
DIPLOMARBEIT

# Analyzing, Labeling and Interacting with SOMs for Knowledge Management

ausgeführt am Institut für
Softwaretechnik und interaktive Systeme
der Technischen Universität Wien

unter der Anleitung von
ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber

durch

**Angela Roiger**
Matrikelnummer: 9825108

Ferd.-Raimund-Str. 21
3032 Eichgraben

Wien, im März 2007

## Abstract

Self-Organizing Maps are a popular way to display complex relations of high dimensional data. This work improves the map visualization by revealing the cluster structures in a Self-Organizing Map: First, different algorithms are implemented to determine the cluster structure. Afterwards, methods are derived to display and suitably label these clusters on the Self-Organizing Map in a way that is intuitive and easy-to-understand for the user. A data set of 20,000 newsgroup postings was used for experimental verification.



## Zusammenfassung

Self-Organizing Maps erfreuen sich großer Beliebtheit, wenn es darum geht, komplexe Zusammenhänge hochdimensionaler Daten vereinfacht darzustellen. Um diese Karten für einen Anwender leichter lesbar zu machen wurden im Rahmen dieser Arbeit Algorithmen implementiert die Clusterstrukturen in einer Self-Organizing Map aufzeigen. Das Wissen über die Struktur alleine ist aber noch nicht ausreichend; diese muss auch verständlich und übersichtlich dargestellt werden. Dazu wurden Methoden entwickelt die Cluster darzustellen und sie mit geeigneten Beschriftungen zu versehen. Die Anwendbarkeit wurde anhand einer Datenbasis von 20.000 Newsgroup Beiträgen untersucht.

# Contents

# Chapter 1

# Introduction

Every day we are confronted with vast amounts of available data, ranging from the output of large-scale sensor networks to document archives on the internet. Company databases also fit into this category, be it customer sales data or documents produced by different departments. Nowadays, various knowledge management techniques are used to capture, organize and structure this information. Representing these large amounts of data in a way suitable for human interaction is a crucial task in this process.

For text documents a classical method is searching by keywords. Usually several keywords can be combined to form one search query as for example in internet search-engines. The results of such queries are typically presented to the user as a list. The documents get ordered by some sort of relevance, for example based on where the keywords appear in the documents. This method is applicable only if the user is familiar with the information sought-after. Another common method is to structure the data in hierarchies, letting the user browse through them. Just as in libraries or book stores one can narrow down the category step by step until the relevant information is found. This way the user can explore the documents but it is difficult to get an overview. Furthermore possible interconnections between various branches in the hierarchy cannot be represented.

A different approach is to use artificial neural networks to organize data. Particularly suitable for user interaction is the Self-Organizing Map (SOM) as it allows to display large amounts of multi-dimensional input data on a two dimensional lattice. The map itself is only a grid on which similar data lies spatially close but enhanced with some visualizations it can give a quick

overview of the structure of the data. Zooming and different layers of displayed information can aid the interactive exploration of the map.

The goal is making the SOM readable for a user as intuitively as a cartographic map. Some features of cartographic maps can be applied to Self-Organizing Maps to make them more easily understandable, for example:

- Relief depiction: A SOM itself does not have any relief information, but visualizations can be applied which generate a relief. A simple example would be to display areas of high data density as an area of high elevation on the map.
- Borders: There are several ways how to split up the SOM into clusters. These clusters can be drawn just like country borders on a cartographic map. If there is more than one clustering level to display, this can be done analog to country, state and district borders.
- Labels: On a cartographic map labels are used to describe for example countries, districts, cities or areas like mountain ranges all on the same map. To recognize what object or area on the map a label belongs to different font sizes and styles are used. The same can be applied to a SOM when displaying several levels of information.

The remainder of this thesis is organized as follows. Chapter 2 explains the concepts and methods on which the remainder of the thesis is built on. Section 2.1 outlines how a Self-Organizing Map works, Section 2.2 deals with labeling of maps in general and in particular the labeling of Self-Organizing Maps and Section 2.3 presents some clustering algorithms. Finally Section 2.4 introduces the software which is used in this thesis - the SOMToolbox.

Chapters 3 and 4 describe the features newly implemented into the SOMToolbox. The implementation is an integral part of this thesis. Besides the description also some design decisions are discussed. The implementation consists of two parts which are explained in detail: Chapter 3 describes the clustering of the SOM into hierarchical clusters. For this three different agglomerative algorithms are implemented. Furthermore in Section 3.2 new color palettes are included in the program to try to make the map look like a topological map with mountains and valleys. The second task described in Chapter 4 is connected to the first as now labels for the clusters are created and placed on the map.

Some experiments are shown in Chapter 5, applying the clustering and labeling algorithm to maps created from 20.000 newsgroup postings.

Chapter 6 provides a summarization and draws conclusions.

# Chapter 2

# Related Work

The work presented in this thesis implements theoretical results about clustering of self-organizing maps by extending an existing software framework and analyzes their applicability in different scenarios. Thus, the preliminary knowledge given in this chapter, covering various topics around Self-Organizing Maps, map labeling and clustering, will provide the foundation for the main contributions of this thesis.

The chapter is structured as follows: First the Self Organizing Map is explained followed by two approaches to generate labels for Self Organizing Maps and a brief general introduction to map labeling. Afterwards an introduction to clustering is given including some clustering algorithms in detail and distance metrics that can be used. The last part of this chapter introduces the "SOMToolbox" program which has been used and extended in this thesis.

## 2.1 Self Organizing Maps

The Self-Organizing Map (SOM; also "Self-Organizing Feature Map" or "Kohonen Map") is a very popular artificial neural network algorithm for unsupervised learning. It has been developed by Teuvo Kohonen and was first described in [Koh82] and discussed in detail e.g in [Koh01]. The SOM projects high-dimensional input data on a two dimensional map. The algorithm is topology preserving, which means that

- similar input data will be mapped to spatially close areas on the map,
- elements which are spatially close on the map should have similar input data.

Furthermore high density areas in input space are mapped onto correspondingly many units on the map.

Figure 2.1 shows that the data points of the dense area in a high dimensional input space $V$ are mapped to several close-by units on the SOM and the other data points are mapped to further away units, corresponding to the distance they have in the input space.

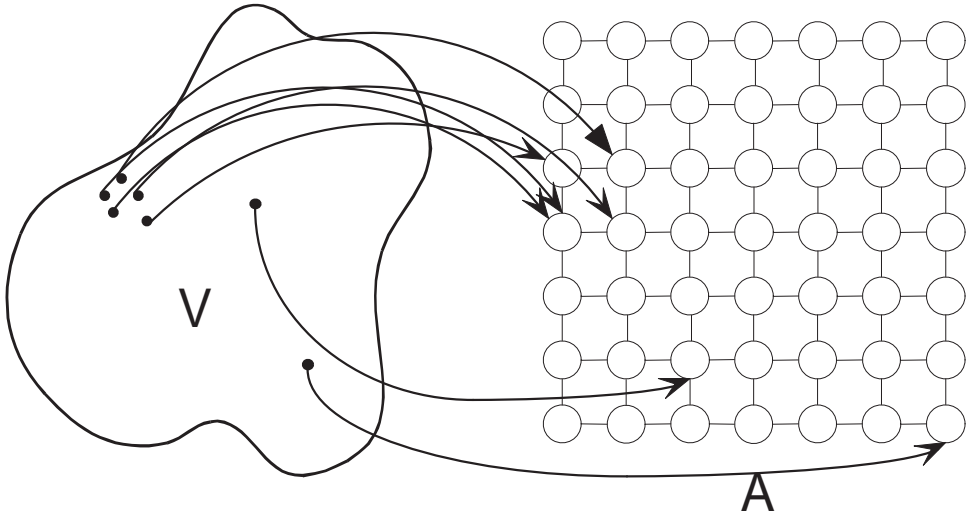



Figure 2.1: Mapping Input Data Onto the Map

SOMs have been used in various fields of science - classification of pictures or music[MLR06][DNR05], organization of document collections[RM03] and controlling robots[HSvdSS94][KMD91] to name just a few.

### 2.1.1 Architecture

The SOM consists of a two-dimensional grid where each node in the grid holds one weight vector (also "model vector", "prototype vector" or "codebook vector"). The weight vector $m_i = [m_{i1}, m_{i2}, \ldots, m_{in}] \in \Re^n$ has the same dimension as the input vectors $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}] \in \Re^n$ which describe $n$ features in the input space. The input vectors are assigned to nodes during the training process. The lattice of the nodes is usually hexagonal or rectangular but can be of any shape. In the SOM implementation used in this thesis a rectangular lattice is used.

### 2.1.2 Training Algorithm

Before the training the map needs to be initialized and then the training process of a SOM consists of a few basic steps which are repeated several times.

1. randomly select an input vector
2. search for the best matching unit
3. adapt weight vector
4. modification of learning rate and neighborhood range

This is done either a predefined number of iterations or until a stopping criterion is met.

#### Initialization

The initial weight vectors can be created for example by using one of the following methods:

- Random Initialization: Each node can be initialized with a random weight vector. It has been shown that initially unordered vectors will be ordered usually during a few hundred initial steps. This does not mean that this is the fastest or best solution. [Koh01]
- Linear Initialization: To start with an ordered initial state of the map one can place the vectors along the x- and y- axis ascending or descending.
- Random Samples of the input dataset: This method is similar to the random initialization, but the vectors are taken randomly from the input data.

Once the initialization is done, the actual training process can be started.

#### Search for Best Matching Unit

In the beginning of each training iteration one vector gets selected randomly from the input data set.

The chosen input vector is compared to each unit's weight vector and the input vector is assigned to the unit which has the smallest distance between the input vector and its weight vector. For a description of distance measures see Section 2.3.7. Very often the Euclidean Metric is used to define the distance between two vectors. The chosen unit is called the "winner".

### Weight Vector Adaptation

To improve the SOM, the winner unit's weight vector is adapted to make it more similar to the input vector. Geometrically the weight vector is moved closer to the input vector. Also the neighboring unit's vectors are adapted. The neighborhood is defined by a function which can typically be a Gaussian bell-shaped curve or a function defining a radius around the winner.

In Figure 2.2 the winner unit is black and the units around are shaded in different levels of gray according to how much they are adjusted. In (a) the neighboring units are all adapted to the same degree and in (b) the units are adapted depending on the distance to the winner unit.

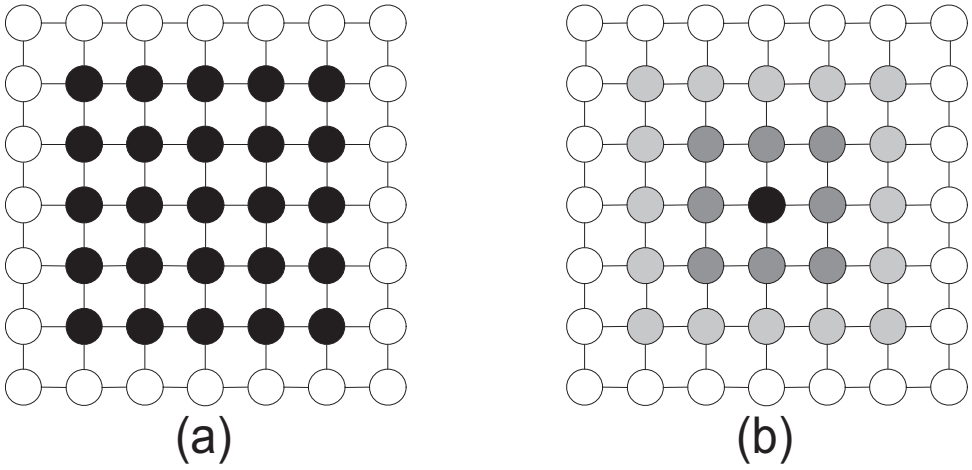



Figure 2.2: Different Neighborhood Functions

The new value of the weight vector $m_i$ can be calculated for the winner unit according to

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [x(t) - m_i(t)] \tag{2.1}$$

where

- $i$ is the current unit
- $c$ is the winner unit
- $x$ is the input vector
- $t$ is the iteration step
- $\alpha$ is the time decreasing learning rate

- $h_{ci}$ is the time decreasing neighborhood function

This function is also applied to all neighboring weight vectors.

#### Modification of Learning Rate and Neighborhood Range

In the training process the map should adjust itself the most to the input data in the beginning to create the approximate structure of the map. Later on it should adapt less to find a stable arrangement of the input vectors on the map.

In order that the training algorithm converges the learning rate is larger in the beginning and and decreases over time. This means that in the beginning the weight vectors are altered a lot and in the end of the training process, when a stable organization of the input vectors within the map is reached, there are only small adaptations made. The neighborhood range can also be decreased during the iterations so that for example in the beginning all units' weight vectors are adapted and in the end only close neighbors are affected.

#### Stopping Criteria

The easiest way to determine the end of the training process is to run it only for a predefined number of iterations. One can also define a mathematical quality criterion and run training until this condition is fulfilled. Another option is to terminate the training process if a stable organization of the input vectors on the lattice is reached.

### 2.1.3 SOMs in Knowledge Management

A common problem in knowledge management is the ordering of textual document collections, enabling the user to easily find information of interest. A well-known technique for this type of information retrieval is to show textual documents in the context of *similar* documents [Hon99], i.e. to group together related topics or concepts of a data collection. It has been shown that suitably-created SOMs satisfy this so-called "proximity condition" [LCN99], and, thus, provide a useful knowledge management tool. Examples for this effect can also be seen in Chapter 5 of this work.

Once this proximity condition is satisfied, a SOM can be used to exploratively search for related documents. Given a suitable visualization, this should be doable interactively by the

user without the need to reformulate search expressions, which would usually be required by traditional information retrieval techniques. [KHLK96] analyzes the advantages of the SOM approach with regard to digital libraries in further detail.

## 2.2 Map Labeling

To be able to interpret a trained map some sort of labeling is necessary. Manual labeling of a map may be sufficient for small maps where the user has knowledge of the data represented but becomes unfeasible with increasing map size. Sometimes class information is available for the data - in these cases the class labels can be used to describe the map.

### 2.2.1 LabelSOM

One method to automatically determine a label for a unit is the LabelSOM method explained in detail in [Rau99] and [RM99]. It describes the units of a SOM with the vector elements (=features of the input space) that best describe the input vectors mapped to this unit. This is done by looking at the quantization error of the vector elements.

The quantization error is the average distance for a feature between the unit's weight vector $m_i$ and all the input vectors $x_j \in C_i$ where $C_i$ is the set of all input vectors mapped to the unit $i$.

$$q_{i_k} = \sum_{x_j \in C_i} \sqrt{(m_{i_k} - x_{j_k})^2} \qquad m_i, x_j \in \Re^n \qquad k = 1 \ldots n \tag{2.2}$$

This means that a low quantization error characterizes a feature that is similar in all input vectors to the weight vector. Thus this feature describes the unit well.

If the input vector contains a lot of attributes which are non existent and therefore 0 those attributes often have a quantization error of almost 0 for a unit. Such features are not appropriate for labeling the unit though, since this would describe what the unit does not contain. Therefore the vector elements need to be determined which have about the same value in the input vectors and have value above a defined threshold.

### 2.2.2 Keyword Selection by Lagus and Kaski

In [LK99] a method for labeling text document maps is presented which strives to find labels that distinguish an area from the rest of the data. The method can be applied to label units as well as clusters. A good descriptor or keyword $w$ for a unit or cluster $C$ should have the following two properties:

1. $w$ should be prominent in $C$ compared to other words in $C$
2. $w$ should be prominent in $C$ compared to the occurrence of w in the whole collection.

Since neighboring units usually describe similar data, the frequent appearance of the word $w$ in close by areas does not make it a bad keyword. Therefore the area immediately surrounding the cluster is excluded in determining the goodness for the keywords of that cluster.

### 2.2.3 Cartographic Map Labeling

The problem of label placement is known in cartography as the map-labeling problem. There are many approaches to automating the process for example with expert systems or simulated annealing. Map labeling can be split up in two problems[WWKS01]:

1. The Label-Size Maximization Problem: Find the maximum factor $\sigma$ such that each feature gets a label stretched by this factor and no two labels overlap. Compute the corresponding complete label placement.
2. The Label-Number Maximization Problem: Find a maximum subset of the features, and for each of these features a label from its set of candidates, such that no two labels overlap.

Both problems are NP-hard [FW91].

The labeling problem arising in this thesis is somewhat different. In contrast to the problems mentioned not a number of points but map areas need to be labeled. Deciding the size for the labels is also an issue because the map size and the number of labels to display can vary a lot.

In [Sku04] commercial geographic information systems software was used to display a self organizing map of text documents like a geographical map.

## 2.3 Clustering

Clustering can be described as finding a natural grouping among objects. The members of a cluster are similar in some way and are dissimilar to members of other clusters. The clustering algorithm has to find a structure in an unlabeled data collection to classify them without supervision.

In [Jai99] clustering is defined as follows: "Clustering is the unsupervised classification of patterns (observations, data items or feature vectors) into groups (clusters)"

There are two ways to cluster: partitional and hierarchical.

- Partitional clustering determines all clusters at once and the clustering gets improved iteratively. A data item can change the cluster it belongs to during this process.
- Hierarchical clustering determines clusters based on already existing clusters. The hierarchy implies that once a data item belongs to a cluster this decision cannot be reversed. It can be done either divisively (top down) or agglomeratively (bottom-up).
  - Divisive algorithms start with one cluster containing the whole set and subsequently divide the cluster into successively smaller clusters.
  - In contrast, agglomerative algorithms begin with each data element as a single cluster and then those clusters are merged into successively larger clusters.

### 2.3.1 Agglomerative Hierarchical Clustering

All methods implemented in this work use agglomerative hierarchical clustering. In each step the two most similar clusters get merged into one new cluster. The similarity of the clusters can be determined using various linkage functions. In the end, there is one big cluster containing a hierarchy of all other clusters down to the level where each cluster contains only one element. The result of this clustering can be visualized in a dendogram. A dendogram is a tree structure where each clustering step is visualized as fusion of two branches. The length of the branches shows the distance between two clusters.

Figure 2.3 shows an example what clustering looks like for two dimensional data points on a plane taking Euclidean distance between the two closest elements of each cluster as the similarity measure. In the beginning, each data item is in a separate cluster. Then the closest

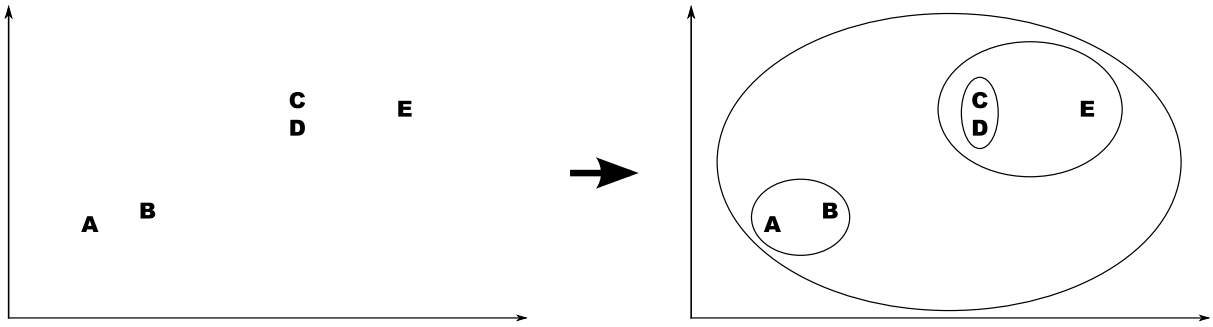



Figure 2.3: Clustering Example

clusters are determined and subsequently joined into bigger clusters. In this example, first the data items C and D are merged into a cluster, then A and B. In the next step, the cluster of C and D is merged with E and in the last step the two remaining clusters A,B and C,D,E are merged.

Figure 2.4 shows the dendogram for this example.

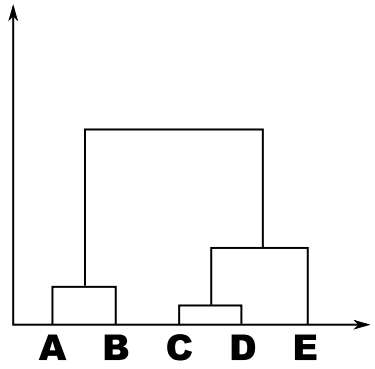



Figure 2.4: Dendogram Example

To determine how close clusters are to each other several linkage functions can be used to calculate the cluster-to-cluster differences. There are some common linkage functions:

### 2.3.2 Single Linkage

This linkage function is also known as nearest neighbor linkage. The distance between two clusters is defined by the distance of the two closest elements in the cluster.

Using single linkage a phenomenon called chaining can be observed. Since the distance between two clusters is only calculated between the two closest elements, it can happen that

"chaining objects" connect clusters that have a big distance. In the example shown in Figure 2.5 the object D is the "chaining object" which connects the two clusters C and E. The shortest distance is represented by the black line, the other distances by a dotted line.

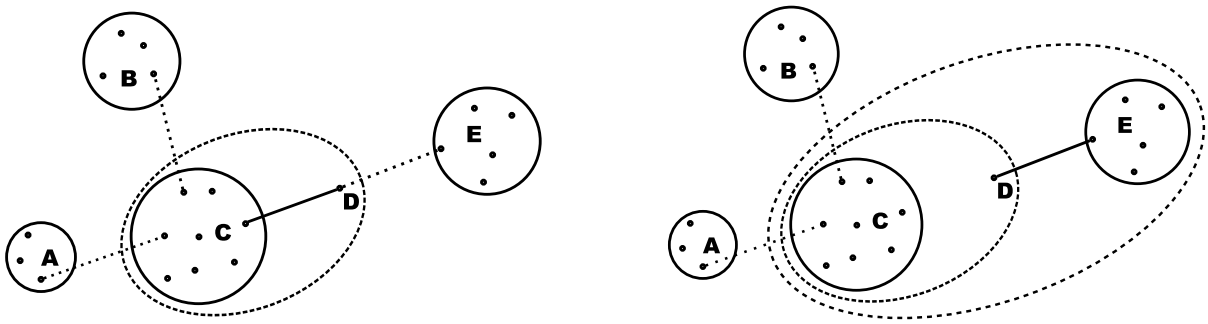



Figure 2.5: Single Linkage with Chaining

In case the areas between the clusters do not contain any chaining objects the single linkage clustering algorithm is able to find even very heterogeneous clusters, e.g. long, u- or s-shaped clusters. [Eck80]

This method tends to create a few big clusters and isolate outlier elements which are only joined in the last steps. The strong point about this characteristic is, that such outliers can easily be found in the dendogram. Because of this property single linkage is often merely used to identify outlier elements.

### 2.3.3 Complete Linkage

The cluster-to-cluster distance is the maximum distance between any one node from one and any node from the other cluster. Therefore it is also called furthest neighbor linkage.

Figure 2.6 shows how the complete linkage method would cluster the same data as in the example above. Since only the longest distances between two clusters are taken into account, chaining is not possible. In the first step D and E are the closest clusters, so they are connected. Cluster D,E now has a big distance to the other clusters and the clusters A and C are merged.

This method tends to create small clusters. All pairs of objects of the two clusters have to be considered to connect two clusters so merging of big classes is only possible in the last steps. Outliers cannot be detected using complete linkage method - they rather distort the result of the clustering.

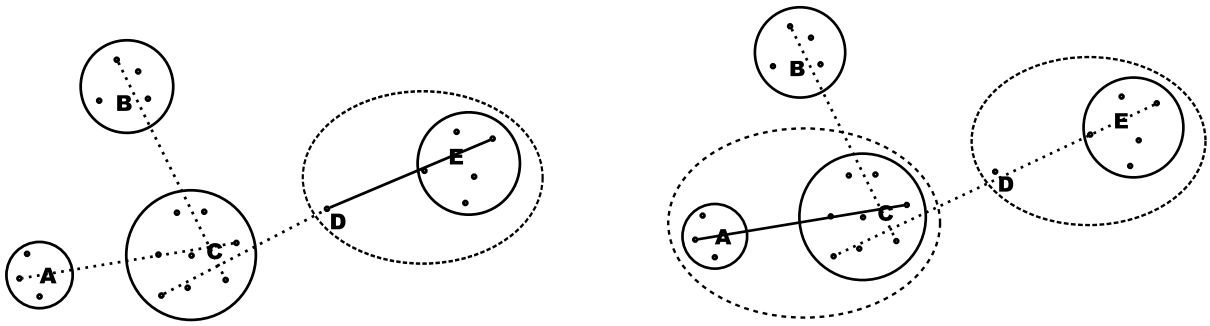



Figure 2.6: Complete Linkage

Single and Complete Linkage have in common that the merging of two clusters can depend upon one distance value.

The remaining methods do not have such anomalies as chaining or creation of small clusters. Furthermore it is not possible to identify outlier objects with these methods.

### 2.3.4 Wards Linkage

This linkage function, first presented in [War63], is also known as minimum-variance clustering because it strives to keep the variance of the cluster elements as low as possible. The distance is determined by the increase in the "error sum of squares" (ESS) if the two clusters are combined.

The ESS of a set $X$ of $N_X$ values is the sum of squares of the deviations from the mean value or the mean vector (centroid). For a set $X$ the ESS is described by the following expression:

$$ESS(X) = \sum_{i=1}^{N_X} |x_i - \frac{1}{N_X} \sum_{j-1}^{N_X} x_j|^2 \tag{2.3}$$

where $|.|$ is the absolute value of a scalar value or the norm (the “length”) of a vector.

The distance between two clusters $X$ and $Y$ is described by the following expression

$$D(X,Y) = ESS(XY) - [ESS(X) + ESS(Y)] \tag{2.4}$$

where $XY$ is the combined cluster resulting from fusing clusters $X$ and $Y$. [sta]

If there are two pairs of clusters which have an equal distance between their centroids, Ward’s method rather joins the two clusters with more different size or if they have equal sizes

the two smaller clusters.

The Ward's linkage function tends to create clusters of equal size.

### 2.3.5 Average Linkage

The distance between two clusters is defined as the average distance between objects from the first cluster and objects from the second cluster.

### 2.3.6 Average Group Linkage

For this linkage method the average values (the mean vectors or centroids) of the clusters have to be calculated. The cluster-to-cluster distance is then defined as the distance between the clusters' average values.

For calculating those linkage functions different distance metrics can be used. The metric is the function which defines the distance between two elements.

### 2.3.7 Distance Measures

To measure the distance between two objects we have to define a distance function. A distance function is a function $d$ which defines a number for each pair of objects $o_i$ and $o_j$ such that:

$$d_{ij} \geq 0 \qquad \text{positivity} \tag{2.5}$$

$$d_{ij} = 0 \text{ if, and only if } o_i = o_j \qquad \text{identity} \tag{2.6}$$

$$d_{ij} = d_{ji} \qquad \text{symmetry} \tag{2.7}$$

If it also meets the following criterion

$$d_{ij} \leq d_{ik} + d_{jk} \qquad \text{triangle inequality} \tag{2.8}$$

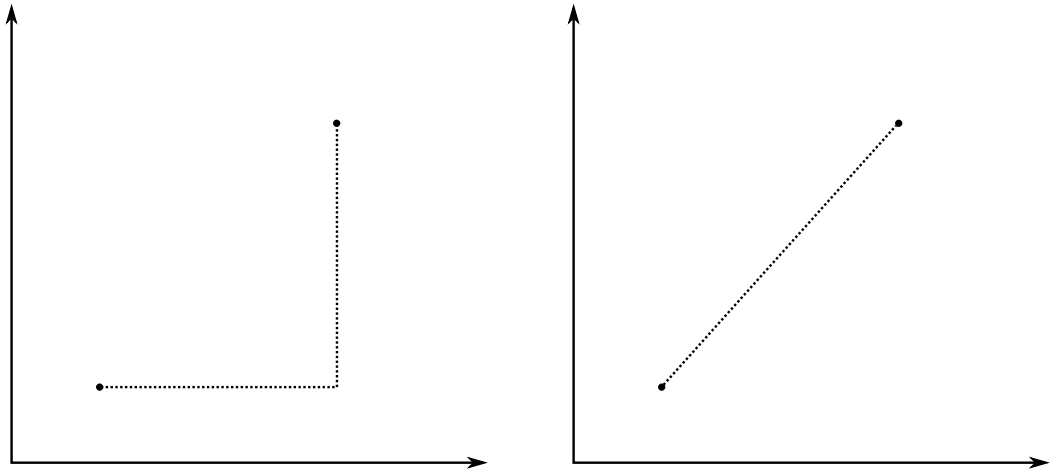
it can be called a distance metric.[Eck80]

### $L_1$ or City Block Metric

The City Block Metric (also Manhattan Metric) is the sum of all distances on each variable. It is defined as follows:

$$d_{ij} = \sum_{l=1}^{m} |x_{il} - x_{jl}| \tag{2.9}$$

where $x_{il}$ stands for the $l$-th variable of the $i$-th object.



(a) City Block Metric (b) Euclidean Metric

Figure 2.7: City Block Metric and Euclidean Metric

The name Manhattan Metric originates in the similarity to walking from one point to another in Manhattan where each distance component can be seen as the number of blocks in each direction.

Figure 2.7 illustrates how the distance is defined in the City Block Metric compared to the Euclidean Metric.

### $L_2$ or Euclidean Metric

The most commonly used metric is the Euclidean Metric. Geometrically it is the direct line between the two points.

$$d_{ij} = \sqrt{\sum_{l=1}^{m} (x_{il} - x_{jl})^2} \tag{2.10}$$

#### $L_r$ or Minkowski-Metric

A generalization of the Metrics mentioned before is the Minkowski Metric.

$$d_{ij(r)} = (\sum_{l=1}^{m}(x_{il} - x_{jl})^r)^{1/r} \quad r \geq 1 \tag{2.11}$$

$r = 1$ and $r = 2$ correspond to the City Block and the Euclidean Metric, respectively.

## 2.4 SOMToolbox

The SOMToolbox is a program developed in Java for working with SOMs which is permanently extended by new features. It supports SOMs with a rectangular lattice. An overview of this program is given here to outline what was implemented before.

### 2.4.1 Features

Only a limited selection of the available features will be described briefly here as an appropriate discussion of every functionality would go beyond the scope of this thesis.

The program supports interactive exploration of a SOM including zooming, selecting units and viewing the data mapped to the SOM if there are viewable files associated with the data items on the map.

In [NDR05] and [NLR05] the interface for exploring music collections are described in detail. The graphical representation of a music collection can be used for example to browse music archives or to create playlists.

#### Training

First of all, with this software a SOM can be trained using several models:

- SOM: the SOM training algorithm as explained in Section 2.1.2
- Growing Grid: a SOM growing dynamically in size during the training process by inserting additional rows or columns into the SOM
- GHSOM (Growing Hierarchical SOM): an incrementally growing SOM consisting of several layers which each consist of a number of independent SOMs

- Mnemonic SOM: a SOM with arbitrary shape

A detailed description and discussion of the algorithms can be found in [May04] and [MMR05].

### Unit Labeling and Zooming

Since a grid of units containing data items is not yet really useful, labels and other additional information can be added for each of these units. Currently only the labelSOM algorithm mentioned in Chapter 2.2.1 is implemented but the program is designed to support several labeling algorithms.

The interface supports semantic zooming, which means that the amount of information displayed depends on the current zooming level. Figure 2.8 shows on the left side a part of a SOM at a low zooming level where the number of data items mapped to the unit is shown as well as the three most dominant labels. The right side of the figure shows a more detailed part of the same SOM where more labels, the number of data items and also the file names are shown. In this case the file names are too long and are only displayed completely if the mouse position is over the truncated file name.

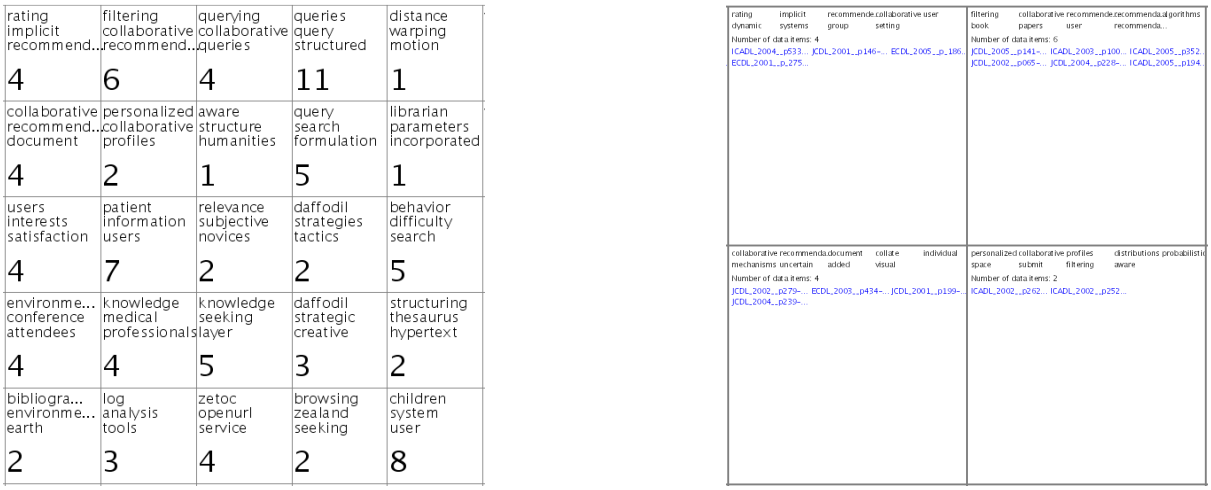



Figure 2.8: Different Levels of Zooming

### Class Information

If class information is available for the data items, it is possible to view where those classes lie on the map. Each class is represented by a color and in each node that contains data items a pie chart is created which shows the distribution of the classes in this node. In Figure 2.9 part of a test dataset is shown which has three classes named “1”, “2” and “3”. From the number of inputs and the colored slices of the pie chart one can determine how many inputs of each class lie in a unit.

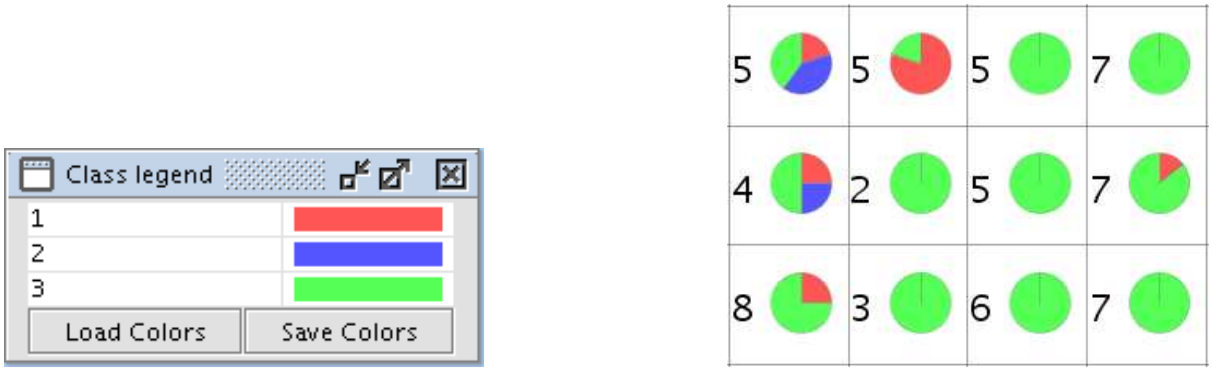



Figure 2.9: Part of a SOM With Pie-charts

## 2.4.2 Visualizations

One of the strengths of SOMs is that it is possible to apply intuitive and meaningful visualizations. There are several visualizations implemented into the SOMToolbox, for example U-Matrix and the Smoothed Data Histograms.

### Iris Dataset

The Iris dataset is used here to show the abilities of the visualizations. It is a widely used and simple data set in data mining literature.

The data describes the three species of Iris flowers: "Setosa", "Virginica" and "Versicolor". Four characteristics of the flowers are represented in the data - "sepal length", "sepal width", "petal length", and "petal width", thus the data vector is four-dimensional. There are 50 data vectors per species in the data set. The Setosa species is distinct from the other two species whereas the Virginica and Versicolor are rather similar.

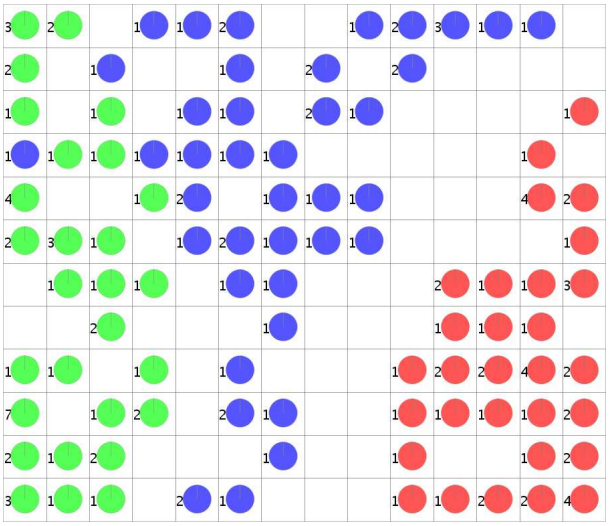


Figure 2.10: Iris Data on the SOM

Figure 2.10 shows the SOM trained with the Iris data set which will be used in this chapter painting the Setosa in red, Virginica in green and Versicolor in blue.

### U-Matrix

U-Matrix stands for "unified distance matrix" and it is a method which visualizes structures of the input space on the SOM by illustrating the distances between the weight vectors of adjacent units.

It calculates a matrix with distances between all neighboring units applying usually the same metric that was used during the training process. An area with low distances in the U-Matrix corresponds to input data lying close to each other. Therefore areas with high values in the U-Matrix can separate sections of similar input data. This characteristic is used to determine cluster boundaries. The distances are mapped to a color palette so that homogeneous clusters are painted in one color and areas of high distance in another. The color gradually changes between those colors to represent the resulting relief of the visualization which can be seen in the example in Figure 2.12.

The distances between the neighboring units are calculated as shown in Figure 2.11 taken from [Ult92]. For each unit three distances are calculated where the distances $dx$ and $dy$ are the distances between $U$ and its neighboring units to the right and to the top and $dxy$ is the mean

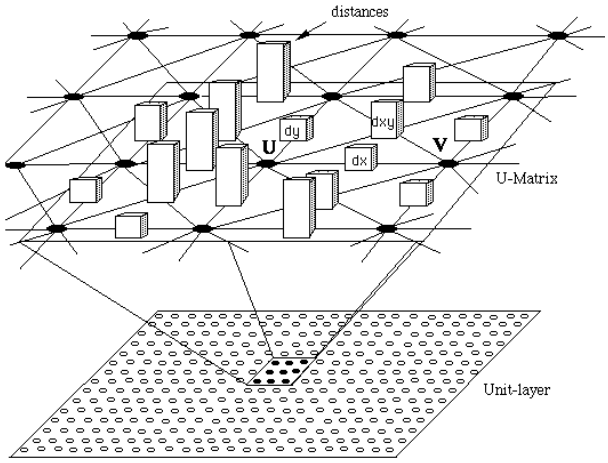



Figure 2.11: U-Matrix Distance Calculation

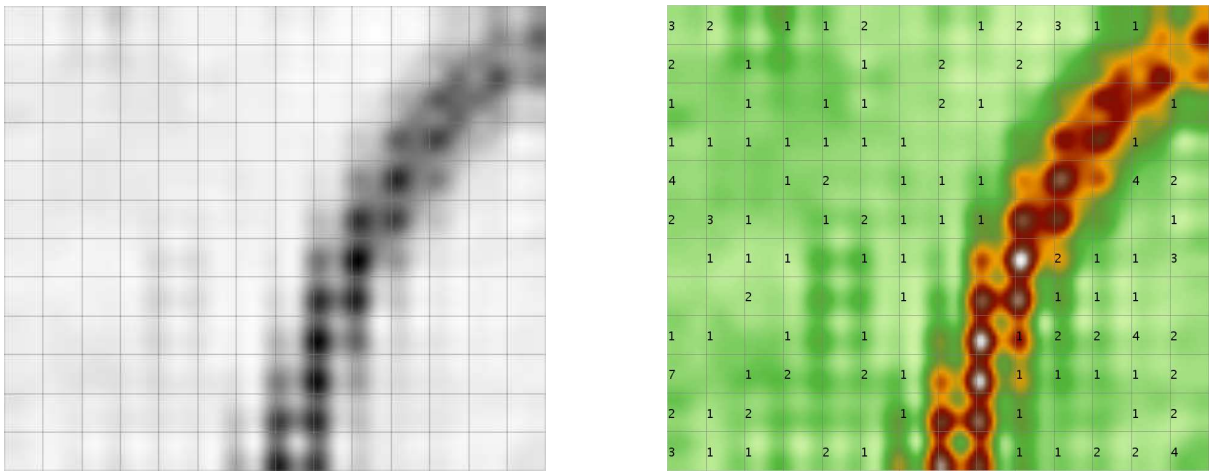


Figure 2.12: U-Matrix in Gray-scale and as Mountains

value of the distance from $U$ to its top right neighbor and the distance between $V$ and its top left neighbor.

Figure 2.12 shows a SOM trained with the Iris dataset to which the U-Matrix visualization has been applied. In the left image the dark areas represent high values in the U-Matrix and therefore a great distance between the units. The right image shows the same visualization but with a color palette imitating cartographic maps. The two clusters can be identified clearly in both images.

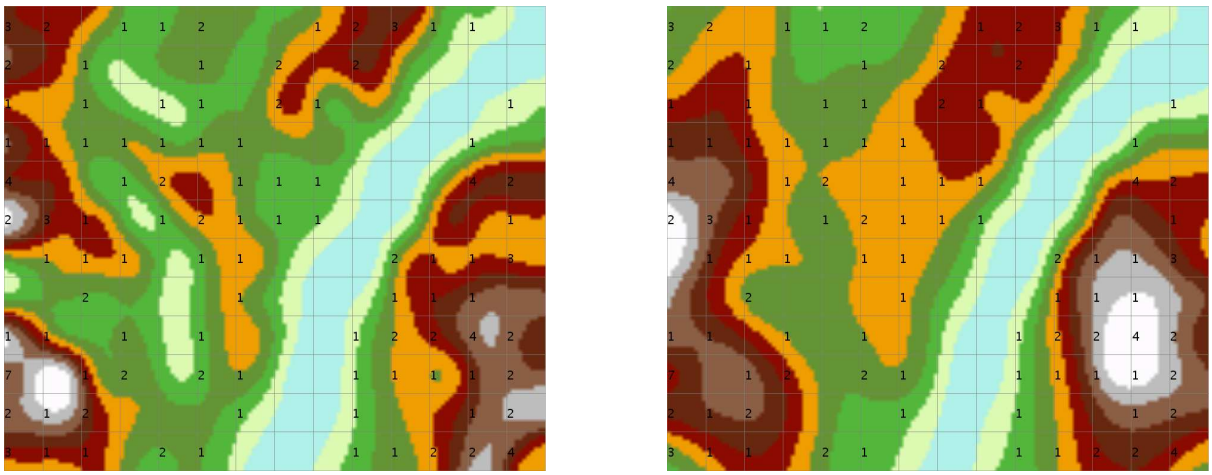


Figure 2.13: SDH With Smoothing Factors 7 and 19

**Smoothed Data Histograms**

The Smoothed Data Histograms (SDH) are a method that strives to unveil cluster structures in a SOM. The idea is to estimate the probability density of the high dimensional input data on the map.

When a SOM is trained the data items are mapped to the unit that best describes the data, which is the one with the lowest distance between the unit's weight vector and the input vector. The SDH algorithm takes the $n$ closest units and assigns points to the units: The best matching unit gets $n$ points, the next best gets $n - 1$ points and so on. The parameter $n$ is adjustable by the user and is called the smoothing factor. All points are summed up and result in a histogram for the map.

Figure 2.13 shows a SDH visualization of the Iris data set. The image on the left uses a smoothing factor of 7 which reveals the separation of the Setosa species from the rest of the data. For the right image a smoothing factor of 19 is used, where the Versicolor and Virginica clusters are somewhat separated. The border of the resulting cluster is not so clear, though.

More details about the SDH and using it for cluster visualization in SOMs can be found in [PRM02]

## 2.5 Summary

This chapter presented an introduction to the fundamental topics that are relevant for the remainder of this thesis. The Self Organizing Map was described briefly as the thesis presents visualization techniques for it. For a more detailed description of the SOM several references to other works were given. Subsequently two algorithms for generating labels for SOM units were presented as well as the problem of labeling cartographic maps. The Label SOM algorithm is of particular interest because it will be applied in Chapter 4 to label clusters. An overview of clustering was given as well as a description of the algorithms that will be used for clustering SOMs. Furthermore an overview of the SOMToolbox was given to present the work that was already done and to explain some features that will be used in the following chapters.

# Chapter 3

# Clustering the SOM

The following three clustering algorithms were chosen and are now implemented into the SOMToolbox to cluster the units on the SOM:

1. single linkage
2. complete linkage
3. Ward's linkage

The linkage functions are described in Section 2.3.1. The program builds up a clustering tree similar to a dendogram but without saving the distance values between the clusters when clusters get merged. Depending on the number of clusters displayed on the map the corresponding borders of the clusters are painted.

This chapter will first compare the results of the clustering algorithms based on some manually generated data sets. Then the possibilities of displaying the clusters will be discussed as well as a new set of color palettes will be introduced.

## 3.1 Comparing the Linkages

To show the characteristics of the available linkage functions some manually generated data sets have been used. They are described here and the linkage functions are applied to them.

The clusters will be visualized by drawing the clusters' borders. To verify the clustering the class information for the data is show in the pie charts.

### 3.1.1 Data Set 1: Two Clearly Separate Clusters

Figure 3.1 shows two data clusters which are completely separated from each other. Each cluster consists of 500 data points with a Gaussian distribution of the points around the cluster center. Those clusters are also clearly separated from each other on the SOM. Therefore, all algorithms should be able to recognize those two clusters.

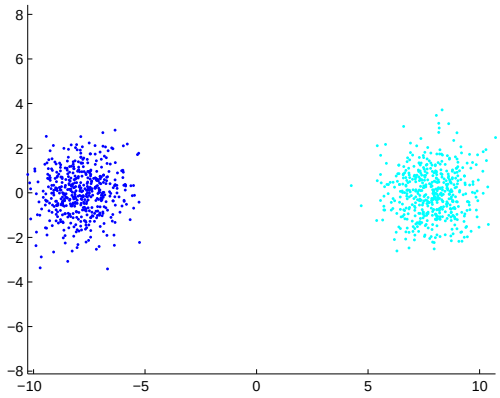



Figure 3.1: Data Set 1: Two Separate Clusters

For this data set all three linkages produce the same result for the clustering in two clusters. This result can be seen in Figure 3.2.

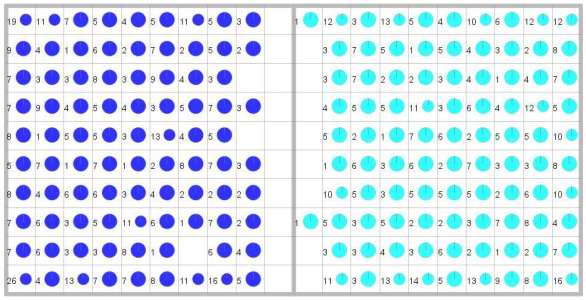


Figure 3.2: SOM of Data Set 1, all Linkages, Clustering Into 2 Clusters

### 3.1.2 Data Set 2: Three Clusters - One Separate

This data set consists of three clusters. One of the three clusters is distinct from the rest, the other two overlap each other as seen in Figure 3.3.

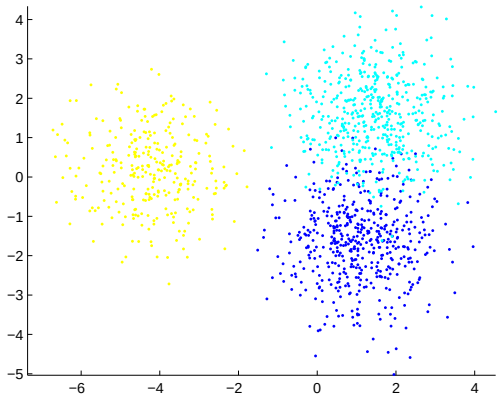



Figure 3.3: Data Set 2: Three Clusters - One Separate

### Single Linkage

Here it is already a bit more difficult to determine the cluster borders as two clusters overlap. The separate yellow cluster is detected by the single linkage function as shown in Figure 3.4 but it cannot find the border between the two other clusters. Figure 3.5(a) shows the clustering into ten clusters where only some single units and two-unit-clusters appear. If clustered even more, for example into 40 clusters as shown in Figure 3.5(b), there are only more small clusters added but the boundary of the two blue clusters is still not detected.

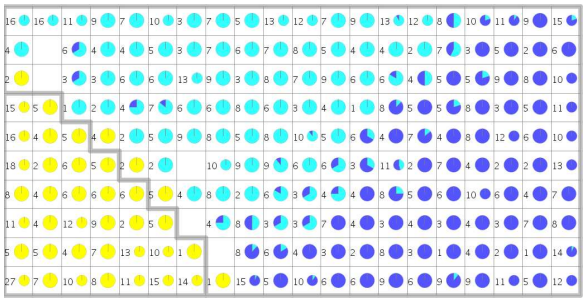
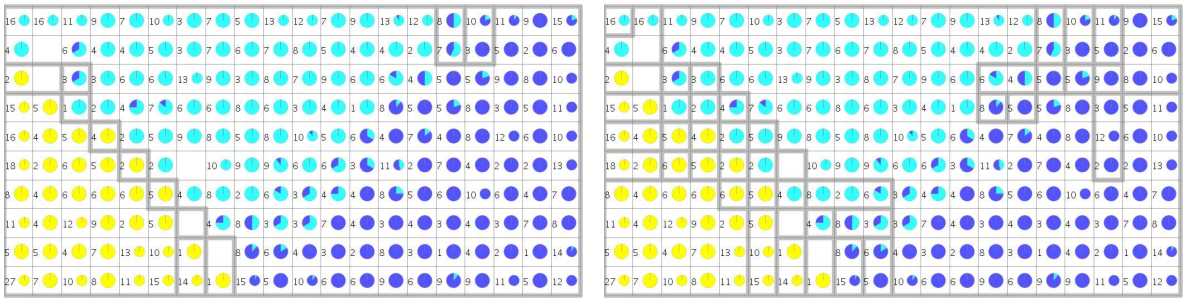


Figure 3.4: SOM of Data Set 2, Single Linkage, Clustering Into 2 Clusters

### Complete Linkage

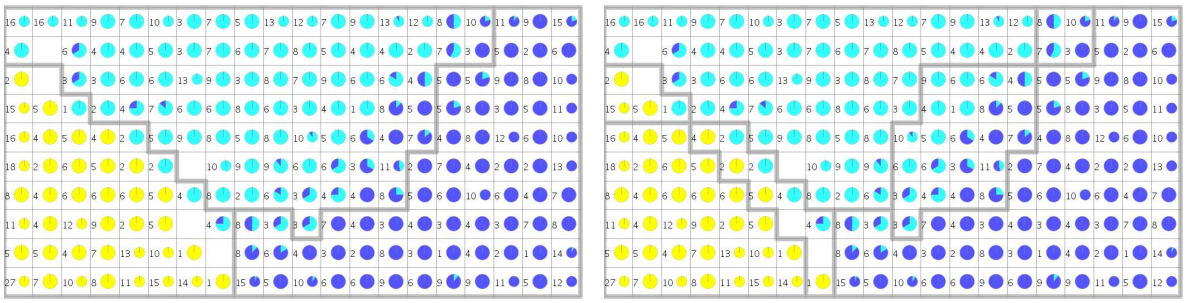
The complete linkage algorithm is able to distinguish the two clusters. Figure 3.6(a) shows that the separation into three clusters yields a very good result. If clustered into five clusters as



(a) Clustering into 10 Clusters (b) Clustering into 40 Clusters

Figure 3.5: SOM of Data Set 2, Single Linkage

shown in Figure 3.6(b) the algorithm separates the border regions between the clusters as own clusters.



(a) Clustering into 3 Clusters (b) Clustering into 5 Clusters

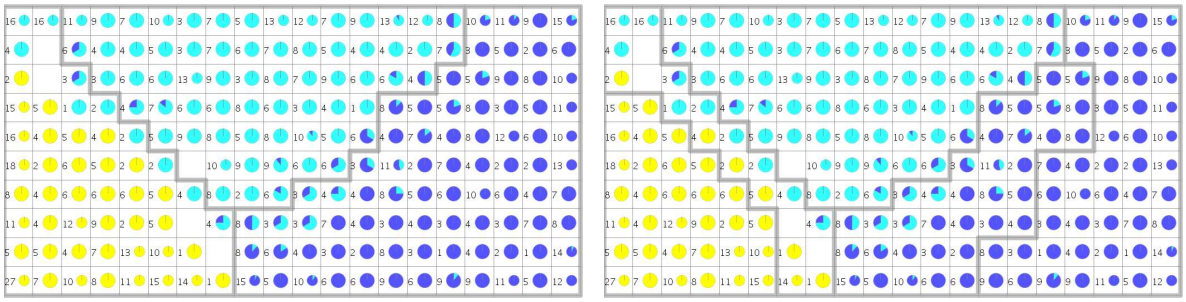
Figure 3.6: SOM of Data Set 2, Complete Linkage

#### Ward's Linkage

With this dataset Ward's linkage behaves very much like complete linkage. As displayed in Figure 3.7(a) the algorithm can distinguish the three clusters. The more sparsely populated areas at the clusters' borders are separated as own clusters when five clusters are viewed as seen in Figure 3.7(b).

### 3.1.3 Data Set 3: Two Rings

The third dataset is more complex and is used to show the strength of the single linkage algorithm. The two clusters each consist of 500 data points which form a ring. The rings are each



(a) Clustering into 3 Clusters (b) Clustering into 5 Clusters

Figure 3.7: SOM of Data Set 2, Ward's Linkage

approximately on a plane in space and have the same radius. The rings are interlocking and each ring passes through the other ring's center.

Figure 3.8 shows a plot of the two rings.

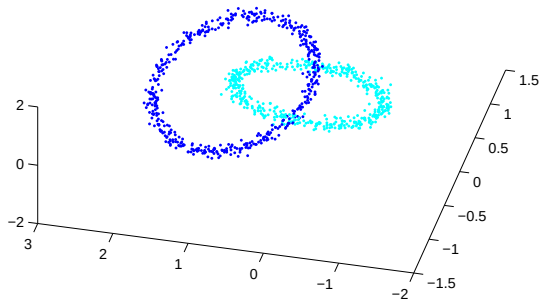


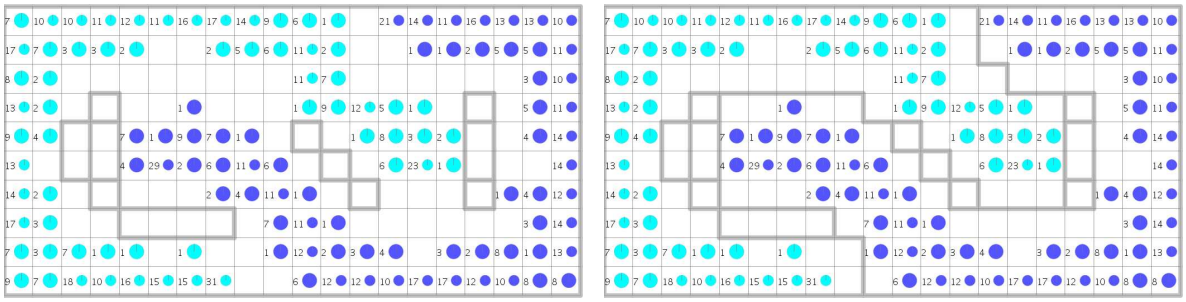

Figure 3.8: Data Set 3: Two rings

The next SOMs show the two ring data set. The data items lie spatially separated on the map in u-shaped clusters. The single linkage algorithm is particularly suited for this kind of data as it clusters two items only depending on their closest distances.

### Single Linkage

Figure 3.9(a) shows the clustering into ten clusters. Those small clusters in the middle are interpolating units which have a high distance to both data clusters. It looks like there are more than ten clusters because some clusters cover units connected only on their corners. If

clustered into eleven clusters as in Figure 3.9(b), the two rings are identified as two separate clusters.

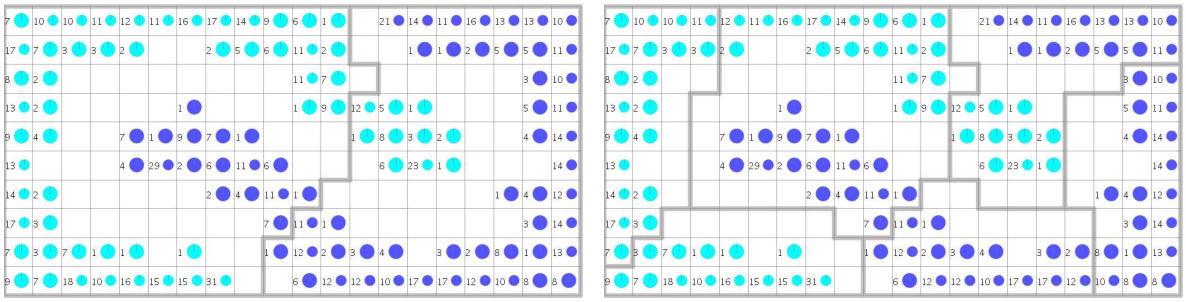


(a) Clustering into 10 Clusters (b) Clustering into 11 Clusters

Figure 3.9: SOM of Data Set 3, Single Linkage

### Complete Linkage

The complete linkage algorithm is not able to determine the two clusters correctly (Figure 3.10). The explanation is quite simple as complete linkage merges the two clusters where the furthest distance between two elements is the lowest. The opposing areas of one ring have a higher distance to each other than each of them has to the other ring.



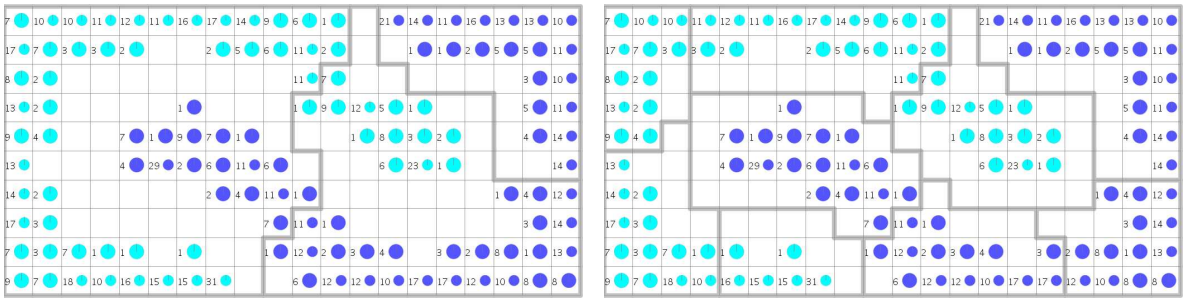
(a) Clustering into 2 Clusters (b) Clustering into 6 Clusters

Figure 3.10: SOM of Data Set 3, Complete Linkage

### Ward's Linkage

Ward's linkage suffers from a similar problem. The variance inside each cluster is lower if the clusters consist of parts of each ring. As an example, Figure 3.11 shows the clustering into three

and into nine clusters. Only when 9 or more clusters are displayed there are just data items of one class in each cluster.



(a) Clustering into 3 Clusters (b) Clustering into 9 Clusters

Figure 3.11: SOM of Date Set 3, Ward's Linkage

### 3.1.4 Data Set 4: Three Overlapping Clusters

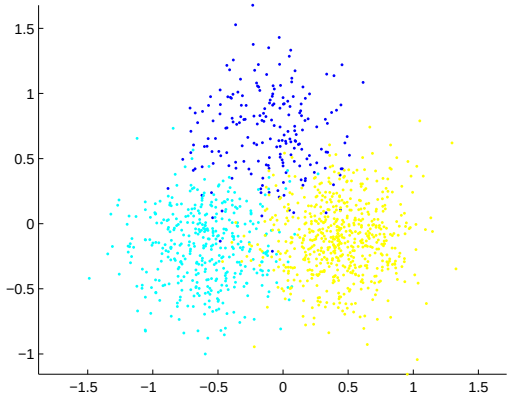
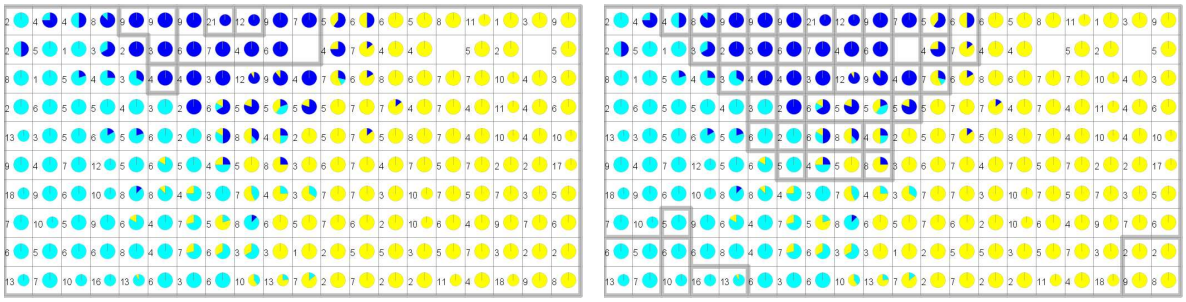



Figure 3.12: Data Set 4: Three Overlapping Clusters

The three clusters in Figure 3.12 have different sizes: 200 (blue), 400 (cyan) and 600 (yellow) data items. They are generated using a Gaussian distribution. The distance between the centers of the clusters is the same for each pair of clusters and chosen so that the clusters slightly overlap.

### Single Linkage

Figure 3.13 shows the result of clustering the close-by clusters with the single linkage function. The clustering into six and into 42 clusters shows that the single linkage algorithm separates single units or small clusters of the blue data items. The reason for that is that there are fewer data items in the cluster and therefore the data items have a greater distance between each other.

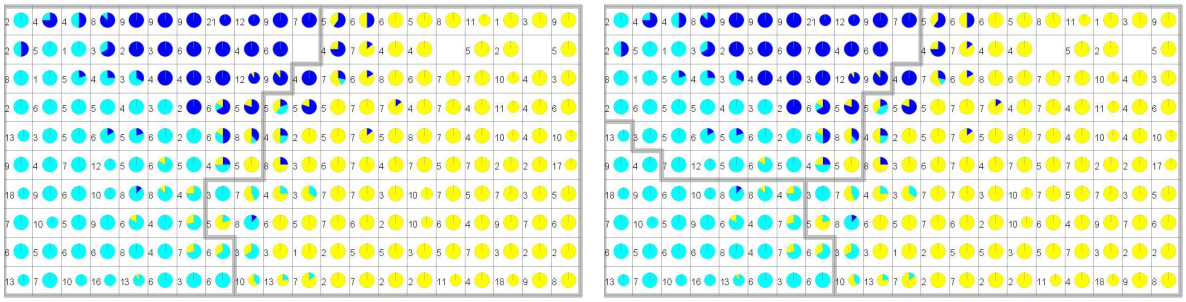


(a) Clustering into 6 Clusters (b) Clustering into 42 Clusters

Figure 3.13: SOM of Data Set 4, Single Linkage

### Complete Linkage

With the complete linkage function it is possible to identify the yellow cluster (Figure 3.14(a)). The other two clusters cannot be separated clearly with this algorithm. The border is drawn through the cyan cluster as shown in Figure 3.14(b).

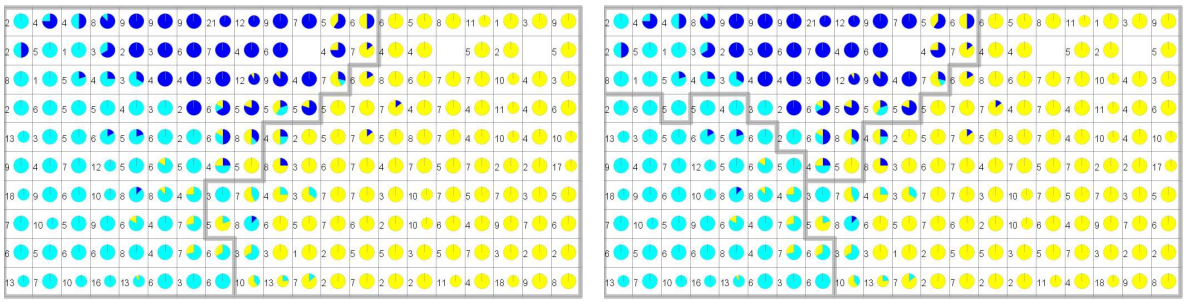


(a) Clustering into 2 Clusters (b) Clustering into 3 Clusters

Figure 3.14: SOM of Data Set 4, Complete Linkage

**Ward's Linkage**

The yellow cluster is again detected correctly as seen in Figure 3.15(a). The blue and the cyan clusters are also detected quite well (Figure 3.15(b)).



(a) Clustering into 2 Clusters (b) Clustering into 3 Clusters

Figure 3.15: SOM of Data Set 4, Ward's Linkage

## 3.2 Adding Color Palettes

The border lines of the clustering split up the map in areas just like borders separate countries on cartographic maps. On top of that, various visualizations are already implemented in the SOMToolbox which produce some sort of relief information for the map. On cartographic maps contour lines and colors are used to visualize the relief of the landscape. Contour lines run through points having the same elevation. Steep terrain can be identified by contour lines lying closer to each other than in flat areas. When a color gradient is used to represent the altitude information, this is called hypsometric tinting. Areas of equal elevation are drawn with the same color. A typical color scheme ranges from green for lowlands, processing through yellow and brown to gray or white for mountain tops.

The same can be applied to Self Organizing Maps to assist the viewer of the map in understanding the complex visualizations. This relief information can be be drawn using several palettes, for example grayscale, redscale or different versions of a palette which make the image look like a picture of islands as shown in Figure 3.16.

As an extension a new set of palettes is introduced which should make the visualization images look like a topological map of a mountainous region . Therefore two different color schemes from maps available online are taken. The first one is a map of Austria found at

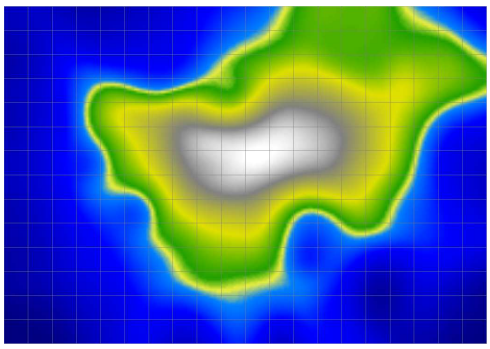
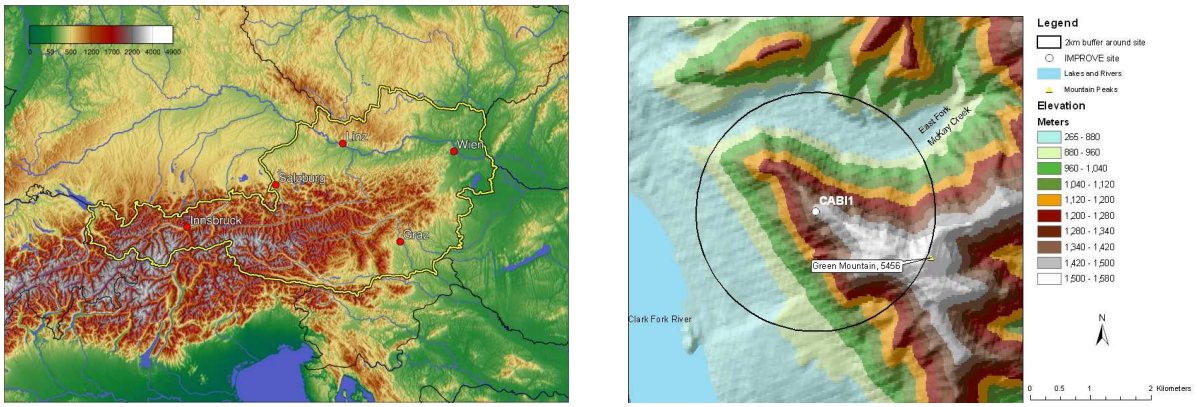


Figure 3.16: Island Color Palette

[mapa] and the second one shows a part of the Cabinet Mountains in Montana and is taken from [mapb].




(a) Map of Austria (b) Cabinet Mountains, Montana

Figure 3.17: Cartographic Maps

Those two color palettes are now implemented in the program with some variations like the number of colors in the gradient or modifications of the ranges of some colors.

Figure 3.18 shows the same data set as Figure 3.16 applying an adaption of the palette used in the map of Austria. The resulting mountain looks unnatural and does not resemble a mountain on a topographical map. This is mainly because this mountain is very smooth and lacks peaks and valleys.

Figures 3.19(a) and 3.19(b) again use the same data set but with the color palette taken from the second map. The image using only ten colors bears more resemblance with a real map as the smooth image using the color gradient of the same palette.

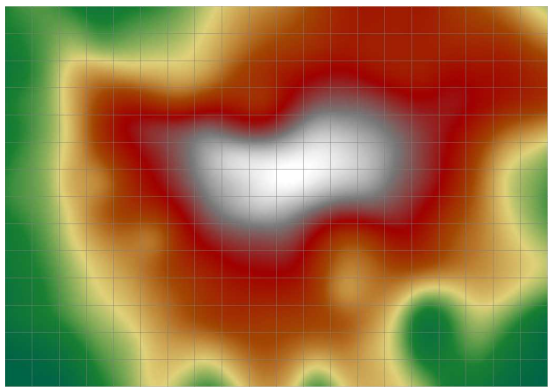
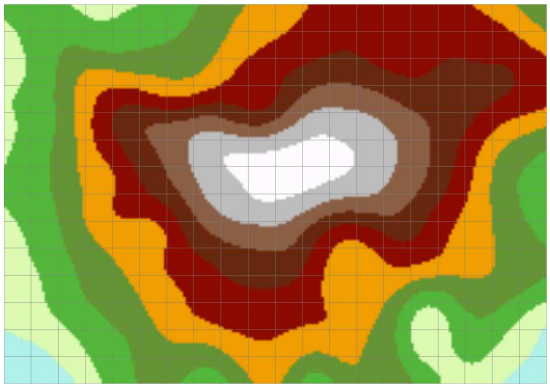
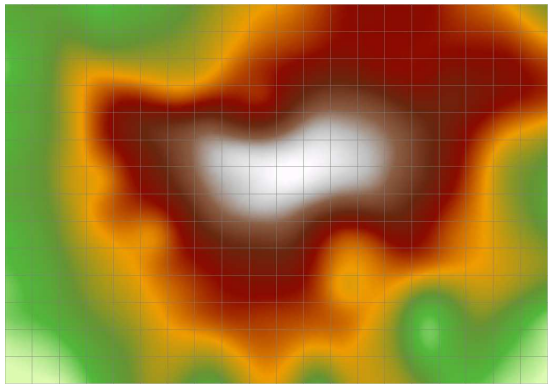


Figure 3.18: SOM with First Color Palette



(a) 10 Colors



(b) 256 Colors, Gradient

Figure 3.19: SOM with Second Color Palette

## 3.3 Coloring the Clusters

To better visualize the clusters they can be colored. The user can choose between all color palettes available in the program.

The coloring algorithm takes the palette and assigns it to the top cluster containing all elements. The two "child" clusters each get one half of the palette which they again split up among their "children". The color which is used to paint a cluster is determined by the middle of the palette.

The Algorithm has the following characteristics:

- If the number of colors is even the color before the middle is taken.

- If the number of colors is odd the first child gets one color less than the second.
- Clusters which are close in the hierarchy have a similar color.
- Outlier clusters have a more different color.
- In the worst case $2^n$ colors are needed to paint $n$ clusters in different colors.

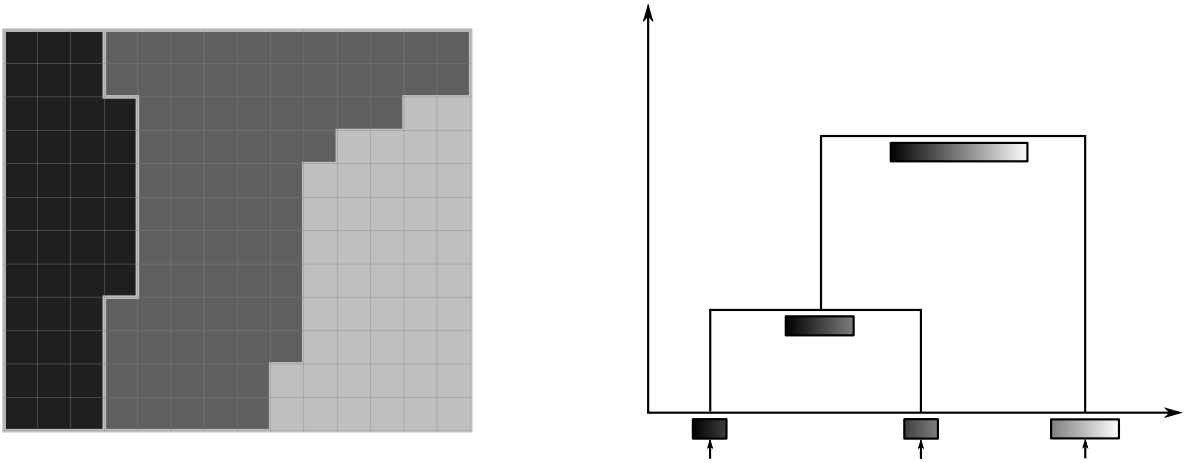


Figure 3.20: Gray-scale Colored Clusters with Dendogram

An example for a situation where many colors are needed is a data set with a lot of outliers clustered with the single linkage algorithm. When the outliers are merged to the cluster one by one in the last steps of the clustering, this means that in each step the coloring algorithm gives half of the palette to the single element and the other half to the rest. Therefore the number of available colors for painting large areas of the map can get rather low.

Figure 3.20 shows a clustering into three clusters and the dendogram for merging the last three clusters into one. The arrows in the dendogram point to the color taken to paint the cluster.

Due to the fact that the contrast between the colors of two neighboring clusters can get very low after only a few steps, an additional palette with random colors has been added. The palette is initialized once when the program is started.

Figure 3.21 shows a clustering into ten clusters. Using the gray-scale palette the clusters in the middle already have very similar colors. In such cases the random palette can be useful to clearly distinguish the clusters.

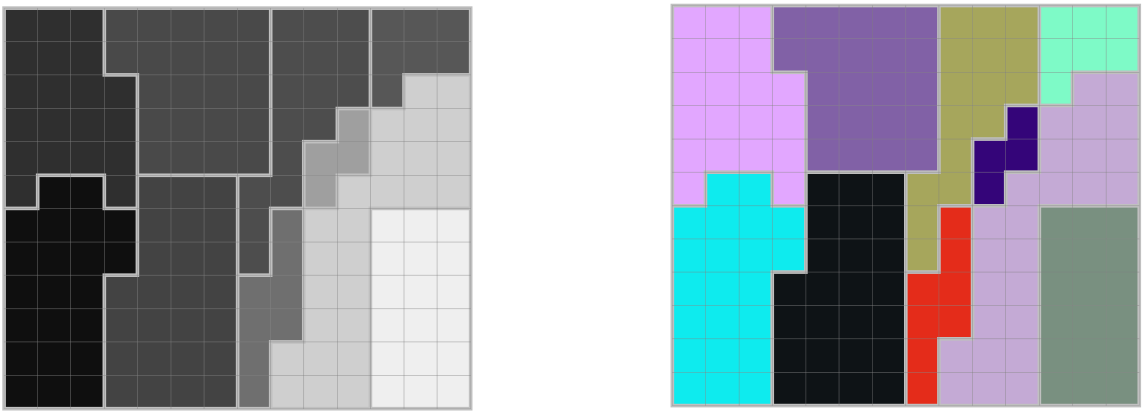


Figure 3.21: Gray-scale and Random Colors for 10 Clusters

## 3.4 Border Color

The color of the border was chosen in a way that labels and other information are still visible when the border line is very wide. Therefore it is not possible to draw the lines black which would make the line very well visible, as this is the default color for all labels on the map. On one hand another color than a shade of gray would highlight the borders but on the other hand it is not useful as very often the maps need to be printed without color.

A light shade of gray keeps other information on the map readable and is still visible enough. Figure 3.22 shows a close up of the map in Figure 5.19 where the border line overlaps other objects on the map.


Figure 3.22: Border Overlapping Labels

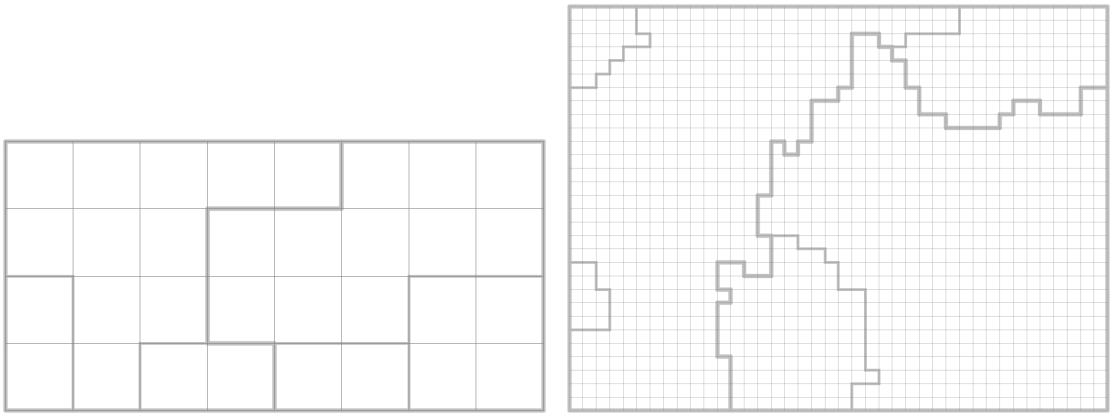
## 3.5 Border Width

The border width is set depending on the width of the map in units. This is done to ensure that borders are visible on large maps and are not too obtrusive on small maps.

It is possible to display several layers of clusters at the same time. To recognize which borders are of which level, the borders of lower levels are one third thinner than the borders of the level above.

Figure 3.23(a) shows a map with 8x4 units. There are two layers of clusters displayed: a separation into two clusters which horizontally separates the map into two parts and the division into six clusters which is drawn with thinner lines.

In Figure 3.23(b) a larger map containing 40x30 units is shown. Again the borders for two and six clusters are drawn. At this zooming level the borders in the larger map appear to have a similar width as in the smaller map. It is also noticeable that the border lines' width compared to the unit size is thicker in the larger map.



(a) Map Sized 8x4 (b) Map Sized 40x30

Figure 3.23: Small and Large Map with 2 Levels of Clusters

## 3.6 Summary

Applying the linkage functions to the four test data sets showed the strengths and weaknesses of the different functions. The single linkage algorithm performs only better then the other ones with data where the distances within a class from one data item to the next are lower than

to the distance to the closest item from another class. As shown in the example data set with the two rings the algorithm can identify clusters of arbitrary shape.

The complete linkage algorithm performs better than single linkage with data where some items of the classes get close to each other. If the classes overlap too much as in the example data set 4 this algorithm also has difficulties identifying the cluster borders.

Except for the data set with the two rings the Ward's linkage algorithm could find the most suitable cluster borders compared with the other algorithms.

The second part of this chapter dealt with displaying the clusters. Since the goal was to make the clustered map resemble a cartographic map, additional color palettes have been added which can be used to make the various visualization images look like a map of a mountainous region. The borders of the clusters can then be interpreted as country or, on lower level, district borders. If there is no background image from a visualization, it is also possible to paint each cluster in a distinct color, which is especially helpful to recognize discontiguous areas of one cluster.

# Chapter 4

# Labeling Clusters

Having a clustered SOM, it is now possible to see a structure in the map. Without any knowledge about the content of the clusters this information is unhelpful. Therefore it is necessary to automatically label the clusters. First of all, it must be decided which parameters should influence the decision process. Then the labels must be calculated using those requirements and finally the labels have to be placed on the map.

## 4.1 Choosing the Label

There are two kinds of data that can be used to determine cluster labels:

- The input data vectors. The disadvantage of this method is that lots of data needs to be processed, which can take much time, especially if the map contains a lot of units.
- The labels of the units. There can be different types of labels and for the calculation of the cluster label it makes no difference how those unit labels were chosen. Of course the unit labels are created based upon the input vectors but this method can be used regardless of the algorithm used to create the units' labels. The only features those labels need to have is either a numerical *value* and/or a *quantization error*.

  The weak point of this method is that information might be lost. Potentially each unit has as many labels as there are dimensions in the input vectors. The unit labeling method makes a selection of these labels and if it only creates few labels, all other potential labels cannot be taken into consideration when labeling the cluster.

We decided to use the latter method because it gives more flexibility. As there are different kinds of labeling algorithms for the units also the labels of the clusters can be different.

The next design decision that had to be taken was when and how many labels should be calculated. Even if they are not shown it is helpful to create more than one label per cluster. Especially if the labeling process is done bottom up several labels are needed to create a meaningful label for a higher level.

- The first approach is to create the labels while clustering the SOM. The labels of the cluster are calculated based on the labels of the two clusters that are merged together. The cluster gets as many labels as each cluster below had, i.e. the clusters have the same amount of labels as the units. One advantage of this method is that it is faster than considering all of this cluster's units' labels. The disadvantage is that data might get lost or blurred because for each cluster only a small amount of labels is created which can thereafter be used to create the labels of a higher level. Therefore it can happen that a common label for a cluster can not be found because it was not stored in a lower level for one or more of the sub-clusters.
- To improve the quality of the created labels it is possible to store all labels during the whole clustering process. This means that to determine the label of the top cluster all labels of all units are available. This method is considerably slower than the first.
- The third option is to only calculate the labels that are currently shown on the map using all of the cluster's units' labels in the calculation.

**Parameters**

If a unit contains input data there are three kinds of information that can be used to decide on the labels.

- Number of data points mapped to the unit:
  Each unit can, but does not have to, contain input data. The number of input data can be used for weighing the influence of a unit's properties to the labeling of the cluster, e.g. labels from more densely populated units get a higher weight.

- Value:

  Each label contains a value which is the mean value of all input vectors' values for the label. The meaning of this value depends on the input data; It could for example be a measure of length as in the Iris data set mentioned in Section 2.4.2 or it could also be a boolean value.

- Quantization Error:

  The quantization error is the average distance between the input vectors' values and the value of the weight vector for one label. The lower this value, the closer the input data lies concerning this feature making it a good descriptor of the similarity of the input data.

It depends on the data whether it is more useful to consider the value rather than the quantization error for ordering the labels. The quantization error is a criterion for the quality of the label, because a low quantization error indicates that the value is similar for all input data mapped on to the unit. Therefore it is in most cases a good ordering criterion. On the other hand there are data sets where the value is more important than the quantization error. Since it cannot be generally decided which is more important it is up to the user to weigh the influence of those two parameters.

Text data sets are a typical example where pure ordering by quantization error can yield unmeaningful labels. Here the value is also important as it describes the frequency of the word occurring in the texts. A label which has a low quantization error because it is non existent in all input texts should not be considered as a label for a cluster. On the other hand a label with a high value means that the label is prominent at least in some documents.

The number of inputs currently does not influence the labeling of clusters.

## 4.2 Structure of a Label

Intuitively, a label is some text describing the common characteristics of the data elements of the cluster. In this implementation a cluster's label is structured as follows:

The label is merely a container that can consist of several texts. Each text can have multiple lines. There are properties of the label which affect the whole label whereas others are set

separately for each text. Those properties can be edited manually and they are described in more detail in Section 4.5.

In Figure 4.1 there are two examples for labels. The left label contains only two text items of which the first one has two lines whereas the right contains three text items each only single lined.


Figure 4.1: Different Labels

**Labels with values**

Depending on the data the name of the label alone might not be meaningful enough to describe the cluster. The user has the option to additionally display the corresponding value to the displayed text. The value is added to the text in a new line. It is up to the user to manually remove the line break to have it in the same line. In most cases the label is wider than high already so it is better not to make the label even broader when adding this additional information.

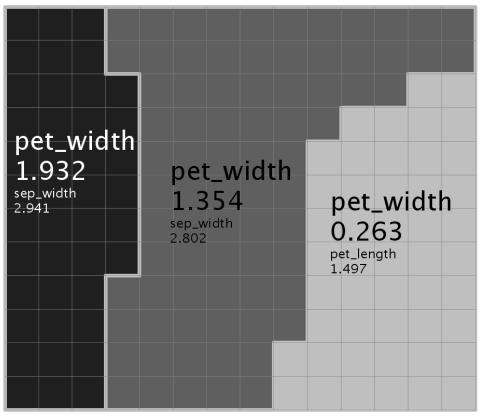



Figure 4.2: Values Add Meaning to the Labels

The example in Figure 4.2 shows the clustering of the iris data set into three clusters. The

words in the labels are the features and the values are the corresponding mean values for their values over all input data inside this cluster. In this case the labels are ordered by quantization error to get the labels which describe the cluster best. Since all clusters are best described with the same first label all three are equally labeled. Obviously this does not give much information, but adding the value shows that the value for this specific feature is different for each cluster. Furthermore it can be seen that the second label of the cluster on the right hand side is different from the second label in the other two clusters.

## 4.3 Font Sizes

Since the size of the map can vary a lot, the first step was to decide on the font sizes of the labels. There is one default font size calculated in the beginning. It is ten times the map's number of units in width. When there are several layers of clusters displayed the size of the font for the smaller clusters needs to be adapted to visualize to which layer they belong. They should not get illegible too fast but still they need to be noticeably smaller. This can be achieved by making the labels one third smaller in each lower layer.

In case the user wants to display more than one label per cluster the labels quickly take up a lot of space. This can be avoided by also making these labels smaller. To avoid mixing up those labels with the labels of another layer they are placed directly under the first label of the cluster having half its font size.

## 4.4 Placing the Label

Now that the content of the label has been determined, we can move on to finding an appropriate position for it on the map. The perfect labels should be placed inside the cluster they belong to in a way that it is obvious which area they describe, even if there are a lot of clusters displayed. Furthermore they need to visually support displaying several layers of clusters and more than one line of text per label. Nevertheless the labels should stay readable and not overlap each other.

Since all this cannot be achieved with reasonable time and effort we decided only to approximate a position and leave it up to the user to manually edit the labels and move them to

their final position.

#### Position inside the cluster

We decided to leave aside any global constraints and only find a position for the label within its cluster. For most cases it is sufficient to place the label in the center of the cluster.

In this implementation there are two options:

- Center of the surrounding rectangle:
  The rectangle surrounding the cluster is calculated during creation of the cluster.
- Centroid:
  The centroid is determined in the label placement process.

In both cases the center of the label's first text is placed in the specified position.

## 4.5 Manual Editing

Since the methods for placing the label do not check whether the label overlaps with another label or exceeds the bounds of the cluster it is possible to manually move and edit the label.

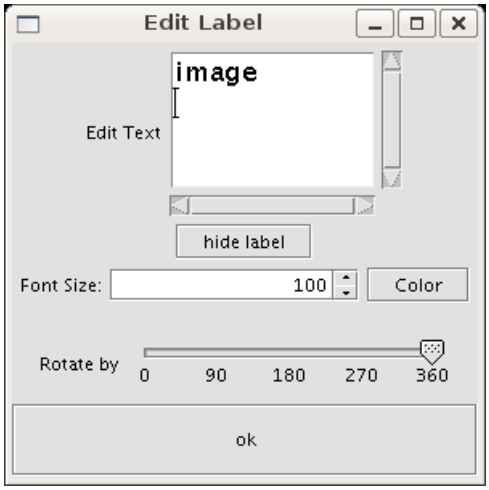



Figure 4.3: Dialog for Editing Labels

- Text:
  The label's texts can be changed to any user-defined text. This text can have multiple lines - all with the same font size.

- Position:
  The user is able to drag and drop the label around the map to any position as desired. The whole label is moved together so the text items of a label cannot be moved separately.

- Font size:
  In case the font size of the label is not appropriate, it can be changed to any integer value. The change of the font size of the first text of a label does not influence the font size of the other texts. Every text's font size can be edited independently. When the editing of a label is finished and the "ok" button is pressed the positions of the label's texts are adapted so they neither overlap nor have too large spaces in between.

- Rotation:
  The label can be rotated to any degree. All the label's texts are rotated together. They are rotated around the center of the first text of the label.

- Color:
  The color of the label can be changed to highlight the label. This is especially useful if the visualization or background image has a similar color as the label. This can be done by choosing one of the predefined colors or choosing the color using either the HSB or RGB color model.

- Hide:
  If a label text should not be displayed, it is possible to hide it. The label is then not visible on the map, but it is still there, so it can also be selected and edited as any other label. It can also be made visible again.

All the changes are immediately applied to the label and visible on the map, except for the repositioning of the texts inside one label in case of font size changes. They are only done when the "ok" button is pressed. If another label is selected then the new label is associated with the dialog and no repositioning takes place.

#### Manually Adding Labels

Since the automatically generated labels might not be sufficient it is also possible to manually add labels. Those labels contain one text field which can have several lines. Due to the fact that it is a single text, all changes of attributes affect all lines of text. If different properties are desired, it is necessary to create several manual labels.

The manual labels always appear in the top left corner of the map and can thereafter be moved into the desired position.

Like any other labels these labels can be edited as desired. If the label is not needed any more it cannot be removed completely but it can be hidden and is therefore not visible any more on the map.

## 4.6 Automatic Changes

There are a few cases where the label's properties need to be adapted to changing conditions of the map.

- When a clustering level above an already displayed level is added to or removed from the map the font sizes of the labels change so that the lower layer gets a smaller font size. The position of the label stays the same, in a way that the center of the label's first text stays in the same position. The font size is adapted for all the label's texts and also the space between the labels is adjusted. If the font size of the label was manually edited it will be changed nonetheless. Other manually changed properties like the text, rotation, color or visibility will stay as set by the user.

- When the user chooses to change the way the labels' texts are decided upon, the texts of the labels are changed even if the text was manually edited by the user. The manually entered text can not be restored.

- If values are added to the labels or removed from them, the text of the label is reset. This makes sense especially when the value is added as the value describes a specific feature of the cluster and it might not make sense with the manual description.

Each text of the label gets a second line where the value is displayed. In order to avoid overlapping text all label texts from the second on are moved down.

The position of the label is not changed, even though the center of the first text is now somewhere else due to the added line in the text. This is not done to prevent moving manually repositioned labels.

- If the font size of one of the label's texts is changed the label adapts the spaces between the texts.

## 4.7 Saving and Restoring Labels

Sometimes it is useful to save the current labeling and restore it later for further editing or working with the map.

When saving, one can choose between two options:

- clustering
  The whole clustering tree is saved including all cluster information, the border lines, the labels and all their transformations. Furthermore all manually added labels are saved.

  This method to store the label is very sensitive to changes in the program's source code. A lot of classes are written to the file and if one of those classes changes between saving and loading it might be incompatible and so it gets impossible to load that file again.

  After loading such a clustering to a map it is possible to edit it again, just like before saving.

- xml
  To avoid the problem of incompatible class versions it is also possible to store the labels in an xml file. In this case no cluster information is stored and it is not possible to assign a label to its cluster. All the recreated labels are treated as manually added labels, with the only specialty that one label can consist of several texts.

The method to save the label data is determined by the extension used for the file. If it equals "xml" then the labels are saved as xml, otherwise the whole clustering is saved. Loading

the labels is done accordingly, so if the file's extension is not xml the program assumes that it contains the complete clustering and will fail if this is not the case.

## 4.8 Summary

Deciding which properties of the units and their labels to use for creating the cluster label depends on which kind data should be labeled. For text collections a high *value* is an indicator for a good label whereas in general a low *quantization error* is desirable.

The main application area for labeling is text collections, especially large collections, where the input vectors can have some thousand dimensions of which most have a value close to or equal zero. Depending on the size of the map, there can be a lot of input vectors mapped to one unit. Since the unit labels are already a selection of terms which describe the content of all inputs on the unit, these labels are taken into account when choosing the cluster labels instead of the raw input vectors.

The labels created for the clusters depend on the quality and number of labels available for the units. If the units have more labels, the cluster labels can be better, but do not have to get better. It depends if the additional labels occur frequently in the units of the clusters or if they are unique and therefore do not help in finding a common label.

Choosing the right size and position for a label is somewhat tricky because it is not known in advance how much space is available for the label. On one level, the sizes of the clusters can range from a single unit to covering the majority of the map. Nevertheless the labels should have the same size. A compromise was implemented which chooses the size depending on the width of the map and positions the label in the center of the cluster. Of course this makes labels overlap each other if the clusters are small or places labels outside their cluster in case of a cluster covering not connected areas. Those labels can be manually corrected, for example by abbreviating the text, moving or resizing the label.

# Chapter 5

# Experiments

The following experiments will be performed using the 20 newsgroups data set. This data set has become very popular for text experiments in machine learning and has been used for example in [Mit97]. The data files can be found at [20n].

From this data set analogies can be drawn to company archives. Large companies are organized in hierarchical structures and in each department of this hierarchy large amounts of documents are managed. The documents from one department will deal with the subject of the department but this subject can have a variety of topics; for example a department for research will have documents on all their different research activities as well as documents dealing with financial or human-resource issues. For an outsider, or even a manager from the company it is not possible to have an overview of all the information resting inside the repository. They have to rely on annual reports to get a general idea of what was worked on. The idea behind the following experiments is to show how a large amount of data can be structured and viewed to the user aiding him in getting an overview of the information. Due to lack of real life data from company archives the 20 newsgroups benchmarking data set is used.

In this chapter first the data set will be described, then the maps will be clustered and labeled as presented in Chapters 3 and 4. The results will be evaluated by comparing them with the U-Matrix or SDH-Visualization and by reviewing how well the cluster structures and labels suit the underlying data.

## 5.1 20 Newsgroups: Data Set Description

A newsgroup is usually a discussion group where users from all over the world can post messages. To simplify finding a group, the newsgroups are often arranged into hierarchies.

From the following top level hierarchies, groups were selected for this data set:

- comp.*: Discussion of computer-related topics
- sci.*: Discussion of scientific subjects
- rec.*: Discussion of recreational activities (e.g. games and hobbies)
- soc.*: Socialising and discussion of social issues.
- talk.*: Discussion of contentious issues such as religion and politics
- misc.*: Miscellaneous discussion (anything which doesn't fit in the other hierarchies)
- alt.*: short for alternative; for groups which were not allowed in the other hierarchies

The 20 newsgroups data set consists of newsgroup postings from the 20 newsgroups listed in Table 5.1. From each newsgroup, one thousand articles from the year 1993 have been taken. Approximately 4% of the articles have been removed because they were posted in more than one of the newsgroups. Therefore it is possible to assign to each article one class corresponding to the newsgroup where it was posted.

From each of the articles a vector has been extracted, which represents the frequencies of around 3500 words that can be found in the articles. The words are chosen from a list of all words occurring in the texts, excluding words that appear in most or very few documents and so-called stop words (like articles or prepositions).

The names of the newsgroups are used as the class labels for the map. The colors used to visualize the newsgroup articles on the maps are shown in Figure 5.1.

As the input vectors result from natural language, it is not possible to spatially separate the newsgroups on the map: Some articles are much longer than others and, therefore, contain more different words. Some newsgroups allow diverse topics, hence the words used in the messages can vary a lot. A couple of articles contain off topic discussions and, consequently, have only few words in common with other articles from the same newsgroup. In their case it

| | |
|---|---|
| alt.atheism | Godless heathens. |
| comp.graphics | Computer graphics, art, animation, image processing. |
| comp.os.ms-windows.misc | General discussions about Windows issues. |
| comp.sys.ibm.pc.hardware | XT/AT/EISA hardware, any vendor. |
| comp.sys.mac.hardware | Macintosh hardware issues & discussions. |
| comp.windows.x | Discussion about the X Window System. |
| misc.forsale | Short, tasteful postings about items for sale. |
| rec.autos | Automobiles, automotive products and laws. |
| rec.motorcycles | Motorcycles and related products and laws. |
| rec.sport.baseball | Discussion about baseball. |
| rec.sport.hockey | Discussion about ice hockey. |
| sci.crypt | Different methods of data en/decryption. |
| sci.electronics | Circuits, theory, electrons and discussions. |
| sci.med | Medicine and its related products and regulations. |
| sci.space | Space, space programs, space related research, etc. |
| soc.religion.christian | Christianity and related topics. (Moderated) |
| talk.politics.guns | The politics of firearm ownership and (mis)use. |
| talk.politics.mideast | Discussion & debate over Middle Eastern events. |
| talk.politics.misc | Political discussions and ravings of all kinds. |
| talk.religion.misc | Religious, ethical, & moral implications. |

Table 5.1: The 20 Newsgroups and their official description from 1993

also makes more sense to group them with articles of other newsgroups dealing with the same topic.

We will now look at different maps. For the maps in Section 5.2 and 5.3 the words were taken from the articles directly whereas in Section 5.4 a stemming algorithm that removes prefixes and suffixes from the words has been applied before creating the vectors.

## 5.2 Map 1

To create the first map, the newsgroup postings have been taken with all their header information, excluding only the line which contains the name of the newsgroup. All words occurring in the newsgroup postings were taken as they are written in the text which makes for example singular and plural of the same word or two different forms of a verb count as two separate

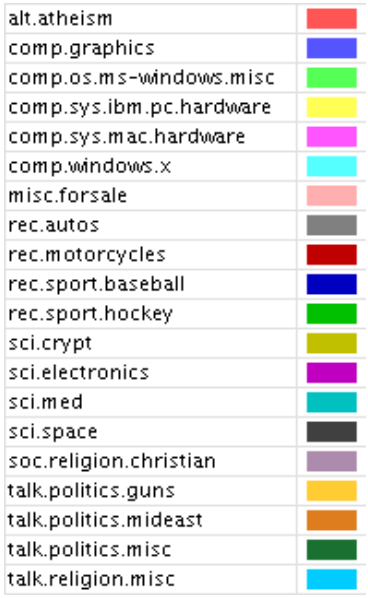



Figure 5.1: The Colors Used for the 20 Newsgroups

words.

This results in a very long list of words of which only the words with a minimal length of 3 letters were used. Words that only occur in less than four percent of the documents have been removed as well to reduce the list. Besides removing terms from the list based on the document frequency and word length, some more have been omitted:

- English and German stop words
- some manually selected, very frequent words which do not help in distinguishing the newsgroups (like “wrote” or “posting”), i.e. corpus specific stop words
- numbers and dates

This results in a final list of 3646 words. That means that the input vector created for each document has 3646 dimensions and holds the frequencies for those words inside the document.

The size of the map was chosen to be 75x55 units. The training process was done with 500.000 iterations.

Figure 5.2 shows the trained map with the pie charts containing the class information to identify where the classes lie on the map. It can be seen that there are areas on the map con-

Figure 5.2: Map 1: 75x55 Map (Without Stemming)

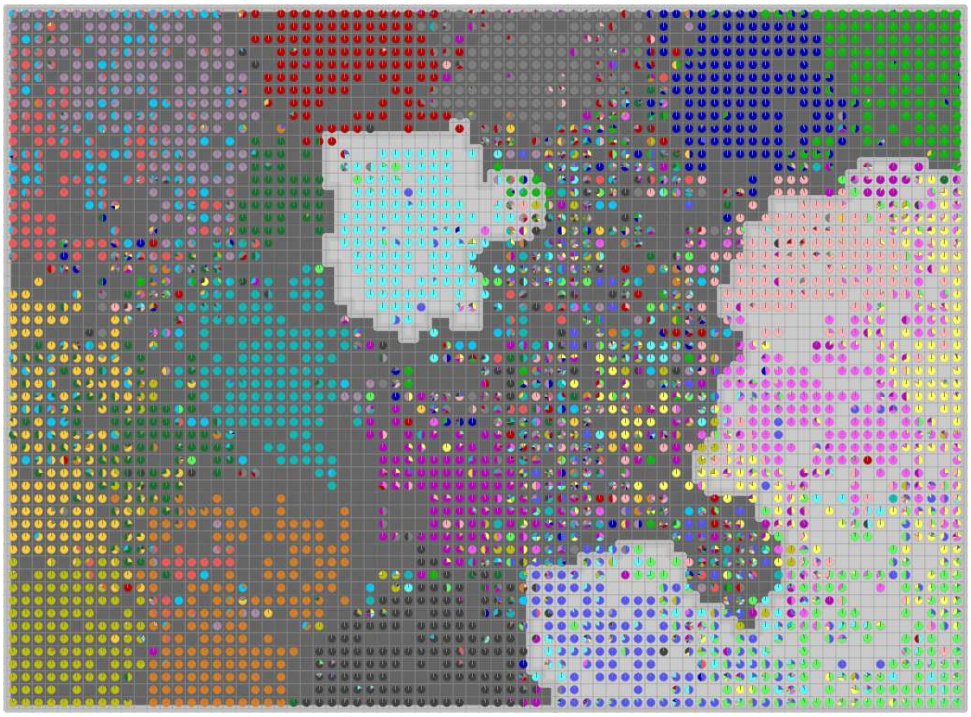


Figure 5.3: Map 1: 2 Clusters

taining mainly items from one class. There is also a large area in the middle which has mixed content.

Figure 5.3 shows two clusters using Ward's linkage of this map. The light gray cluster is interesting because it contains all newsgroups dealing with computers and also the group misc.forsale. A lot of messages of the newsgroup misc.forsale deal with computers or computer parts which are for sale. Some example messages from this newsgroup can be found in the Appendix. The smaller area of this cluster covering basically comp.windows.x is spatially separated from the rest of this cluster but still these areas have been joined by the clustering algorithm to form one cluster.

Ten clusters are shown in Figure 5.4, the random color palette is used to better visualize the clusters consisting of two disconnected areas. The cluster in the top left corner contains most parts of the groups alt.atheism, talk.religion.misc and soc.religion.christian but also parts of the group talk.politics.misc. Note that there is a small, separated area slightly below the cluster but still belonging to it.

The blue cluster in the middle contains the newsgroup comp.windows.x and the other part of this cluster in the lower right contains comp.graphics. In the lower right corner the cluster comprises most of comp.os.ms-windows.misc and some postings from other computer-related newsgroups. The blue cluster on the right side contains both hardware newsgroups and the "for sale" group.

The postings of the newsgroup talk.politics.mideast are already split up in three clusters: the big one in the lower left-hand corner and the two adjacent red and cyan clusters. The big cluster is shared with talk.politics.guns, talk.politics.misc and sci.crypt, whereas the other two clusters mainly contain items solely from talk.politics.mideast. The cyan cluster is a nice example for a cluster surrounded by units with no data items. This is a strong indication for this cluster containing a separate topic.

Right beside the politics clusters, there is a small cluster containing a part of the postings from the newsgroup sci.med. Even though it lies next to the rest of the texts from this newsgroup, it is separated because it contains short postings from one person and answers to him.

Figure 5.5 shows 30 clusters of the same map with labels for these clusters. There are two clusters consisting of two areas: In the top left there are 2 small blue areas which are labeled "morality" and the dark green area without label next to the "morality" cluster belongs to the

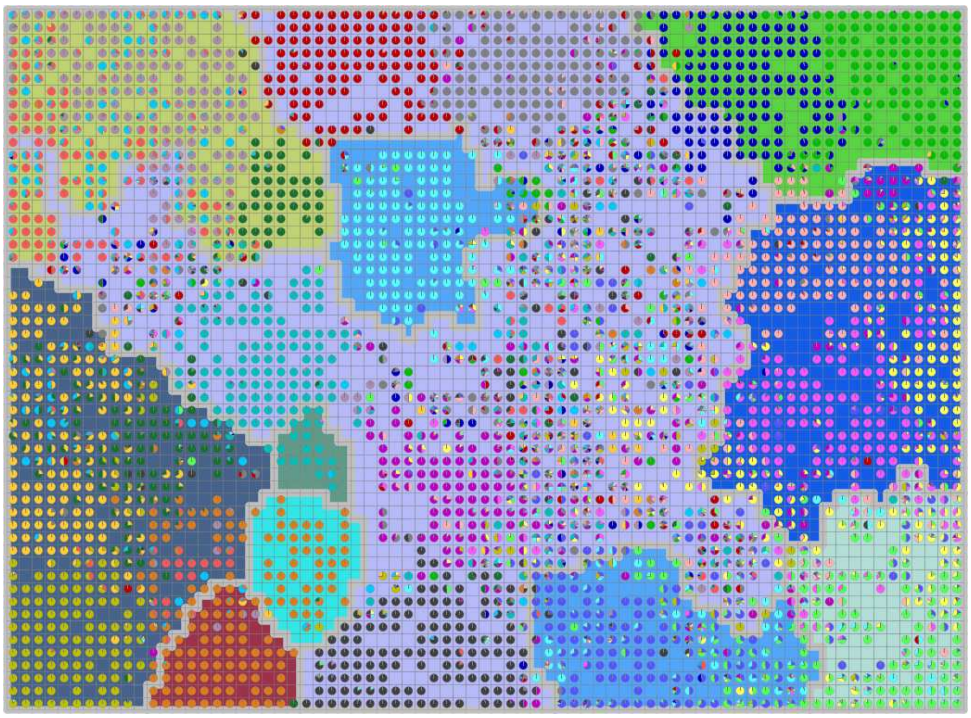


Figure 5.4: Map 1: 10 Clusters

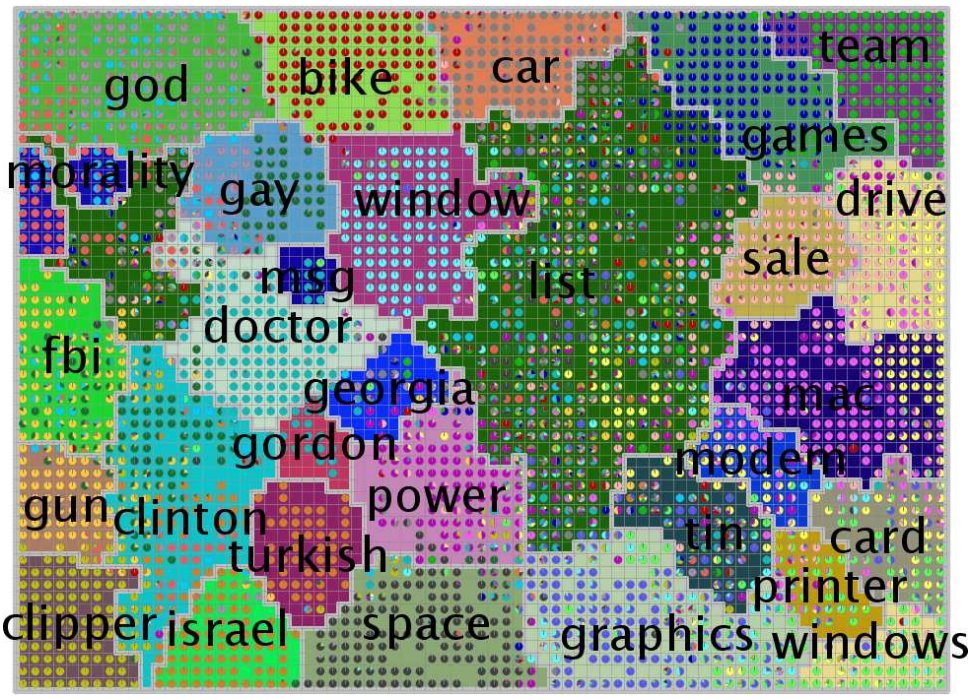



Figure 5.5: Map 1: 30 Clusters With Label

large green area in the middle labeled “list”.

The labels for the clusters of Figure 5.5 are in most cases describing the topic of the contained newsgroups well. Table 5.2 lists some of the labels and the newsgroup(s) which are dominant in the cluster. The largest cluster (labeled “list”) as well as the smaller clusters labeled “georgia” and “tin” contain input data from all classes, especially short postings. Three postings from the cluster labeled “list” can be found in the Appendix.

| Label | Dominant Newsgroup(s) |
|---|---|
| god | alt.atheism, talk.religion.misc, soc.religion.christian |
| morality | alt.atheism, talk.religion.misc |
| bike | rec.motorcycles |
| car | rec.autos |
| team | rec.sport.hockey |
| games | rec.sport.baseball |
| sale | misc.forsale |
| space | sci.space |
| mac | comp.sys.mac.hardware |
| graphics | comp.graphics |
| window | comp.windows.x |
| gun | talk.politics.guns |
| doctor | sci.med |
| israel | talk.politics.mideast |
| turkish | talk.politics.mideast |

Table 5.2: Map 1: Some Labels of the 30 Clusters

Figure 5.6 shows the map clustered into only three clusters with four labels for each cluster. There is a small cluster in the lower left-hand corner which has the labels “turkish”, “armenians”, “armenian”, and “turkey”. This shows that the cluster contains postings from a very distinct topic. The fact that the units containing data items inside this cluster are surrounded by units without data items also indicates that there is a well-defined topic in this area. Furthermore, when looking at the U-Matrix visualization of this map in Figure 5.9 it can be seen that there are high distances between the units in the border area of that cluster.

The cluster in the middle without label belongs to the cluster on the right side. It contains all newsgroups dealing with computers and misc.forsale. The labels are “windows”, “sale”, “drive” and “card”, which match the content of this cluster quite well.

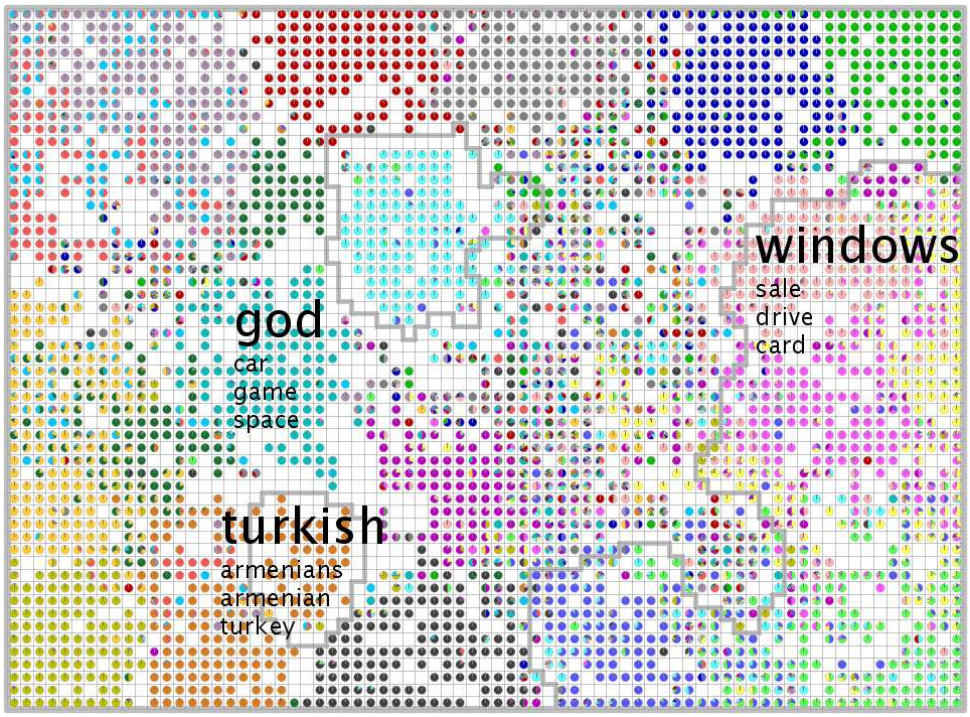



Figure 5.6: Map 1: 3 Clusters With 4 Labels

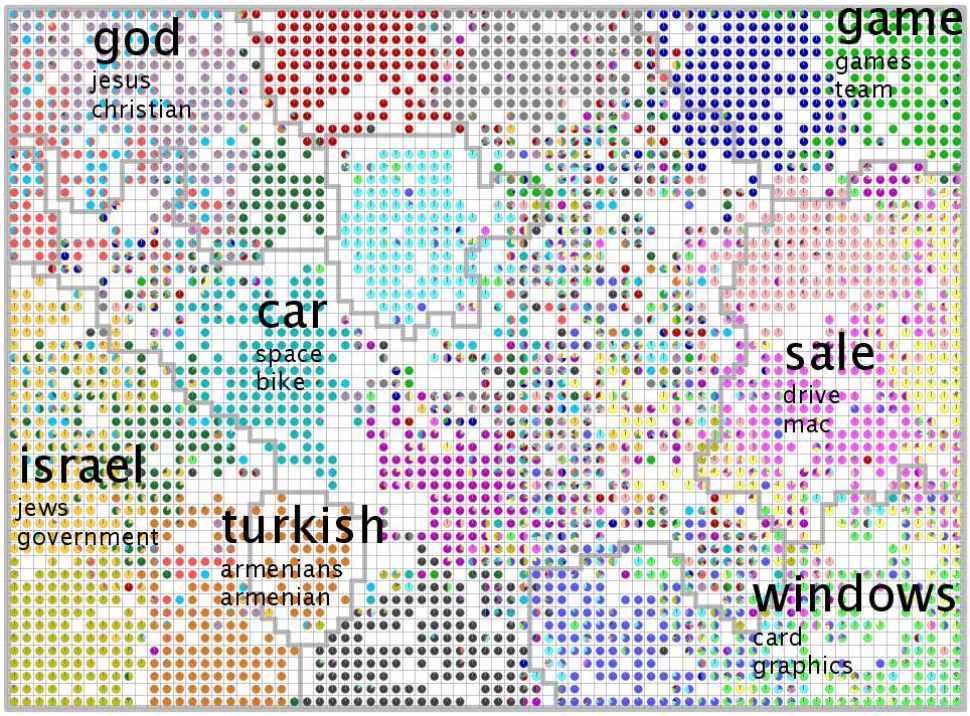



Figure 5.7: Map 1: 7 Clusters With 3 Labels

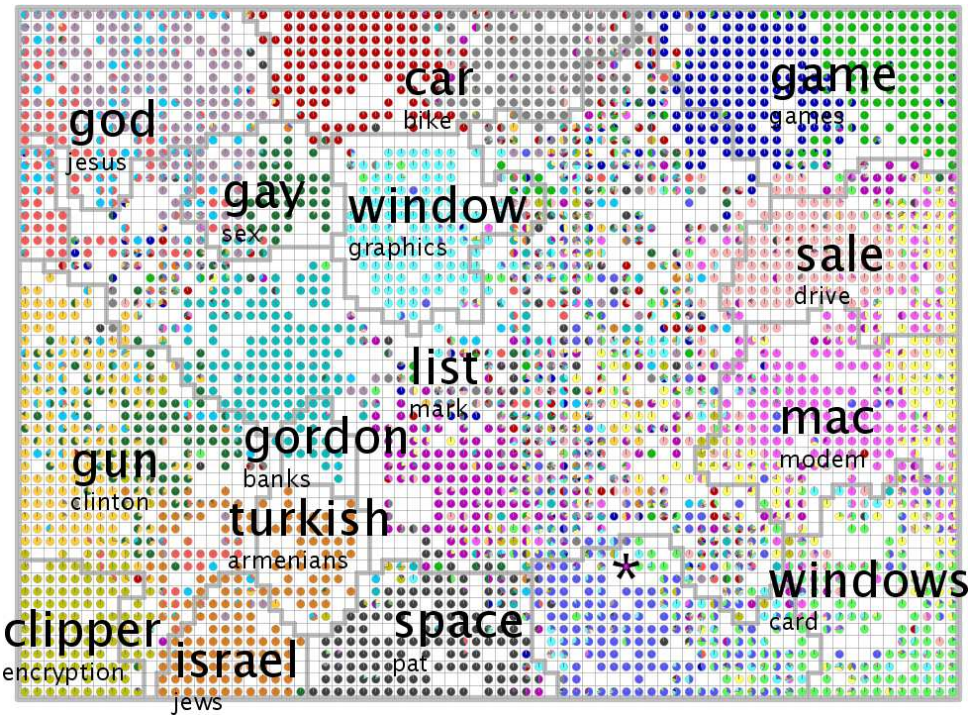



Figure 5.8: Map 1: 15 Clusters With 2 Labels

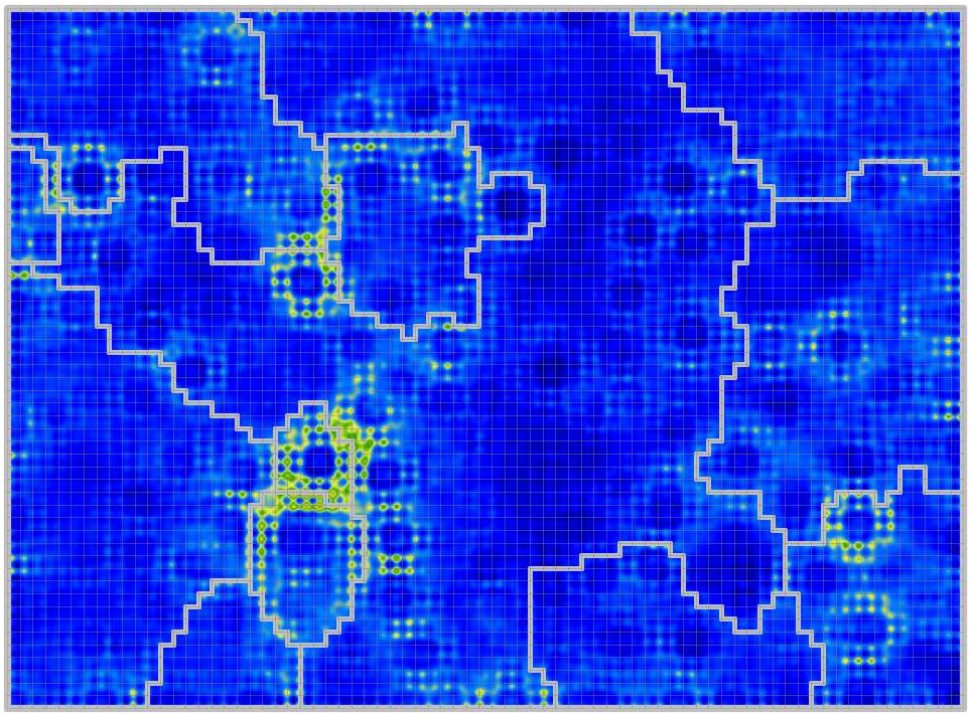


Figure 5.9: Map 1: U-Matrix Visualization

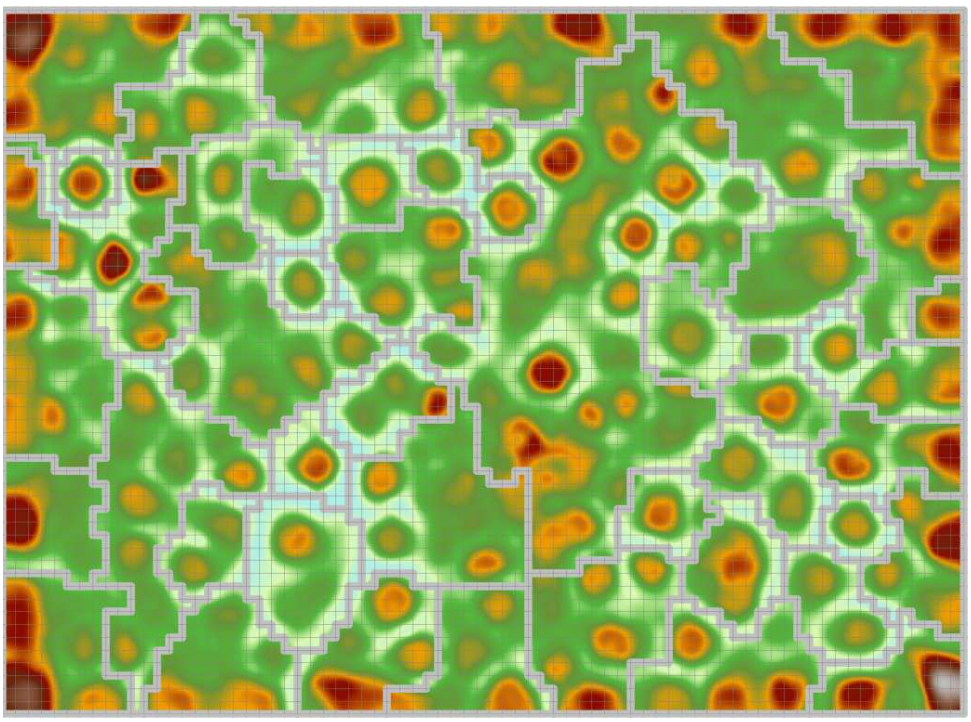


Figure 5.10: Map 1: SDH (Island Palette) and 45 Clusters

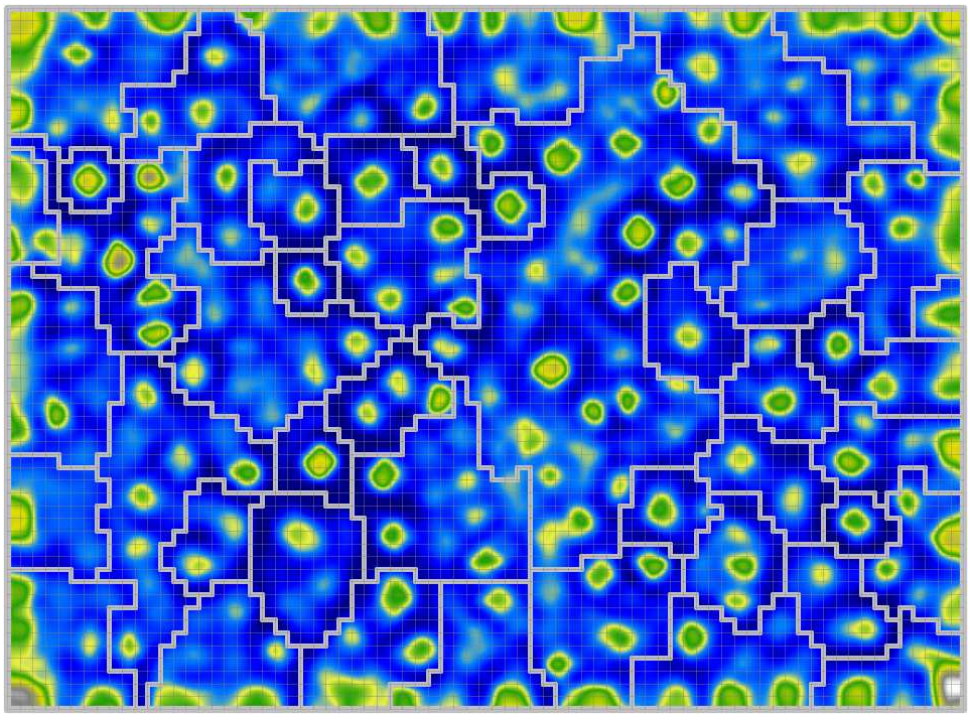


Figure 5.11: Map 1: SDH (Mountain Palette) and 45 Clusters

The large cluster containing the rest has labels from various topics.

Figures 5.7 and 5.8 show some other levels of the clustering and labels for these clusters. These examples are chosen because the number of clusters is still low enough to display more than one label per cluster. In Figure 5.7 the unlabeled area in the center belongs to the cluster labeled “windows”. The cluster in 5.8 with the labels “window” and “graphics” consists of two areas on the map. The second area is manually labeled with a “*”.

In Figure 5.9 the U-Matrix visualization is applied to the map as well as Ward’s linkage algorithm showing ten clusters. It can be seen that cluster borders lie in areas of high distances between the units. Furthermore there is a large area in the middle which has low distances between the units. This indicates a homogeneous cluster in that area. Ward’s linkage algorithm also finds a large cluster in that area which can be seen for example in Figure 5.5.

Figures 5.10 and 5.11 show 45 clusters and also a SDH visualization. The first map uses the island color palette and a smoothing factor of 15. To the second one a smoothing factor of 9 has been applied and a color palette representing mountains has been used. The borders of the clusters lie in areas which are low in the SDH visualization.

## 5.3 Map 2

This map uses the same input vectors as the first map, but this time the size of the map was chosen to be 60x45. In Figure 5.12 the map is shown including the pie charts to visualize where the newsgroups lie on the map.

Some classes have different neighboring classes from the ones they had in map 1. For example, the newsgroup sci.crypt is now located in the lower left corner next to rec.motorcycles, rec.autos and the two hardware newsgroups. Furthermore talk.politics.mideast is now separate from the the other two politics newsgroups and is located next to the groups dealing with religion. Still, Ward’s clustering algorithm joins sci.crypt with talk.politics.misc and talk.politics.guns (which were spatially close in map 1) in one cluster when clustering this map into five clusters as shown in Figure 5.13.

As before, the groups dealing with hardware, the sports groups and the motorcycle and car groups are each located spatially close. On this map the two hardware groups are in the bottom of the map but the other groups dealing with computers are in the top right corner.

Figure 5.12: Map 2: 60x45 Map (Without Stemming)

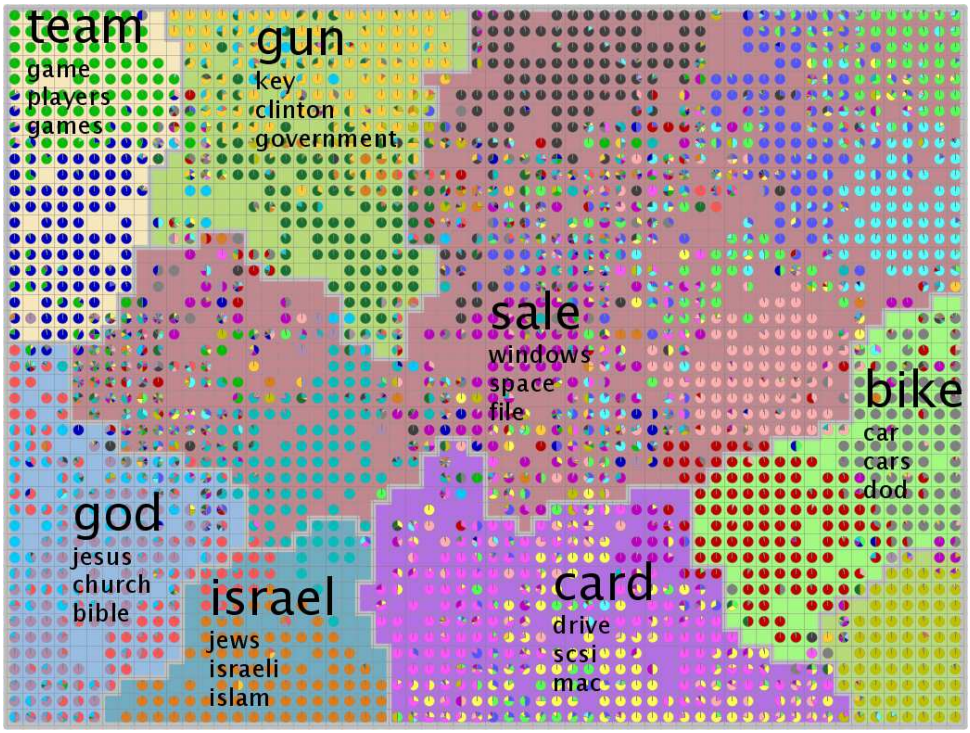



Figure 5.13: Map 2: 7 Clusters With 4 Labels

Figure 5.13shows the map as well as seven clusters, labeled with four labels each. There are eight separate areas because the cluster labeled “gun” is composed of two spatially separate parts. The groups dominating in this cluster are talk.politics.misc, talk.politics.guns and in the lower right-hand corner sci.crypt. Except for the label “key” the labels are all politics related.

The cluster labeled “god” contains most of soc.religion.christian and talk.religion.misc and parts of alt.atheism. Most other parts of alt.atheism are together with talk.politics.mideast in the cluster labeled “israel”.

As in the first map the two newsgroups dealing with sports are joined in one cluster which here has the labels “team”, “game”, “players” and “games”. All these labels are good descriptions of the content of the cluster.

rec.autos and rec.motorcycles lie in one cluster on the right side of the map labeled “bike”. The first two labels together relate nicely to the two newsgroups dominating in this cluster. A small amount of units containing input data from rec.motorcycles is split-off and belongs to the cluster labeled “sale”.

The two hardware newsgroups are joined in one cluster labeled “card” in which also small parts of sci.electronics, misc.forsale and comp.os.ms-windows.misc are mapped.

Figure 5.14 shows the map clustered into 25 clusters and labeled with one label each. Like in the first map there is a large cluster approximately in the middle containing a lot of different data. It consists of two separate areas: The pink area labeled “list” and the area manually labeled with two stars slightly to right of this area. This cluster contains a lot of short postings from all groups. This data is so diverse that it is not possible to find a suitable label.

The labels indicate the topic of the cluster well, just like in the first map. There are some labels which, on first sight, appear peculiar, but when looking at the postings mapped to that area they make perfect sense. “Gordon” is the label of the area covering postings from a person called Gordon, to Gordon or quoting Gordon. Example postings can be read in the Appendix. The area labeled “Koresh” contains mainly postings from the newsgroup talk.politics.guns dealing with the raid of the church of the religious leader David Koresh.

Figure 5.15 shows 15 steps of the clustering including one label for each cluster. Technically, the clustering process works bottom-down, i.e., starting from one cluster for each unit and merging the clusters until only one is left (agglomerative clustering). For presentation purposes, these steps are reversed in the figure. Although the new labels calculated “from

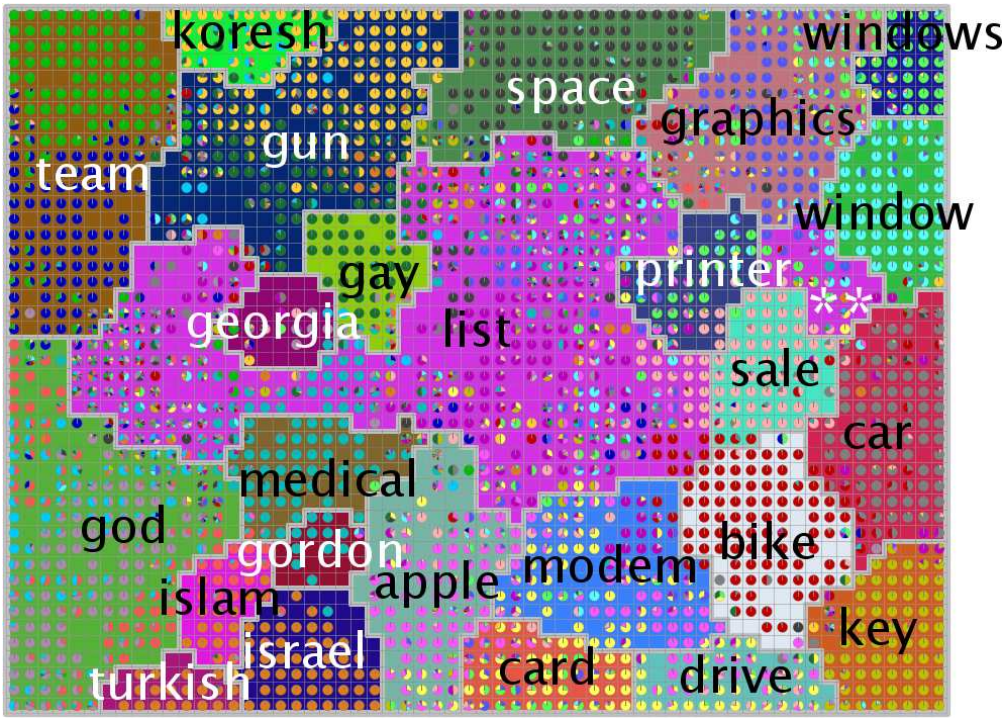



Figure 5.14: Map 2: 25 Clusters

scratch" whenever a new cluster is created (by merging two lower level cluster), it is often the case that a the new labels are similar to the labels of one of the subclusters, creating the impression that the label is "inherited" from one subcluster.

## 5.4 Map 3

The creation process of this map is somewhat different from the one of the maps before. From the header information in the postings, only the "Subject" and the "From" line have been taken. Furthermore, Porter's stemming algorithm was used to remove prefixes and suffixes to obtain word stems. The stemming algorithm is described in detail in [Por80].

In addition to the omitted words from 5.2, e-mail addresses were removed from the list of words because they are usually redundant with the name of the author of the posting and lower the quality of the labels.

The trained SOM is shown in Figure 5.16. It has the same size and was trained with the same amount of iterations as Map 1.

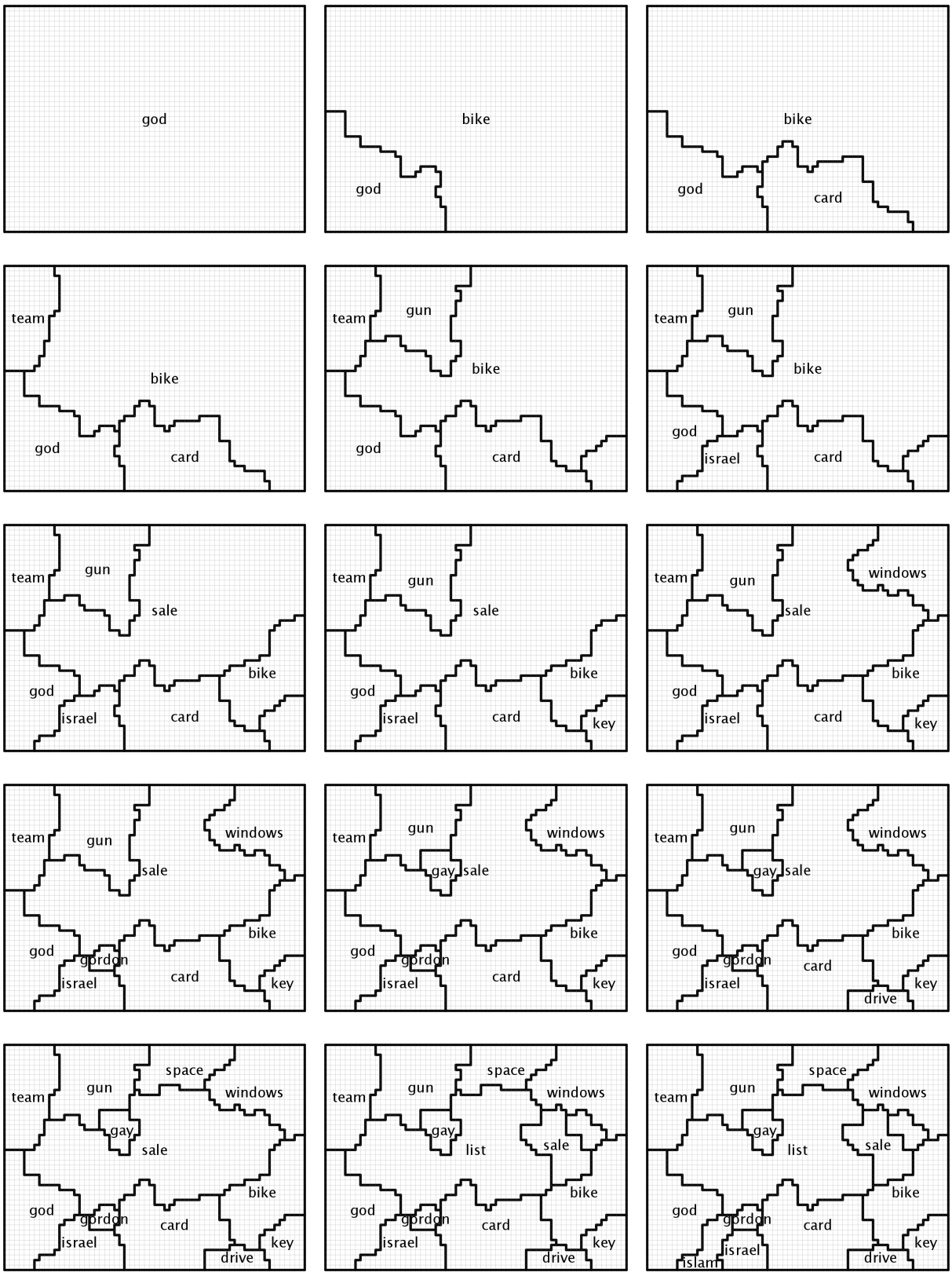



Figure 5.15: Map 2: 1 - 15 Clusters With Labels

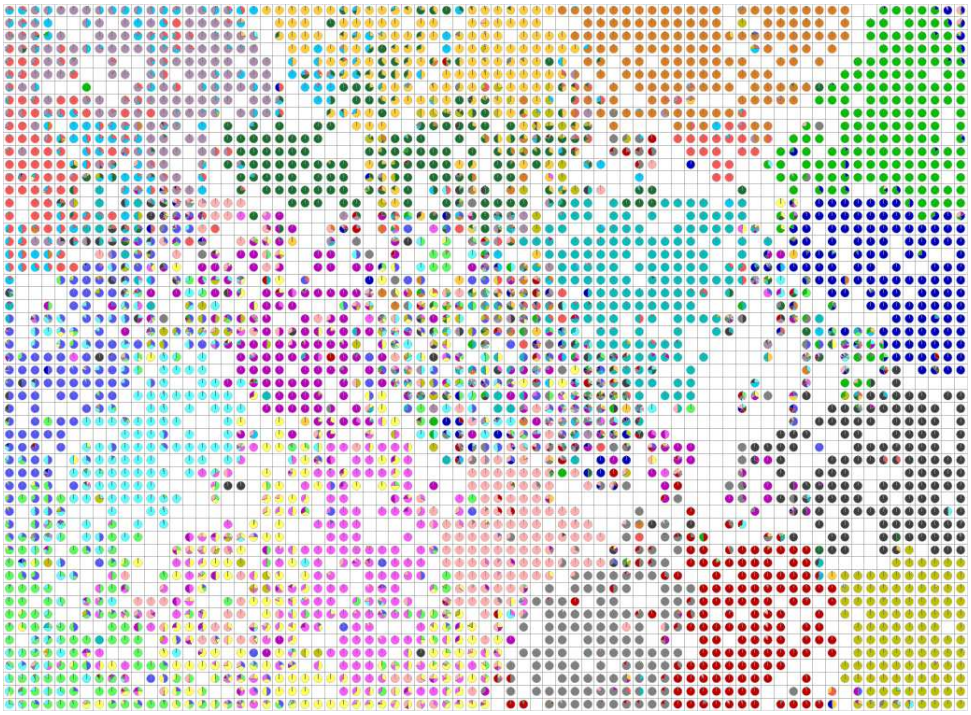


Figure 5.16: Map 3: 75x55 (With Stemming)

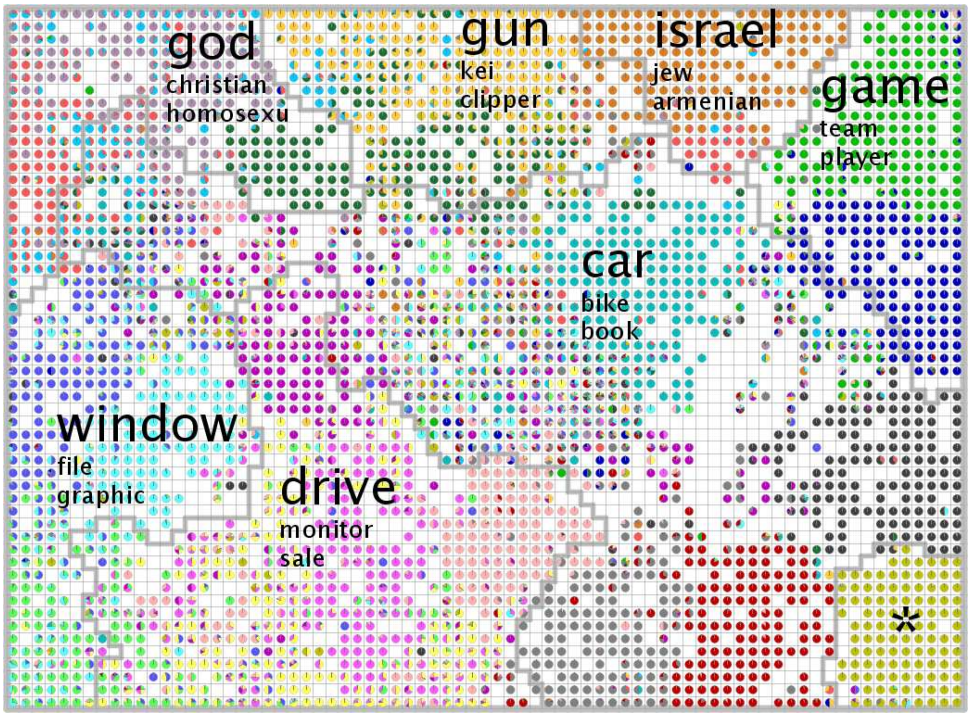



Figure 5.17: Map 3: 7 Clusters With 3 Labels

In Figure 5.17 the map is clustered into seven clusters. Even though the classes lie on different positions on the map there are a lot of similarities to the first two maps divided in seven clusters. Again the two sports newsgroups are in one cluster and there is a cluster containing religious topics. As in the second map, sci.crypt is put in one cluster together with talk.politics.guns and talk.politics.misc although they are not close on this map.

The map in Figure 5.18 shows 15 clusters with one label per cluster. The area labeled with a "*" belongs to the cluster labeled "mail". There are a lot of labels similar to the first two maps and, as before, the labels suit the underlying data.

The maps in Figure 5.19 and 5.20 do not show the class information; nevertheless, it is possible to determine where a lot of the newsgroups are located by looking at the labels.

Figure 5.20 shows nine colored clusters with labels and in addition 67 clusters with smaller labels. The cluster manually labeled "*" is part of the cluster labeled "david". As a result of the stemming algorithm, words ending with an *y* now end with an *i*, for example the labels "kei" or "batteri". There are also some labels where obviously the suffixes of the original words have been removed, as in the labels "imag" or "insur". Those labels can of course be manually edited to display full words, but this is not done here to show the terms based on which the map has been created.

The large cluster in the middle labeled "book" contains a lot of small clusters of which only a few have meaningful labels. The small clusters labeled "insur" and "doctor" suggest that they contain postings from the sci.med newsgroup and the cluster label "orbit" relates to the newsgroup sci.space. The labels containing names such as "gordon", "david" or "bill" do not help in identifying the underlying topics in those areas of the map. Names cannot be automatically removed as common names like Mark or Bill are also a verb or noun respectively. Table 5.3 shows some more labels for some of these small clusters.

Looking at one of the newsgroup postings inside the cluster labeled "gordon" reveals that most labels come from a signature used by one of the frequent posters of sci.med:

```
Gordon Banks  N3JXP      | "Skepticism is the chastity of the intellect, and
geb@cadre.dsl.pitt.edu   |  it is shameful to surrender it too soon."
```

Only the sixth label relates to a topic inside this cluster.

The cluster labeled "henri" can be linked to the cluster labeled "orbit" when looking at both clusters' next labels. The labels "gamma", "rai", "detector" and "burst" for the "pat"-cluster

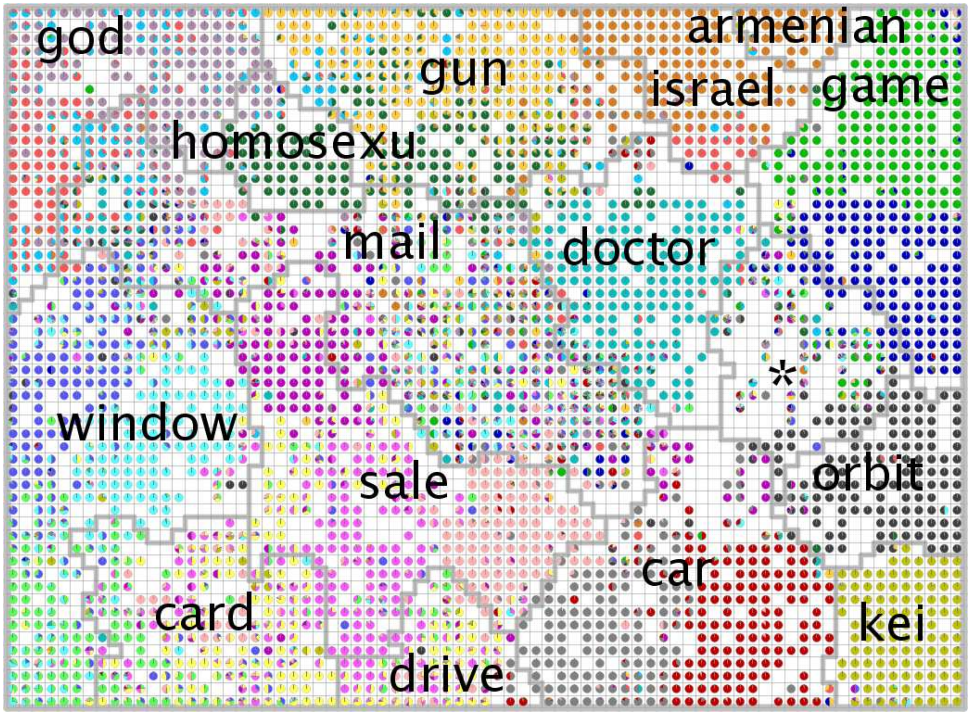



Figure 5.18: Map 3: 15 Clusters With Labels


Figure 5.19: Map 3: 22 Clusters With Labels

| Label | Further Labels |
|---|---|
| gordon | bank, n3jxp, chastiti, intellect, pysician |
| henri | moon, spencer, sky, billion, launch |
| orbit | space, launch mission, shuttl, satellit |
| pat | gamma, rai, max, detector, burst |
| mail | address, info, phone, internet, pleas |
| bill | post, group, newsgroup, discuss, faq |

Table 5.3: Map 3: Some Further Labels

reveal that the topics of this cluster might also be similar.

The map shown in Figure 5.20 is not yet well suited for presentation purposes. To enhance the comprehensibility of the map some labels are manually edited, which is shown in Figure 5.21. Word endings are edited or added to make the words more easily understandable, for example the label "imag" was changed to "image". From Figure 5.19 we know that in the cluster with the label "car" lie two sub-clusters, one dealing with cars, the other one dealing with bikes. To make this fact visible the label is changed to "car & bike". The cluster in the lower left-hand corner is labeled "graphic & windows" because it contains the groups dealing with computer graphics, the X Window System and the operating system Windows. We know from the class information that the large cluster labeled "game" contains the newsgroups dealing with sports, so the label is adapted accordingly. For cluster containing mainly the newsgroup sci.crypt the label is changed from "kei" to "crypt" and the cluster containing middle east related topics is renamed to "mid.east", and so on. The large cluster in the middle containing various topics has three labels in Figure 5.21: "medicine" where the newsgroup sci.med lies, "space" for the newsgroup sci.space and "misc." for all the miscellaneous topics arranged in this cluster.

## 5.5 Interpretation

There are a few results that can be derived from the clustering of the maps: Usually postings whose input vectors are close (i.e. containing similar words in similar frequencies) cover similar topics. Therefore, we can assume that postings that are (1) spatially close on the SOM and/or (2) clustered into the same cluster also concern related topics. Examples (see Table 5.2):

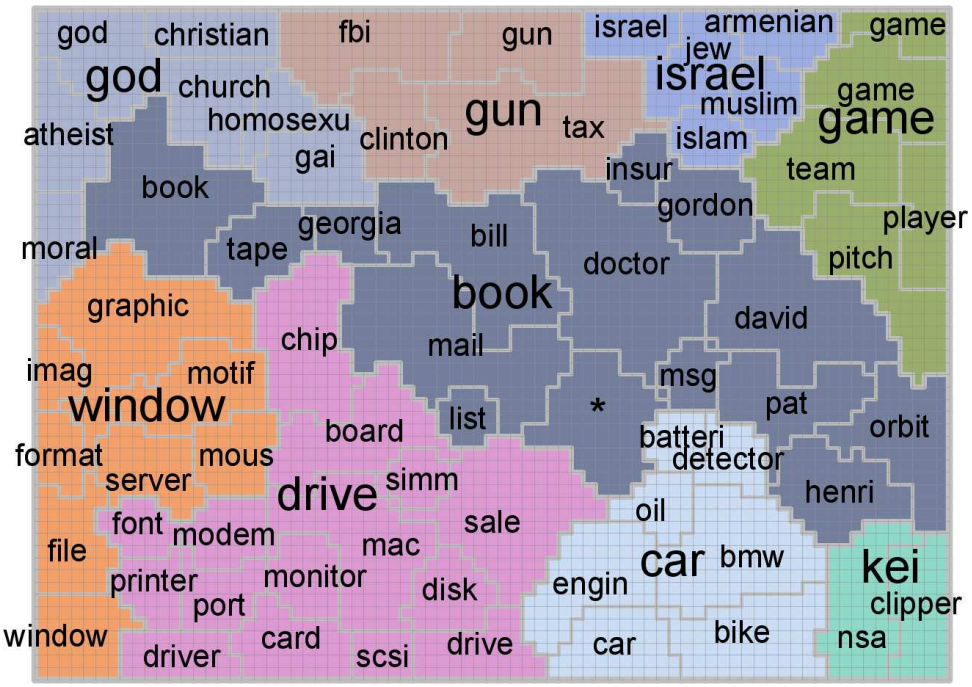



Figure 5.20: Map 3: 9 Clusters With Color and 67 Clusters (Both Labeled)

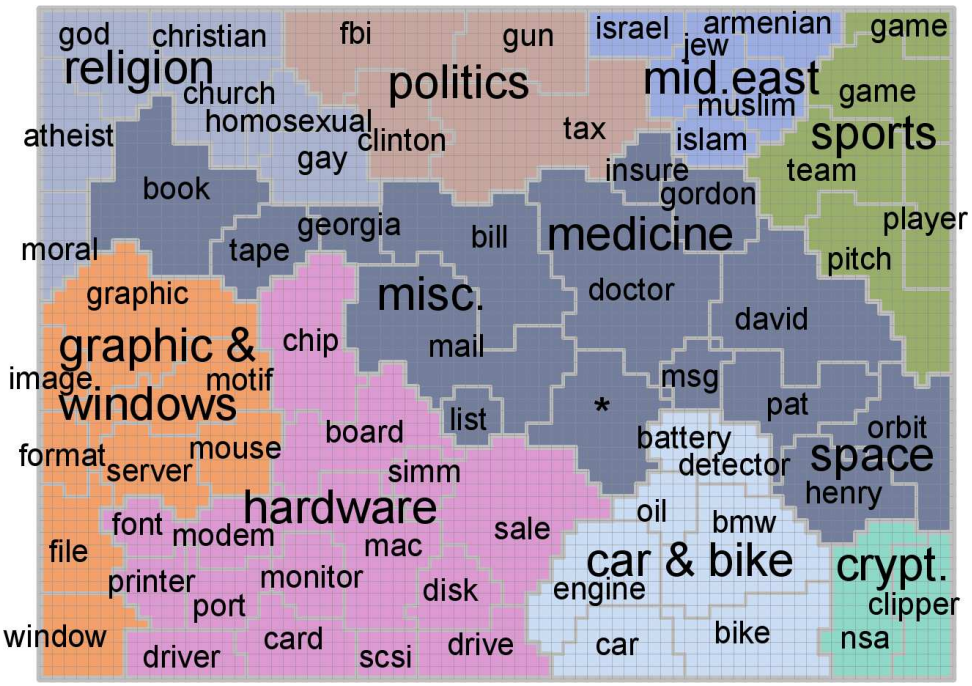



Figure 5.21: Map 3: 9 Clusters With Color and 67 Clusters (Edited Labels)

- Not surprisingly, postings from alt.atheism, talk.religion.misc and soc.religion.christian are often spatially close on the SOM and grouped into the same cluster. Note that the (hierarchical) newsgroup naming scheme does not cover this similarity, as all three newsgroups belong to different top-level hierarchies.
- Another example is the group sci.crypt. Especially the first two maps show that the sci.crypt topics are closer to the computer and politics groups than to the other science-related (sci.*) newsgroups. The clustering in Figure 5.20 shows an example for one of these similarities in more detail: The cryptography cluster (labeled "kei", stemmed from "key") in the lower-right hand corner contains a sub-cluster "nsa", whereas the politically-oriented cluster at the top (labeled "gun") contains a sub-cluster named "fbi".
- The SOMs show the major topics being discussed in talk.politics.mideast by splitting it into clusters labeled "israel", "turkish" or "armenian" and "islam".

However, there are also examples of areas where the topics might be different although the words are the same. The word "drive" can be used, for example, in the meaning of "hard disk drive" or in the meaning of "to drive a car or motorcycle". As the semantics of words are not taken into consideration, this effect can have an undesired impact on the placements of postings on the SOM. An example can be seen in Figure 5.20: The cluster labeled "drive" (bottom, middle) is adjacent to "disk" and "scsi", but also to "car" and "engin[e]".

From the labels one can identify topics which are discussed in a newsgroup. For example in Figure 5.20 the cluster in the top left-hand corner consisting of the religion related newsgroups contains topics such as god, moral or church and also homosexuality. The latter is in an area overlapping with politics related newsgroups.

## 5.6 Summary

In this chapter a data set consisting of around 20 000 postings from 20 different newsgroups has been used to present the labeling and clustering algorithms. Therefore, three SOMs have been generated and were presented. There are several ways how a map can be viewed using clusters and labels and this allows different possibilities to read the map.

Every map has been used to emphasize a different aspect: On the first map we compared the output of the clustering with the U-Matrix and SDH visualizations, detailed clustering steps were shown for the second map in Figure 5.15. The third map was used to show a way to make the map look like a political map.

Finally the previous section outlines the lessons learned from the experiments.

# Chapter 6

# Conclusion

In this thesis an approach to make the exploration of data on a SOM easier for the user has been introduced: The SOM should become as intuitively readable as a cartographic map allowing efficient information retrieval. An emphasis has been put on textual documents to demonstrate the use of this technique in the domain of knowledge management.

In Chapter 2 some preliminary knowledge has been explained. The SOM was introduced as well as methods for labeling SOM units, and algorithms for clustering were discussed. Furthermore, the software used in this thesis, the SOMToolbox, was presented with some features and visualizations which were used in the remainder of the thesis.

Chapter 3 described the implementation and design issues of clustering a SOM and how to display it in a user friendly way. The clustering algorithms were compared to each other by applying them to manually generated data sets. This has shown that Ward's linkage results in an appropriate partitioning of the map in most cases whereas single linkage only has advantages in very specific cases, i.e. the dataset consisting of two rings in our examples. Moreover, in Section 3.2 some additional color palettes based upon hypsometric tinting of cartographic maps were introduced. As the clustering alone does not reveal information about the content of the clusters, the second part of the implementation covers the labeling of the clusters. Chapter 4 explains how the labels are generated and placed on the map.

Finally, the implemented methods were applied to a dataset consisting of 20 000 newsgroup postings in Chapter 5. Three different maps have been generated to demonstrate some possible ways to interact with the SOM. The first map was used to compare the automatically generated

labels with the newsgroups that are mapped to the cluster. Moreover other visualizations have been applied to the map and displayed together with the clustering to compare the results. In addition to presenting several clustered and labeled versions of the second map, a series of images is presented adding one labeled cluster in each step to exemplify how the user can explore the map. The last map was used to show the SOM without the class information but only with labeled clusters. Figure 5.20 showed how the structure and relation of the data on the map can be revealed when displaying more than one layer of cluster information.

Of course there are a lot of improvements and new features which can be implemented in the future to aid the user in working with the program, for example:

- additional color schemes for various applications
- a simple inteface for the user to adapt automatically chosen values (e.g. color for a cluster, border color, default font size,...)
- show a different number of clusters depending on zooming level
- adapt font sizes to zooming level
- automatic text summarization for the documents inside each cluster
- automatically determine a number of clusters to display

We would like to point out that information retrieval is not the only possible application of the clustering, coloring and labeling algorithms presented in this work. As, in our implementation, the automated labeling algorithm is extended by features allowing labels to be added, modified and removed, our tool can be used to create visually appealing and easy to understand graphical representations of the input data. This way, a SOM containing a lot of textual data can be used, for example, to reveal underlying cluster structures, to visualize and explore the contained information and to present the contents interactively.

This work has contributetd to taking the SOM one step further away from an abstract scientific analysis method towards a tool that can be used by end-users in real life scenarios: A manager can get an overview of the documents created by different working groups, a scientist

can cluster a large set of papers or an author can browse for his works in the context of other documents.

These are but a few examples of application domains for this work. We believe that our contribution not only provides an added value to existing SOM tools but opens up new usage scenarios.

# Appendix A

# Example Newsgroup Postings

## From misc.forsale

### Example 1

```
Subject: Intel i486DX-33 CPU:  $300 + shipping
From: awlin@eagle.wesleyan.edu
Organization: Wesleyan University
Nntp-Posting-Host: wesleyan.edu
Lines: 11

***** FORSALE *****

Intel i486DX-33 CPU

Price: $300

Must sell immediately.

Andie Wei-Ku Lin
----------------
awlin@eagle.wesleyan.edu
```

### Example 2

```
From: antonio@arezzo.oas.olivetti.com (Antonio Maiuolo)
Subject: HONDA ACCORD (82) FOR SALE
Organization: Olivetti Research California
Lines: 27
Distribution: world
Reply-To: antonio@arezzo.oas.olivetti.com (Antonio Maiuolo)
NNTP-Posting-Host: arezzo.oas.olivetti.com
Keywords: honda, sale, 82
```

```
Honda Accord_4_Cyl. LX Hatchback  3D   5 Speed   ’82    126,000 Miles

        with :  AM/FM Stereo Cassette

                Clutch          ’89
                Alternator      ’90
                Battery         ’93
                Carburator      ’93 Rebuilt

        Registration expires    FEB ’94


        Asking price            : $ 2150.00


        Please call 408-366-3570 (Fulvio)

        Location : 20300 Stevens Creek Blv.  Cupertino CA
```

**Example 3**

```
From: walter@psg.com (Walter Morales)
Subject: Nitendo game wanted
Organization: Pacific Systems Group, Portland Oregon US
Lines: 24

Hi,
I am one those uncles that try to please my nephews whenever possible,
so.. they have asked me to find them some Nitendo games, no, it is
not for the super nitendo.. it is for whatever model came prior to
that.

Since they are overseas, I will first ask them if they already have the
games you would have to offer me. Please send me a list, or whatever and
the price you are asking so I can send to my nephews and find out what
they have and what they want.. so bare with me, I will respond, but it
will take me a while.

Thanks,
Walter
walter@psg.com


Please respond directly.

--
  _____________________________________________________________________
 /                                          Portland, Oregon USA      \
 |  WALTER T. MORALES                       45 31 25 N 122 40 30 W    |
 |  internet: walter@rain.com               Pop. 366383               |
```

## From different newsgrous but inside the same cluster

**rec.sports.hockey**

```
Subject: NCAA finals...Winner????
From: ktgeiss@miavx1.acs.muohio.edu
Organization: Miami University Academic Computer Service"
Lines: 1

Lake State/Maine in finals...WHO WON?   Please post.
```

### misc.forsale

```
From: khiet@crystallizer.ecn.purdue.edu (Peter Thanh Khiet Vu)
Subject: WANTED:  FUTON
Keywords: WANTED:  FUTON
Organization: Purdue University Engineering Computer Network
Lines: 5

 I am looking for a large futon and frame.

call Peter 495-2056
or e-mail me "khiet@cn.ecn"
```

### sci.crypt

```
From: morgan@engr.uky.edu (Wes Morgan)
Subject: Re: I have seen the lobby, and it is us
Organization: University of Kentucky Engineering Computing Center
Lines: 61

ns111310@LANCE.ColoState.Edu (Nathaniel Sammons) wrote:
>2) If some kind soul out there would write a letter, and upload it to
>the net, everyone could capture it, print it out, and snail-mail it
>out to their local congressional critter.
>
>BTW>> I'm working on one.

You should realize that form letters are the *worst* way to influence
your congresscritters; exact copies are routinely placed on the lowest
rung of the opinion ladder.

If you want to write (and I think you should!), take the time to really
*write* a letter.  Things to emphasize:

- It's been said that Usenet is available to the 'technical
  elite', i.e. the techies at corporate sites and universi-
  ties.  Emphasize that you are part of the group that will
  be making/developing/using Clinton's 'data superhighway.'

- Explain how you are intimately familiar with both computing
  and data communications. (if this is the case; don't call your-
  self an 'expert' after wiring in a 1200 bps modem.)   This
  will distinguish your letter from the random flamers.  Don't
  turn it into braggadocio; just tell them that you know the
  technical sides of the issue.

- Don't overdo jargon and gobbledygook.  Remember, your letter
  will be first read (in all likelihood) by a staffer who may
  even be a college student; if they don't understand it, your
```

```
  views won't even make the 'running total' sheets.

- Be concise; don't ramble.  Rants are *definitely* out of place.
  Cite references, if necessary, but only use "accepted" references
  like academic journals.  "My neighbor Jim" is *not* a real refer-
  ence.  8)

- Unfortunately, very few Congresscritters *really* understand
  electronic communications.  Encourage them to pick up access
  to Compuserve, America Online, or one of the Free-Nets.  Offer
  to send them samples.  If you are in a position to do so, offer
  them (or their staffers back in the home state) access to your
  systems.  Offer to give a demonstration the next time they're in
  town.  Your offer to get *personally* involved in helping them
  *will* give your opinions more credence.

- In addition to sending mail to your representatives, send mail
  to the members of the committee (or subcommittee) that is dealing
  with the issue.  If your Congresscritter isn't on the committee,
  they can't be of much help until the matter comes to the floor.

--Wes

ps> I'd suggest drawing analogies between digital communication and the
    more traditional media, but Usenet doesn't have a decent track record
    in the analogy department.  8)

--
Rachel Elizabeth Morgan  --  4/13/93, 7:00 am    | Oh yeah, I can be reached as
9 pounds 4 ounces (despite coming 3 weeks early) |     morgan@engr.uky.edu
                                                 |----------------------------
To netters who gave constant encouragement and prayer - thanks for everything!
```

## From "gordon" Cluster

### Posting from Gordon

```
From: geb@cs.pitt.edu (Gordon Banks)
Subject: Neurasthenia
Reply-To: geb@cs.pitt.edu (Gordon Banks)
Organization: Univ. of Pittsburgh Computer Science
Lines: 15

In article <1993Apr21.174553.812@spdcc.com> dyer@spdcc.com (Steve Dyer) writes:

>responds well, if you're not otherwise immunocompromised.  Noring's
>anal-retentive idee fixe on having a fungal infection in his sinuses
>is not even in the same category here, nor are these walking neurasthenics
>who are convinced they have "candida" from reading a quack book.

Speaking of which, has anyone else been impressed with how much the
descriptions of neurasthenia published a century ago sound like CFS?

--
------------------------------------------------------------------------
```

```
Gordon Banks  N3JXP      | "Skepticism is the chastity of the intellect, and
geb@cadre.dsl.pitt.edu   |  it is shameful to surrender it too soon."
-------------------------------------------------------------------------------
```

### Reply to Gordon

```
From: mcg2@ns1.cc.lehigh.edu (Marc Gabriel)
Subject: Re: How to Diagnose Lyme... really
Organization: Lehigh University
Lines: 44
X-Newsreader: TIN [version 1.1 PL9]

Gordon Banks (geb@cs.pitt.edu) wrote:
: In article <1993Apr12.201056.20753@ns1.cc.lehigh.edu> mcg2@ns1.cc.lehigh.edu (
Marc Gabriel) writes:

: >Now, I'm not saying that culturing is the best way to diagnose; it's very
: >hard to culture Bb in most cases.  The point is that Dr. N has developed a
: >"feel" for what is and what isn't LD.  This comes from years of experience.
: >No serology can match that.  Unfortunately, some would call Dr. N a "quack"
: >and accuse him of trying to make a quick buck.
: >
: Why do you think he would be called a quack?  The quacks don't do cultures.
: They poo-poo doing more lab tests:  "this is Lyme, believe me, I've
: seen it many times.  The lab tests aren't accurate.  We'll treat it
: now."  Also, is Dr. N's practice almost exclusively devoted to treating
: Lyme patients?  I don't know *any* orthopedic surgeons who fit this
: pattern.  They are usually GPs.

No, he does not exclusively treat LD patients.  However, in some parts of the
country, you don't need to be known as an LD "specialist" to see a large
number of LD patients walk through your office.  Given the huge problem of
underdiagnosis, orthopedists encounter late manifestations of the disease just
about every day in their regular practices.  Dr. N. told me that last year,
he sent between 2 and 5 patients a week to the LD specialists... and he is not
the only orthopedists in the town.

Let's say that only 2 people per week actually have LD.  That means at the
*very minimum* 104 people in our town (and immediate area) develop late stage
manifestations of LD *every year*.  Add in the folks who were diagnosed by
neurologists, rheumatologists, GPs, etc, and you can see what kind of problem
we have.  No wonder just about everybody in town personally knows an LD
patient.

He refers most patients to LD specialists, but in extreme cases he puts the
patient on medication immediately to minimize the damage (in most cases, to
the knees).

Gordon is correct when he states that most LD specialists are GPs.

-Marc.
--
--
-------------------------------------------------------------------------
              Marc C. Gabriel        -  U.C. Box 545  -
```

```
            (215) 882-0138         Lehigh University
```

### Quoting Gordon's message including parts of the signature

```
From: paulson@tab00.larc.nasa.gov (Sharon Paulson)
Subject: Re: food-related seizures?
Organization: NASA Langley Research Center, Hampton VA, USA
Lines: 48
<C5uq9B.LrJ@toads.pgh.pa.us> <C5x3L0.3r8@athena.cs.uga.edu>
NNTP-Posting-Host: cmb00.larc.nasa.gov
In-reply-to: mcovingt@aisun3.ai.uga.edu's message of Fri, 23 Apr 1993 03:41:24 GMT

In article <C5x3L0.3r8@athena.cs.uga.edu> mcovingt@aisun3.ai.uga.edu
(Michael Covington) writes:

   Newsgroups: sci.med
   Path: news.larc.nasa.gov!saimiri.primate.wisc.edu!sdd.hp.com!elroy.jpl.nasa.gov!
swrinde!zaphod.mps.ohio-state.edu!howland.reston.ans.net!europa.eng.gtefsd.com!
emory!athena!aisun3.ai.uga.edu!mcovingt
   From: mcovingt@aisun3.ai.uga.edu (Michael Covington)
   Sender: usenet@athena.cs.uga.edu
   Nntp-Posting-Host: aisun3.ai.uga.edu
   Organization: AI Programs, University of Georgia, Athens
   References: <PAULSON.93Apr19081647@cmb00.larc.nasa.gov> <116305@bu.edu>
<C5uq9B.LrJ@toads.pgh.pa.us>
   Date: Fri, 23 Apr 1993 03:41:24 GMT
   Lines: 27

   In article <C5uq9B.LrJ@toads.pgh.pa.us> geb@cs.pitt.edu (Gordon Banks) writes:
   >In article <116305@bu.edu> dozonoff@bu.edu (david ozonoff) writes:
   >>
   >>Many of these cereals are corn-based. After your post I looked in the
   >>literature and located two articles that implicated corn (contains
   >>tryptophan) and seizures. The idea is that corn in the diet might
   >>potentiate an already existing or latent seizure disorder, not cause it.
   >>Check to see if the two Kellog cereals are corn based. I'd be interested.
   >
   >Years ago when I was an intern, an obese young woman was brought into
   >the ER comatose after having been reported to have grand mal seizures
   >why attending a "corn festival".  We pumped her stomach and obtained
   >what seemed like a couple of liters of corn, much of it intact kernals.
   >After a few hours she woke up and was fine.  I was tempted to sign her out as
   >"acute corn intoxication."
   >------------------------------------------------------------------------
   >Gordon Banks  N3JXP      | "Skepticism is the chastity of the intellect, and

   How about contaminants on the corn, e.g. aflatoxin???



   --
   :-  Michael A. Covington, Associate Research Scientist        :    *****
   :-  Artificial Intelligence Programs      mcovingt@ai.uga.edu :  *********
```

```
   :-  The University of Georgia              phone 706 542-0358 :   *  *  *
   :-  Athens, Georgia 30602-7415 U.S.A.     amateur radio N4TMI :  ** *** **  <><

What is aflatoxin?

Sharon
--
Sharon Paulson                       s.s.paulson@larc.nasa.gov
NASA Langley Research Center
Bldg. 1192D, Mailstop 156            Work: (804) 864-2241
Hampton, Virginia.  23681            Home: (804) 596-2362
```

### Another Gordon in the same Cluster

```
From: glang@slee01.srl.ford.com (Gordon Lang)
Subject: Please help find video hardware
Article-I.D.: fmsrl7.1pqf9oINN88e
Organization: Ford Motor Company Research Laboratory
Lines: 19
NNTP-Posting-Host: slee01.srl.ford.com
X-Newsreader: Tin 1.1 PL5

[ Article crossposted from comp.sys.hp ]
[ Author was Gordon Lang ]
[ Posted on 5 Apr 1993 23:25:27 GMT ]

[ Article crossposted from comp.sys.ibm.pc.hardware ]
[ Author was Gordon Lang ]
[ Posted on 5 Apr 1993 23:19:01 GMT ]

I need a device (either an ISA board or a subsystem) which will
take two RGB video signals and combine them according to a template.
The template can be as simple as a rectangular window with signal
one being used for the interior and signal two for the exterior.
But I beleive fancier harware may also exist which I do not want
to exclude from my search.  I know this sort of hardware exists
for NTSC, etc. but I need it for RGB.

Please email and or post any leads....

Gordon Lang (glang@smail.srl.ford.com  -or-  glang@holo6.srl.ford.com)
```

# List of Figures

# Bibliography


[20n] 20 newsgroups,. `http://people.csail.mit.edu/jrennie/20Newsgroups`. 15 Jan 2007.

[DNR05] M. Dittenbach, R. Neumayer, and A. Rauber. PlaySOM: An alternative approach to track selection and playlist generation in large music collections. In *Proceedings of the First International Workshop of the EU Network of Excellence DELOS on Audio-Visual Content and Information Visualization in Digital Libraries (AVIVDiLib 2005)*, pages 226–235, 2005.

[Eck80] Thomas Eckes. *Clusteranalysen*. Kohlhammer Standards Psychologie, 1980.

[FW91] Michael Formann and Frank Wagner. A packing problem with applications to lettering of maps. In *SCG '91: Proceedings of the seventh annual symposium on Computational geometry*, pages 281–288, New York, NY, USA, 1991. ACM Press.

[Hon99] Timo Honkela. Connectionist analysis and creation of context for natural language understanding and knowledge management. In *CONTEXT '99: Proceedings of the Second International and Interdisciplinary Conference on Modeling and Using Context*, pages 479–482, London, UK, 1999. Springer-Verlag.

[HSvdSS94] Ted Hesselroth, Kakali Sarkar, P. Patrick van der Smagt, and Klaus Schulten. Neural network control of a pneumatic robot arm. *IEEE Transactions on Systems, Man, and Cybernetics*, 24:28–37, 1994.

[Jai99] A.K. Jain. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, September 1999.

[KHLK96] Samuel Kaski, Timo Honkela, Krista Lagus, and Teuvo Kohonen. Creating an order in digital libraries with self-organizing maps. In *Proceedings of WCNN'96, World Congress on Neural Networks, September 15-18, San Diego, California*, pages 814–817. Lawrence Erlbaum and INNS Press, Mahwah, NJ, 1996.

[KMD91] S. Kieffer, V. Morellas, and M. Donath. Neural network learning of the inverse kinematic relationships for a robot arm. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2418–2425, April 1991.

[Koh82] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

[Koh01] Teuvo Kohonen. *Self-Organizing Maps*. Springer, 2001.

[LCN99] Chienting Lin, Hsinchun Chen, and Jay Nunamaker. Verifying the proximity hypothesis for self-organizing maps. In *HICSS '99: Proceedings of the Thirty-Second Annual Hawaii International Conference on System Sciences-Volume 1*, Washington, DC, USA, 1999. IEEE Computer Society.

[LK99] Krista Lagus and Samuel Kaski. Keyword selection method for characterizing text document maps. In *Ninth International Conference on Artificial Neural Networks, 1999. ICANN 99*, volume 1, pages 371–376, 1999.

[mapa] `http://mygeo.info`. 03 Feb 2006.

[mapb] `http://www.coha.dri.edu`. 15 Feb 2006.

[May04] Rudolf Mayer. Text mining with adaptive neural networks. Master's thesis, TU Wien, 2004.

[Mit97] Tom Mitchell. *Machine Learning*. McGraw Hill, 1997.

[MLR06] R. Mayer, T. Lidy, and A. Rauber. The map of Mozart. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, pages 351–352, Victoria, Canada, 2006.

[MMR05] Rudolf Mayer, Dieter Merkl, and Andreas Rauber. Mnemonic soms: Recognizable shapes for self-organizing maps. In Marie Cottrell, editor, *Proceedings of the Fifth Workshop on Self-Organizing Maps (WSOM'05)*, pages 131–138, Paris, France, September 5–8 2005.

[NDR05] Robert Neumayer, Michael Dittenbach, and Andreas Rauber. PlaySOM and PocketSOMPlayer: Alternative interfaces to large music collections. In *Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005)*, pages 618–623, London, UK, September 11-15 2005.

[NLR05] Robert Neumayer, Thomas Lidy, and Andreas Rauber. Content-based organization of digital audio collections. In *Proceedings of the 5th Open Workshop of MUSICNETWORK*, Vienna, Austria, July 4-5 2005.

[Por80] Martin Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[PRM02] E. Pampalk, A. Rauber, and D. Merkl. Using smoothed data histograms for cluster visualization in self-organizing maps. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN 2002)*, pages 871–876, 2002.

[Rau99] Andreas Rauber. LabelSOM: On the labeling of Self-Organizing Maps. In *Proceedings of IJCNN 99, International Joint Conference on Neural Networks*, volume 5, pages 3527–3532, 1999.

[RM99] Andreas Rauber and Dieter Merkl. Automatic labeling of Self-Organizing Maps: Making a treasure-map reveal its secrets. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 228–237, 1999.

[RM03] A. Rauber and D. Merkl. Text mining in the SOMLib digital library system: The representation of topics and genres. *Applied Intelligence*, 18(3):271–293, May-June 2003.

[Sku04] Andre Skupin. A picture from a thousand words. *Computing in Science and Engineering*, 6(5):84–88, 2004.

[sta] `http://www.statistics.com/resources/glossary/w/wardslnkg.php`. 09 Nov 2006.

[Ult92] A. Ultsch. Self-organizing neural networks for visualization and classification. In *Proceedings of the conference for information and classification*, 1992.

[War63] J.H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, March 1963.

[WWKS01] Frank Wagner, Alexander Wolff, V. Kapoor, and Tycho Strijk. Three rules suffice for good label placement. *Algorithmica*, 30(2):334–349, 2001.