Abstract

Interactive outlier detection in latent variable models using virtual reality.

Author: Tore Aurstad, Ms. Tech. graduate student, Algorithm and Visualization Group, Department of Computer and Information Science (IDI), Norwegian University of Science and Technology (NTNU).

This report investigates different methods in computer graphics and virtual reality that can be applied in a system that provides analysis for the changes that occur when removing outlier points in plots that display principal component analysis. The main result of the report is the conclusion that the use of animation gives a better understanding for the movement of individual points in the plots, before and after removal.

Summary

This report investigates methods in *computer graphics* and *Virtual Reality* or VR, which provide a better analysis of *multivariate data*. Methods for enhancing *principal component analysis* or PCA are investigated in particular. To research the different gains of introducing new methods in computer graphics and Virtual Reality for visualization systems with PCA, a *system* has been developed. This system provides basic PCA in 2D and 3D. A special task for this project is to animate the removal of *outlier points*. This has successfully been implemented in the system. The system can also filter out rotation using *Procrustes analysis*, a combined *transformation* that provides maximum overlap between two geometric configurations. The system handles basic data that has been sampled from *chemometric* experiments. *Chemometry* is also known as multivariate data analysis and it is a field of chemistry with strong linkage to statistics and computer science. The system does however work with multivariate numerical data alone, and can treat whatever multivariate data which needs to be investigated using principal component analysis. This report discusses the system which has been developed, and the results and conclusions that can be gained from it. The major findings of this report are the benefits gained from using *animation* to provide better understanding of the movement of data points in the *scores plots* of PCA, when removing outlier points. The reports discusses the systems abilities to inspect and understand changes of multivariate data in two or three dimensions, and the use of Procrustes analysis, which gives a better understanding of the *stretching* of the data that occur after the removals. The understanding of *dynamics* in multivariate data is clearly enhanced using animation. Use of *red-blue stereoscopy* together with *Polaroid glasses*, also gives a better depth view of the data points. Additional use of VR, for example use of data gloves, have not been implemented into the system, as the use of *spin plots* and standard input devices (mouse and keyboard) have provided sufficient and fast for the interaction in the system. The main conclusion of this report is that principal component analysis should use animation, spin plots and stereoscopy, and additional methods in VR and computer graphics. More research should be done to investigate additional methods of computer graphics and VR for chemometric visualization that can provide a better analysis. Since analysts have got different computer skills, multiple applications have been developed to provide a variable degree of interaction in two or three dimension for visualization of PCA on multivariate data in the system. This has created a flexible environment for analysts to work with their multivariate data.

Preface

This report is the written work by author Tore Aurstad, for the partial fulfilment of the degree *Master of Technology* in computer engineering, at the Norwegian University of Science and Technology (NTNU), spring 2005, in the course TDT4900. The work has been performed at the Algorithms and Visualization Group, Department of Information and Computer Science (IDI), NTNU.

The task of this project was defined by university lecturer Odd E. Gundersen, Algorithms and Visualization Group, IDI, NTNU, in collaboration with professor of chemometrics Bjorn. K. Alsberg, Chemometrics group, Department of Physical Chemistry, NTNU. The author contacted Gundersen and Alsberg, after choosing the project. The projected started 20.01.2005.

1.1.1 Acknowledgements

I would like to thank Odd E. Gundersen for proof-reading the report and giving advices throughout the entire project, and providing me with insight of the strategy of writing scientific reports, plus providing me with resources and pointers to information for relevant material in virtual reality or VR, and computer graphics (visualization). I would also like to thank Bjorn K. Alsberg for helping me to understand the nature of the task of this project, plus teaching me in complex mathematical models like principal component analysis, singular value decomposition or SVD and other relevant fields of chemometry, for the task at hand, and providing resources and pointers to relevant material in chemometry or multivariate data analysis. I would also like to thank my friends and my family for their added support.

To the future readers of this report, I hope this report is interesting reading material for all of you.

Tore Aurstad,
Trondheim, 06.06.2005.

Table of Contents

List of Figures

List of Tables

1 Introduction

This chapter will introduce the project, presenting motivation for the project in 1.1, research questions in 1.2, research approach in 1.3, requirements which are raised by the research questions from 1.4 and organization of the thesis in 1.5.

1.1 Motivation

Human observers have little problems visualizing objects in two and three dimensions. In particular fields of science, samples are taken in experiments with possibly more than three dimensions. If the fourth dimension is not the time dimension, more than three spatial coordinates are given and a direct visualization of the data points corresponding to the spatial coordinates is not possible. The field of chemometric is known as multivariate data analysis. In this field, large data sets with dimensionality much larger than three are difficult to comprehend for an analyst. The possible method is to view a subset of the data, for example choosing two or three dimensions at a time, but this will also mask information for the analyst. It is a much better approach to view all the data simultaneously, but in a more comprehensive and understanding way. Many methods already exist for viewing higher dimensional data simultaneously in either the 2D plane or the 3D space. This report will discuss a system that has been developed for using *principal component analysis* - PCA – to visualize the higher dimensional data. The particular new features of this system are the animation of removal of possible multiple outlier points (in use, the system accepts any point selected by a user) in the *scores plot* of PCA. The system uses also additional methods in computer graphics and Virtual Reality (VR).

1.2 Research questions

Based on the motivation in 1.1, the main research question is the following:

Q-1

In what way should a system for analysis of multivariate data, using principal component analysis (and possibly related methods in multivariate data analysis), utilize new methods of computer graphics and Virtual Reality (VR) to give the users optimal information viewing capabilities?

The main research question poses no trivial task. It is impossible for a human to directly visualize (in an abstract way) large data sets (larger than spatial dimensionality three), for example 30x200 (30 rows and 200 columns, which could be represented as 30 points in a hyperspace of 200 dimensions). The following questions follow up the main research question of this report.

Q-2
What combinations of hardware interfaces will provide the best information viewing capabilities for multivariate data?

Q-3
What are the qualitative changes of viewing the data, when using the added functionality of the new interfaces?

Q-4
What degree of knowledge of the information in the multivariate data should be expected, when using principal component analysis (PCA) with the new interfaces of the system to be developed?

Q-5
What degree of knowledge of the dynamics that occur in the multivariate data should be expected, when removing outliers of the score plot?

The research questions Q-1 - Q-5 cannot be answered until a system for performing PCA on multivariate data, supporting animation of removal of outlier points have been developed. This project represents pivotal work in this task at hand, and there are no existing systems that to the author’s knowledge implement exactly such functionality.

1.3 Research approach

The main research question can be solved in two different ways. Since this project consists of applied computer science, a system will be developed to investigate the possible answers of the research questions. It is possible to develop a large system that supports many different data formats and multivariate data analysis methods (methods of chemometry). It is also possible to use a wide variety of virtual reality input and output devices, available at NTNU. However, the author must develop the system on an individual basis (due to the demands for this particular project), there are no previous system that combines chemometry and VR / computer graphics, and the project is only lasting six months, which suggests that a small and specific system will be developed. The results and conclusions of this small system could then provide new ideas for future development of a larger system. An alternative solution for the main research task has been to use the open source code of *Scicraft*¸ which is a large general analysis tool for use in chemometry, to develop the animation functionality. The author has not selected this solution, since a small, but dedicated solution would be simpler to develop, test and reuse by future developers. A small prototype is also more suited to conduct specific research on, than using a massive integrated environment like *Scicraft.*

One further point is that the calculations needed for animating and performing PCA on large multivariable data sets are best solved in a small system, which demands less computer resources like system memory.

1.4 Requirements

The requirements for the system that must be developed for this project to answer the research questions in 1.2 can be summed up as:

R-1:
The system should be able to handle at least one format of data files for chemometric experiments, which means opening the data file, parsing the data into variables letting the user manipulate the data in a non-volatile way, i.e. remove data points without changing the data files.

R-2:
The system should be able to perform PCA on the variables from the data files. The system should let the user select the principal axes for PCA and orient the view of the data.

R-3:
The system should be able to remove outliers in the scores plots, and animate the changes that occur between before and after removal.

Additional details can be established for the system itself, using software requirements standard like [IEEE98], but the main requirements are summed up in requirements R-1 – R-3.

1.5 Organization of the thesis

This report is divided into several chapters and appendices.

- Chapter 1 *Introduction* contains the introduction.

- Chapter 2 *Theoretical framework* explains the relevant theory from virtual reality, computer graphics and multivariate data analysis or chemometry, especially PCA.

- Chapter 3 *Design of the system* discusses the conceptual design and construction of the system to be developed for this project.

- Chapter 4 *Implementation of the system* discusses the implementation of the system, and show figures showing screenshots of the system and discusses the applications in the system implementation.

- Chapter 5 *Results and evaluations* discuss the results from the implementation of the system, and include an evaluation of these results.

- Chapter 6 *Conclusions and further work* draws conclusion from the results and evaluations.

- Appendix A *Mathematical methods in PCA* quickly explains the mathematical details behind PCA. This appendix contains mathematic methods that PCA is based upon, and therefore is not included in the theoretical framework itself (which in this report takes a high-level perspective).

- Appendix B *Technical terms and abbreviations* lists technical terms and abbreviations in the report, together with a short explanation.

- Appendix C *Additional resources* explain the content of the CD-ROM, which this report contains.

- Appendix D *Software Requirements Specification (SRS)* is the IEEE 830 std. document for the system. It is included in the appendices, since it is a rather extensive SRS, and could interfere with the readability for this report if moved in front of Chapter 3. Please note that the SRS was developed before the system was designed in collaboration with Professor B. K. Alsberg.

- Appendix E *Code listings* lists the source code modules of the system. Additional details for each module are also included.

The source code itself, installation manual, system documentation, PDF file of the report, byte compiled modules, screenshots, videos, Application Programming Interface (API) documentation and user manual can be read using the CD-ROM included with this report. Refer to Appendix C for a complete overview of these resources.

As a special note, this project adopts many strategies in [HART98] as a guide for the research process.

2 Theoretical Framework

In this chapter, the theoretical framework in this project will be presented. This framework consists of relevant theory from virtual reality (VR) and computer graphics. Instead of making a division the two fields, theory belonging to the field of *visualization* is a better description of theory from computer science included in this report. The report will however divide VR into *desktop VR* [ROST02a] and *immersive VR* [ROST02b]. Desktop VR is applied in standard computer systems that uses low-cost equipment like stereoscopy (Polaroid glasses for passive stereo or LCD active stereo glasses), software providing computer graphics (animations, 2D and 3D scenes) or similar to create a *desktop virtual reality*. Immersive VR is applied in more expensive computer systems that uses additional head-mounted displays (HMDs), positional trackers and Data Gloves or similar to create and immersive, artificial or *immersive virtual reality.* Since this project should develop a system for multivariate data analysis, also known as chemometry [UIB05], relevant theory from this field will also be included. Especially PCA, the main focus of this report, will be investigated. Mathematical theory, which defines PCA, is included in Appendix A, *Mathematical methods in PCA.*

2.1 Introduction

In 2.2, principal component analysis will be explained. This method is an analysis method for multivariate data analysis which is a rotational and reduction transformation. Scientific visualization is discussed in 2.3. There are no previous systems that can provide the specific functionality that is required for answering the research questions of this report. A new system will therefore be developed in this project. The type of system to be developed is a scientific visualization system, using desktop VR. In 2.4, stereoscopy is discussed. This is a collection of method in computer graphics or with the help of special equipment to provide observers with depth vision when viewing a 3D scene on a 2D canvas. In 2.5, linear interpolation is discussed. This is a mathematical method that can be applied for animations.

2.2 Principal component analysis

Principal component analysis - PCA – is a method for analyzing multivariate data sets. The goal of PCA is to rotate the axis system for the multivariate data set in the *correct orientation* or *most explanatory way* to express the data. More detailed, the axis system in PCA is composed of *principal axes* which are sorted after their explanatory level of the *variance* in the data. PCA can be solved algebraic, but is used for describing multivariable data sets in the new, optimized axis system. Generally, the data will not be correctly aligned initially, and the rotation brings the axis system to the most descriptive orientation. After this rotation, individual plots can be drawn to visualize the multivariate data, along the principal axes. Each principal axis is given a percentage value, which describes the ratio of explanation each axis can provide of the cumulative (summed) value of explanation values to the other principal axes for the variance in the data. The principal axes are sorted into PC1, PC2 and so on, up to PC*n*, where n is the number of principal axes. After PCA is run on the multivariate data, all the principal axes are orthogonal to each other in the n-dimensional hyper plane [JOHN02].

An example of PCA run on a collection of six points in the 2D plane is shown in Figure 1.

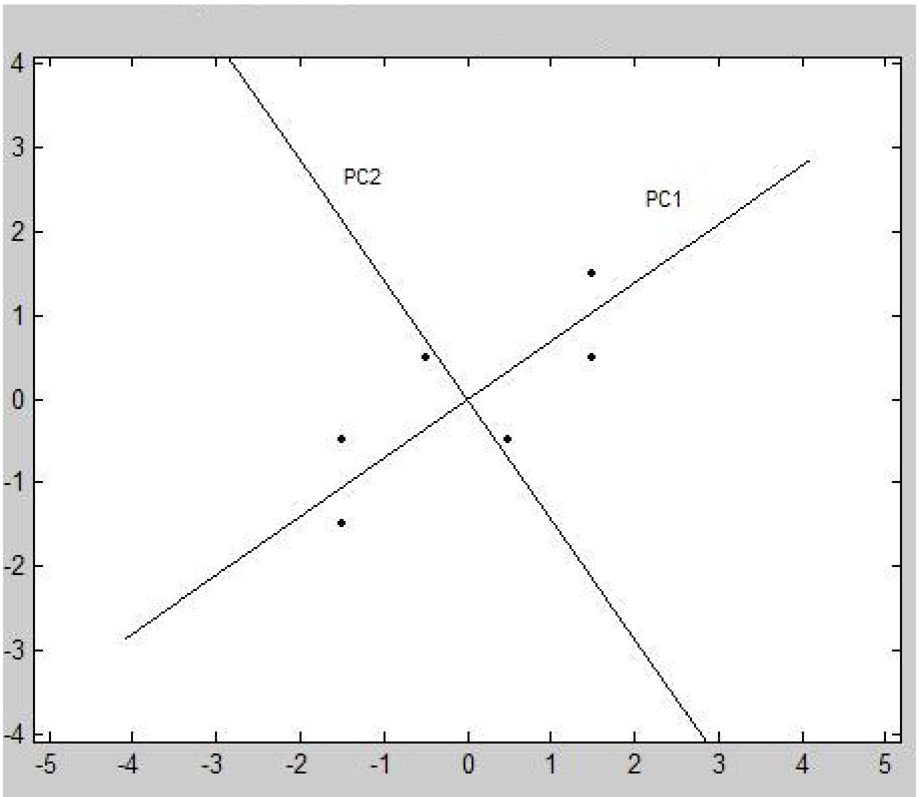


Figure 1 - PCA run on a dataset with six points in the 2D plane.

Figure 1 shows an initial test of PCA on simple 2D data with the PC1 and PC2 axes, which are the only two principal axes for the multivariate data. The number of principal axes are the same as the original dimensionality for these six 2D points (n=2). As Figure 1 shows, the PC1 axis is *oriented in the principal direction of variance* of the data set. The PC2 axis is orthogonal to the PC1 axis as in every case for principal axes. Figure 1 was generated with the Matlab software package from Mathworks[1] on a simple 2D data set from [RICH86]. A complete PCA run on the data would rotate the axes and the points in the correct amount, to align the PC1 and PC2 axes with the standard horizontal and vertical axes. Not every data set need to be rotated either, if there are no covariance or correlation between the dimensions of the data points. An example of this case is shown with another simple 2D data set from [RICH86] in Figure 2. In this case, the principal axes point in the standard horizontal and vertical direction. There are no specific orientations of the data points (*neutral alignment*).

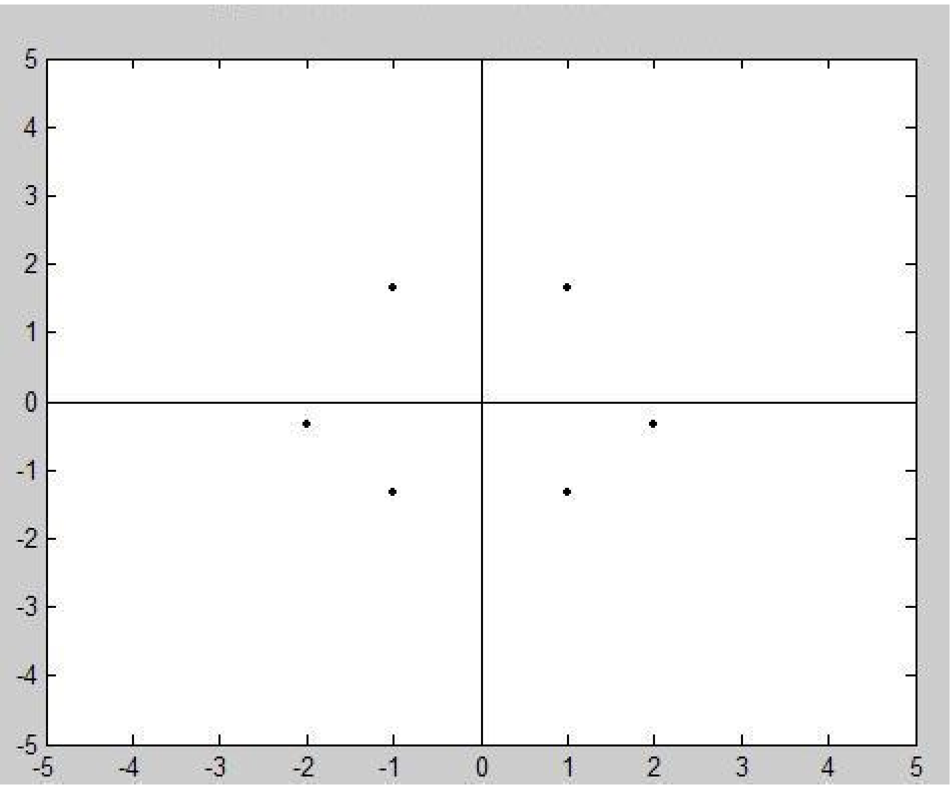


Figure 2 - PC-axes already aligned for data sets showing no covariance or correlation.

The real strength of PCA is shown in interpreting multivariate data with high dimensionality.

[1] Matlab from Mathworks – http://www.mathworks.com

2.2.1 PCA as a reduction transform

To understand the information in the multivariate data, the main correlation and covariance between the dimensions of each sample point should be calculated. For example, a data point in the 20-dimensional space will have a coordinate value consisting of 20 values, which are intersects between the dimensional axes and the data point. A collection of data points in the 20-dimensional space will typically be aligned more in certain directions than others, and after the PCA analysis, the explanation value of each principal axis for the data will be known. In visualizations of the multivariate data in the plane, the PC1 and PC2 axes will be used to plot the data along these axes. In the three-dimensional space, the PC3 axis will be added. The new intersects between the data points and principal axes define the *scores plot*, which most typically are a *scatter plot* (a scatter plot is a point cloud) of the coordinate value of each intersection. Figure 2 is an example of a scores plot for two-dimensional data. When scores plots visualize higher dimensional data in only 2D or 3D, some information in the data is lost. The calculation of this loss is calculated with PCA using the explanation value of each principal axis. This explanation value is calculated as follows [JOHN02]:

$$\lambda_i^{expl.} = \lambda_i / (\lambda_1 + \lambda_2 \ \ldots + \lambda_n) \qquad (2.1)$$

Eq. (2.1) states that the explanation value of each principal value equals the ratio of the summed explanation values for all principal axes. The following algorithm is used to calculate the explanation values:

ALGORITHM 2-1: Calculate explanation values for principal axes [JOHN02]

1. *For each column j* in multivariate array **A** with *M* columns, *sum (j)*, and find the mean vectors by dividing *sum (j) / N*, where *N* is the number of rows. Put the results in **M_A**, the mean matrix for **A**.

2. *For each row i* in **A**, subtract the **M_A** matrix (which has got dimensions Mx1). Put the results in the **C_A**, the centred matrix for **A**.

3. *Perform PCA* on the **C_A** matrix. *Find the eigenvalues* from the diagonal matrix from Singular Value Decomposition (SVD) of **C_A** or run an eigenanalysis on the covariance matrix of **C_A**.

4. Calculate the ratio of each eigenvalues compared to the cumulative sum of eigenvalues. Express this ratio as a percentage. This is the explanation value for the individual principal axis (on a row-basis).
End of Algorithm.

Algorithm 2-1 can also be used to show the scree plot of the eigenvalues, which is a graph method for showing the number of principal axes that should be taken into consideration during PCA on a multivariate set. Step 3 in algorithm 2-1 will be explained when discussing methods to solve PCA later in this chapter. A general scree plot is shown in Figure 3, modelled after the description of scree plots in [JOHN02]. The λ_i values are the eigenvalues resulting from running Algorithm 2-1. A typical scree plot looks like Figure 3. Since the principal values are sorted, the plot is a 2D graph which will decrease monotonic. An important feature of the scree plot is showing where the eigenvalues are *levelling out*, indicated by the dashed line in Figure 3, which occurs in the figure after the third principal component (axis). This feature of the scree plot is an indication for the analyst of the number of principal axes that should be included into consideration of the PCA on the multivariate data. Including additional principal axes will not give considerable added information. However, the explanation values must also be considered. A guide line for the number of principal axes that should be considered is to include at least 80-90% of the sum of explanation values.

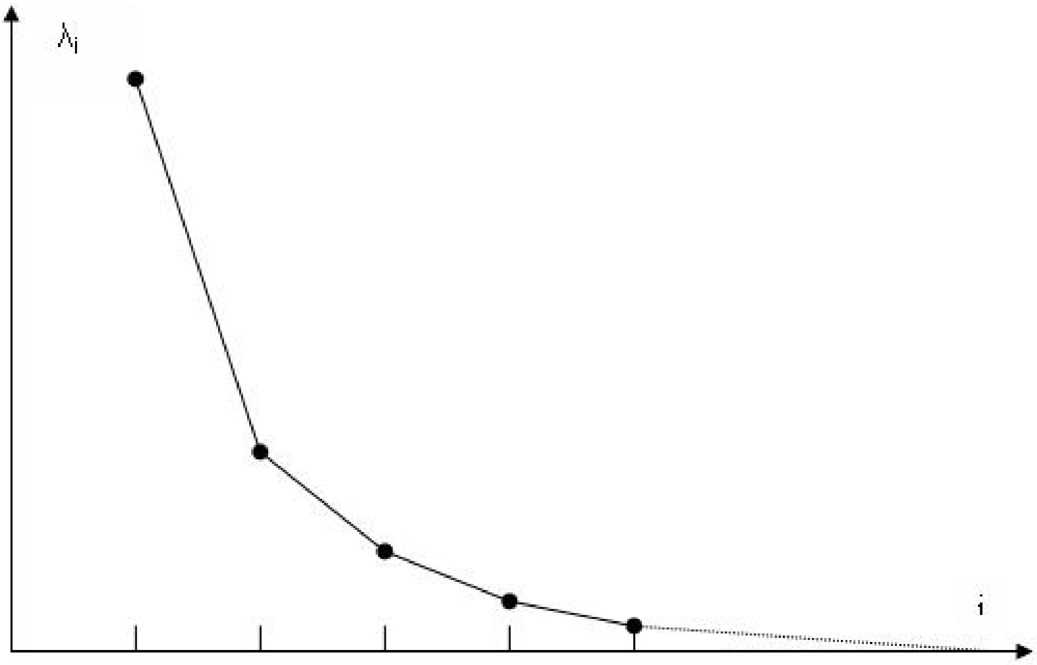


Figure 3 - General scree plot.

PCA is not only a reduction transform, but also a rotational transform. The change of the axes system will rotate the data, when the principal axes are aligned to the standard axes in 2D and 3D plots to visualize the scores plots.

2.2.2 PCA as a rotational transform

Multivariate data that have no covariance and correlation will not rotate the data from its original positions, since the principal axes are in such cases aligned to the standard axes which define the positions of the points in the first case.

An example of calculation of covariance will now be shown. The concepts of covariance and correlation will be explained in Appendix A. Consider the matrix **Y** from [RICH86]:

$$\mathbf{Y} = \begin{bmatrix} 2 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 3 \\ 5 & 4 \\ 5 & 5 \end{bmatrix}$$

The PCA analysis of **Y** is shown in Figure 1. First the matrix **Y** is centred to the origin, by subtracting the mean vector (step 1-2 in Algorithm 2-1). The centred matrix **Yc** from running **Y** through these steps can now be used to calculate the covariance. The result is:

$$cov(Yc) = \begin{bmatrix} 1.9 & 1.1 \\ 1.1 & 1.1 \end{bmatrix}$$

It is the indices of the matrix off the main diagonal which describes the amount of covariance in the matrix. Theses indices must be nonzero, or else there will be no covariance and correlation, and hence no need for rotation off the original standard axes (Example is given in Figure 2). To plot the principal axes of **Y**, the eigenvectors of the matrix transpose of covariance of **Yc** is used. The rotation matrix **G** for **Yc** is following the rule (**C_A** is a general centred matrix), based on the discussion of rotation for PCA in [RICH86]:

$$\mathbf{G} = [eigs\ (cov\ (\mathbf{C\_A}))]^{T} \qquad (2.2)$$

The function *eigs* will calculate the eigenvectors *V* of the argument, while the function *cov* will calculate the covariance matrix of the argument.

Additional calculations comparing the eigenvectors V (positioned in the columns of V^T) with the expression following expressions, which will find the rotational angles Θ:

$$V_x = V\cos(\Theta),\ V_y = V\sin(\Theta),\ \mathrm{V} = (\mathrm{V}_y, \mathrm{V}_x) \qquad (2.3)$$

Eq. (2.3) and by inspection of Figure 1, where V in this case is the individual rows of the eigenvector matrix of cov (Yc) shows that PC1 axis is rotated 35.0 degrees with the horizontal axis and the PC2 axis is rotated -55.0 degrees, making PC1 and PC2 axes orthogonal. The two principal axes of **Y** have planar vector orientation:

$$PC1 = (-0.819, -0.574),\ PC2 = (0.574, -0.819).$$

A geometric inspection with Figure 1 will confirm these vector orientations. Figure 1 also shows that both principal axes intersect with origin. This is a general fact; all principal axes intersect with the origin.

In general, PCA will work both as a reduction transform and orientation transform. PCA should only be considered as a valid reduction transform if the principal axes under consideration have cumulative explanation value larger than 80-90% of the total.

After a discussion of the scores plots in PCA, it is time to consider the loadings plots and residual plots. These are additional plots to describe the geometry of the hyper plane that are applied in chemometrics.

2.2.3 Positional description of PCA

In chemometry, three plots are used to describe PCA, together with the scree plot.
The multivariate data is described with the score matrix **T**, loadings matrix **P** and residual matrix **E**.

The scores matrix **T** is the basis for the scores plot, which has already been explained. The dimensionality of the **T** matrix is of size NxM, where N is the number of rows and M is the number of columns of the multivariate data (also an NxM matrix). If K represents the number of principal values to consider in the plots, the dimensionality will be reduced to NxK, where the analyst usually selects the principal axes to consider, usually the first two or three principal axes (PC1, PC2 and in 3D the PC3 axis). A plot of the values in the NxM or NXK matrix in the 2D plane or 3D space will show the score matrix in the score plot for the analyst.

The loadings matrix **P** describes the dimensions or columns of the multivariate data. This matrix describes the orientation of the hyper plane (the M- or K-dimensional hyper plane that is spanned by the principal axes) compared to the individual principal axes. The elements of the loadings matrix **P** is the directional cosine of the angles between the hyper plane and the principal axes. The loadings matrix **P** is of dimensionality MxM or MxK for the NxM matrix, where K still represents the number of principal values to consider in the plots.

The residual matrix **E** describes the distance from the individual data points to the common hyper plane. Usually this distance is very close to zero, and the residual matrix **E** can be ignored. The cases where the residual matrix **E** should be considered, is data gathered from process industry, which usually contains high levels of temporal variance. [UMET05].

The mathematical model for multivariate data in a general matrix **X** is described by the following formula [JOHN02]:

$$X = TP^{T} + E \qquad (2.2)$$

The geometric interpretation of the scores and loadings matrices are shown in Figure 4 [UMET05]:

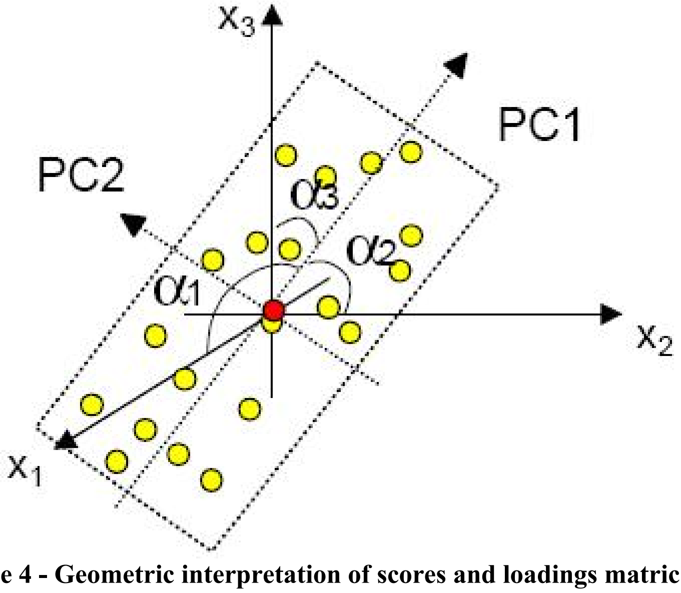


Figure 4 - Geometric interpretation of scores and loadings matrices.

In Figure 4, two principal axes define a 2D plane in 3D space, and the intersection between the data points and the 2D plane define the scores plot. The geometric definition of the loadings plot can also be seen; it is the directional cosine of the angles between all the principal axes with the individual data points (in this case, the loadings matrix **P** is of dimensionality MxK, 2x3 transposed which equals 3x2). The residual plot for matrix **E** is not shown in Figure 4, but the geometric interpretation is the distance from the common hyper plane (or just *plane* in 2D) between the points. In Figure 4, these distances equals zero, since all points intersects with the plane spanned by PC1 and PC2.

There are two basic methods to calculate the **T** and **P** matrices. Both methods first centre the multivariate data to the origin by subtracting the mean vector. The first method then calculates the covariance matrix of the centred multivariate data and performs an eigenanalysis.
Eq. (2.2) can be used to calculate the rotation matrix, which then can rotate the original multivariate data and calculate the scores matrix. The second method calculates the Singular Value Decomposition of the centred matrix, which will be explained further in Appendix A. The SVD of the centred matrix calculates three matrices U, S, V, which are orthogonal, diagonal and orthogonal matrices of the centred matrix. These matrices are the decomposition or factorization of the centred matrix after SVD is calculated. The scores matrix **T** loadings matrix **P** are calculated as [JOHN02]:

$$T = US \tag{2.3}$$

$$P = V^T \tag{2.4}$$

Eq. (2.5) - (2.6) show that SVD can easily calculate the scores and loadings matrices. The residual matrix **E** can be calculated from subtraction in Eq. (2.4).

2.2.4 Outlier points

In PCA, the orientation of the principal axes can be misaligned by outlier points. Usually, these points are non-correlated to the remaining data points, and can be removed to align the principal axes in a more correct way. An analyst should focus on the most isolated outlier points. The points distributed nearer to the midpoint of the data should not be removed.
The further from the origin an outlier point is situated, the higher impact that outlier has on the principal axes of the data.

Outlier points are usually divided into *strong* and *moderate* outlier points. The strong points belong to the scores or loadings matrix. The moderate outlier points belong to the residual matrix, and are recognized by high values in this matrix, describing a high distance between the common hyper plane defined by the principal axes and the corresponding data point.

A main task of this project is to develop a new system that supports removal of outlier points (or points in general), and is able to animate the changes in the scores plot before and after the removal. Removal of outliers in the loadings plots will not be considered in this project.

The animations in the system need to turn on or off rotation in the removal process. *Procrustes analysis* will be used to filter out the rotation. This method is explained in Appendix A.

Relevant theory in computer science for this project will now be discussed.

2.3 Scientific visualization

Scientific visualization is a field of visualization that focuses on research of systems for scientific calculations. A related field of visualization, *information visualization*, explains how users acquire knowledge from the information in the visualizations. This subchapter will discuss both fields and their relevance for the system to be developed.

2.3.1 Goals

Scientific visualization is oriented towards visualization of large scientific databases and applications. The data is often multivariate arrays. Since PCA is a central multivariate analysis method, this indicates that PCA is an important method also in scientific visualization. The goal of scientific visualization is suggested in [HAAS98]: *'Scientific Visualization has the goal to leverage existing scientific methods by providing new scientific insight through visual methods*'. The core goal of scientific visualization is according to this quote to give the analyst better insight, or increased level of information knowledge of their data, through the use of *visual methods*. In this project, animation, stereoscopy and spin plots will be used to provide new visual methods for the user. The main goal of the system, to be considered a scientific visualization system, should therefore be to provide better insight or knowledge of the multivariate data for the users of the system.

2.3.2 Additional methods for visualizing multivariate data

The system will probably use spin plots to visualize the multivariate data. Spin plots are scatter plots with the ability to zoom, rotate and pan. There are however alternatives to view multivariate data. These alternatives are basic methods in *data mining*, which are field of computer science focused on the *extraction of information* in data.

Parallel axes [INSE90] are able to visualize multivariate data without dropping information. The different columns are mapped as parallel, translated axes. Each row in the multivariate data represents as usual a sample multivariate data, and is plotted as a line segment that intersects the parallel axes. The intersections are usually scaled to [0, 1] and colour coding is frequently used to separate the lines from each other. This method is powerful in its capability to show the entire multivariate data simultaneously. A related method, *star plots,* share many properties with parallel axes. These plots are also known as *radar plots*, and show the dimensions or columns as rays emanating from a central point (star pattern). The individual samples are shown as edge segments intersecting the dimensions according to their values [SPENC01].

Additional methods for representing multivariate data is shown in [SPENC01], for example *focus-context*, multiple camera angles, *hyper cubes* and *mosaics plot*. The focus-context method shows an overview of the multivariate data, together with details. The hyper cubes and mosaics plot divide the data into further divisions in the 3D space or 2D plane. Multiple camera angles show many plots of subsets of the multivariate data. All these additional methods have strong capabilities to show large multivariate data. Parallel axes method is a very scalable method for large data sets. It is however a fact that principal component analysis (PCA) is able to investigate much larger data sets with less complex spin plot visualizations in either 2D or 3D, and still be able to describe most of the variance or information of the multivariate data.

2.3.3 Interaction models

Empirical experiments of user interaction and quality of scientific visualizations systems are presented in [HAAS98]. The experiments focus on the cognitive challenges that occur when users interacts with a scientific visualization system. These systems are usually complex, especially for untrained users. The experiments also show that users want a high image quality of the visualization, and a fast interaction with the system. These two goals for users are difficult to achieve. Large and complex scientific visualizations systems cannot provide high image quality and fast interaction. The best combination of the two must be found. However, users are different in the way they interact with scientific visualization systems. One group of users prefer to investigate the visualizations in a *holistic* manner that is done by viewing as much as possible of the data as the same time to understand the data at a high level, and then move the camera view. These users would rather prefer to be provided with high image quality and have fewer demands to interaction. However, these users also prefer to move the camera view in small amounts. Such users would clearly benefit from stereoscopy, to see the spatial extent of the data. The other group of users prefers to navigate through the data, to change the camera view often and have fewer demands of image quality.

The quality of a scientific visualizations system is measured in the level of quality of the data, image and interaction of the system. The data quality measures support for data formats and data integrity and precision. The image quality measures colour and screen resolution, and rendering quality. The interaction quality measures the support for interaction devices and the ease of use. [HAAS98]

The next sub chapter will discuss stereoscopy and stereo projection.

2.4 Stereoscopy

Stereoscopy provides *stereo vision* in otherwise flat viewing panels, like a computer monitor. The method simulates the stereo vision observed in a real environment, e.g. when a human walks in nature and uses the visions of the left and right eye to melt together a *binocular view* (or stereo vision) of the nature scene. Since the eye pupils are separated around 6,5 cm, the different view origins for each eye gives stereo view up to a distance around 30 metres. Most predatorily animals share this ability with humans to view in stereo vision.

To give the users stereoscopy in a scientific visualization system, *passive* or *active* stereoscopy can be used. Passive stereoscopy uses colour coding for a 3D scene on the 2D monitor to represent different views for the left and right eye. The effect is much more easily seen with Polaroid glasses. The system will use most likely use this cheap method. A more expensive solution is to use LCD shutter glasses in active stereoscopy, which are synchronized and applied voltage to block the view for each eye alternately, with the refresh rate of the monitor or visualization system in general. LCD shutter glasses are much more expensive than Polaroid glasses. The halved refresh rates also result in a darker scene compared to the one which uses Polaroid glasses.

The calculation of the stereo projection for each eye defines the colour coding or calculation of the individual scenes to show to left and right eye for an observer. These formulas are as follows [VINC95]:

$$\frac{x_{pl}}{(x - \frac{S_e}{2})} = \frac{d}{(z + d)} \qquad (2.7)$$

$$\frac{y_{pl}}{y} = \frac{d}{(z + d)} \qquad (2.8)$$

$$\frac{x_{pr}}{(x + \frac{S_e}{2})} = \frac{d}{(z + d)} \qquad (2.9)$$

$$\frac{y_{pr}}{y} = \frac{d}{(z + d)} \qquad (2.10)$$

Eq. (2.7) – (2.8) state the projected x and y coordinates of the vertices in the scene for the left eye pupil. The z value stands for the z depth of each object in the scene. The d value is the projection plane. Eq. (2.9) – (2.10) state the projected x and y coordinates for the right eye. Note that the left and right eyes share the same projected y values, while the left eye pupil will observe a translation to the right, and the right eye pupil will observe a translation to the left.

The S_e value is the eye separation between the pupils. These formulas can be proven using geometric congruence. An example is given in [WATT98].

The next subchapter will discuss linear interpolation, for use in animation.

2.5 Linear interpolation

Linear interpolation is a straightforward method for use in animation. It can be expressed as [VINC95]:

$$V(t) = (1-t)\, V_1 + t\, V_2,\; t \in [0, 1] \tag{2.11}$$

The parameter t will vary from 0 to 1. Eq. (2.11) expresses a parameterized line segment, starting at v_1 and ending in v_2. The number of time steps for the animation will decide the rate at which the position of the object following a line expressed as in Eq. (2.11) is moving. Usually, a number of time steps corresponding to at least 12 frames per second would give the user a continuous impression of the movement, and not sequential (time-lapsed).
Linear interpolation is usually performed in all three directions when animating in 3D space. This gives the following equations [VINC95]:

$$X(t) = (1 - t)\, X_1 + t\, X_2 \tag{2.12}$$

$$Y(t) = (1 - t)\, Y_1 + t\, Y_2 \tag{2.13}$$

$$Z(t) = (1 - t)\, Z_1 + t\, Z_2 \tag{2.14}$$

The animation will also make use of Procrustes analysis to remove unwanted rotation when running animation between the states before and after removal of outlier points in the system. This mathematical method and covariance, correlation, eigenanalysis and SVD is explained in Appendix A. The next chapter will discuss the design of the system. The software requirements specification (SRS) for the system is shown in Appendix D.

3 Design of the system

The design of the system which will be developed in this project will be established in this chapter. The conceptual design is discussed in 3.1, which presents the main components of the system. The parser component of the system is constructed in 3.2. In 3.3, the mathematical handler of the system is explained. The visualization handler, which will handle the graphical components of the system, is presented in 3.4. This part includes a visual prototype of a typical user interface for basic PCA analysis. In 3.5, the software design of the system is explained, which explains which software development tools will be used to implement the system.

3.1 Conceptual design

An overview of the core components of the system is shown in the *UML sequence diagram* in Figure 5:

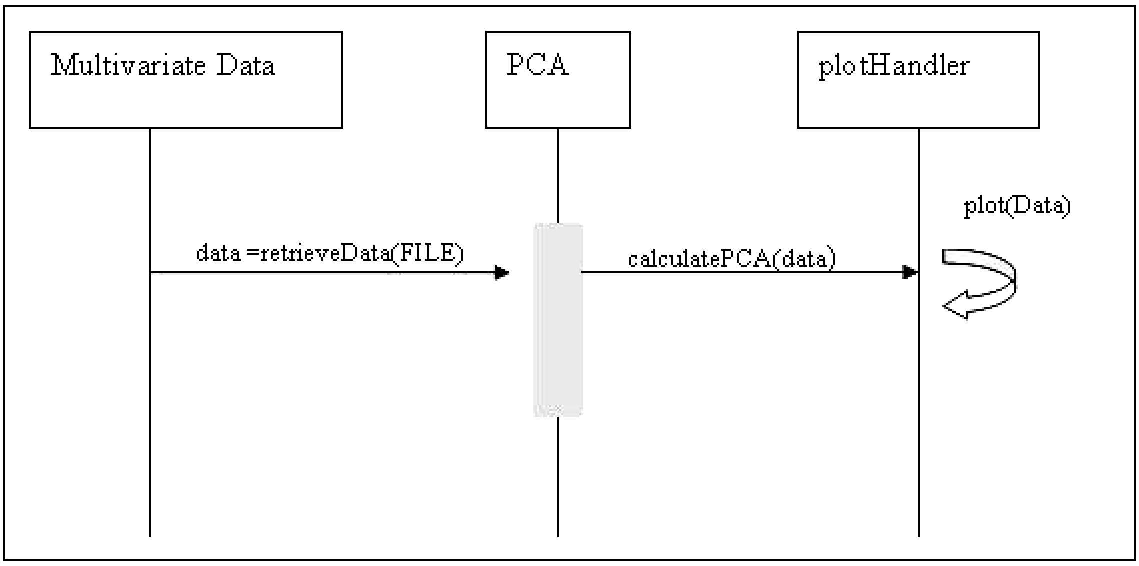


Figure 5 - Sequence diagram of PCA computation.

The multivariate data will be stored on a file, and the system will retrieve this data with a *parser* to interpret the file contents and store the components of the multivariate data into variables in memory. A *black box* method will then perform PCA (and Procrustes analysis) on the retrieved variables. The *plot handler* in the system will visualize the processed variables from PCA in the different plots for the system.

The following plots will be available in the system:

- Scree plots
- Scores plots
- Loadings plots

The scores plots will also support animation and removal of (outlier) points.

The following system components will be developed for the system:

- *A parser* to handle the input files for the system
- *A mathematical handler* for calculating PCA of the parsed data
- *A visualization handler* for visualizing the PCA data

Each component or module can consist of several subcomponents or functions.

The parser of the system will be explained next.

3.2 Parser

The parser will provide the data input functionality of the system. The parser will only support a special *Octave formatted* data files. This format is used by the Scicraft program. Parsers are available with Scicraft, but the system will be developed independent from the Scicraft source code. The parser is built to recognize a reduced octave format, and will be tested with example files from Scicraft. The following files will be tested:

- ampicillin_small.oct
- colon.oct
- cushings.oct
- ovarian.oct

These files contain multivariate data in arrays, but also string lists and scalars. Only arrays will be visualized in the system. The common file structure for these four Octave files is as follows:

Field	Syntax
Definition of variable:	
Name field	name: X
Type field	type: matrix | scalar | string
Rows of variable X	U
Columns of variable X	V
Value fields	
UxV fields of the same type, separator = text space, row change = line break.	

Table 1 - Reduced octave file format.

The parser must interpret the data of the files following the reduced octave file format. The parser must also handle the basic I/O stream and interact with the system.

The mathematical handler will be discussed next.

3.3 Mathematical handler

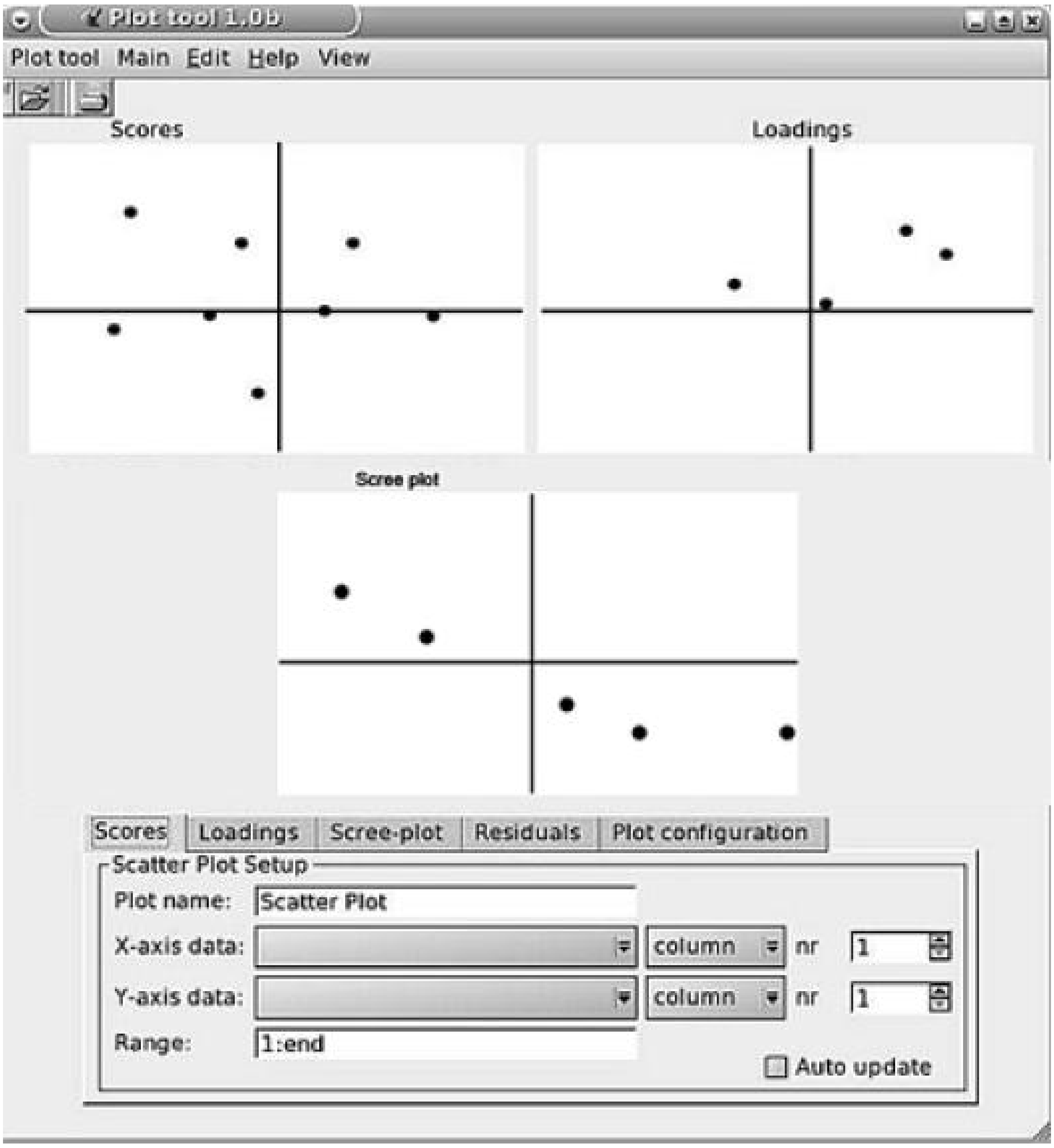
The mathematical handler will calculate PCA for the multivariate data, on variables received from the parser. The calculation of PCA should use SVD, since this is considered a fast method. The following functionality should be included:

- Calculation of SVD on the data
- Centring the data to the origin by subtracting the mean
- Calculations of the dimensions of the data
- Calculation of Procrustes analysis (rotational transform)

The calculation of the scores matrix, loadings matrix and scree matrix follows from the SVD calculation. These calculations can also be done in the mathematical handler.

3.4 Visualization handler

The visualization handler will visualize the data from the calculated PCA data and provide the user with the scores plot, loadings plot and scree plot. It will also provide the basic user interaction, enabling the user to select principal axes to view the data, pick objects to view the coordinates, rotate, zoom and pan the plots, pick objects for removal of outliers in the scores plots, choose to animate the shift in data positions in the scores plot, enable on and off red-blue stereoscopy in the system and select Procrustes analysis filtering. A visual prototype of the visualization handler is shown in Figure 6:



Scatter Plot Setup

Plot name: Scatter Plot

X-axis data: column nr 1

Y-axis data: column nr 1

Range: 1:end

☐ Auto update

Figure 6 - Visual prototype of the visualization handler.

Figure 6 shows the visualization handler will show the scree plot, loadings plot and scores plot simultaneous. The user will be able to select the data for the two axes in the 2D case and the three axes in the 3D case (not shown), and also select the array of the multivariate data set to analyze. The finished implementation will also include selection of Procrustes analysis filtering and a button to animate the removal of outliers in the scores plot. Additional changes can also be expected from this visual prototype, but the prototype is representative for the graphical design of the system. The visualization handler should provide functionality for both 2D and 3D viewing of the multivariate data, and support use of stereoscopy and animation.

3.5 Software design

The system will be developed with software packages that the Scicraft team uses. This selection makes the system easier to maintain and further develop by the Scicraft team. The software packages are available free of charge with the General Public License (GPL) for open source software.

The system will be implemented in the Python[2] programming environment [PYTH03]. The system will consist of several program modules and multiple runtime modules that will provide the user with a flexible choice of different applications with different degrees of level of interaction. The system will use the Python extension Numerical Python[3]. This extension provides the necessary PCA functionality like SVD to calculate the scores matrix and loadings matrix for multivariate data, and calculate the scree matrix. The system will use The Visualization Toolkit [VKT04], VTK, to provide the visualization functionality of the system. There are program bindings between VTK and Python in the PyVTK package. The graphical user interface (GUI) will be developed with Qt[4] Designer and written in the Qt programming language. PyQt provides program bindings between Qt and Python. The entire system will be developed on the Linux platform, using Linux Debian[5] as the operating system.

The next chapter will present the resulting system and discuss the implementation.

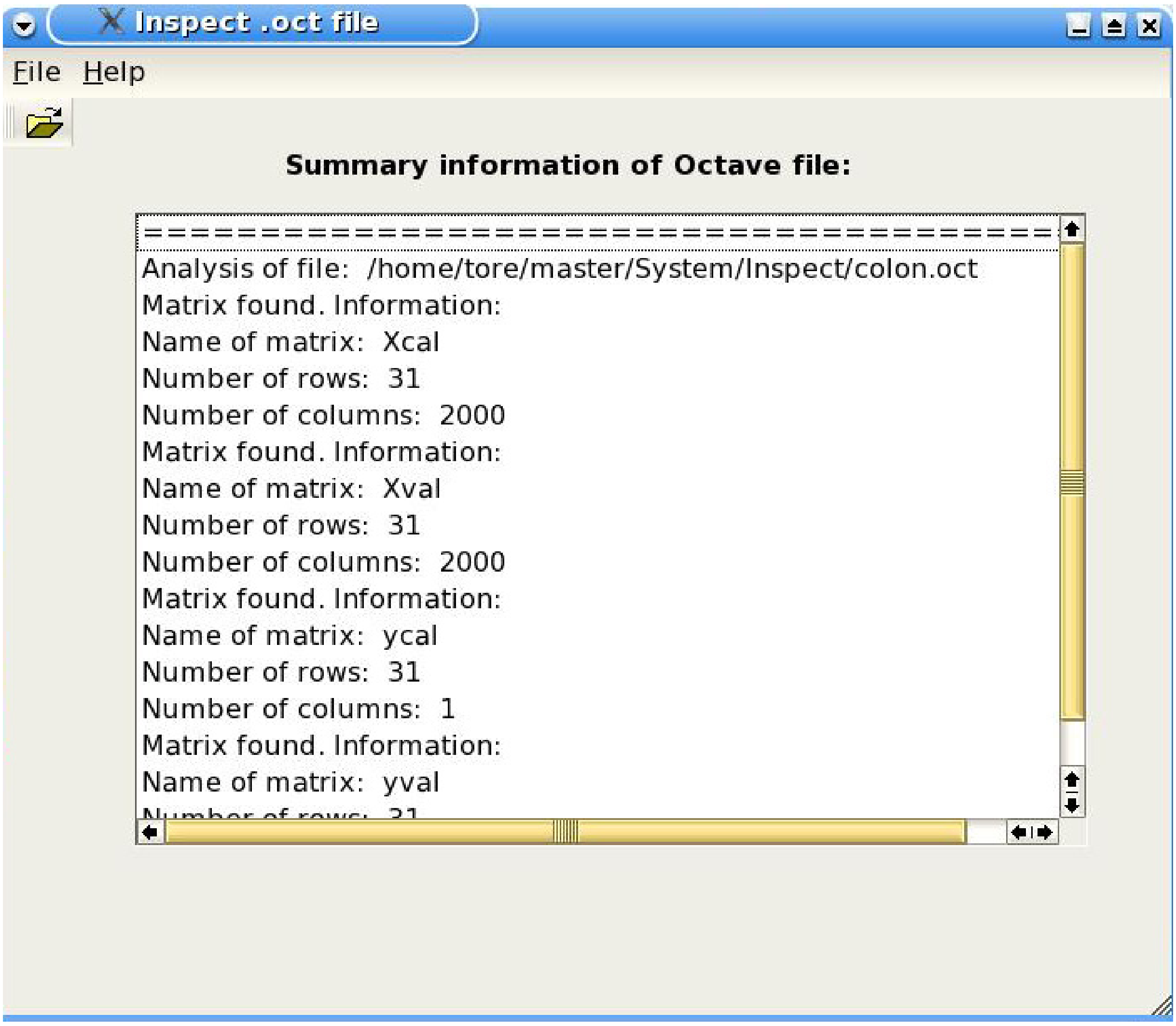
[2] Python website – http://www.python.org
[3] Numerical Python website – http://sourceforge.net/projects/numpy
[4] Qt website – http://www.trolltech.com
[5] Linux Debian website – http://www.debian.org

4 Implementation of the system

The system implementation will be presented in this chapter. In 4.1, the parser in the system is presented and explained. In 4.2, basic applications that provide 2D and 3D PCA analysis in the system are presented. Applications with 2D and 3D PCA analysis and additional VR support are presented in 4.3. The VR support is of type desktop VR and consists of red-blue stereoscopy, animation and spin plots.

4.1 Parser

The parser inspects the files in the reduced Octave format. A runtime module (application) has been developed to provide a front-end GUI to open files in and display the parsed information of the files. A screen image of the front-end parser is shown in Figure 7:



```
=========================================
Analysis of file:  /home/tore/master/System/Inspect/colon.oct
Matrix found. Information:
Name of matrix:  Xcal
Number of rows:  31
Number of columns:  2000
Matrix found. Information:
Name of matrix:  Xval
Number of rows:  31
Number of columns:  2000
Matrix found. Information:
Name of matrix:  ycal
Number of rows:  31
Number of columns:  1
Matrix found. Information:
Name of matrix:  yval
Number of rows:  31
```

Figure 7 - Parser of the system.

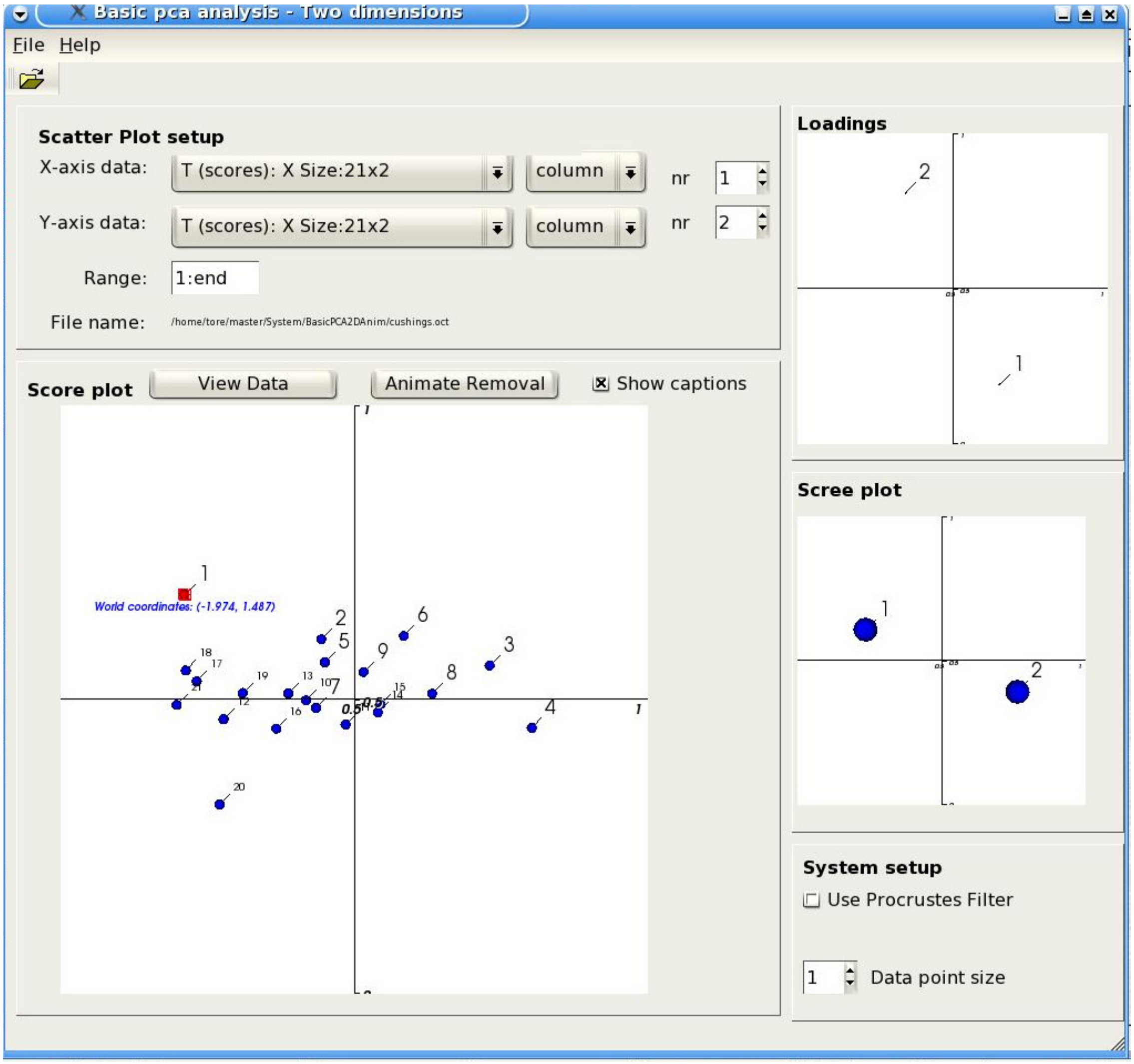
The parser application in Figure 7 shows the parsing of the file *colon.oct*. This file contains several arrays, scalar values and string lists. It is the arrays that will be visualized in the system. The user selects the file in reduced Octave format to analyze, and the parsing process is run automatically, listing the contents of the file in the list box of the application.

The parser and its front-end GUI application are written in the Python programming language. The GUI application was created in Qt Designer as with the rest of the GUI-based modules of the system, and converted into Python code with the *pyuic* converter (tool in PyQt software package). The additional code was programmed using the default Python integrated development environment (IDE), *Idle*[6]. This IDE have been used for the entire project.

4.2 Basic 2D and 3D PCA tools

The basic tools for running PCA on files in the reduced Octave format enables the user to select the input file, select the array to view and select the principal axes for the horizontal and vertical axes (x-axis and y-axis), and in the 3D case the z-axis. The user can also zoom, pan and rotate the plots (also in the 2D case). One of the applications for 2D and 3D PCA analysis is shown in Figure 8.

[6] Idle is the default IDE for Python. http://www.python.org



Basic pca analysis - Two dimensions

Scatter Plot setup

X-axis data: T (scores): X Size:21x2 column nr 1

Y-axis data: T (scores): X Size:21x2 column nr 2

Range: 1:end

File name: /home/tore/master/System/BasicPCA2DAnim/cushings.oct

☒ Show captions

World coordinates: (-1.974, 1.487)

System setup

□ Use Procrustes Filter

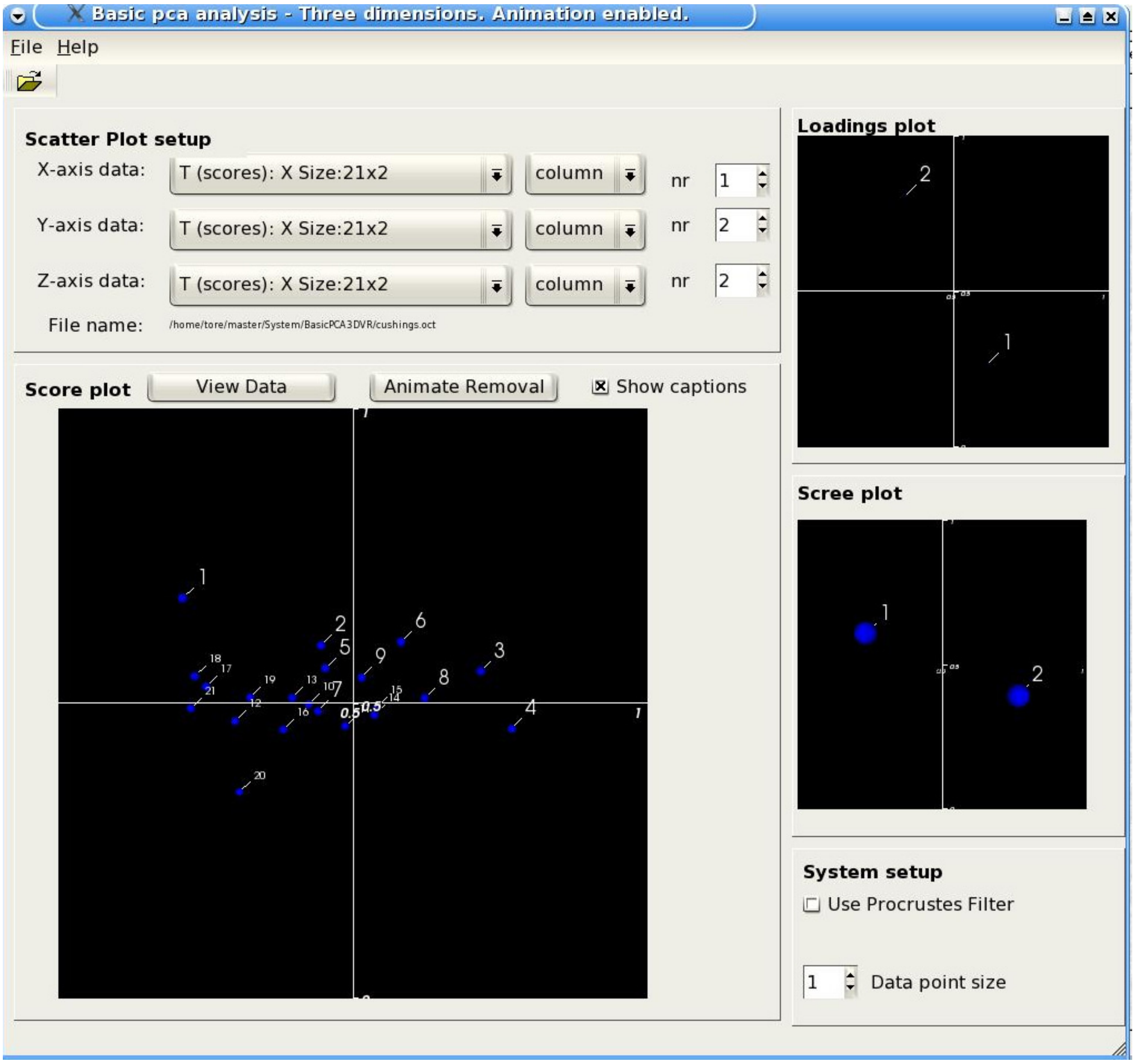
1 Data point size

Figure 8 - Basic 2D and 3D PCA analysis tool.

In this case, the application is a 2D PCA analysis tool with support for animation, Procrustes analysis filtering and removal of outliers in the score plot. The user selects the file to open and the parsing process is automatically run. The calculation of PCA for the selected array is also run. The user should first define the principal axes for the horizontal and vertical axes (and z-axis in the 3D case). The principal axes are preset to PC1 and PC2 (plus PC3 for the 3D case). The user then clicks the button *View Data* to start the visualization handler to visualize the processed and parsed PCA array in the scores plot, loadings plot and scree plot. The user can select the outliers for removal in the scores plot, by moving the mouse over the object and pressing the key button *P*. This runs the pick operand method. This should only be done in the scores plot. The user selects as many outliers as needed (it is not suggested to remove nearly all objects), which will be marked with red colour (default colour is blue). The user can see the world coordinates of each picked object when performing a pick operation. The user then clicks the *Animate Removal* button. This starts up the procedure for the animation. First, the corresponding rows in the input array are removed. Then PCA is run on the reduced array. If the user has marked the *Use Procrustes Filter* checkbox, rotation in the animation is removed. The animation cannot be interrupted and can be a computational demanding process and should require 3D acceleration. The animation will run at least two seconds, possibly longer if the computations are demanding. The frames per second are kept above a minimum of 12 frames per second (FPS) to provide continuous and not sequential animation. This will not happen in a system lacking 3D acceleration. The loadings and scree plots are animated. The reasons for this are the high demands of concentration required by the user to follow three animations simultaneous. Therefore, only animation of the scores plot is included. Animating three render windows makes also the animation much slower. To read the position of an object, it must be picked. To reset all three plots in the basic 2D and 3D PCA applications, click the *View Data* button.

4.3 Basic 2D and 3D PCA tools with VR

There is a collection of basic PCA tools in the *System* folder of the CD-ROM. The reason for this is to provide the user with a flexible selection of PCA tools. Some users will prefer to view the data only in 2D, while others will prefer 3D, and also animation. The only VR support functionality of the system (besides scientific visualization and animation, which can be considered as *desktop VR*) is red-blue passive stereoscopy. To activate or deactivate the red-blue passive stereoscopy, the user only has to press the key button *3*, to turn on or off this functionality. An example of a fully functional 3D PCA system with VR functionality is shown in Figure 9:



Basic pca analysis - Three dimensions. Animation enabled.

Scatter Plot setup

X-axis data: T (scores): X Size:21x2 column nr 1

Y-axis data: T (scores): X Size:21x2 column nr 2

Z-axis data: T (scores): X Size:21x2 column nr 2

File name: /home/tore/master/System/BasicPCA3DVR/cushings.oct

☒ Show captions

System setup

☐ Use Procrustes Filter

1 Data point size

Figure 9 – Basic PCA 3D tool with VR (red-blue stereoscopy) functionality.

Most users will probably use the basic PCA 3D tool with VR functionality, since this is the fully functional PCA tool available in the system.

4.4 Interaction functionality

This part will explain how to interact with the system, using keyboard and a mouse. The key buttons and mouse events recognized by the system are listed in Table 2:

User goal	**Interaction**
Animate removal.	Click the *Animate Removal* button. Wait for animation to end (About 2-5 seconds run time).
Activate red-blue stereoscopy.	Move mouse pointer over render window, press key button 3.
Activate Procrustes filtering in animations	Click to activate checkbox *Use Procrustes Filter.*
Choose PC axes.	Use the spin fields. Click up or down arrow to select PC axes for x-, y- and z-direction.
Deactivate red-blue stereoscopy.	Move mouse pointer over render windows, press key button 3 again.
Deactivate Procrustes filtering in animations	Click to deactivate checkbox *Use Procrustes Filter*.
Define data point size	Use spin box labelled *Data point size.* Click up and down arrow.
Exit the application	Click the close button for the window.
Open file	Click the open file button or select File and Open from the main menu and select the input file.
Pan the camera.	Press middle mouse button and drag in the pan direction. Then release.
Pick outlier points.	Move the mouse pointer over the objects in the scores plot to define as outlier points. Press key button P. Multiple points can be selected. To animate, see *Animate Removal* action above.
Read the position of a point (world coordinates)	Pick the point by pressing key P. To reset the marked points, click the *View Data* button.
Rotate the camera.	Press left mouse button and drag in the rotation direction. Then release.
Show PCA for selected array	Click the *View Data* button, after selection of file and array.
Select array	Use the combo box and select the array.
Zoom the camera.	Press right mouse button and drag upward to zoom in, and drag downward to zoom out. Then release.

Table 2 - Interaction functions of the system.

The system provides a special purpose analysis tool for multivariate data analysis running PCA with relatively few interaction methods. The next chapter will discuss results and evaluations of the system.

5 Results and evaluations

The system implementation has resulted in a special purpose system, which will be discussed in this chapter. There are no metric evaluations of the system in the results, but the results will be discussed and the qualitative level of main features of the product is presented in 5.1.

The evaluations of the system are summed up after testing of the system, performed by the author. The evaluations presented for the system in 5.2 are not user evaluations, but follow the framework for evaluation of scientific visualization systems in [HAAS98].

5.1 Qualitative results of the system

Results of important features of the system results are shown in Table 3:

System feature	Results
Animation (3D accelerated environment)	Animation runs fast and stable. No flickering. Precise linear interpolation.
Animation (3D non-accelerated environment)	Animation runs first at medium speed, then decreased. Finally, the system becomes little responsive. Flickering visible. This system clearly requires 3D acceleration.
Parsing	The parser handles reduced Octave formatted files from Scicraft. Additional files in the same format have not been tested.
PCA	The PCA plots have been compared with Scicraft. There are nearly identical plots, but mirroring of horizontal axes is visible. This is due to different software packages used for the calculation of PCA. Positional information correct (shown when picking).
Picking	The pickings of objects are precise when picked in front. When the objects (data points) are picked from a narrow angle, the wrong point is picked.
Principal axes	Flexible selection of principal axes, with error checking enabled. Presets set to PC1, PC2 (and PC3).
Procrustes analysis	Evaluation of the removal or filtering of rotation in the animation requires complex analysis, but the method clearly removes rotation.
Stereoscopy	Activating and deactivating red-blue passive stereoscopy is very easy, and fast. When zooming very near data points, the depth effect can give eye strain.

Table 3 - Results for main features of the system.

5.2 Evaluation of the system

The quality of the system will now be evaluated according to the framework of [HAAS98], as mentioned in the introduction of this chapter. The interaction, graphics and data quality is evaluated for scientific visualization systems with this framework.

5.2.1 Interaction quality

The system is special purpose and non-extensive. The interaction qualities are sufficient for a prototype system. It should be intuitive to use for most users already familiar with the Scicraft package, which is the case for the target users of the system (chemometry scientists). The GUI interfaces of the graphical modules of the system have been modelled to look similar to the Scicraft user interface. The system can quickly turn on or off red-blue stereoscopy and pick objects. The key bindings for red-blue stereoscopy are somewhat illogical (key button *3*), but is the default button to press when activating stereoscopy in the VTK rendering windows of the graphical PCA applications. The user can zoom, spin and rotate the camera in the render windows easily with a mouse. To zoom the camera in or out requires clicking and dragging the right mouse button, and dragging the mouse up or down to zoom in or out. Rotating the camera requires clicking the left mouse button and dragging the left mouse button in the desired direction to shift the camera position. Panning the camera required clicking the middle mouse button and dragging the mouse in the desired direction to pan the camera. The system accepts wrong inputs from the user and should not create critical errors, since the system runs in the Python interpreter environment. The available interaction features of the system should be possible to learn within a few minutes time of learning for new users to the system.

5.2.2 Graphics quality

The system has sufficient graphics quality for a prototype system. The data points of the spin plots are represented as blue spheres with white or black background, which is a sufficient representation of the point cloud. The system was developed using a resolution of 1200x1024 pixels and 32 bit colour depth. On a 3D accelerated system, the system runs at the desired frame rate, even when animating large plots. The system was developed on a computer lacking correct driver support for its ATI Radeon graphics board on the Linux Debian Software. The system have been tested on another computer (the workstation located in the visualization lab for the Chemometry Group at NTNU), which have got a NVIDIA graphics board with correct driver support. On this computer, the system runs stable and with the required speed for animations. The system should therefore be installed only on systems with proper 3D acceleration support, although basic functionality will still be available in non-accelerated environments, running at reduced frame rates.

5.2.3 Data quality

The data qualities of the system are sufficient for a prototype. It is possible to run files in the reduced octave format. The GUI front-end application of the parser shows correct parsing of the four files.

The next chapter will present the conclusions for this report and the project, and discuss suggested further work for the project.

6 Conclusions and further work

This chapter will discuss conclusions for the project in 6.1 and suggested further work in 6.2. The conclusions will answer the research questions of this report.

6.1 Conclusions

The project has resulted in the development and implementation of a special purpose multivariate data analysis or chemometry tool. This tool is innovative in the use of desktop VR to visualize PCA in 2D and 3D. The system of this project could be used at scientific congresses, to demonstrate how chemometry can benefit from methods in computer graphics and desktop virtual reality, according to Professor Alsberg. The report itself can be used as a resource for science projects for graduate students and scientists working with projects which are researching use of desktop VR in PCA. Answers to the research questions from 1.2 will now be provided. These questions will be repeated in this part of the report.

Q-1

In what way should a system for analysis of multivariate data, using principal component analysis (and possibly related methods in multivariate data analysis), utilize new methods of computer graphics and Virtual Reality (VR) to give the users optimal information viewing capabilities?

The system of this project uses animation, red-blue stereoscopy and general purpose computer graphics to provide a simple desktop VR system. The creation of a usable PCA tool with basic tool confirms that these features should be applied for a better quality of PCA. Especially the movements of data points in the scores plot are easier to understand with the use of animation. The system does not use immersive VR or special purpose hardware for the interaction (E.g. Data Glove or Head Mounted Displays). Adding such functionality to the system would result in a system with higher level of *natural interaction*, but would also demand more training of the users.

Q-2

What combinations of hardware interfaces will provide the best information viewing capabilities for multivariate data?

The system uses standard hardware interfaces, a computer with a CRT screen and keyboard and mouse as input devices. Polaroid glasses are used to view the red-blue stereoscopy of the system. For desktop virtual reality, this combination gives sufficient information viewing capabilities, at the easiest level of interaction and highest level of trainability.

Q-3

What are the qualitative changes of viewing the data, when using the added functionality of the new interfaces?

The movement of points in the scores plot is clearly visible when using animation. The dynamics of the changes are much easier to understand with the use of animation. The red-blue stereoscopy adds depth vision to the system, helping to understand spatial distances in the 3D case.

Q-4

What degree of knowledge of the information in the multivariate data should be expected, when using principal component analysis (PCA) with the new interfaces of the system to be developed?

The system presents a clear perspective on the multivariate data, providing analysis to acquire the information. However, a solid understanding of PCA is required.

Q-5

What degree of knowledge of the dynamics that occur in the multivariate data should be expected, when removing outliers of the score plot?

The system shows the removal of outliers in the score plot more visible than previous systems like Scicraft, using animation to show the movements of the data points. A higher degree of knowledge of the dynamics is acquired. The use of Procrustes analysis to filter out rotation has also been beneficial to understand the scaling of the data.

The final conclusion of the project and this report is that multivariate data analysis is easier to conduct, using the desktop VR methods animation, stereoscopy and spin plots. The additional methods do provide more insight and knowledge of the multivariate data. The main conclusions have been gathered from experiences in using the developed system for this project.

6.2 Further work

The system is still a prototype, and its source code is available on the CD-ROM with this report. The graphical quality of the system is sufficient. Using higher screen resolution and better rendering quality should only be applied if the frame rate can be kept at an interactive frame rate above 12 frames per second, which have proven sufficient for this system. The animations should be run above this frame rate to provide continuous and not sequential view, plus avoid flickering. The interaction quality of the system is sufficient for a desktop VR system. However, using special purpose hardware like Data Gloves can provide a more natural interface of the system. The system must be fully redesigned to be controllable with Data Gloves. A better alternative is to allow the user use one Data Glove for selection and commands (e.g. *View Data* or *Animate Removal* or picking data points), while the other hand controls the mouse and gives keyboard inputs. This combination of Data Glove and mouse or keyboard will not be beneficial for all systems, and could result in developing a feature which the user will not like, because it is cumbersome. The main reasons for not developing Data Glove support in the system are the technical challenges of acquiring drivers and programming interfaces in Python to the available Data Gloves from 5DT[7], and the sufficient and user friendly and quick interaction this desktop VR system provides. If however data glove functionality is required, the *Cgkit* can be applied in the system.[8] The data quality of the system is sufficient. However, the support for only one data format results in a system that cannot handle many data files. The Scicraft application is available in open source and provides several file readers written in the Python programming language. A standard data format like XML is a good candidate for future supported files in the system.

[7] 5DT website – http://www.5dt.com

[8] Cgkit, the Python Computer Graphics Kit – http://cgkit.sourceforge.net

A Appendix A - Mathematical methods in PCA

This appendix will introduce mathematical methods used in PCA.

A.1 Eigenvectors and eigenvalues

Eigenvalues are a mathematical method used to find the eigenvectors and eigenvalues. The mathematical expression for eigenvalues and eigenvectors is stated in (A.1) [PENN98].

$$\lambda x = Ax \tag{A.1}$$

The expression (A.1) contains two unknowns, λ and x, which are the eigenvalues and eigenvectors respectively. To find the non-trivial solutions (the zero values of the two unknowns), the matrix must be solved as a singular matrix. Singular matrices are not invertible and have determinants equal to zero [WEISS95]. The expression (A.1) is reorganized into expression (A.2):

$$Ax - \lambda x = 0 \tag{A.2}$$

The fact that singular matrices with determinant zero are not invertible is stated in Cramer's Rule [PENN95]:

$$A^{-1} = \frac{adj A}{|A|} \tag{A.3}$$

The inverse matrix is undefined if the determinant |A| of a general matrix A is zero as the divisor of (A.3). To find the non-trivial solutions, expression (A.2) must be set to zero, and its determinant calculated (the singular matrix will be used). The adj stands for the adjoint matrix of A in (A.3).

$$|A - \lambda I| = 0 \tag{A.4}$$

The expression (A.4) is also known as the *characteristic equation*. I stand for the NxN identity matrix.

An example will now be given to demonstrate calculation of eigenvalues and eigenvectors, which is also known as eigenanalysis. Consider the following matrix A:

$$A = \begin{bmatrix} 5 & 7 \\ -2 & -4 \end{bmatrix}$$

This matrix has the following characteristic equation (A.4):

$$\begin{vmatrix} 5-\lambda & 7 \\ -2 & -4-\lambda \end{vmatrix}$$

This equals zero, and solving the determinant equation gives a second degree equation that can be solved manually. For larger matrices, numerical routines in e.g. Numerical Python can be used to find the eigenvalues. Calculations give the two eigenvalues:

$$\lambda_1 = -2\,,\ \lambda_2 = 3.$$

The eigenvalues can be inserted into (A.2), and the eigenvectors can be found. Calculations give the two eigenvectors:

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} 7 \\ -2 \end{bmatrix}$$

The first and second eigenvector in this example is not normalized, which is the method of dividing vectors with its Euclidean length. Most calculation packages, Numerical Python included, will normalize the eigenvectors.

The next sub chapter will explain covariance matrices.

A.2 Covariance matrices

Covariance matrices are used to statistically describe the amount of which dimensional variables (usually the columns) increase or decrease together. This amount is always considered between two dimensional variables. The mathematical expressions of covariance for a matrix X are listed in (A.5) [JOHN02] and (A.6) [RICH86].

$$\Sigma = E((X-\mu)(X-\mu)^T) \qquad \text{(A.5)}$$

$$\Sigma = \frac{1}{K-1}\Sigma_{j=1}^{K}(x_j-\mu)(x_j-\mu)^T \qquad \text{(A.6)}$$

(A.5) and (A.6) are equal. In (A.6), the expectancy operator E is expanded to its *unbiased average.* The main diagonal of covariance matrices **Σ** contains the variance for the matrix X. An example will now be given to demonstrate calculation of covariance matrices. Consider the matrix **X** [RICH86]:

$$\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 4 & 1 \\ 5 & 2 \\ 4 & 4 \\ 2 & 4 \end{bmatrix}$$

The geometric representation of X is six points in the 2D plane. The mean matrix of **X** must be calculated. These averages of each column is calculated, and then subtracted from all rows in the corresponding columns. This centres the matrix **X** to the origin. The matrix **Xc** is the mean centred matrix of X where μ is the mean vector:

$$Xc = (X - \mu) = \begin{bmatrix} -2.00 & -0.33 \\ -1.00 & -1.33 \\ 1.00 & -1.33 \\ 2.00 & -0.33 \\ 1.00 & 1.67 \\ -1.00 & 1.67 \end{bmatrix}$$

Expression (A.5) or (A.6) is then used to calculate the covariance matrix of **Xc**. Because the mean vector is already subtracted, and the matrix **X** is mean centred, the mean vector now is moved to the origin and equal to zero. This simplifies (A.5) and (A.6) by removing the need to subtract the mean vector μ. The covariance matrix of X is:

$$cov(X) = \Sigma = \begin{bmatrix} 2.40 & 0 \\ 0 & 1.87 \end{bmatrix}$$

The variance is as stated above along the main diagonal. In this example, the off-diagonal elements of the matrix equals zero. This means that there is no joint decrease or increase among the columns or variables (in this case, the x- and y- variables of the six points in the 2D plane). This means that there is no covariance for the points **X** and there is no special covariance between the first and second dimensional variable (x and y).

The *correlation matrix* is often used together with the covariance matrix. The expression for the correlation matrix is [RICH86]:

$$\sigma_{ij} = \frac{v_{ij}}{(v_{ii}v_{jj})^{1/2}} \qquad \text{(A.7)}$$

The v_{ij} elements belong to the covariance matrix. The correlation matrix is a scaled covariance matrix, expressing the covariance in percentage. The main diagonal of the correlation matrix will only contain value 1. The correlation matrix of for **X** equals the identity matrix of dimension 2x2. In general, matrices can contain non-zero or zero covariance and correlation. In Figure 2, the matrix **X** was shown with its principal axes. These were calculated in Matlab by mean centring the matrix **X** and calculating the covariance matrix. The eigenvalues and eigenvectors of the covariance matrix were found. The principal axes equal the two eigenvectors from this calculation.

SVD is a faster and more applicable method than performing eigenanalysis on the covariance matrix, especially since SVD accepts non-square matrices. This method will be explained in the next sub chapter.

A.3 Singular Value Decomposition

SVD is a decomposition or factorization of a general matrix A. It is stated as follows [STRANG99]:

$$A = U\Sigma V^T \qquad \text{(A.8)}$$

The matrix A is written in (A.8) as a product of the orthogonal matrices U and V^T and the diagonal matrix Σ. The main diagonal of Σ contains the singular values, which are the square roots of the eigenvalues. This means that Σ can be used to find the eigenvalues and generate the scree plot. The orthogonal matrices are found first using the following identities (uses the fact that orthogonal matrices that are multiplied with their transpose matrices give identity matrix) [STRANG99]:

$$AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T \qquad \text{(A.9)}$$

$$A^T A = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T\Sigma V^T \qquad \text{(A.10)}$$

Once the orthogonal matrices U and V^T, are found using (A.9) and (A.10), the diagonal matrix Σ can be found using (A.8).

An example will now be given to demonstrate calculation of SVD. Consider the following matrix **A**:

$$A = \begin{bmatrix} 4 & -3 \\ -3 & 4 \end{bmatrix}$$

The matrix **A** is symmetric, which means that the matrix products AA^T and A^TA are equal:

$$AA^T = A^T A = \begin{bmatrix} 25 & -24 \\ -24 & 25 \end{bmatrix}$$

Performing an eigenanalysis on this matrix will produce the eigenvalues $\lambda_1 = 49$, $\lambda_2 = 1$. The singular values are the square root of the eigenvalues, and are the elements of the main diagonal of the diagonal matrix Σ. This gives the following matrix:

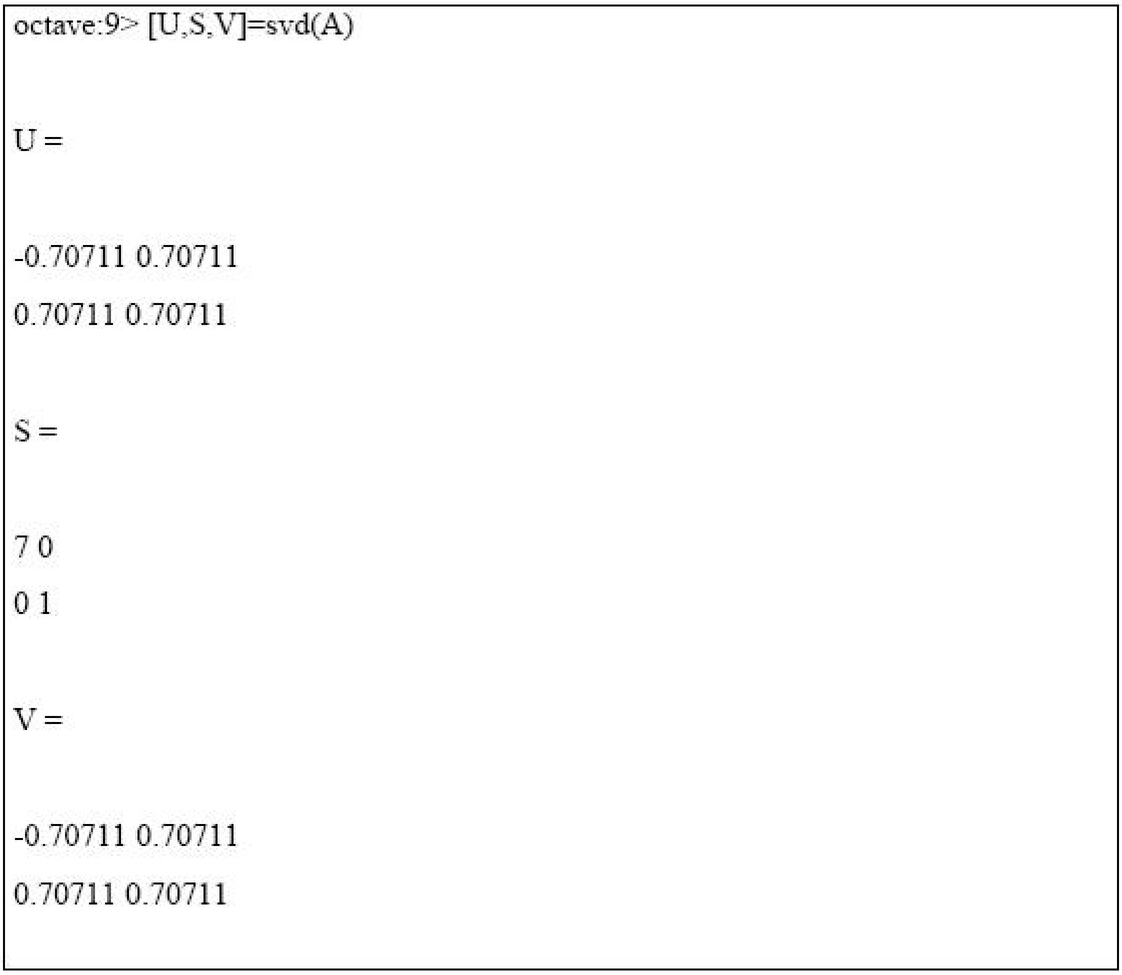
$$\Sigma = \begin{bmatrix} 7 & 0 \\ 0 & 1 \end{bmatrix}$$

The connection between the diagonal matrix Σ and calculating the eigenanalysis of the matrix products AA^T and A^TA gives Σ (a result from SVD theory not discussed in this report). Once the diagonal matrix Σ is found, U and V^T are calculated using (A.9) and (A.10).

SVD is requires more steps of calculations, but there are fast numerical implementations in e.g. Numerical Python. An example of running SVD on the matrix A in *Octave*[9] is shown in Figure 10:

The next sub chapter will discuss Procrustes analysis.

[9] Octave website – http://www.octave.org



```
octave:9> [U,S,V]=svd(A)

U =

-0.70711 0.70711
0.70711 0.70711

S =

7 0
0 1

V =

-0.70711 0.70711
0.70711 0.70711
```

Figure 10 - Calculating SVD on matrix A using Octave.

A.4 Procrustes analysis

The last mathematical method of this appendix is Procrustes analysis. This is a composite transformation, using translation, rotation or reflection and with generalized Procrustes analysis (GPA) scaling. The aim of Procrustes analysis is to fit two geometric configurations, to provide maximum overlap. The Procrustes name is derived from a character in Greek Mythology. The background story can be read in [JOHN02].

Procrustes analysis fits a matrix X towards a matrix Y. An alternative is to fit Y towards X, which is in the opposite direction. A third alternative is to try fitting both matrices X and Y towards an average. An example of how to use Procrustes analysis with animations is setting the initial geometric configuration (the *start frame*) as Y and final geometric configuration (the *end frame*) as X, and then fitting X towards Y. An algorithm for calculating Procrustes analysis for matrix X and Y is shown next. This algorithm is standard Procrustes analysis, not allowing scaling. The algorithm will remove rotation from an animation (such that an observer only is shown the scaling) [JOHN02].

ALGORITHM A-1: PROCRUSTES ANALYSIS

1. Centre X and Y to the origin by subtracting the mean. Replace X and Y.
$X = Xm = X - MEAN(X)$. $Y = Ym = Y - MEAN(Y)$.
2. Calculate the SVD of X^TY. Set $SVD\ (X^TY) = U\Sigma V^T$.
3. Rotate X by UV^T. $X' = XUV^T$.
4. Calculate the average $Z = (X'+Y)/2$.
5. Calculate the *PCA* of Z, using eigenanalysis on Z^TZ (covariance matrix).
6. Use the eigenvectors from the PCA of Z to form the matrix V.
Rotate all matrices Z, X' and Y with this rotational matrix V. $Z = ZV$, $X' = X'V$, $Y = YV$.
End of Algorithm.

It is possible to skip executing step 4-6 in Algorithm A-1, if there is no need to calculate the average geometric configuration Z. Algorithm A-1 lacks the additional step of scaling the two geometric configurations X and Y as in GPA.

Procrustes analysis can also be used for pattern recognition between a template object and search objects. [JOHN02]

B Appendix B - Technical terms and abbreviations

In this appendix technical terms and abbreviations in the report will be listed, with a short description. Refer to the *index* of this report for further reading of the terms in the report, or see the *references* for additional reading in resources.

Chemometry	Multivariate data analysis. A field of chemistry analysing multivariate data, using frequently statistical methods and computer science.
Covariance	Statistical concept. For a general matrix X, the covariance of X is stated as E $((X - \mu)(X - \mu)^T)$. Measures the degree dimension variables of a matrix are increasing or decreasing together (usually columns).
Eigenvalues	Mathematic concept. For a general matrix A, in the expression $\lambda X = A X$ λ is the eigenvalues. Eigenvalues are scalar values, either real or complex.
Eigenvectors	Mathematical concept. For a general matrix A, in the expression $\lambda X = A X$, X is the eigenvectors. Eigenvectors are vectors, either real or complex.
GPA	Generalized Procrustes Analysis. Refer to Procrustes, defined below in this table.
Linear Interpolation	Mathematical concept. Can be used for a multitude of field within computer science, for example animation.
Outliers	Outliers are points that are separated with a distance from the collection of remaining objects in plots. In PCA outliers are divided into moderate and strong outliers.
PCA	Principal component analysis. Reduction and rotational transform that orients the principal axes in the direction of highest variance, to describe the information of the plot (basically its variance) with the fewest possible principal axes.
Procrustes	Procrustes analysis. Composite transform for fitting to geometric configuration to be most similar, to create a best possible fit or overlap. Allowed transforms are translation, rotation or rotation, and for the Generalized Procrustes Analysis, scaling.
Stereoscopy	Different methods to provide stereovision or depth vision for 2D and 3D scenes on a 2D viewing canvas, such as a computer screen.
SVD	Singular Value Decomposition. Mathematical concept, factorization or decomposition of a general matrix. Generates a product of orthogonal, diagonal and orthogonal matrices (three factors). For a general matrix A the decomposition is $A = U \Sigma V^T$
VR	Virtual reality. Different software and hardware in computer science aimed at creating a virtual reality mirroring real or abstract scenes. Usually divided into desktop and immersive VR.

Table 4 - Technical terms and abbreviations.

C Appendix C - Additional resources

The additional resources for this project are collected on the CD-ROM that is available with the report.

C.1 Overview of the resources

The resources available on the CD-ROM are listed in Table 5:

Resources	File path on the CD-ROM
Documentation	
API-documentation	/Documentation/Apidoc
Installation manual	/Documentation/Manuals/
Report	/Report/
System manual	/Documentation/Manuals/
User manual	/Documentation/Manuals
Screen images	/Documentation/Screens
Videos	/Documentation/Videos
System	
Compiled byte code of modules	/System/
Source code (modules)	/System/
Supporting software	/System/Packages/

Table 5 - Resources on the CD-ROM.

C.2 System

The *System* folder on the CD-ROM contains the modules of the system. To run the system, the supporting software packages must be installed first. This is explained in the installation manual on the CD-ROM. The software packages is included on the CD-ROM in the subfolder *Packages* of the system. Windows and Linux Debian based software packages are included. The necessary Qt package is not freely available for Windows (but Linux), and must therefore be acquired from e.g. the Internet. The system is developed with Linux Debian operating system, and is most convenient to install with this operating system. The system itself must be run with the Python interpreter. Shell script files are included in the *System* folder to start the different applications of the system (Windows users must manually start the Python interpreter and open the runtime modules). The source code files are Python module files with the extension .py. The byte code compiled source code has extension .pyc or (optimized) .pyo.

C.3 Documentation

The *Documentation* folder on the CD-ROM contains documentation of the system. Also included are videos[10] and screens images[11] of the system running.

The API-documentation for the source modules is available in HTML format. This documentation is written for future developers of the system. The API-documentation has been generated with Epydoc[12]. The installation manual, user manual and system manual is also available in the HTML format. The installation manual is aimed for a system installation on a Linux Debian system. The user manual explains how to use the system. The system manual introduces the system features for new users. The report is available as a PDF file in the *Report* folder.

C.4 Resources available on the Internet for the project

It is available to acquire software packages for the system by use of the Internet.

It is also suggested to use the package tool *Apt* or *Synaptic* available with Linux Debian. These package tools quickly install the necessary software to copy the system to the target or

It is suggested to install Scicraft, since this installation will also install the required software packages to the system. Use the internet to visit websites that contains the software packages required by the system, listed in Table 6.

[10] The videos have been created with Xvidcap and are MPEG-1 formatted.
Xvidcap website – http://xvidcap.sourceforge.net
VLC - http://videolan.org or acquire other MPEG video viewers.

[11] The screen images are JPEG formatted.

[12] Epydoc website – http://epydoc.sourceforge.net

Software Package	**Website**
Linux Debian	http://www.debian.org
Numerical Python	http://sourceforge.net/projects/numpy
Python	http://www.python.org
PyQt	http://www.riverbankcomputing.co.uk/pyqt/
PyVTK	Install VTK.
Qt	http://www.trolltech.com
Scicraft	http://www.scicraft.org Scicraft is not required by the system, but will install the required software packages for the system.
VTK	http://www.kitware.com

Table 6 - Internet resources for packages to the system.

Start the browsing of the CD-ROM by opening readme.txt in the root folder.

It is possible to run the system from the CD-ROM, but it is suggested to copy the *System* folder to an available location of the hard drive to the target computer.

D Appendix D – Software Requirements Specification (SRS)

The SRS for the system to be developed in this project will be explained in this appendix.

D.1 Introduction

This SRS is modelled after the IEEE Standard 830-1998 [IEEE98]. Some sections of IEEE 830-1998 are not applicable to the system, and will not be included.

D.1.1 Purpose

The purpose of this SRS is to specify the properties and requirements of the system, and to document the common agreement between the developer and the *customer*, in this case the customer is NTNU (independent research project). The SRS will also be referred to in later stages of development (design and implementation), but not discussed in these chapters of the report for brevity.

D.1.2 Scope

This SRS will only describe the requirements and properties of the system to be developed in this project.

D.1.3 System description

The system is a client program to analyze multivariate data with principal component analysis in 2D and 3D. The data will be visualized in spin plots. The system will use virtual reality methods (desktop VR). Possible desktop VR-methods to apply to the system are stereoscopy, Data Gloves and animation.

D.1.4 System applicability

The system will only be used for multivariate data analysis (principal component analysis). The system can also be used for demonstrations that display how to use desktop VR with multivariate data analysis.

D.2 System overview

The system overview will be presented in this part of the SRS.

D.2.1 Main components

The system will be based upon a *parser, mathematical handler* and *visualization handler*. The system can use existing code from Scicraft, and use Octave for mathematical calculations (suggestive). The system will be written in Python. Numerical Python can also be used to implement the mathematical handler. The parser can use readers from Scicraft or develop own readers. The visualization handler should either use Scicraft modules as a code base, or develop an entirely new solution. Operative system interoperability can be implemented with the *OS module* for Python. The visualization handler should use VTK and Qt. Stereoscopy and use of VR must also interact with VTK. VR operability for Data Gloves should use the VR libraries *Cgkit*, VRJuggler or *Glisa.*

D.2.2 System interfaces

The system will cooperate with the operative system using the OS module as mentioned. This module should retrieve and store files from the file system of the computer the system is installed on. VTK and Qt should provide the graphical user interface, visualization and user interaction. Octave or Numerical Python should provide calculations. Python will be the programming interface of the system. Glisa, Cgkit or VRJuggler should be used to program the Data Gloves.

D.2.3 User interfaces

The system should visualize the loadings-, scores- and scree-plot when calculating PCA on multivariate data sets. The user should be able to define PCA axes and the active array of a data set. The core functionality should be easily accessible and trainable.

D.2.4 Program interfaces

The system will be programmed with Python and will be module based. Specific modules from Scicraft can be imported and modified to provide Octave file readers or other relevant functions. The program bindings PyQt and PyVTK will be used to program Qt and VTK based programs in Python. For the Data Glove, Cgkit, Glisa or VRJuggler should be applied.

D.2.5 Hardware interfaces

The system should use 5DT Data Glove 5 (one or two gloves) from Fifth Dimension Technologies and the Flock of Birds positional devices from Ascension Technology Corporation. The Flock of Birds is magnetic based positional tracking devices to be used with the Data Gloves to provide positional information of the Data Glove. Polaroid glasses should be used for the red-blue stereoscopy of the system. The system will either use computer monitor device (CRT or LCD), or a projector with canvas (stereo projectors are available).

D.2.6 User operations

The central user operations available in the system are the following:

1. The user can open a data file written in the reduced Octave format of Scicraft and load its variables in the system. PCA should be calculated automatically.
2. The user can pick or mark several objects to indicate outliers in the scores plot.
3. The user can set up the principal axes for the visualization plots.
4. The user can turn on or off stereoscopy in the system (red-blue).
5. The user can turn on or off VR-navigation in the system.
6. The user can control (initiate) animation in the system with the user interface.

D.2.7 System functionality

The central system functionalities available in the system are the following:

1. The system can parse Octave-formatted data files and create a list of the containing variables of the file.
2. The system can calculate PCA on the variables of the data (arrays).
3. The system can visualize in two or three dimensions.
4. The system can navigate the dataset using input device
(Data Glove or mouse and keyboard can be used).
5. The system can modify the data sets by removing outliers in the scores plot and recalculate PCA.

6. The system should animate the changes in the plots when outliers are removed and the user want to animate the changes. Rotation should be possible to turn on or off using Procrustes filtering.
7. The system should control the animation setup.
8. The system should acquire data and events from the system input devices, i.e. mouse, keyboard, Data Glove or Flock of Birds.

The system should also include the normal user operations, like starting and stopping the system.

D.2.8 User characteristics

The users of the system will mainly be employees (scientists) at the Institute of Chemistry, Group of Physical Chemistry, NTNU. The common properties of these users are extensive education in multivariate data analysis, and variable level of knowledge of using computers. The system should be possible to use for all the employees, it must therefore be a user friendly system.

D.2.9 Limitations

The system will be developed with approximately the same development tools as Scicraft. The user interface should also not deviate too much from the Scicraft design. An advantage with developing the system with the same tools as Scicraft is the fact that most future users already have training in using Scicraft, and the system will be easier to understand with a familiar design.

D.2.10 Future changes

The system will be not use Data Gloves if the libraries for Data Gloves (Cgkit, VRJuggler or Glisa) are erroneous or not implementable with Python. Initial tests show that Cgkit works best with the 5DT and Flock of Birds hardware, but Glisa and VR Juggler is either not working or is not providing enough functionality. Another possibility is the design choice of not using Data Gloves for a simpler implementation and interaction style.

D.3 Specific requirements

The specific requirements will be defined and then listed in tables. The functional requirements are listed first, then the non-functional requirements.

D.3.1 External interfaces

The system will operative with the OS or Octave externally. The system will use the OS module of Python to retrieve the data files stored on the file system. The following requirements are established:

EX-1	The system will use the OS module of Python for OS operability.
EX-2	The system will interpret the variables in the Octave files with a parser module (Scicraft or new parser module).
EX-3	The system should be able to use VR-based devices like 5DT Data Gloves and Flock of Birds.
EX-4	The system should be able to use Octave or Numerical Python to calculate PCA (and Procrustes Analysis).

Table 7 - External interfaces of the system.

D.3.2 Functions

The specific requirements for the system will be discussed in this part. In most SRS, this description is elaborate, but will be shorter in this report for brevity. The following requirements are established:

FC-1	Descriptive summary of data files should be available.
FC-2	Lists of variables from data files should be available.
FC-3	Principal axes should be possible to set.
FC-4	PCA should be possible to calculate.
FC-5	Outliers should be possible to mark in the scores plot.
FC-6	Active variable in the data sets should be possible to set.
FC-7	PCA should be recalculated when outliers are removed.
FC-8	Stereoscopy should be possible to active or deactivate.
FC-9	VR-based devices should be possible to active or deactivate in the system.
FC-10	Navigation of the plots should be able to perform.
FC-11	Procrustes analysis should be calculated when filtering rotation from the animations.

Table 8 - Functional requirements of the system.

There will be several other helping functions in the system and additional code. See the API documentation on the CD-ROM for the system for more elaborate explanation of each function in the modules.

D.3.3 Performance

The system should handle large multivariate data sets, up to arrays with 70 rows and 3000 columns. These matrices should be calculated with PCA and visualized in real-time. To achieve this, the minimum frame rate of 12 FPS is set as a performance marker for this system to support continuous, not sequential visualization. It is also important that the VR-equipment should handle user interaction precise (when calibrated) and the mouse interaction should record pick operations of objects in the plots. The following requirements are established:

PF-1	The system should handle matrices with maximum size of 70 rows and 3000 columns.
PF-2	The system should animate at a refresh rate of 12 FPS minimum.
PF-3	The system should recognized calibrated input device interaction at least 70 percent of the time.

Table 9 - Performance requirements for the system.

D.3.4 Requirements of data

The files for this system should be in the reduced Octave format (.oct). There are additional formats that would be desired. Program files in the Matlab format (.m) and R-files. Scicraft contain reader modules for this functionality. The following requirements are established:

I-1	The system should present the contents of Octave files.
I-2	The system should return the variables in Octave files.
I-3	The system should calculate PCA of the variables returned.
I-4	The system should be able to run Matlab format program files or R files.

Table 10 – Specific requirements for the data i the system.

Note that I-4 is suggestive. This demand is a natural extension of the system.

D.3.5 Design limitations

The system should be developed with the same developing tools as Scicraft utilizes. The user interface should have the same design. The following design limitations are established:

DS-1	The user interface should have the same design as Scicraft (approximate).
DS-2	The system should use Qt and PyQt to provide the graphical user interface.
DS-3	The system should use PyVTK and VTK to visualize the data.
DS-4	The system should use Numerical Python for calculations.
DS-5	The system should be programmed as modules in Python.
DS-6	The system should use Glisa, VRJuggler or Cgkit as VR libraries.
DS-7	The system should parse and read files in the reduced Octave format of Scicraft.

Table 11 - Design limitations of the system.

D.3.6 Use of standards

The system will use the Python Enhancement Proposals (PEPs) for the programming.

The documentation will be written using the document generation tools PyDoc or EpyDoc.

The user interface will follow Scicraft standard. The data files parsed by the system will be the reduced Octave format from Scicraft. The following requirements are established:

SD-1	The documentation of the source code (API docs) will be generated with EpyDoc or PyDoc.
SD-2	The development tools should be the same as in Scicraft.
SD-3	The system should apply the Python PEPs.
SD-4	The SRS should use IEEE830-1998 std.

Table 12 - Standards in the system.

D.3.7 Properties of the system

This part will discuss the properties of the system.

Reliability

This project has too little time span to exclude all errors by use of the system. But the system should not create critical errors or terminate. These demands should be met by using the safe-keeping in the Python interpreter environment. The following requirements are established:

RE-1	The system should not abort or terminate without the users intention.
RE-2	The system should accept user errors.
RE-3	The system should provide error messages.
RE-4	The system should correct wrong user input.

Table 13 - Specific requirements for the reliability of the system.

Availability

The user will decide when to start and end the system. The system is a client system for one user. There are no specific demands of the availability of the system.

Safety and integrity

The system should be safe to use and the original contents of the data should not be altered. The following requirements are established:

SI-1	The system should not perform harmful operations.
SI-2	The data in the files should not permanently be overwritten.

Table 14 - Safety requirements for the system.

Maintainability

The system should be accessible as source code and byte compiled code. It should not be necessary to recompile and the modules should be possible to use in other applications. This will be covered by basic Python functionality. The following requirements are established:

MB-1	The system should be available as both source code and byte compiled code.
MB-2	The system source code should be commented.
MB-3	The source code should use Python PEPs.
MB-4	The source code should be documented with API-docs.
MB-5	The system should contain a user manual.
MB-6	The system should contain a system manual.
MB-7	The system should contain an installation manual.

Table 15 - Maintainability requirements for the system.

Portability

The system will be developed with Linux Debian and use Linux as the OS platform.

The following requirements are established for OS portability:

PO-1	The system should be applicable in Linux in general.
PO-2	The system should be applicable in Linux Debian in special.
PO-3	The system should be applicable in Windows or Mac OS X.
PO-4	The system should be applicable in other OS from the above.

The main OS for this system is Linux Debian. Running the system on Windows or Mac OS X requires a license of Qt, which the GUI of the system is built on
(Qt is free for Linux platforms).

System modes

The system has four special modes, besides its general modes. The first mode is when analyzing in two dimensions. The second mode is when analyzing in three dimensions. The third mode is when running VR to navigate the data sets in three dimensions. The fourth mode is when selecting one or multiple data points in the scores plot and then animating. In addition, the system has the usual system modes, launch, data reading, and data calculation, variable storing and exiting the system. The following requirements are established:

SM-1	The system should have a system mode for 2D PCA analysis.
SM-2	The system should have a system mode for 3D PCA analysis.
SM-3	The system should have a VR mode.
SM-4	The system should have an animation mode when removing outliers in the data.

Table 16 - System modes.

User classes

There is only one common user class for this system. It is the following:

BR-1	The system should have a common user class called *chemometricians*.

Table 17 - User classes of the system.

Physical objects
The physical objects integrating with the system is the VR-devices, Polaroid glasses and the assorted computer equipment, which is the computer, key board, mouse and monitor device or stereo projectors. The physical objects are as follows:

PO-1	The system should use 5DT Data Glove 5 VR interaction devices (1-2).
PO-2	The system should use Ascension Flock of Birds positional tracking devices (1-2).
PO-3	The system should provide red-blue stereoscopy, for use with Polaroid glasses.
PO-4	The system can use stereo projectors.
PO-5	The system must use additional computer equipment of the *customer*'s visualization lab(standard workstation equipment)

Table 18 - Physical objects in the system.

Stimulus-response
When picking an object in the plots, it should change colour. When picking, the coordinates of the object (world-coordinates) should also be provided. The additional stimulus-response actions should follow from the user interactions. Especially the animation is an important S-R feature of the system.

External stimuli
There are no other external stimuli to the system, besides the input devices.

Responses
The responses in the system are suggestive. When moving the cursor with the VR-devices, a cone object should be visualized to represent the 3D cursor. The following requirements are established for the responses in the system:

RP-1	The system should display a cone as a 3D cursor when using VR devices.
RP-2	The system should use Qt and PyQt event model for the GUI.
RP-3	The system should show an animation of the removal of outliers.
RP-4	The system must show its axes in the plots.
RP-5	The system position should be possible to read from the axes or objects (pick operation).

Table 19 - Responses in the system.

Functional hierarchy

The system will be implemented as modules and there will be functions using other modules or execute as internal functions, using one module only. The system will use the Python import mechanisms for including code from other modules. Help variables will also be used, and the GUI-based applications will contain large GUI application classes.

Comments

This SRS has been developed after discussion with Professor Alsberg. The requirements follow the specifications given in these discussions. Some detailed requirements have been deduced by the author, but only if they did not disagree with the overall requirements given by Professor Alsberg. The system will be implemented according to this SRS, but technical challenges and difficulties may reduce the number of requirements covered by the implemented system.

E Appendix E – Code listing

This appendix will show code listing of the modules of the system. Summary information of the functions belonging to the modules is also included. The modules of the system are listed in Table 20. The modules are located in application subdirectories of the *System* folder.

Module	**Functionality**
basicpca2d.py	Application which handles basic PCA in 2D. Visualization handler of the system.
basicpa2d_anim.py	Application which handles basic PCA in 2D, with animation.
basicpca2d_VRanim.py	Application which handles basic PCA in 2D, with animation, VR (red-blue stereoscopy).
basicpca3d.py	Application which handles basic PCA in 3D. Visualization handler of the system.
basicpca3d_anim.py	Application which handles basic PCA in 3D, with animation.
basicpca3d_VRanim.py	Application which handles basic PCA in 3D, with animation, VR (red-blue stereoscopy).
myinspect.py	Application which provides parsing of .oct files.
myparser.py	Parser for .oct files. Retrieves the variables.
mypca.py	Calculates PCA using SVD. Mathematical handler of the system.

Table 20 - System modules.

The graphical applications are coded in a common GUI class, and use the two modules *myparser.py* and *mypca.py* as their parser module and mathematical handler module. They also perform mathematical calculations internally. All calculations are performed using Numerical Python and standard Python. The summary information of the two modules *myparser.py* and mypca.py is listed below.

Module mypca.py

This module performs SVD on an input array and includes helper functions.

Function name:	**mysvd**
Input arguments:	inputarray (object)
Output arguments:	U, S, V (objects)
Intended objects:	array
Call example:	mysvd(an_array)

The function *mysvd* calculates the SVD of an input array. The function returns the SVD decomposition (three arrays). The data type array is part of the Numerical Python library specification.

Function name:	**myrowcount**
Input arguments:	inputarray (object)
Output arguments:	rowcount
Intended objects	Input: array. Output: integer.
Call example:	myrowcount(an_array)

The function *myrowcount* retrieves the number of rows of an input array.

Function name:	**mycolumncount**
Input arguments:	inputarray (object)
Output arguments:	columncount
Intended objects	Input: array. Output: integer.
Call example:	mycolumncount(an_array)

The function *mycolumncount* retrieves the number of columns of an input array.

Module myparser.py

This module parses files in the reduced Octave format and returns summary and content information.

Function name:	**inspect**
Input arguments:	filename (object)
Output arguments:	I/O printing out stream
Intended objects:	Input: String. Output: print stream.
Call example:	inspect('myfile.oct')

The function inspect a file (specified in the input argument), and prints the contents out to the standard I/O printing out stream.

Function name:	**summary**
Input arguments:	filename (object)
Output arguments:	I/O printing out stream
Intended objects:	Input: string. Output: print stream.
Call example:	summary('myfile.oct')

The function *summary* prints the summary information of an Octave file, and prints the contents out to the standard I/O printing out stream.

Function name:	**summarylist**
Input arguments:	filename (object)
Output arguments:	summarylisting
Intended objects:	Input: string. Output: list.
Call example:	summarylist('myfile.oct')

The function *summarylist* opens an Octave file and puts the summary information into a list.

Function name:	**retrieve**
Input arguments:	filename (object)
Output arguments:	contentlist
Intended objects:	Input: string. Output: list.
Call example:	retrieve('myfile.oct')

The function retrieve opens an Octave file and puts the content information into a list. It is the central function of this module, which parses an input file (in the reduced Octave format previously mentioned) and assembles all the information in a list, which then is processed by the visualization handlers in the graphical applications for PCA.

References

[HART98] G. Hartvigsen, *Forskerhaandboken*, Norwegian Academic Press, 1998.

[HAAS98] H. Haase, M. Goebel, P. Astheimer, K. Karlsson, F. Schroeder, T. Fruuhauf, R. Ziegler. How scientific visualization can benefit from virtual environments. 7:15, 1998.

[IEEE98] IEEE. *IEEE 830-1998 - Recommended Practice for Software Requirements Specification*.

[INSE90] Alfred Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *IEEE Visualization*, pages 361–378, 1990.

[JOHN02] D.W. Wichern, R.A. Johnson. *Applied Multivariate Statistical Analysis*. Prentice Hall, 5th edition, 2002.

[PENN98] David E. Penney C.H.Edwards Jr. *Elementary Linear Algebra*. Prentice Hall, 1st edition, 1998.

[PYTH03] Learning Python, D. Ascher, M. Lutz, *Learning Python, 2nd edition,* O'Reilly, 2003.

[RICH86] J.A. Richards. *Remote Sensing Digital Image Analysis - An Introduction*. Springer-Verlag, 1st edition, 1986.

[ROST02a] Rostock University, Online lexicon – *Geoinformatik service* *http://www.geoinformatik.uni-rostock.de/einzel.asp?ID=620332223*

[ROST02b] Rostock University, Online lexicon – *Geoinformatik-Service* http://www.geoinformatik.uni-rostock.de/einzel.asp?ID=747778018

[SPENC01] R. Spence. *Information Visualization*. Addison-Wesley, 1st edition, 2001.

[STRANG99] Massachusetts Institute of Technology. *Professor Strang's Linear Algebra Class Lecture*. Open Courseware (Videos) - `http://web.mit.edu/18.06/www/Video/video-fall-99-new.html`.

[UIB05] University of Bergen. *Chemometrics in Bergen - Introduction*. `http://www.kj.uib.no/chemometrics/intro_chemometrics.htm`.

[UMET05] UMetrics software. *PCA*. http://www.umetrics.com/pdfs/books/MVDABook.pdf.

[VINC95] J. Vince. *Virtual Reality Systems.* Addison-Wesley, 5th edition, 1995.

[VTK04] Kitware, Inc., *The Visualization Toolkit User's Guide*, 2004.

[WATT99] A. Watt. *3D Computer Graphics*. Addison-Wesley, 3rd edition, 1999.

[WEISS95] Math World – A Wolfram Web Resource. *Eric W. Weisstein. Singular Matrix.* http://mathworld.wolfram.com/SingularMatrix.html.

Index