Visualization Tool for Tri-Space concept and SOM

by

Gwendolin Lehrer

2nd Bachelorthesis

Submitted in partial fulfillment of the requirement of the degree Bachelor of Science in Engineering

Carinthia University of Applied Science
School of Geoinformation

Supervisors

Internal Supervisor: Dr.-Ing. Karl-Heinrich Anders

School of Geoinformation, Carinthia University of Applied Sciences, Villach, Austria

External Supervisor: Dr. André Skupin

Department of Geography, San Diego State University,
San Diego, USA

San Diego, June 2013

Science Pledge

By my signature below, I certify that my thesis is entirely the result of my own work. I have cited all sources I have used in my theses and I have always indicated their origin.

Villach, 15.06.2013 

Name

Acknowledgments

First of all I want to thank my external supervisor Dr. André Skupin. He introduced me to a lot of interesting visualization techniques and kept my mind open for new ideas and methods. By discussing other topics with similar problem in his seminar I got a whole lot of new ideas to work with. He also gave me many interesting literature references to keep me awake at night. Another thank-you goes to Dr.-Ing. Karl-Heinrich Anders for the advices on this thesis. I also want to thank my colleagues Manuel Rainer and Michael Spöcklberger for the good ideas and code parts, as well as for many creative discussions about SOMs and good places to see in San Diego. Of course I also want to thank all my colleagues in the USA for making these three months so unbelievably great and for every time you helped me. Thank you for a delicious St. Patrick's Day and for finding restaurants to go to even if you are not 21 years old yet. And last but not least a special thank-you to my friends in Villach at CUAS, for three great years with a lot of work but a lot more fun. Hours in the cold measuring the university, days in the sun enjoying the Silbersee, nights of fun partying at Brauhof, I will never forget these three years and I hope to meet you all again.

Zusammenfassung

Diese Arbeit beschreibt die Entwicklung und Implementierung einer Applikation, die selbst organisierende Karten und das tri-space Konzept mit einer geografischen Karte und einem parallelen Koordinatengrafen verbindet. Der Grund das zu realisieren ist, dass selbst organisierende Karten sehr hilfreich beim Auffinden versteckter Muster in Daten sind. Das ist vor allem der Fall bei großen Datensätzen mit multidimensionalen Daten. Allerdings verwenden Laien selbst organisierende Karten kaum weil es viel Zeit und Arbeit braucht um sich gut einzuarbeiten und damit Muster, Ähnlichkeiten und Unterschiede aufzuspüren. Das Zeil ist es, ein Programm zu entwickeln, das den Benutzern helfen wird, von selbst organisierenden Karten zu profitieren. Zusätzlich soll durch die anderen Visualisierungsmethoden das Verständnis vereinfacht werden. Die weiteren Möglichkeiten der Darstellung wurden nach eingehendem Studieren der Fachliteratur zum Thema Visualisierung ausgewählt. Außerdem kann das Tool auch von Experten genützt werden, um Muster in den selbst organisierenden Karten leichter zu erklären und um weitere versteckte Informationen aufzudecken. Zusätzlich wurde das tri-space Konzept genutzt um mehr Informationen zu erhalten indem die Daten in die verschiedenen Perspektiven transformiert werden. Dafür wurde ein eigenes Programm entwickelt, welches aus allen CSV Dateien, die das richtige Format haben und die entsprechenden Daten enthalten, tri-space Dateien in allen Perspektiven erstellen kann und auch Dateien in dem Format, das von SOM_PAK benötigt wird. Um die Applikationen zu testen wurden Daten der Weltbank verwendet. Diese enthalten sozioökonomische Indikatoren der afrikanischen Länder. Zusätzlich wurde ein shapefile der afrikanischen Länder verwendet, welches vom Hoelzel Verlag zur Verfügung gestellt wurde und mit Hilfe von ArcGIS in das KML Format transformiert wurde. Die Applikation wurde mit der Hilfe von eclipse und Processing 2.0 entwickelt. Außerdem wurden einige Java-Libraries verwendet, zum Beispiel zum Auslesen der KML Datei und zum Darstellen des parallelen Koordinatengrafen.
Das Ergebnis ist ein Java-Applet, das auf einer beliebigen Website eingebunden werden kann.

Abstract

This paper describes the development and implementation of an application which combines self-organizing maps and tri-space with a geographic map and a parallel coordinate plot. The reason to do that is that self-organizing maps are good for finding underlying patterns in data; especially huge data sets which contain multidimensional data. Non-experts normally do not use them because it takes a lot of time and work to understand them in detail and use them to detect patterns, similarities and dissimilarities. This tool will help the users to benefit from self-organizing maps and simplify the comprehension with the help of the other visualization methods. The other possibilities to display the data were chosen after a detailed research in the visualization literature. The application can also be used by experts to simplify the explanation of patterns in the self-organizing map and to detect other underlying patterns in the multidimensional data. Additionally the tri-space concept was used to gain more information by converting between the different perspectives. To be able to do that a tri-space converter library was created, this library can also be used as stand alone program to convert CSV files into tri-space files in different perspectives as well as to convert them into the format used by SOM_PAK. To test the applications data from the world bank was used. These data contain socioeconomic indicators of African countries. Additionally a shapefile of the African countries was used; it was deployed by the Hoelzel publishing company and transformed to KML. The application was created using eclipse and Processing 2.0. There were also some java-libraries needed for example to read the KML file and to display the parallel coordinate plot.

The result is a Java-Applet which can be integrated in any website.

Table of Contents

List of Abbreviations

ANN	Artificial neural network
AT-L	Attribute-Time over Location perspective
COD	Output code book file format of SOM_PAK
CPU	Central Processing Unit
CSV	Comma / Character Separated Values
CUAS	Carinthia University of Applied Sciences
DAT	Input file format of SOM_PAK
GIS	Geoinformation System
GUI	Graphical User Interface
KML	Keyhole Markup Language
LA-T	Location-Attribute over Time perspective
LT-A	Location-Time over Attribute perspective
MDS	Multidimensional Scaling
RGB	Red Green Blue
SOM	Self-Organizing Map
SDMX	Statistical Data and Metadata eXchange
SWE	Snow-Water Equivalent
USA	United States of America

1. Introduction

In the introduction you will get a brief overview about the reasons, methodology and the planned results.

1.1 Motivation

As the population of Africa grew by about 792 million people since the 1950s it is very important to take a look at the other changes that result from that. In Africa the population spreads over the continent very irregular. On one hand there are cities like Cairo (ca. 15 million inhabitants) and Lagos (ca. 11 million inhabitants) and on the other hand there is the nearly uninhabited desert Sahara.

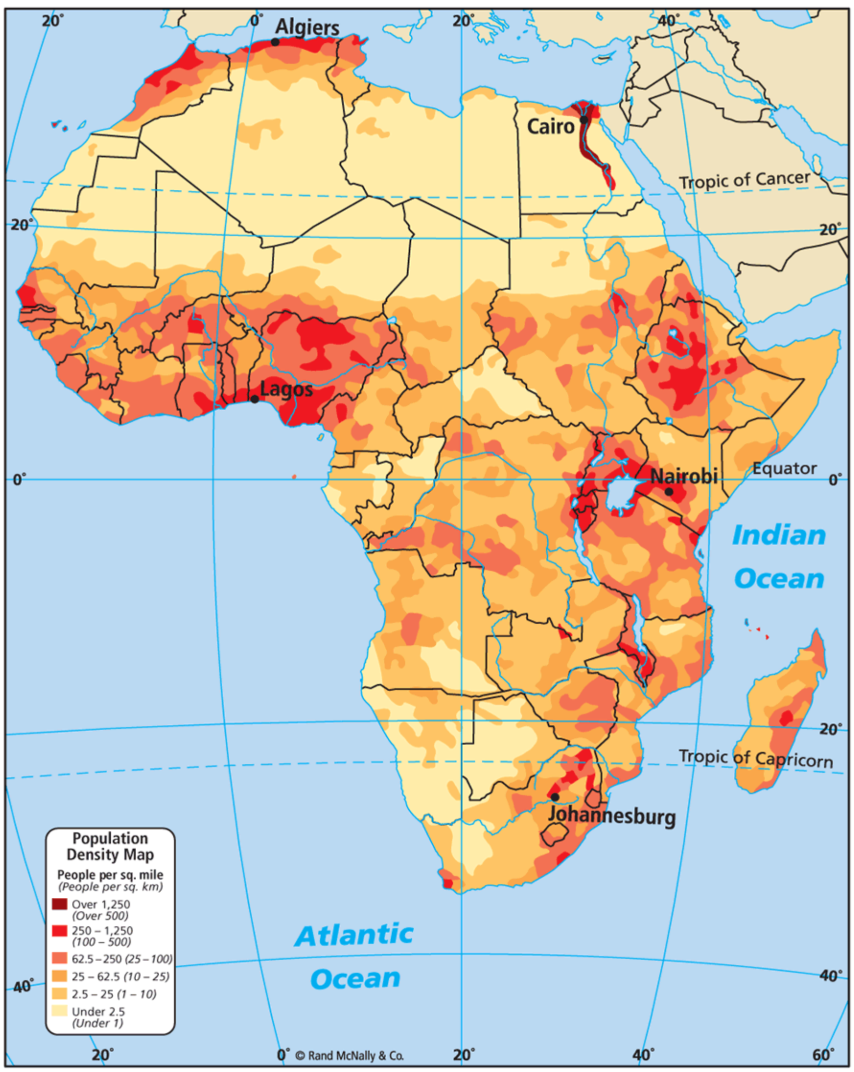


Figure 1: Population density in Africa in 2000

There are also many different peoples in Africa like the Arabs in the North, the Tuareg in the Sahara, the Boers in the South and the Maasai in the East, just to mention a few. Africa can not only be stratified on ethnic grounds but also on the basis of land use, such as by distinguishing among forest, plot areas and agricultural land. The data to be shown comes from the World Bank. The so called socio-economic indicators can be divided into the following categories:

- Agriculture & Rural Development
- Aid Effectiveness
- Climate Change
- Economic Policy & External Debt
- Education
- Energy & Mining
- Environment
- Financial Sector
- Gender
- Health
- Infrastructure
- Labor & Social Protection
- Public Sector
- Science & Technology
- Social Development
- Urban Development

These indicators are influenced by changes in law, economics, land use, politics, the physical environment and the development of new technologies. The data is multivariate; it consists of about 1250 attributes, about 50 countries and about 50 years. There are different methods to show multivariate data: For example self-organizing map (SOM) and multidimensional scaling (MDS) methods. The reason to use SOMs is that they provide a clearer view of the data in comparison to MDS which may be more truthful but harder to visualize, additionally the SOM method is scalable for much larger data sets than traditional MDS. Another reason is that MDS is based on dissimilarity matrices, which the given data doesn't include. (Skupin and Fabrikant, 2003) The more traditional ways, like for example parallel coordinate plots and geographic maps, of visualizing such data will not be replaced by using SOMs; they will be used to examine the results of the SOMs. Another method that will be used to make patterns more obvious to the users is the tri-space concept (Skupin, 2010). The goal is to provide the user with a solution the enables him or her to detect patterns, similarities and developments.

1.2 Problem Definition

Using SOMs to evaluate multivariate data is nearly impossible for non-expert users. Only presenting the data with the help of the traditional methods underlying structures and correlations may be not obvious enough. Transforming the tabular form of the data into different perspectives with the help of the tri-space is useless without a good presentation of the data afterward. Combining all these concepts in one user interface enlarges the chances of understanding the data and taking advantage of it.

1.3 Method of Solution

To give the user the chance to learn about correlations between the socio-economic indicators, the data has to be presented, using the principles of SOM and tri-space. This will enable the user to understand correlations between the three spaces of attribute, location, and time. This application will be developed in eclipse[1], with the help of the Processing[2] 2.0 library to simplify the graphical representation. To help as many people as possible with this development it will be provided as web applet. Additionally to the display of the SOM the data will be visible in a map window as well as in a parallel component plot to make it easier to understand for the non-expert user. All these parts of the application will be interactive; meaning that if the user chooses one input-vector it will be highlighted in all other views.

1.4 Expected results

- An application, that will enable users to develop new perspectives on the temporal and spatial patterns of the socio-economic development of African countries between 1960 and 2012
- A map view and an attribute view will be provided, to make the application more usable for non-experts.
- The application will give the users the ability to switch between tri-space perspectives.
- The switching between the perspectives makes it necessary to provide different controls, depending on the selected view. For example time-sliders and attribute lists.

1 http://www.eclipse.org/downloads/ (accessed: 12 June 2013)

2 https://www.processing.org/download/ (accessed: 12 June 2013)

1.5 Structure of the Thesis

At first the theoretical background of the work will be described like the concepts of SOMs and tri-space. Existing, similar projects will be mentioned and examined, including reflection on their pros and cons. Afterward the methods are discussed, the problem definition and the approach to solve it will be described in detail and the project area and the used data will be presented. The next chapter deals with the implementation in detail and then gives a summary. Afterward the solutions and their interpretation are presented and it is discussed if it was possible to implement the approach and if all goals were reached. In the end a conclusion will be drawn and the future work will be outlined.

2. Theoretical background

This chapter describes the used methods in detail, gives basic information about them and explains why to use them.

2.1 Self-Organizing Maps

The Self-Organizing Map (SOM), commonly also known as Kohonen network (Kohonen, 2001) is a computational method for the visualization and analysis of high-dimensional data. SOMs belong to the category of artificial neural networks (ANN), which consist of input and output elements, also called nodes. ANNs can fulfill many purposes but the main goal of a SOM is to cluster information. Clustering methods are normally used on geographical information and cluster close points. If clustering methods are used on attributes, they will calculate similarities instead of proximities. The SOM defines an ordered mapping: a kind of projection from a set of given data items, also known as vectors, onto a regular, usually two-dimensional grid. A model m_i is associated with each grid node. These models are computed by the SOM algorithm. A data item will be mapped into the node whose model is most similar to the data item, i.e., has the smallest distance and therefore the highest similarity to the data item using a suitable metric. (Kohonen and Honkela, 2007) That does not only affect the node which the data item is mapped on but also the surrounding nodes, because of that data vectors which are similar to each other will be placed close to each other on the SOM grid (Yan and Thill, 2008).

The input data are n-dimensional Euclidean vectors:

$$x(t)=[\xi_1(t),\xi_2(t),\ldots,\xi_n(t)].$$

In this case t is the index of the data item in its sequence. Let the ith model be

$$m_i(t)=[\mu_{i1}(t),\mu_{i2}(t),\ldots,\mu_{in}(t)].$$

Now t is another index in the sequence in which the models are generated. This sequence is defined as a smoothing-type process in which the new value $m_i(t+1)$ is computed iteratively from the old value $m_i(t)$ and the new data item $x(t)$ in this way:

$$m_i(t+1)=m_i(t)+a(t)h_{ci}(t)[x(t)-m_i(t)].$$

Here $a(t)$ is a scalar factor that defines the size of the correction; its value decreases with the step index t. The index i refers to the model under processing, and c is the index of the model that has the smallest distance from $x(t)$ in the Euclidean signal space. The factor $h_{ci}(t)$ is a kind of a smoothing kernel, also called the neighborhood function. It is equal to 1 when $i=c$ and its value decreases when the distance between the models m_i and m_c on the grid increases. Moreover, the spatial width of the kernel on the grid shall decrease with the step index t. These functions of the step index, which determine the convergence, must be chosen very delicately: cf. , e.g., (Kohonen 2001). Also the initialization of the $m_i(1)$ is a problem discussed in (Kohonen 2001). (Kohonen and Honkela, 2007)

A SOM can be split into component planes (Figure 2). Each of these represents one of the input vectors components. By looking at the different component planes, one can see similarities and differences between the spreading of various components.

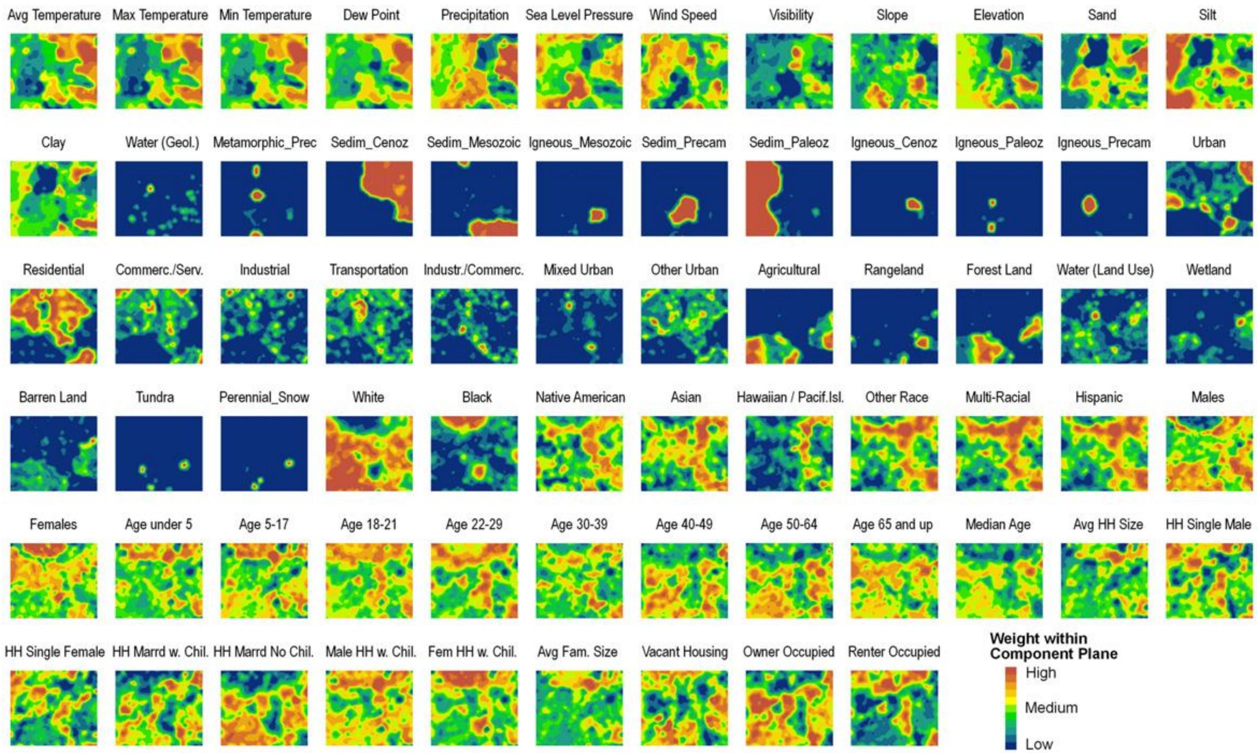


Figure 2: Component planes of a high-resolution SOM (500 * 500 neurons) trained with climate, topography, soil, geology, land use/cover, and population (Skupin and Esperbé 2011)

2.1.1 Examples using Self-Organizing Maps

In this chapter a few examples, showing what you can do with SOMs are presented and illustrated, as well as discussed if they are suitable methods for the application.

2.1.1.1 An alternative map of the United States

The alternative map of the United States based on an n-dimensional model of geographic space (Skupin and Esperbé 2011) is a holistic, high-resolution SOM of geographic features of the USA (United States of America). The focus was not only on finding clusters in the SOM but also showing the clusters in a regular map (Figure 3).

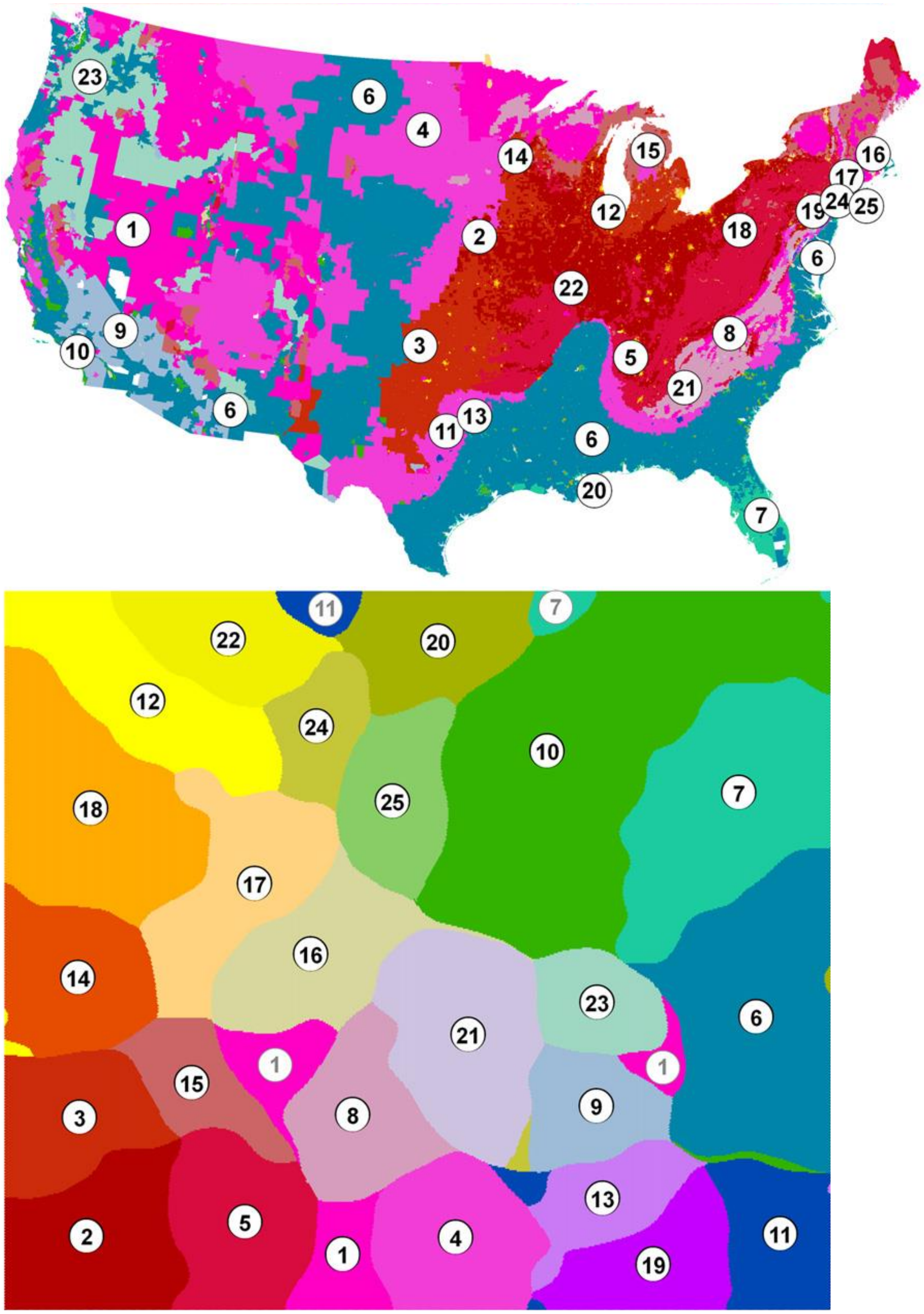


Figure 3: k-means clustering of n-dimensional neuron vectors projected into SOM space and geographic space. Large cluster enclaves indicated with gray labels. (Skupin and Esperbé 2011)

This shows how the SOMatic application will be constructed. When a user examines a SOM, he or she has the possibility to open the according map window to get a geographic overview.

2.1.1.2 SeeSOMe

SeeSOMe[3] is a basic visualization of a SOM as Java Applet (Figure 4). It was developed in Processing and shows demographic data about the US-American population. The user has the possibility to switch between component planes and vary the symbol size.

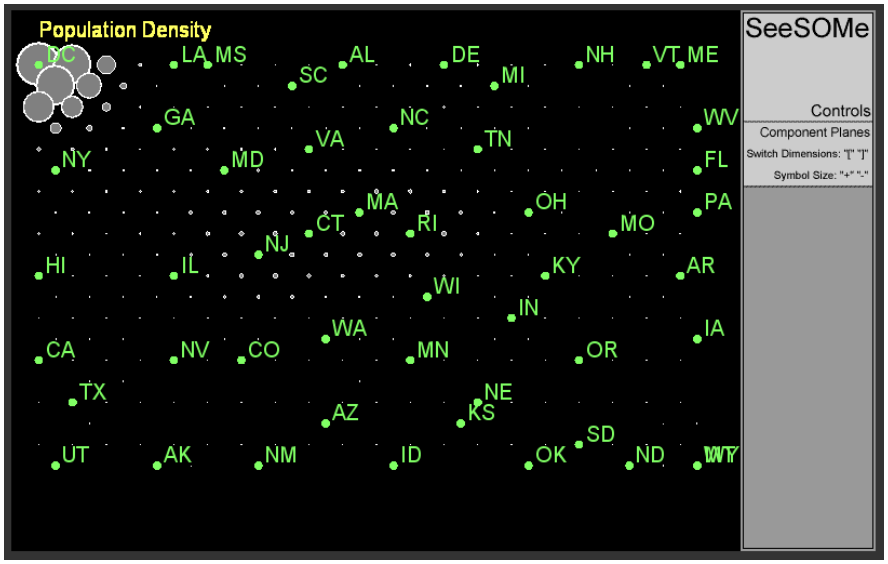


Figure 4: SeeSOMe showing population density component of a SOM trained with U.S. demographic data

In comparison to the alternative map of the USA it shows you less, but it is interactive. The goal for the developed application was to combine the interactivity and give the user more ways of seeing a SOM.

2.1.1.3 SOM and Visual Data Mining

Visual data mining (VDM) can be used to gain more insight into a SOM. Jun Yan and Jean-Claude Thill created the SOMs using SOM Toolbox 2[4] and set up their own VDM environment with ArcObjects[5] and Visual Basic[6]. In addition to SOM maps and geographic maps, the prototype VDM environment also includes other common techniques of exploratory data analysis to assist the understanding of original data and the evaluation of the results such as scatter plot, star coordinate plot and parallel coordinate plot (Yan and Thill, 2008). They implemented four ways of linking the different visualizations (Figure 5):

---

3 http://geography.sdsu.edu/People/Pages/skupin/research/software/SeeSOMe/ (accessed: 12 June 2013)
4 http://www.cis.hut.fi/projects/somtoolbox/ (accessed: 12 June 2013)
5 http://resources.arcgis.com/en/help/arcobjects-java/api/arcobjects/ (accessed: 12 June 2013)
6 http://microsoft-visual-basic-2008.softonic.de/ (accessed: 12 June 2013)

- position: the position of data items remains fixed across visualization forms
- color: the same color is used for the same group of data items
- line: the same data items are connected by explicit lines
- motion: groups of data items are displayed one after another using animation

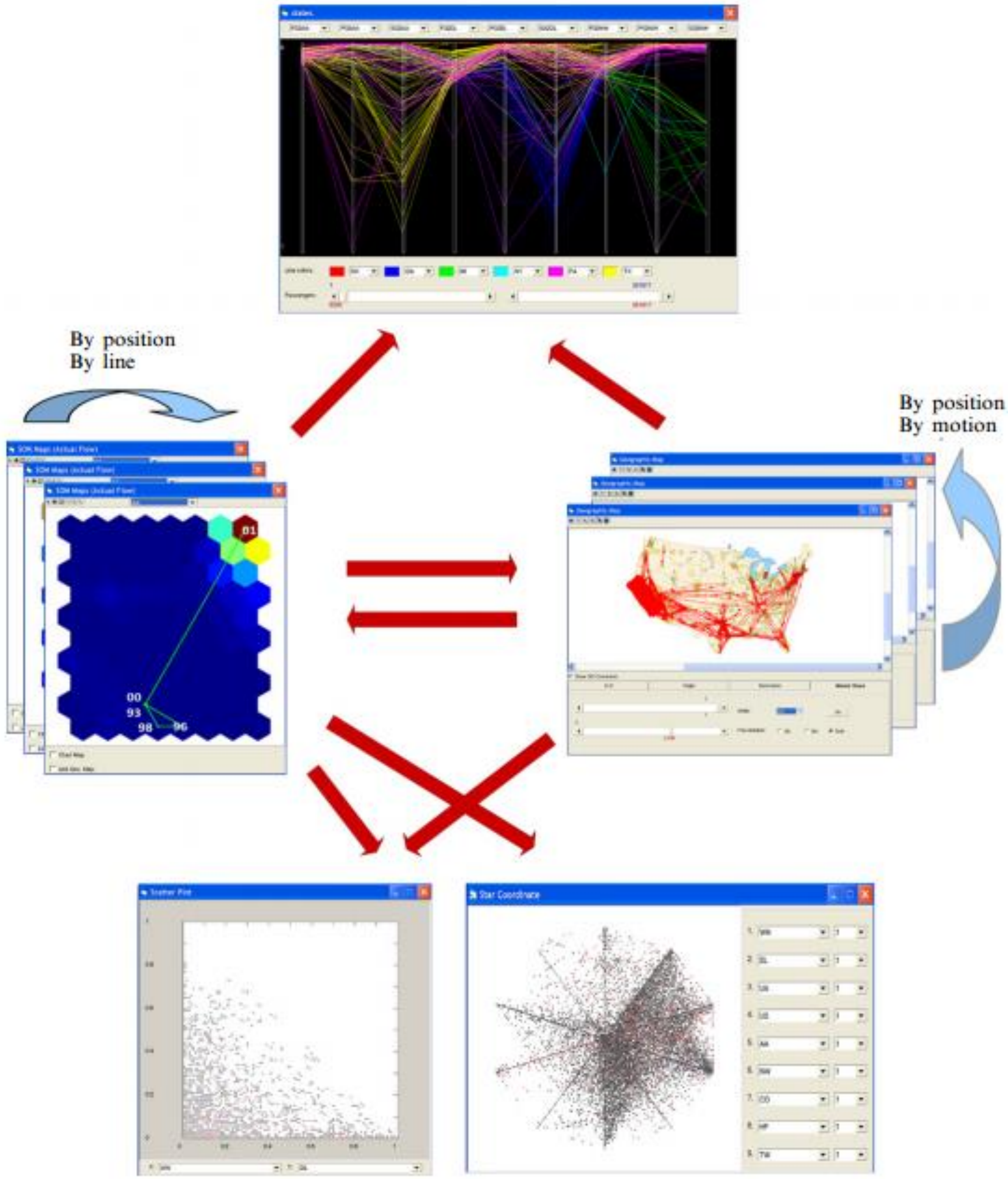


Figure 5: Different types of visualization (Yan, Thill, 2008)

2.2 Tri-space concept

This concept is based on the fact that data tables can be more informative when they are transformed into another perspective. You could have a table with all your locations listed and for each location you have a special attribute at a certain time (L-AT perspective). The same information could be presented as listed combinations of time and space which are assigned special attributes (LT-A perspective). The challenge with tri-space data is the transformation between different perspectives, especially in the presence of missing values (Skupin, 2010).

Table 1: Examples for the LA-T perspective (left) and the LT-A perspective (right)

	Temperature	Humidity
Villach / 12:00	12°	75%
Villach / 13:00	13°	76%
Klagenfurt / 12:00	10°	80%
Klagenfurt / 13:00	11°	81%

	12:00	13:00
Villach / Temp.	12°	13°
Villach / Hum.	75%	76%
Klagenfurt / Temp.	10°	11°
Klagenfurt / Hum.	80%	81%

In the two parts of the table (Table 1) the same values are represented. Here you see that easily, but if you imagine a data set of seven crime types in fifty states for forty years, this table would hold 14.000 values. The goal of the tri-space concept is to make it possible to switch between the different representations without losing or aggregating data.

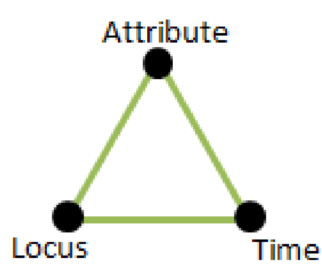
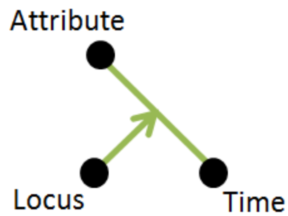


Figure 6: Dimensions of the tri-space concept

The tri-space concept consists out of 6 dimensions, the three single elements (e.g. attribute) and the three combinations of two elements (e.g. attribute and time). The single elements are represented by the corners of the triangle and the combinations are the edges (Figure 6).
There are also 6 perspectives, which are combinations of the dimensions. You have to combine a single element with a combined element to get a perspective; of course the combination has to contain attribute, time and location.



	A1	A1
	T1	T2
Loc1		
Loc2		
Loc3		

Figure 7: L-AT perspective

In this case (Figure 7) the locations are the objects and the attribute-time combinations are the properties. This perspective is perfect for presenting the data on a map like we now it.

2.2.1 Multi-year snow dynamics

Wang et al. (2013) demonstrate an implementation of the tri-space concept for environmental data. A medium-resolution SOM was applied to a snow water equivalent (SWE) data set covering the Northern Hemisphere over a 20-year period, with high temporal resolution. One of the resulting visualizations was similar to Figure 3, the other approach shows how development over time can be made visible with the help of the tri-space concept (Figure 8).

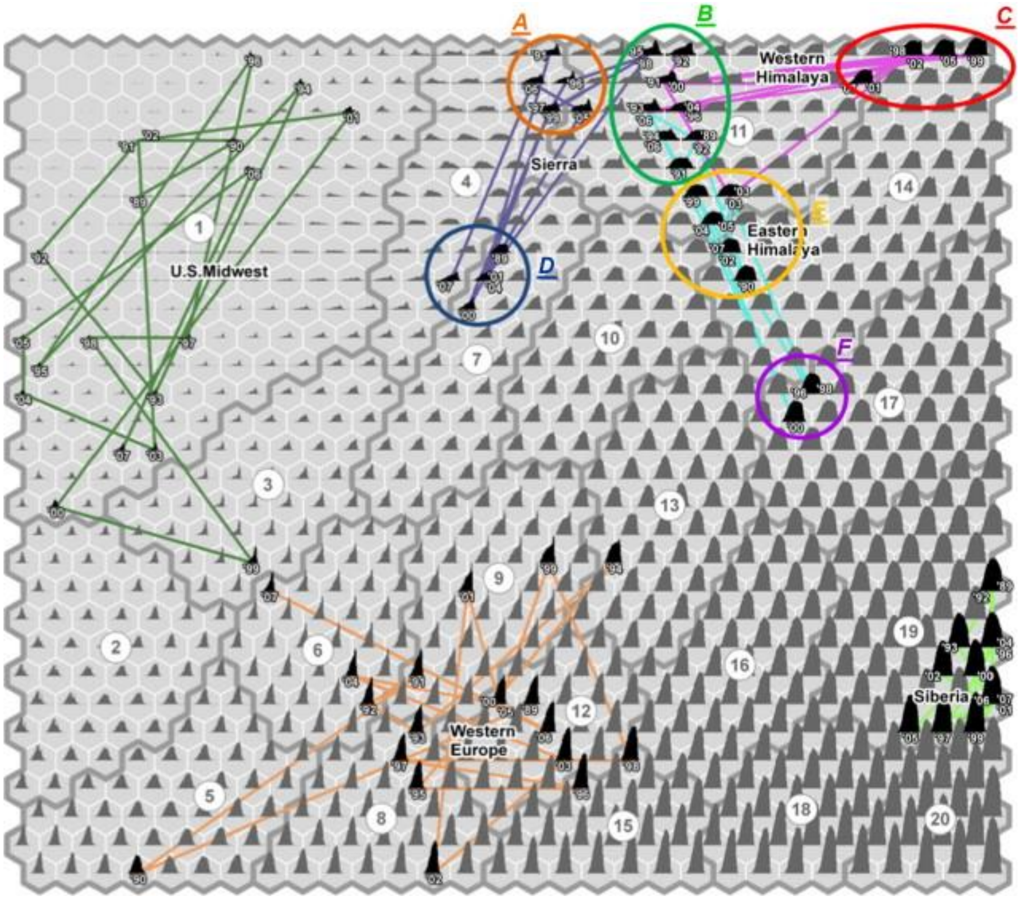


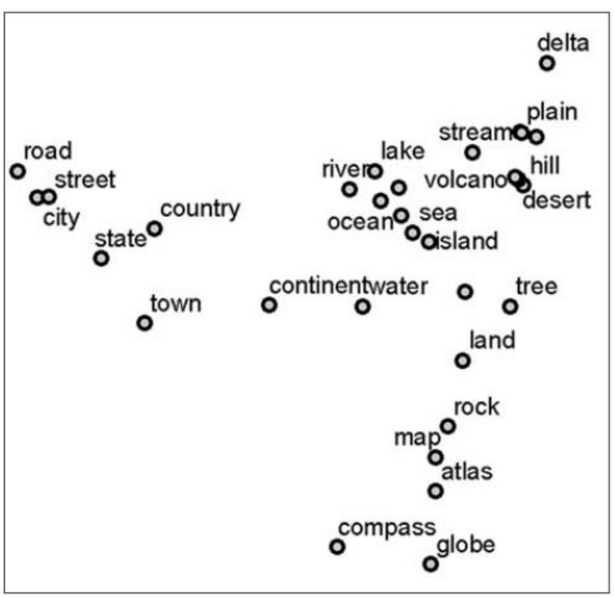
Figure 8: SOM with a niveograph for each neuron with trajectories of six sites in different colors presenting the temporal pattern. (Wang et al. 2013)

3. Methodology

This chapter contains the present problems to deal with and the way to solve them, as well as the description of the project area, the data and the implementation.

3.1 Problem definition

Data manipulation basically means derivation of new data from existing data for more convenient or more comprehensive analysis. (Andrienko, Andrienko, 2006) As the project is based on multivariate data of 50 countries, 50 years and 1250 attributes it is hard to discover novel patterns with basic global statistical methods, such as correlation coefficients. There are methods which are specialized on working with multivariate data like MDS (Kruskal and Wish, 1978). MDS operates on similarity matrices by calculating the stress function between the input matrix and a matrix with the calculated differences in n-dimensional space. It continues doing that until the stress value is low enough. In comparison to MDS SOMs tend to tear the input vectors apart when laying them out to make them fit the (most of the times) square view. But the problem with MDS is that input vectors tend to lie above each other and therefore it is harder for the user to get all the information he could (Figure 9).



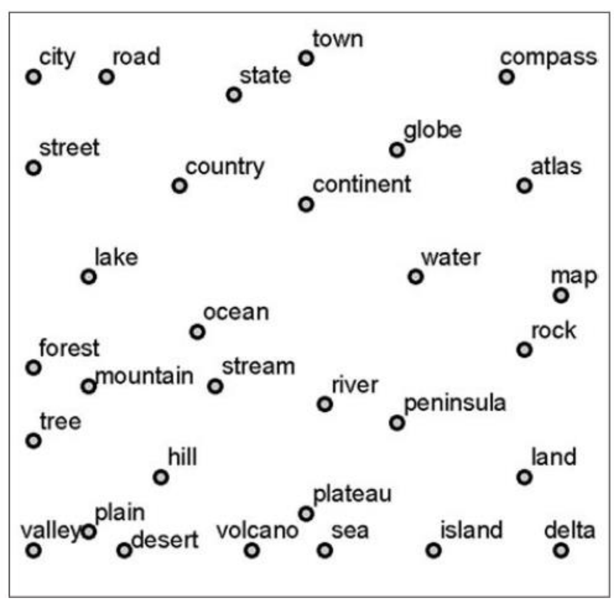


Figure 9: Comparison of MDS and SOM for a human subject test, based on five variables (Skupin and Fabrikant, 2003)

For this purpose the decision was for SOM to provide a visualization, which provides more information on the first sight. For non-expert users it is hard to read information out of a SOM without help. That is the reason why other methods had to be evaluated to find out which of them are useful to provide more information about the given data. As the socio-economic indicators are given on a country scale it makes sense to use a geographic map. Deciding which visualization method to use for the attributes was a challenge, because there are many different methods. Using charts would only make sense if the data set didn't have so many values, they are limited in the attributes they can show by their size, the more attributes you put in there the bigger the chart has to be. A visualization that can be re-sized easily would be a scatter plot. But even if you use different colors and sizes for the points in it and the two axes to show different values as well as a time slider for showing development it would only be able to show 4 variables plus the time, which is definitely not enough for this purpose. The solution for this problem is a parallel coordinate plot (PCP) (Inselberg and Dimsdale, 1991), which is able to show as many variables as wanted and can be put into a scrollable panel to avoid size issues. One of the most important advantages of the display of multivariate data sets in parallel coordinates is that it transforms the search for relations among the variables into a 2-D pattern recognition problem. (Inselberg, 1999) This example (Figure 10) shows some of the advantages and disadvantages of parallel coordinate plots. It is immediately (i.e. pre-attentively) apparent that there is a "trade-off" between attributes A and B as well as a strong correlation between B and C. Nevertheless, limitations can be identified. For example, for the data shown in Figure 10, it is not apparent that there is also a "trade-off" between B and E and a strong correlation between C and G; in other words, the ordering of the attributes can significantly affect the ease with which relationships can be identified. (Spence, 2007)

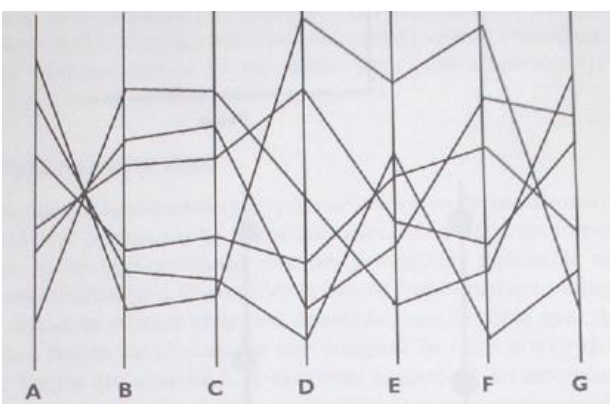


Figure 10: Example for a parallel coordinate plot (Spence, 2007)

Figure 11 shows how parallel coordinate plots work in combination with different categories which are presented in different colors. Here you can see the patterns more clearly, but the order is still very important for the chance to see patterns.

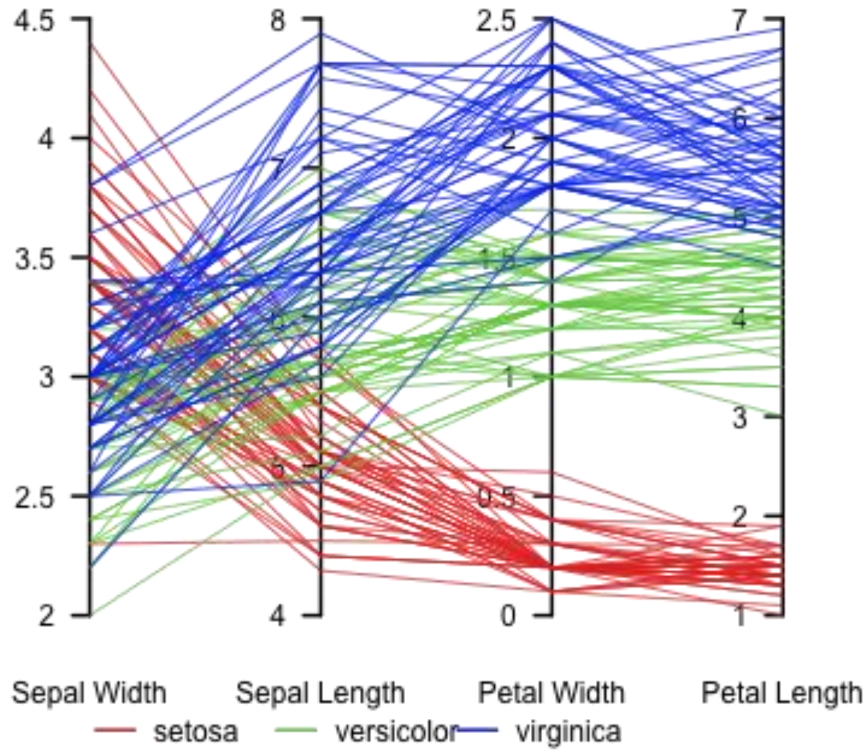


Figure 11: Example for a parallel coordinate plot with different categories[7]

To help the user understand the patterns which are presented with the help of the different methods even more, they will have to be interactive. To solve this problem all different views will be sharing the same color scheme and a list to choose what will be highlighted in all views. Another challenge is the changing between tri-space perspectives. To be able to provide a version to the user that is fast enough all perspectives and their data have to be loaded in the beginning to allow quick switching between them.

7 http://orange.biolab.si/screenshots/ (accessed: 15 June 2013)

3.2 Method of solution

To fulfill the requirements a project structure had to be defined. This structure was used to create the packages and classes in eclipse. The created classes are shown in Figure 12:

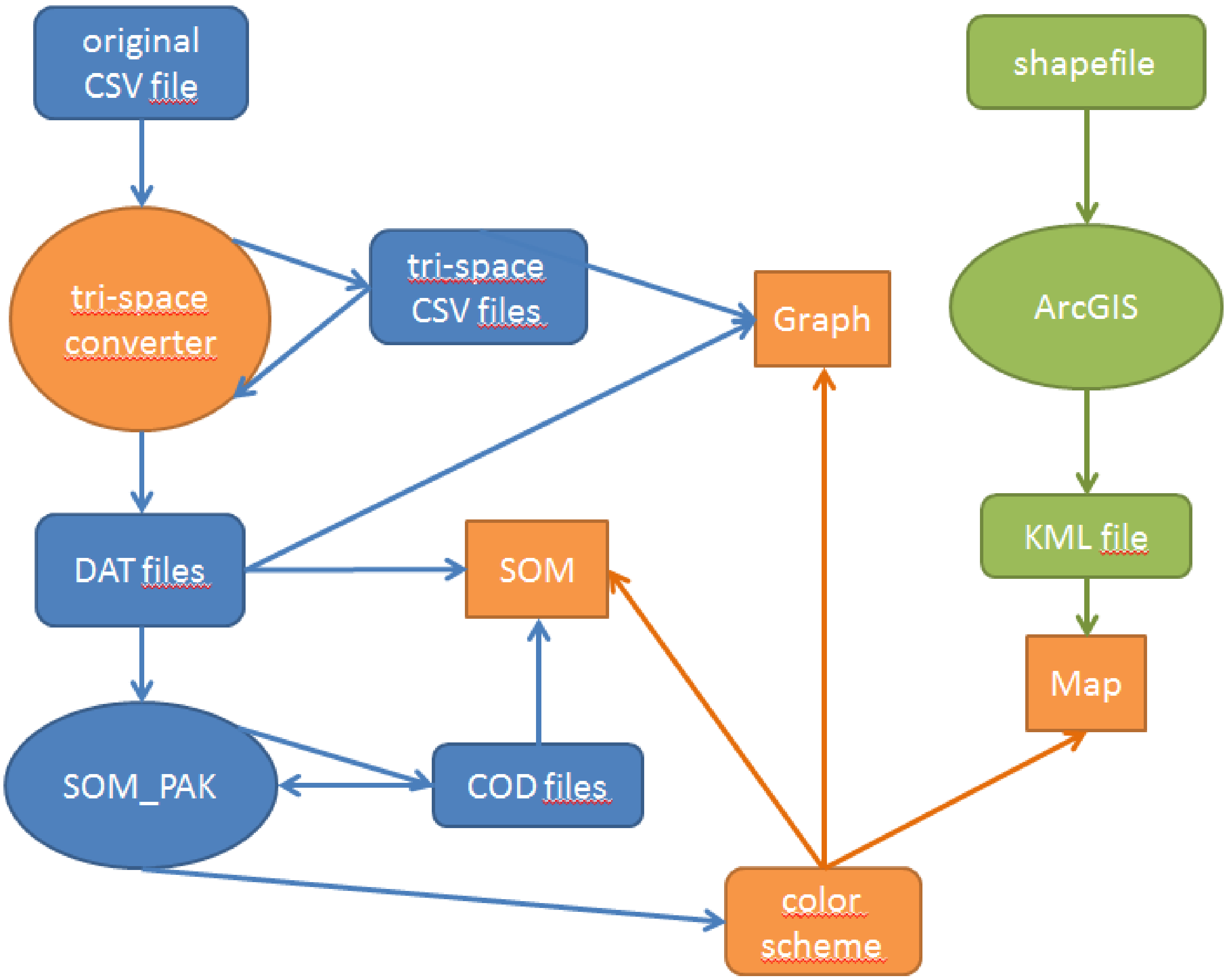


Figure 12: Scheme of the programs and data used

The orange parts of the scheme are the programs that were created during the work on the thesis. The blue parts are the data and the program used to transform it (SOM_PAK). The green parts are the data and program (ArcGIS) used for the geodata. The oval shapes are programs, the rounded rectangles are files containing the used data and the rectangles are parts of the graphical user interface.

The process which is described here works this way: the original CSV file is run through the tri-space converter application to create tri-space CSV files. Those are run through the tri-space converter again to create the DAT files, which are needed for the external program SOM_PAK. SOM_PAK creates

COD files out of them and then the COD files are fed into SOM_PAK again to create the smaller SOMs building the color scheme. The results of these processes are needed for the parts of the graphical user interface. The SOM part is made out of the COD and DAT files, the graph out of the tri-space CSV files and the DAT files. The color scheme is needed for the two already mentioned parts and the map part. To complete the map part there is one file missing. Therefore the shapefile was converted to a KML file with the help of ArcGIS.

As the application should help the user to understand the correlations between different socio-economic indicators in Africa, the users is not supposed to be able to upload new data. Instead, the application has to provide sample data sets to work on. In this case, the different data sets contain data from the World Bank. These data are provided in CSV (Comma/Character Separated Values) format. With the help of a self-developed tri-space converter tool these data are converted into the different perspectives. Afterward the files are converted into DAT (input file format of SOM_PAK) files with the same tool. Out of the DAT files, SOM_PAK (Kohonen, Hynninen, Kangas, Laaksonen, 1995) can create the SOMs in COD (output code book file format of SOM_PAK) format. The reason to use SOM_PAK was easy: to make it easier for others to create their own sample data and show it with the application. Afterward the COD files are used as a kind of DAT file and projected onto a one by twelve SOM to compute the color scheme. Additionally an ESRI shapefile had to be converted into KML (Keyhole Markup Language) with the help of ArcGIS to be usable as geographic map basis.

The application is deployed as Java applet. The reasons for that are:

- It can be put into a website easily, which means that users can access it without being forced to installing something on their computer.
- It is faster than the alternatives Java standalone application and JavaScript (Table 2, Figure 13)

Table 2: Drawing Hexagons in Seconds

	Java Applet	Java Standalone	JavaScript
100x100	2	3	4
1000x1000	20	31	45

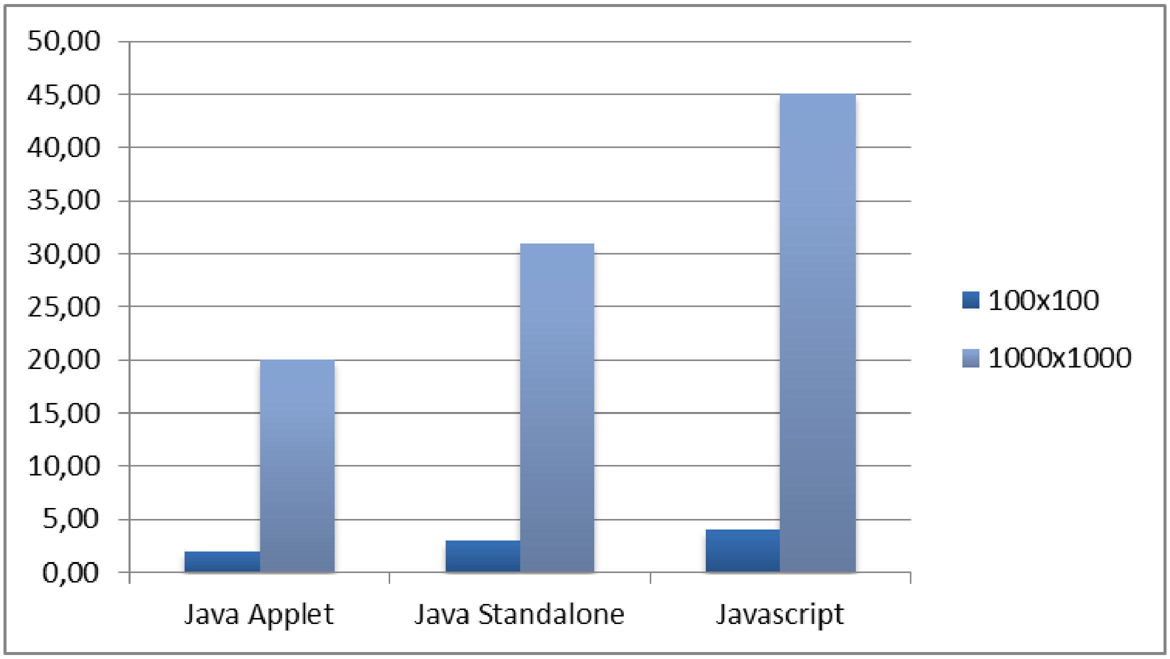


Figure 13: Drawing Hexagons in Seconds

These times were measured using the three different deployment types in a simple website, because it was a main goal to give the users the chance to access the application via internet.

3.3 Project area

The project area is Africa. The reason for that is that the application can help understanding developments and changes over time which is very helpful for developing countries. To have the chance to see a development over the years the used data reaches from 1960 to 2012. It also can help seeing structures and similarities between countries which are not obvious in the first place.

3.4 Data

The data sets to calculate the SOMs and present the socio-economic indicators were downloaded from one of the World Bank databases called world development indicators[8]. These data are available for the whole world from 1960 to 2012 and for all 1287 indicators. The indicators can be divided into 18 categories[9] that range from agriculture and rural development over health to science and technology. That enables the users to get broad overviews as well as detailed information about countries. The data can be

[8] http://databank.worldbank.org/data/views/variableselection/selectvariables.aspx?source=world-development-indicators# (accessed: 12 June 2013)

[9] http://data.worldbank.org/indicator/all (accessed: 12 June 2013)

downloaded as Excel (2003), tabbed text, SDMX (Statistical Data and Metadata eXchange) or CSV files. For our purpose the CSV files were used, because the Excel files of 2003 have a limit of 65536 rows by 256 columns which are not enough for all the data and they are very easy to read for humans as well as for a program.

As already mentioned this data was transformed into DAT files to calculate SOMs with SOM_PAK. A DAT file starts with a number which specifies the dimension. This dimension defines the values from which the SOM will be calculated. For example in Figure 14 the computation will only be based on the first three double values and will ignore the color string in the end. DAT files can also contain comment lines, which start with a #. The COD files contain as many lines as the given dimension told it to, in this example there are six lines because the dimension is 3 by 2. The lines contain the values calculated based on the algorithm described in chapter 2.1.

```
3
# First the yellow entries
181.0   196.0    17.0   yellow
251.0   217.0    49.0   yellow
# Then the red entries
248.0   119.0   110.0   red
213.0    64.0    87.0   red
```

```
3 hexa 3 2 bubble
191.105   199.014   21.6269
215.389   156.693   63.8977
242.999   111.141   106.704
241.07    214.011   44.4638
231.183   140.824   67.8754
217.914   71.7228   90.2189
```

Figure 14: Examples for DAT and COD file (Kohonen, 2001)

Additionally color scheme files were created, those are basically a DAT and a COD file as well, because for assigning the color we also used the concept of SOM. The original SOM is used as DAT file and projected onto a one by twelve SOM grid which represents the RGB (Red Green Blue) color scheme (Table 3).

Table 3: RGB scheme used in the application

Name	Red	Green	Blue	Color
Red	255	0	0	
Orange	255	125	0	
Yellow	255	255	0	
Spring Green	125	255	0	
Green	0	255	0	
Turquoise	0	255	125	
Cyan	0	255	255	
Ocean	0	125	255	
Blue	0	0	255	
Violet	125	0	255	
Magenta	255	0	255	
Raspberry	255	0	125	

3.4.1 Tri-Space

To give the user a chance to explore the data in different tri-space perspectives it was necessary to convert the data from one perspective to the other. The data from the World Bank was in the LA-T (Location-Attribute over Time) perspective (Table 3).

Table 4: A part of the data in the LA-T perspective

Country	Indicator	1990	1991	1992
ZMB	Armed forces personnel, total	16000	16000	16000
ZWE	Armed forces personnel, total	45000	45000	48000

Out of the different tables the SOMs are created. The transformation of the data into the LT-A (Location-Time over Attribute) and the AT-L (Attribute-Time over Location) perspectives was easy with the help of the self-built library for tri-space conversions. The conversions into the L-AT, T-LA and A-TL perspective also worked perfectly fine, but with the creation of the SOM a problem occurred. For example in the L-AT perspective, each input vector has 660 dimensions. Creating a SOM with 100 x 100 neurons out of this data means creating 10.000 lines with 660 values in each line, but as there are only 25 input vectors in this format, 100 x 100 neurons were not needed so those perspectives were calculated as 5 x 5 SOMs.

3.4.2 Geodata

To make the loading of geodata as easy as possible and give other programmers the chance to change the geographic map view the format KML is used. There is an enormous amount of KML files available online for nearly any part of the world and additionally ArcGIS and many other GIS (Geoinformation System) packages are able to convert different file formats into KML. For our purposes an ESRI shapefile of all the countries in the world was reduced to the countries of Africa and converted to KML. This shapefile was created by the Hoelzel publishing company (Figure 15) and was already used in other Geoinformation projects of the Carinthia University of Applied Sciences (CUAS). The reason for that is that the polygons in there are not as detailed as in most of the other data sets found but the data is still topologically correct. As there are fewer points for each polygon, the map view is able to load very fast.

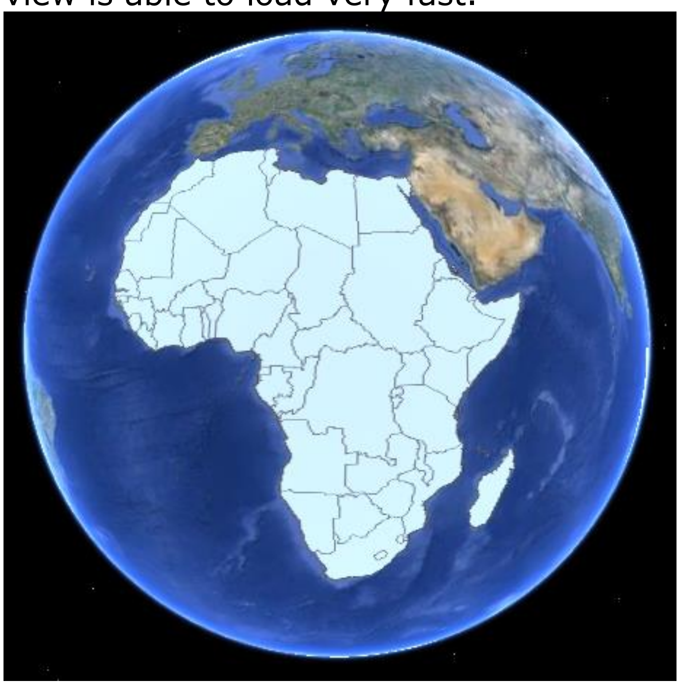


Figure 15: Google Earth Representation of the geodata

3.5 Implementation

The whole project, consisting out of this application and the work of my two colleagues Manuel Rainer and Michael Spöcklberger is called SOMatic. This Application itself has the title SOMatic – Socio-economic Indicators of African Countries. The implementation was realized in eclipse 3.7. The processing environment is great for small projects but to be able to manage more classes eclipse is perfect with the structure of its Package Explorer. Nevertheless, the processing 2.0 library was used. There are still some problems because most of the other libraries that normally work with processing are not yet available for the 2.0 version, but that is only a matter of months. The basic layout was created with the Swing library but for the additional Panel which contains all the controls the G4P[10] 3.2.3 was used. This is a Graphic library for Processing and great for creating control elements. To create the geographic map it was necessary to parse a KML file and especially its HTML part, which was done with the help of JSOUP[11] 1.7.2. The last part in the Implementation needing a library was the attribute view with the parallel coordinate plot, for that the JFreeChart[12] 1.0.14 library was used. This library also needs the help of JCommons and iText, but that was no additional effort, as it can be downloaded on the same website and will be linked to the project as soon as JFreeChart is added to the referenced libraries.

3.5.1 Graphical User Interface (GUI)

As already mentioned the GUI is based on the Java Swing layout, which is based on the Abstract Window Toolkit (AWT) and much more flexible and modular. (Loy, Eckstein, Wood, 2002) To implement the layout (Figure 16) a JFrame is filled with JPanels, each one contains a PApplet to make sure that the different components are working independently.

The three components are re-sizable and their colors correspond to each other. Those colors are received from the color file created during the data processing as mentioned in chapter 3.2. To make it possible that all the different applets know which colors are needed and which tri-space perspective is active right now, a so called PerspectiveHandler was created. The PerspectiveHandler is responsible for reading the DAT and COD files in the beginning and controlling the tri-space perspectives as well as coloring the geographic map accordingly. To make sure that all applets are on the same status of information, it is a so called Singleton. This concept results in the PerspectiveHandler only being instantiated once and all the applets get so called instances when they call it. (Gamma, Helm, Johnson, Vlissides, 1994)

Although the PerspectiveHandler is responsible for the timing when the COD and DAT files are loaded, the work is done by the FileHandler. This Java class and also most of the other classes handling the reading, behavior and calculation of SOM relevant topics were created by two colleagues of mine: Manuel Rainer and Michael Spöcklberger. Those classes contain the Best Matching Unit (BMU) Test and different Topologies like hexagonal and rectangular as well as the 2D Grid which is basically the graphical presentation of the SOM. Also the InputVector and Neuron class were created by them.

---

[10] https://code.google.com/p/gui4processing/downloads/list (accessed: 12 June 2013)

[11] http://jsoup.org/download (accessed: 12 June 2013)

[12] http://sourceforge.net/projects/jfreechart/files/ (accessed: 12 June 2013)

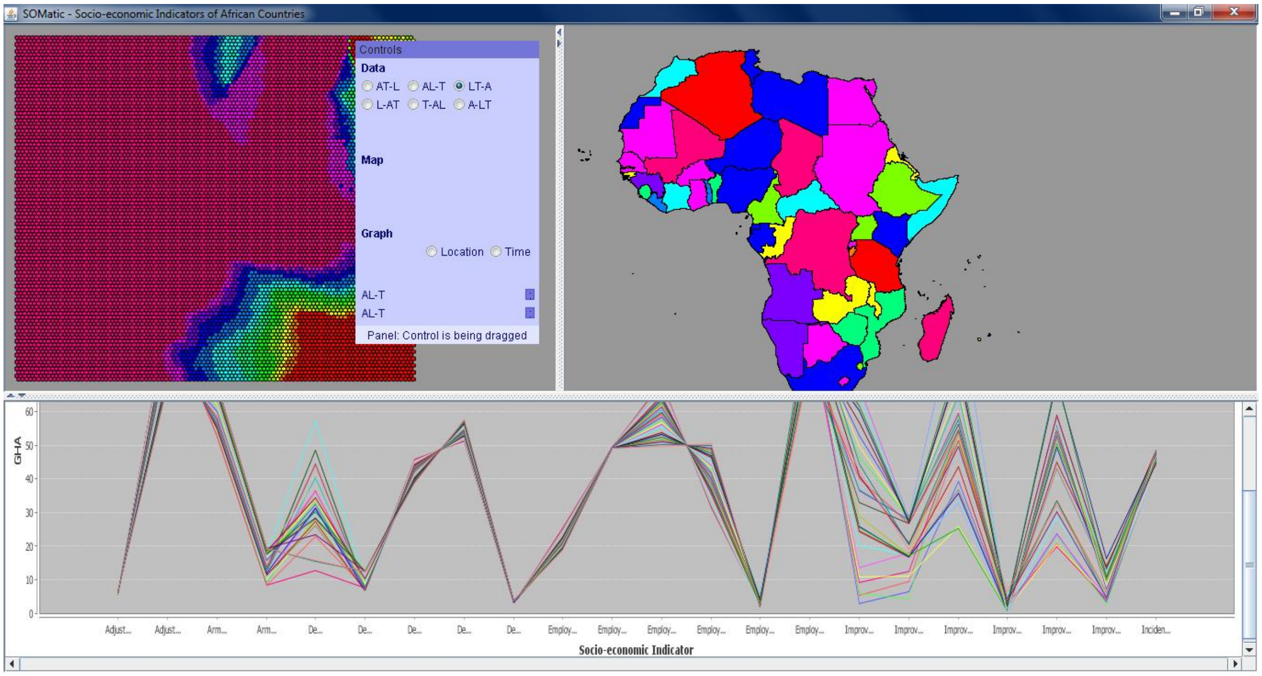


Figure 16: GUI of the SOMatic App

3.5.2 Self-Organizing Map

The SOM class is the applet seen in the upper left corner in Figure 16. As already mentioned a huge part of that was realized by my colleagues, this Applet puts all the parts together and adds the color component. The three implemented tri-space perspectives (AT-L, AL-T and LT-A) of course look differently when shown with the color scheme:

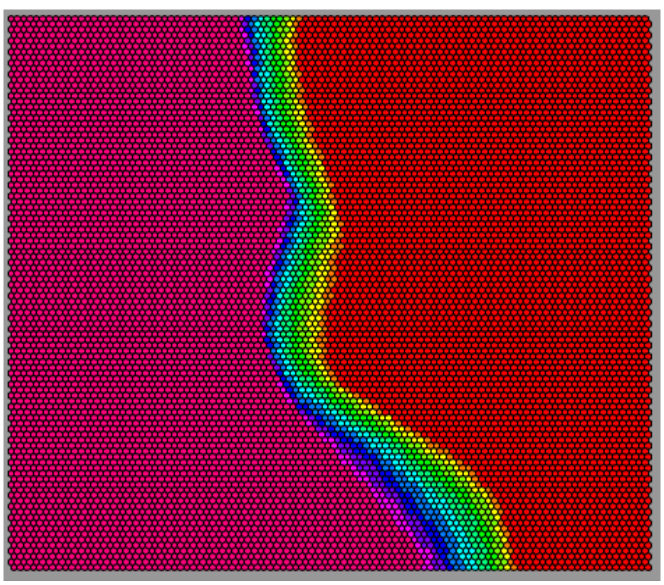


Figure 17: AT-L SOM color coded

The AT-L SOM (Figure 17) shows a clear border between the pink and red part. If you look at the data that makes sense, as for this SOM it consisted out of the urban population over the years for all countries and the rural population over the years for all countries. Of course the data for rural and urban population is exactly reverse. Although red and raspberry are similar if we think in normal human schemes, if you take a look at the color scheme (Table 1) you will see that red and raspberry are the parts which are as far apart as possible.

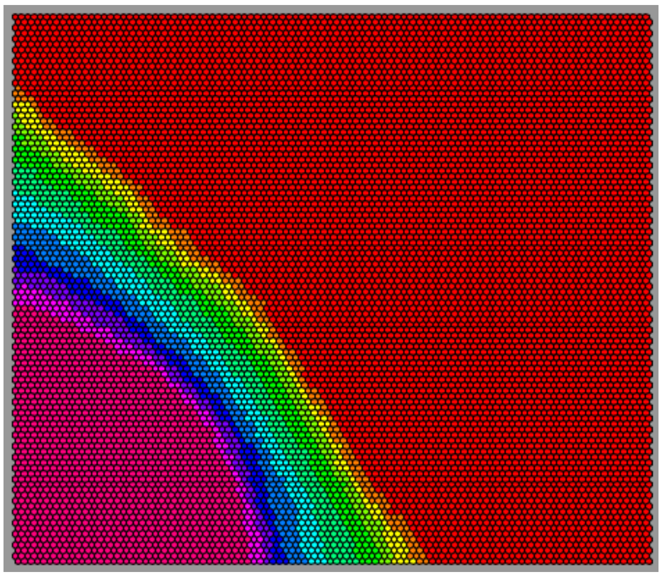


Figure 18: AL-T SOM color coded

The pattern of the AL-T SOM (Figure 18) is not as easy to interpret as the first one, but it is quite clearly bordered as well. The data contains basically 2 types of data, population statistics and land use. That those two types of variables correlate, makes sense as well. Especially as the tri-space perspectives creates input vectors, which show the development over time.

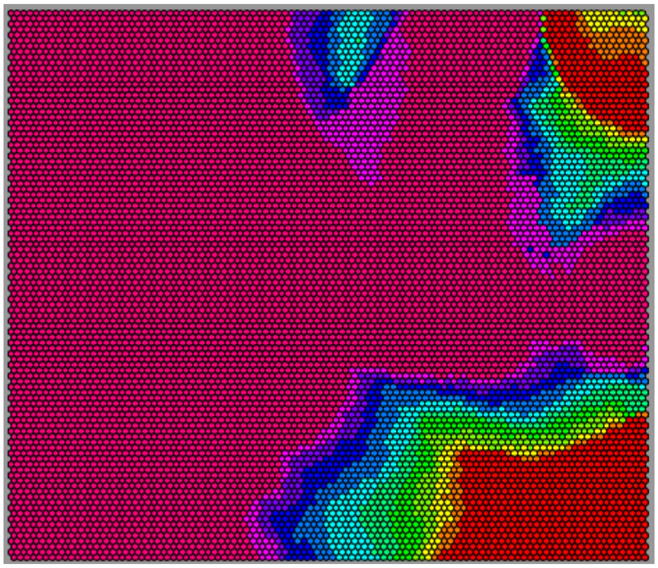


Figure 19: LT-A SOM color coded

The LT-A SOM (Figure 19) is even harder to interpret. But that also can be explained by looking at the data. It contains a lot of very different attributes and the patterns are those of a country in a year for all those attributes. But if you look at the map for this perspective for 1995 you see that many of the countries are colored in the dominant color raspberry (Figure 20). You also see that the color is especially dominant in the north western and western parts. With further analyzes and taking a look at the component planes it would be possible to find the reason for the pattern in the SOM more in detail.

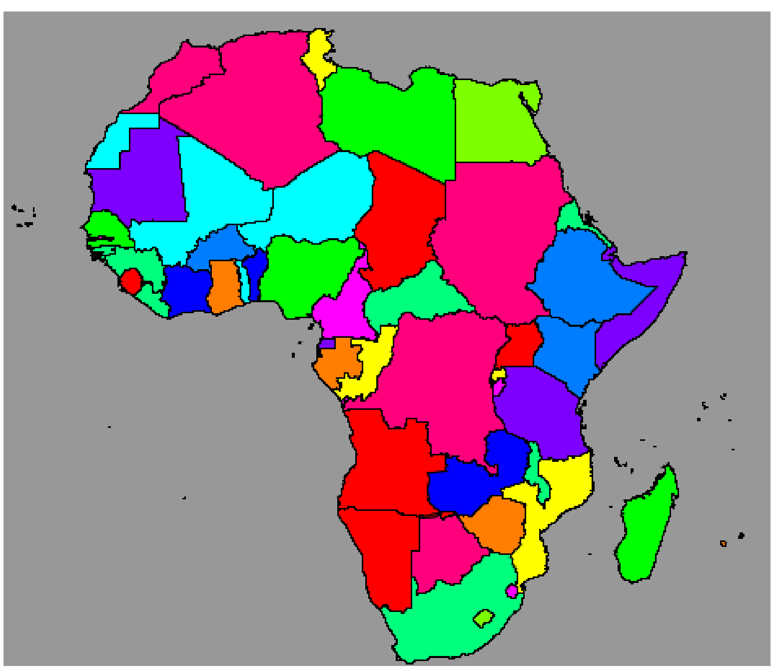


Figure 20: Map of 1995 for LT-A perspective

The L-AT SOM (Figure 21) is only five by five neurons, the reasons for that are mentioned in chapter 3.4. Here it is really difficult to see anything else than the colors running from the lower left edge to the right, upwards and back to the left. In this case each input vector represents one country.

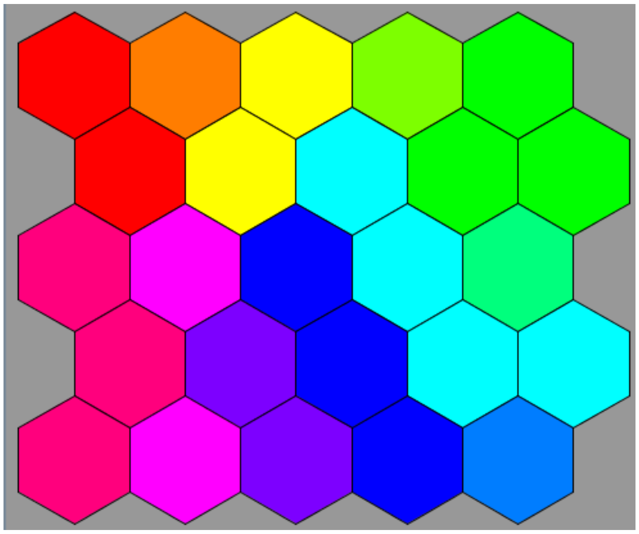


Figure 21: L-AT SOM color coded

3.5.3 Geographic Map

As already mentioned in chapter 3.4.1 the map was produced using a KML file. The map can be used for finding geographic patterns which are not clear when you look at the SOM. The map can only be colored when the perspectives AL-T, LT-A or L-AT are chosen, because only their input vectors contain the location which the map needs. Based on the BMU test of the input vector the color can be found and saved as meta data. Additionally to the correct perspective, the user has to choose either a year (for the LT-A perspective) or an attribute (for the AL-T perspective). Of course he does not have to choose that if the L-AT perspective is chosen, because this results in only one map. Only with that additional information the exact input vector can be determined and its color can be used. In Figure 20 you already saw the map of 1995 for the LT-A perspective. Some other examples are the "Agricultural land (% of land area)" for the AL-T perspective (Figure 22).

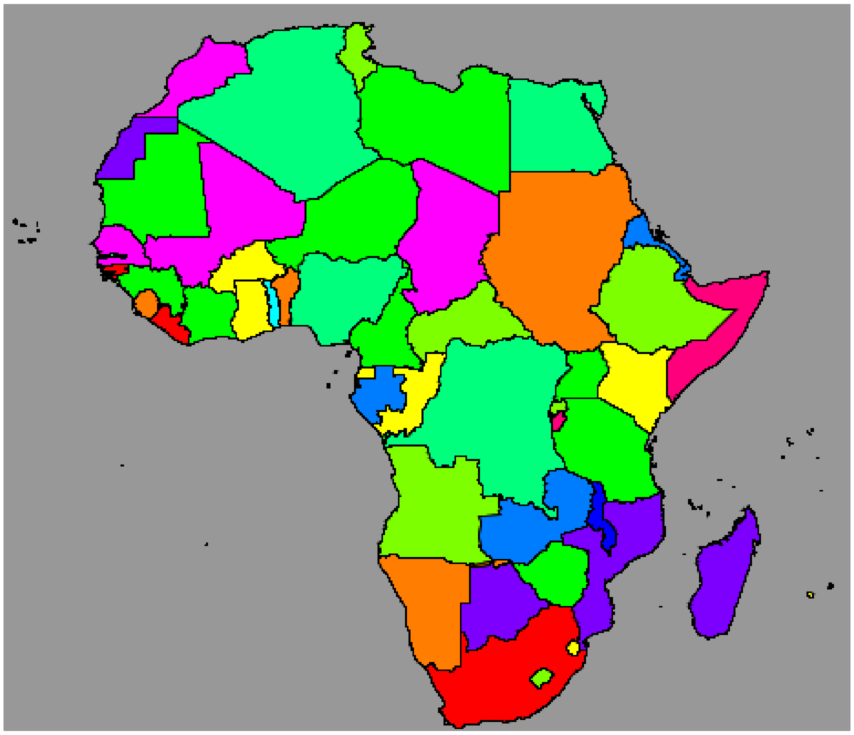


Figure 22: Map of Agricultural land (% of land area) for AL-T perspective

As the data is received from a KML file generated for Google Earth, not only the library for decoding the KML part is necessary but also a part for decoding the HTML part of the description to get the meta data about the polygon. The KML parsing was done with the help of the Java XML parser library. For the HTML parsing the JSOUP library was used.
For the polygons a self-made class consisting of 2D vectors was used. Those 2D vectors get rescaled to fit the screen (Figure 23).

```
private ArrayList<Vector2D> createBoundaries(String coords){
    String[] xyzs = coords.split(" ");
    ArrayList<Vector2D> vecList = new ArrayList<Vector2D>();
    for(int i = 0; i < xyzs.length; i ++){
        String vector = xyzs[i];
        String[] xyz = vector.split(",");
        double x = Double.valueOf(xyz[0]);
        double y = Double.valueOf(xyz[1]);
        x = (x + 28) * 5.4;
        y = (y * -1 + 42) * 5.4;
        Vector2D vec = new Vector2D(x, y);
        vecList.add(vec);
    }
    return vecList;
}
```

Figure 23: Code sample showing the createBoundaries() method

3.5.4 Parallel Coordinate Plot

For the parallel coordinate plot the user has to choose one out of the available dimensions. For example if the active tri-space perspective is LT-A the two dimensions are location and time. Then one of the possibilities listed up has to be chosen. For this example the location Ghana was chosen. The resulting plot contains as many lines as there are different attributes and is shown for all the different years (Figure 25).

As already mentioned the parallel coordinate plot is based on the java library JFreeChart. It was created using the feature of categories in a simple line chart (Figure 24). In a DefaultCategoryDataset when creating a value you have to mention a series. All parts of the data set with the same series are one line, and those series can be colored different. So this solution is perfect for showing correlations by using the same colors. When you hover over a line at one of the kinks you get information about the series at that point. If you look at Figure 25 you will see, that that can be tricky sometimes because the series lie extremely close together.
That is where the fact that all of the applets are re-sizable comes in handy. The bigger the height of the applet gets, the bigger the differences between the series are.

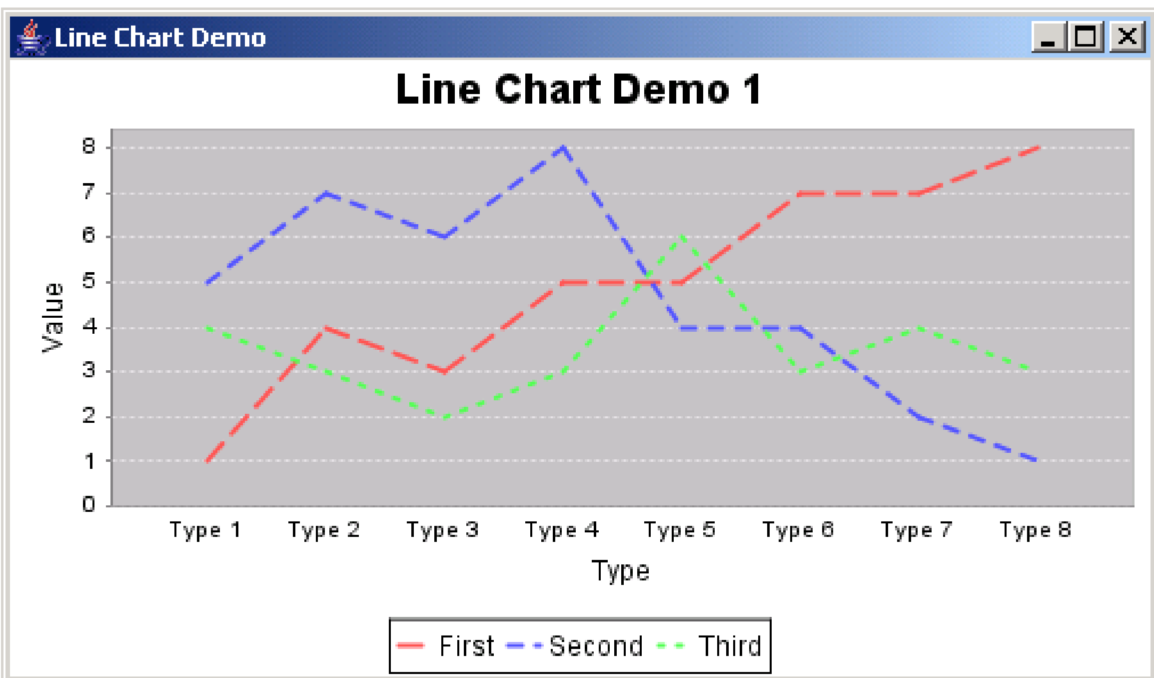


Figure 24: simple Line Chart[13] based on JFreeChart

To interpret the graphs you have to understand the principles of parallel coordinate plots. They were mentioned in chapter 3.1. Every time the lines

13 http://www.java2s.com/Code/Java/Chart/JFreeChartLineChartDemo1.htm (accessed: 12 June 2013)

show extreme differences between two dimensions they are negatively correlated, for example in Figure 25 between the first two dimensions. A great example for two positively correlated dimensions is shown in Figure 26 between DJI and EGY; those abbreviations stand for Djibouti and Egypt, so it makes sense that their rural population behaves the same as they are both located in the north-east of Africa. The order in which the dimensions are presented is based on the order in which they are read from the CSV files. As also already mentioned in chapter 3.1 the information gain would be a lot bigger if the user would be able to change the order of the dimensions to see the differences between for example Angola and Uganda.

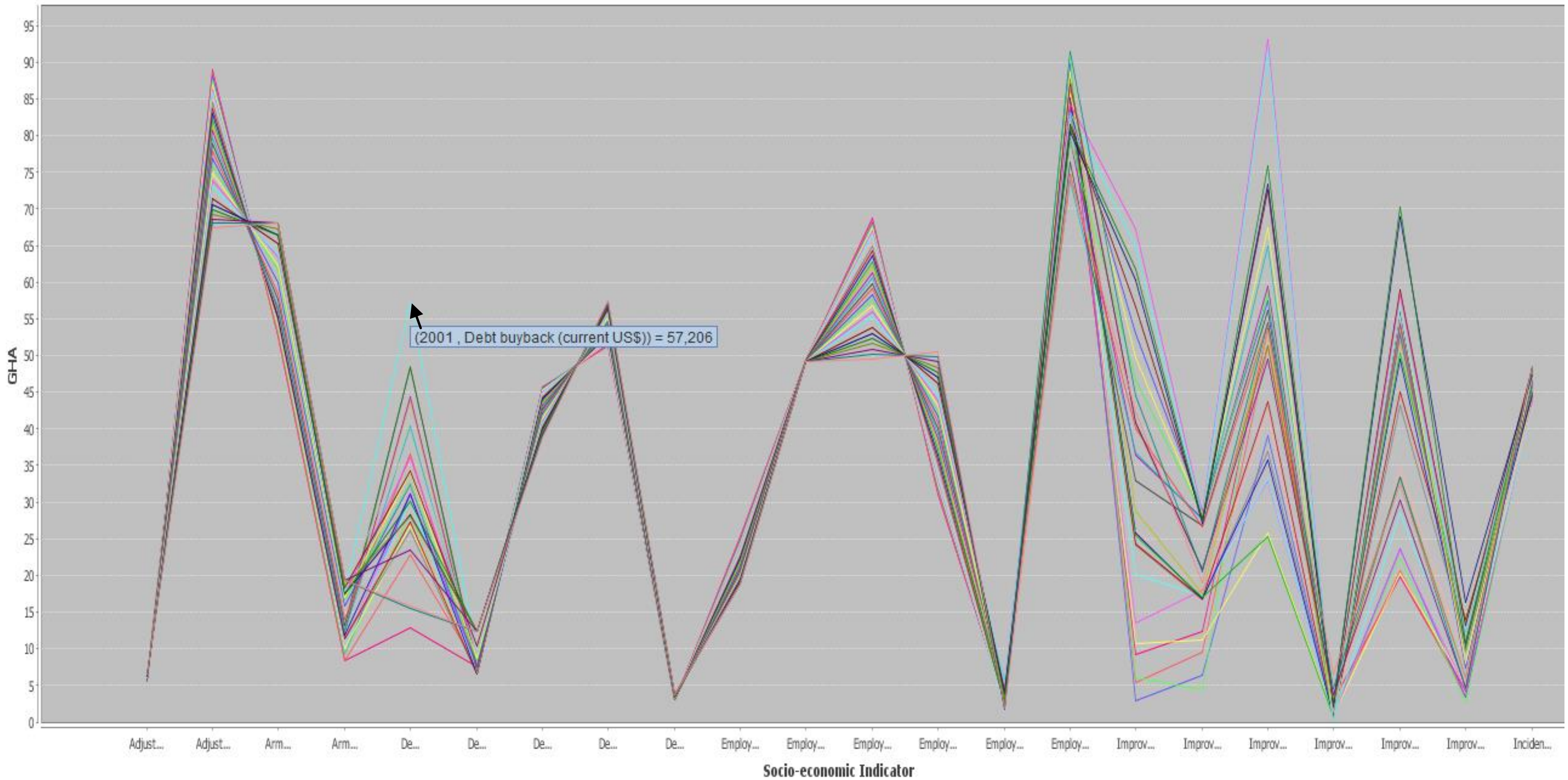


Figure 25: Parallel Coordinate Plot for Ghana in the LT-A perspective

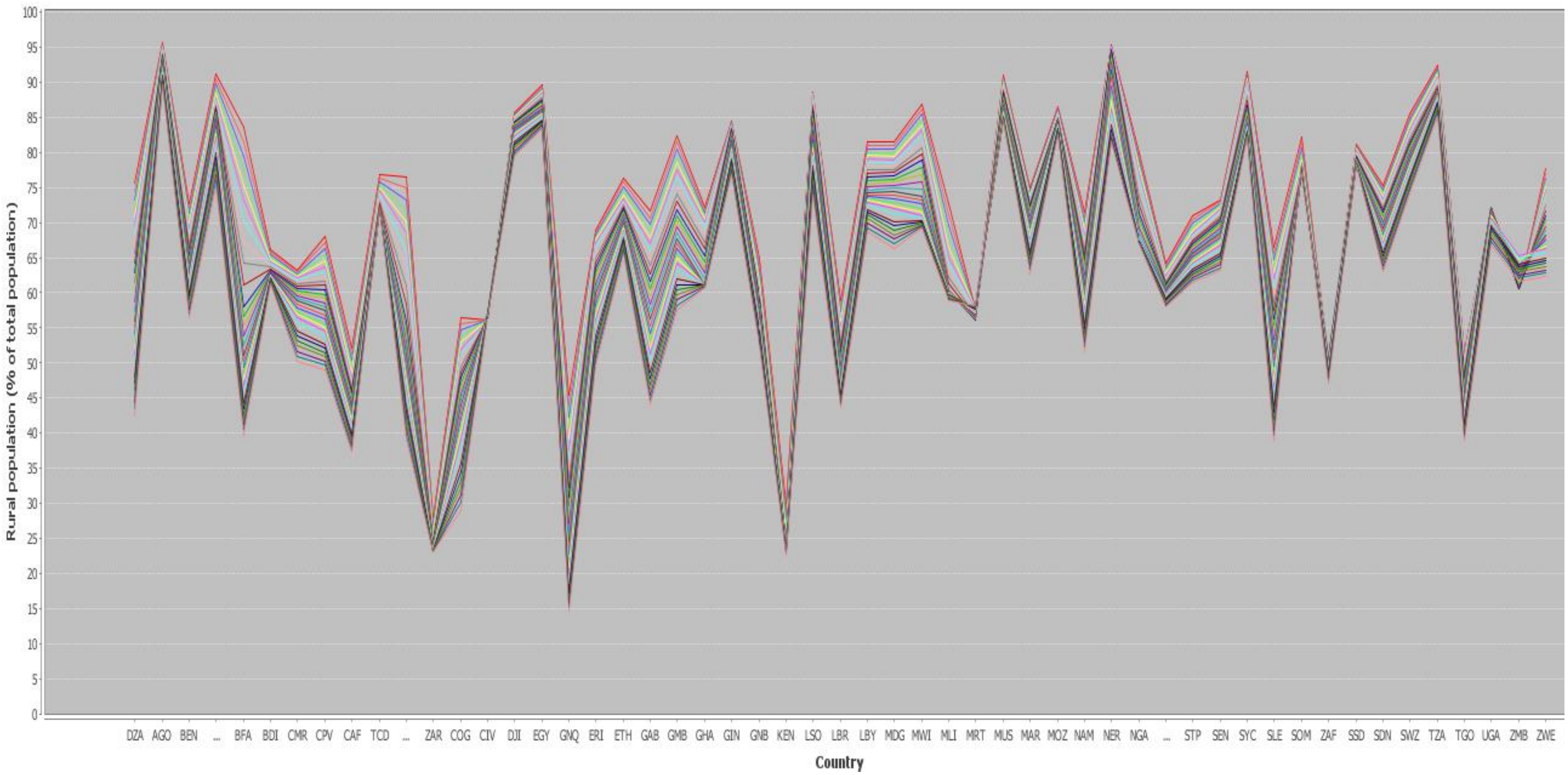


Figure 26: Parallel Coordinate Plot for Rural Population in the AT-L perspective

3.5.5 Control Panel

The control panel (Figure 28) was created with the G4P library. This library contains a huge amount of different control elements for nearly all purposes. In this application the controls used are shown in Figure 27.

```
private GPanel pnlControls;
private GLabel lblData, lblMap, lblAction, lblGraph;
private GOption optAToverL, optALoverT, optLToverA, optAttribute,
                optLoverAT, optToverAL, optAoverLT,
                optLocation, optTime;
private GToggleGroup tg;
private GDropList listCountry, listAttr, listLoc, listTime;
private GSlider slideYear;
```

Figure 27: Code sample showing parts of the control panel

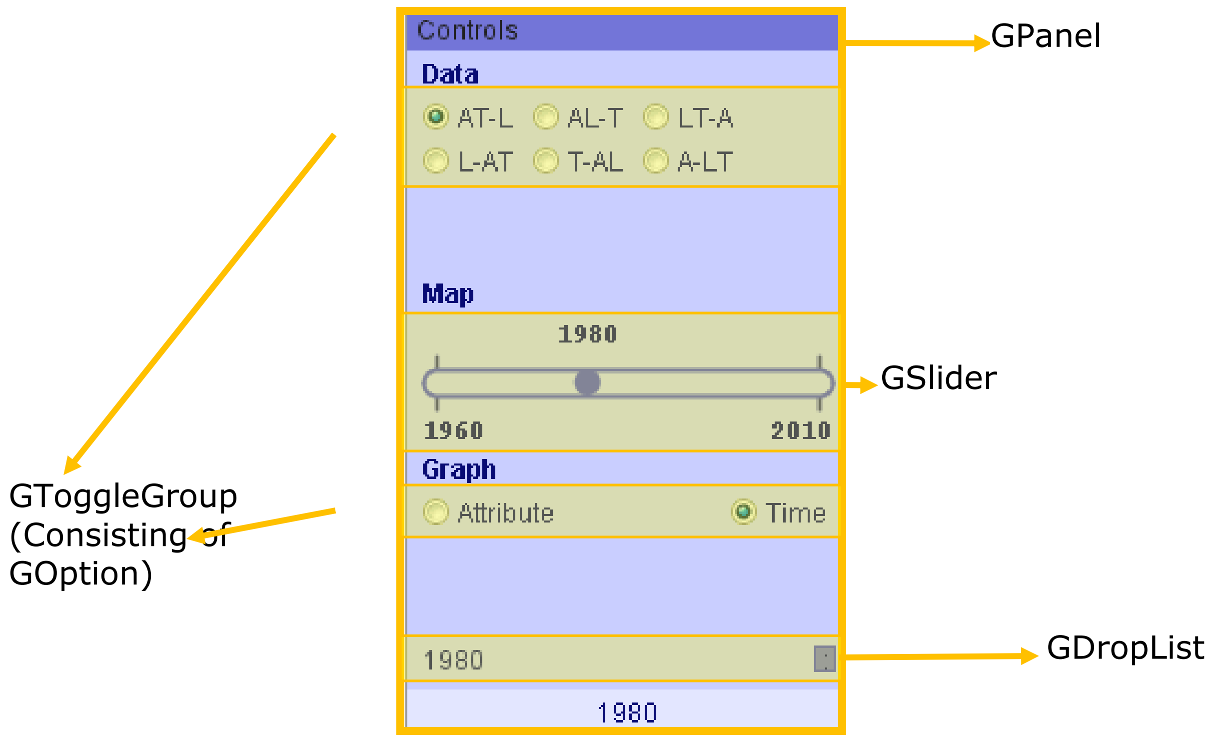


Figure 28: Control Panel with all its components

The controls on the panel change according to the chosen tri-space perspective. For example when the AT-L perspective is chosen the user sees a slider with the available years to change the map accordingly. If the AL-T perspective is chosen this slider will be substituted by a drop down list for the available attributes to be shown in the map. Also the graph part of the controls changes. As it makes sense for the graph to only show one attribute at the time (different attributes would cause problems with the units) the

user can choose which attribute he wants to see. If he or she prefers to see one time slice in particular the radio button "Time" changes the drop down list to a list of available years. This only makes sense if all the attributes have values in the same range for example 0-100%. If another perspective is chosen, for example AL-T the two radio buttons which are visible will be "Attribute" and "Time".
I would recommend this library to everybody who is experienced in using the Java Swing GUI, because it is very similar. I would also recommend it to beginners because it is easy to use and there are many tutorials and samples online and in the library itself.
The library also gives the programmer different color schemes to use which makes it appropriate for any software without complicated color changing for each component.

3.5.6 Converter Library

To be able to create all the different SOMs the CSV data had to be converted into the different tri-space perspectives. For this purpose a whole new library was created. This library consists of a TriSpaceReader. This class reads the CSV file, no matter what format it is in, and saves it as a TriSpaceObject. One TriSpaceObject consists of an attribute, a location, a time and the value. This creates the possibility to put the objects together in whatever format, order and selection you need. If the data should be saved as another CSV file the TriSpaceWriter is used. It takes the TriSpaceObjects and puts them together however needed. If a DAT file is needed to be able to calculate a SOM out of it the DatWriter is used. It basically does the same thing as the TriSpaceWriter except for one big difference: It ignores lines with missing values. It needs to do that because SOM_PAK is not able to work with missing values. In this case all the transformation have been made into CSV as well as DAT files, but as already mentioned, only some of the perspectives are used in the application.
For this library there was created a GUI (Figure 29) as well, this makes it possible to use it from outside and without the SOMatic application.

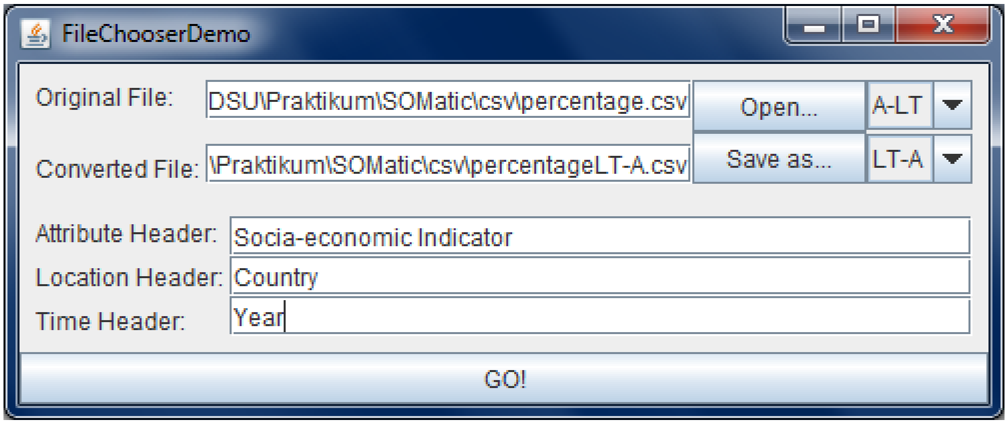


Figure 29: tri-space converter GUI

3.6 Summary

The processing library was very helpful, because the PApplet class contains a setup() and a draw() method. The setup() method is called only once per sketch run, at the start. It can be used to define the base sketch setup, like its size, its mode or to create an instance of the PerspectiveHandler. The draw() method is called on each frame: a sketch is like a movie, it is made of successive frames, images, and their quick succession (by default around 60 per second) makes the eye believe there is a continuous movement. That is also the reason that the control panel is moving around in the PApplet, it doesn't work like a normal window but it is drawn so often that you think so. Those two functions make sure that things that only have to be done in the beginning are not repeated and therefore they need less memory space and are less CPU-intensive (Central Processing Unit).

4. Results and Interpretation

This chapter describes the created results, the SOMatic application and the tri-space converter library.

4.1 SOMatic – Socio-economic Indicators of African Countries

This application is able to give the user more information about a sample data set with the help of SOM, a geographic map and a parallel coordinate plot. Those three views are controlled by one panel and share the same color scheme to make correlations more obvious.
The application is divided into different packages to make it more reusable and modular. The packages are:

- somatic.attr: contains everything that is needed to create the parallel coordinate plot

- somatic.gui: contains everything that has to do with the GUI: all the applets, the color scheme and the PerspectiveHandler
- somatic.map: contains everything that is needed to create the geographic map: the HTML and KML parser, the polygon and 2D vector classes
- somatic.rainer: contains everything that my two colleagues created
- somatic.som: contains the class that puts all the parts of somatic.rainer together and creates the SOM

4.2 Tri-space Converter Library

This library helps the user converting CSV files from one tri-space perspective into another. It can also be used to transform files into the DAT format which is needed if you use SOM_PAK for creating SOMs.

5. Discussion

Here the results are reflected, it is reviewed if the used methods are appropriate for the requirements and if those requirements have been reached.

5.1 Critical reflection

The SOMatic Application is a huge step towards one application that can help you understand your data better and combines different ways of representation. By now it was not possible to perform a user evaluation, that way, not so obvious bugs and possible misunderstandings can be eradicated. The important point of the ability to highlight one input vector in all visualizations is not yet implemented. Another point that is not yet completely satisfying is the parallel coordinate plot; it would increase the chance to find underlying patterns a lot if it would provide the possibility to change the order of the dimensions. To be helpful for as many users as possible, it should be able to work with other data as well. In addition to that the users either have to upload their own geographic map file, which could be hard to handle or there should be a map representation to which the CSV files can be linked via country names or coordinates. Another point that is not completely satisfying by now is the possibility to zoom in the three applets. The zooming would enlarge the information gain for the user.

5.2 Are the applied methods appropriate?

For the SOM and the graph the implemented programs are definitely appropriate. In the short time span it was not possible to implement an appropriate method for the geographic map. Right now it is not able to zoom and the user is not able to choose a country by clicking on it to highlight the related data in the other applets. A more appropriate method would include that, either self-coded or not. It makes sense to wait for the Unfolding Maps library[14] to work with the processing 2.0 environment; that will save a lot of time and work and will create a lot more possibilities. The recent library for processing 1.5 is not only able to create a map and an overview map and markers, but it can also color countries and read GeoJSON[15] files.

5.3 Have the expected results and goals of the thesis been reached?

The goal to use the processing library to create a web-deployable tool to give users the chance to gain information about temporal and spatial patterns of socio-economic indicators of Africa was reached. There are some possibilities to improve that tool, but the basic requirements were reached. The application is able to show the data in different views and to switch between three tri-space perspectives. The three applets share the same color scheme and there is a control panel which changes depending on the chosen perspective.

6. Conclusions and future work

This chapter gives a short summary and presents the future work to improve the created application.

6.1 Summary

The requirements were fulfilled and the user gets more information about the given data set than before. There are some open issues which would improve the product a lot. Still, based on this application some meaningful conclusions about the data were made. The application can not only help non-expert users but also people who have been working with SOMs for a long time. Without any clustering methods this tool is able to show basic correlations and differences.

---

14 http://unfoldingmaps.org/ (accessed: 12 June 2013)

15 http://www.geojson.org/ (accessed: 12 June 2013)

6.2 Future work

- Implement the possibility to highlight the same input vector in all perspectives to enlarge the information gain for the user. To be able to do that a maps library for the geographic map would help a lot because implementing a point in polygon test just for that is too time consuming.
- Creating a platform for users to upload their own data: Therefore a basic website linked to the application has to be created where users can not only upload their data but also find example data and templates to know what the data has to look like to be successfully uploaded.
- Giving the user a chance to define a new color scheme for the SOM: That can be done with a simple additional editing window. As long as the new color scheme still has twelve colors it makes no difference for the application.
- Including a real map instead of the KML file: For this goal it would be good to wait until the Unfolding Maps library is available for processing 2.0. Then the main work would be to give the users a chance to upload their geographic data in the same way as the SOM data.
- Component planes: Providing an additional view that shows the different component planes would supply even more information about the data, especially for expert users.
- Trajectories: This special way of presenting SOM data in a temporal way is another way of giving more information to the (expert-) user.

References

Andrienko, N. and Andrienko, G., 2006. *Exploratory Analysis of Spatial and Temporal Data - A Systematic Approach,* Springer

Gamma, E. and Helm, R. and Johnson R. and Vlissides, J., 1994. *Design Patterns – Elements of Reusable Object-Oriented Software*, Amsterdam, Addison-Wesley Longman

Inselberg, A. and Dimsdale, B., 1991. Parallel Coordinates. In Human-Machine Interactive Systems (pp. 199-233). Springer.

Inselberg, A., 1999. Multidimensional Detective, *In*: S. K. Card and J. D. Mackinley and B. Schneiderman, eds. *Readings in Information Visualization Using Vision to think*, San Francisco, CA: Academic Press, 107-110

Kohonen, T. and Hynninen, J. and Kangas, J. and Laaksonen, J., 1995. Self-Organizing Maps Program Package [online], Available from: http://www.isegi.unl.pt/ensino/docentes/fbacao/som_pak.pdf [Accessed 15 May 2013]

Kohonen, T., 2001. *Self-Organizing Maps*, Springer

Kohonen, T. and Honkela, T., 2007. *Kohonen network* [online]. Scholarpedia, Available from: http://www.scholarpedia.org/article/Kohonen_network [Accessed 8 March 2013].

Kruskal, J. B. and Wish, M., 1978. *Multidimensional Scaling*. Sage University Paper Series on Quantitative Applications in the Social Sciences 07-001. Beverly Hills, Calif.

Loy, M. and Eckstein, R. and Wood, D., 2002. *Java Swing*, O'Reilly Media

Skupin, A. and Fabrikant, S., 2003. Spatialization Methods: A Cartographic Research Agenda for Non-Geographic Information Visualization. *Cartography and Geographic Information Science*, 30(2), 99-119

Skupin, A. and Esperbé, A., 2011. An alternative map of the United States based on an n-dimensional model of geographic space*. Journal of Visual Languages and Computing*, 22(4), 290-304

Spence, R., 2007. *Information Visualization Design for Interaction*, Pearson

Wang, N. and Biggs, T. and Skupin, A., 2013. Visualizing gridded time series data with self organizing maps: An application to multi-year snow dynamics in the Northern Hemisphere. *Computers, Environment and Urban Systems*, 39(5), 107-120

Yan, J. and Thill, J.-C., 2008, Visual Exploration of Spatial Interaction Data with Self-Organizing Maps In: Agarwal, P. and Skupin, A., eds. *Self-Organising Maps: Applications in Geographic Information Science*, Wiley, 67-83

List of Figures

List of Tables