almost anybody in the western world a quick idea of what GIS is all about by mentioning one of these tools. A further, much more scientifically relevant point is the fact that many researchers started using these tools to display their research results [Bleisch and Nebiker, 2008; Butler, 2006]. Several studies have investigated the usefulness of 3D globes as part of linked view tools. The results are controversial. On the one hand researchers say that the 3D environment enhances the readability of the observed landscape and its features [Bleisch and Dykes, 2008; Jones et al., 2009]. On the other hand, the augmented cognitive load is cited as the main negative factor in navigating such environments [Rase, 2003]. In a study regarding 2D/3D integration, Bleisch and Nebiker [2008, p:1] conclude that

“It seems that connecting 2D data displays to the 3D views and updating or changing them dynamically allows overcoming some of the shortcomings of using stand-alone 3D views of information. This includes difficulties with navigation, occlusion of content, lower information density or projective distortion of the display. We assume, supported by combined 2D/3D displays that have already been tested as mentioned in the introduction and by our own experiences with the prototypal implementation described in this paper, that combining 2D displays with data representations in 3D virtual environments will certainly lead to new possibilities and ways for explorative analysis of spatial information. This may especially be true for data sets

that have a direct relation to the landscape, landform, the 3D shape of objects such as bridges or buildings or layers of the atmosphere."

Bleisch et al. [2009] further analyses the problem of visualisations over 3D globes by testing the usability of thematic map symbols in a prototypical study and concludes with a call for research on the subject.

Stereo 3D

Stereoscopic 3D visualisation is a very broad subject on which entire book chapters [eg. Kirschenbauer, 2005; Shepherd, 2008] and articles [eg. Wang et al., 2009; Cockburn and McKenzie, 2001, 2002] have been written. Due to the limited importance of this subject for this project and the controversial research results, a detailed explication is here skipped and the reader is directed to the literature above mentioned.

2.5 Linked views

When dealing with multivariate (spatio temporal) datasets, often one visualisation is not enough for efficiently exploring a dataset. To overcome this problem, the multiple views technique can be used. This technique consists in having multiple windows with different visualisations linked to each other usually by the mouse position. On mouse movement or selection, the data displayed on each visualisation is updated according to the current selection criteria [Roberts, 2005; Buja et al., 1991]. Views linking used in combination with brushing and the overview - filter – detail on demand paradigm [Shneiderman, 1996], can create very complex and powerful interfaces as for example the OECD explorer [OEC, 2011; Jern et al., 2007] or the HEALTH GeoJunction [geo, 2011b] [see figure 2.7].

The main concerns in using multiple linked windows are not how to technically solve their implementation but much more how to keep the GUI clean and when to use at all such paradigm. Kosara et al. [2003, p:5] state:

"Given today's graphical user interfaces, using several views on the same data instead of just one is trivial. Doing it right, and providing

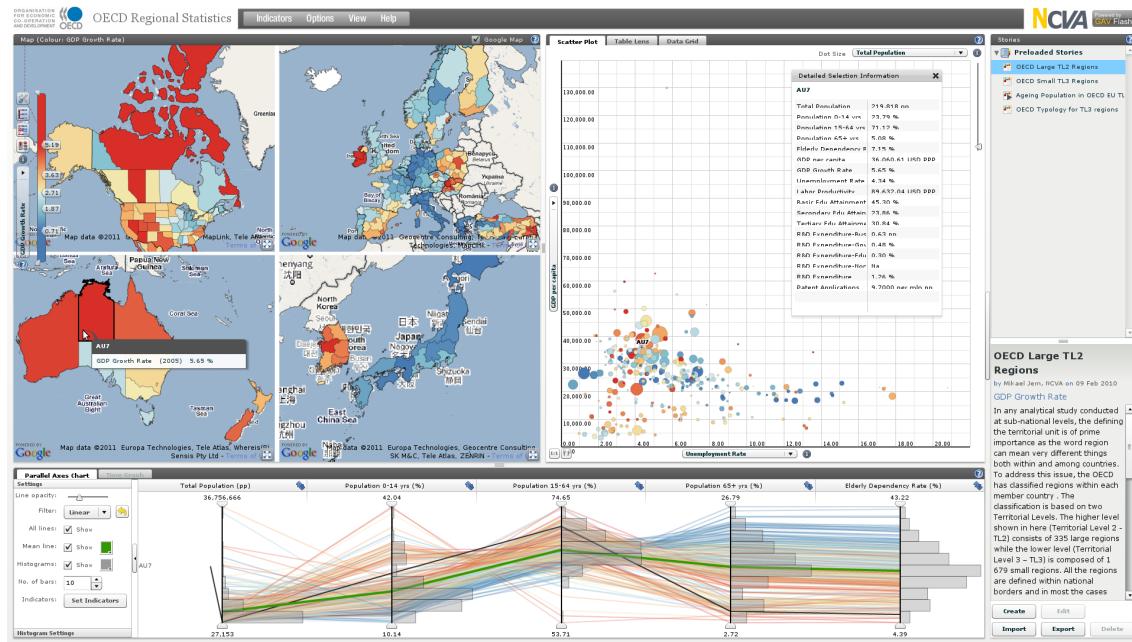


Figure 2.7: Linked views example [source: OEC [2011]]

the necessary amount of interaction, however, is not."

To help develop applications using multiple linked windows, Wang Baldonado et al. [2000] developed a set of eight guidelines that should allow a developer to assess if a multiple linked windows implementation is a sensible solution to his/her problem. Table 2.1 summarizes the guidelines giving a short description of each one and the main pros and cons of following or disregarding the indication.

2.6 Available tools

Many multivariate temporal data visualisation tools are available nowadays, and some of them deal with spatio temporal data as well. Most packages are focused towards visualisations and less toward mature GIS integration. The work-flow to use such tools requests therefore the separate usage of a GIS to prepare the data, an export/import step and finally the visualisation and analysis of the data with the chosen tool. Below a list can be found, added for reference only, of some

Rule	Summary	Major Positive Impacts on Utility	Major Negative Impacts on Utility
Diversity	<i>Use multiple views when there is a diversity of attributes, models, user profiles, levels of abstraction, or genres.</i>	memory	learning computational overhead display space overhead
Complementarity	<i>Use multiple views when different views bring out correlations and/or disparities.</i>	memory comparison context switching	learning computational overhead display space overhead
Decomposition	<i>Partition complex data into multiple views to create manageable chunks and to provide insight into the interaction among different dimensions.</i>	memory comparison	learning computational overhead display space overhead
Parsimony	<i>Use multiple views minimally.</i>	learning computational overhead display space overhead	memory comparison context switching
Space/Time Resource Optimization	<i>Balance the spatial and temporal costs of presenting multiple views with the spatial and temporal benefits of using the views.</i>	comparison computational overhead display space overhead	
Self-Evidence	<i>Use perceptual cues to make relationships among multiple views more apparent to the user.</i>	learning comparison	computational overhead
Consistency	<i>Make the interfaces for multiple views consistent, and make the states of multiple views consistent.</i>	learning comparison	computational overhead
Attention Management	<i>Use perceptual techniques to focus the user's attention on the right view at the right time.</i>	memory context switching	computational overhead

Table 2.1: Summary of rules and areas of major impact on utility [source: Wang Baldonado et al. [2000]]

available open source tools with a short description of the functionality as given per their product description.

Protovis [pro, 2011]

"is an extensible toolkit for constructing visualizations by composing simple graphical primitives. In Protovis, designers specify visualizations as a hierarchy of marks with visual properties defined as functions of data. This representation achieves a level of expressiveness comparable to low-level graphics systems, while improving efficiency - the effort required to specify a visualization - and accessibility - the effort required to learn and modify the representation." [Bostock and Heer, 2009]

GeoVISTA [geo, 2011a]

"The fundamental goal of the GeoVISTA Studio project is to improve geoscientific analysis by providing an environment that operationally integrates a wide range of analysis activities, including those both computationally and visually based." [geo, 2011a]

"GeoVISTA studio is a component-oriented system, focused on: ease of program construction (visual programming), an open (non-proprietary) architecture, simple component-based integration." [Takatsuka and Gahagan, 2002].

VIS-STAMP [vis, 2011]

"VIS-STAMP is a geovisual analytic software package that couples computational, visual, and cartographic methods for exploring and understanding spatiotemporal and multivariate data. It can help analysts investigate complex patterns across multivariate, spatial, and temporal dimensions via clustering, sorting, and visualization. Specifically, VIS-STAMP builds on a self-organizing map, a parallel coordinate plot, several forms of reorderable matrices, and a geographic small multiple display." [vis, 2011]

GAV Flash tools [GAV, 2011]

"The Geovisual Analytics Visualization or GAV component sharing toolkit and application building environment is based on the principles behind the Visual Analytics research program including aspects of explorative and collaborative visual data analysis. It contains a collection of visual components, data analysis algorithms, tools that connect the components to each other and data providers that can load data from various sources. The framework is fully integrated with Adobe's Flex Framework." [GAV, 2011]

JavaScript InfoVis Toolkit [jsi, 2011]

"The JavaScript InfoVis Toolkit provides tools for creating Interactive Data Visualizations for the Web. [jsi, 2011]"

3

Design

This chapter explains the design process that resulted during the implementation of the tool as it is now. After an introduction about the used design methodology, section 3.1 will illustrate the outcome of the first brainstorming sessions with my two advisers while planning what was needed. Section 3.2 will then depict the results of the focus-group session held after two months the project had started. Finally, in section 3.3 I will present the formal requirements for the developed tools based on the theoretical insight gained from the literature research and from the focus-group inputs.

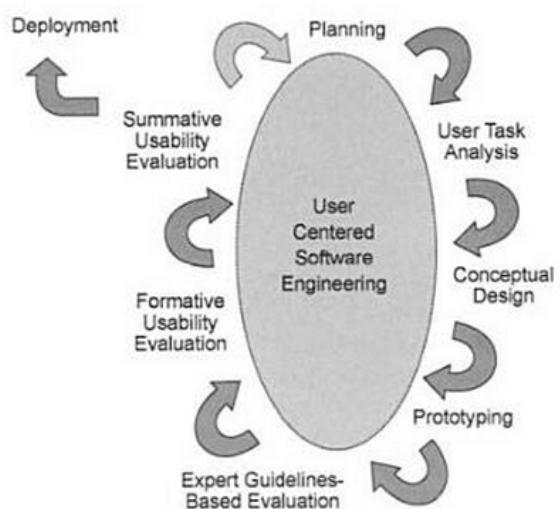


Figure 3.1: User centred design methodology after Torres [2002]

The main design priority was to create a tool that would not only help answer the research questions but would be useful for the researchers in dealing with their specific use cases. Trying to strive as much as possible towards user satisfaction, it was decided to design the tool using a user-centred development methodology [Torres, 2002]. Following this development mantra [figure 3.1] the development process is separated in seven iterative steps [Fuhrmann and Pike, 2005]:

- Planning: timetables, milestones and deliverables are planned
- User task analysis: the real users' needs are analysed
- Conceptual design: software features needed are identified and formal requirements written
- Prototyping: iterative implementation of the prototype
- Expert guideline-based evaluation: use expert heuristics to identify potential problems
- Formative usability evaluation: Task based expert evaluation on the prototype and subsequent iterative ameliorations of the prototype
- Summative usability evaluation: broader standardized user tests conducted when a prototype has reached a certain maturity

A last step in user-centred development is the final deployment of the software to a broad user base. In this project, the agile development methodology [see section 4.1.2] was used along the user centred design, therefore the working code (the one of the prototype as well) was deployed and publicly released as early as possible and thus not only after evaluation.

3.1 Requirements from user tasks analysis

The main tasks that need to be performed by a scientist working with geoTOP model and sensor data are the following:

- Search for temporal patterns at a given location in cyclic data
- Search for spatial patterns at a given time
- Search for outliers and data anomalies
- Interpret data clusters in a scattergram and find correlation to space and time

Based on literature research [see chapter 2] and on multiple brainstorming sessions the following visualisations were considered as appropriate for the needed tasks and were submitted to the focus group for evaluation: dataview, scatterplot, Parallel Coordinates Plot [Inselberg, 2009], Multivariable-Time cube¹ [Li and Kraak, 2005], PCP-Time-Cube¹ [Li and Kraak, 2005], pencil glyphs [Tominski et al., 2005], helix glyphs [Tominski et al., 2005] and stereo 3D terrain.

3.2 Focus Group

The rationale behind the choice of submitting the requirements from user task analysis to an expert users focus group can be best explained by quoting Lindlof and Taylor [2002, p:183]:

"The most compelling reason for using the focus group method is to exploit the "group effect" (Carey, 1994). The method takes advantages of the fact that in both ordinary conversations and guided discussions, people draw upon a shared fund of experiences. What occurs in this context is a kind of "changing" or "cascading" effect in which each person's turn of the conversation links to, or tumbles out of, the topics and expressions that came before it. As Morgan (1988) explains, "the explicit use of the group interaction [produces] data and insights that would be less accessible without the interaction found in a group" (p. 12). This dynamic seldom, if ever, arises in one-on-one interview."

¹With hindsight, it was a mistake to submit the MTC and PTC to the focus group since the project has no movement data.

The focus group was modelled according to the suggestions made in Courage and Baxter [2005] and Monmonier [1994] obtaining the following protocol:

- Brief project introduction by outlining its aims and requirements.
- Participant introduction: everyone was asked to shortly introduce him/herself and to address these points:
 - Who are you?
 - What do you do in your work?
 - How exposed are you to multivariate data and/or the GEOTop project
 - What are your likes and dislikes in your current way of dealing with such data?
- Presentation of the eight chosen visualisations [see section 3.1] and discussion under the motto “All ideas are correct”
- Interface mock-ups design: the participants received a set of mini visualisations print-outs and were asked to create a GUI mock-up by gluing the visualisations on a A4 sheet.
- Thanks and refreshments

The Focus group turned out to be very successful and productive. The five experts that participated delivered an incredibly high amount of information using heuristics. A detailed review of the results obtained from the focus group is skipped here since the inputs were directly used to define the formal requirements of the project as found in the next section. The GUI mock-ups produced by the participants can be seen in appendix A.6.

3.3 Formal requirements

The goal of the thesis to implement a beta quality application as a Quantum Gis Plugin (Python and C++). The application should have multiple views (independent modules) with brushing and linking (select a location in one view and

highlight in all views [Rogowitz and Treinish, 1996]). This architecture will allow for a flexible distinction between “must-have” and “nice-to-have” features and therefore for a smoother time management. The application interface will be built based on the Visual Information Seeking Mantra “Overview - zoom and filter - details on demand” [Shneiderman, 1996] implementing standard UI controls (zoom, pan, rotate, tilt, reset), layer filters and time sliders. Furthermore, the application will take advantage of the possibility offered by linking a 2D map canvas and visualisations with a 3D module [Bleisch and Nebiker, 2008].

3.3.1 Must-have

- A globe-like terrain model module with stereo 3D capabilities. The stereo capabilities should at least support the anaglyph mode. Although this stereo mode does not yield the best results possible [Volbracht et al., 1997], it has the advantage of running on a wide range of hardware and of being virtually cost free.
- A visualisation framework with at least the following visualisations:
 - A time vs. value plot should allow comparing the trends of multiple variables over time for the selected location.
 - A 3D Helix visualisation should be implemented to provide the details on demand in the analysis of cyclic data [Tominski et al., 2005]. Differently from the original implementation where multiple helices were placed on a vectorial map to indicate the data of a region, the tool should implement only one helix that shows the values of the selected pixel on the globe.
- A scattergram tool with a functionality to plot the values on the globe.

3.3.2 Nice-to-have

- A temporal resolution selector for the visualisation framework. This should allow a quick switch between different temporal resolutions of the same

variables.

- A Time manager tool that allows filtering the data on a certain time period and animating the raster layers.
- A multivariate selector to work in combination with the time manager to allow filtering the data based on the variables.

3.4 Evaluation

This project only treats the first six steps of the user centred design methodology since adding a summative usability evaluation at the end of the project would have requested much more time resources that are unavailable during a master thesis. To test our approach we will perform qualitative expert interviews based on SEE metrics investigating user performance and satisfaction [Fuhrmann and Pike, 2005; Tobon, 2005]. Details regarding the interview will depend upon the final implementation of the tool.

4

Implementation

This chapter will explain how I implemented the features planned in chapter 3. The first section of the chapter will be dedicated to the explanation of the programming principles I adhered to. In section 4.2 I will shortly explain the function and position in the system architecture and interaction flow of each framework and library used. The last sections will then be dedicated to a detailed description of how I implemented each of the tools developed for this thesis.

4.1 Development strategy

This project has been developed following modern development models and using industry standard open source code management and versioning systems. Furthermore, a strong accent was put to the use of coding standards.

4.1.1 Software used

The whole development and testing process took place in Ubuntu Linux 10.10¹ using exclusively open source software and tools. For the Qt-C++ development and GUI design a mix of the Qtcreator² IDE and Geany³, a lightweight IDE, has

¹<http://www.ubuntu.com>

²<http://qt.nokia.com/products/developer-tools/>

³<http://www.geany.org/>

been used. For the python development a mix of Aptana⁴ and a customised version of the lightweight texteditor Gedit⁵ where successfully alternated depending on the machine it was being developed on. The thesis itself is written using open office for typing, L^AT_EX⁶ for typesetting, Bibtex⁷ for literature management and is stored in an SVN⁸ repository.

4.1.2 Agile software development

The development method used for all the programming work performed for this thesis leans strongly toward the Agile development process [Agi, 2011]. Although the Agile programming methodology [see figure 4.1] is mainly tailored to work for small teams, some key points of this process can be applied to this project as well. The main ones used in this project are:

- Stakeholder satisfaction
- Simplicity is essential
- Welcome change
- Early and frequent software delivery
- Working software as measure of productivity

Beside these principles, the Agile software development process suggests many techniques to improve development quality and efficiency from which I mainly used three. The first is the creation of user stories early in the planning phase of the project. This technique consists in asking the stakeholders to describe the typical task they would need to do with the software. This allows the identification of the real stakeholders needs and can easily be converted into more formal requirements. The second technique I used is the project management through

⁴<http://www.aptana.com/>

⁵<http://live.gnome.org/Gedit>

⁶<http://www.latex-project.org/>

⁷<http://www.bibtex.org/>

⁸<http://subversion.apache.org/>

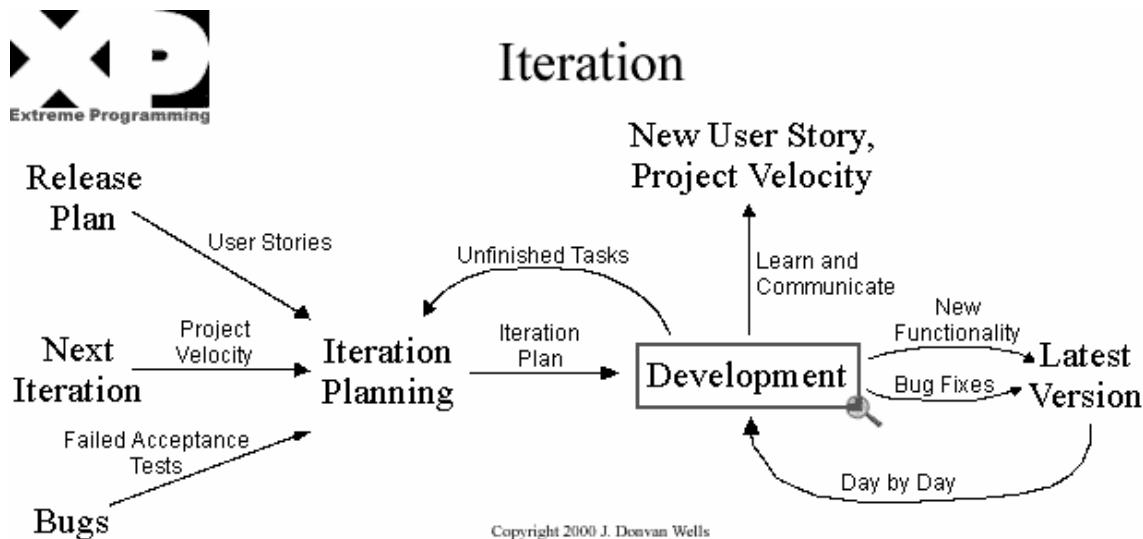


Figure 4.1: Agile iteration workflow [source: <http://extremeprogramming.org>]

feature. This technique consists in focusing on the features that need to be implemented in order of importance rather than on the project as a whole and its fixed goals. This technique is directly related to the Agile wish of welcoming early and frequent change during the whole development of a project. The third technique is iterative planning, this technique allowed me to adapt and implement quickly new idea that were brought into the team either during a demo, by new literature, by external feedbacks and partly after the expert interview.

The agile approach to development allowed me to easily integrate the continuously evolving nature of such a complex project, furthermore the frequent delivery of working code made the structure of the whole project truly modular.

4.1.3 Code management with GIT

GIT⁹ is an open source distributed version control system [DVCS] used by many multi-developer projects¹⁰. Its strengths lie in speed and efficiency of the code-base management independently from the project size. The distributed¹¹ nature of GIT makes it ideal for network independent developing since every clone is a

⁹<http://git-scm.com/>

¹⁰Projects using GIT include: the Linux kernel, Qt, Gnome, KDE, Android, PostgreSQL, ...

¹¹For further information on the advantages of DVCS see de Alwis and Sillito [2009]

fully fledged repository and needs no central server.

All code produced for this project is managed in various GIT repositories. The choice of GIT over other version control system¹² was based on three main personal requirements.

Network independence: I move a lot and I'm often working far from my desk so I needed a solution that would allow me to pick up a computer from my desk, leave and know that I would need no network to be able to access the main repository. This point made me decide for DVCS over CVCS.

Collaboration: I wanted a system that would easily deal with branching and merging the code-base and that would easily allow sharing the work with others and accept others' contributions. One of the key decision factor here was the outstanding quality of GitHub¹³ and all the social coding possibilities that it enables.

Integration into QGis developing standards: Since the start of the project, QGis is maintained in an SVN repository. When I started looking into how to develop my thesis, the QGis developer also started discussing the migration to a new development infrastructure based on GIT. To avoid a future migration I decided to start from the beginning using the GIT on GitHub to then easily move to the new GIT based QGis infrastructure as soon as it was usable. Meanwhile, the infrastructure¹⁴ is well grown and it will be publicly advertised after the next release of QGis¹⁵. All Python plugins reside now on this infrastructure while globe is still being developed on GitHub since an installer script¹⁶ references the GitHub repository.

The main advantage of developing code using a (D)VCS was for me the ability of experimenting on a new feature knowing that I will not break anything and that the worst case scenario would have been reverting all the changed files back to a working state. For this reason I followed the frequent commits paradigm to minimize non working code being developed.

¹²The main alternative would have been SVN as it is used to manage the thesis documents

¹³<http://github.com/>

¹⁴<http://hub.qgis.org>, <http://plugins.qgis.org>

¹⁵QGis 1.7 Wrocław is planed to release on 25.4.2011

¹⁶<http://www.bernawebdesign.ch/byteblog/2010/12/01/qgis-globe-plugin-installer-script/>

4.2 System architecture

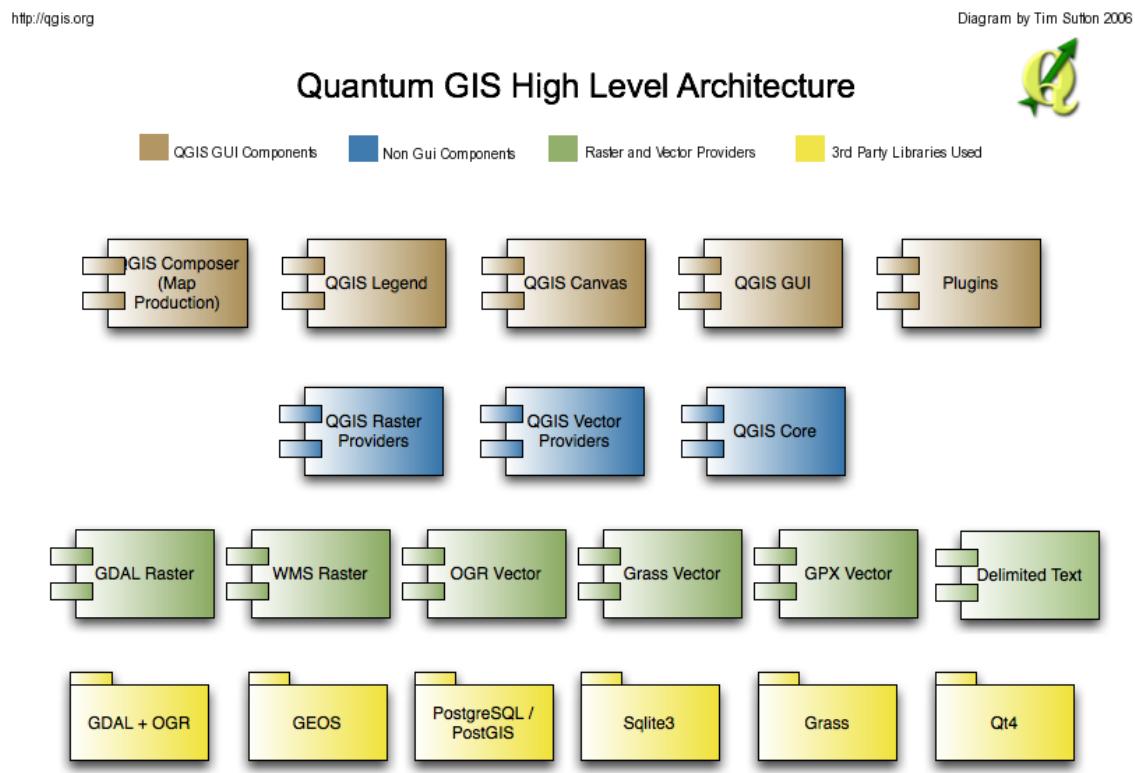


Figure 4.2: QGis high level archiutecture [source: <http://www.qgis.org>]

QGis leverages many libraries and frameworks to achieve its full functionality. In this section I will first explain the high level QGis architecture and then I will explain for what each component is necessary and how it relates to the others.

As figure 4.2 clearly shows, QGis is build of multiple modules that deliver specific functionality to the whole system. Five main GUI components manage the whole application: Map composer, Legend, Canvas, GUI and plugins. Map composer is responsible for the creation and printing of maps, the legend and canvas components are used to maintain the state of the data while the GUI manages the layout of the application. Finally a vast ecosystem of plugins contributes to QGis flexibility by allowing everyone to program custom plugins to suit their needs.

The main GUI components are independent and do not directly provide GIS

capabilities. All the actual GIS work is performed by the non GUI components. Thanks to this approach, QGis can be used as a library to build other applications or to add GIS capabilities to an already existing application.

The non GUI components are: QGis raster providers, QGis Vector providers and QGis core. The raster providers and vector providers are a series of classes and libraries binding that allow QGis to deal with raster and vector data of different origins and format. These components function as an abstraction layer between the data and the application, thus allowing the use of a unique API for all data types. The QGis core is the GIS library at the heart of QGis performing all the GIS work tasks.

4.2.1 QT framework

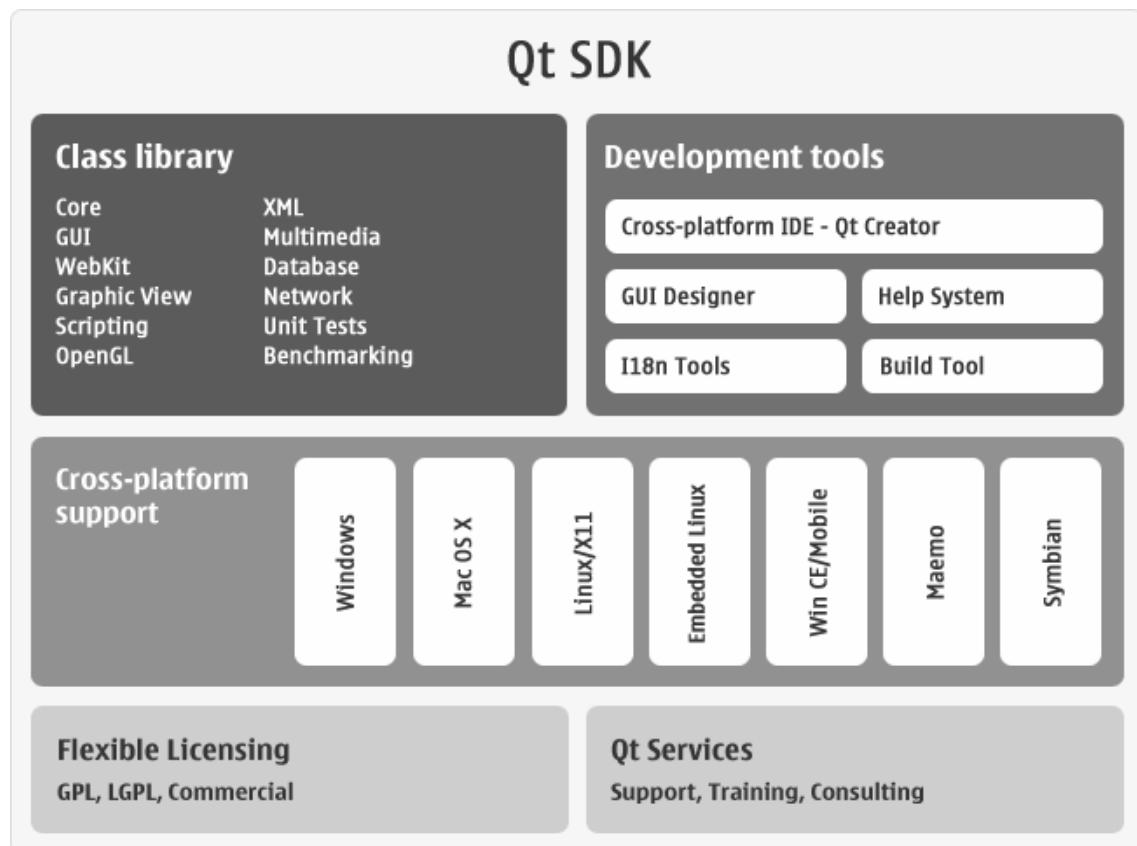


Figure 4.3: Qt framework [source: <http://qt.nokia.com>]

The QT framework is a cross-platform application and UI framework that includes a cross-platform class library, integrated development tools and a cross-platform IDE¹⁷. The use of Qt allows QGis to be platform independent and to allow the developers to focus on GIS functionality and not to have to deal with repetitive tasks such as file management, windowing and infrastructure coding.

Qt has native support for GUI design, internationalisation, help system and building [see figure 4.3]. Furthermore, the coherent and intuitive class design makes it easier for the developer to implement the features he needs. Furthermore, the developer will benefit from bindings to the Qt library in several different programming languages such as PyQt for Python. A very interesting mech-

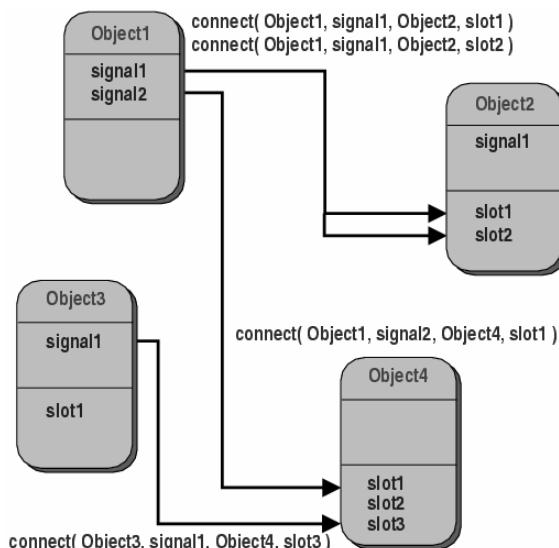


Figure 4.4: Qt signal/slot framework [source: <http://qt.nokia.com>]

anism available in the Qt framework and widely used in QGis is the signal/slot mechanism. This paradigm allows a loosely coupled communication between objects that is type safe but which is still easy to create re-usable software components. The basic idea [see figure 4.4] is that each object can emit a signal when a certain event happens without needing to know if any other object will be listening to the signal. If an object is listening to the signal, then the connected slot (method) will be executed obtaining the same effects as callbacks but

¹⁷source: <http://qt.nokia.com/>

by maintaining the two objects completely unaware of each other¹⁸.

4.2.2 PyQt framework

PyQt framework is a set of python bindings for the Qt framework produced by riverbank computing ltd.¹⁹. These bindings allow to trigger most of the Qt classes and functionality from within Python. PyQt is used in QGis to expose the C++ API to the Python plugin infrastructure.

Thanks to the flexibility and simplicity of Python and the many functionalities of Qt, PyQt allows the developer to quickly and easily produce high quality code specially when used in combination with the Agile programming methodology [PyQ, 2011].

4.2.3 QGis API

The Qgis API is the main access point for plugins to the core QGis library. In appendix A.1 a high level UML schema of the whole API illustrates the complexity of the API. This API offers the developer all the tools needed to deal with geographic datasets and operations. A detailed discussion of the API would be out of scope here and therefore skipped. Further information can be obtained from the QGis API documentation ²⁰ and in the developer wiki ²¹

4.2.4 QGis plugin architecture

QGis Plugins are modular components that can be added to the main program by the user depending on its necessities. When no available plugin offers the needed functionality, a developer can create a plugin that suits fills the needs and, if desired publish it for others to use. The plugins can be programmed using C++ or Python. C++ is mostly suited for core plugins that need to leverage other

¹⁸further information about the signal/slot mechanism can be found at <http://doc.qt.nokia.com/latest/signalsandslots.html>

¹⁹<http://www.riverbankcomputing.com/software/pyqt/intro>

²⁰<http://doc.qgis.org/>

²¹<http://www.qgis.org/wiki/Development>

C++ libraries and it is not the recommended way of creating new plugins since distributing them involves creating new QGis installers or requiring the users to compile their own QGis version from sources. Globe plugin was written in C++ because it is expected to become part of QGis just after the release of QGis 1.7. The Python bindings allow on the other hand a much quicker development and release cycle since Python is an interpreted language and does not require compilation. Distributing a Python is as simple as uploading a zipped version to a central plugin repository via a web interface. Every QGis user will then receive a notification of the new plugin from the QGis plugin manager. The introduction of the Python plugins infrastructure made the number of user contributed plugins increase significantly mostly due to the much simpler nature of Python and the quicker development and distribution.

4.3 Globe C++ plugin

The Globe plugin²² is the only C++ plugin implemented for this project. The development of the plugin was the result of a collaboration with Pirmin Kalberer from sourcepole.ch that by the time I started my project had already implemented the basics of the globe QGis plugin. His initial version was capable of displaying the globe in a QWidget and could overlay the QGis layers on the globe. Anyhow, the plugin lacked a GUI, Stereo support, extent synchronization and its configuration was based on a configuration XML file parsed at start up. The target strived was to implement these missing features during the duration of the thesis.

4.3.1 Architecture

The plugin bases on the OSGEarth²³ rendering tool-kit which on its own builds on top of the OpenSceneGraph²⁴ [OSG] graphics toolkit. The OSG toolkit uses C++ and OpenGL to create high performance 3D renderings. Figure 4.6 shows

²²The full code base can be found at <https://github.com/mbernascocchi/qgis/tree/threading-globe/src/plugins/globe>

²³<http://osgearth.org/>

²⁴<http://www.openscenegraph.org>



Figure 4.5: Evolution of the Globe plugin. Basic globe, globe with GUI, globe with GUI and stereo 3D

the high level architecture of the plugin highlighting where adapter classes had to be programmed to allow the OSGEarth globe to communicate with QGis. The used architecture bases on the wish to have the globe running as a Qt widget directly inside QGis and not to call it as an external program. This approach allows us leverage the QGis rendering engine for the globe as well which results in the same layer symbology as on map canvas being used automatically. The main drawback of this approach was the relative difficulty of implementing the adapter layers, but once this was done, the benefits were worth the effort.

As it can be seen in figure A.2 all the features listed early have been implemented in two main groups, on one hand on one hand all the ControlHandlers structs²⁵ take care of the on screen GUI controls that allow zooming, panning, tilting, rotating and syncing of the globe [see figure 4.5]. The control handlers all extend either the class NavigationControlHandler or ControlEventHandler from OSGEarth to take advantage of the built-in capabilities of the library. On the other hand, the QgsGlobePluginDialog deals with the runtime configuration of the plugin allowing the user to add or remove DEMs of various sources, formats²⁶ and resolutions, set stereo 3D settings, and as a still experimental feature, add 3D models for objects. The OSGEarth toolkit takes internally care of the generation of the terrain model, the textures and, if activated, the stereo pairs according to the user settings [see figure 4.7]. All stereo settings are saved in QGis settings and

²⁵in C++ a struct is the same as a class except that its members are public by default

²⁶<http://osgearth.org/wiki/TileSourcePlugins>

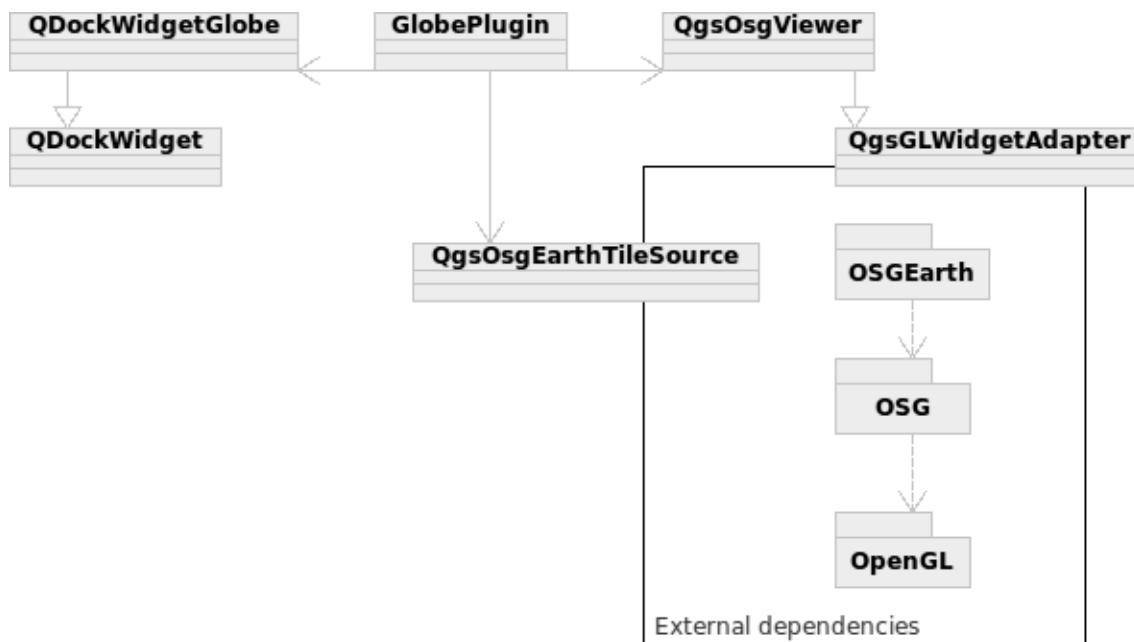


Figure 4.6: Globe plugin dependencies and high level architecture [Own illustration]

they are applied to each opened subsequently opened project. This was decided so that a user would not have to reset the globe stereo settings for each project he was working on. On the other hand, the elevation and models settings are saved on a project basis so that the project files are self contained units and can we easily shared.

At beginning of March, Globe has been updated to use the newly developed OSGEarth 2.0 toolkit making globe much stabler and with much better performance than with version 1.4. The main implementation is therefore completed but some bugs still require further work [see section 8.1] to see Globe as a standard part of QGis 1.8.

4.4 MultiView Python plugin

The MultiView plugin²⁷ is a Python plugin that allows the import and visualisation of multivariate spatio-temporal data into QGis. The main concerns during

²⁷The full code base can be found at <http://hub.qgis.org/projects/multiview/repository>

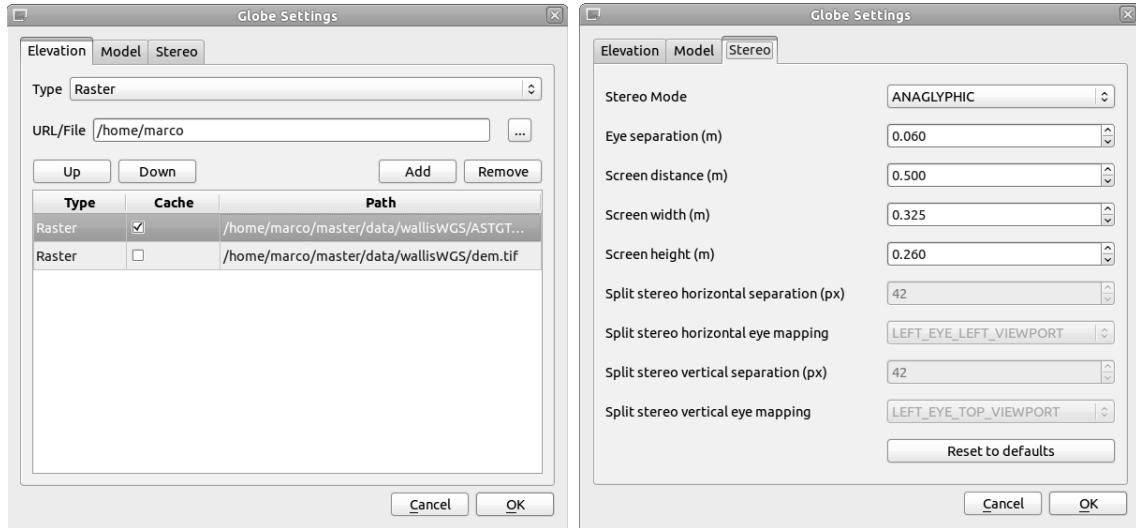


Figure 4.7: Globe plugin's stereo 3D and DEM settings dialogs

the design phase were to design it as a flexible framework for spatio-temporal data visualisations and to try to reduce its complexity to a minimum.

4.4.1 Architecture

MultiView consists of a main class, a main widget, a data loader, an extensible system of dockable visualisations and some support classes [see figure 4.8 and appendix A.3]. The main class (MultiView) is instantiated by QGis automatically

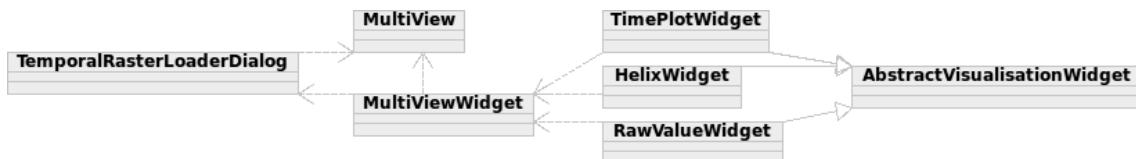


Figure 4.8: MultiView architecture overview [Own illustration]

on start-up and is used only for few functions. Its main role is to listen to the QGis toolbar signals and to start the correct widget. The other important function performed by this class is the management of the interval length for each variable loaded in the project [see listing 4.1] and to save the values to the project file. This approach had to be used because the time intervals are created by the data loader and need to be available to the main widget as well.

Listing 4.1: MultiView timestep duration example array

```
{
  'S0003': 86400,
  'S0004': 14400
}
```

The data loader is an almost standalone²⁸ data importer that takes a list of files with some temporal information encoded in the file name and creates layers in QGis with custom properties that indicate time, interval step number and length of any given layer. The data loader has a maximum resolution of one second.

The main widget is where all the preparatory work is performed. This class takes care of creating the user interface on plugin start up and then manages all user input. Once the class received the needed input, it reads the data values and prepares a multidimensional array to pass on to the currently active visualisation. This approach allows a complete separation of the visualisations widgets from the main application. The result is a fully extensible visualisation system. On each activated variables change, the main widget recalculates the lowest date of all active variables and stores this as a variable called timeMin. The array [see listing 4.2] delivered to the visualisations is an array of arrays (one for each activated variable) containing lists of seconds from timeMin and value pairs.

Listing 4.2: MultiView multidimensional example result array

```
{
  'snowdepthS0003': [(0, 0.0), (86400, 5.5), (172800, 0.0), (259200, 0.0), (345600, 7.0)],
  'snowdepthS0004': [(0, 0.0), (14400, 0.0), (28800, 0.0), (43200, 0.0), (57600, 0.0)],
  'tempL0001S0003': [(0, 1.8), (86400, 0.6), (172800, 0.7), (259200, 8.8), (345600, 3.0)],
  'tempL0001S0004': [(0, 2.7), (14400, 1.7), (28800, 9.5), (43200, 21.2), (57600, 18.2)]
}
```

Each of the visualisation implemented in MultiView is an independent widget that needs to implement the AbstractVisualisationWidget class. This class assures that the visualisation widget implements the required methods to properly run in MultiView. Currently, the inclusion of a further visualisation requests only to alter two lines of code in the main widget's code. This could be reduced to no

²⁸Converting the data loader to a stand alone plugin would require very little work, but for distribution simplicity the two pieces of code are maintained together. It might be possible that future plugins require a temporal data-loader, as it might be the case with TimeManager plugin, and then the loader could be separated from MultiView

modifications needed by creating a scanning system that would include all available visualisations, however this feature has been noted as marginally important and thus not implemented.

4.4.2 Rawvalue widget

The raw value widget is the most basic visualisation possible realised for Multi-View, where the result array is displayed in text form [see figure A.5]. This widget has a triple functionality, first of all it was the first widget to be implemented and thus it defined the base requirements and signal infrastructure for the further development, second it can be used to export the values for further analysis by simply copy-pasting and third it is a simple example of how the framework works for future developers that might want to implement a further visualisation widget.

4.4.3 TimePlot widget

The TimePlot widget [see figure A.6] is a time vs value plot. The X axis of the plot indicates the time of a measurement while on the Y axis the value of the current selected location is displayed. The plot can show an unlimited amount of time steps and of variables using different colours per variable and allowing zooming into the data. The TimePlot plot is a QwtPlot object that implements a custom QwtScaleDraw class on its X axis. This class (TimeScaleDraw) is responsible for setting the labels and the intervals depending on the data received. The TimeScaleDraw class converts the received seconds values into a time-date object and displays it as a label. The class instantiates as well a QwtScaleDiv object to only show labels where there is a time value, obtaining so a regularly distributed labelling. The same approach is used for the TimeScalePicker, a specialized QwtScalePicker class that displays the values under the cursor when moving over the plot.

Finally, various Qt classes and widgets are used to allow changing the appearance of all the lines. It is planned but not implemented yet to allow individual line styling to improve the readability of the plot.

4.4.4 HelixView widget

The HelixView widget is based on a port and adaptation of the LandVis²⁹ code by Tominski et al. [2005] and it is implemented using Python, PyQGLViewer and the Python OpenGL module. This combination of technology layers made the programming of the helix one of the most complicated tasks of the whole project. The biggest hurdle was certainly the OpenGL language and its very low level concepts.

At the heart of the helix [see figure A.7] there are the two main methods of the HelixWidget class, setData and drawWithNames. The setData method receives the raw array from MultiView and preprocessed it for the helix. The preprocessing is needed when data with different temporal resolutions are displayed at the same time. In this case, the setData method finds the greatest common denominator, calculated using the euclidean algorithm³⁰, of the temporal resolutions and expands every array where needed so that each value for each time-step of each variable is present [see listing 4.3]. setData can either just fill the slots with null values or it can interpolate the data using different methods. Finally, the method calculates some values needed for the correct functioning of the helix.

Listing 4.3: MultiView multidimensional example array before and after preprocessing in setData

```
{
    'snowdepthS0003': [(0, 0.0), (57600, 5.5)],
    'snowdepthS0004': [(0, 0.0), (14400, 0.0), (28800, 0.0), (43200, 0.0), (57600, 0.0)]
}
#preprocessing with linear interpolation
{
    'snowdepthS0003': [(0, 0.0), (14400, 1.1), (28800, 2.2), (43200, 3.3), (57600, 4.4)],
    'snowdepthS0004': [(0, 0.0), (14400, 0.5), (28800, 9.7), (43200, 0.0), (57600, 6.3)]
}
```

The drawWithNames method, actually draws the helix on the screen. The method goes over the preprocessed array and subsequently creates the helix object. For each time-step a GLQuad³¹ is drawn and coloured using the hue of the variable and the saturation calculated depending on its value. If the user selected in the GUI to smooth the helix, then more quads are drawn per time-step. After

²⁹<https://svn-tec.informatik.uni-rostock.de/landvis/LandVis/trunk>

³⁰http://en.wikipedia.org/wiki/Euclidean_algorithm

³¹OpenGL Quadrilateral

the quad(s) of a time-step and variable is (are) drawn, the helix is translated and the new variable's quads are drawn. Once all variables are drawn, the helix is rotated to start drawing the next time-step and the process continues until the whole helix is drawn. Listing 4.4 shows a pseudo code implementation of the drawing algorithm.

Listing 4.4: HelixView's DrawWithNames algorithm in pseudocode

```
foreach timestep
    set time label if at half cycle
    foreach quadsPerTimestep
        foreach variable
            calculate saturation
            draw GLQuad
            traslatehelix //move to next variable
    rotatehelix //move to next time-step
```

4.5 Scattergram Python plugin extension

The Scattergram tool³² [see figure 4.9] is a scatter-plot tool that allows to plot the values of two layers against each other. The tool can track the mouse position over the map canvas and highlight the correspondent value on the scatter-plot. Scattergram allows as well to select a polygonal region on the map canvas and then highlights all the plot points included in the selection. Basically all the scattergram functions work from the map canvas to the scattergram. The Scattergram plotToMap extension does exactly the contrary, it allows the user to select point(s) on the scattergram and to have them plotted on the map canvas.

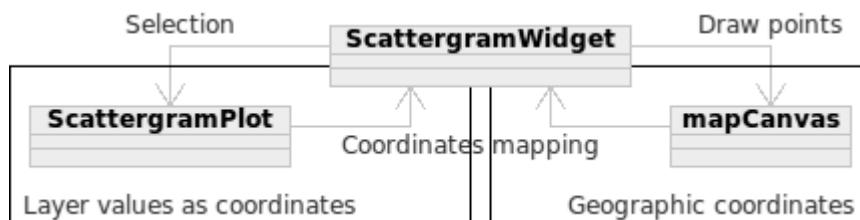


Figure 4.9: High level architecture of Scattergram's plotToMap functionality [Own illustration]

³²The full code base can be found at <http://hub.qgis.org/projects/scattergramidentify/repository>

The extension [see figure A.9] has been programmed by adding and extending to the existing code-base at three main parts. The first part takes care of creating a new vectorial layer in the map canvas when plotToMap is activated and to remove it when the tool is deactivated. The second main change was to add a support for cross-referencing the coordinates. The original system created an XY pair of values for the plot by reading the value of the two layers at any given point and then displayed it on the plot. The extension saves the geographic coordinates of each analysed pixel on the map in an array indexed using the plot coordinates [see listing 4.5].

Listing 4.5: Scattergram's coordinates mapping algorithm in pseudocode

```

Foreach pixel in map canvas
    vx = getValueAtPoint(layerX, point)
    vy = getValueAtPoint(layerY, point)
    valuesX.append(vx)#values plotted on scattergram
    valuesY.append(vy)
#map plot coords to map coords
    coords[vx, vy] = point

```

The last main modification was the addition of a plot selection tool (single value and polygonal selection) that convert the plot coordinates of the selected point(s) into geographic coordinates and would draw points on the map canvas.

4.6 Plugins starter Python plugin

PluginStarter³³ was my first code in Python and my first contact with the QGis python API and is therefore not too complex but it was a very good way to start understanding all the different layers involved in a QGis plugin. The plugin is formed by two main parts [see figure A.8, a configuration dialog and the main action. The dialog lists all the available plugins actions³⁴ and allows the user to select which actions should be executed when the run button is clicked. Once the user decides what actions should be executed on click, the configuration is saved and the plugin is ready to use. The main action of the plugin executes then all the

³³The full code base can be found at <http://hub.qgis.org/projects/pluginsstarter/repository>

³⁴other plugins attach their own actions to the QGis toolbars

actions saved in the settings' dialog at once as many times as needed.

The very simple functionality of the plugin is very useful when a user needs to repeatedly execute multiple actions at once and could easily be extended to support multiple profiles so to emulate a real working environment starter.

5

Results

This chapter is dedicated to the analysis of the results obtained in the thesis. I chose to evaluate the produced tool with the help of expert interviews due to the high amount and quality of information generated by this method in a relatively short time. The chapter is a thorough evaluation of the expert interviews and will report failures and achievements of the implemented solutions.

5.1 Expert interview setup

In this section I will review the expert interviews that were performed after the completion of the coding part of the project. The developed tools have been expert tested to find out if they satisfy the user's needs and the usability requirements expressed in the design phase of the project. Basing on the requirements resulted from literature research, the various brainstorming sessions and the focus group [see chapter 3], a test environment has been set up where five experts had to solve a series of tasks and evaluate the difficulty and effectiveness of the tool they used to solve the task. The test scenario was based on real analysis tasks inspired by the original design requirements, using model data from the GeoTOP model.

The choice to perform expert interviews instead of a broadband user testing was driven by three main arguments. First, since the tool is mainly tailored toward expert users, it is a sensible choice to test it with such kind of users. Second the quality and quantity of the feedback coming from an expert interview is

much higher than the results of a mass survey. Furthermore, interviews give the interviewer the unique opportunity of following-up and clarifying participants' responses Courage and Baxter [2005]. Third, a fairly complex tool has an inherent learning curve. It therefore needs some time to get acquainted to it. In wide user testings the available time with each participant is usually relatively short, whereas in expert interviews each session can last up to two hours Courage and Baxter [2005]. It was chosen to perform a semi structured interview since it ideally fulfills the requirements of such a program test. Thanks to this format, the user can behave freely and comment on his or her actions while still following a given structure Courage and Baxter [2005]

5.1.1 Test setup

On session begin the users where asked few questions regarding their experience with GIS, QGis, environmental data and multi-temporal data to assess their knowledge. After a short explanation about the project (only one expert had not been in the focus group) the participants were presented a printed script [Appendix A.7] that led them through the whole session. Furthermore, a verbal explanation about what was expected from them and how the system was designed and supposed to work was given to them. Finally, when needed, a quick overview of basic GIS concepts and QGis use was given to assure a minimal work ease during the whole session.

5.1.2 Test tasks

The first tool the participants were asked to use was the data loader. After loading the required data the first analysis task was undertaken. Using only the layers visibility settings¹ present in QGis they were asked to analyse the data and to find temporal patterns. After an exploration time of about ten minutes the user was prompted to move over to the next task. The next two tasks were aimed to the research of temporal patterns using specially crafted visualisations. Time plot visualisation was the first visualisation the participants were asked to use.

¹This is very similar to the way the geoTOP data is analysed currently

The task they received was the same as in the previous step. Similarly they were asked to look for time patterns using the time helix. The fourth task asked to look for outliers and clusters in the scattergram plot of two layers and to find out if these values were following a spatial pattern. This is a typical challenge researchers face when exploring new spatial datasets. Finally the experts were shown the implementation of the globe plugin and they were asked to evaluate the navigation controls and the general look and feel of the tool. Unfortunately due to some software version problems it was impossible to have the participants test the globe implementation thoroughly. Once all the tasks had been performed, the participants were asked to freely comment the tools they had been using, to rate them for efficiency as well as intuitiveness and to give inputs on possible ameliorations.

5.1.3 Participants structure

The participants were chosen in such way that a broad spectrum of specialists in the field was covered. Because different experience was very important as well I tried to maximize the heterogeneity of the five participants [Courage and Baxter, 2005]. A further very important criteria was to have somebody who is an expert in user interfaces but who did not know the type of data used in the tests very well. The final selection of experts was composed as follows.

Participant 1 is a mathematician enrolled as Ph.D. student at the Glaciology, Geomorphodynamics and Geochronology department [3G] with no experience in GIS and intermediate experience with temporal and environmental data.

Participant 2 is a Post-doc fellow at the 3G department with lot of experience and day to day exposure to GIS (but no experience with QGis), temporal data and environmental data. This participant knew the model data used very well since these are the data she usually works with.

Participant 3 is a senior researcher at the 3G department with much experience but middle daily exposure to GIS (no QGis), temporal data and environmental data. Participant 3 was involved from the beginning of the project's design phase but had not had the chance to effectively test the tools before this session.

Participant 4 is a research associate at 3G department where he is mainly pro-

gramming the models used in this test. Participant 4 has much experience in GIS/QGis and uses these tools regularly in his daily work. Furthermore participant 4 uses regularly temporal and environmental datasets.

Participant 5 is a senior research associate at the department for Geographic Information Visualisation and Analysis. Participant 5 has been using GIS for a long time but does not use it a lot any more. His main interest lies in interface design. He has little exposure to environmental and temporal data and has been invited to the test for this reason.

Table 5.1 below shows the result of the questions asked to the participants before the beginning of the test session.

P	GIS PK	QGis PK	ED PK	TD PK	GIS DU	QGis DU	ED DU	TD DU
1	0	0	2.5y	2.5y	0/10	0/10	4/10	6/10
2	10y	0	10y	10y	3/10	0/10	8/10	9/10
3	14y	0	16y	16y	1/10	0/10	3/10	3/10
4	15y	7	15y	15y	5/10	5/10	5/10	5/10
5	15y	0	0	0	2/10	0	0	0

Table 5.1: Experts previous knowledge (PK) in years of experience and daily usage (DU) of GIS, QGis, environmental data (ED) and temporal data (TD)

5.2 Results evaluation

5.2.1 Data loader

The first tool the participants were asked to use was the temporal data loader. As seen before the interface only allows little interaction: the user can set two regular expressions to match his file naming scheme and the start date of the import. As it can be seen in figure 5.1 the loader has a big log area at the top. To load data, the user has to click a button to get a file selection mask and choose the files that he or she wants to be loaded. In all five experiments, the users hesitated deciding what to do once they opened the loader. Asked what they where waiting for they all answered that they expected something to happen in the log window. Further questions about what they would have had expected to happen revealed

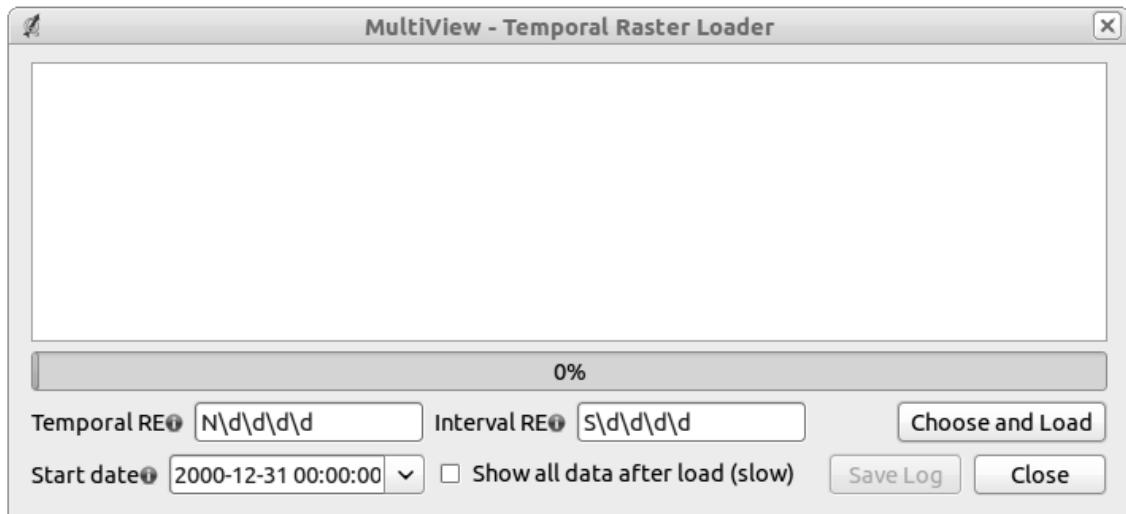


Figure 5.1: Temporal Data loader interface pre expert testing.

that all expected a file browser or some messages to be displayed instead of the white area. After telling them that the white area was meant as a log display they all moved on and clicked the “choose and load button”, selected the files and started the import. It must be remembered that three out of five participants did not know anything about the file naming scheme used in the experiment. While the three users who did not know the data spent very little time evaluating the available options of the interface, the two users that actually knew the data showed more interest in what could be set. Once the import started, a question box appeared in the loader [figure 5.2] asking if they wanted to override the manual settings with the ones found in the data header files present in the data folder. Every user chose “yes” with no hesitation and when asked why they chose to override the manual settings they all agreed that this had to be a file generated during data production and thus it would be accurate to use it. At the end of the import process no user choose to save the log of the import, closed the loader and moved on to task number two. The user interface specialist also noted that the data loader icon was too similar to the MultiView plugin one and that this could make people think that the loader only works if you use the MultiView plugin. A further remark on this topic was the suggestion that the loader button could be added to the list of buttons that import data instead of the list of the plugins

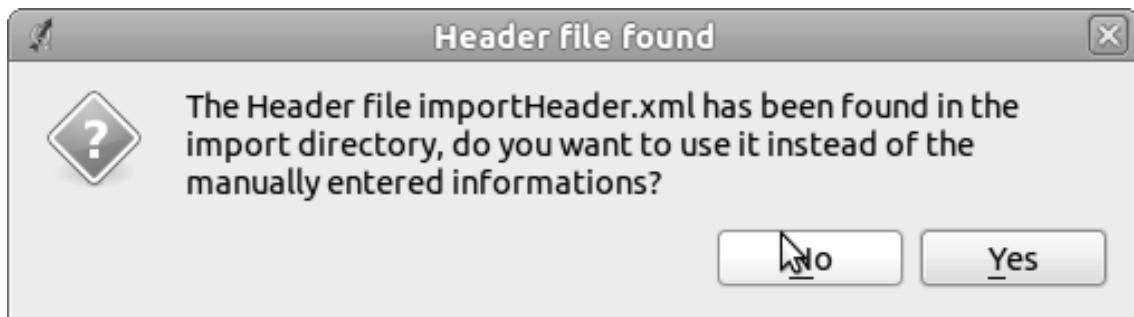


Figure 5.2: Dialog presented to users when an header file is found in the data import folder

As a direct result of this test task, the data loader has been improved by putting an explanatory text in the log area when the loader is opened. This should allow future users to easily understand how the loader works and make them more efficient. Also, two users noted that this is a minor issue since it only poses a problem during the first usage and that even without my input they would have, by default of other options, clicked on the correct button. Furthermore, the loader icon has been changed to a more meaningful icon as can seen in figure 5.3.



Figure 5.3: MultiView plugin, old dataloader and new dataloader icons

5.2.2 QGis layer visibility

When trying to identify temporal patterns using only the standard QGis layer visibility options the participants started switching on and off subsequent layers but mostly gave up within 3 minutes complaining that the method was very inefficient and that it was very hard to remember one time-step and compare it to the next or previous. Furthermore all the participants said that it was almost impossible to remember and compare more than two steps at a time. No user could identify any temporal pattern but the could correlate the variations of the variables values to the topography.

5.2.3 TimePlot visualisation

After the short experience using the QGis Layer visibility settings the participants were prompted to turn on the MultiView plugin and start using the time plot. Each participant had different issues while using the MultiView plugin but everybody had a major problem seeing the warning box. The warning box is a fixed location on the dialog where warnings and hints are shown to the user. No participant actually saw it and could thus not get the needed hints on how to use the tool. This problem was so important that it was fixed directly after the test sessions by simply displacing the warning box from the lower left corner to the middle just over the plot where the user actually expects something to happen [figure 5.4].

Directly connected to the bad visibility of the warnings/hints was the difficulty of knowing what to do. Since the MultiView plugin needs multiple inputs to work properly, the user has to select the variables first and then move to the map canvas to select the location they want the variables to be shown. Furthermore the user can choose if to select the location by right click (left click is reserved for map panning in QGis) or to continuously update it via mouse move.

Once they had discovered the warning box and thus understood better how to use the tool, most of the participants used both the tracking systems for different subtasks. The right click mode was mainly used when the participant wanted to have a detailed look at the data for a specific location, for example when wanting to use the zoom on the plot or to change the displayed variables. On the other hand they used the mouse move tracking when they wanted to quickly scan the whole region shown on the map. All the participants found that both modi were very useful for the different tasks and that, once acquainted to the tracking system, it was very easy to look for temporal patterns.

As stated before the users had many issues with and suggestions for this tool since they all found it very useful and spent a lot of time trying out for a better understanding of its potentiality. In the following paragraphs each issue will be shortly described and possible solution analysed. The list is prioritized by importance for the future development of MultiView.

Multiple y axis or normalisation: Most users had problems with the Y axis

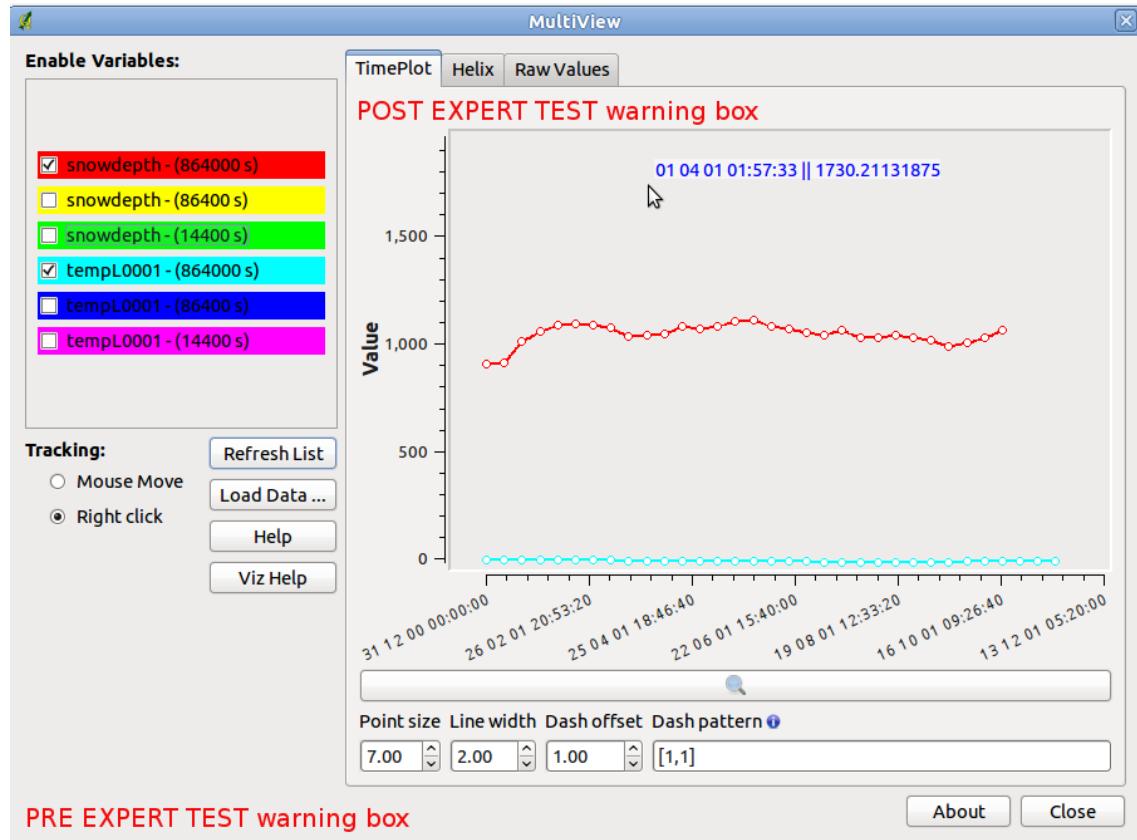


Figure 5.4: TimePlot visualisation as used during the expert testing with added pre and post expert testing warning box

of the plot. This axis only shows one value scale for all the active variables and thus if the range of a variable is for example 50-3000 and for another it is 0-10, the differences in the second variable will be flattened out by the big combined range (0-3000). This was a known problem and already a top development priority. Thanks to the user feedbacks some alternative solutions to this problem have been found and will be implemented in the near future. Two users proposed to use multiple axes (one for each variable type) but then agreed that having more than two-three Y axes on a plot would render it very difficult to read. The solution that will be implemented will allow the user to toggle Y value normalization so that the different variables can be better analysed for scale independent trends. Further, the tool should allow to display the real value on demand using the cursor tooltip.

Subsequent locations' analysis: Currently the time plot tool allows the analysis of one location at a time (or multiple location by moving the mouse on the map canvas) for multiple time steps and multiple variables. It was suggested by the participants to allow spatial analysis by leaving the last visualised graph's lines on the plot and setting their appearance to dashed or dotted lines [see figure 5.5] so that a direct comparison of two different locations could be performed much more easily. This feature still requires a substantial amount of work and is therefore planned for future development.

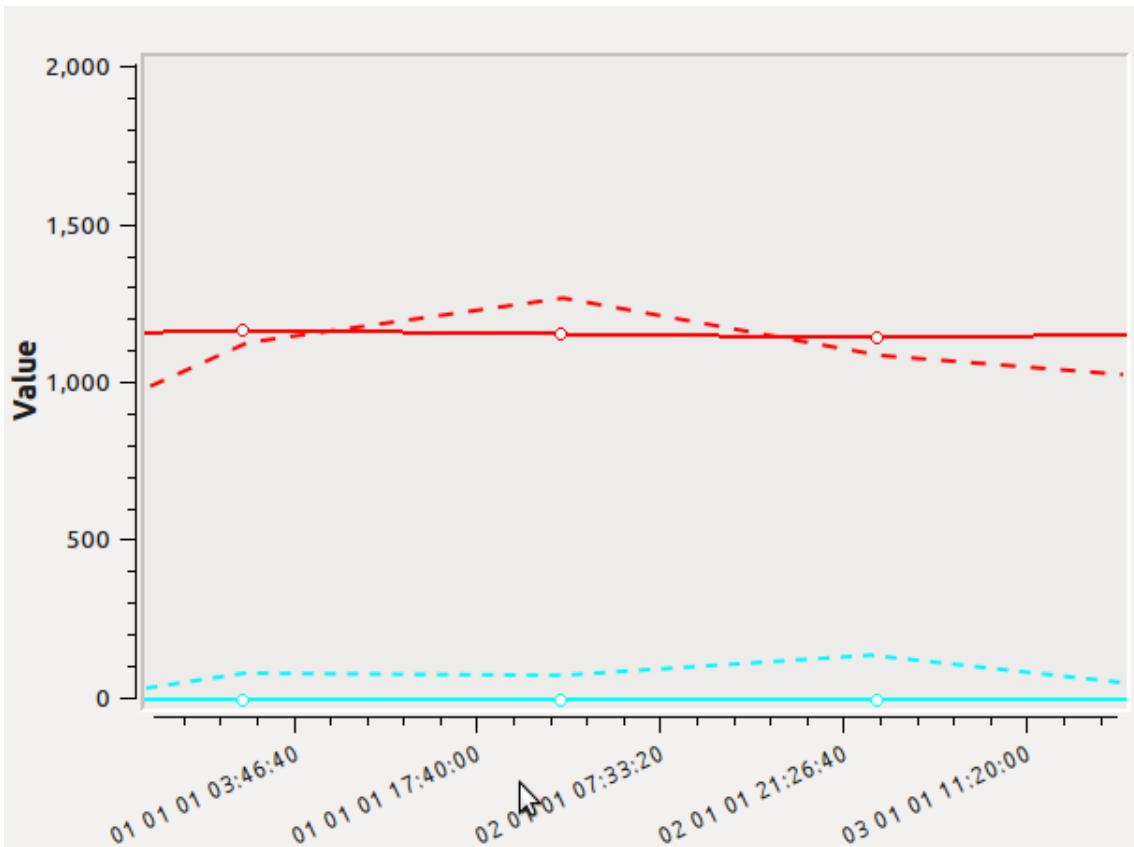


Figure 5.5: TimePlot visualisation sketch with added last selected values lines

Time axis labels reduction: The amount of date labels on the time axis was found to be too high by most participants. Furthermore, the experts suggested to put the labels only where data was present instead of at regular intervals. This feature was considered very important by the participants and the work load for the implementation was estimated to be acceptable so the feature was imple-

mented after the expert testing.

Visualisation Printing² : one of the participants said that it would be very useful to be able to print the current visualisation. Due to the very simple implementation, basic printing support was directly added to the code-base. None the less it would be important to extend this feature to add more information, such as location, enabled variables and timestamps to the printout.

Zooming related issues: during the testing, the implementation of the zooming class seemed to be a little unstable and to cause random bugs to all the users. This inconsistent behaviour forced the participants to use less a feature that they said they would have used more due to its usefulness. Notwithstanding once the participants knew what manipulations they could execute without generating errors, they regularly used the zoom to analyse the data closely. While analysing the zoomed data the participants wished they could pan the plot to keep the same zooming level while browsing the data.

Colour scheme editing² : The last issue reported by the testers was the impossibility of editing the colouring scheme of the variables. As seen in section 4.4 the colouring algorithm is a basic one but still delivers satisfactory results in most cases. When dealing with many variables, or if external constrains would impose a specific colouring scheme, the possibility of user defined colours would be highly appreciated. Since this issue does not limit the functionality of the tool it was decided to leave it for future developments.

5.2.4 HelixView visualisation

Once the participants were satisfied with the TimePlot they were prompted to move on to the Helix visualisation. The first comment by most users was satisfaction because the controls on the left side (variable selection, tracking system selection, ...) stayed the same. The participants could so profit of the confidence they had built up during the TimePlot usage period and solely focus on the usage of the helix. As for TimePlot, the users spent much time trying out the available features and functionalities as well as to see how they could use the tool to solve

²This issue was observed during the usage of TimePlot but applies as well to HelixView and other future visualisations.

their problems.

Differently from for TimePlot, every user needed some time at the beginning to understand what actually the visualisation showed and how to read it. The main problem the participants had was to actually understand how to set-up the visualisation. The helix visualisation needs one more setting than TimePlot to work, the duration of one cycle. While TimePlot can just show the data without caring about how long a cycle has to be, the implicit cyclical nature of the Helix imposes a design constraint that is data dependent. If the data is sampled at a daily resolution it cannot be visualised at an hourly resolution without having to interpolate data. Interpolation of data was not performed on purpose since it does not make sense for this user case. The only exception is when data with different resolutions were displayed. In this case, the interpolation of the lower resolution data is needed to allow comparison among the different resolutions.

The users had to be made aware of this and they all reacted saying: “oh yes, logic” and promptly set the duration to the wanted value. This is actually a feature of the Helix since it allows to show data with various temporal resolution and to look for patterns in different periodicity.

Disabled interpolation box when not needed: As can be seen in figure 5.6 the UI of HelixView allows the user to select a type of “interpolation for multiple temporal resolutions”. This selector disturbed the users heavily since it only has an effect when data with different temporal resolutions are displayed but it was always selectable. On suggestion of two users, the interface has been updated so that when there is no need for the switch, then it is deactivated.

Date readability improvements, to many labels : Similarly to what noted for TimePlot, the participants found that the amount of time labels was too high and that it was very hard to read them. A basic solution has been implemented after the session, but problems still exist when very big datasets are loaded.

Zooming inconsistent with qgis: The default zooming binding to the mouse-wheel in QGis zooms in when scrolling up whereas the default of PyQGLViewer is the opposite. The amount of work needed to substitute the default PyQGLViewer behaviour was considered too big for the first working version and is thus left as a possible future work.

Keep zooming on repaint: A further zooming related issue is the continuous

zoom reset when the viewer is repainted. A viewer repaint is triggered either every-time a new variable is activated or deactivated or when a new location on the map is chosen.

Colour saturation legend: Two participants suggested adding a colour legend to the side of the helix so that the absolute values of each saturation level can be estimated without having to use the value picker tool on each helix step. This idea seems very interesting but it must be further investigated on how to display the values of many variables without clogging the visualisation.

Helix manipulation buttons: One of the participants suggested to add buttons to the UI to allow manipulating the Helix in some constrained ways. For example the test user suggested to add controls to rotate the Helix around the time axis and to set it frontally to the user. These ideas could improve the usability and intuitivity of the helix visualisation a lot. The general feedback experts gave about the usability of the helix as a 3D object was that they were used to just use trial and error approach when dealing with 3D manipulation of objects, thus the Helix was no harder to use than any other 3D visualisation.

Size instead of saturation: The last point on this list was suggested by one participant as a way of experimenting with visualisations. The participant suggested to try to display the value of each step on the helix by changing the ribbon thickness instead of the colour saturation. This approach would [according to the participant]: “maybe help in seeing transition s from one value to another”. It would be highly interesting to try out such a visualisation but an effective evaluation of the benefits and deficits of such an application would call for another user testing. It is probably appropriate to leave this for future extensions of the MultiView plugin.

5.2.5 Comparison of temporal visualisations

After having spent around 50 minutes of the session, every participant had finished all the temporal tasks and they were asked to resume their experience and to compare the three different methods they used. Of all the participants, nobody appreciated looking for temporal patterns using only the layers' visibility switches. On the other hand, it was an unanimous statement that TimePlot was

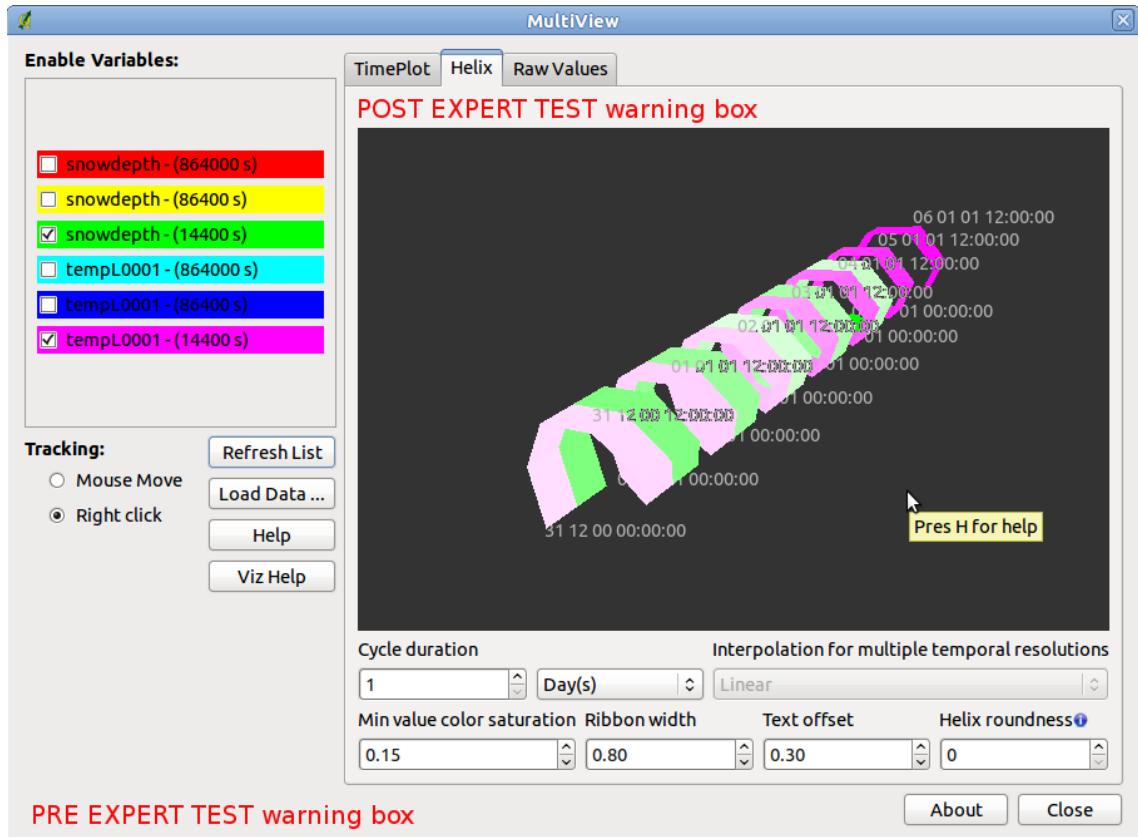


Figure 5.6: HelixView visualisation as used during the expert testing with added pre and post expert testing warning box

a very intuitive way of displaying data and that it was very easy to identify patterns in the data. Indeed, the participants performed very well when asked to identify patterns using this visualisation. This is a very satisfactory result since one of the aims of this project was to create a tool that would allow to replace the currently used visualisations.

The general opinion about HelixView was that it is difficult to understand how it works at the beginning but might be very helpful for cyclic data. The users could not easily identify complex patterns immediately, but easier patterns like the diurnal cycle were easily discovered. All testers stated that TimePlot was their preferred tool but that they would like to try using HelixView with data they know so to be able to exactly understand how it works and what its potential is.

This result was almost expected since all the participants were used to deal with statistics and plots. Since TimePlot is very similar to a standard plot, the user would have a very flat learning curve to start using the tool proficiently. On the other hand, the helix visualisation is mainly unknown and thus requires more effort to be understood.

After this comparison, the participants were asked to close MultiView and to move on to the next test subject, the scattergram plugin.

5.2.6 Scattergram

As seen in section 4.5 the scattergram plugin allows users to plot the values of two layers against each-other. Furthermore, the plugin allows the user to select a point on the map and display it on the scattergram canvas. The extension created during this MSc thesis enables the user to select points on the scattergram and to obtain their location on the map canvas. The task the users were supposed to solve after getting acquainted to the interface was to look for outliers and/or clusters in the data and to see if they were connected to a specific spatial pattern.

The experts had little trouble getting used to the interface and could easily start using the tool to look for patterns in the data. Their success was impressive, everyone could quickly detect clusters on the scattergram, have them plotted to the map and identify the topographic characteristics of the locations³.

During the usage, the testers found three main issues. The first was the absence of a variable name label on each Axes. This issue should be easy to fix since all needed functionalities are already existent. The other discovered bug was similar to the zoomer bug in TimePlot. The interaction between the zoomer class and the scale range seems to create random bugs here as well. The third problem is the text icon instead of a real icon. This problem is related to the impossibility of reaching the author of the original scattergram plugin and thus not being able to edit the original UI files⁴.

³Unfortunately, due to temporary software incompatibility at the moment of the test, the participants could not use the tool together with globe plugin and could thus not test its efficacy in a 3D DEM

⁴As of 10.04.2011 the Original author has contacted me with the wish of merging our codebases

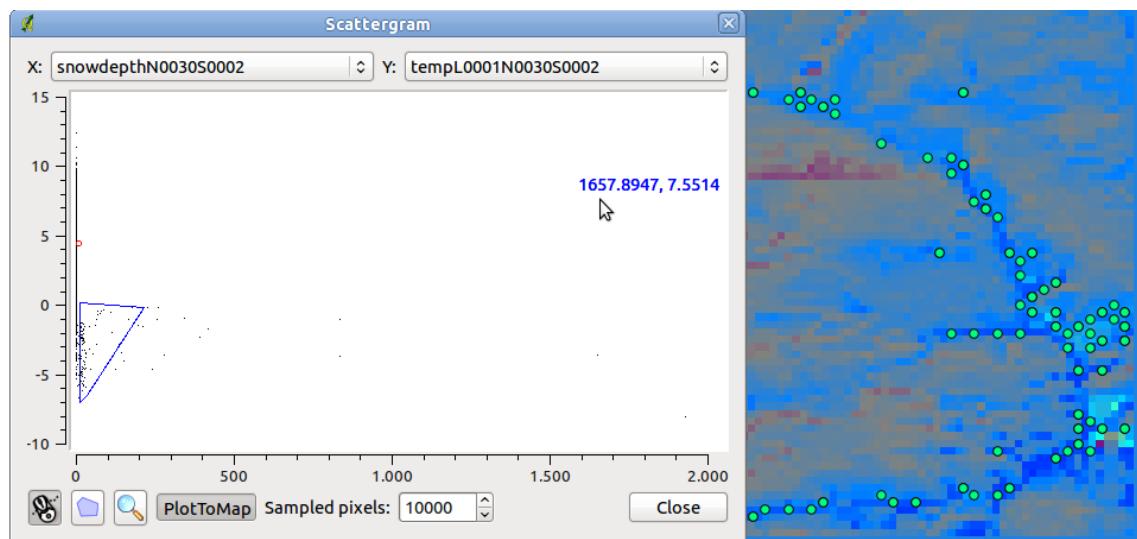


Figure 5.7: Scattergram visualisation used during the expert testing

5.2.7 Globe

The last task the users were asked to perform was a superficial evaluation of the UI of the Globe plugin, by telling what they thought each button of the UI would do. This task had to be kept minimal since at the moment of the test, due to versions incompatibilities, the plugin was running incredibly slow and only on an old branch of QGis. As seen in sections 4.3, 4.4 and 4.5 the globe needs a multi-threaded version of QGis which nowadays is a branch of QGis 1.4 but MultiView and some functionalities of Scattergram require at least QGis 1.7.

Most users found all the buttons to work as expected. The only button that was confusing the users was the one to reset rotation and tilting to perpendicular view with north above. As it can be seen in figure 5.8 this button was the same as the tilt down button and thus it was hard to understand if it was a bug or actually a button with functionality. The button has been changed to reflect the testers suggestions.

It was very unfortunate that the participants could not test the interaction with globe, since previous undocumented quick tests on friends had shown positive results.

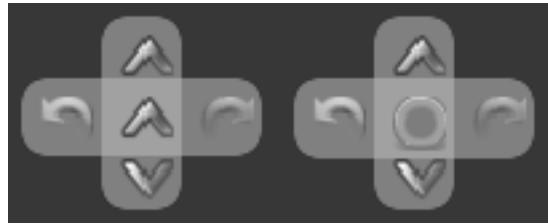


Figure 5.8: Globe tilt and rotate controls Pre- and Post- expert interview

5.2.8 General remarks

At the end of the session the users were asked to freely comment on the tools they had been using, what impression they got from the software and how the system could generally be improved. On top of the many points already quoted for each tool, the participants noted three more things.

Info ball inspires clicking: For users not used to Gnome (the desktop manager of Ubuntu), the whole graphic environment was new and thus the typical power users routines were absent. For example most users tried to click on the info balloons whereas the typical routine would be just to hover the element and wait for a tool tip to appear. This has nothing to do with the direct implementation of the tools, but it certainly had an impact on the users' performance since they first thought that the info balloons were not working and therefore not used them.

Expected inputs and warnings: A very common problem the participants had, was the initial difficulty of knowing which input was expected at what time. This problem was extremely sharpened by the very bad visibility of the warning box. The new placement of the warning box should soothe the problem.

Favourites tools: Most participants found the combination of scattergram and time plot to be their favourite. On top of this, their expectations on the working version of the globe plugin is very high. Although it is usually not very relevant what people think about future features Courage and Baxter [2005] it is still exciting to see that people have high expectations on the Globe plugin.

6

Discussion

In the first section of the chapter I will evaluate the produced software based on the formal requirements set in the design phase [see chapter 3]. The second part will be dedicated to answering the research questions formulated at the beginning of this thesis [section 1.3].

6.1 Formal requirements analysis

In section 3.3 a series of formal requirements for the tools were formulated to steer the development of the tools. In this section I will review the produced code, basing my analysis on the formulated requirements. The result of this analysis are summarised in table 6.1.

Requirement	Impl.	Major issues	Minor issues	Beta
Globe	yes	**	**	*
Visualisation framework	yes	no	*	***
TimePlot	yes	no	**	**
Helixview	yes	no	**	**
Scatterplot	yes	no	**	***
Temporal resolution selector	no	N/A	N/A	N/A
Time manager tool	no	N/A	N/A	N/A
Multivariate selector for TM	no	N/A	N/A	N/A

Table 6.1: Summary of the implemented features from the formal requirements. * Means low and *** high.

Requirement 1: A globe-like terrain model module with stereo 3D capabilities

The code developed for the globe plugin fulfils the stated requirements. All the features that were wished have been implemented along with some further features that make the globe easier to use. The only shortcoming of the globe plugin might lie in its performance and stability. As stated, the wished code quality was beta, meaning that it is fairly performing and stable but still has some glitches. I would say that Globe plugin falls into a late alpha code quality since there still are some major performance issues that were “show-stoppers” at times. Nonetheless, the code-base is very good and the potentiality for further development of the tool outstanding.

Requirement 2: A visualisation framework

The MultiView plugin has been successfully implemented as a modular visualisation formal Requirements framework. Its architecture reflects the wish of keeping the visualisations as a separate entity from the main data processing classes while still providing an easy method to implement a new visualisation. As shown in section 5.2.3 the code quality of the MultiView framework has only minor glitches that do not limit working with the tool. It can definitely be said that the wish for beta quality code is completely fulfilled.

Requirement 3: A time vs. value plot should allow comparing the trend of multiple variables over time for the selected location

The TimePlot visualisation has received very good reviews in the expert interviews. It was one of the favourite tools in the test. All the wanted features have been implemented and the code has very few bugs. The beta code quality requirement is fulfilled.

Requirement 4: A 3D Helix visualisation should be implemented to provide the details on demand in the analysis of cyclic data

Similar to the TimePlot visualisation, the HelixView plugin has posed no technical problems to the tester. Their difficulty in dealing with the tool was mainly conceptual. The functionality prescribed by the formal requirements is fully implemented and only few minor bugs are still present in the code. The code quality

requirements are fulfilled.

Requirement 5: A scattergram tool with a functionality to plot the values on globe

The functionality implemented through the Scattergram extension allows the user to select points on the scattergram and to see their location on the globe. This solution covers the exact requirement formulated for a tool that would allow the location of outliers and data inconsistencies. The code has very few problems, most of which are just related to performance optimisation, the requirements are therefore completely fulfilled.

Requirement 6: A temporal resolution selector for the visualisation framework
Not implemented due to time constrains.

Requirement 7: A Time manager tool that allows filtering the data on a certain time period and to animate the raster layers

Not implemented due to time constrains. A project of collaboration with the authors of TimeManager plugin for QGis exists and options will be evaluated after the completion of this thesis.

Requirement 8: A multivariate selector to work in combination with the time manager to allow filtering the data based on the variables

Not implemented due to time constrains.

6.2 Answering the research questions

Research question 1: How can a 3D terrain be successfully linked to multivariate spatio-temporal data visualisations?

The system implemented in this project bases heavily on the linking and brushing paradigm. The globe is used to chose the location of the data that needs to be analysed and the coordinates broadcast to all visualisation tools. The advantage

of this approach is that it allows extreme flexibility and loose coupling of the visualisation tools. The globe plugin could (and in fact is) be used in connection with any other plugin that needs a coordinate as input. This has the nice side-effect of allowing a steadily growing ecosystem of plugins¹ to trigger the features of the globe plugin and thus to deliver more feedbacks about its usability.

The main observation done analysing the users evaluating the system is that their interaction with a globe as location controller appears as comfortable as with a 2D map canvas. Of course this is the results of a small scale qualitative analysis and has no aspiration of statistical validity whatsoever. One possible explanation of this is the relatively large availability of globe software nowadays. Although globe renderers have existed for a long time, it is only in the last years that the development, led by the “google earth effect”, increased drastically. Thanks to this evolution and the similarities between the globe plugin and Google Earth, it could be speculated that the tool would perform well even in a broad band user testing.

The biggest uncertainty about Globe’s usability lies in the mouse controls used to select the location. On one hand, the mouse-hover concept has worked very well when the data needed to be quickly swept but cannot be used when a location had to be temporarily saved, since the coordinates would change as soon as the user moved the mouse. On the other hand, the right click solution to temporarily save the location seemed to confuse the users. This question, should be further investigated with the help of a larger scale summative usability evaluation.

Research question 2: *Which temporal data visualisations would make our system intuitive, efficient and easy to use with the specific type of data (i.e. temporal and cyclic data) generated by environmental monitoring projects?*

The visualisations that were considered fit for the task and successively submitted to the evaluation of the focus group where a heterogeneous group of classical and more modern and abstract visualisations. The focus group participants clearly leaned towards the more classical visualisations, their point being that if they were to perform data exploration and thus did not know the data they where

¹At the time of writing QGis has about 120 published plugins

looking at they would rather be using tools they are comfortable with. Nevertheless the participants still suggested implementing the HelixView to evaluate its potential with cyclic data.

The expert interviews confirmed the problem with unknown visualisations, the participants had significantly fewer problems understanding the TimePlot than they had understanding the HelixView. The interviewed experts suggested that the original difficulties they were having with the HelixView could probably be overcome by “playing around” with data they knew.

This results show how hard it can be to introduce a new visualisation in a work context where the users are acquainted to certain work-flows and tools. Nevertheless, all interviewed experts agreed that although difficult to grasp at first, the combination Globe-HelixView could yield a very intuitive system for cyclic data exploration and that it complements the easier to understand Time-Plot very well.

Research question 3: *How can data anomalies and outliers be visually correlated to the topography?*

A tool that would allow visualising values clusters, outliers and data anomalies their topographic context was considered very useful from all users since it would allow them to discern if a value was real or an anomaly (such as a sensor or model failure).

The implemented solution makes use of a very simple but, as noted by the interviewed experts, efficient idea. The selected scattergram data are back referenced to the Globe and displayed as points wrapped on the Globe’s DEM and on the 2D map canvas.

The technique could have been extended by changing the size, hue, saturation, shape and so on of the projected points depending on the variables values of each point to allow multivariate analysis on the globe. Such extension could on one hand be very beneficial for some use cases but could on the other hand make the tool much more complicate and maybe less effective since its main strength was (according to the expert interviews) the simplicity of use.

This results confirm the response to the research question number two that says that the interviewed experts preferred the easier tools to perform their data

explorations. As for the other research questions, this question would tremendously profit of a large summative usability evaluation.

Research question 4: *Does adding the surface topography (on a DEM), in comparison to more abstract visualisations (various 2D or 3D plots) make a (positive) difference in inference making in environmental (and similar) monitoring projects?*

Due to time constrains and to amount of coding needed for the implementation of the tools needed for this project, we had to discard the idea of a large scale comparative study between the developed tool and other already available visualisation packages. Such a task would probably require a whole thesis by itself to be properly conducted on a large user base due to the necessity to create test cases, complex datasets and evaluate which other tools should be used in the test.

7

Conclusions

This thesis demonstrates the complete process of building an ad-hoc application embedded in a mature GIS software to solve an existing problem within an environmental monitoring project implementing a novel combination of multivariate spatio-temporal data visualisations.

The whole user centred design process is followed and explained from the early planning stages, passing through the users task analysis, conceptual design, prototyping, expert guideline-based evaluation up to a formative usability evaluation. The last step of this methodology, a summative usability evaluation was not part of this thesis due to the imposed time constraints and should be considered for a future quantitative analysis project.

The developed applications are an example of how a (stereo) 3D visualisation can be used as input controller in a mixed 2D/3D environment. Furthermore, the developed globe plugin simplifies the GIS work-flow for terrain based data visualisation by allowing a direct (almost real-time) visualisation of any 2D data readable by QGis. The data export and preparation steps are no longer needed and the user can instantaneously see the results of its analysis in a terrain based (stereo) 3D visualisation.

The modular implementation of the MultiView visualisation framework has opened a vast possibility for future spatio-temporal data visualisation in QGis by creating a common development platform where independent developer can easily implement new visualisations. Lastly, the scattergram extension shows how the simple idea of backward georeferencing the scattergram data has generated a

useful and intuitive tool for the localisation of data clusters, anomalies and outliers.

7.1 Known limits

All the developed tools have some known limits, mainly easily fixed. Most of the Globe limitations are due to the performance issues of the code base, once its performance is stabilised, the tool-set will become highly interactive and further studies could be undertaken on it. One further important limitation of the globe plugin is that it currently supports only projects in WGS84. This should no longer be a major problem since the next release of QGis will support raster and vector re-projection on the fly and thus data in any projection will be loadable. On the side of the MultiView plugin, the main concern lies in testing the system with huge data amounts to test how it scales. The scattergram extension needs some performance improvement as well, but in general all the tools programmed in the thesis are usable and stable enough. For further details on the needed bug-fixes, extensions and proposed solution see the next chapter.

8

Future work

The whole code developed during this thesis is released to the public domain under the GPL license and is available in various repositories online [see chapter 4 for details]. As such, the code is available to anybody interested in fixing bugs, extending it or make it perform better as they wish. Often such work gets contributed back to the original code so that a common code-base is kept alive and growing. This one of the advantages of Open Source code development.

In this chapter I will present the still outstanding bugs and issues reports as well as some new ideas that could be implemented in the future of all the presented tools. A short hint on how to implement the idea is given as well to facilitate the work of future developers.

8.1 Globe plugin

Globe Plugin aims at the real time visualisation of the QGis map canvas on a 3D (true 3D as well) globe to allow data analysis, exploration and visualisation all using the same Software package. This approach has the advantage that all data can be studied while looking at the topography of the terrain. Therefore the topography is directly present and does not need to use a further visual variable. Furthermore this approach allows GIS users to produce high quality 3D output and visualisations without having to export or convert their data to other formats.

8.1.1 Google Summer Of Code Program [GSoC]

I have submitted Qgis-Globe to the GSoC 2011¹ program to be able to continue developing it after my thesis is completed since there is a great interests in the QGis community for such a tool. If the proposal [GSo, 2011] is accepted I'll have twelve further weeks to dedicate to the stabilisation of the code-base and to implement new features².

As seen in section 4.3, the globe plugin needs the multi-threaded version of QGis which is not yet part of the current development trunk. After the imminent release of QGis³, the first concern will be helping the merging of the threading branch into the current development trunk. Once this is done, the first part of the GSoC project would aim at a stabilization, performance increasing and bug fixing of the current Globe code-base. The second part would then be dedicated to the following features implementation:

- user selectable Real time Synchronisation with map canvas
- Creation of Signals for symbolization updates
- Use of 3D models in QGIS symbolization
- Python bindings
- Packaging
- Nicer “fly to” paths
- Navigation with Wiimote or Kinect⁴

Once all these features are implemented, Globe Plugin could be the precursor for a complete 3D data visualisation and analysis tool in Qgis.

In case the Globe project is not accepted for the GSoC program, all these improvements remain part of future developments of the Globe plugin but will need

¹http://www.google-melange.com/gsoc/document/show/gsoc_program/google/gsoc2011/home

²Unfortunately for the globe plugin, the GSoC proposal was not accepted in favour of another proposal of mine, QGis for Android

³QGis 1.7 Wrocław is planed to release on 2.5.2011

⁴This feature could be implemented as part of a current MSc Thesis at the GIVA department

more time to be implemented. Due to the very nature of Open Source it might as well be that further developers join the development team and that these features get implemented quicker than thought. It is in any case planned to complete the first part of the GSoC project before the release of QGis 1.8 so that Globe can be included in the default QGis install.

8.1.2 Globe on geowall

Up to present, Globe plugin's stereoscopic capabilities have been used only in anaglyph mode and performed well in one-two persons set-ups. One of the secondary goals of the project was the possibility of using large displays such as a GeoWall to allow collaborative work and presentation to large audience of such datasets in a true 3D manner. The possibility of displaying such data in a true 3D environment would allow on one hand collaboration among scientists and on the other hand it would allow the scientists to create involving and convincing presentations to laypersons by triggering what could be called the "WOW effect"⁵. The set up of this feature should theoretically be effortless since the functionality has already been included into the code-base, but dependencies and driver issues could always arise and this is why this feature needs further testing⁶.

8.2 MultiView plugin

As seen in section 5.2, many features can still be added to the MultiView plugin and some bugs are to be fixed. In this section I will resume which features should come into consideration for a latter implementation and how these could be designed to be as effective as possible.

⁵This point is very important foremost for non-specialistic public presentations

⁶This testing will be performed just after the delivery of the thesis in preparation to the thesis defence

8.2.1 Data loader

From the four suggested improvements for the Data Loader, two have been immediately implemented due to their easy implementation. The remaining improvements are the direct inclusion of a file browser into the dialog and the placement of the loader icon in the “add new data” bar instead of the plugin bar. While the former improvement would be straightforward to implement, the latter would need further research to have a proper implementation. A further feature that needs to be implemented is a real header file parser.

8.2.2 TimePlot visualisation

Of all the issues noted during the expert testing of the TimePlot visualisation, six could not be implemented due to time constraints and are listed here as possible extensions of the plots functionality.

A solution to the different values range problem could lie in normalization of the values and the display of relative values as it was done in the Helix visualisation. This would allow the user to analyse patterns of different variables with different value ranges. A further extension in this direction could be the possibility to allow the user to enter a scaling function for each variable.

The next feature that should be added to the TimePlot is the ability of saving the last selected (or even multiple) location on the map and to show the lines in the plot thus allowing the comparison of multiple location at multiple times as shown in figure 5.5.

The addition of plot panning capabilities when the plot is zoomed should as well be considered very high in the priority list of new features to be added to TimePlot. Furthermore, the workload for the implementation of such a feature is estimated to be only moderate since the QWTplot library has already all of the needed components to implement this successfully. Furthermore, the interaction between the zoomer class and the TimePlot, as noted before, should be proofed and debugged since the use of the zoom created random errors in the visualisation forcing the users to use it less than they wished. The possible errors lie here in the concurrent modification of the X and Y axes ranges.

The printing system currently implemented in MultiView⁷ allows for basic printing of the plot without any legend. To produce valuable printouts, this system should be expanded to at least add the colour legend of the variables to the print. Further enhancements would include the ability of adding a title and a user defined text to the plots.

Finally, the customizability of MultiView⁷ would greatly benefit of user selectable colouring schemes or individual colour assignment. This could be implemented by placing a colour picker near each variable name in the MultiView legend. The main advantage of this approach over placing the colour selection in the TimePlot itself is that these settings would then automatically propagate to all visualisations.

8.2.3 Helix visualisation

The most important feature to fix/extend in the Helix visualisation is the readability of the date labels when very big datasets are loaded. The implemented algorithm sets a label every half a cycle. This should be improved by making the labelling engine aware of the labels quantity by counting how many cycles will be generated and then adapt the amount of labels to a maximum allowed value by leaving more cycles or steps between each label.

The next bug that should be taken care of is the inconsistency of the scroll zooming with the default QGis behaviour. The solution here would be to program a custom mouse scroll event handler for PyQGLViewer that would work in the same way as QGis does. As well related to zooming of the helix is the loss of the zoom level on each canvas repaint. This issue might be one of the most complex to solve since it requires to save the actual zooming and viewing angle of the helix and then reapply it after a canvas repaint. The functionality is definitely available in PyQGLViewer an OpenGL but its implementation will need some more research.

After the implementation of the previous fixes is completed, three more features have been requested. The first one is the implementation of a colour sat-

⁷Although this issue applies to HelixView as well, it is listed here to be consistent with the order used in section 5.2.3

uration legend. One basic solution would be to implement a legend per each variable, but such a solution would work only for a limited number of variables on the cost of clogging the canvas. Implementing successfully this feature will require an evaluation and research of possible solution for the clogging problem. The second feature request is the addition of buttons that would allow the user to manipulate the helix under certain constrains. The requested buttons should at least allow the user to rotate the helix on the time axis and to rotate it by 90 degree on the Y axis so that it is frontal to the user. The implementation of these buttons should not pose big problems since PyQGLViewer listens to the QT signals framework⁸ and it already implements rotation controls constrained on an axis. The last feature is more an experiment request than a proper feature request, as explained in section 5.2.4, it would be interesting to try to use other methods to display the value of a step in the helix. For instance, it was suggested to try to implement a helix that shows values by changing the thickness of the ribbon instead of by changing the colour saturation. If such feature will be implemented, it would be interesting to user test the two (or more) methods to see which performs best.

8.3 Scattergram extention

The most important future development of Scattergram is the re-merging of the code-base with the author of the original plugin. This would allow the unification of code to maintain and allow the scattergram plugin to grow. After merging, two of the three issues listed in section 5.2.6 will become trivial to solve since the original UI files will be available. The only issue that will need more attention is the zoomer bug. As for TimePlot, fixing this issue will require more testing and debugging to try to identify the problem. Since the public release of the scattergram tool, some feedback of QGis users suggested to save the values of the selected points as attribute in the vectorial dataset. This could then be used for further analysis. Two further improvements could be implemented to augment the tool usability. A custom mouse event handler to allow a polygonal selection

⁸thus buttons can be easily added to the UI and made to emit signals

using the same key-bindings as the QGis standard tools would probably be the most important piece of code to add to the scattergram code-base. The other improvement would be the implementation of a quicker closest point algorithm during the look up of the values coordinates on a click.

8.4 Plugins starter plugin

PluginStarter was developed as first plugin to get acquainted to the PyQGis API and has thus limited functionalities. The main idea of the plugin was to create a sort of working environments that could be quickly launched. The plugin accomplishes this in a very easy manner by listing all the available plugins actions and allowing a user to select which plugins should be started on a click. The limitation of this approach is that only one environment can be saved at a time. The plugin would profit of a saving system where a user could save and load different profiles on one mouse click. Implementing such a feature should not be too complicated since it is only matter of saving a settings array to a permanent QSettings object. A more complicated task would be to extend the functionality not only to open plugins but to position them in a defined manner to create a real work environment starter.

8.5 Time manager plugin raster support

A further development of the software tools would be the implementation of raster support into the time manager plugin⁹. The main advantage of having such an infrastructure would be the possibility of easily explore the different time steps in complex datasets. The initial implementation could be based on the switching on and off of different layers. The time slider of the plugin would then turn on only the layers that apply to a given time and a variable selector would allow to choose which variable to display. This basic implementation would probably already suffice to augment even more the productivity of the tools-set.

⁹<http://www.geofrogger.net/trac>

The main foreseeable problem will be performance related. Canvas redrawing can be very expensive when dealing with many layers but with the due attention this problem should be manageable. Another way of facing the performance could lie in consideration of the implementation of a larger system based on the RASDAMAN¹⁰ server package. This approach would substantially increase the complexity and dependences of the project, but it might improve the tools performance.

8.6 Concluding personal note

This project allowed me to greatly improve and learn many skills. During this year I had the chance to give presentations and talks, attend GIS meetings, lead a focus group, perform various one-on-one interviews, design a tool based on user needs, write formal requirements and last but not least, write this thesis.

The development of the plugins exposed me to intense coding in languages (Python and C++) that I didn't use before and introduced me to the realm of distributed development in a large team. Being able to work in such manner and having been introduced to the open source GIS development in Switzerland will certainly benefit my career.

It has been a long hard road, but it was definitely worth the effort and I'm now ready for the next challenge.

¹⁰<http://rasdaman.eecs.jacobs-university.de/trac/rasdaman>

Bibliography

Free Software vs. Open Source [online, last visited: 2011-01-09]. Available from:

<http://www.gnu.org/philosophy/free-software-for-freedom.html>.

GAV Flash Tools [online, last visited: 2011-04-22]. Available from: <http://ncva.itn.liu.se/tools>.

GeoVISTA Studio [online, last visited: 2011-04-22]. Available from: <http://www.geovistastudio.psu.edu/>.

Google Summer of Code Proposal [online, last visited: 2011-04-14]. Available from: <http://www.google-melange.com/gsoc/proposal/review/google/gsoc2011/mbernasocchi/1001>.

HEALTH GeoJunction [online, last visited: 2011-04-22]. Available from: <http://www.apps.geovista.psu.edu/hgj/>.

JavaScript InfoVis Toolkit [online, last visited: 2011-04-22]. Available from: <http://thejit.org/>.

Manifesto for Agile Software Development [online, last visited: 2011-01-12]. Available from: <http://agilemanifesto.org>.

OECD eXplorer [online, last visited: 2011-04-22]. Available from: <http://vitagate.itn.liu.se/GAV/eXplorer/OECDRegional/>.

Protovis [online, last visited: 2011-04-22]. Available from: <http://vis.stanford.edu/protovis/>.

PyQt Whitepaper [online, last visited: 2011-03-02]. Available from: <http://www.riverbankcomputing.com/static/Docs/PyQt4/pyqt-whitepaper-a4.pdf>.

VIS-STAMP: A Visualization System for Space-Time and Multivariate Patterns [online, last visited: 2011-04-22]. Available from: <http://www.spatialdatamining.org/software/visstamp>.

Aigner, W., Miksch, S., Müller, W., Schumann, H., and Tominski, C. (2008). Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics*, 14:47–60. Available from: <http://dx.doi.org/10.1109/TVCG.2007.70415>.

Aigner, W., Miksch, S., Müller, W., Schumann, H., and Tominski, C. (2007). Visualizing Time-Oriented Data – A Systematic View. *Computers & Graphics*, 31(3):401–409. Available from: <http://dx.doi.org/10.1016/j.cag.2007.01.030>.

Andrienko, G., Andrienko, N., Demšar, U., Dransch, D., Dykes, J., Fabrikant, S. I., Jern, M., Kraak, M.-J., Schumann, H., and Tominski, C. (2010). Space and Time. In Keim, D., Kohlhammer, J., Ellis, G., and Mansmann, F., editors, *Mastering the Information Age – Solving Problems with Visual Analytics*. Eurographics Association. Available from: <http://www.vismaster.eu/book>.

Andrienko, N., Andrienko, G., and Gatalsky, P. (2003). Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing*, 14(6):503–541. Visual Data Mining. Available from: <http://www.sciencedirect.com/science/article/B6WMM-49H1102-1/2/e0ae4bae5f05168db1b15cabce525e36>.

Bertin, J. (1983). *Semiology of graphics : diagrams, networks, maps*. University of Wisconsin Press, Madison, Wis. : translated by William J. Berg.

Bleisch, S., Burkhard, J., and Nebiker, S. (2009). Efficient Integration of Data Graphics into Virtual 3D Environments. In *24th International Cartography Conference*. Available from: http://www.3dgi.ch/publications/bls/ICC2009_Paper_BleischBurkhardNebiker.pdf.

- Bleisch, S. and Dykes, J. (2008). *Using web-based 3D visualization for planning hikes virtually - an evaluation*, chapter 21. CRC Press, Florida. Available from: http://www.3dgi.ch/publications/bls/2008_BleischDykes_InnovationsInGIS13.pdf.
- Bleisch, S. and Nebiker, S. (2008). Connected 2d and 3d visualizations for the interactive exploration of spatial information. In *XXI ISPRS Congress*. Available from: http://www.3dgi.ch/publications/bls/2008_BleischNebiker_ISPRS.pdf.
- Bostock, M. and Heer, J. (2009). Protovis: A graphical toolkit for visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1121 –1128.
- Buja, A., McDonald, J. A., Michalak, J., and Stuetzle, W. (1991). Interactive data visualization using focusing and linking. In *Proceedings of the 2nd IEEE conference on Visualization*, pages 156–163. Available from: <http://portal.acm.org/citation.cfm?id=949633>.
- Butler, D. (2006). Virtual globes: The web-wide world. *Nature*, 439. Available from: <http://dx.doi.org/10.1038/439776a>.
- Cockburn, A. and McKenzie, B. (2001). 3d or not 3d?: evaluating the effect of the third dimension in a document management system. In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '01*, pages 434–441, New York, NY, USA. ACM. Available from: <http://doi.acm.org/10.1145/365024.365309>.
- Cockburn, A. and McKenzie, B. (2002). Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, CHI '02*, pages 203–210, New York, NY, USA. ACM. Available from: <http://doi.acm.org/10.1145/503376.503413>.
- Compieta, P., Di Martino, S., Bertolotto, M., Ferrucci, F., and Kechadi, T. (2007). Exploratory spatio-temporal data mining and visualization. *J. Vis. Lang. Comput.*, 18:255–279. Available from: <http://portal.acm.org/citation.cfm?id=1244473.1244708>.

- Courage, C. and Baxter, K. (2005). *Understanding Your Users: A Practical Guide to User Requirements Methods, Tools, and Techniques*. Morgan Kaufmann, 1 edition. Available from: http://www.bibsonomy.org/bibtex/2cf46ac7201fcb402dccd79822bbab8e3/projekt_bui.
- Daassi, C., Nigay, L., and christine Fauvet, M. (2006). A taxonomy of temporal data visualization techniques. Available from: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.122.8704>.
- de Alwis, B. and Sillito, J. (2009). Why are software projects moving from centralized to decentralized version control systems? In *Cooperative and Human Aspects on Software Engineering, 2009. CHASE '09. ICSE Workshop on*, pages 36–39. Available from: <http://www.computer.org/portal/web/csdl/doi/10.1109/CHASE.2009.5071408>.
- Feller, J. and Fitzgerald, B. (2002). Understanding open source software development. Available from: <http://portal.acm.org/citation.cfm?id=513726>.
- Fuhrmann, S. and Pike, W. (2005). User-centered design of collaborative geovisualization tools. In Dykes, J., MacEachren, A. M., and Kraak, M.-J., editors, *Exploring Geovisualization*, pages 591 – 609. Elsevier, Oxford. Available from: <http://www.sciencedirect.com/science/article/B861G-4P9TW7H-G/2/e2afc439d2d99c35e622479309a35e94>.
- Grinstein, G., Trutschl, M., and Cvek, U. (2001). High dimensional visualizations. In *Data mining conference KDD Workshop*. Available from: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.130.90>.
- Gruber, S. (2011). Personal communication.
- Hautle, S. (2010). Visualisierung multidimensionaler Daten mit Zeitbezug: Eine 3D-Erweiterung der Parallelen Koordinaten.
- Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer*, 1:69–91. 10.1007/BF01898350. Available from: <http://dx.doi.org/10.1007/BF01898350>.

- Inselberg, A. (2009). *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. Available from: <http://www.springer.com/mathematics/numerical+and+computational+mathematics/book/978-0-387-21507-5>.
- Jern, M., Johansson, S., Johansson, J., and Franzen, J. (2007). The gav toolkit for multiple linked views. In *Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV '07. Fifth International Conference on*, pages 85–97. Available from: <http://www.computer.org/portal/web/csdl/doi/10.1109/CMV.2007.21>.
- Jones, R. R., McCaffrey, K. J. W., Clegg, P., Wilson, R. W., Holliman, N. S., Holdsworth, R. E., Imber, J., and Waggett, S. (2009). Integration of regional to outcrop digital data: 3d visualisation of multi-scale geological models. *Comput. Geosci.*, 35:4–18. Available from: <http://portal.acm.org/citation.cfm?id=1453266.1453568>.
- Kirschenbauer, S. (2005). Applying “true 3d” techniques to geovisualization: An empirical study. In Kraak, J. D. A. M. M.-J., editor, *Exploring Geovisualization*, pages 363–387. Elsevier, Amsterdam. Available from: http://www.elsevier.com/wps/find/bookdescription.cws_home/703524/description.
- Kjellin, A., Pettersson, L. W., Seipel, S., and Lind, M. (2008). Evaluating 2d and 3d visualizations of spatiotemporal information. *ACM Trans. Appl. Percept.*, 7:19:1–19:23. Available from: <http://doi.acm.org/10.1145/1773965.1773970>.
- Kosara, R., Hauser, H., and Gresh, D. L. (2003). An interaction view on information visualization. In *Eurographics 2003 Proceedings*, volume 22, page 123–138. Available from: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.1078>.
- Kraak, M. (2003). The space-time cube revisited from a geovisualization perspective. In *Proceedings of the 21st International Cartographic Conference (ICC)*, pages

- 1988–1996. Available from: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.102.9231>.
- Lakhani, K. R. and von Hippel, E. (2003). How open source software works: 'free' user-to-user assistance. *Research Policy*, 32(6):923–943. Available from: <http://www.sciencedirect.com/science/article/B6V77-479TM54-1/2/5672b73de696a2d8a1d68e6d5747a2cb>.
- Li, X. and Kraak, M.-J. (2005). New views on multivariable spatio-temporal data: the space time cube expanded. In *International Symposium on Spatio-temporal Modeling, Spatial Reasoning, Analysis, Data Mining and Data Fusion*, volume XXXVI, pages 199–201. Available from: <http://www.isprs.org/proceedings/XXXVI/2-W25/source/NEW%20VIEWS%20ON%20MULTIVARIABLE%20SPATIO-TEMPORAL%20DATA%20THE%20SPACE%20TIME%20CUBE%20EXPANDED.pdf>.
- Lindlof, T. R. and Taylor, B. C. (2002). *Qualitative Communication Research Methods*. Sage, Thousand Oaks, CA, 2 edition. Available from: <http://www.uk.sagepub.com/books/Book233741>.
- Monmonier, M. (January 1994). Focus groups for design improvement in dynamic cartography. *Cartography and Geographic Information Science*, 21:37–47(11). Available from: <http://www.ingentaconnect.com/content/acsm/cagis/1994/00000021/00000001/art00004>.
- Muller, W. and Schumann, H. (2003). Visualization methods for time-dependent data - an overview. In *Proceedings of the 2003 Winter Simulation Conference*, volume 1, pages 737 – 745 Vol.1. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1261490.
- Noirhomme-Fraiture, M. (2002). Visualization of Large Data Sets: The Zoom Star Solution. *The Electronic Journal of SYMBOLIC DATA ANALYSIS*, 0(0).
- Rase, W.-D. (2003). Von 2D nach 3D – perspektivische Zeichnungen, Stereogramme, reale Modelle. *Kartographische Schriften*, 7. Available from: http://www.bbsr.bund.de/nn_103116/BBSR/DE/Raumbeobachtung/

Werkzeuge/Visualisierung/Veroeffentlichungen__Artikel/von2Dnach3D,templateId=raw,property=publicationFile.pdf/von2Dnach3D.pdf.

Riedl, A. (2005). Digital Globes - from Virtual to Real. In *Proceedings, 22. ICA Cartographic Conference*. Available from: http://icaci.org/documents/ICC_proceedings/ICC2005/htm/pdf/oral/TEMA24/Session4/ANDREASRIEDL.pdf.

Roberts, J. C. (2005). Exploratory visualization with multiple linked views. In Dykes, J., MacEachren, A. M., and Kraak, M.-J., editors, *Exploring Geovisualization*, pages 159 – 180. Elsevier, Oxford. Available from: <http://www.sciencedirect.com/science/article/B861G-4P9TW7H-14/2/a7547f1403d73bd87341e61e7ca60417>.

Rogowitz, B. E. and Treinish, L. A. (1996). How not to lie with visualization. *Computers in Physics*, pages 268–273. Available from: <http://drona.csa.iisc.ernet.in/~vijayn/courses/DAV/papers/RogowitzTreinishHowNotToLieVis.pdf>.

Shepherd, I. D. H. (2008). *Travails in the Third Dimension: A Critical Evaluation of Three-Dimensional Geographical Visualization*, pages 199–222. John Wiley & Sons, Ltd. Available from: <http://dx.doi.org/10.1002/9780470987643.ch10>.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Visual Languages*, pages 336–343, College Park, Maryland 20742, U.S.A. Available from: <http://dx.doi.org/10.1109/VL.1996.545307>.

Siirtola, H. (2007). *Interactive Visualization of Multidimensional Data*. PhD thesis, University of Tampere. Available from: <http://acta.uta.fi/pdf/978-951-44-6939-8.pdf>.

Takatsuka, M. and Gahegan, M. (2002). Geovista studio: A codeless visual programming environment for geoscientific data analysis and visualization. *Computational Geoscience*, 28:1131–1144.

- Tobon, C. (2005). Evaluating geographic visualization tools and methods: an approach and experiment based upon user tasks. In Kraak, J. D. A. M. M. M.-J., editor, *Exploring Geovisualization*, pages 591–609. Elsevier, Amsterdam. Available from: http://www.elsevier.com/wps/find/bookdescription.cws_home/703524/description.
- Tominski, C., Abello, J., and Schumann, H. (2004). Axes-Based Visualizations with Radial Layouts. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, pages 1242–1247. ACM. Available from: <http://doi.acm.org/10.1145/967900.968153>.
- Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2005). 3D Information Visualization for Time Dependent Data on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 175–181. IEEE Computer Society. Available from: <http://doi.ieeecomputerociety.org/10.1109/IV.2005.3>.
- Tominski, C. and Schumann, H. (2008). Enhanced Interactive Spiral Display. In *Proceedings of the Annual SIGRAD Conference, Special Theme: Interactivity*, pages 53–56. Linköping University Electronic Press. Available from: http://www.ep.liu.se/ecp_article/index.en.aspx?issue=034;article=013.
- Torres, R. (2002). *Practitioner's Handbook for User Interface Design and Development*. Prentice Hall. Available from: <http://portal.acm.org/citation.cfm?id=516081>.
- Torvalds, L. and Diamond, D. (2001). *Just for fun. The Story of an Accidental Revolutionary*. HarperCollins, New York. Available from: <http://www.bibsonomy.org/bibtex/2d6090f165540c4c26b3805be69103b1a/topyli>.
- Volbracht, S., Domik, G., Shahrabaki, K., and Fels, G. (1997). How effective are 3d display modes? In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '97*, pages 540–541, New York, NY, USA. ACM. Available from: <http://doi.acm.org/10.1145/258549.259022>.

- Wang, G., Mercier, L., Collins, D. L., and Cooperstock, J. R. (2009). A comparative study of monoscopic and stereoscopic display for a probe-positioning task. In *Medicine Meets Virtual Reality (MMVR) Medical Education Conference*. Available from: <http://www.cim.mcgill.ca/sre/publications/2009-MMVR-Wang.pdf>.
- Wang Baldonado, M. Q., Woodruff, A., and Kuchinsky, A. (2000). Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '00, pages 110–119, New York, NY, USA. ACM. Available from: <http://doi.acm.org/10.1145/345513.345271>.
- Weber, M., Alexa, M., and Muller, W. (2001). Visualizing time-series on spirals. In *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on*, pages 7 –13. Available from: <http://citeseervx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.65>.
- Wegenkittl, R., Löffelmann, H., and Gröller, E. (1997). Visualizing the behavior of higher dimensional dynamical systems. In *in Proceedings IEEE Visualization '97*, pages 119–125.
- Williams, S. (2002). *Free as in Freedom*. O'Reilly Media, Sebastopol, CA (USA). Available from: <http://www.oreilly.com/openbook/freedom/>.

A

Appendix

A.1 QGis API UML

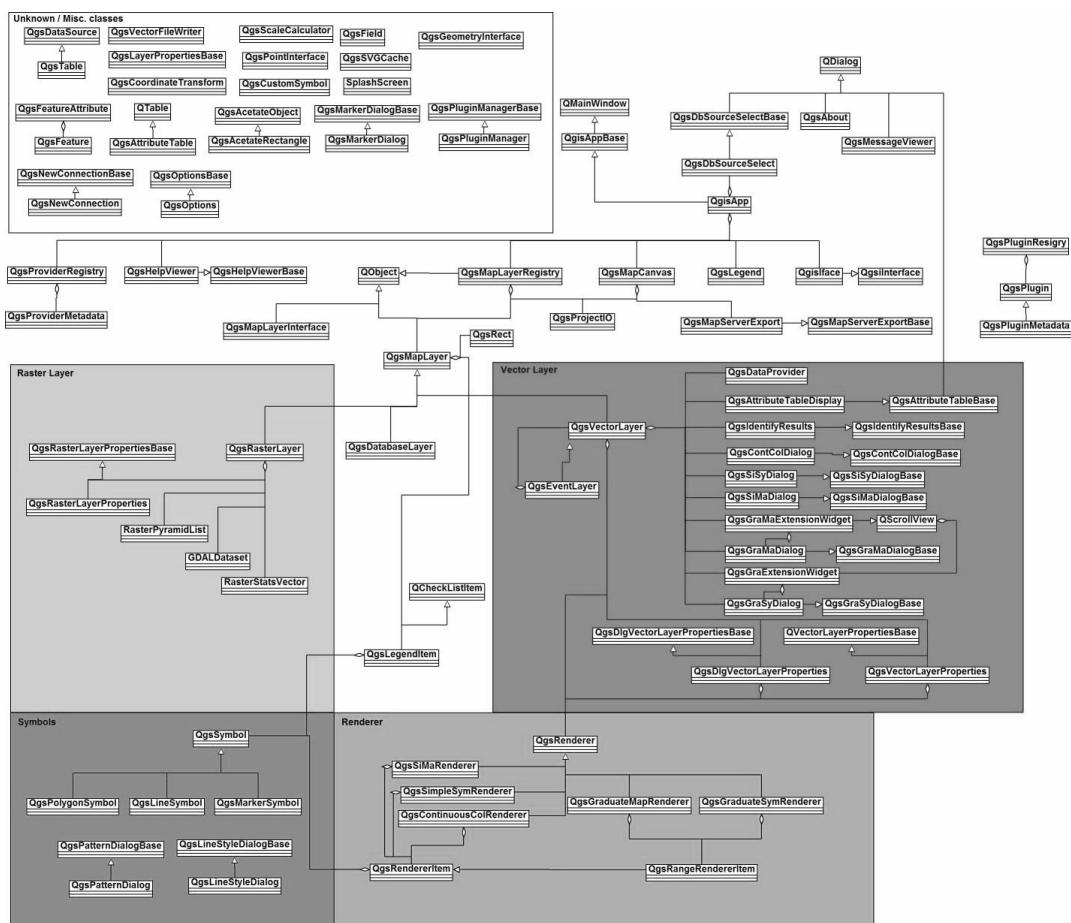


Figure A.1: QGis API UML diagram [Source: <http://www.qgis.org>]

A.2 Globe plugin

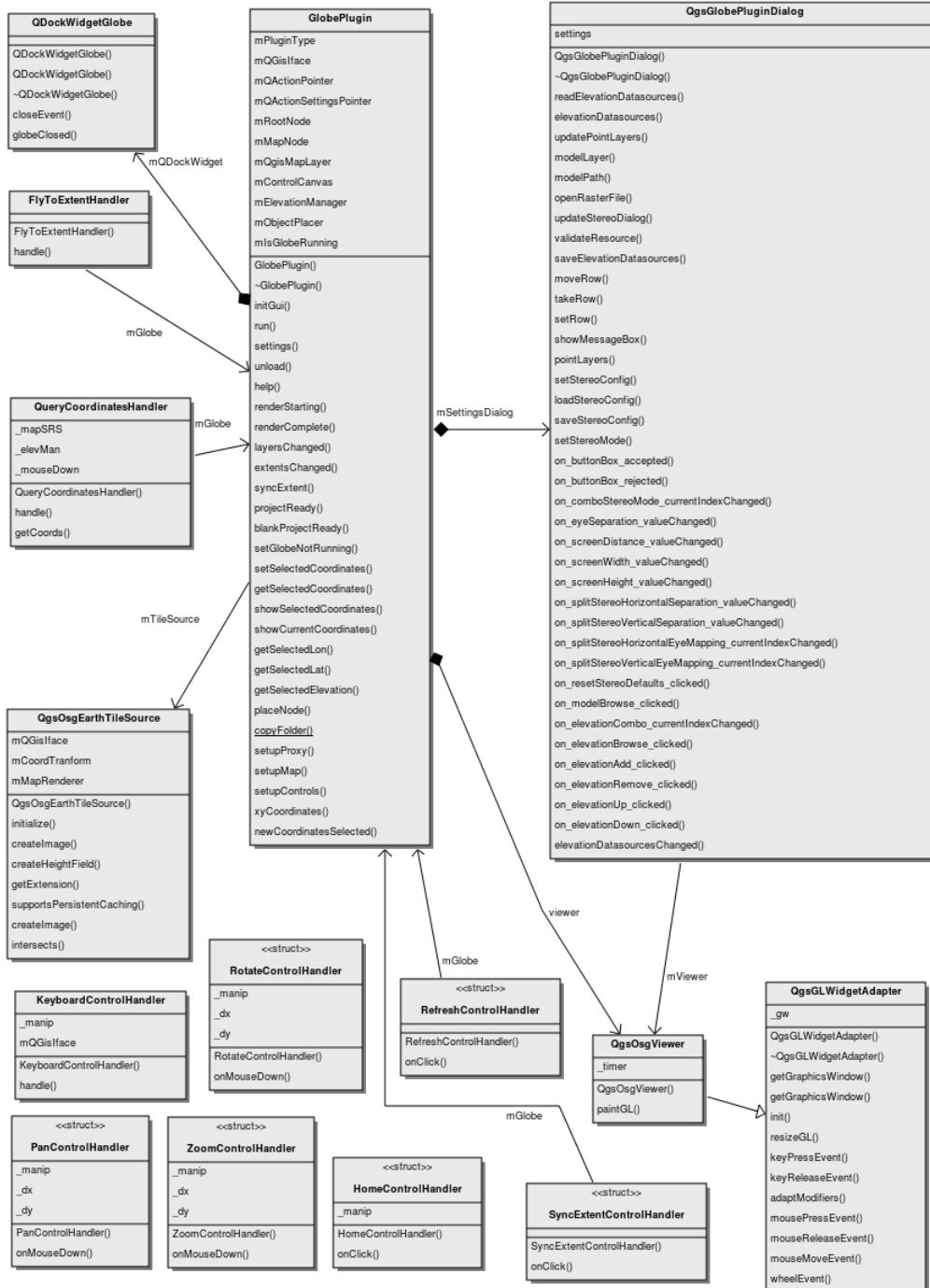


Figure A.2: Globe plugin UML class diagram [Own illustration]

A.3 Multiview

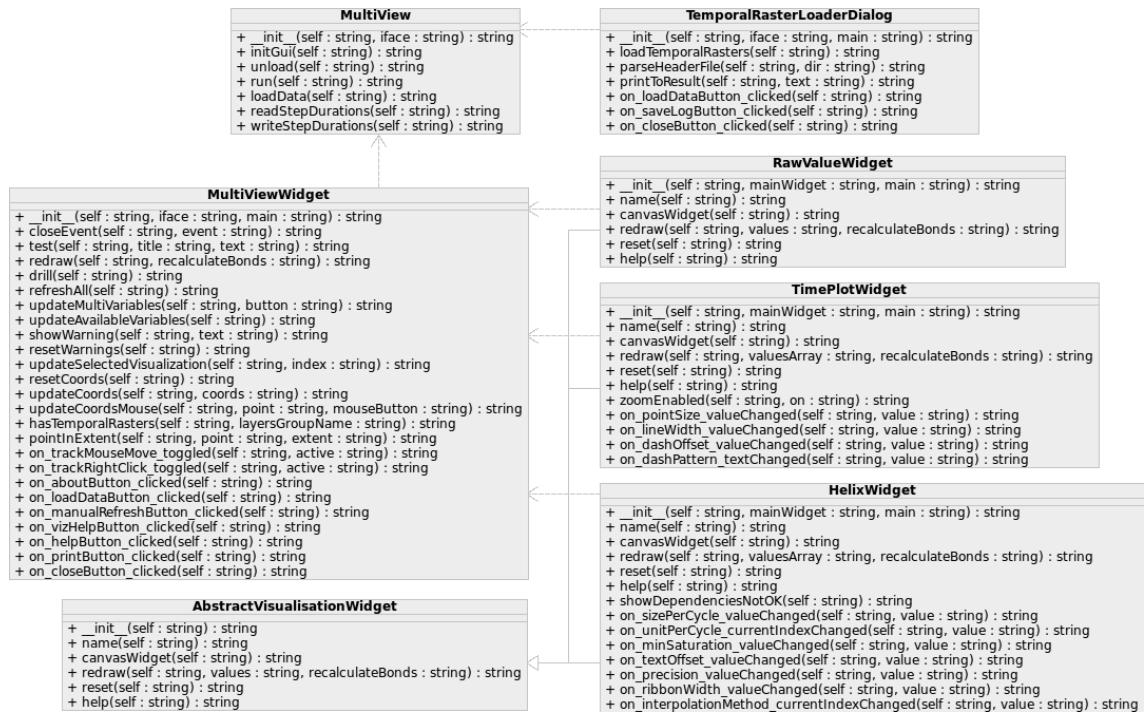


Figure A.3: MultiView plugin high level architecture UML class diagram [Own illustration]

A.3.1 Dataloader

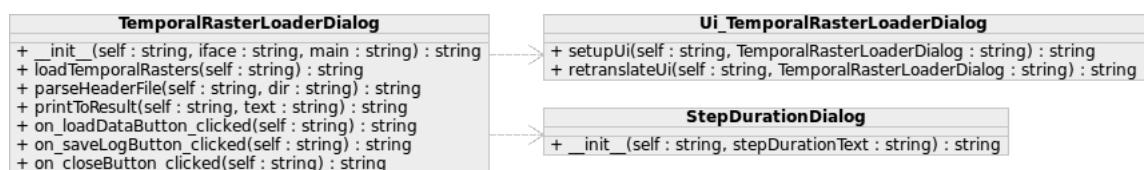


Figure A.4: Dataloader UML class diagram [Own illustration]

A.3.2 Raw value widget

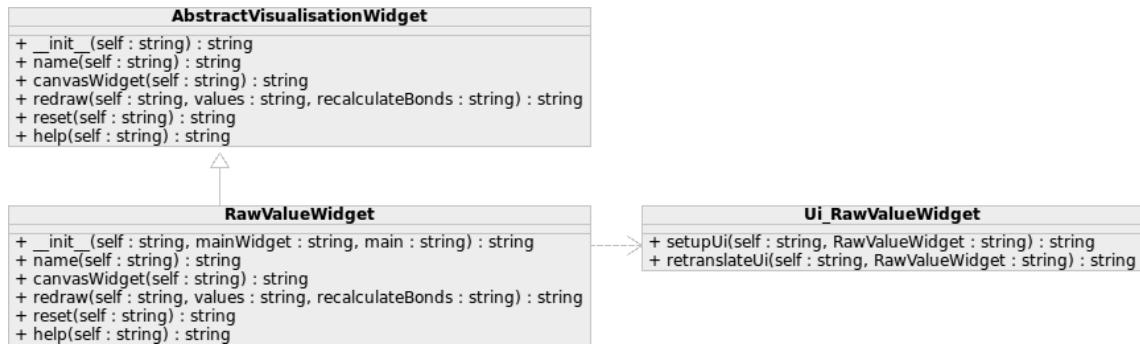


Figure A.5: Raw value UML class diagram [Own illustration]

A.3.3 TimePlot widget

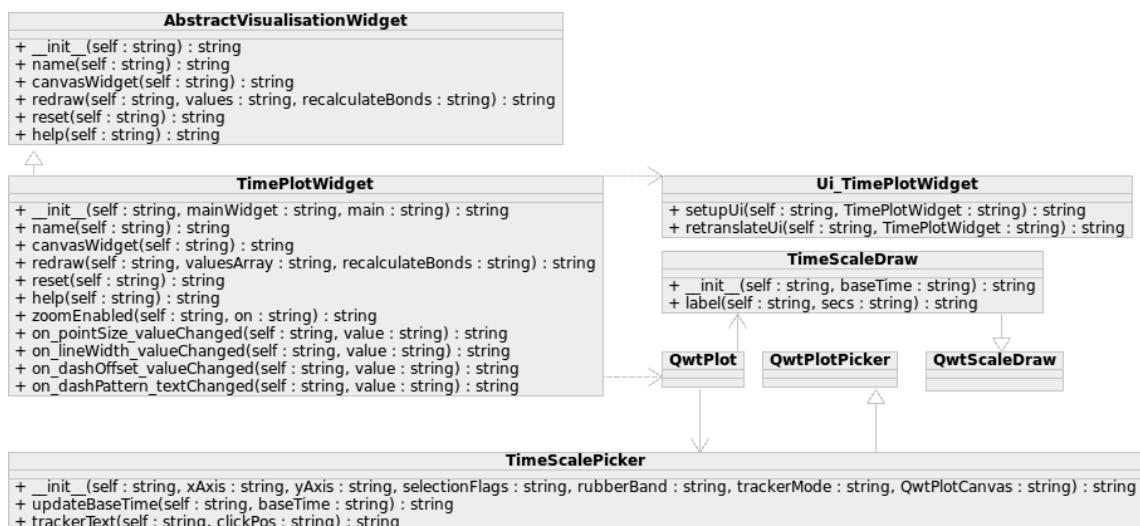


Figure A.6: TimePlot UML class diagram [Own illustration]

A.3.4 HelixView widget

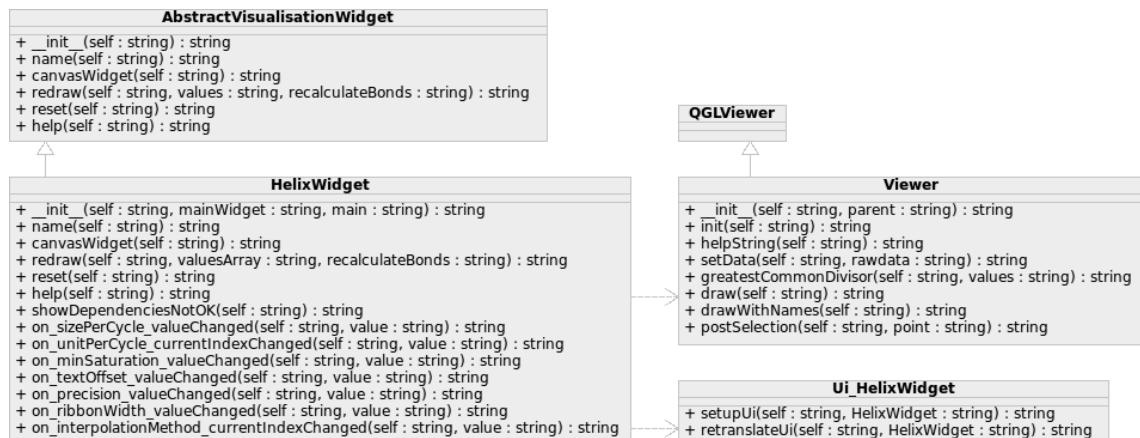


Figure A.7: HelixView UML class diagram [Own illustration]

A.4 Plugins starter plugin

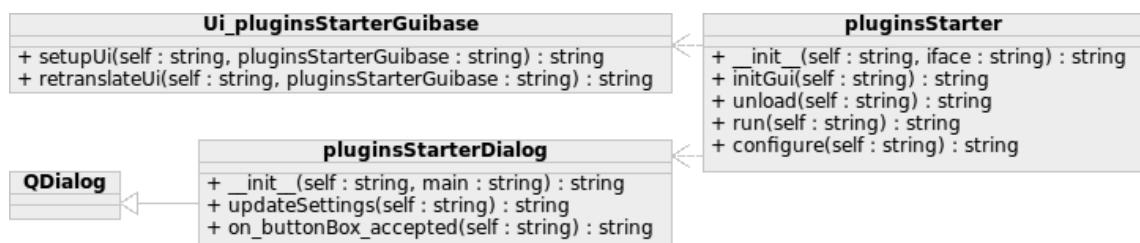


Figure A.8: Plugins starter UML class diagram [Own illustration]

A.5 Scattergram plugin

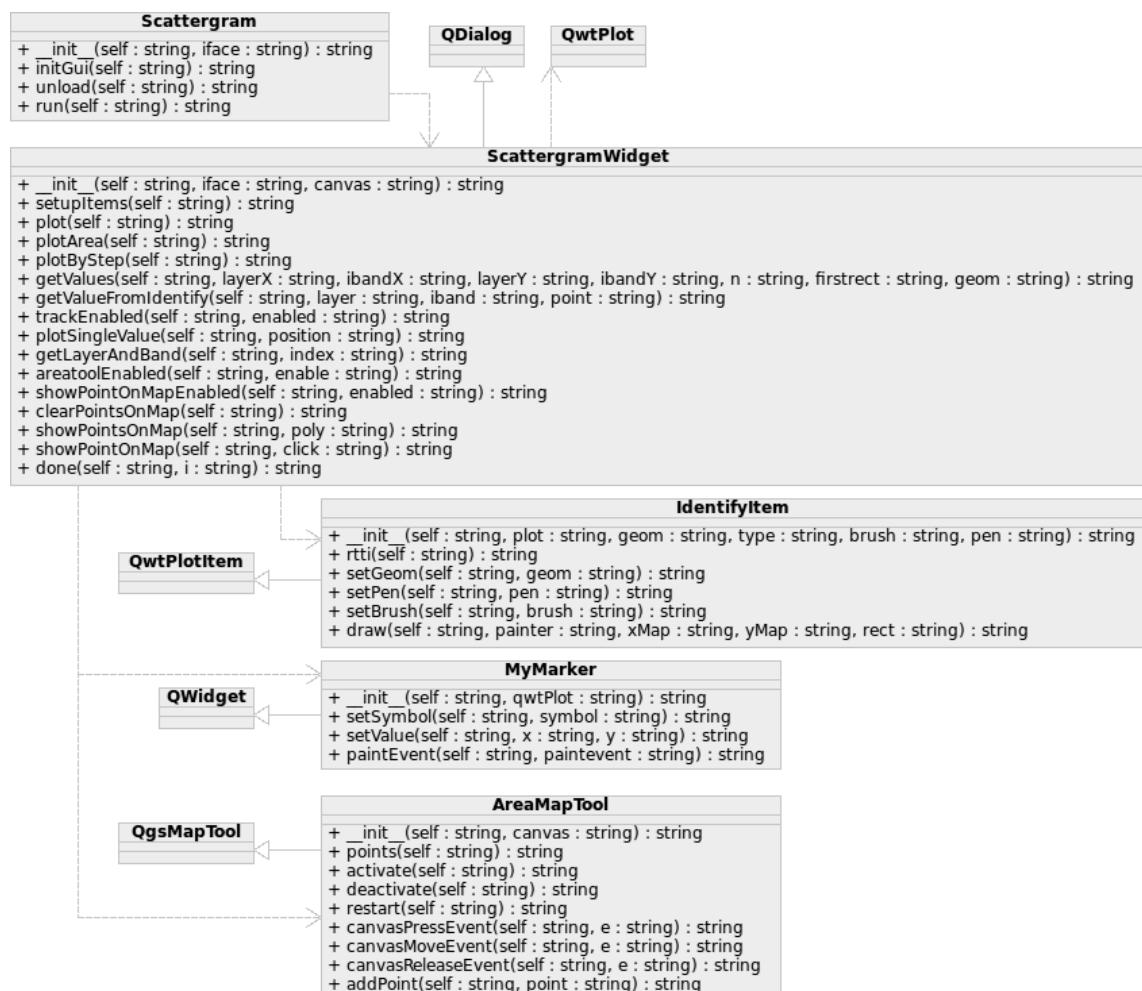


Figure A.9: Scattergram UML class diagram [Own illustration]

A.6 Focus group participant's mockups

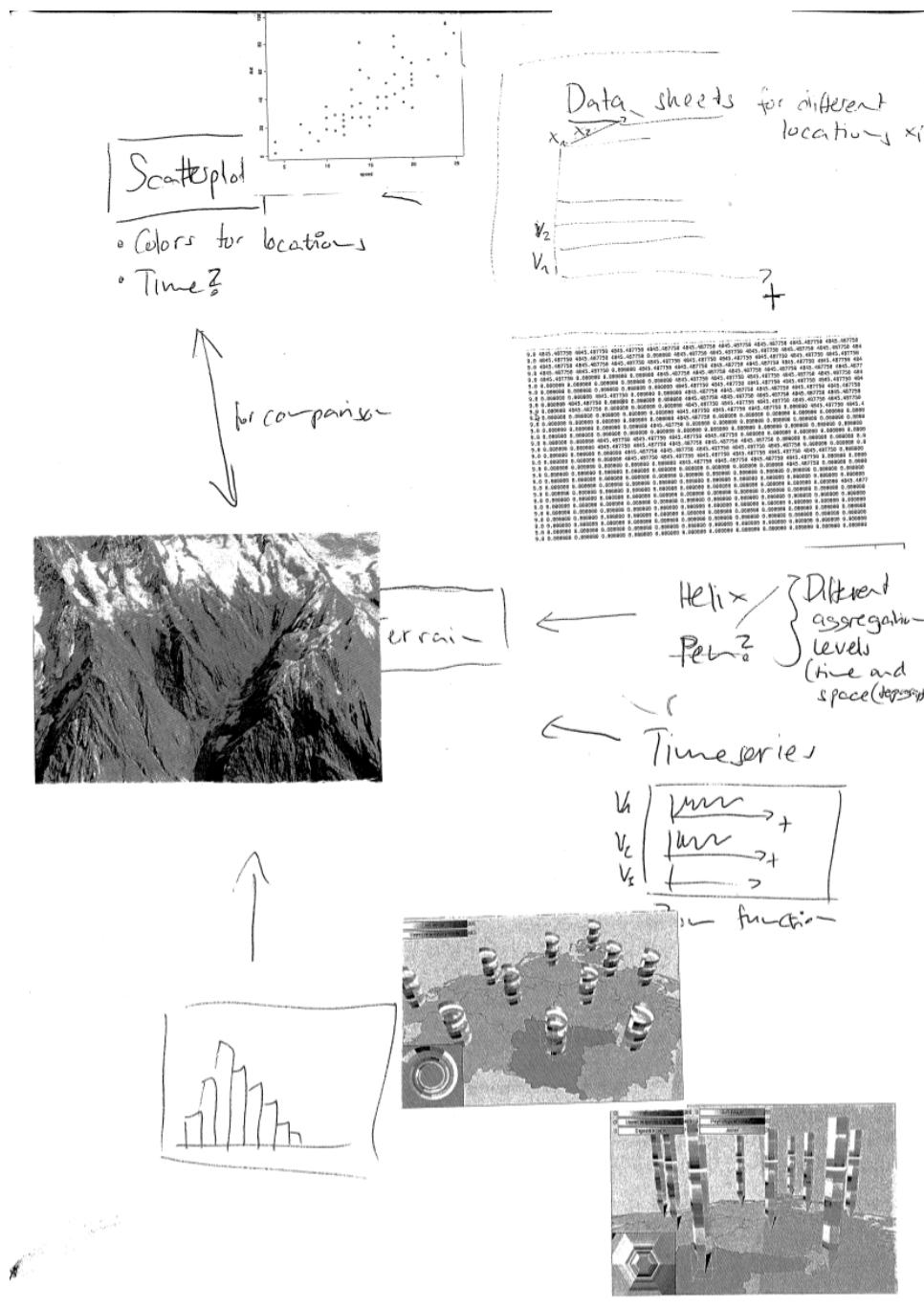


Figure A.10: Participant 1 GUI mock-up

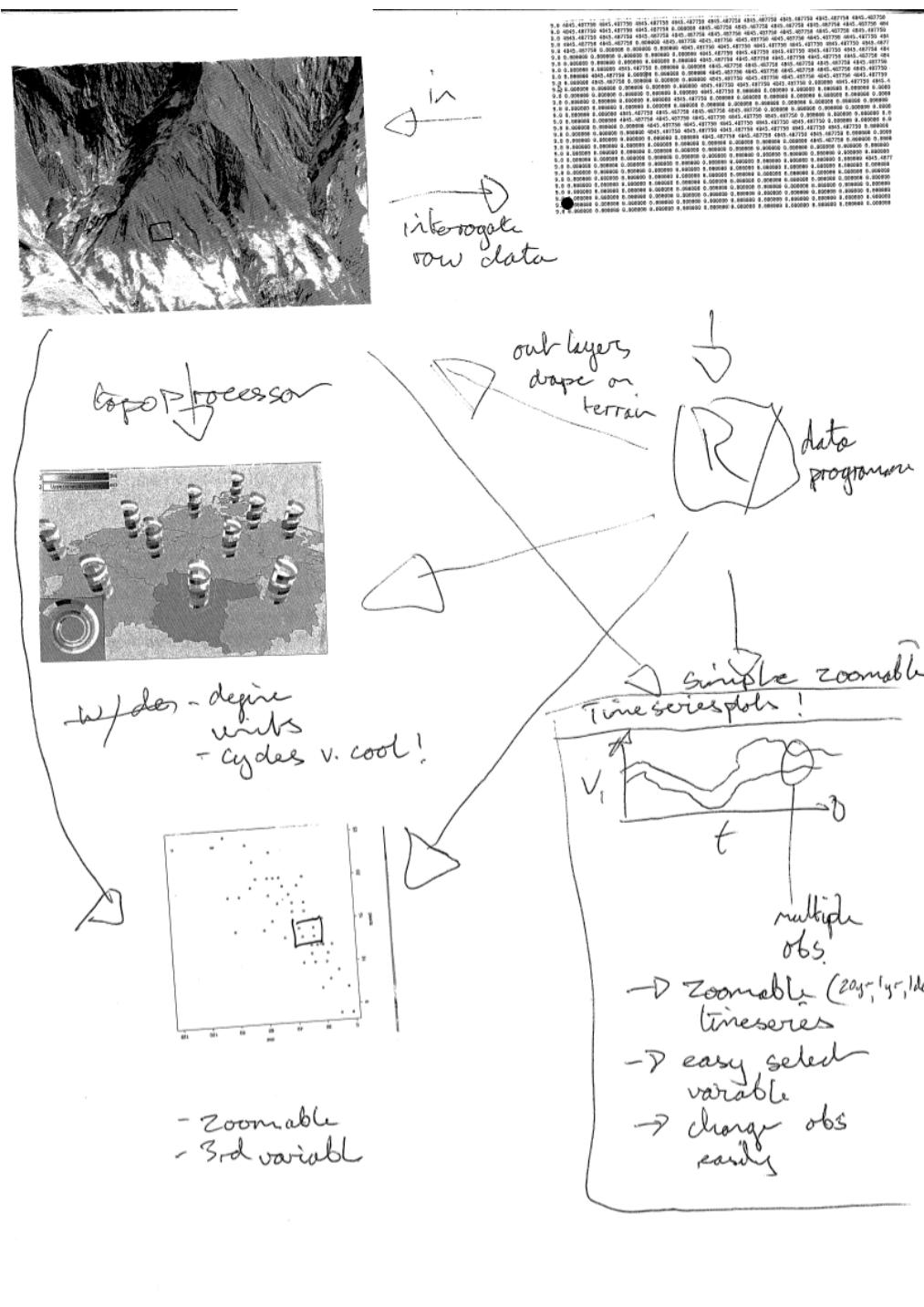


Figure A.11: Participant 2 GUI mock-up

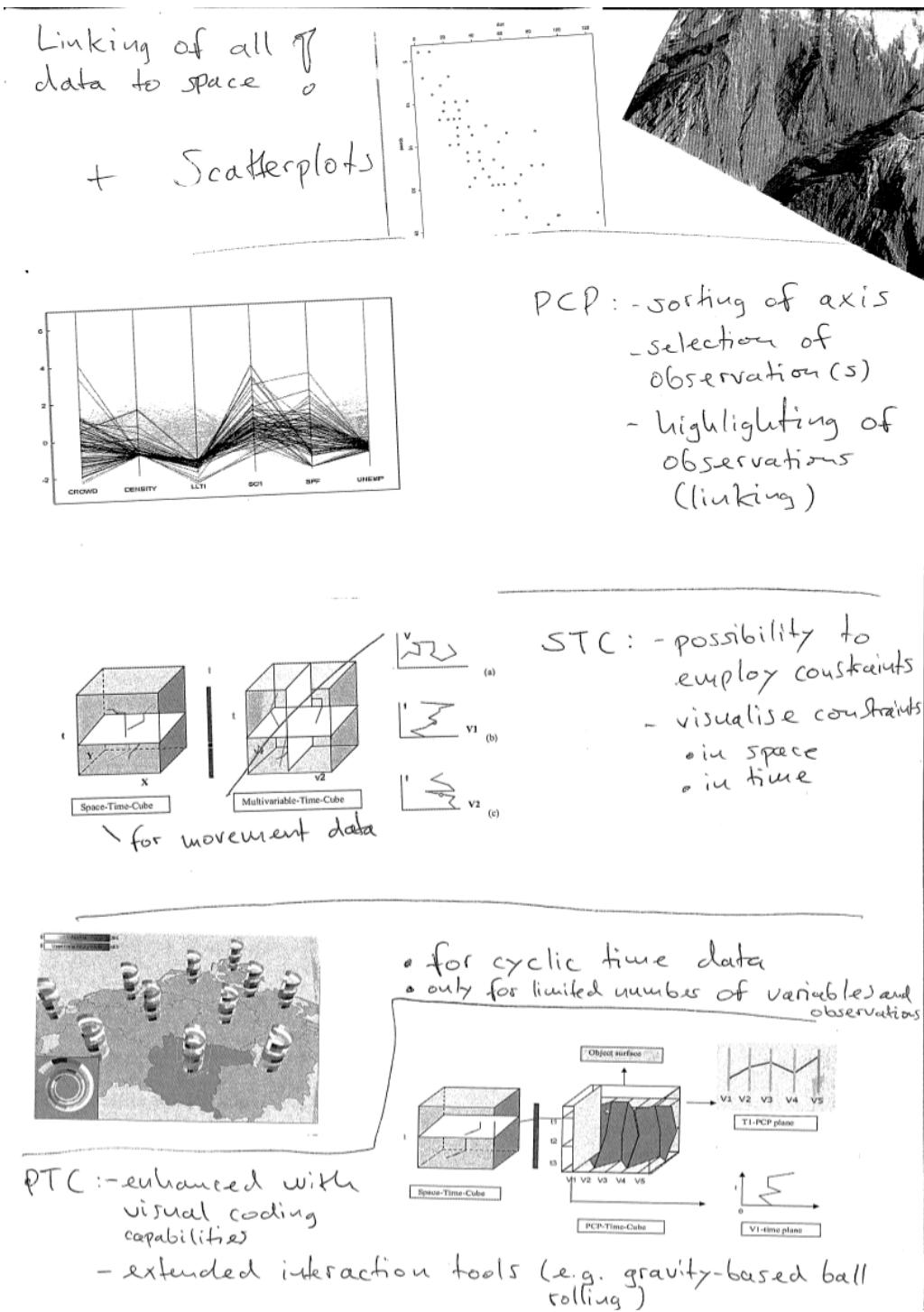


Figure A.12: Participant 3 GUI mock-up

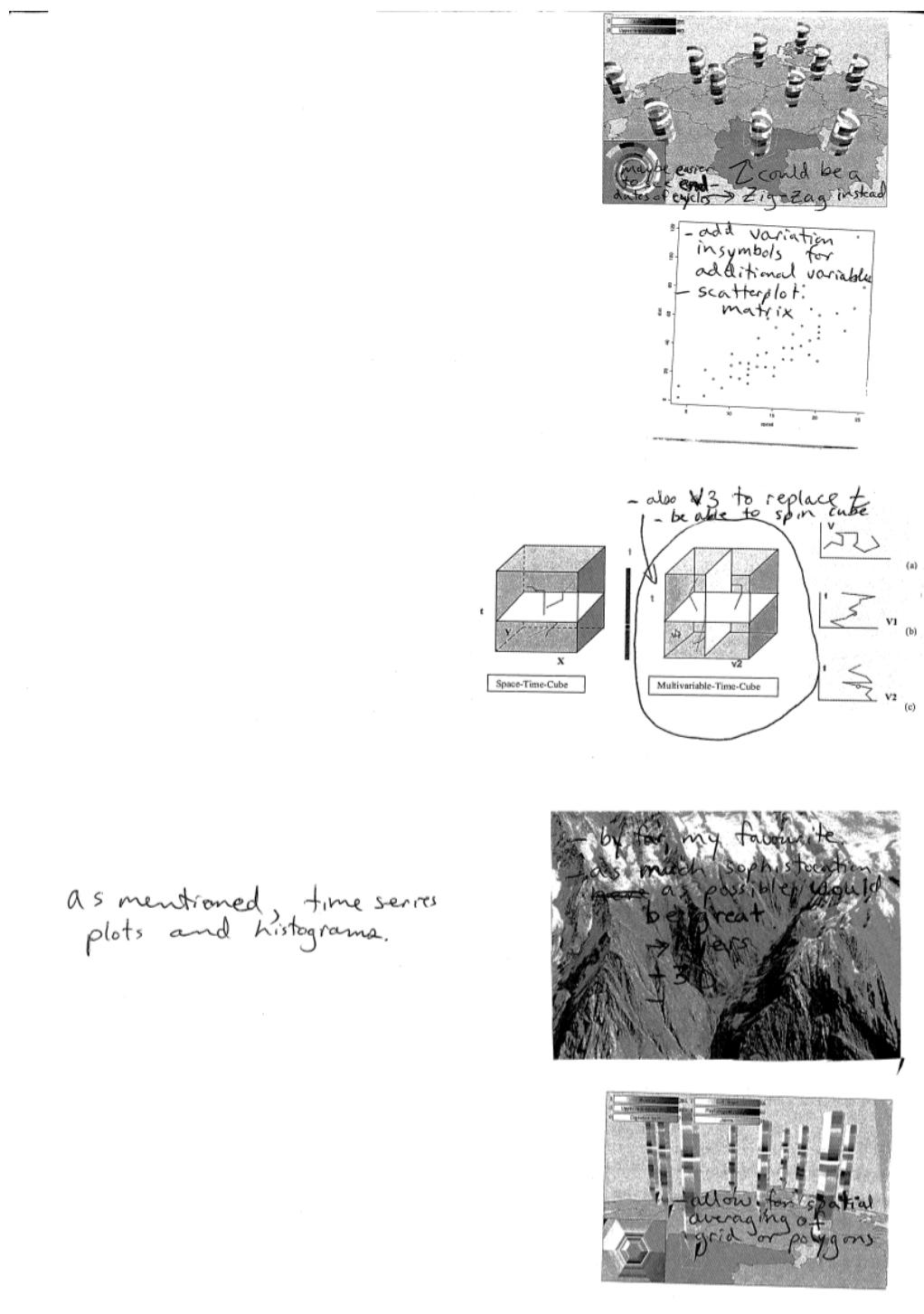


Figure A.13: Participant 4 GUI mock-up

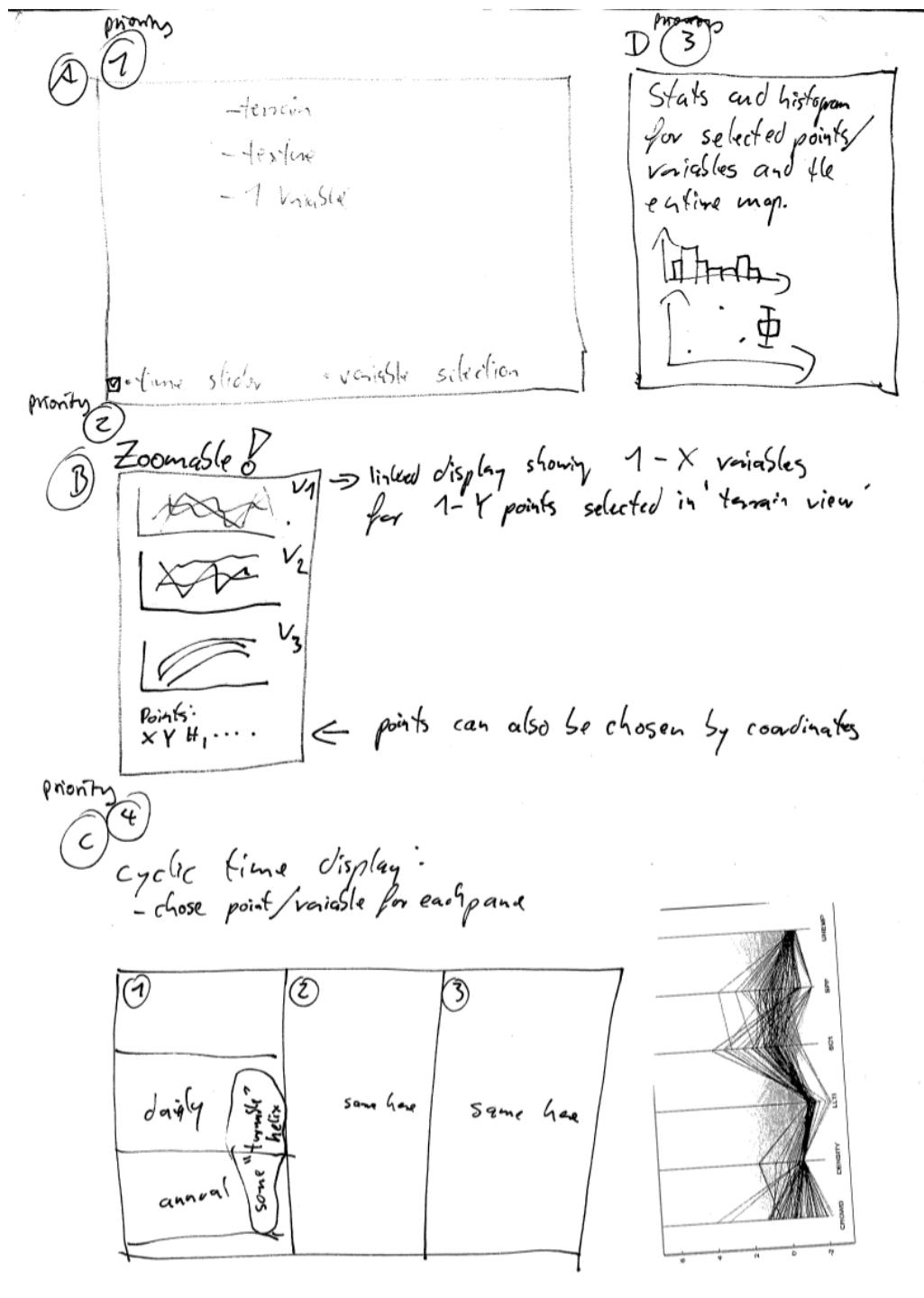


Figure A.14: Participant 5 GUI mock-up

A.7 Expert Interview Script

Expert Testing – Qgis multispatial / multitemporal plugins

1. Preliminary phase

1.1. Welcome

Speak whichever language you prefer

1.2. Background:

a) How many years of experience do you have in the following area:

- a) Working with GIS
- b) Working with QGis
- c) Working with Environmental data
- d) Working with temporal data

b) How often do you :

- a) Work with GIS
- b) Work with QGis
- c) Work with Environmental data
- d) Work with temporal data

c) From 1 to 10 how experienced are you in this field

2. Multi temporal tasks

2.1. Start Qgis Trunk using the link on the desktop , Get used to the interface, layer properties (double click or right click a layer) and the plugins menu.

2.2. Load the following datasets using the plugin MultiView Loader:

- SnowdepthS0002, SnowdepthS0003, SnowdepthS0004: ALL files
- TODO (which data)

2.3. Study the datasets using only the layers visibility switches

- Explore data and possible settings
- Do you notice a temporal pattern?
- What pattern is it?

2.4. Study the datasets using only the MultiView TimePlot

- Explore data and possible settings
- Do you notice a temporal pattern?
- What pattern is it?
- What is this best suitable for?

2.5. Study the datasets using only the MultiView HelixView

- Explore data and possible settings
- Do you notice a temporal pattern?
- What pattern is it?
- What is this best suitable for?
- Would you rather use Helix or TimePlot

2.6. Compare the three methods

- Which was more intuitive
- Which was more effective
- Was the most intuitive tool the most effective as well? Or after getting used to them, the more complex visualizations are more effective
- What is this tool best suitable for?

2.7. Study the datasets using Scattergram

- TODO (which data)
- Look for outliers/anomalies/big value. Are those anomalies or big values?
- How can you tell?
- Do you notice a spatiotemporal pattern between the two variables?
- What pattern is it?
- What is this best suitable for?

3. Globe Tasks

- Open Globe_qgis and start the globe plugin
- navigate around
- perform the same procedure as in 2.3

4. Evaluations**4.1. DataLoader:**

- general feedback

4.2. MultiView

- Does the coloration of the checkboxes and names disturb?
- Are the help texts helpful?
- Where the warnings visible enough
- General feedback

4.3. MultiView-TimePlot:

- Do you like the placement of the time labels on the axis?
- Zooming is OK
- General feedback

4.4. MultiView-Helix:

- Do you like the placement of the time labels on the axis?
- General feedback

4.5. Scattergram:

- Do you like the placement of the time labels on the axis?
- Are the controls understandable
- General feedback

4.6. Globe:

- Are the controls understandable?
- is it easy to read the information?
- General feedback

5. Conclusion

- What did you like/dislike most?
- How much (1-10)?
- Why?
- What did you miss?
- General feedback

Personal Declaration:

I hereby declare that the submitted thesis is the result of my own, independent, work. All external sources are explicitly acknowledged in the thesis.

Marco Bernasocchi