Advanced Topics in Data Analytics

-

Self-Organizing Maps Visualizations

Andreas Rauber

Department of Software Technology and Interactive Systems

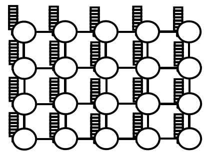
Vienna University of Technology

rauber@ifs.tuwien.ac.at

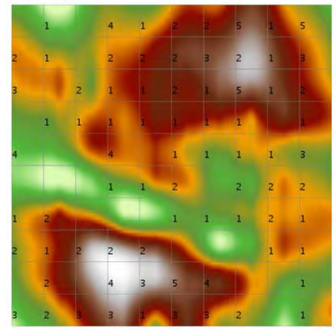
http://www.ifs.tuwien.ac.at/~andi

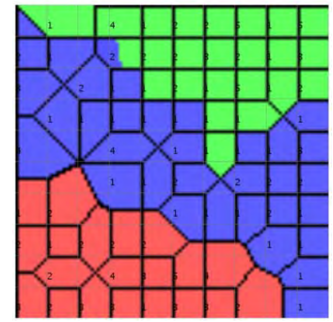
Outline

- SOM Basics

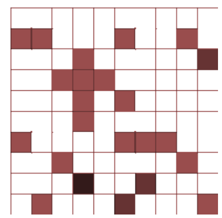


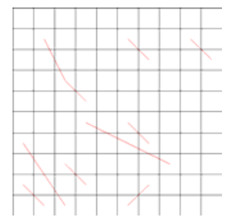
- **Visualizations**



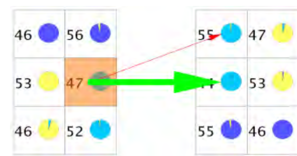


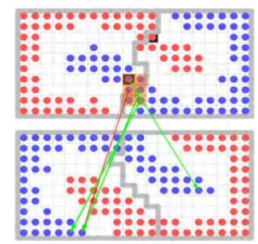
- SOM Quality Measures



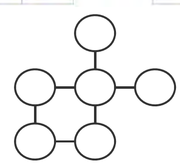


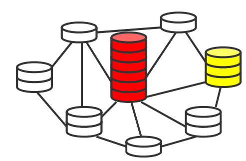
- SOM Comparison



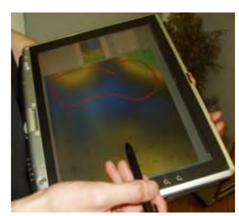


- Related Architectures and Methods





- Applications



SOM Visualizations

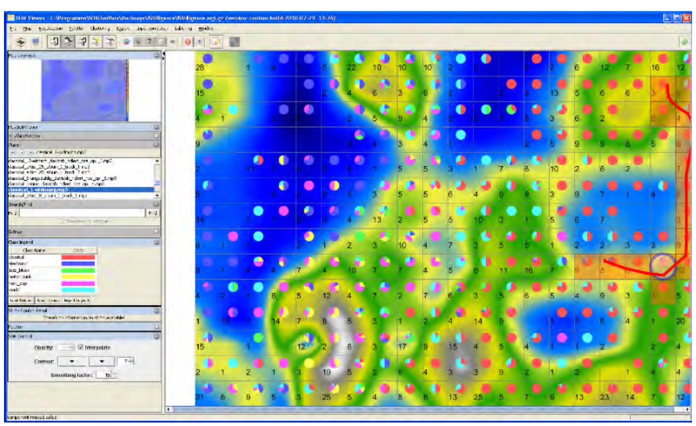
- SOM only basis for further analysis and applications
- Visualizations mostly based on map display
- Different ways of organizing visualizations:
  - Information used:
    - using only the codebook (weight vectors)
    - using codebook and data (with/without class info)
  - Type of visualization
    - coloring / background
    - overlay information
  - Type of information:
    - data analysis: density, topology, class distribution, quantization
    - quality analysis of SOM
- Important: combination of visualizations to be able to interpret information provided by the SOM!

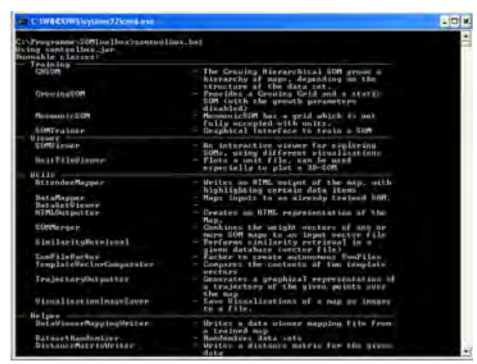
SOM Visualizations

- Juha Vesanto. SOM-based Data Visualization Methods. Intelligent Data Analysis 3(2):111-126. Elsevier Science.1999

Java SOMToolbox

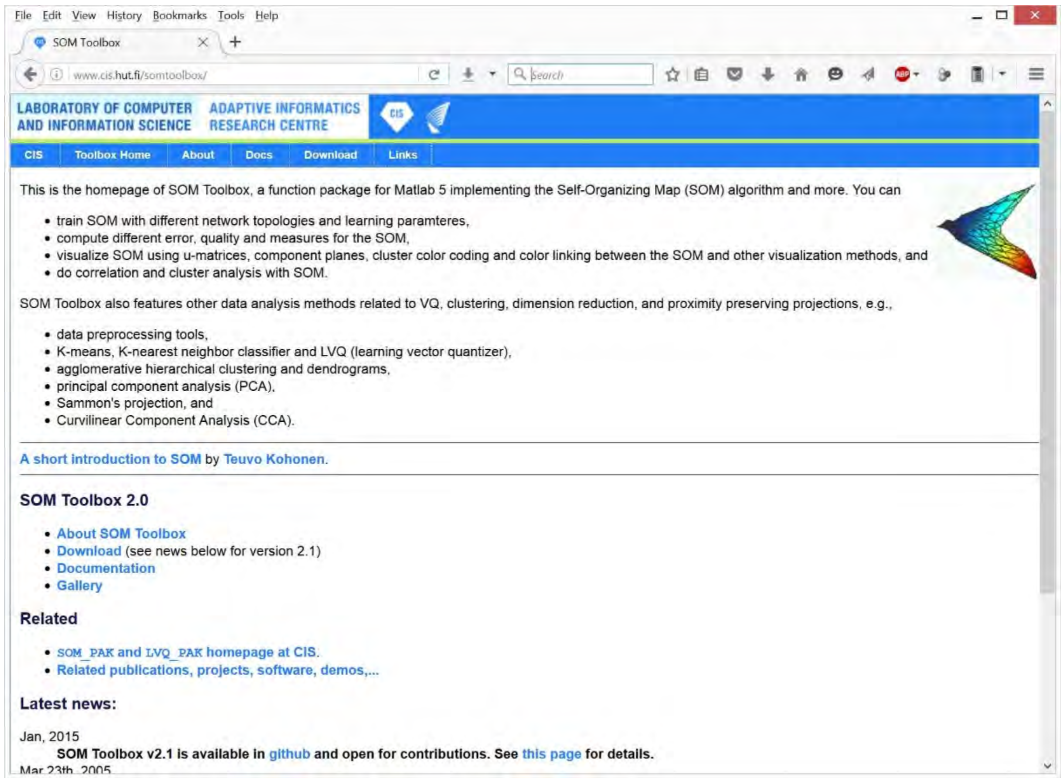
- Java, Apache License
  - http://www.ifs.tuwien.ac.at/dm/somtoolbox/download.html
- Graphical installer for Windows & Mac, package for Linux (Debian & Ubuntu)
  - Requires installed Java Runtime (JRE, http://java.sun.com)
- CL and graphical interface, Step-by-step guide:
  - http://www.ifs.tuwien.ac.at/dm/somtoolbox/somtoolbox-guide.html





Matlab SOMToolbox

- Toolbox for Matlab 5, Helsinki Univ. of Technology
- SOM training and visualization
- http://www.cis.hut.fi/somtoolbox/



Outline

- Overview of visualization types
- Visualizing the SOM
  - Codebook projection
  - Adaptive Coordinates
- Visualizations on the SOM
  - Textual information
  - Density
  - Distances
  - Class info
  - Attributes
  - Clustering of the SOM

Types of Visualizations

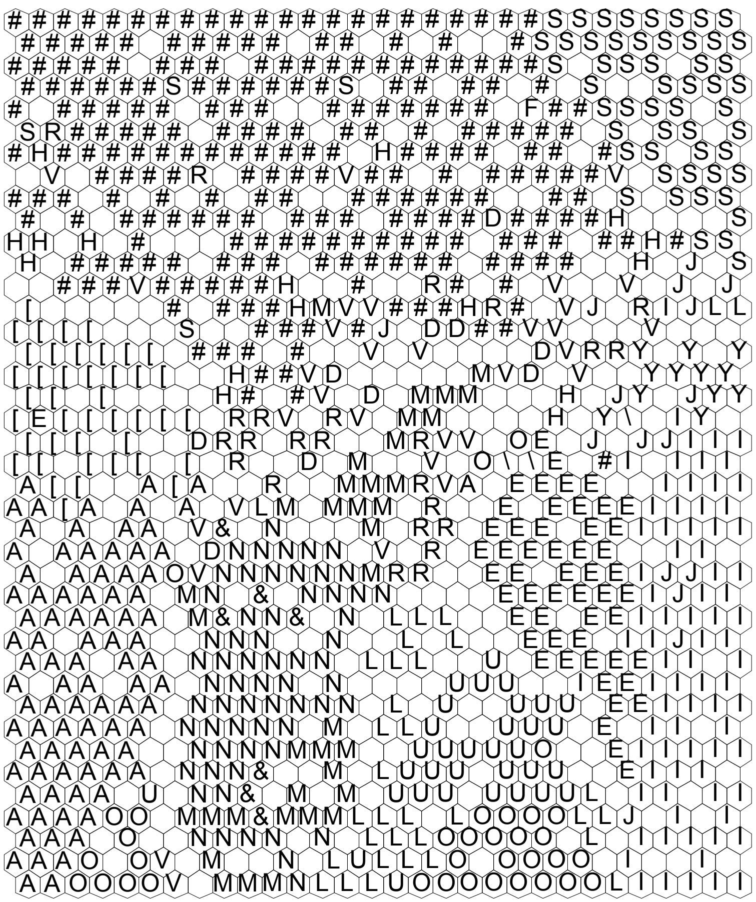
Overviw of visualization types

- Many possibilities to display information on the SOM
  - textual /numeric info
  - (colored) symbols, metaphor graphics
  - coloring the units
  - coloring via interpolating over units
  - lines, graphs as overlays
  - other (e.g. 3D worlds)
- Can be used to visualize a range of information
  - input vectors / data
  - classes
  - quality measures

Types of Visualizations

- Textual info on units
  - Number of vectors
  - Names of vectors
  - Class labels
  - Quality measures
- Can be combined as overlay on top of coloring

Types of Visualizations



Class-IDs on Units

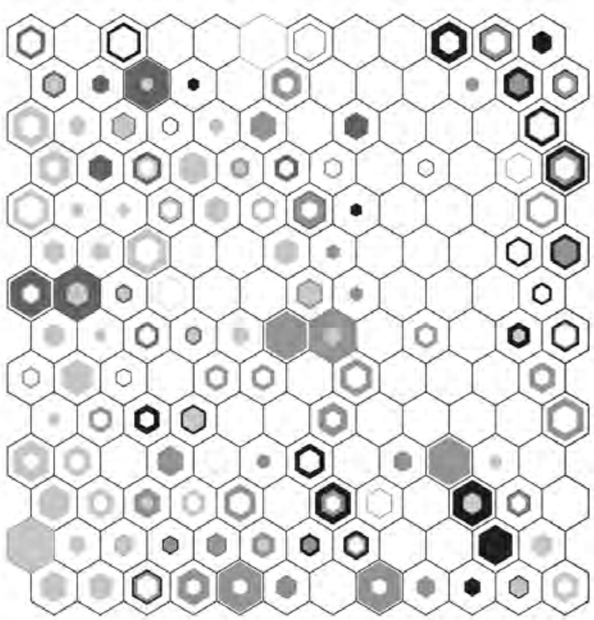
	1		4	1	2	2	5	1	5
2	1		2	2	2	3	2	1	3
3		2	1	1	2	1	5	1	2
	1	1	1	1	1	1	1		1
4			4		1	1	1	1	3
			1	1	2		2	2	2
1	2				1	1	1	2	1
2	1	2	2	2				1	1
	2		4	3	5	4			1
3	2	3	3	1	3	3	2		1

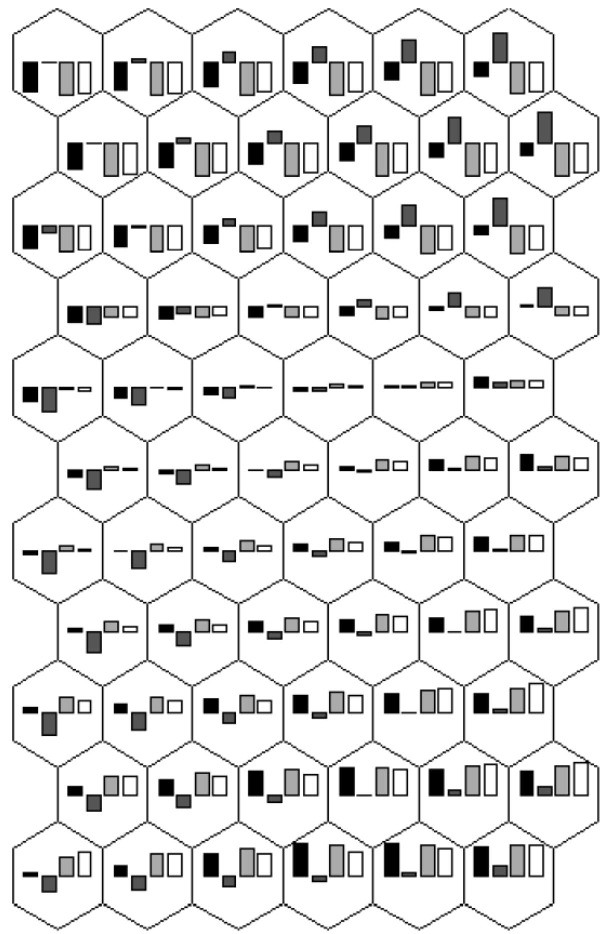
Number of Vectors on Units

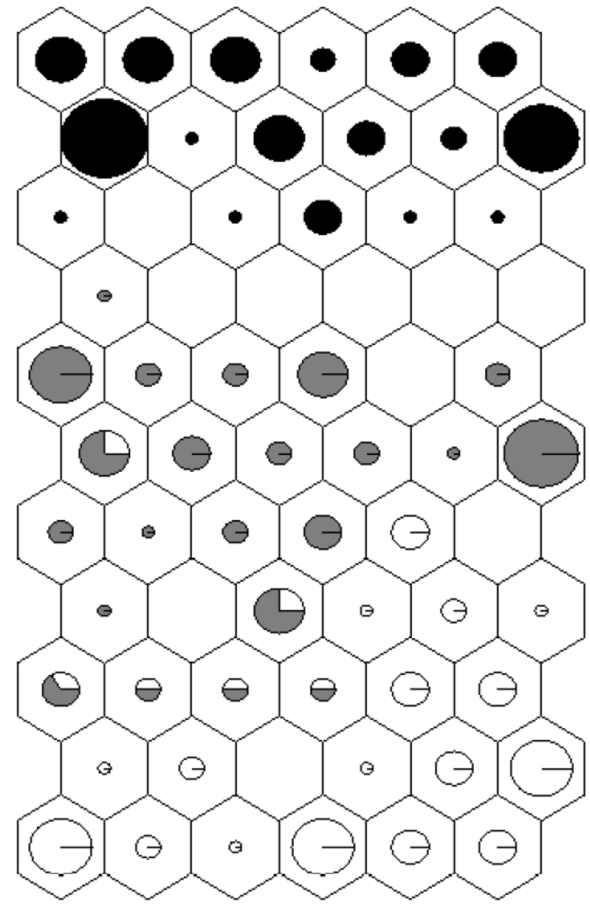
Types of Visualizations

- Symbols on units
- Information from codebook or input vector mapped onto graphical representations
- Diagrams or symbols (e.g. Chernoff Faces)
- Examples:
  - Class distribution
  - Attribute values

Types of Visualizations



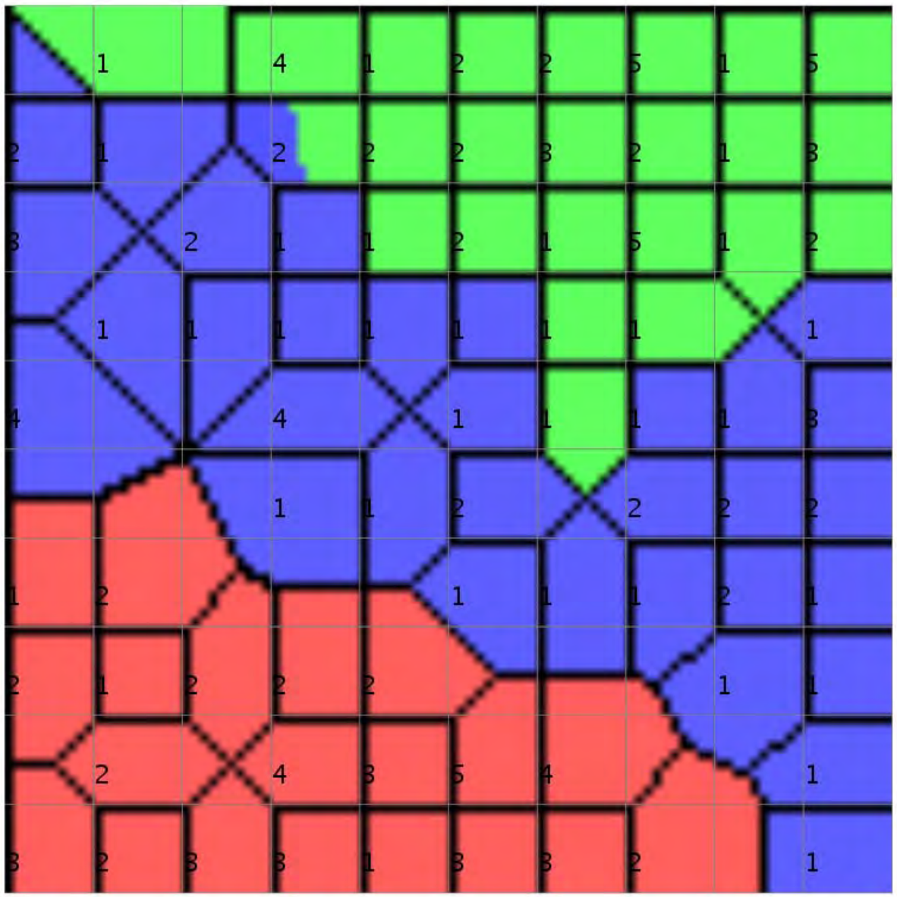


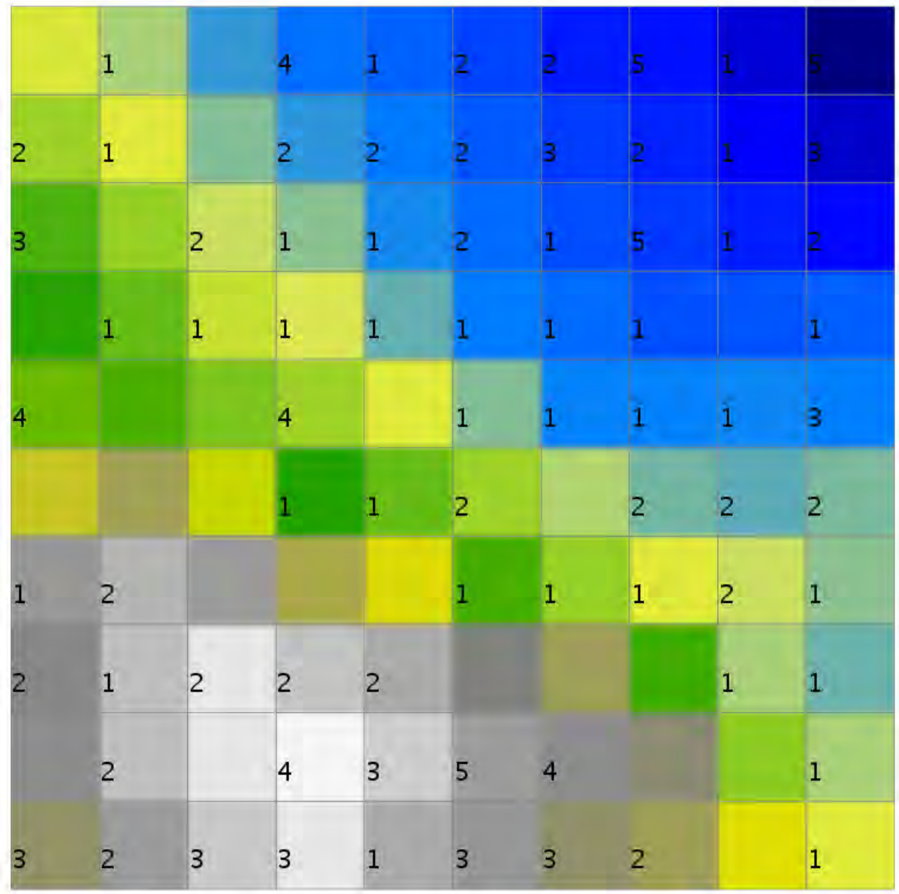


Types of Visualizations

- Coloring of units
- Numeric info as basis for coloring
- Different color palettes have huge impact on interpretation
  - choice of color range
  - gradients
  - inverting color space
- 2 types
  - coloring units
  - interpolating over numeric values on units
- Background image for text and graphs -> combination

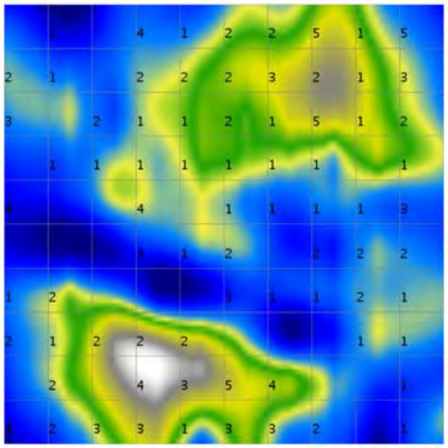
Types of Visualizations

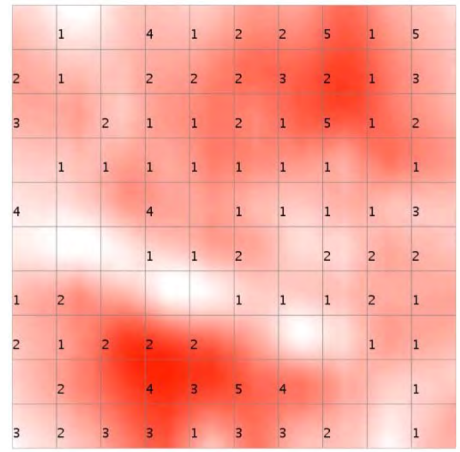


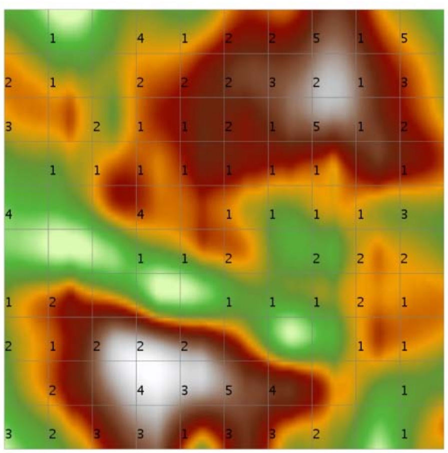


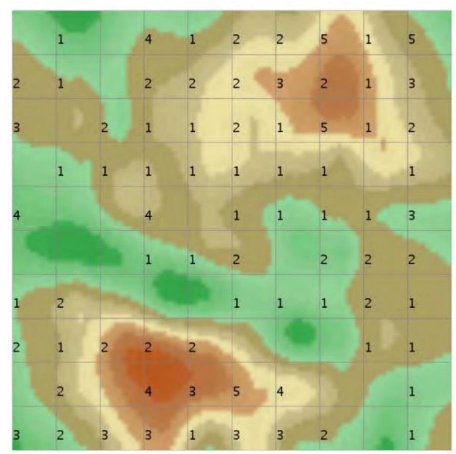
Types of Visualizations

- Interpolating over units: color palettes



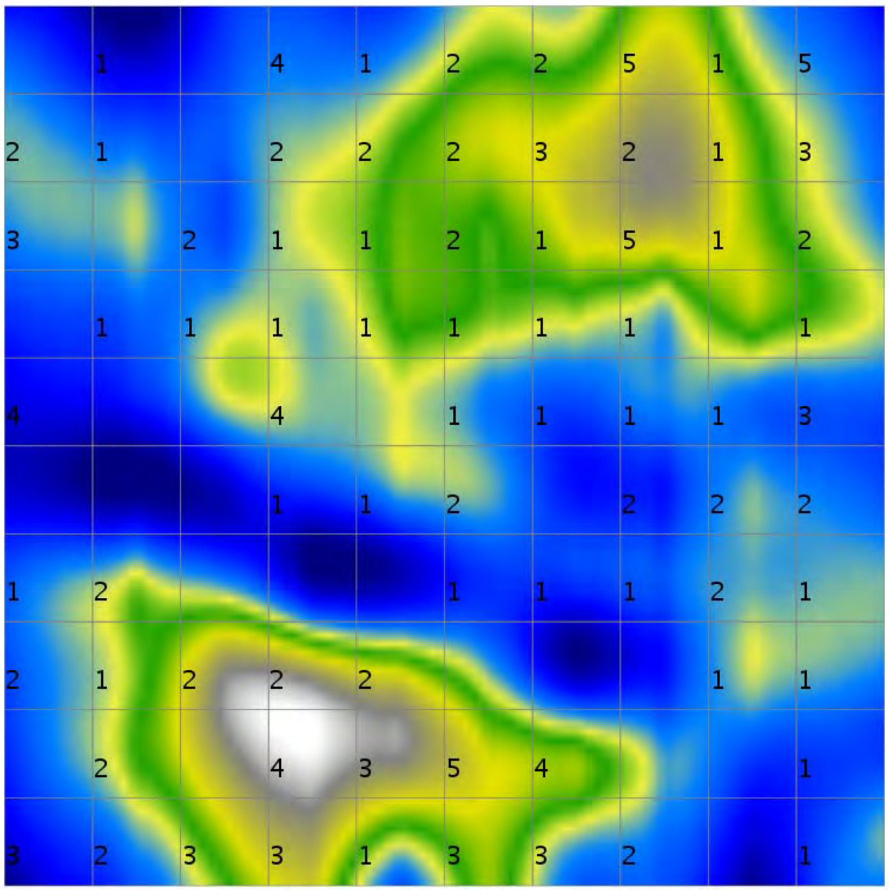


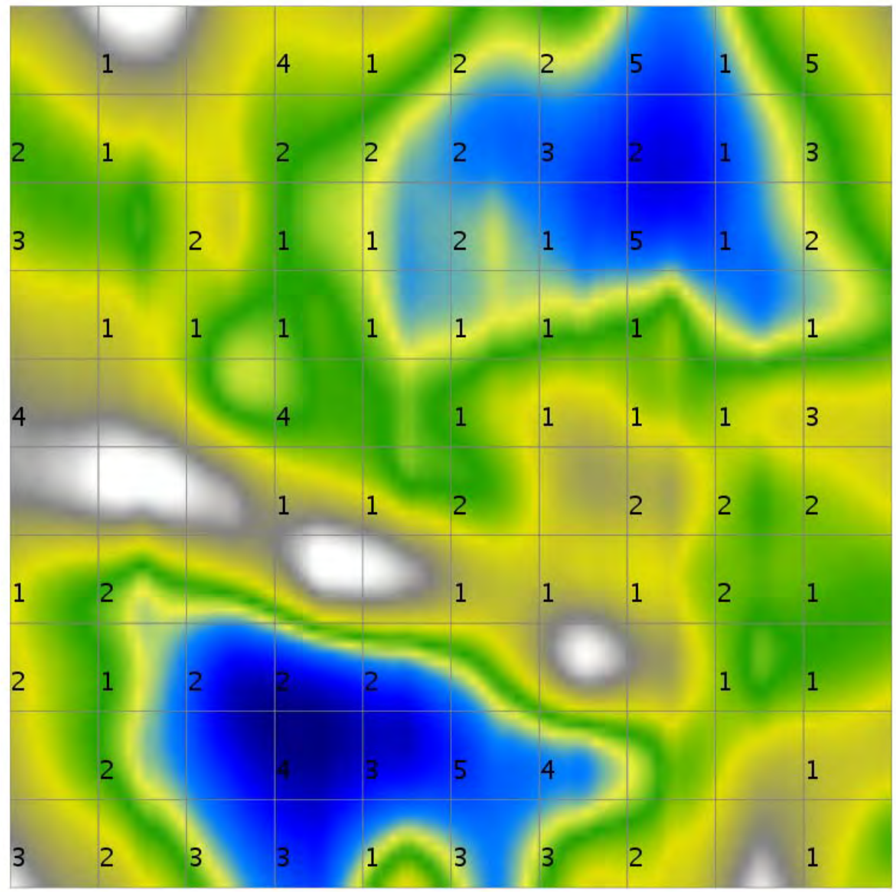




Types of Visualizations

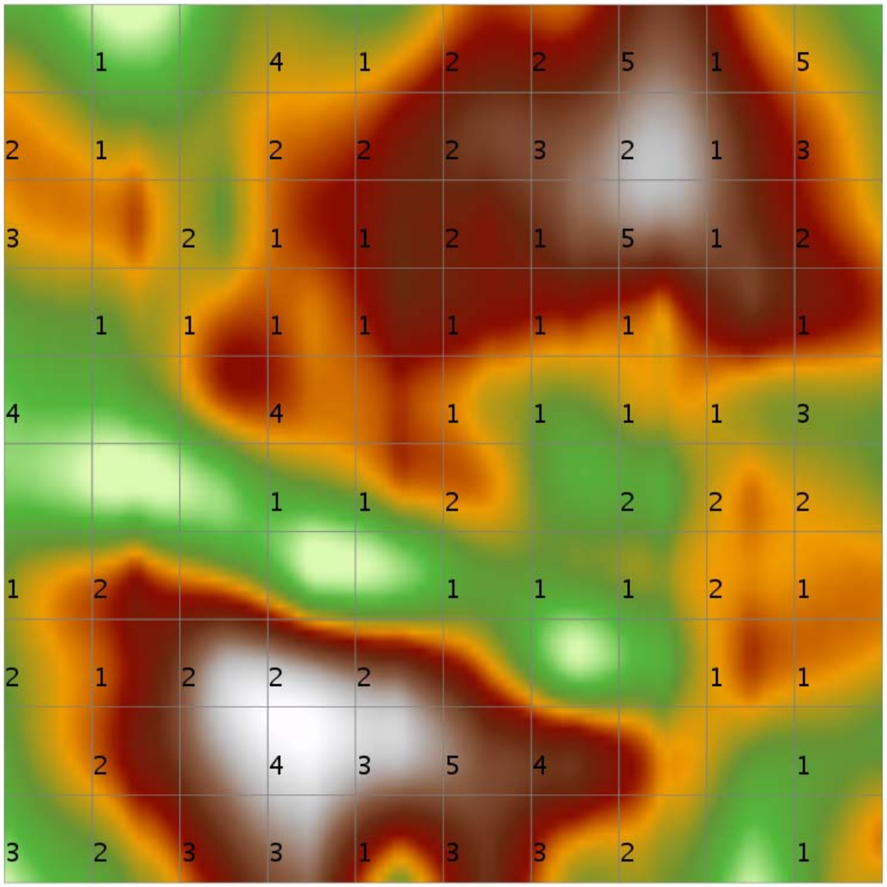
- Interpolating over units: inverting color palette

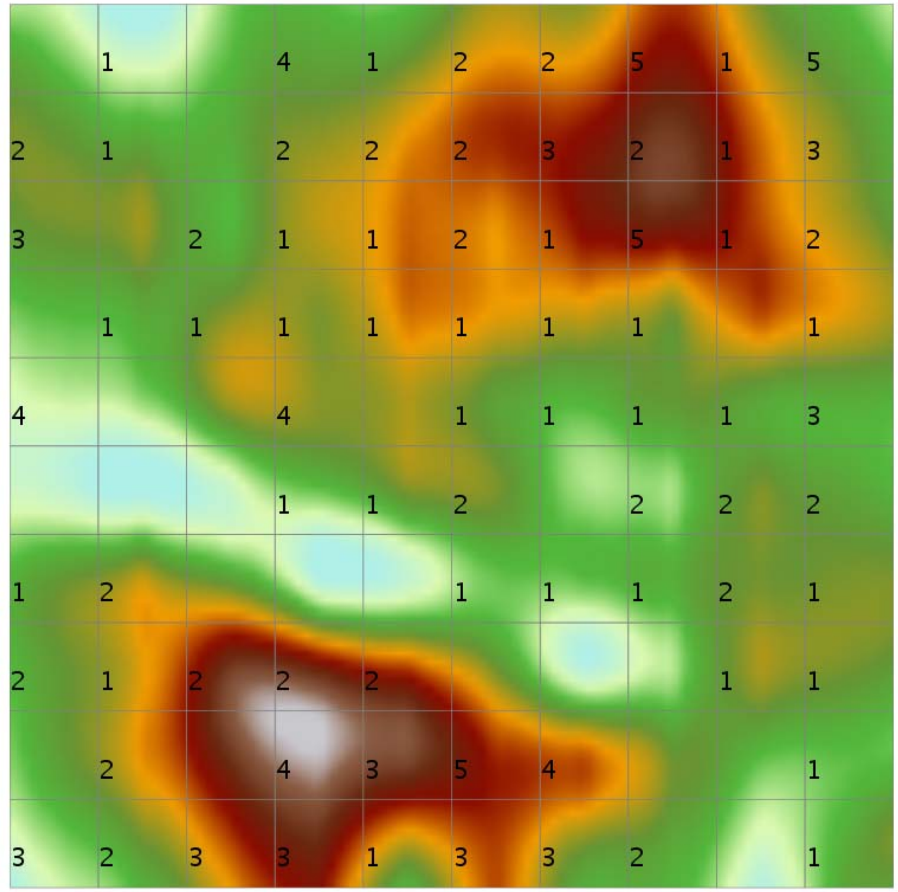




Types of Visualizations

- Interpolating over units: gradient in palettes





Excursion: Accountability

- **ACM Statement on Algorithmic Transparency and Accountability**, May 25 2017

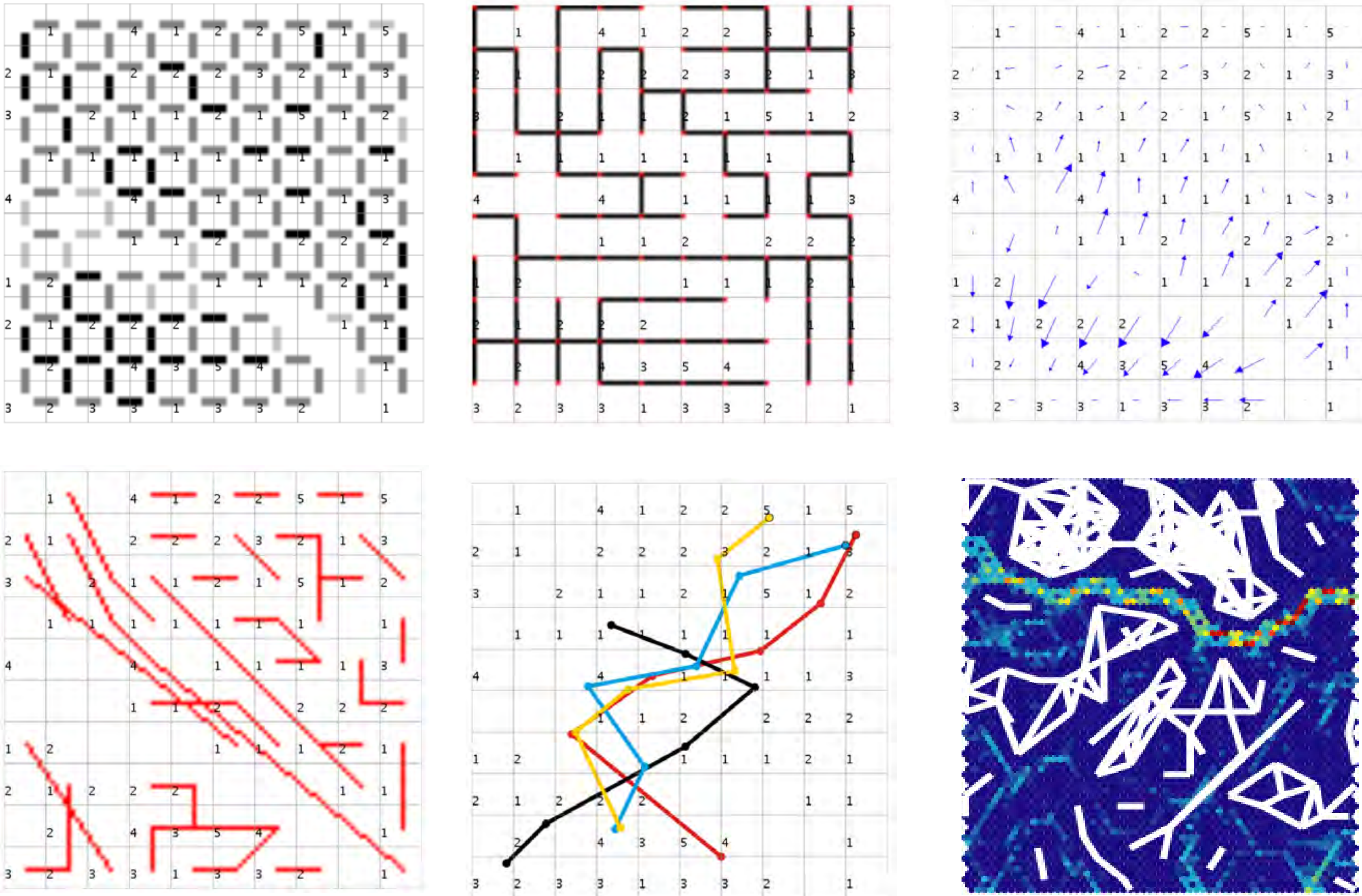
  http://www.acm.org/binaries/content/assets/public-policy/2017_joint_statement_algorithms.pdf

1. **Awareness**: potential bias
2. **Access and redress**: for individuals and groups
3. **Accountability**: responsible for decisions made by algorithms
4. **Explanation**: encouraged to explain procedures, decisions
5. **Data Provenance**: data collection, bias analysis, …
6. **Auditability**: models, data, algorithms recorded
7. **Validation and Testing**: rigorous, routinely, public

Types of Visualizations

- Graphs / connecting lines on SOM
- Drawing structures and connections
- Type of visualization parameters
  - direction of lines
  - color
  - thickness
  - form
- can be used as overlay on colorings & text -> combining visualizations

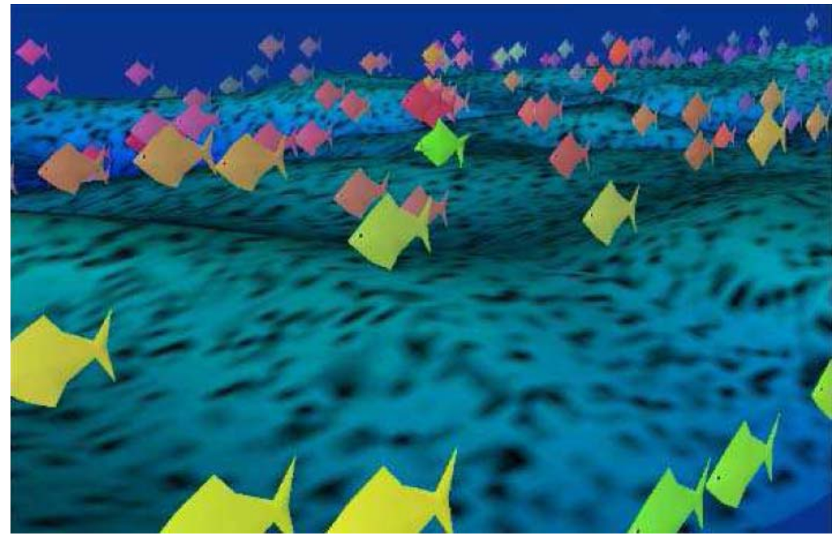
Types of Visualizations

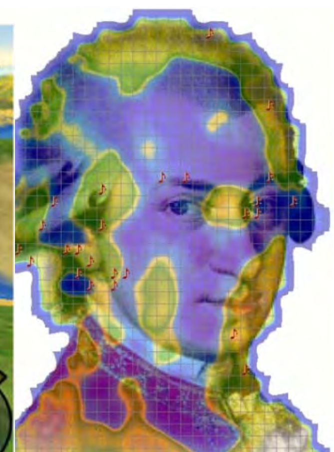


Types of Visualizations

- Many further types
- SOM as basis for visualizations
- SOM creates lower-dim data space as basis for
  - 3-D worlds
  - Metaphor graphics
  - Basis for interpolations
- Domains specific visualizations

Types of Visualizations





Types of Visualizations

Summary

- Many possibilities for visualizations
  - SOM as basis
  - textual /numeric info
  - (colored) symbols, metaphor graphics
  - coloring the units
  - coloring via interpolating over units
  - lines, graphs as overlays
  - other (e.g. 3D worlds)
- Basis for
  - analysis of SOM
  - using the SOM
- What information can be mapped?

Outline

- Overview of visualization types
- Visualizing the SOM
  - Codebook projection
  - Adaptive Coordinates
- Visualizations on the SOM
  - Textual information
  - Density
  - Distances
  - Class info
  - Attributes
  - Clustering of the SOM

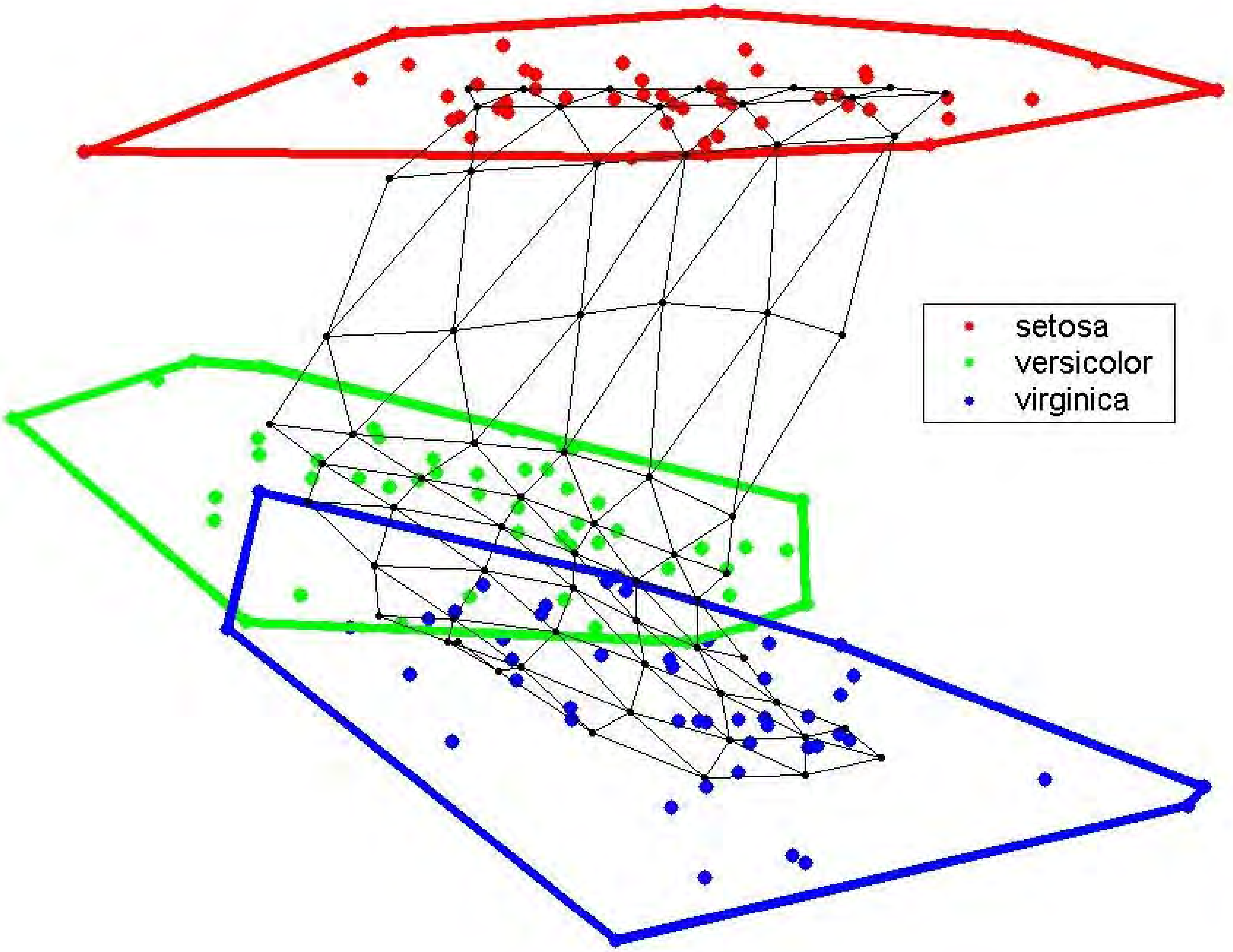
Codebook-Projektion

Projection of SOM Codebook

- Mapping the relationship between input and weight vectors
- Not mapping ON the SOM!
- Only for 2-dim data the codebook would have a 2-dim position
- Project data into 2-dim space by other means (e.g: PCA, Sammons Mapping)
- Distorted visualization of relationship between data and codebook vectors

Codebook-Projektion

PCA Projection of Iris Dataset



Adaptive Coordinates

Adaptive coordinates:

- Mimick the relative movement of weigth vectors through input space during training
- Details: c.f. lecture on “related architectures”

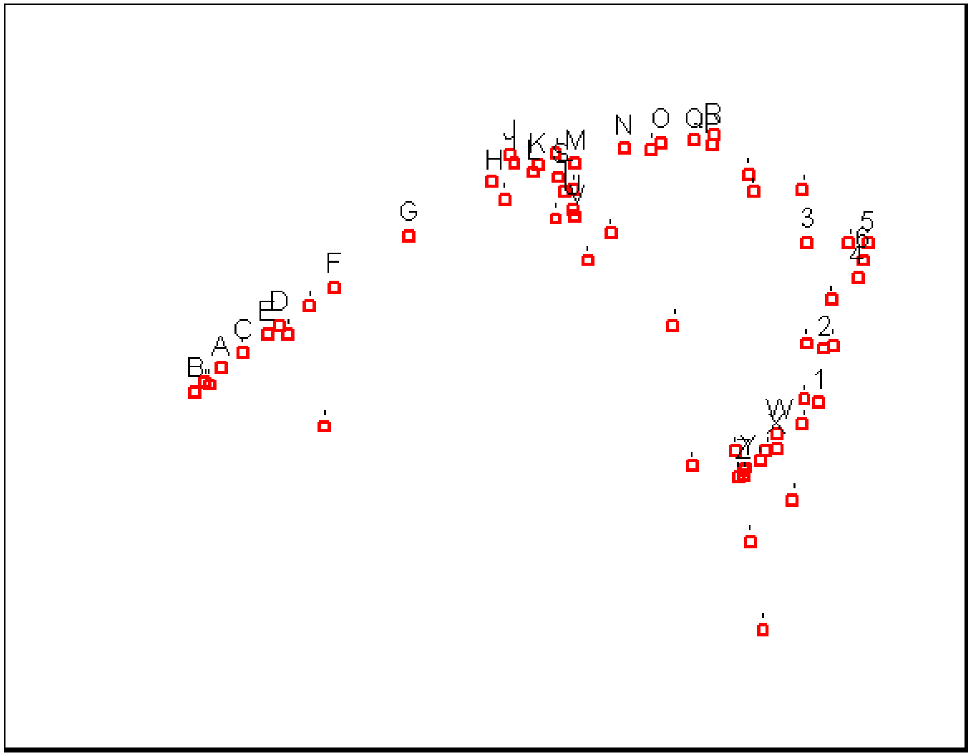
– Dieter Merkl, Andreas Rauber: **Alternative Ways for Cluster Visualization in Self-Organizing Maps**. Proc of the Workshop on Self-Organizing Maps (WSOM97), Helsinki, Finland, 1997

Adaptive Coordinates

- Example dataset:

A	.	B	.	.	.	.	.
C	.	.	.	Z	Y	.	.
D	E	.	.	.	.	X	.
F	.	.	V	.	W	.	1
G	.	T	U	.	.	2	.
H	L	S	.	.	3	.	4
I	K	M	.	P	.	.	6
J	.	N	O	Q	R	.	5

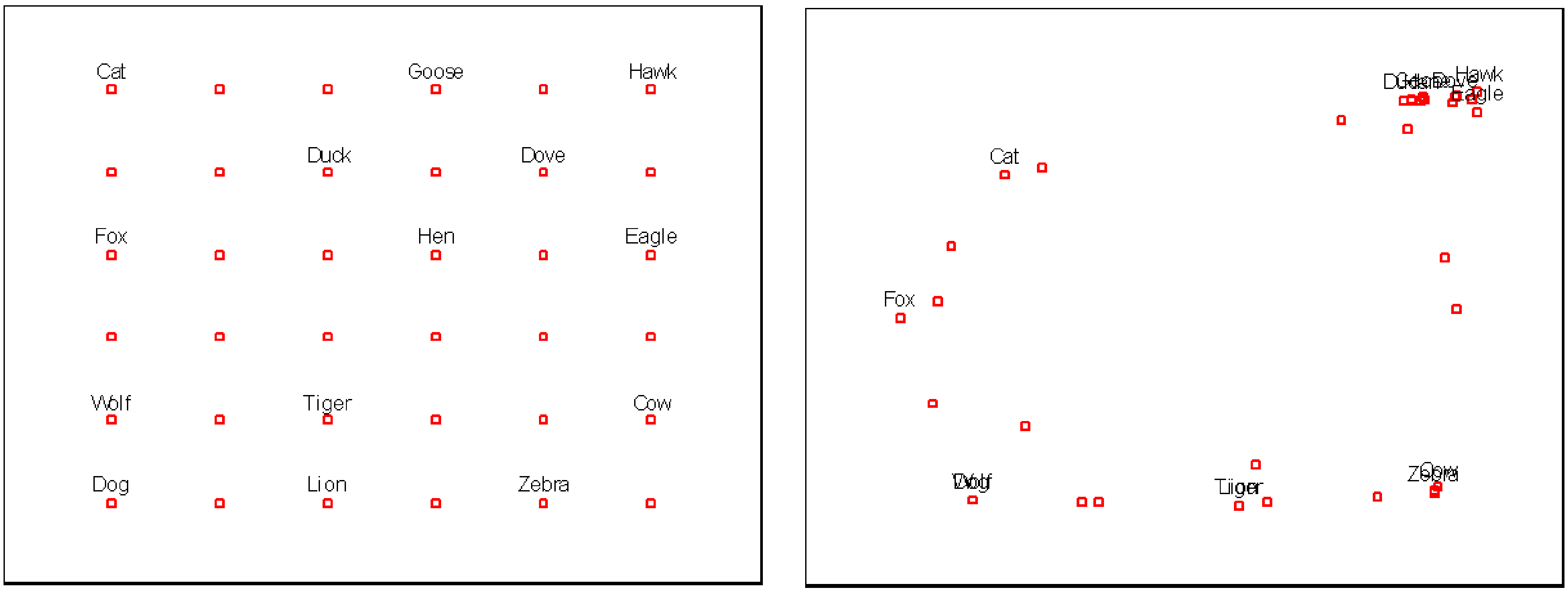
Standard SOM



AC representation

Adaptive Coordinates

- Example dataset: Animals



a.) b.)

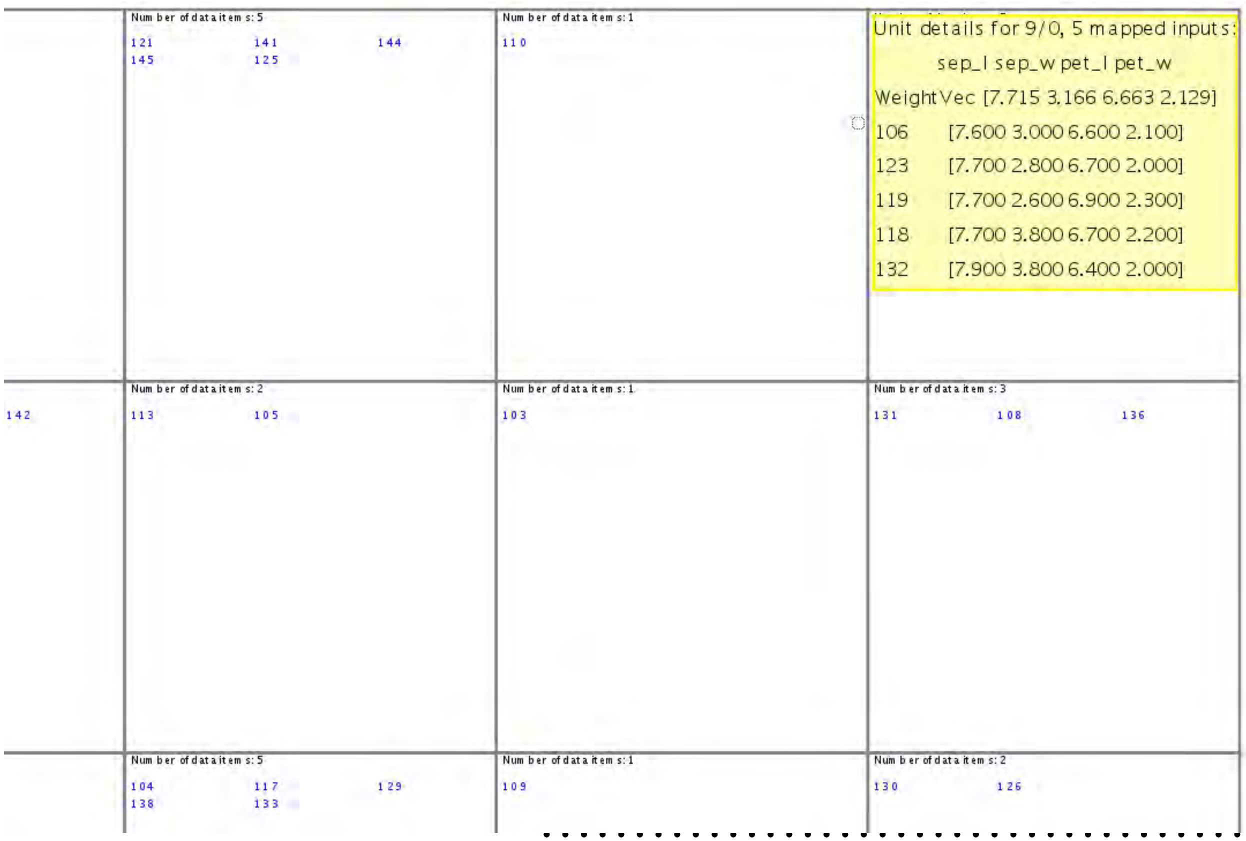
Standard SOM AC representation

Outline

- Overview of visualization types
- Visualizations on the SOM
  - Textual information
  - Density
  - Distances
  - Class info
  - Attributes
  - Clustering of the SOM

Text-Informationen auf der SOM

- Vector infos
  - Vector IDs, mapping distances, quality measures, classes,…
  - Semantic zooming



Outline

- Overview of visualization types
- Visualizing the SOM
  - Codebook projection
  - Adaptive Coordinates
- Visualizations on the SOM
  - Textual information
  - Density
  - Distances
  - Class info
  - Attributes
  - Clustering of the SOM

Outline

Visualization of the SOM

- Textual Information
- Density
  - Hit Histogramm
  - Smoothed Data Histograms
  - P-Matrix
  - Sky Metaphor
  - Neighborhood Graphs
- Distances
- Class info
- Attributes
- Clustering of the SOM

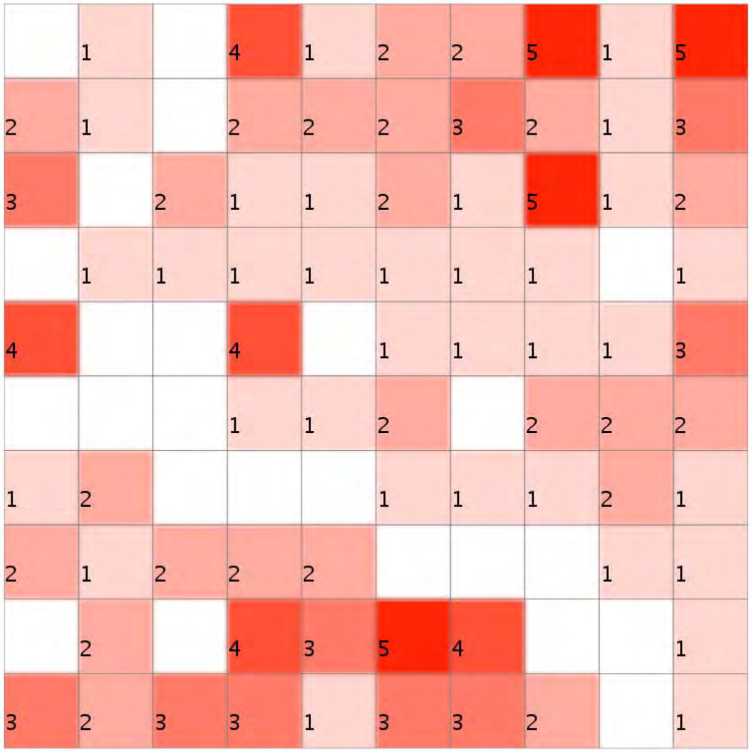
Density

- Density: Distribution of data items on the map
- Each input vector mapped onto its best-matching unit
- Empty units = „interpolating units“, i.e. transitional areas between denser regions -> cluster boundaries?
- Magnification Factor
- Different types
  - Textual: number of vectors
  - Hit Histogram: color, patch sizes
  - Smoothed Data Histograms
  - P-Matrix

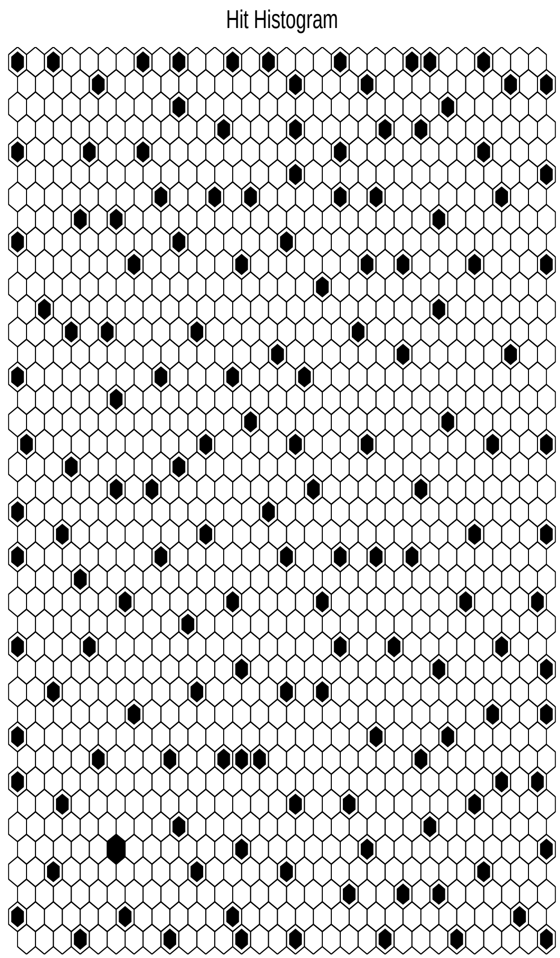
Density – Hit Histogramm

- Vector Infos
  - Hit histogram: number of vectors per unit
  - Iris Dataset:

	1		4	1	2	2	5	1	5
2	1		2	2	2	3	2	1	3
3		2	1	1	2	1	5	1	2
	1	1	1	1	1	1	1		1
4			4		1	1	1	1	3
			1	1	2		2	2	2
1	2				1	1	1	2	1
2	1	2	2	2				1	1
	2		4	3	5	4			1
3	2	3	3	1	3	3	2		1



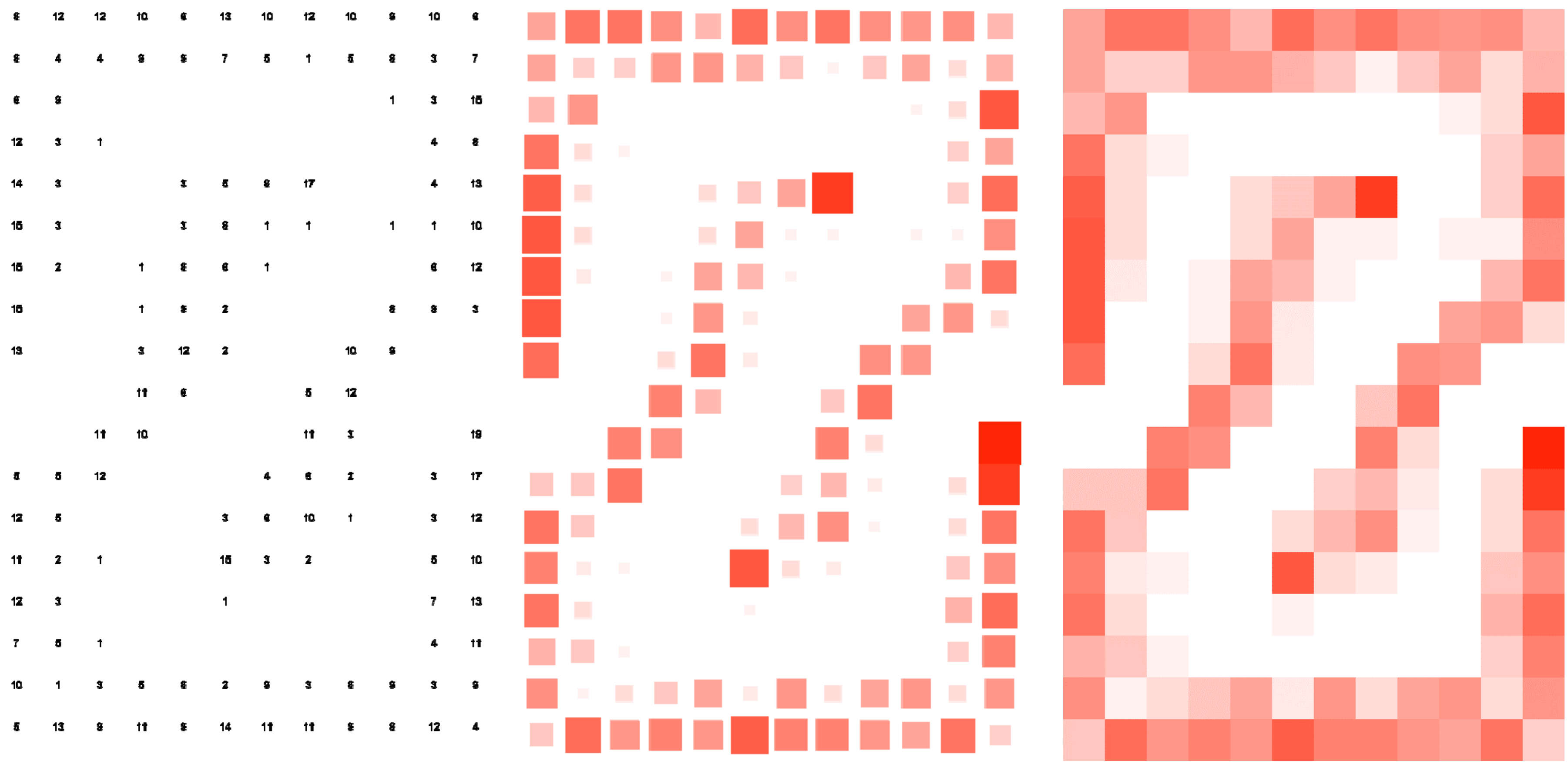
Density – Hit Histogramm



Iris (Emergent SOM)

Density – Hit Histogramm

- Chain Link Dataset



Outline

Visualization of the SOM

- Textual Information
- Density
  - Hit Histogramm
  - Smoothed Data Histograms
  - P-Matrix
  - Sky Metaphor
  - Neighborhood Graphs
- Distances
- Class info
- Attributes
- Clustering of the SOM

Outline

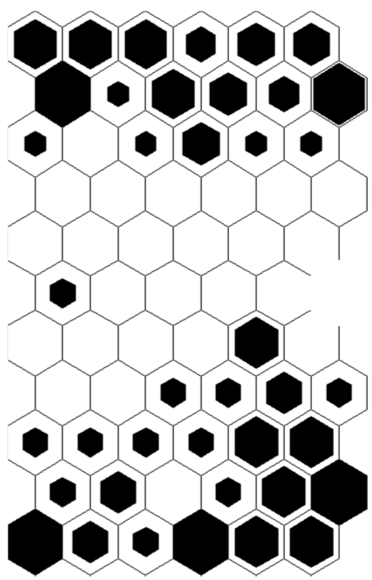
Visualization of the SOM

- Textual Information
- Density
  - Hit Histogramm
  - Smoothed Data Histograms
  - P-Matrix
  - Sky Metaphor
  - Neighborhood Graphs
- Distances
- Class info
- Attributes
- Clustering of the SOM

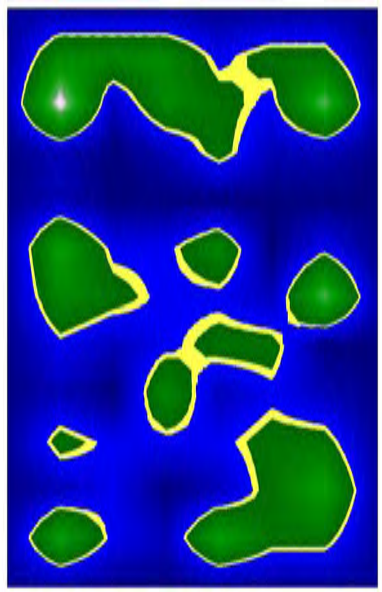
Density: Smoothed Data Histogram

- Extension of Hit Histograms
- Each input vector is mapped not only onto best-matching unit, but onto n-best matching units
- 4 different methods
  - counting all n units equally
  - weigth depends on distance: 1/d
  - normalized: 1/n: for 1. unit 1, then 1/2, 1/3, ...
  - normalized distance weight: $1/d_n$ (d_n : min-max normalized distance)
- creates smoothing effect
- Parameter n: controls granularity of cluster structures
- E. Pampalk, A. Rauber, D. Merkl: **Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps**. In: Proceedings of the Intl.Conf on Artificial Neural Networks (ICANN 2002), pp.871-876.

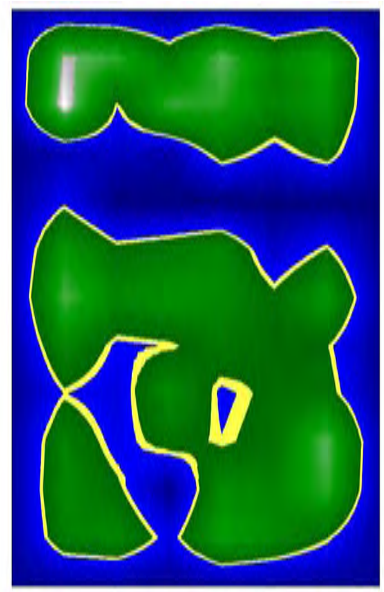
Density: Smoothed Data Histogram



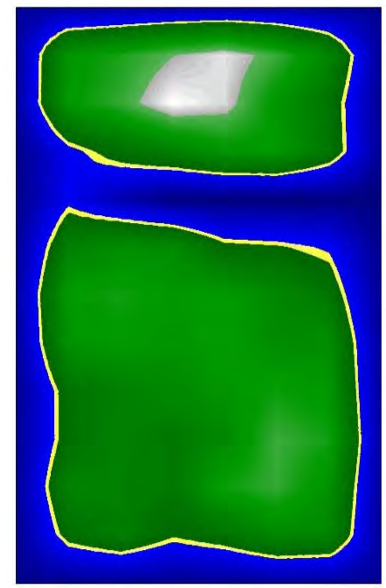
Hit-histogram



n = 1



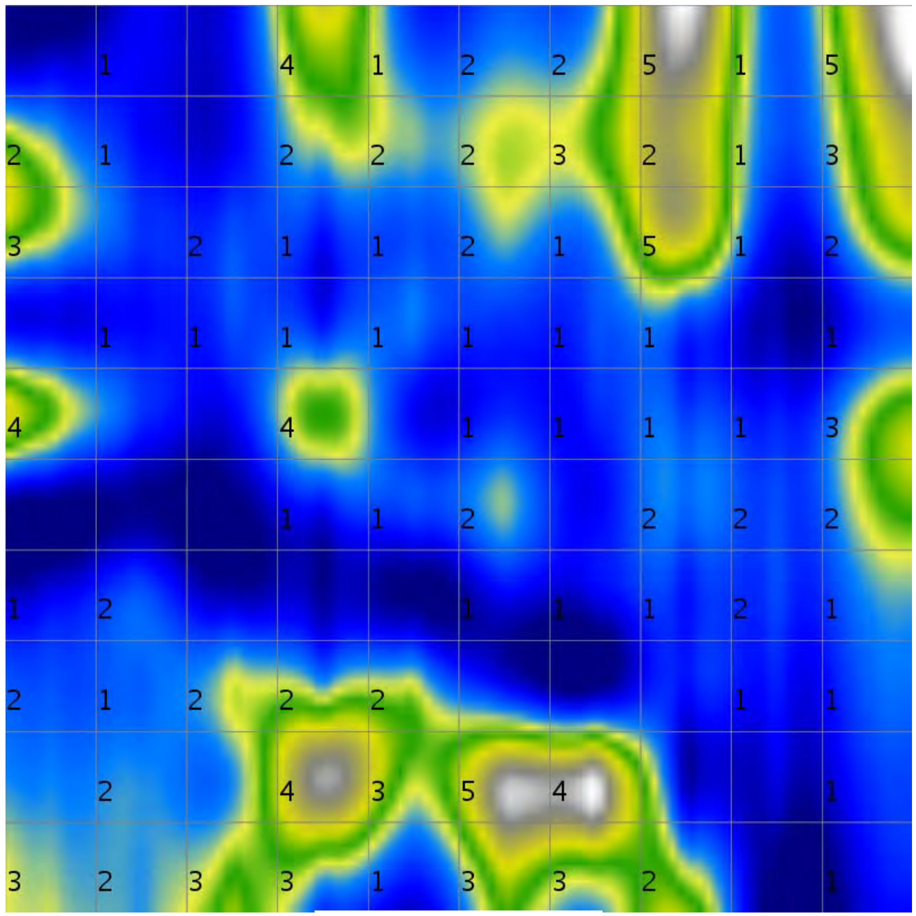
n = 3



n = 10

Density: Smoothed Data Histogram

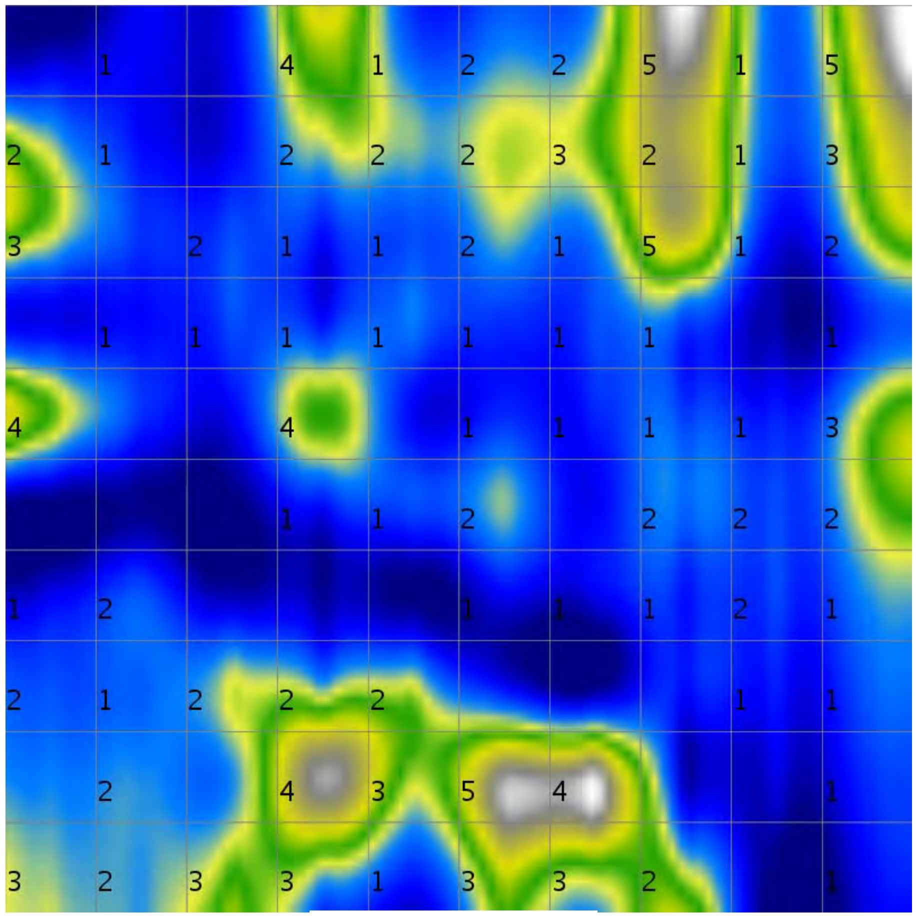
- Iris Dataset



n = 2

Density: Smoothed Data Histogram

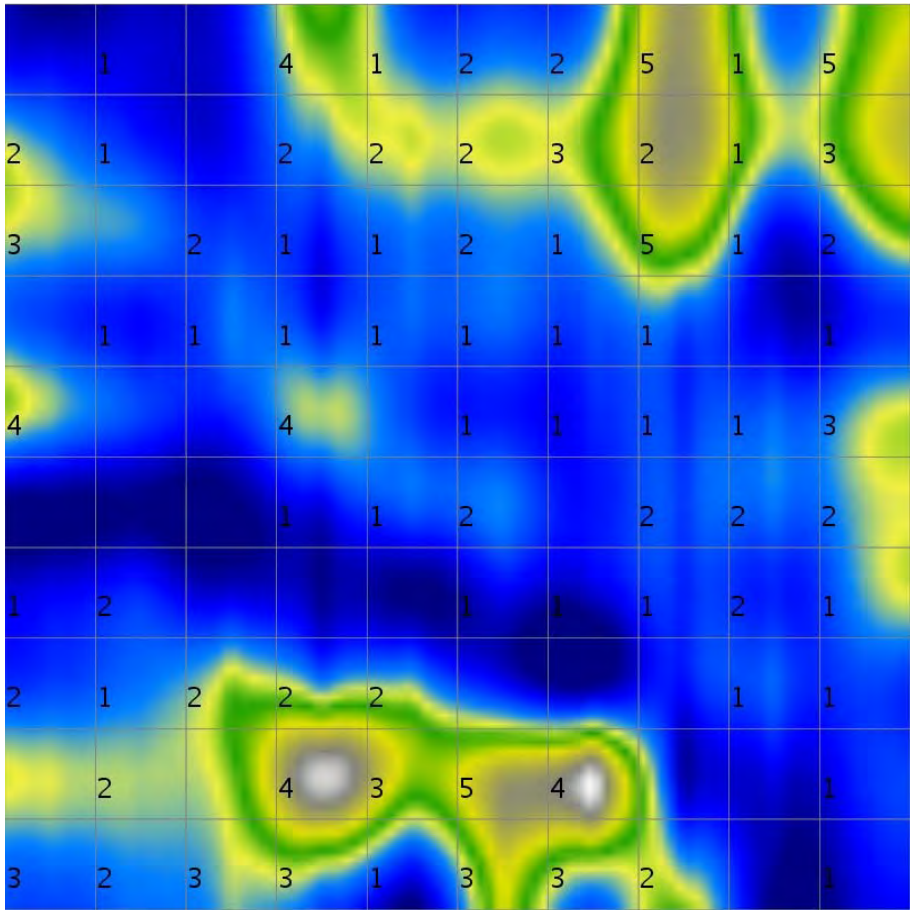
- Iris Dataset



n = 2

Density: Smoothed Data Histogram

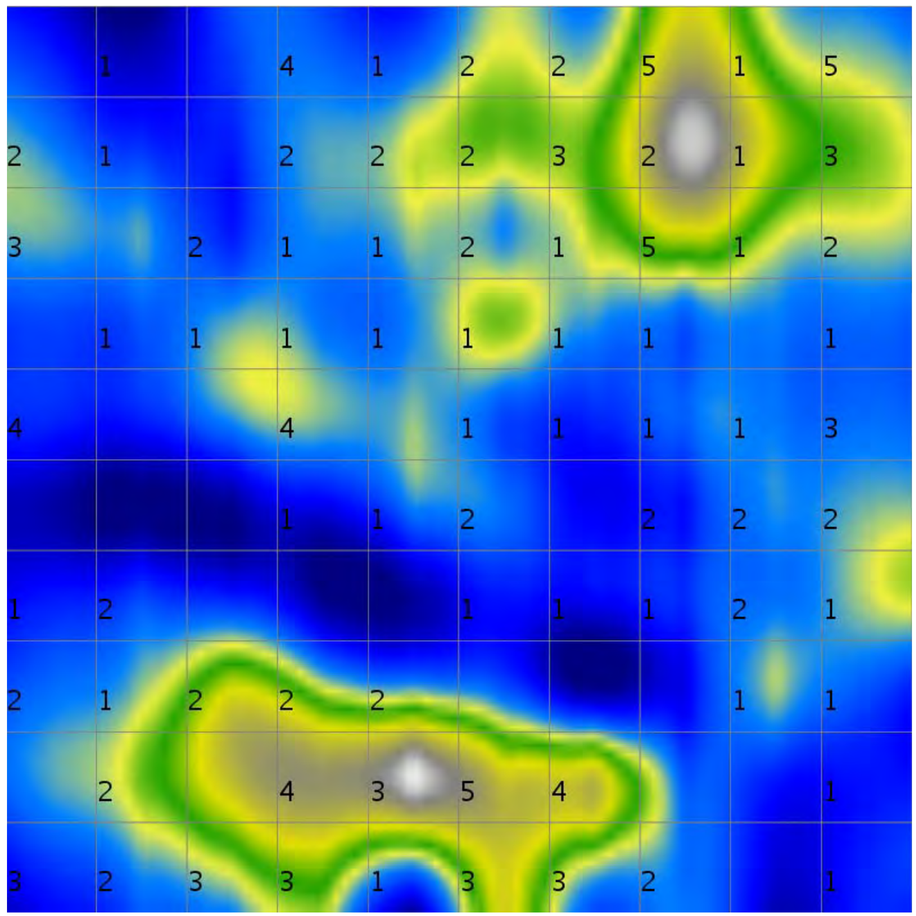
- Iris Dataset



n = 3

Density: Smoothed Data Histogram

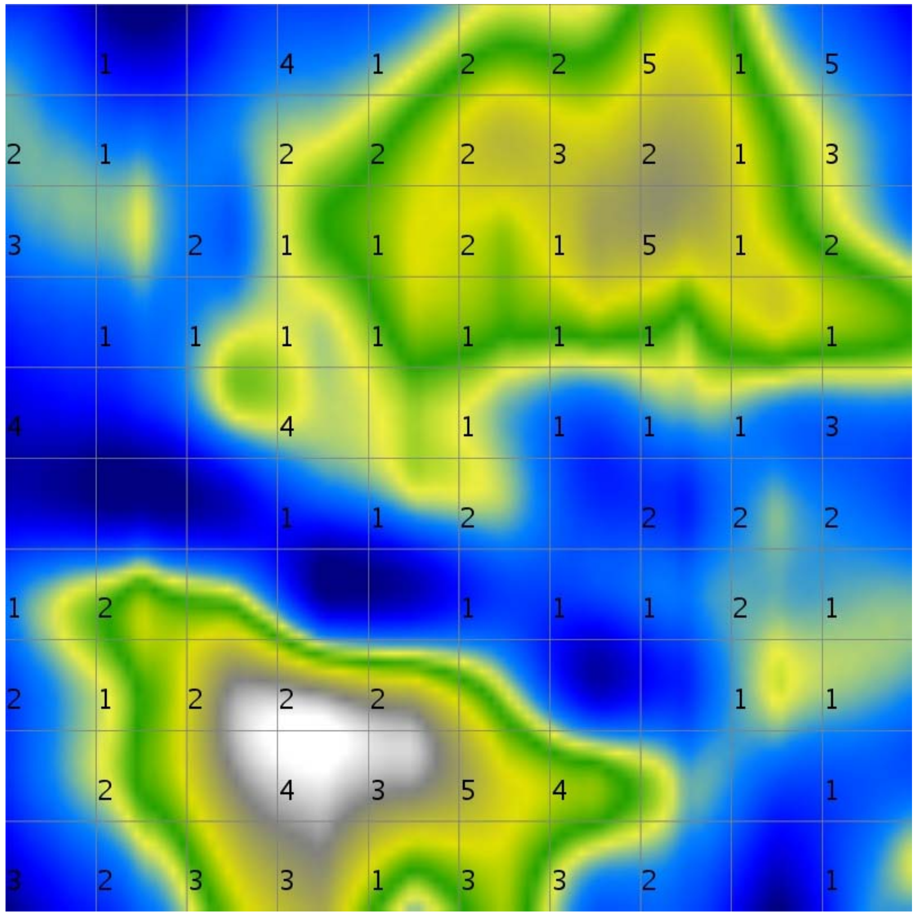
- Iris Dataset



n = 8

Density: Smoothed Data Histogram

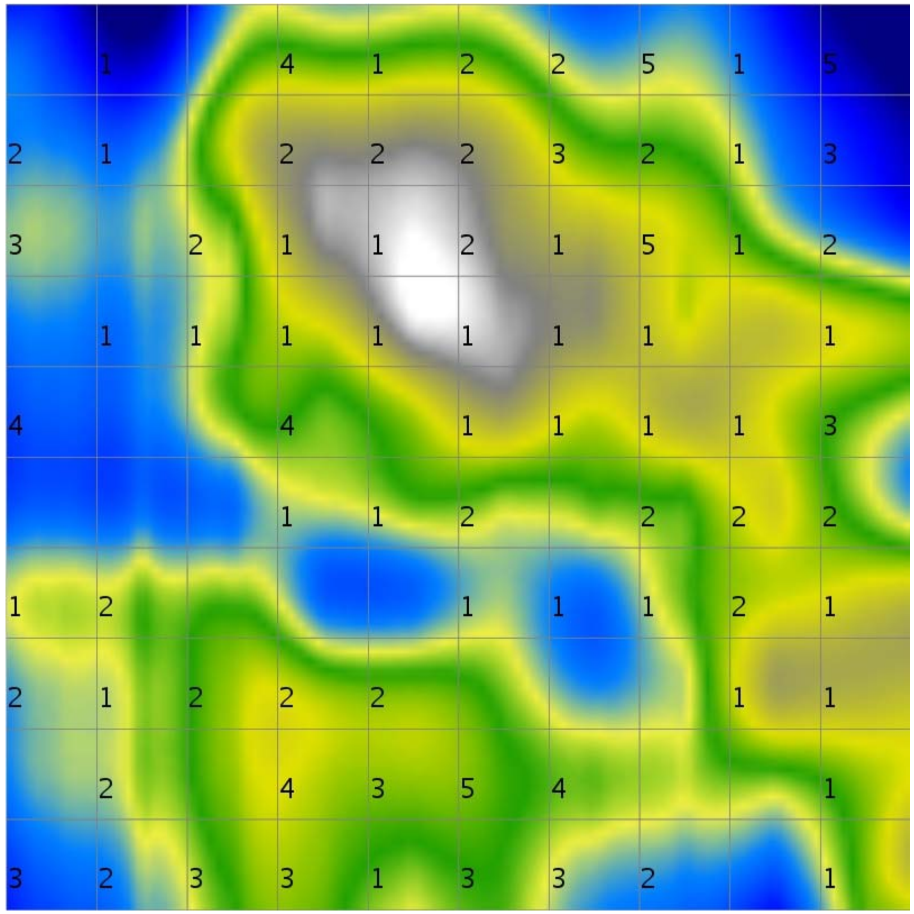
- Iris Dataset



n = 20

Density: Smoothed Data Histogram

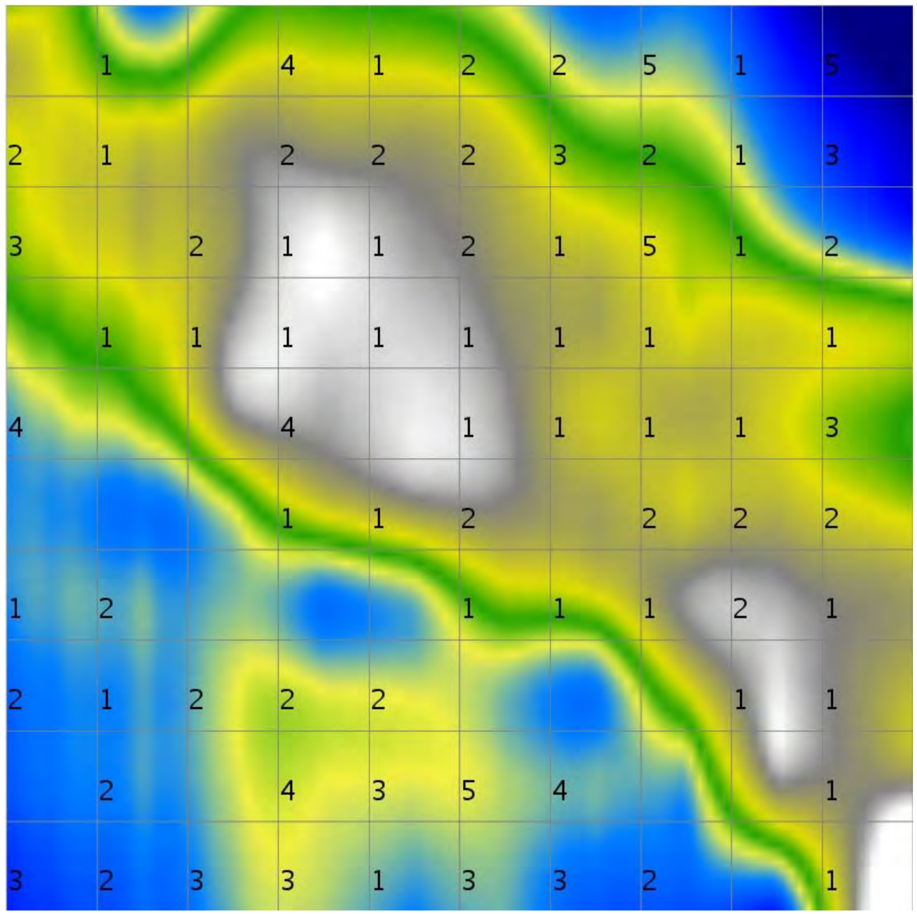
- Iris Dataset



n = 50

Density: Smoothed Data Histogram

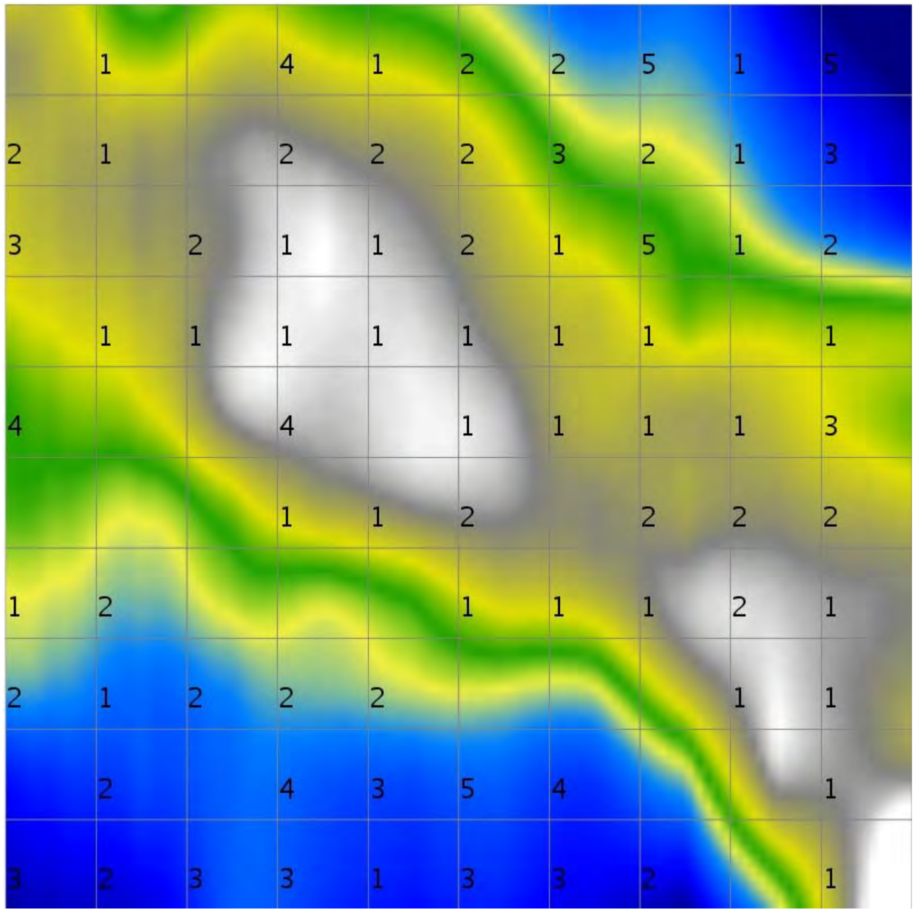
- Iris Dataset



n = 70

Density: Smoothed Data Histogram

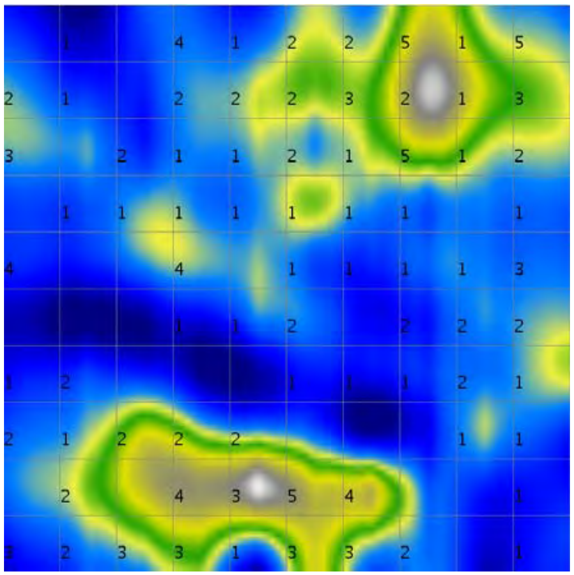
- Iris Dataset



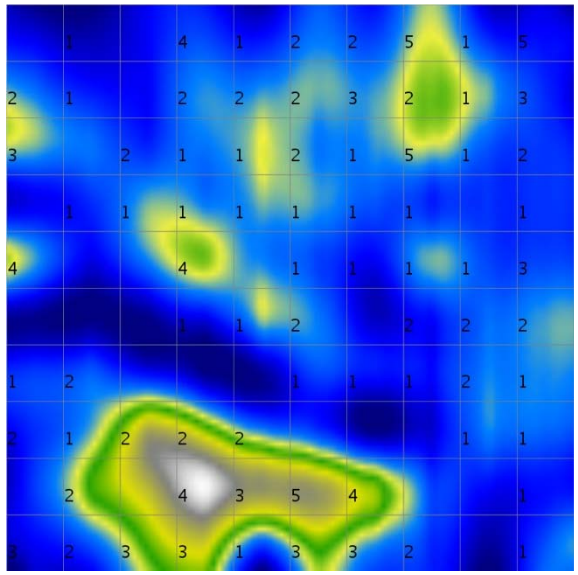
n = 90

Density: Smoothed Data Histogram

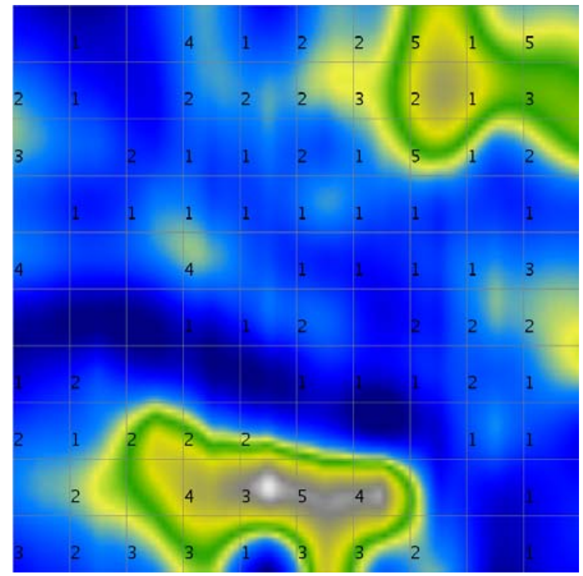
- Iris Dataset, SDN, n=8



SDH



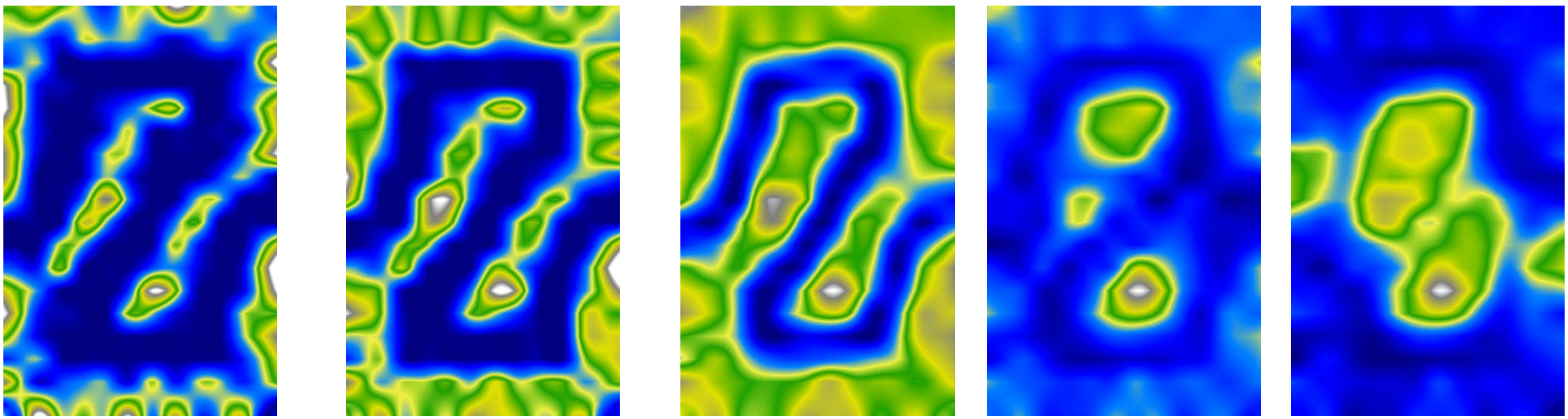
SDH-weighted



SDH-weighted, normalized

Density: Smoothed Data Histogram

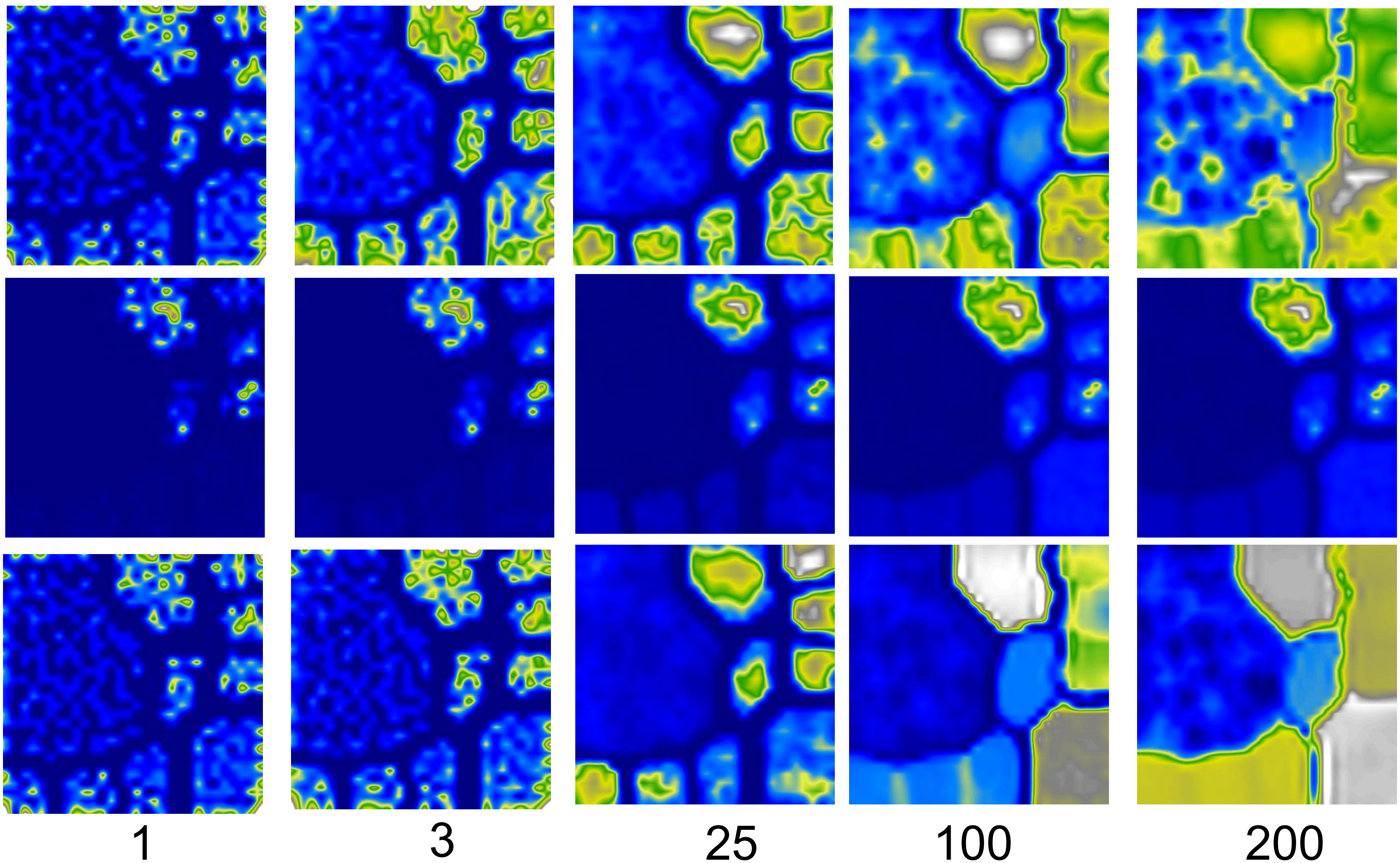
- Chainlink Dataset, weighted SDH



1 2 10 50 100

Density: Smoothed Data Histogram

- 10 clusters dataset: SDH, weighted, weighted-normalized



Density: Smoothed Data Histogram

Questions

- What happens at large smoothing factors? Why?
- differences between standard SDH, weighted SDH – when will visualizations differ? where?
- Which statements can be made when the two visualizations differ?
  (hint: cluster sizes, magnification factors, number of units per cluster)

Outline

Visualization of the SOM

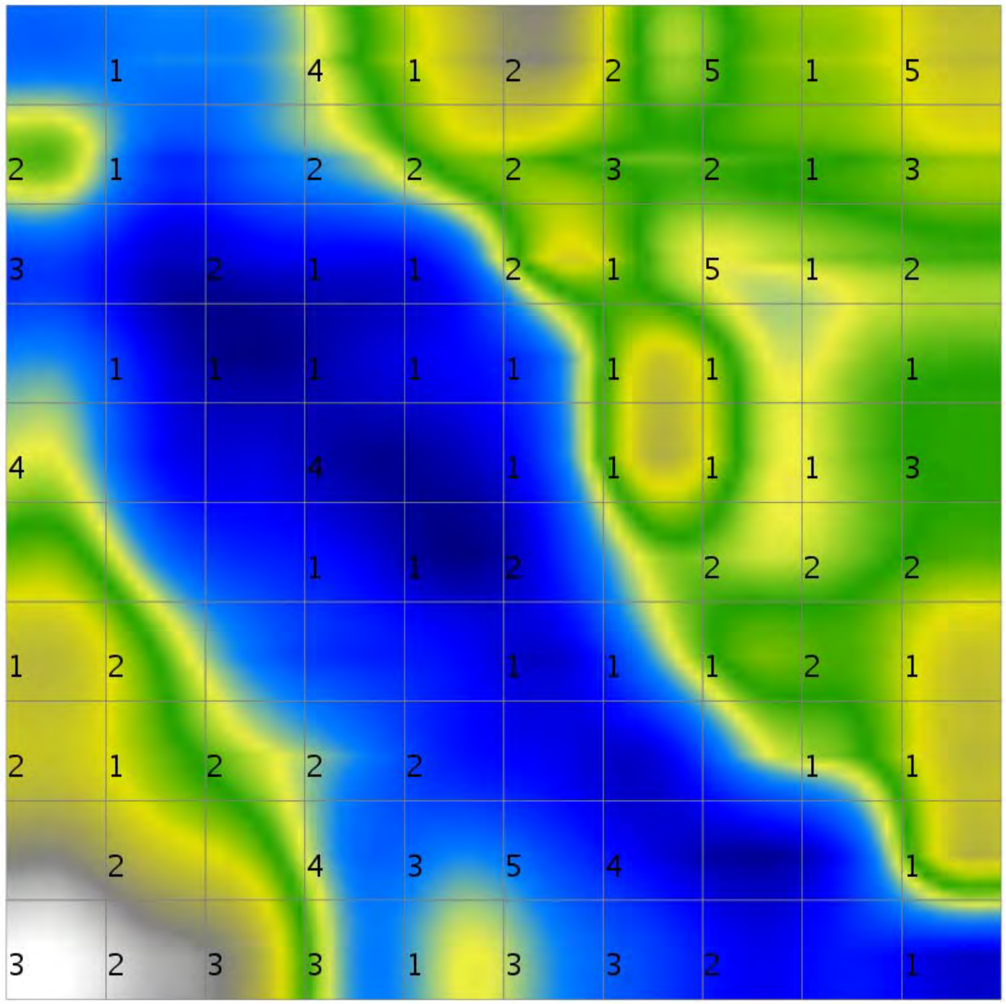
- Textual Information
- Density
  - Hit Histogramm
  - Smoothed Data Histograms
  - P-Matrix
  - Sky Metaphor
  - Neighborhood Graphs
- Distances
- Class info
- Attributes
- Clustering of the SOM

Density: P-Matrix

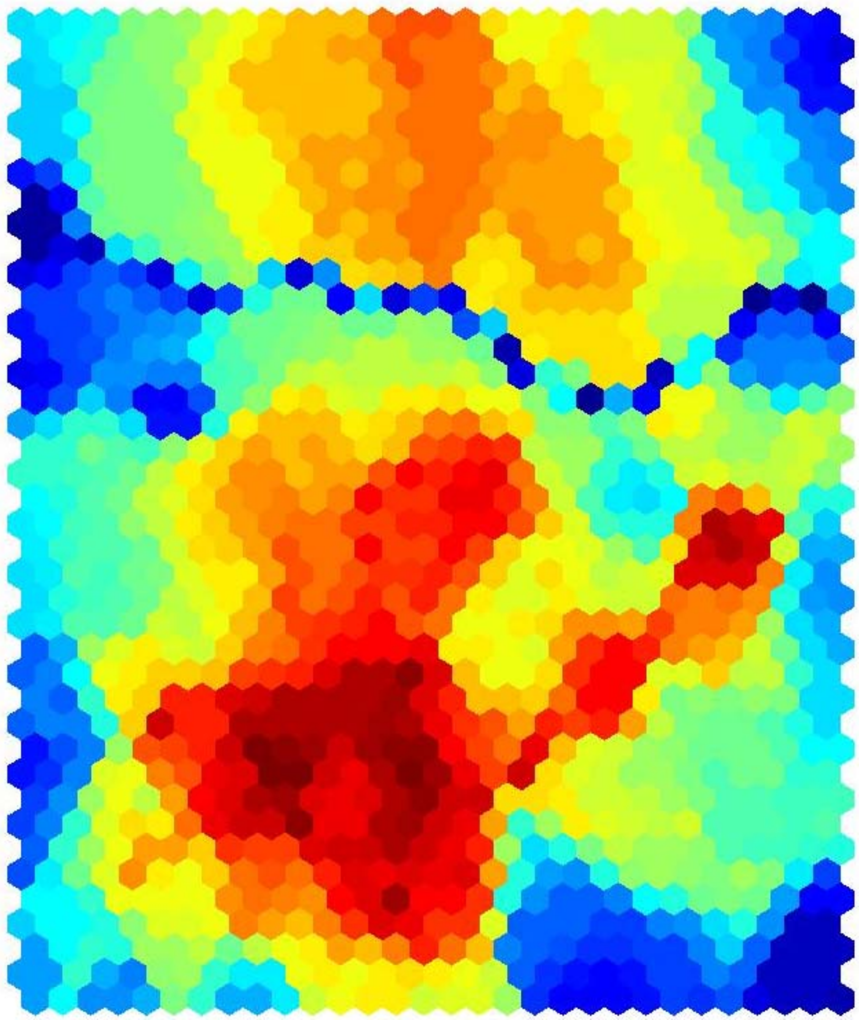
- Pareto-Matrix
- Hypersphere around each weigth vector in data space
- Counting the number of input vectors in this hypersphere provides estimate of density
- Very well suited for SOMs with large number of units (> number of data points, Emergent SOMs)
- Ultsch, A.: Maps for the Visualization of High-Dimensional Spaces. In: Proceedings of the 2003 Workshop on Self-Organizing Maps (WSOM), Kyushu, Japan, 2003. pp.91-100.

Density: P-Matrix

- Iris-Datenset, P-Matrix



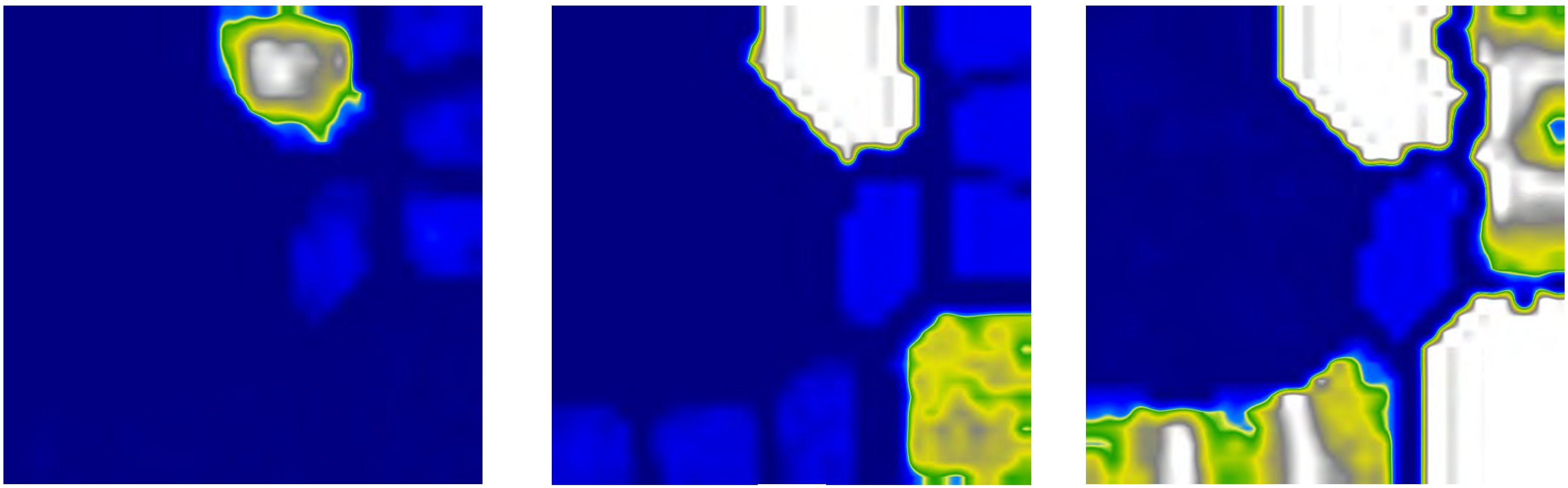
Density: P-Matrix



Iris (groß)
P-Matrix

Density: P-Matrix

- 10-clusters -Datenset, P-Matrix (varying percentile parameter)



1 5 10

Outline

Visualization of the SOM

- Textual Information
- Density
  - Hit Histogramm
  - Smoothed Data Histograms
  - P-Matrix
  - Sky Metaphor
  - Neighborhood Graphs
- Distances
- Class info
- Attributes
- Clustering of the SOM

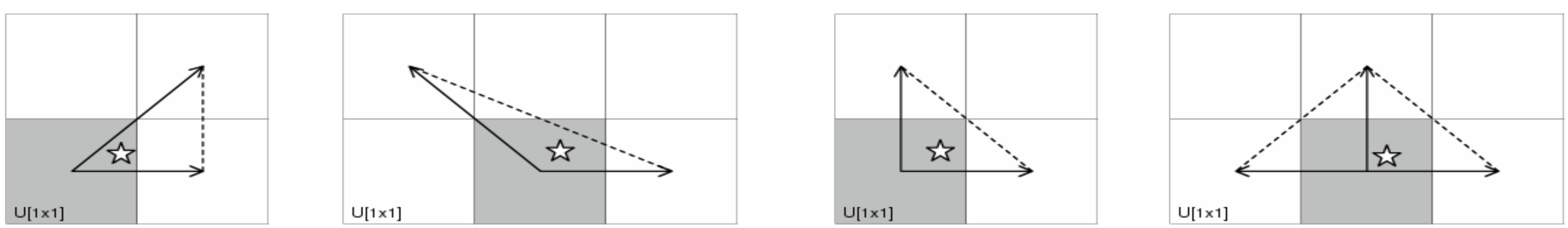
Density: Sky Metaphor

- Conventionally, data items mapped “on unit”
- Sky: display them on their “exact” position within the unit, not just in the centre
- Visualise similarity of an input with other inputs
  - within the same unit
  - across the neighbouring units
- Triangulation
- Khalid Latif and Rudolf Mayer.
  Sky-Metaphor Visualisation for Self-Organising Maps. In Proceedings of the 7th International Conference on Knowledge Management (I-KNOW'07), Graz, Austria, September 5 - 7 2007.

Density: Sky Metaphor

- Apply pull force to determine exact position
  - Relative to the distance of the input to BMU
  - Inverse proportional to the distance of the input to the other units

$$\mathbf{F}_i \quad \propto \quad \frac{d(x, U_1)}{d(x, U_i)} \quad \text{for} \quad i > 1$$



Density: Sky Metaphor

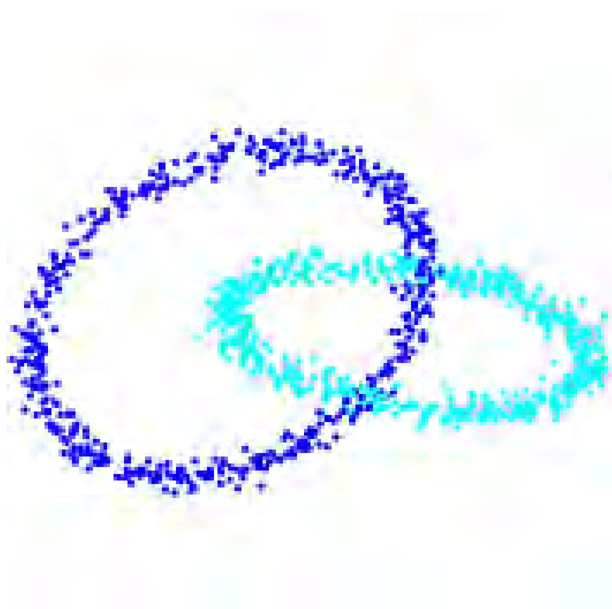
- X and y coordinates of the exact position p of input x on unit U_1 calculated as

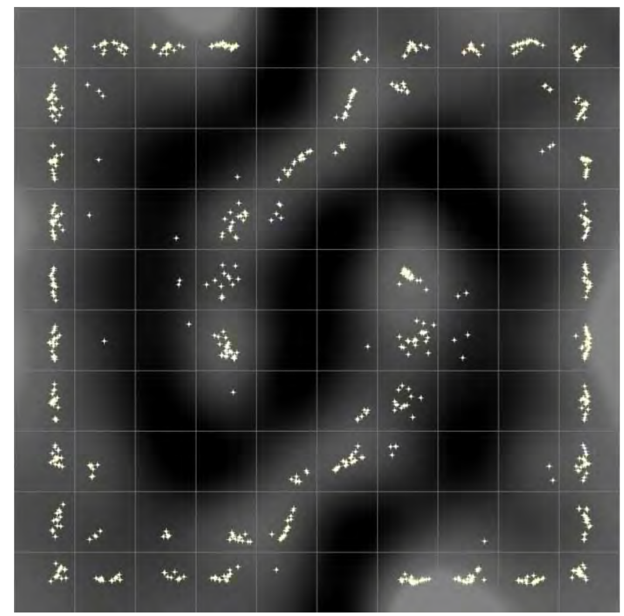
$$
\begin{aligned}
\mathbf{p}_{<x,y>} \quad = \quad & \langle \lambda * \sum_{i=2}^{k} \mathbf{F}_i * \frac{1}{U_{i<x>} - U_{1<x>}}, \\
& \lambda * \sum_{i=2}^{k} \mathbf{F}_i * \frac{1}{U_{i<y>} - U_{1<y>}} \rangle
\end{aligned}
$$

- k: index over the 2 or 3 nearest units U_2 ..$U_{4\ \mathrm{to}}$ to unit U_1
- λ: grid-constant to reconcile the displacement according to the display co-ordinates
- Combined with U-mat (or SDH) as background (b/w palette)

Density: Sky Metaphor

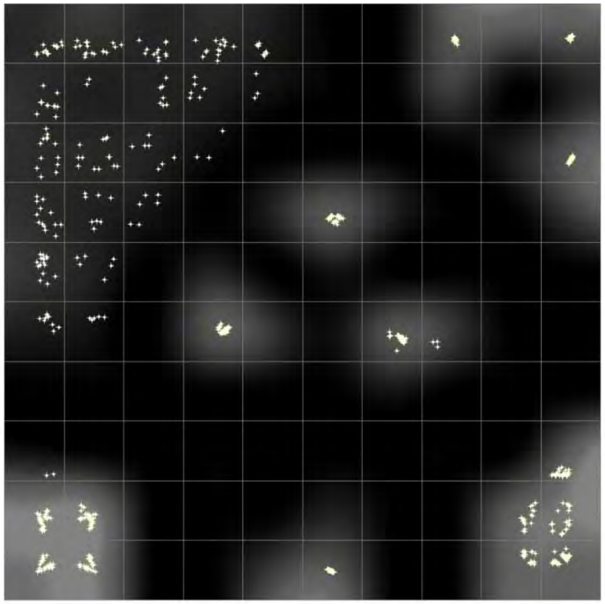
- Chain-link data set





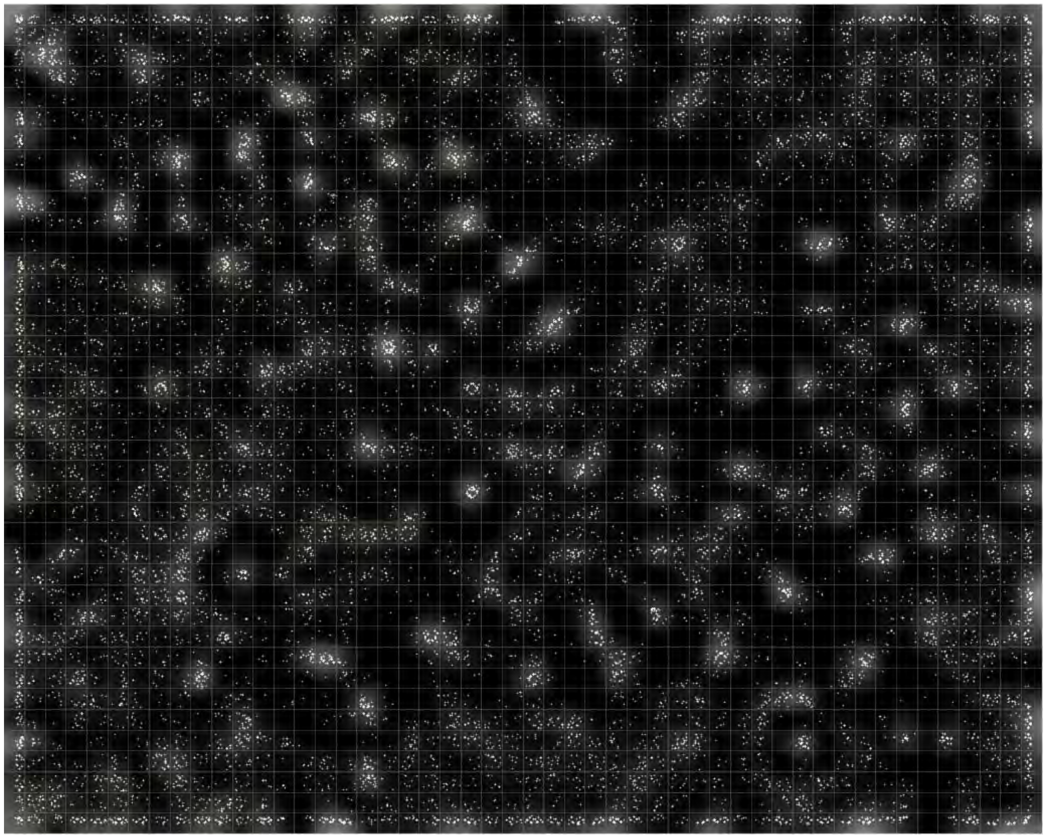
Density: Sky Metaphor

- 10-dimensional Gaussian distributions



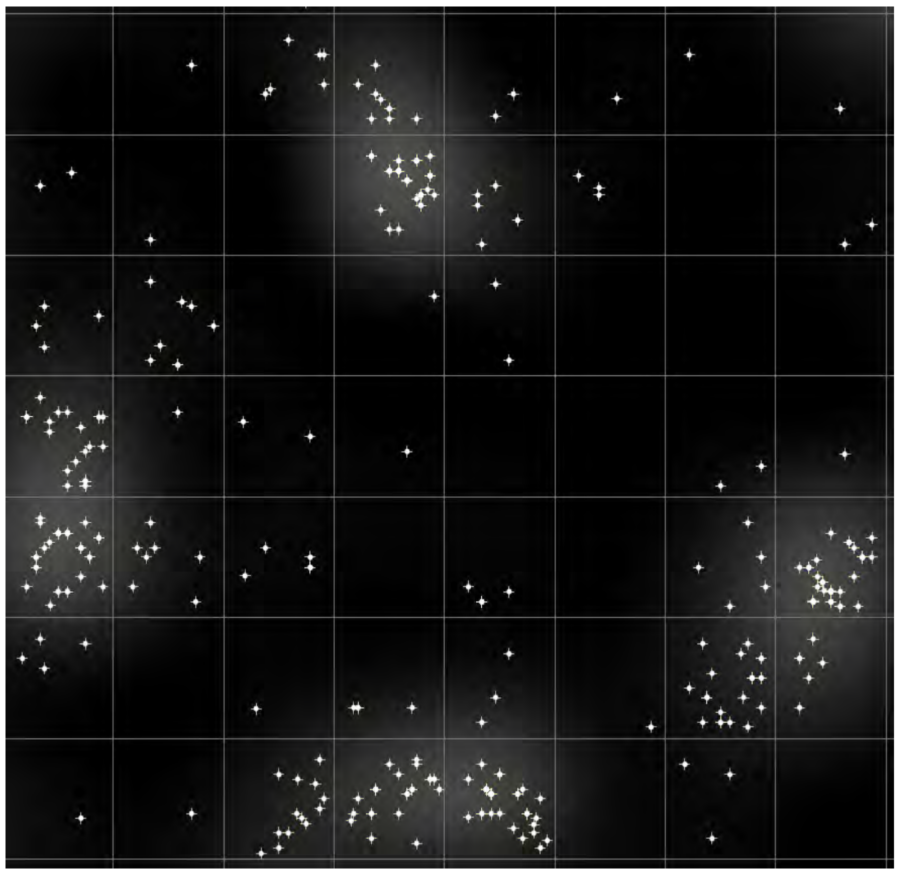
Density: Sky Metaphor

- 20 newsgroups data set



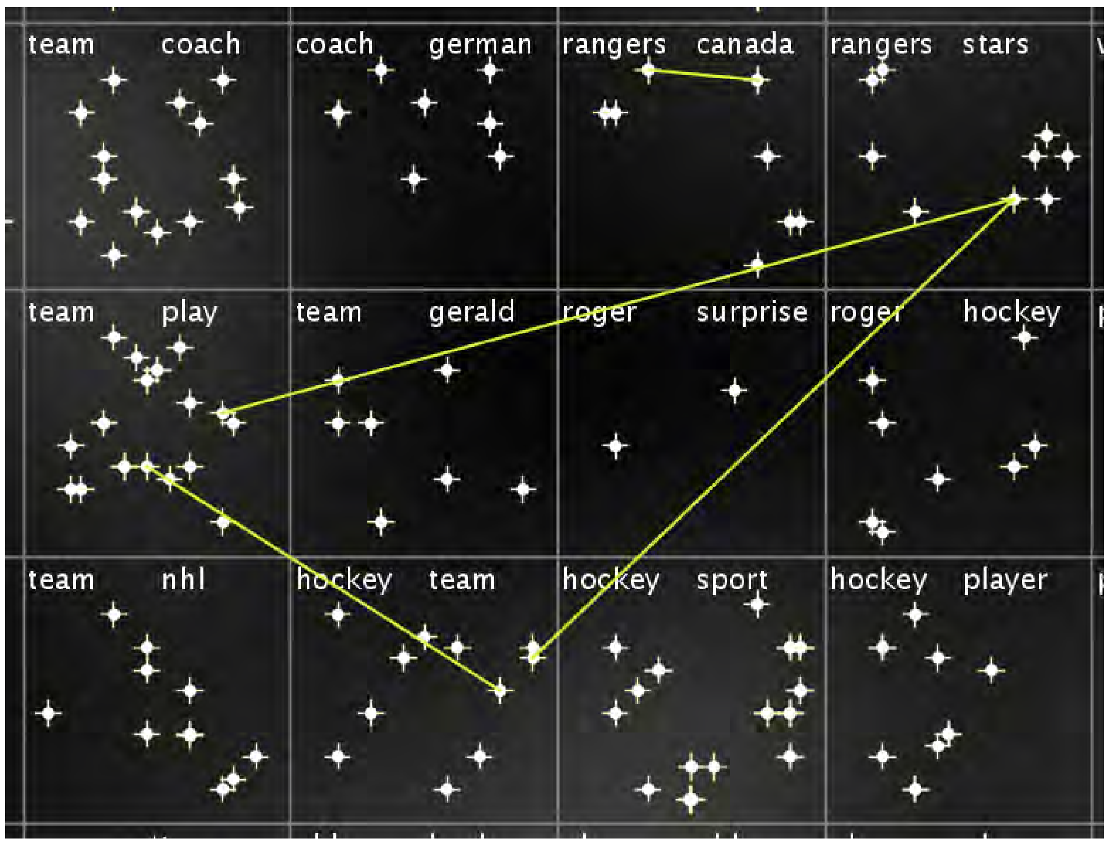
Density: Sky Metaphor

- 20 newsgroups data set



Density: Sky Metaphor

- 20 newsgroups data sets – semantic links



Density: Sky Metaphor

- 20 newsgroups data sets – semantic links

Outline

Visualization of the SOM

- Textual Information
- Density
  - Hit Histogramm
  - Smoothed Data Histograms
  - P-Matrix
  - Sky Metaphor
  - Neighborhood Graphs
- Distances
- Class info
- Attributes
- Clustering of the SOM

Density: Neighbourhood Graphs

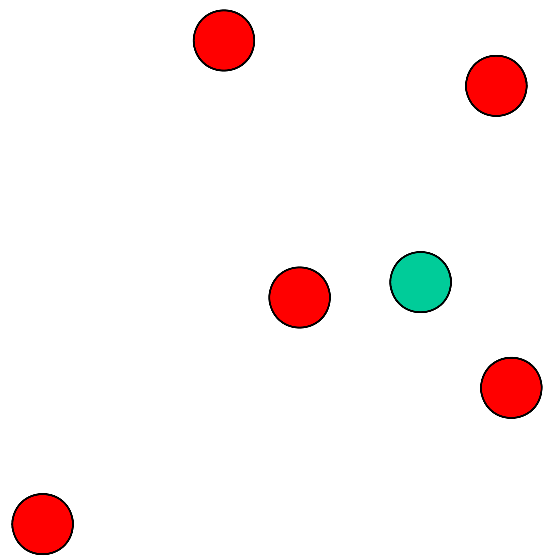
Show,

- which areas of the map are close to each other based on density in input space
- how well the topology is preserved
- the location of dense or sparse regions
- also reveals topology violations
- Georg Pölzlbauer, Andreas Rauber, and Michael Dittenbach. **Graph projection techniques for self-organizing maps.** In Michel Verleysen, editor, Proceedings of the European Symposium on Artificial Neural Networks (ESANN'05), pages 533-538, Bruges, Belgium, April 27-29 2005. d-side publications.

Density: Neighbourhood Graphs

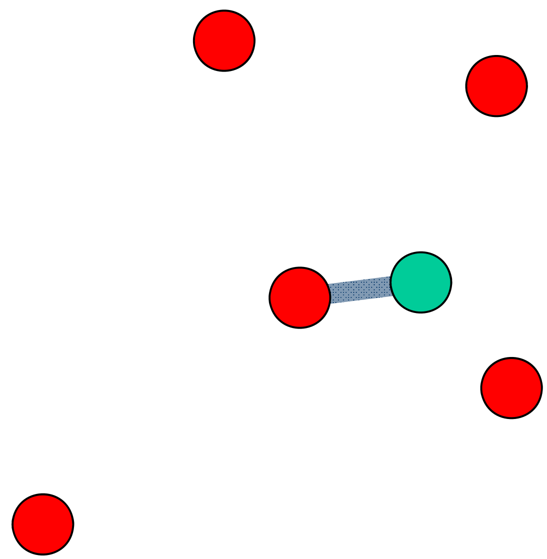
- Similar to P-matrix, but for pairs of units
- Different levels of granularity interactive analysis
- 2 approaches:
  - knn - based distances
  - radius-based distances

Density: Neighbourhood Graphs



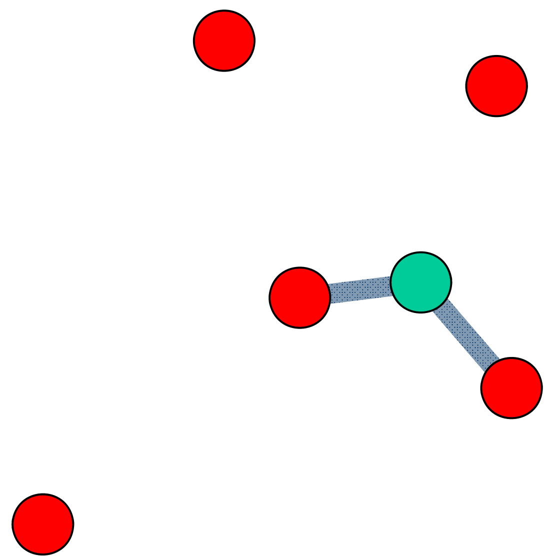
KNN-based distances in input space

Density: Neighbourhood Graphs



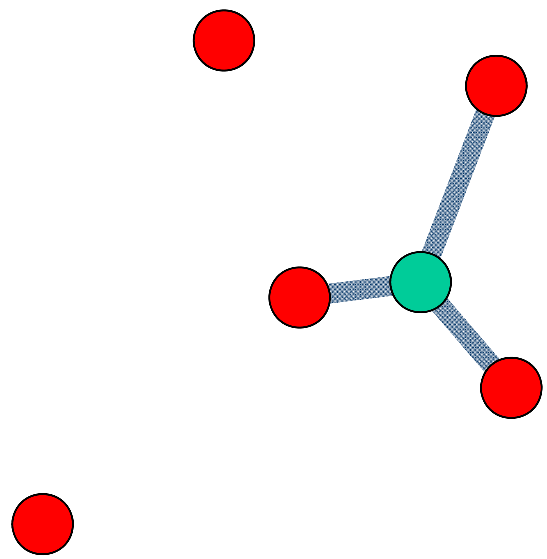
1-nearest neighbor

Density: Neighbourhood Graphs



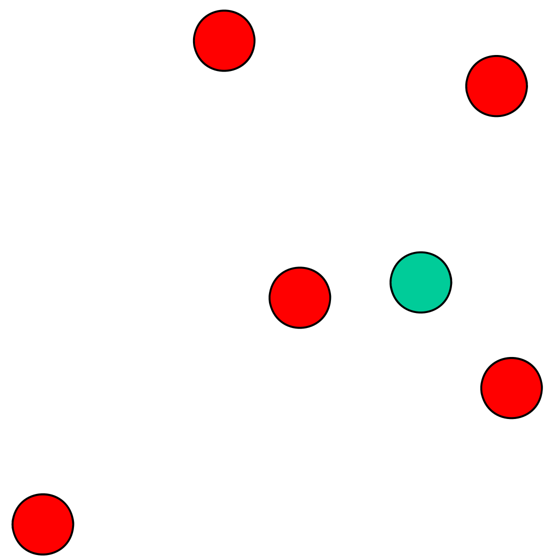
2-neares neighbors

Density: Neighbourhood Graphs



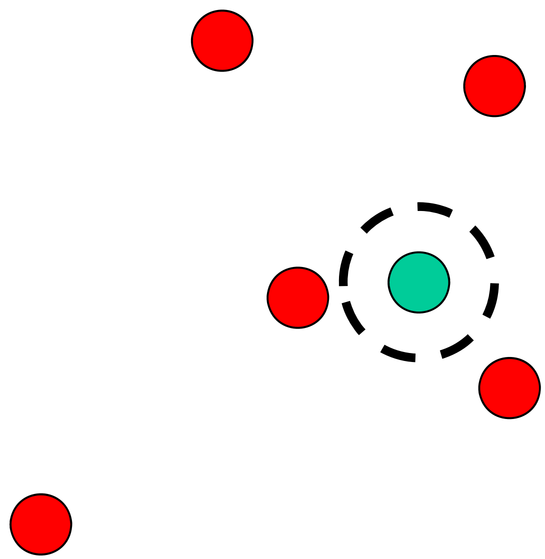
3-nearest neighbors

Density: Neighbourhood Graphs



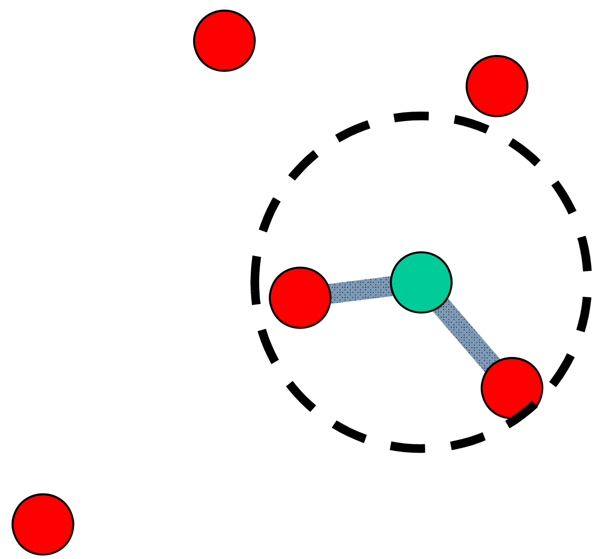
Radius-based distances in input space

Density: Neighbourhood Graphs



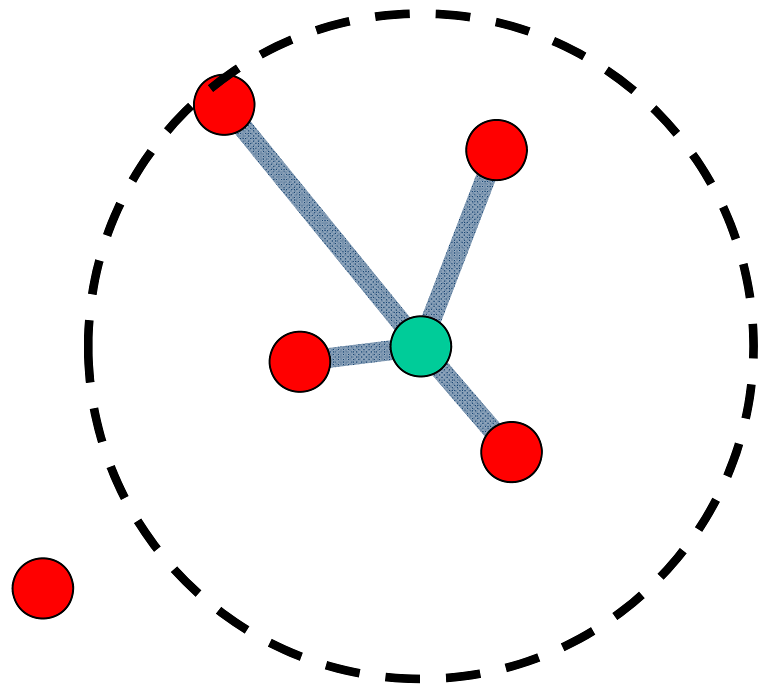
Radius: 1.0

Density: Neighbourhood Graphs



Radius: 2.0

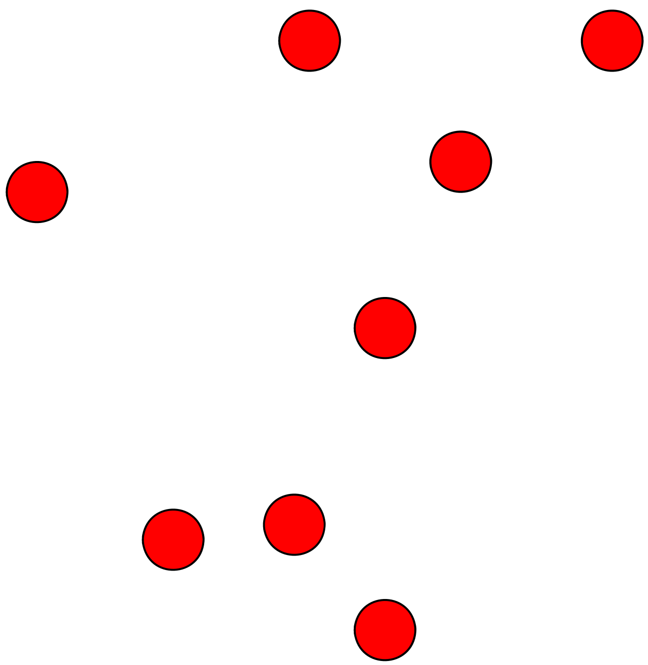
Density: Neighbourhood Graphs



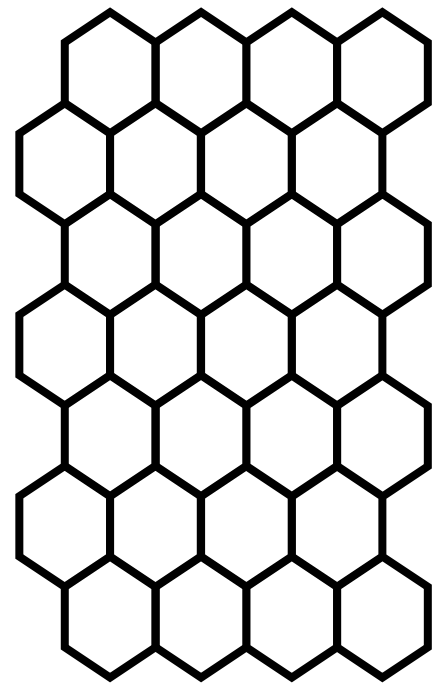
Radius: 3.0

Density: Neighbourhood Graphs

- **Projection:**



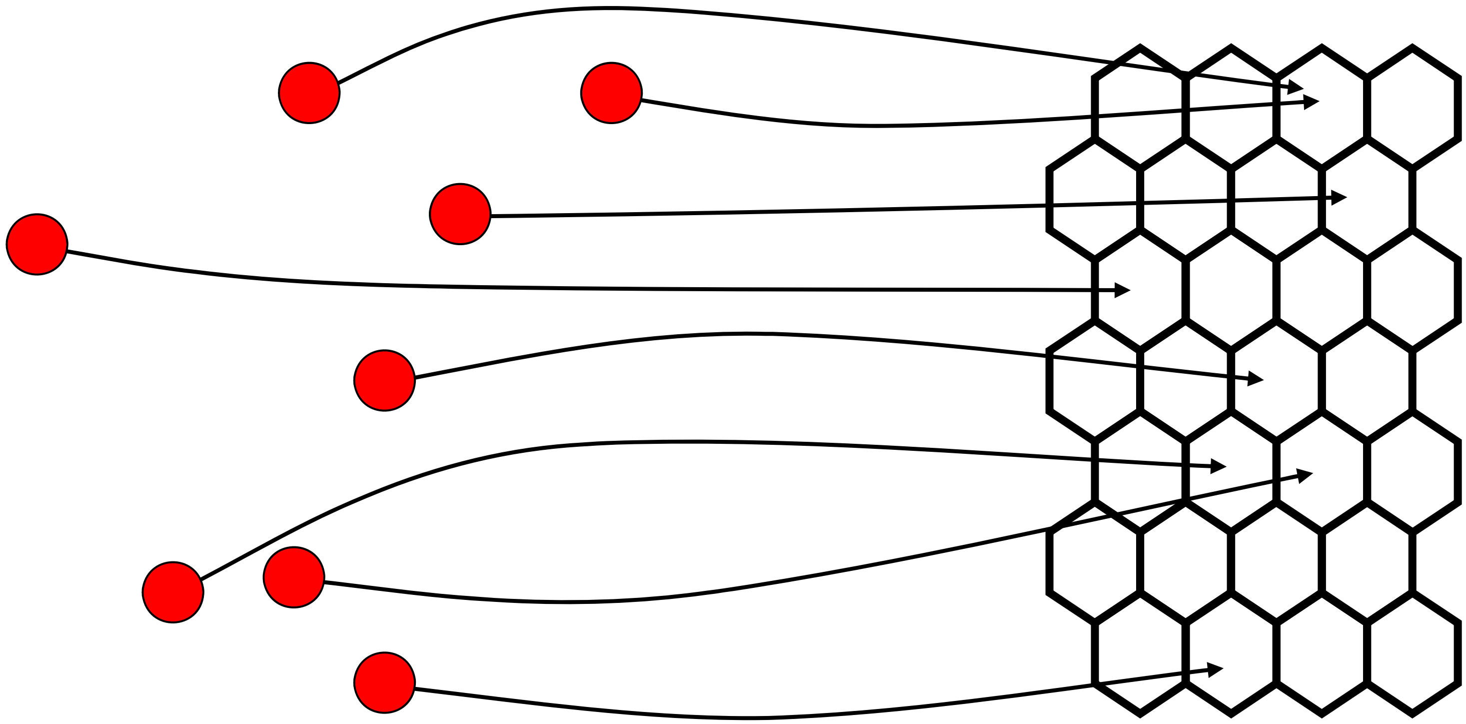
Input Space



Map Lattice

Density: Neighbourhood Graphs

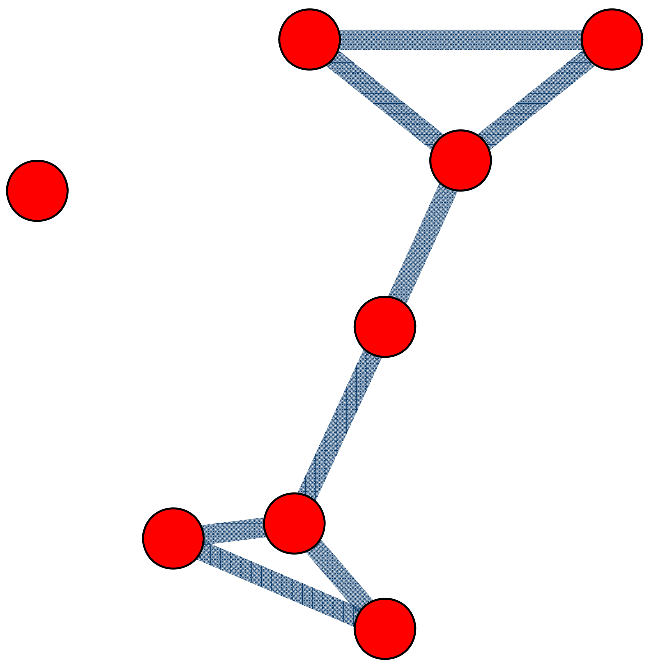
- **Projection:**



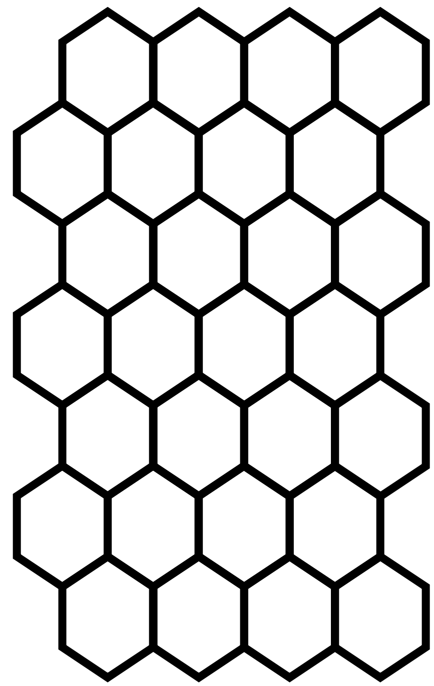
Input Space **Map Lattice**

Density: Neighbourhood Graphs

- **Projection:**



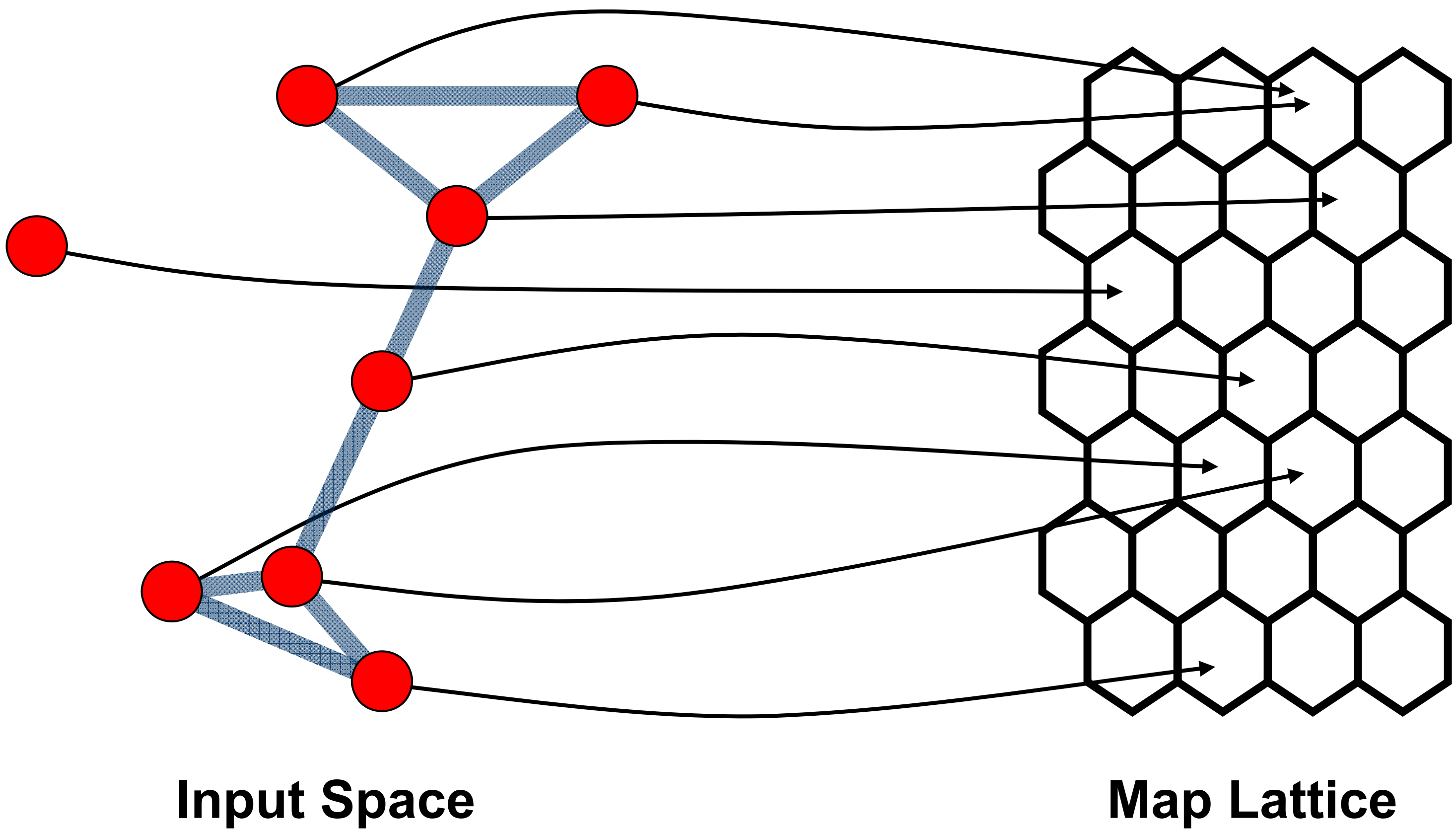
Input Space



Map Lattice

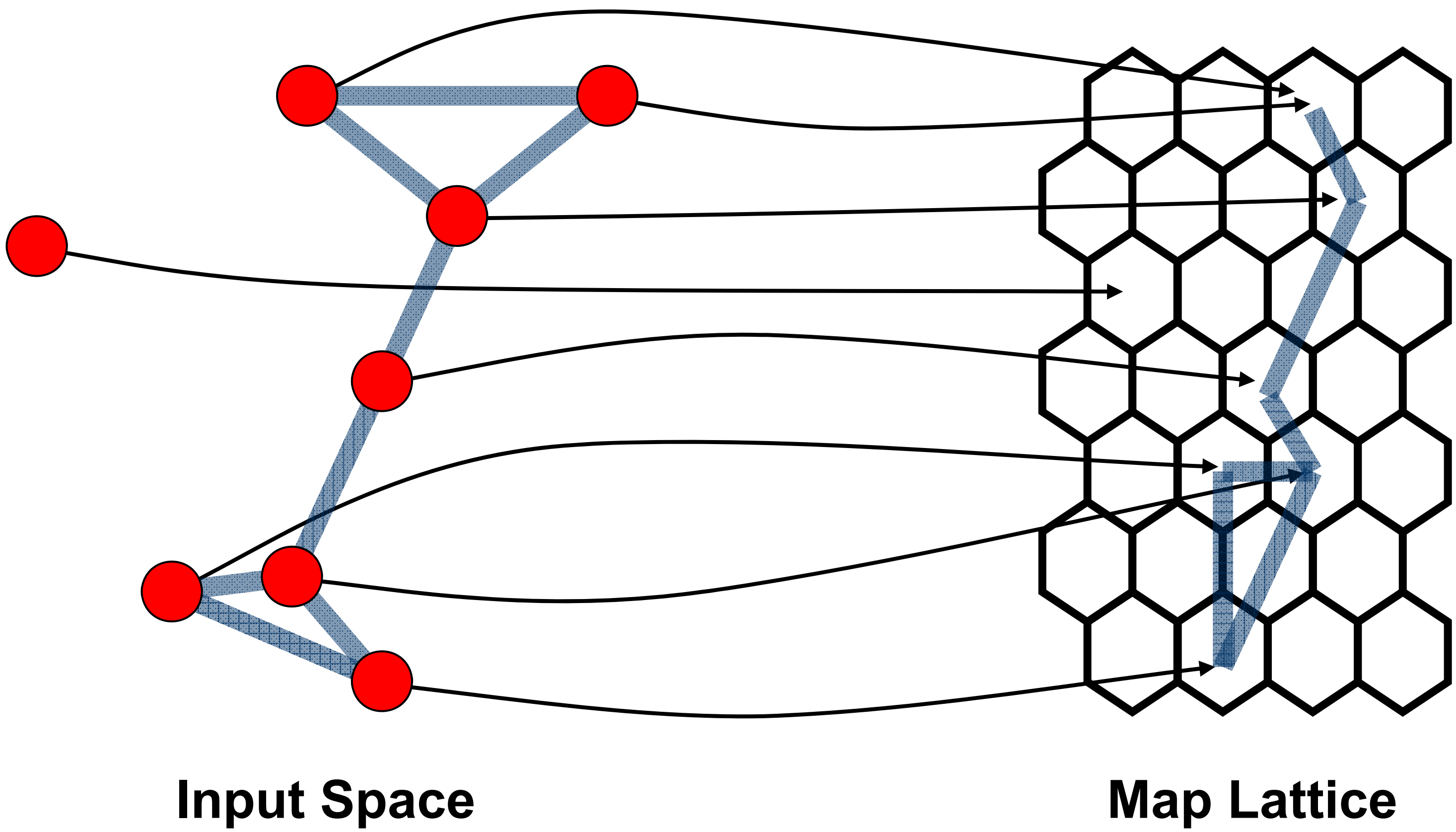
Density: Neighbourhood Graphs

- **Projection:**



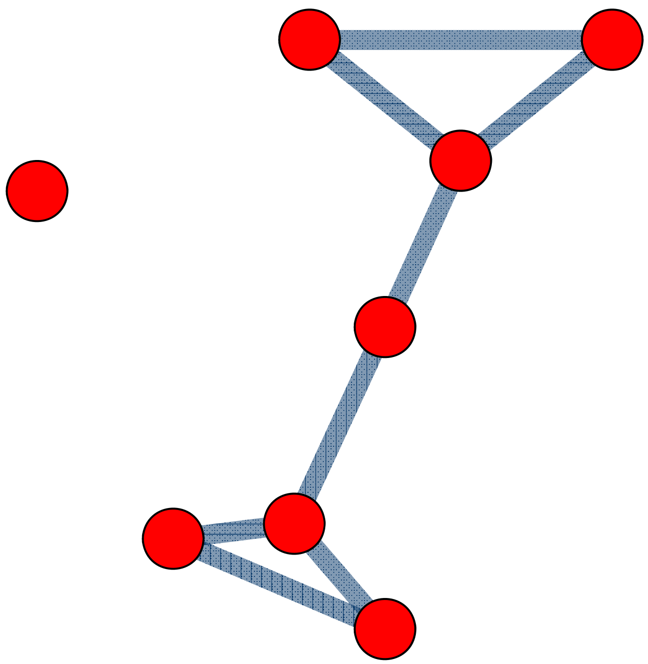
Density: Neighbourhood Graphs

- **Projection:**

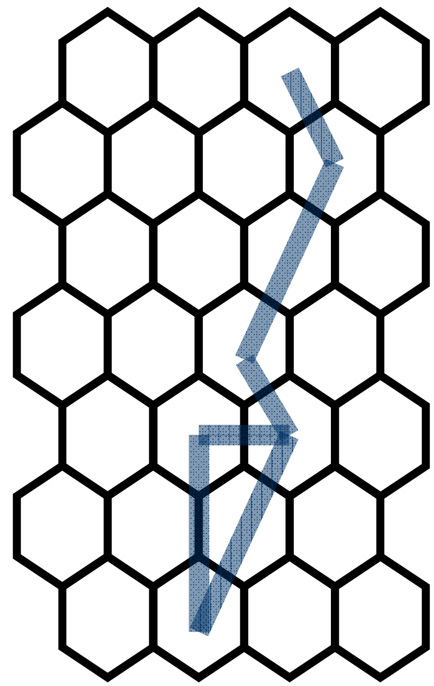


Density: Neighbourhood Graphs

- **Projection:**



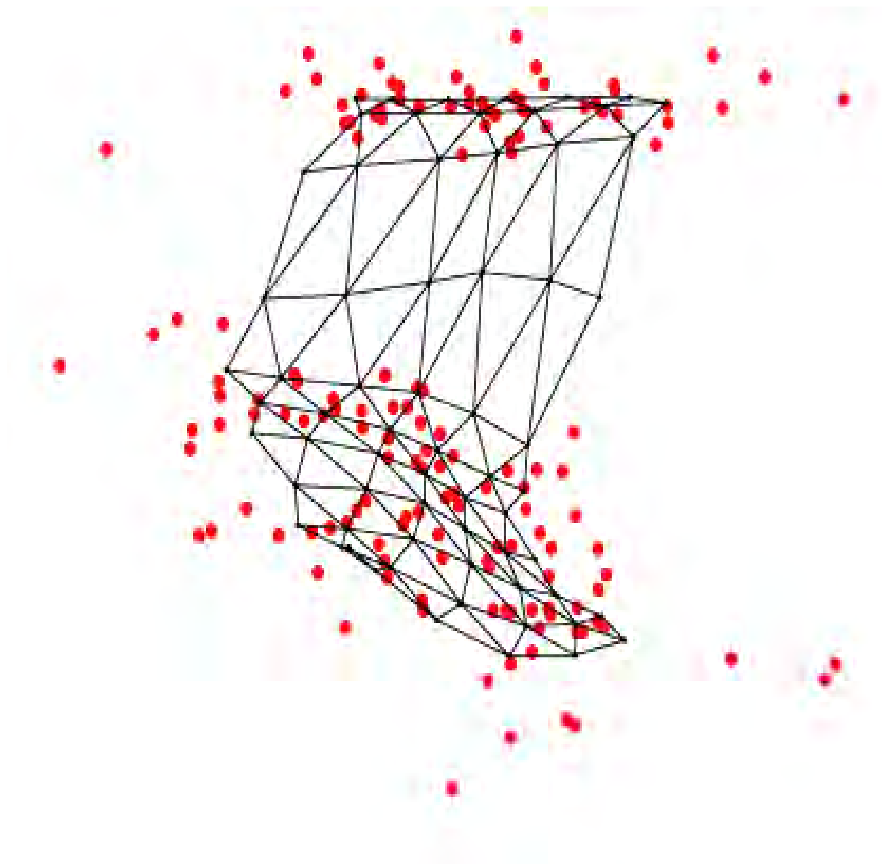
Input Space



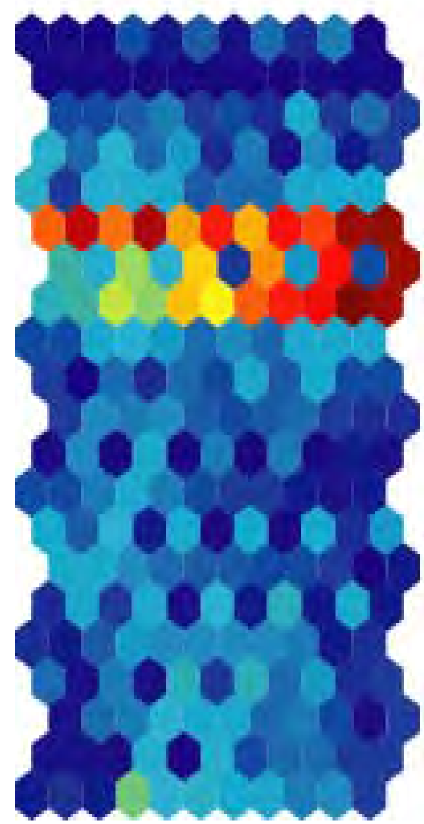
Map Lattice

Density: Neighbourhood Graphs

- **Example: Iris Dataset:**



PCA

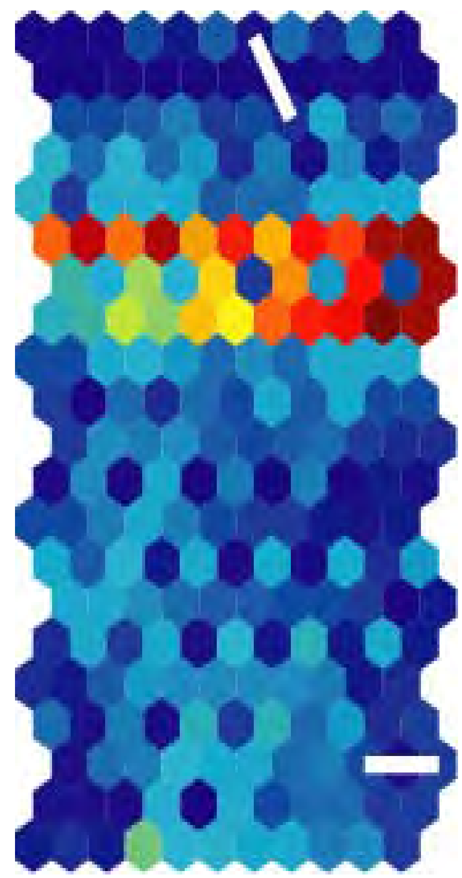


SOM

Density: Neighbourhood Graphs

- **Example: Iris Dataset:**

1-Nearest Neighbors



Radius: 0.2

Density: Neighbourhood Graphs

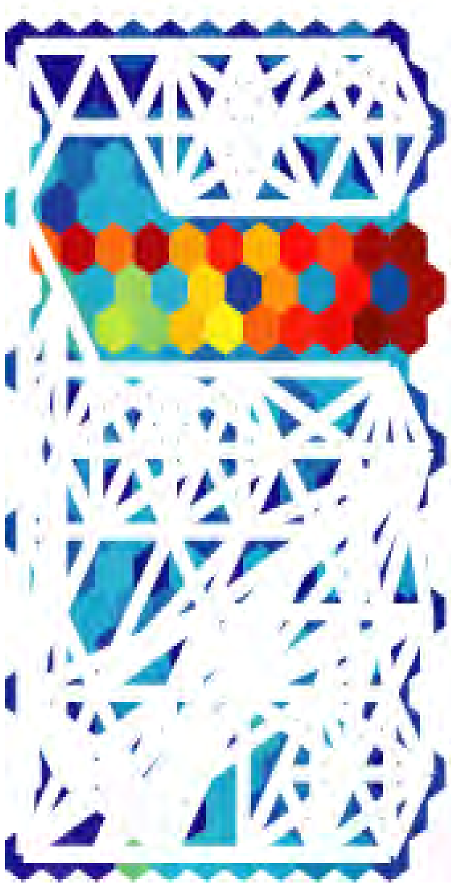
- **Example: Iris Dataset:**

3-Nearest Neighbors

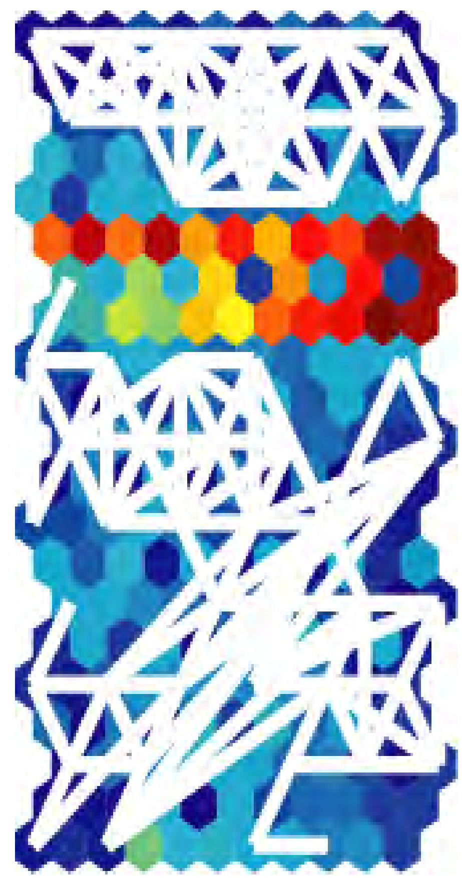
Radius: 0.4

Density: Neighbourhood Graphs

- **Example: Iris Dataset:**



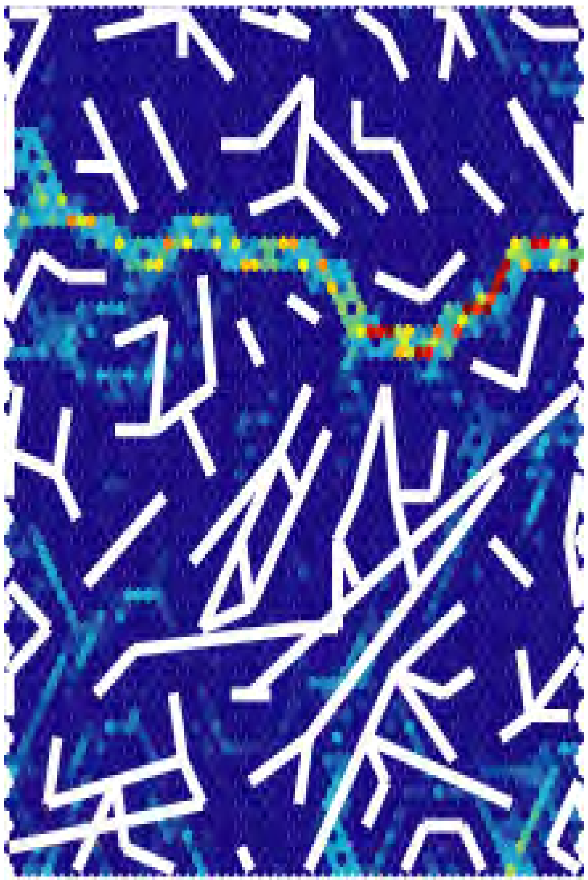
10-Nearest Neighbors



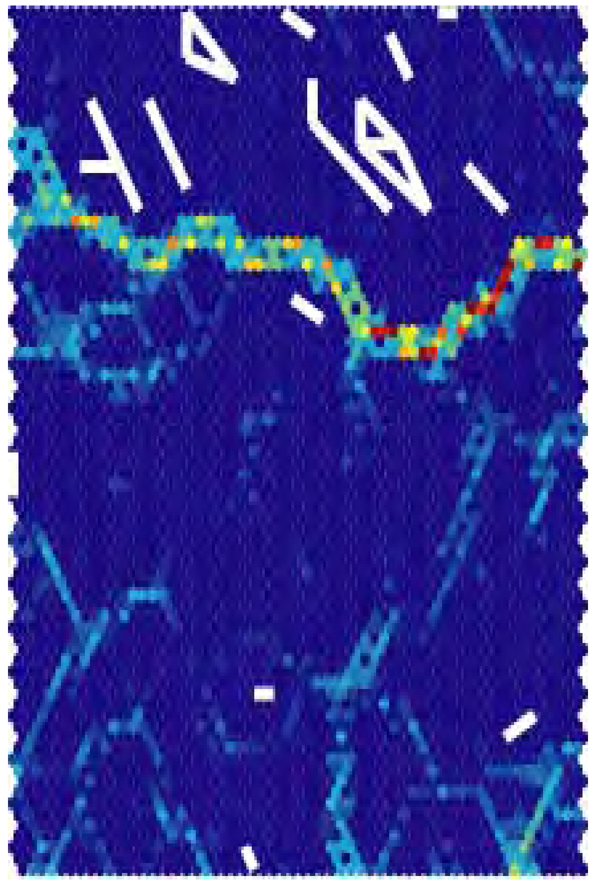
Radius: 0.6

Density: Neighbourhood Graphs

- **Example: Iris Dataset:**



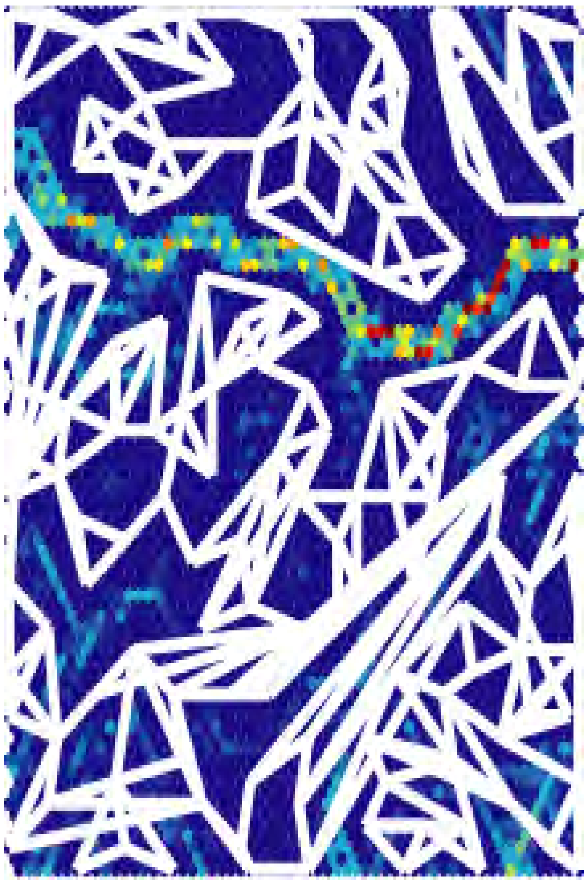
1-Nearest Neighbors



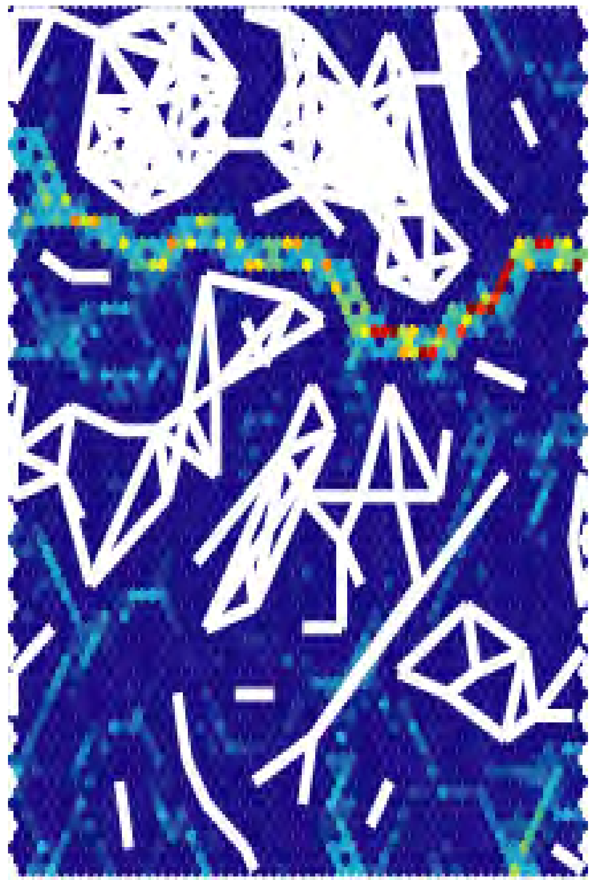
Radius: 0.2

Density: Neighbourhood Graphs

- **Example: Iris Dataset:**



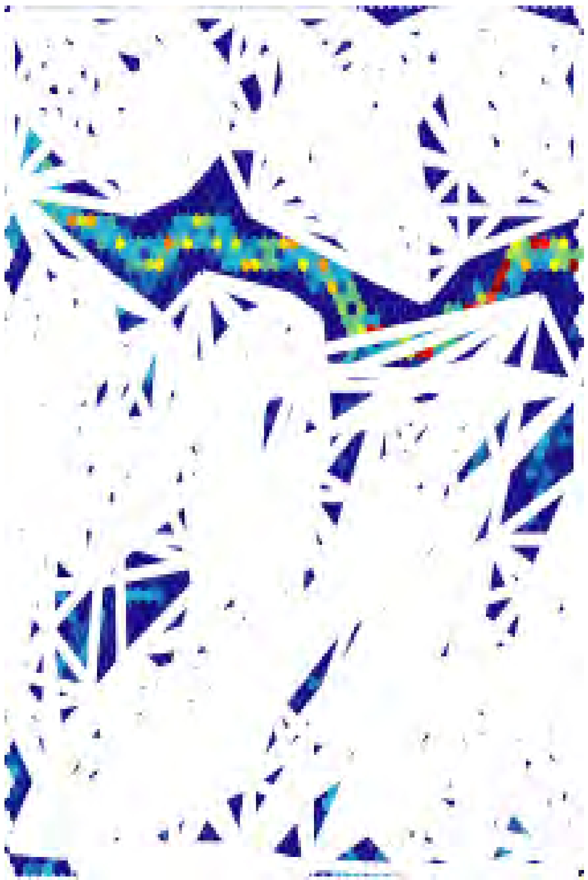
3-Nearest Neighbors



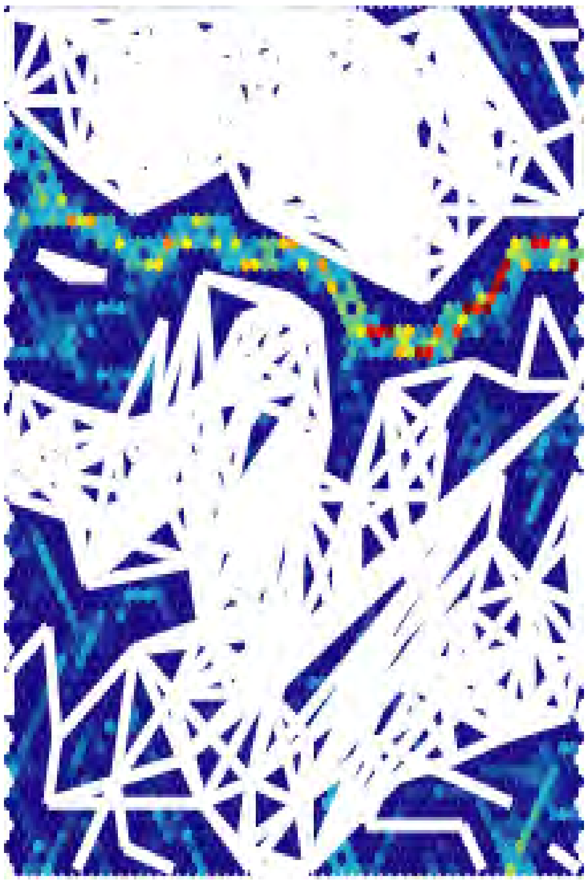
Radius: 0.4

Density: Neighbourhood Graphs

- **Example: Iris Dataset:**



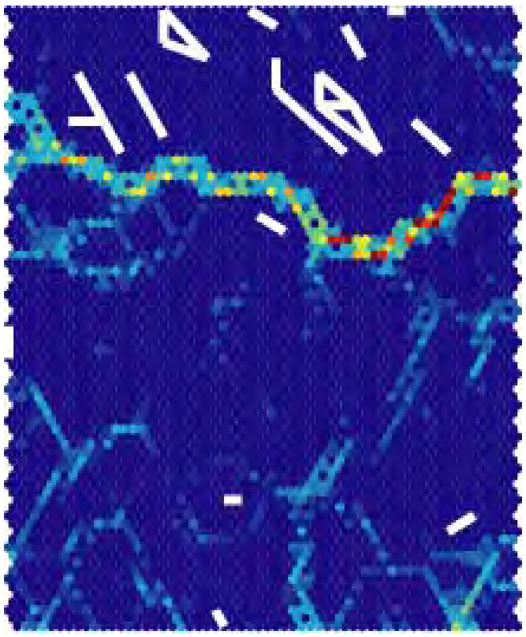
10-Nearest Neighbors



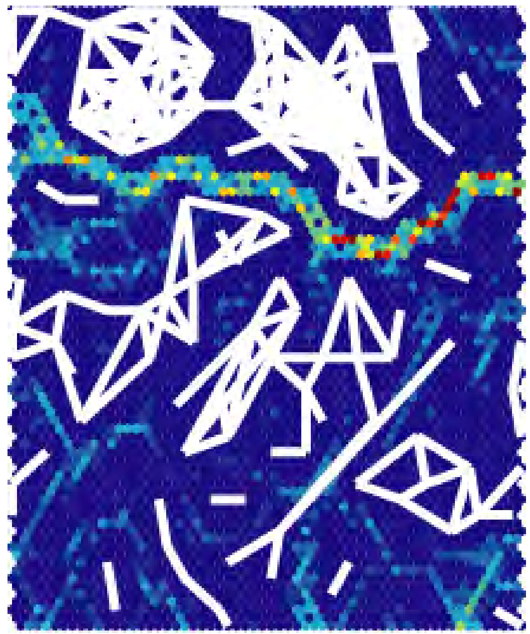
Radius: 0.6

Density: Neighbourhood Graphs

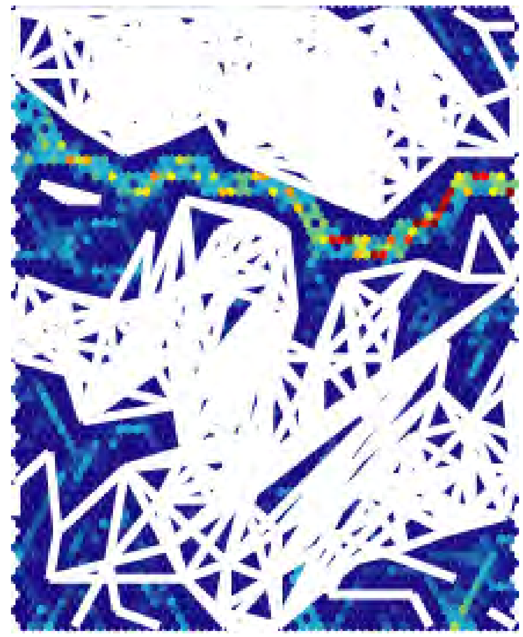
- **Example: Iris Dataset:**



Radius: 0.2



Radius: 0.4



Radius: 0.6

Density: Neighbourhood Graphs

- Show relationship between input & map space
- Based on topology, but (also) reveals cluster densities
- Radius method: focus on density
  - for clusters of different cardinality
- Nearest neighbors: focus on cluster cardinality
  - for clusters with different densities
- High parameter values:
  show clusters and their connections
- Long lines: **topology violation**
  (quality indicator)

Neighborhood Graphs

Questions

- What's the difference between using knn or the radius method for determining neighborhood?
- When will the two methods deliver different results?
- What can we deduce if the results differ?
- When will ´knn result in a dense graph, when the radius?
- What do long lines tell you that cross the entire map?

Outline

- Overview of visualization types
- Visualizing the SOM
  - Codebook projection
  - Adaptive Coordinates
- Visualizations on the SOM
  - Textual information
  - Density
  - Distances
  - Class info
  - Attributes
  - Clustering of the SOM

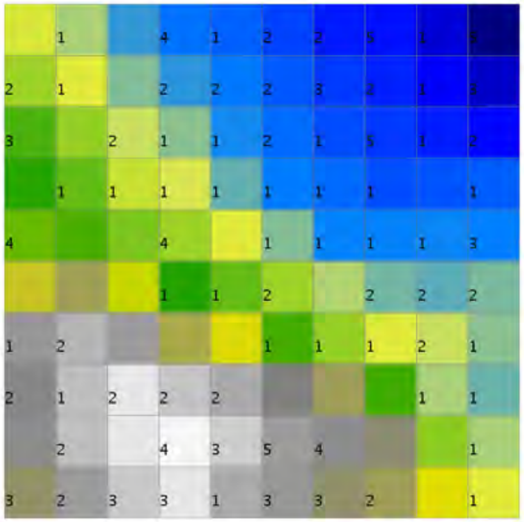
Outline

Visualizations on the SOM

- Textual information
- Density
- Distances
  - Activity Histograms
  - Minimum Spanning Trees
  - Cluster Connections (CC)
  - D-Matrix, U-Matrix
  - U* Matrix: U-Matrix + P-Matrix
  - Vectorfields: Flow / Borderline
- Class info
- Attributes
- Clustering of the SOM

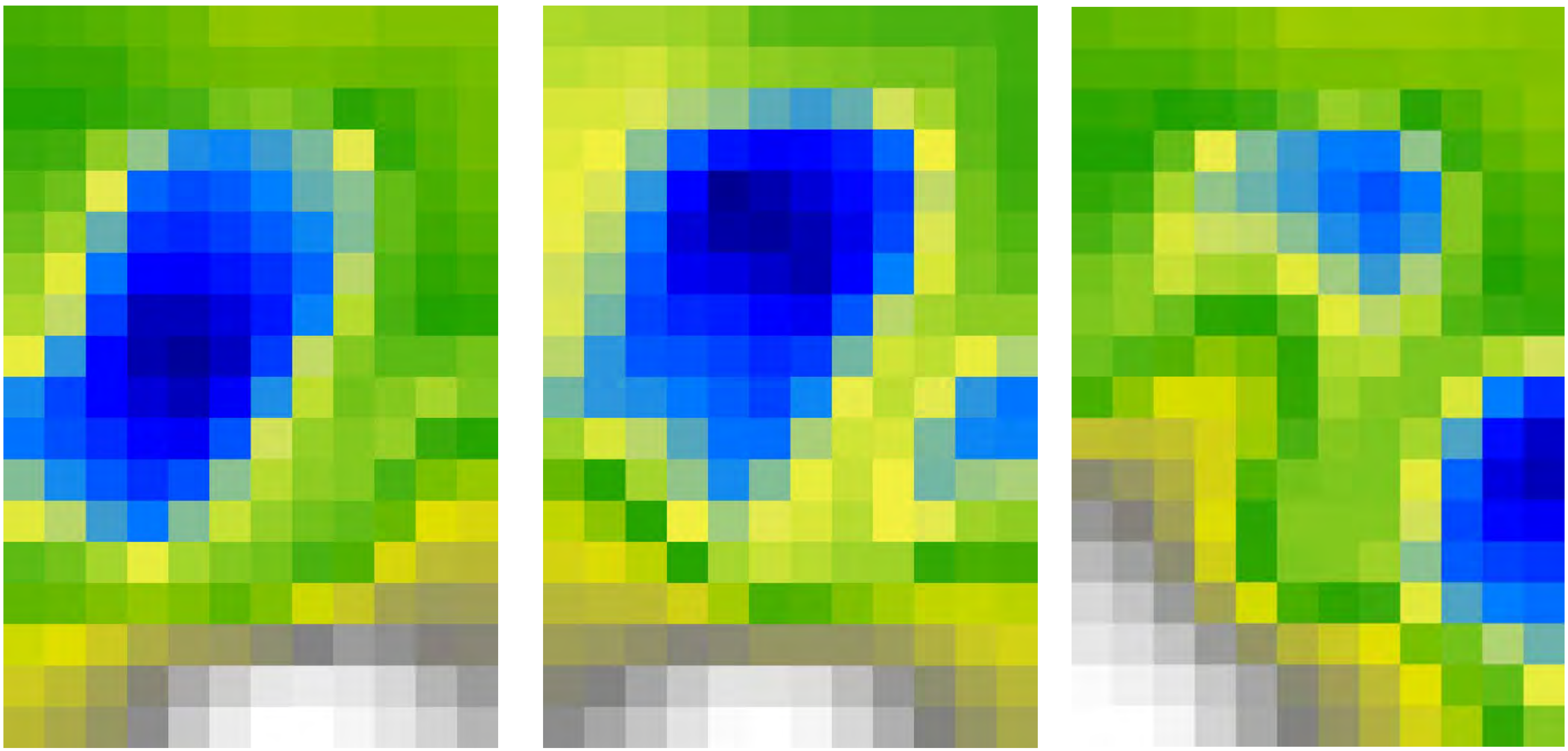
Activity Histogram

- Input vector is mapped onto best-matching unit
- How well fitting would the second or n-best matching unit have been? Where are they located?
- Activity Histogram **per data point** visualizes distance between input vector and all weight vectors (codebook)
- Can reveal cluster homogeneity/ topology violations -> **how?**
- What can we learn about cluster structures and shapes?



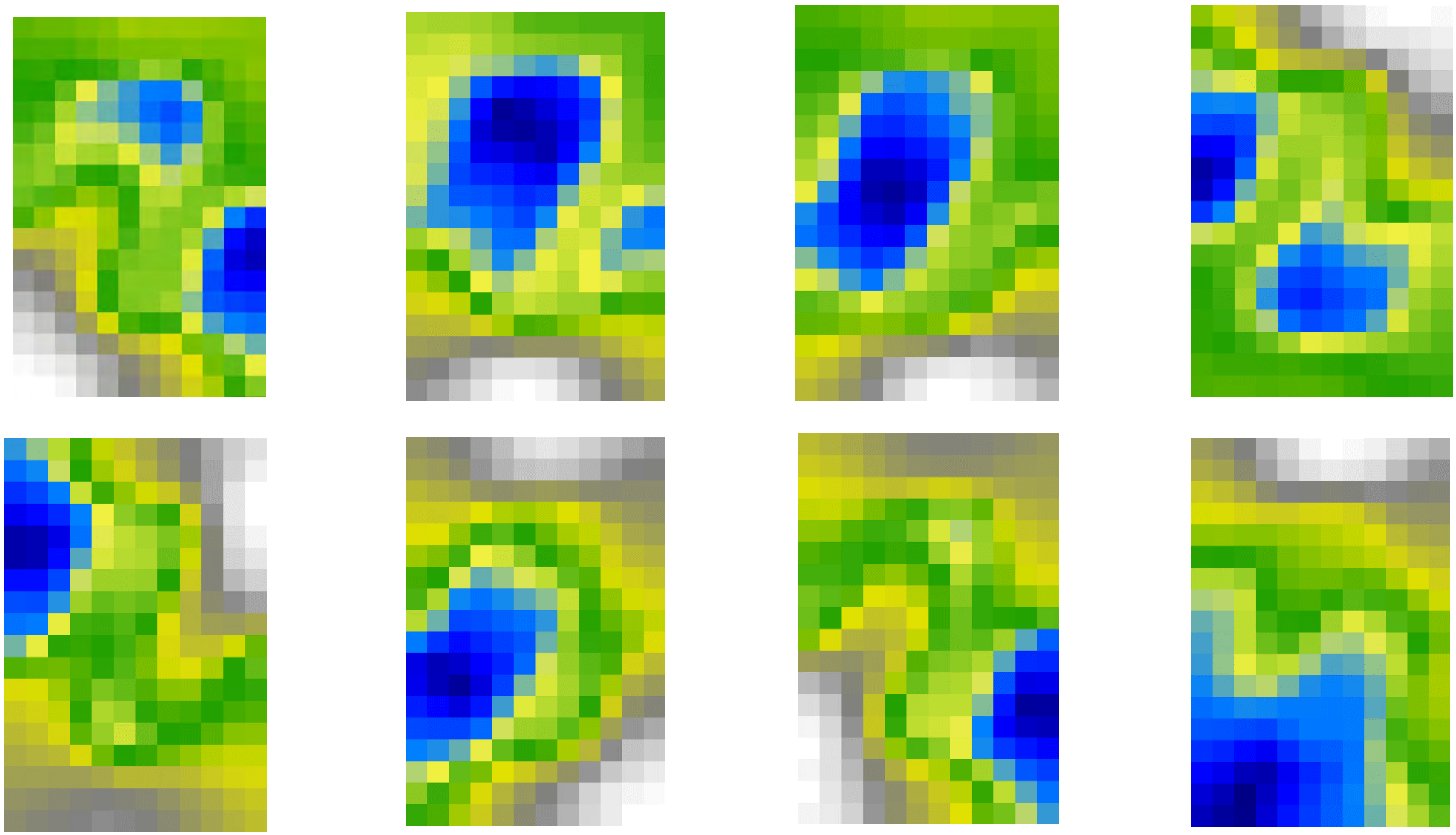
Activity Histogram

- Plot activation per input vector onto map



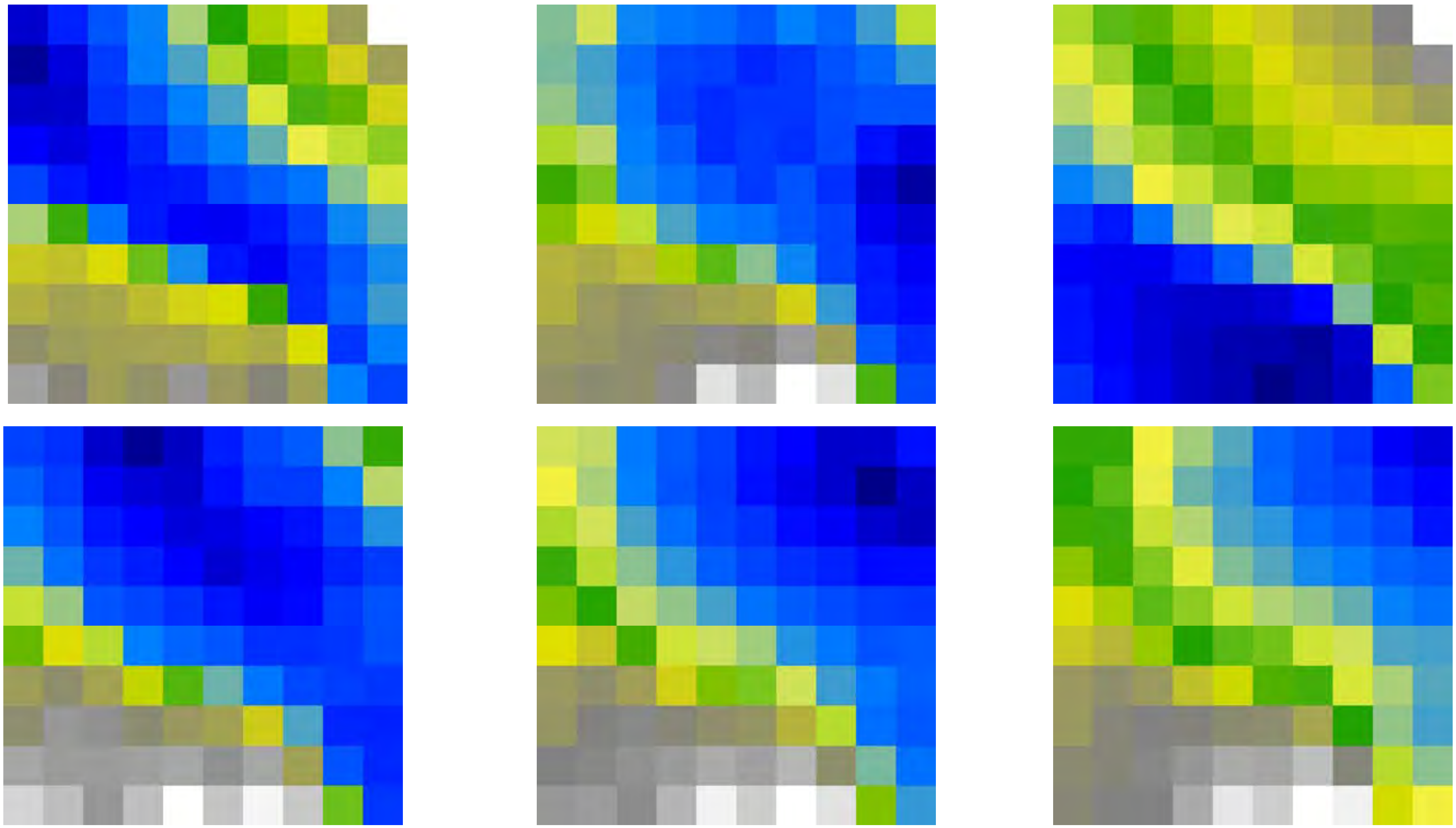
Activity Histogram

- Plot activation per input vector onto map



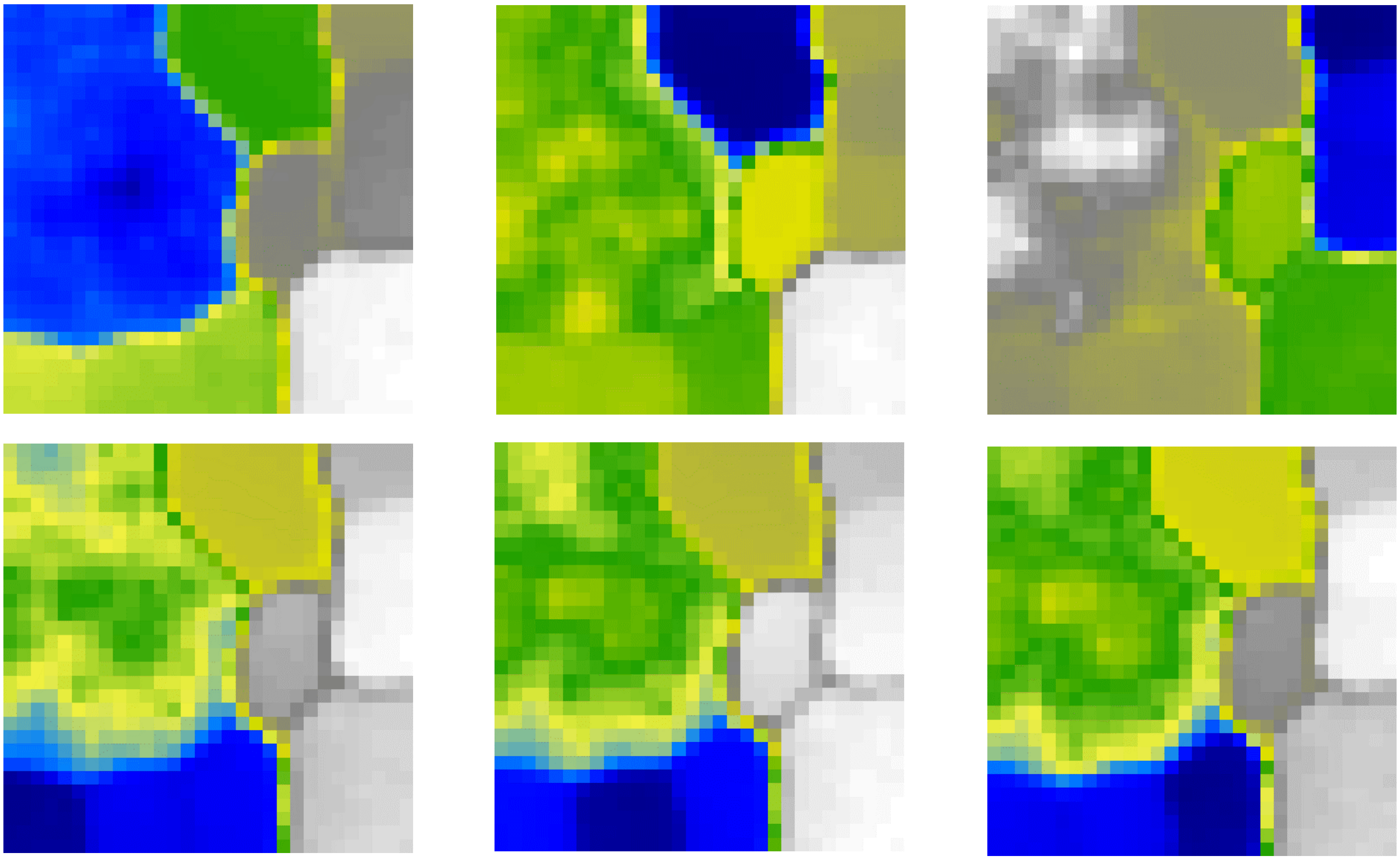
Activity Histogram

- Iris Dataset



Activity Histogram

- 10-Gaussians Dataset



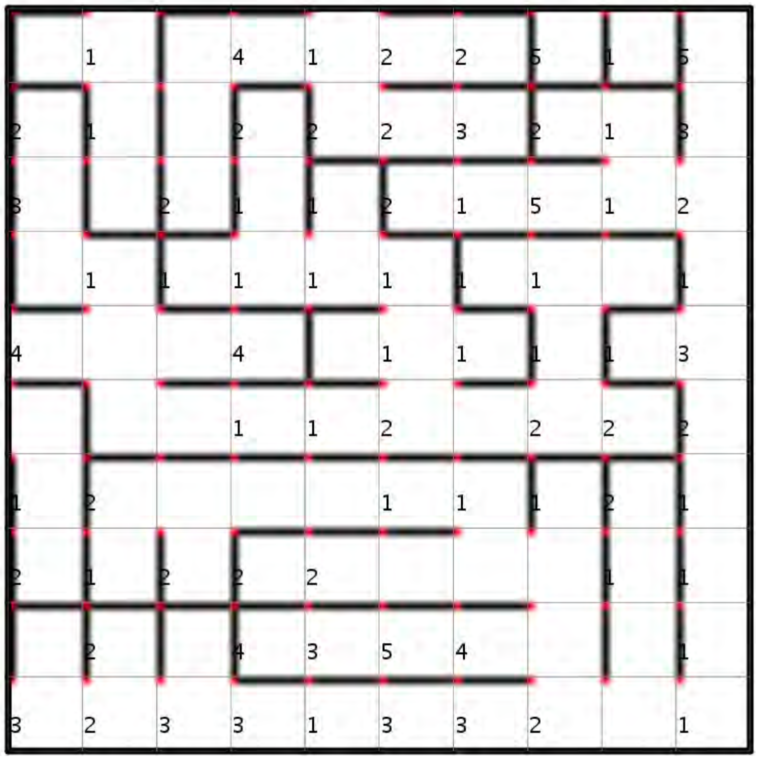
Outline

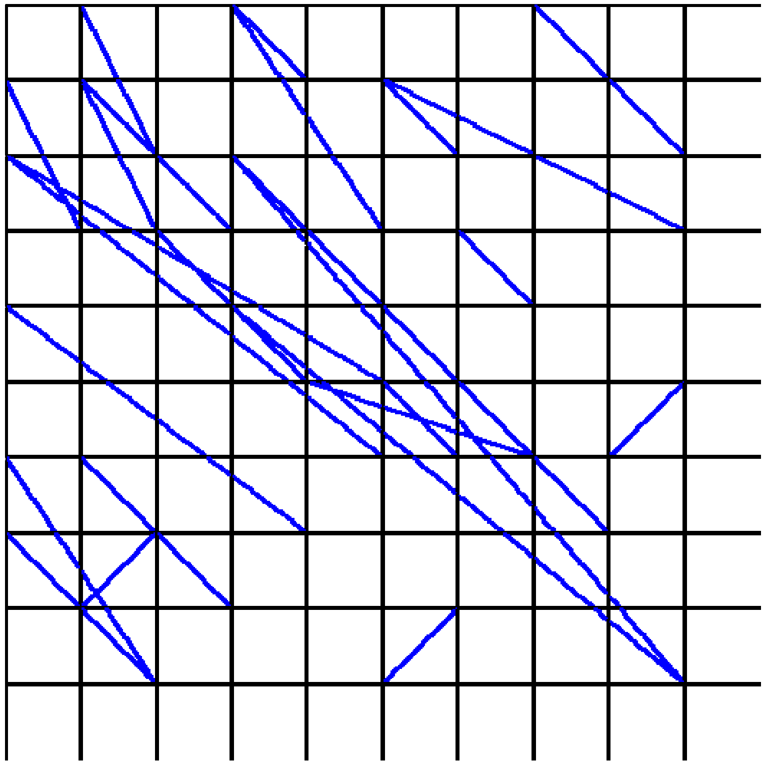
Visualizations on the SOM

- Textual information
- Density
- Distances
  - Activity Histograms
  - Minimum Spanning Trees
  - Cluster Connections (CC)
  - D-Matrix, U-Matrix
  - U* Matrix: U-Matrix + P-Matrix
  - Vectorfields: Flow / Borderline
- Class info
- Attributes
- Clustering of the SOM

Distances: MSPT

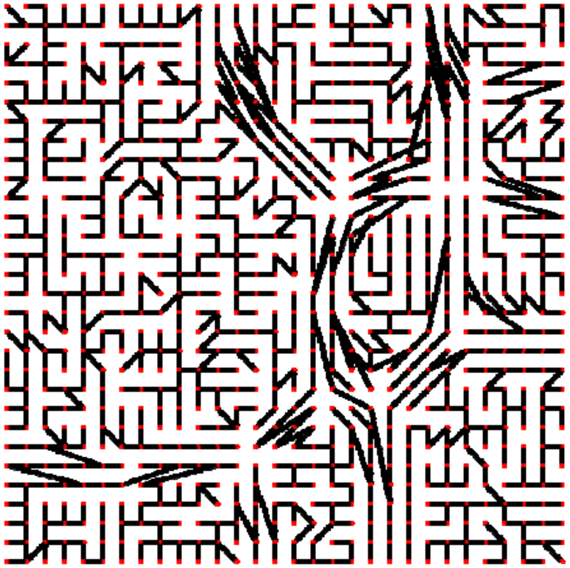
- Minimum Spanning Tree (Iris Data)
  - MSPT of input data projected onto SOM
  - MSPT of weight vectors



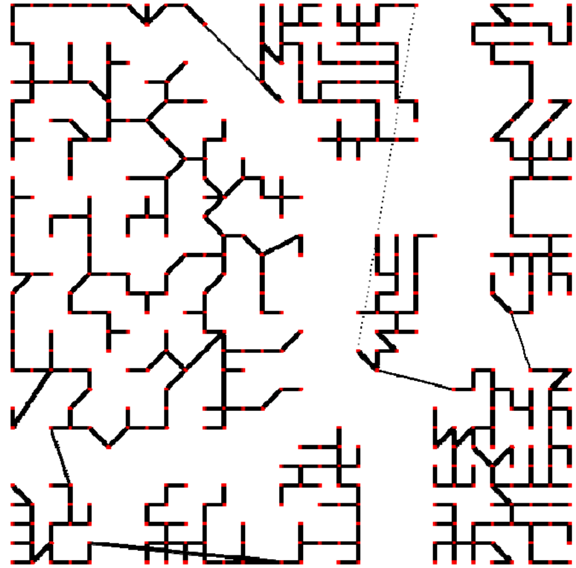


Distances: MSPT

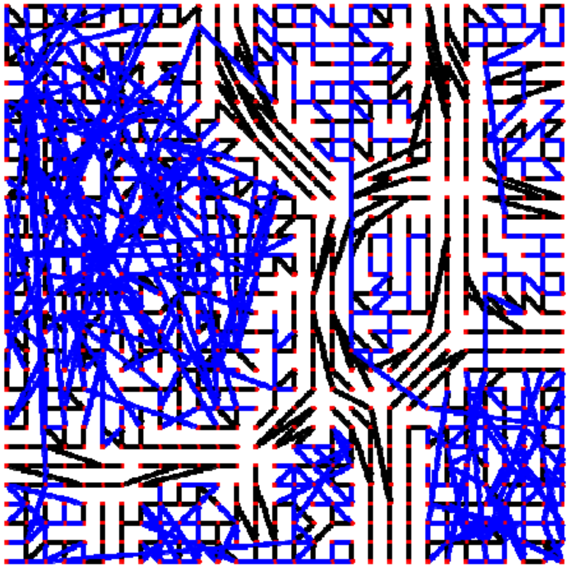
- Minimum Spanning Tree (10-Clusters Data)



MST of units



MST of units weighted linewidth



MST of units and data

Outline

Visualizations on the SOM

- Textual information
- Density
- Distances
  - Activity Histograms
  - Minimum Spanning Trees
  - Cluster Connections (CC)
  - D-Matrix, U-Matrix
  - U* Matrix: U-Matrix + P-Matrix
  - Vectorfields: Flow / Borderline
- Class info
- Attributes
- Clustering of the SOM

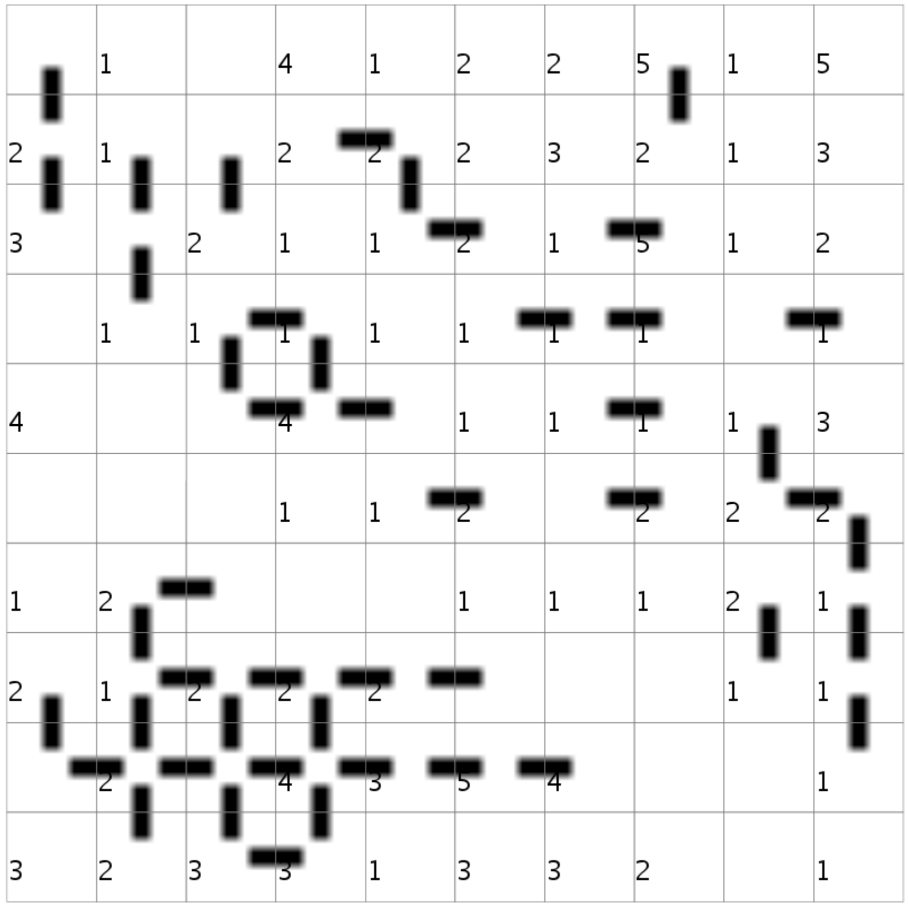
Distances

- Calculating the distance in input space between neighboring units
  - small distances: similar area of data space
  - large disances: located far apart in input space -> cluster boundaries
- Different approaches
  - Cluster Connections (CC)
  - D-Matrix
  - U-Matrix
  - U* Matrix: U-Matrix + P-Matrix
  - Vectorfields: Flow / Borderline

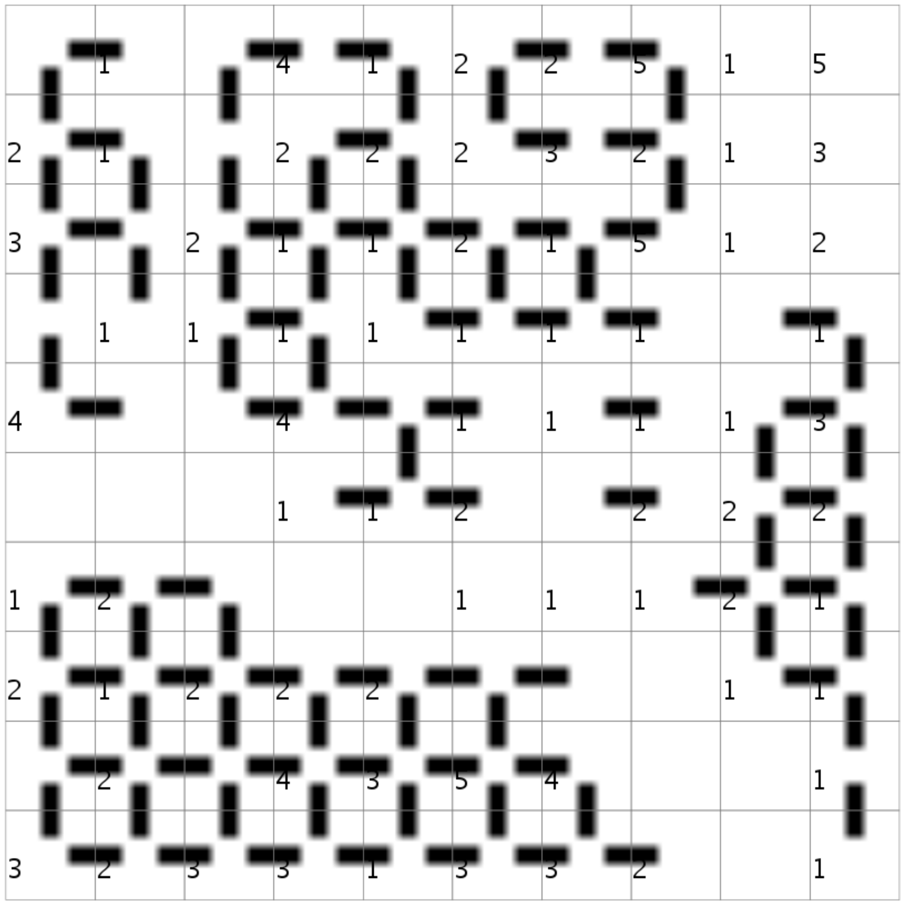
Distances: CC

- Cluster Connections (CC)
- Calculate distance between neighboring units
- Draw connecting lines between units according to thresholds
- Graph
- Allows interactive exploration of cluster structure
- Dieter Merkl, Andreas Rauber: Cluster Connections -- A visualization technique to reveal cluster boundaries in self-organizing maps. In: Proc 9th Italian Workshop on Neural Nets (WIRN97), Vietri sul Mare, Italy, Springer, 1997.

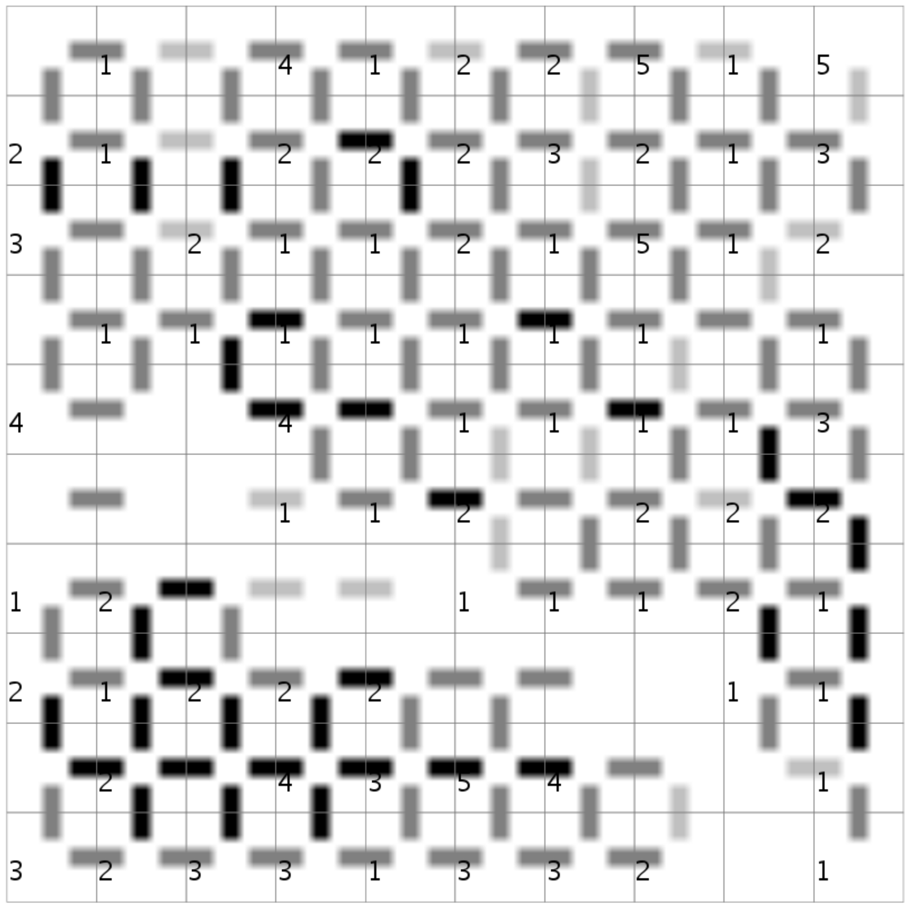
Distances: CC



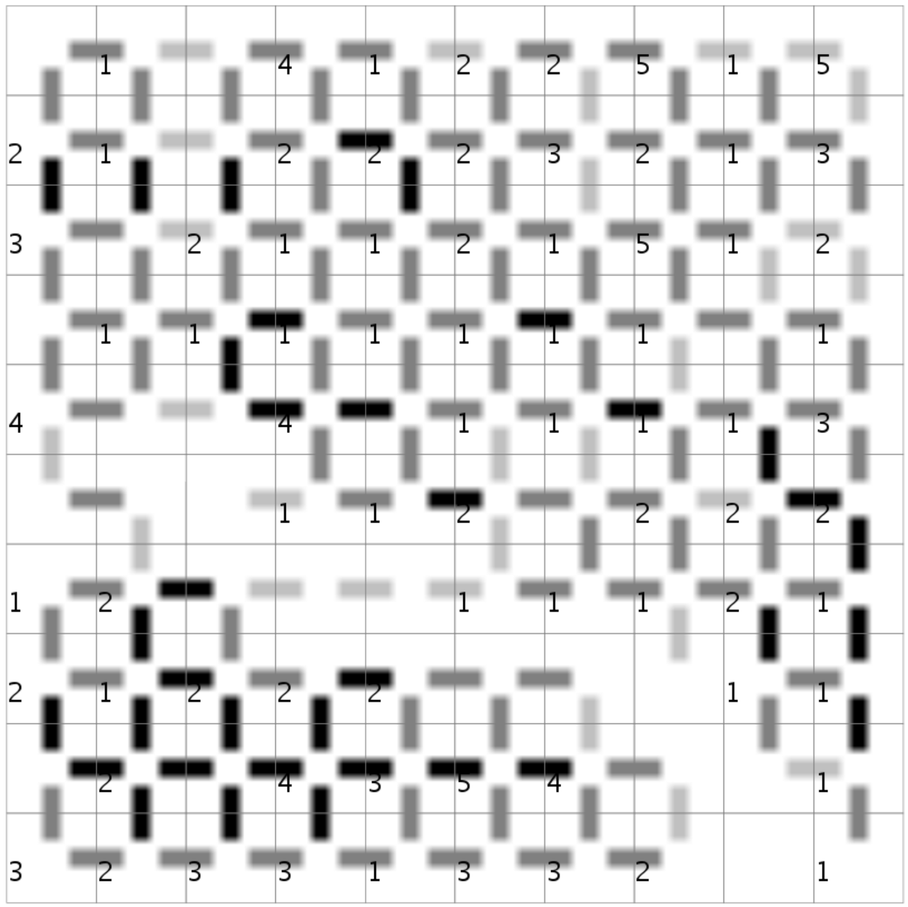
Distanzen: CC



Distanzen: CC



Distanzen: CC



Outline

Visualizations on the SOM

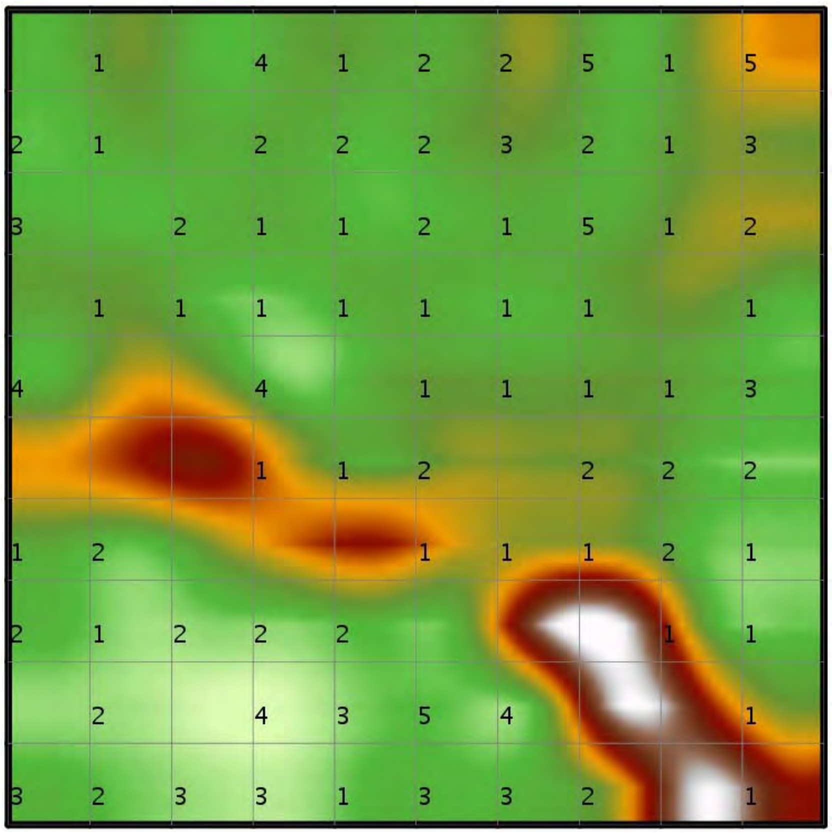
- Textual information
- Density
- Distances
  - Activity Histograms
  - Minimum Spanning Trees
  - Cluster Connections (CC)
  - D-Matrix, U-Matrix
  - U* Matrix: U-Matrix + P-Matrix
  - Vectorfields: Flow / Borderline
- Class info
- Attributes
- Clustering of the SOM

Distances: D/U-Matrix

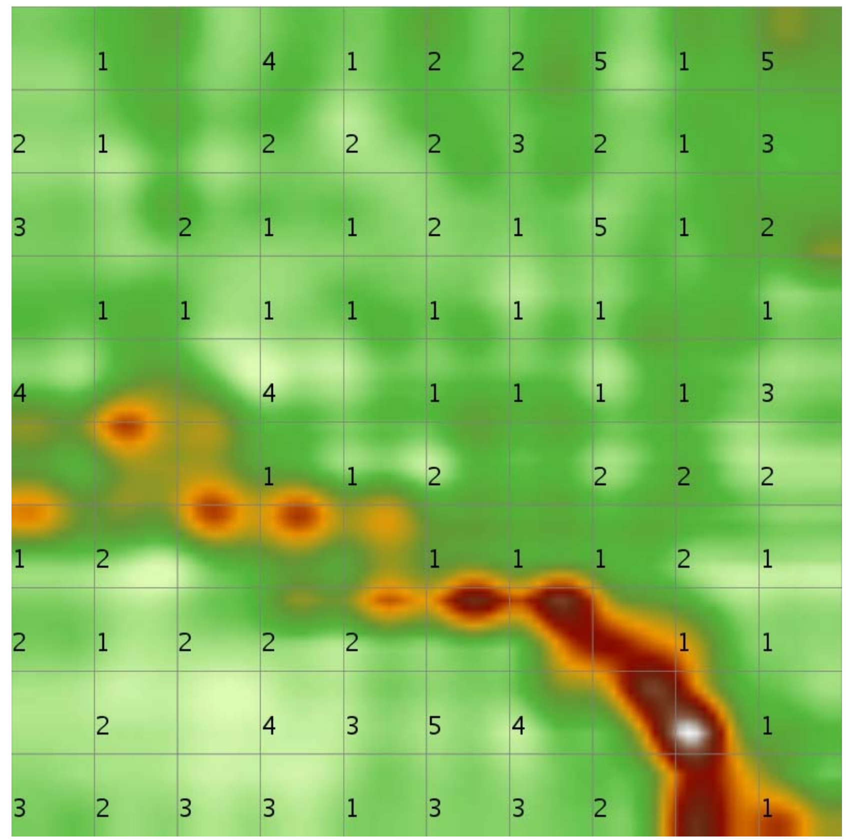
- Calculate distance between neighboring units
- D-Matrix: average distance to all neigboring units
- U-Matrix: distance to each neighboring units, interpolate
- reveal cluster structure
- „Mountains“ (high values): Cluster boundaries
- „Valleys“ (low values): coherent regions, clusters
- 2D or 3D visualization
- A. Ultsch.: Self-organizing neural networks for visualization and classification. In *Information and Classification. Concepts, Methods and Application*. Springer Verlag, 1993.

Distances: D/U-Matrix

- Iris Dataset:



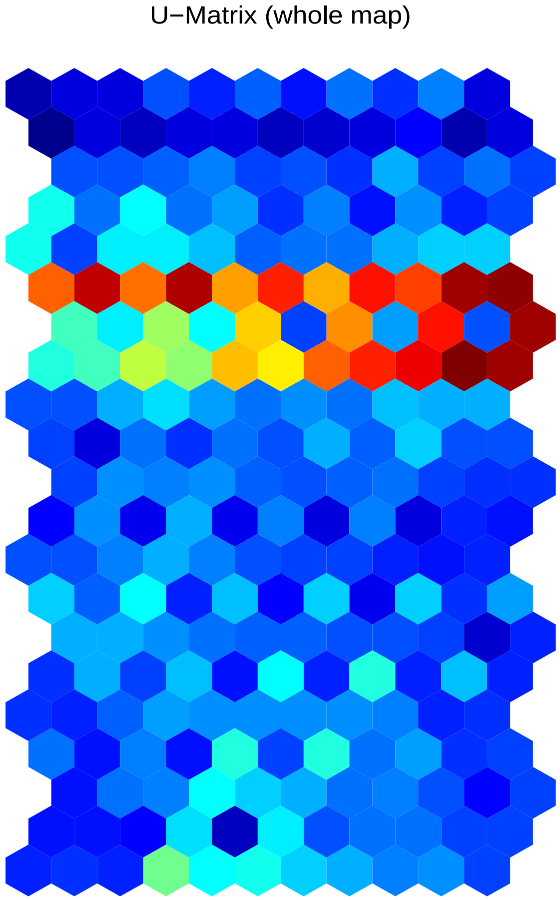
D-Matrix

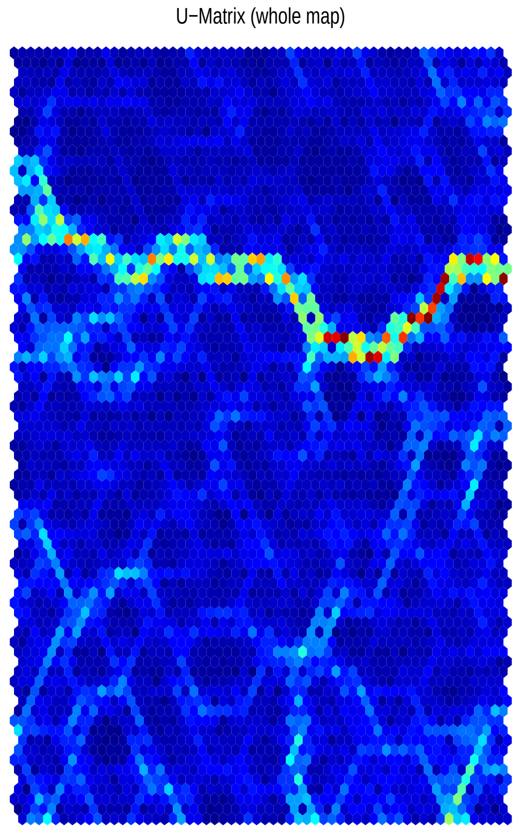


U-Matrix

Distances: U-Matrix

- Iris Dataset: small / large SOM





Outline

Visualizations on the SOM

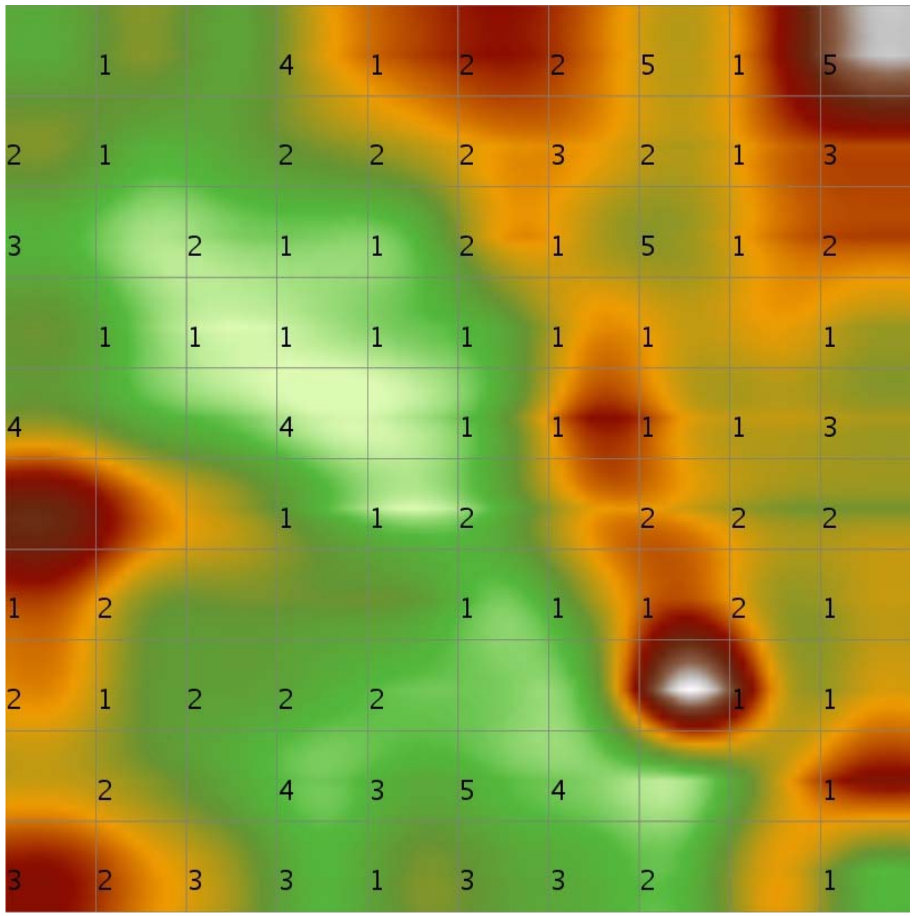
- Textual information
- Density
- Distances
  - Activity Histograms
  - Minimum Spanning Trees
  - Cluster Connections (CC)
  - D-Matrix, U-Matrix
  - U* Matrix: U-Matrix + P-Matrix
  - Vectorfields: Flow / Borderline
- Class info
- Attributes
- Clustering of the SOM

Distances + Density: U*

- P-Matrix: shows density of data on units
- U-Matrix: sows distances / cluster boundaries
- U*-Matrix: combination of U-Matrix and P-Matrix
- Ultsch A. U*-Matrix: A Tool to Visualize Cluster in High-Dimensional Data. Tech. Report, Dept. of Mathematics and Computer Science, University of Marburg.

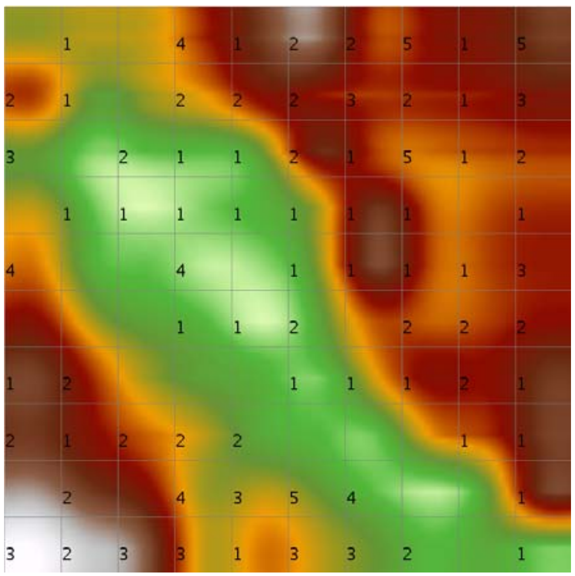
Distances + Density: U*

- Iris Dataset:

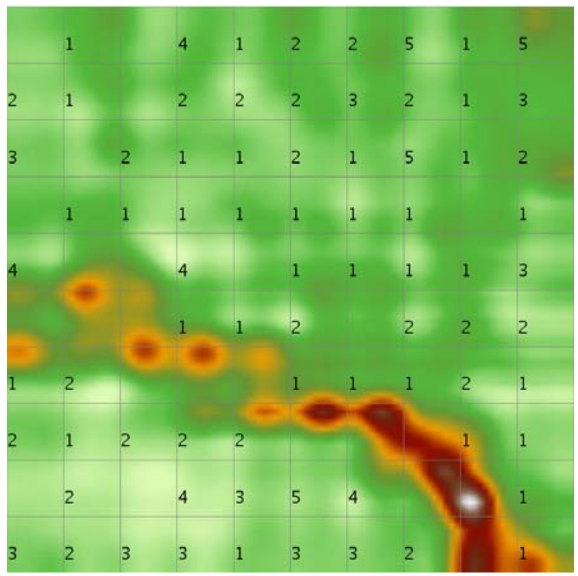


Distances + Density: U*

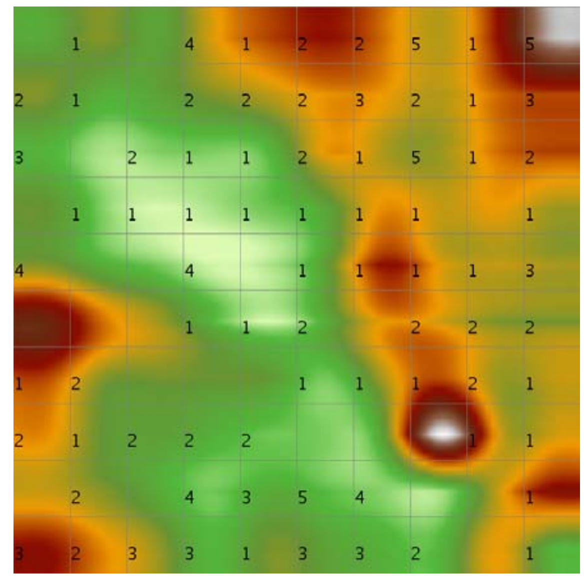
- Iris Dataset:



P-Matrix



U-Matrix



U*-Matrix

Distances + Density: U*

Questions

- which information can you reveal from U-matrix, which from P-matrix
- When is the U* matrix (more/most) useful?

Outline

Visualizations on the SOM

- Textual information
- Density
- Distances
  - Activity Histograms
  - Minimum Spanning Trees
  - Cluster Connections (CC)
  - D-Matrix, U-Matrix
  - U* Matrix: U-Matrix + P-Matrix
  - Vectorfields: Flow / Borderline
- Class info
- Attributes
- Clustering of the SOM

Distances: SOM Vector Fields

Vector field graphs show

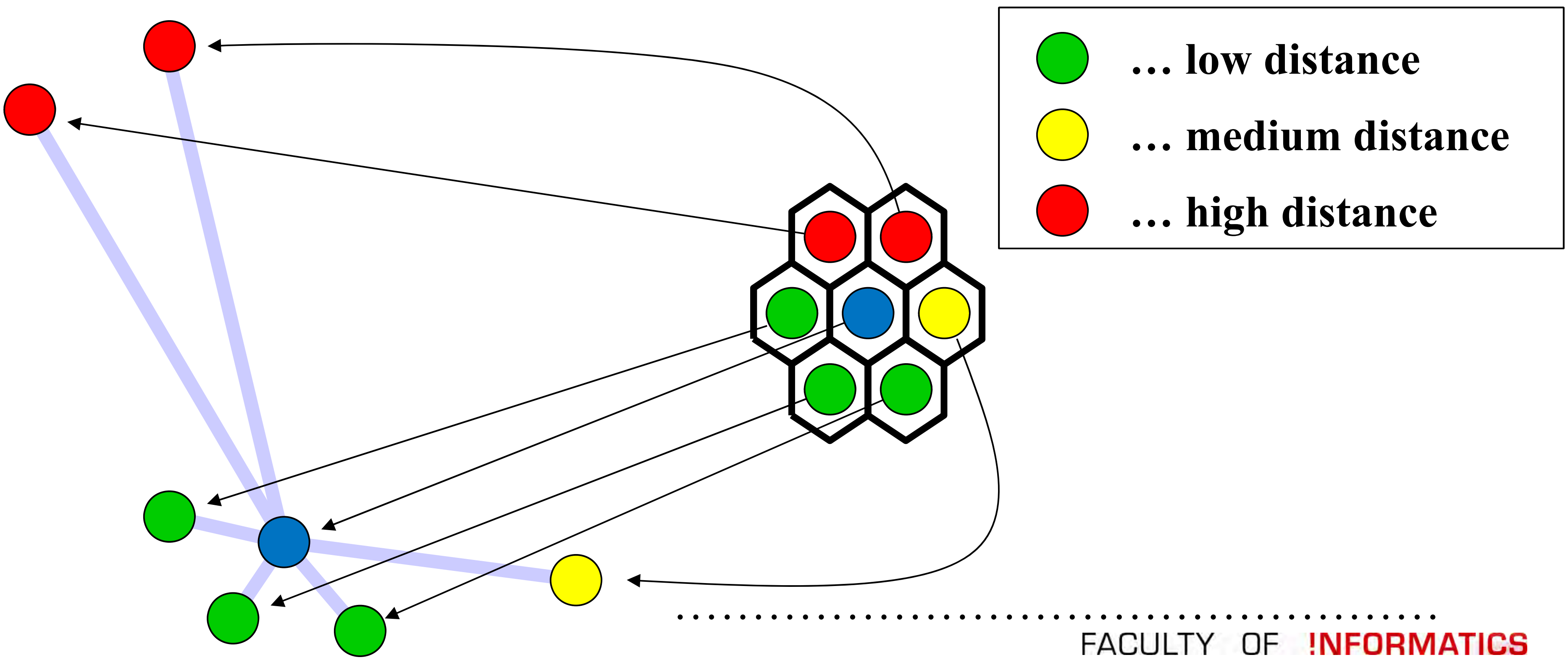
- which areas of the map are close to each other
- relationships of groups of attributes
- Georg Pölzlbauer, Andreas Rauber, and Michael Dittenbach. **A vector field visualization technique for self-organizing maps** In Tu Bao Ho, David Cheung, Huan Li, editors, Proceedings of the Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05), pages 399-409, Hanoi, Vietnam, May 18-20 2005. Springer-Verlag.
- Georg Pölzlbauer, Michael Dittenbach, Andreas Rauber. **Advanced visualization of Self-Organizing Maps with vector fields**. Neural Networks, 19(6-7):911-922, July-August 2006

SOM Vector Fields

- similar to U-matrix, but for pairs of units
- based solely on units' weight vectors
- different levels of granularity (interactive analysis)
- optimized for engineering disciplines

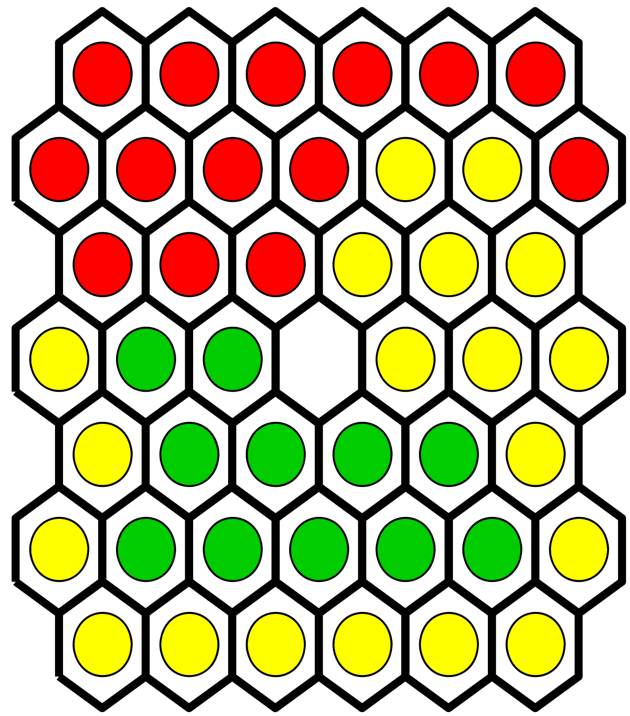
Vector Field Principles

- Vector field representation
- Vectors pointing to cluster centers
- Smoother version of U-Matrix



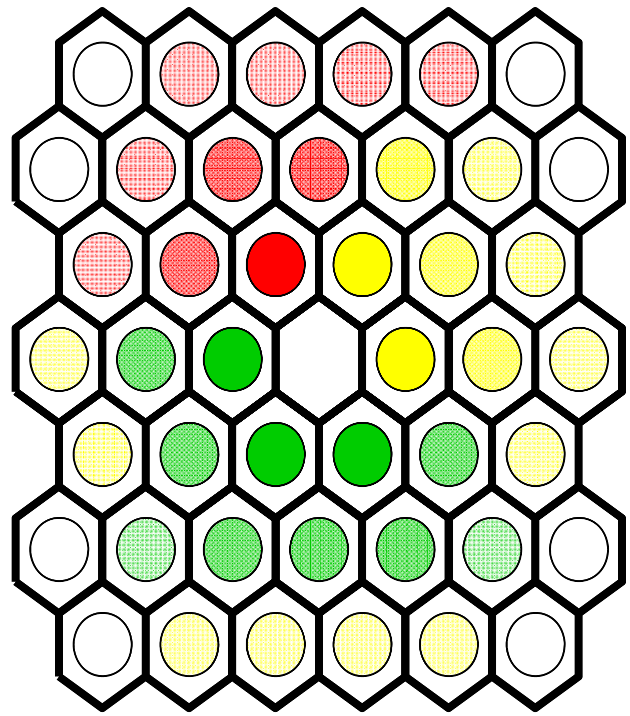
Flow Computation

1. calculate distances



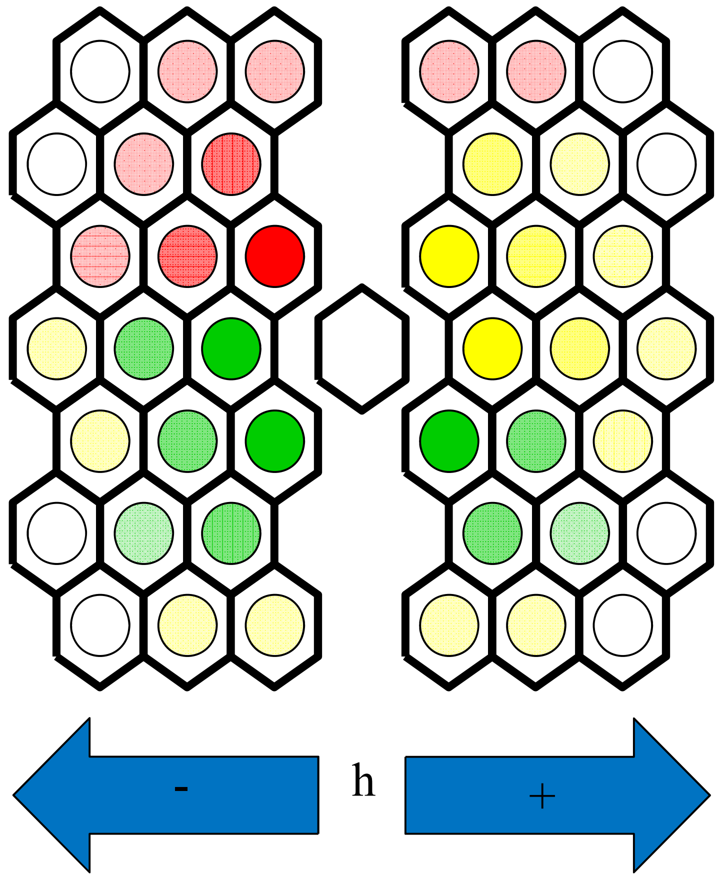
Flow Computation

1. calculate distances
2. weight with kernel function
   (different levels of granularity)



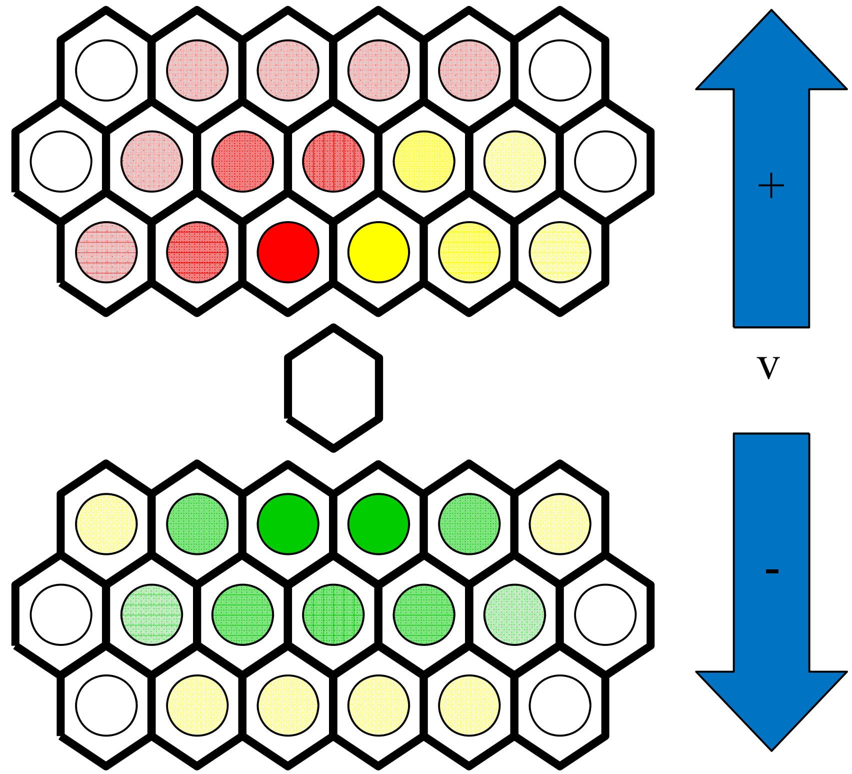
Flow Computation

1. calculate distances
2. weight with kernel function
3. divide into positive and negative h/v directions



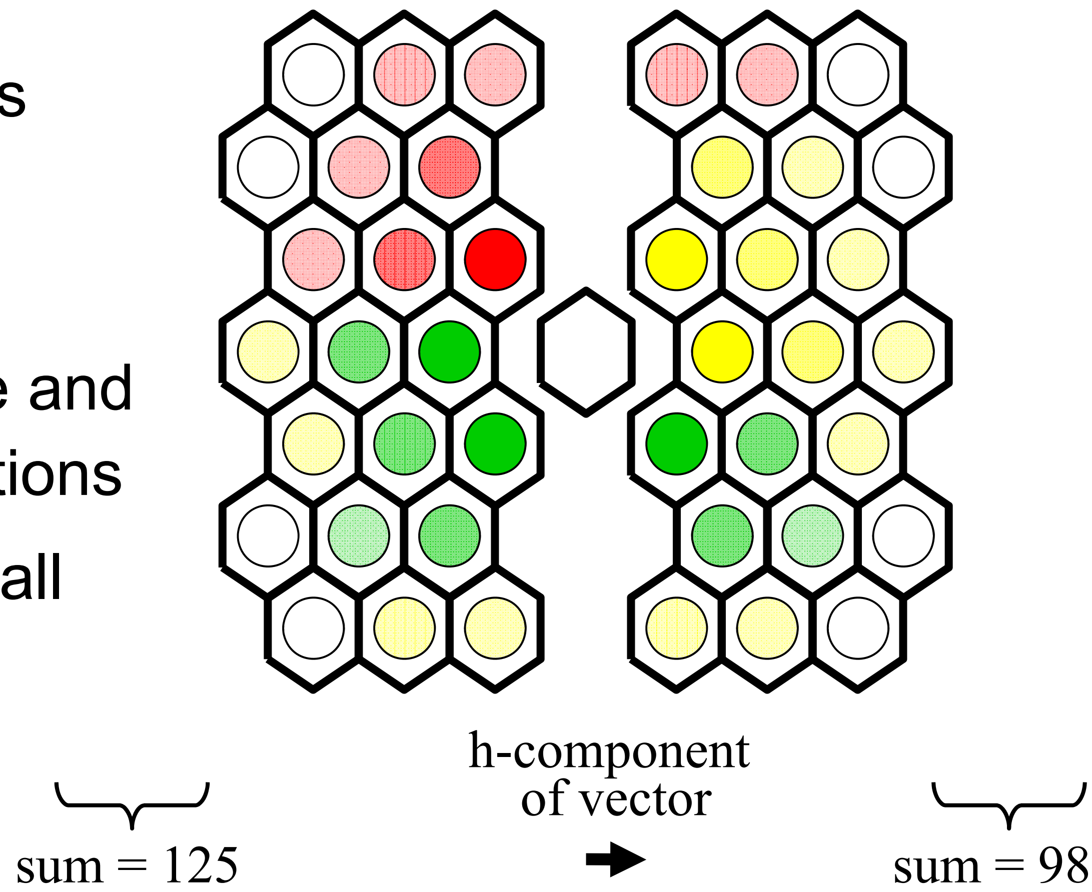
Flow Computation

1. calculate distances
2. weight with kernel function
3. divide into positive and negative h/v directions



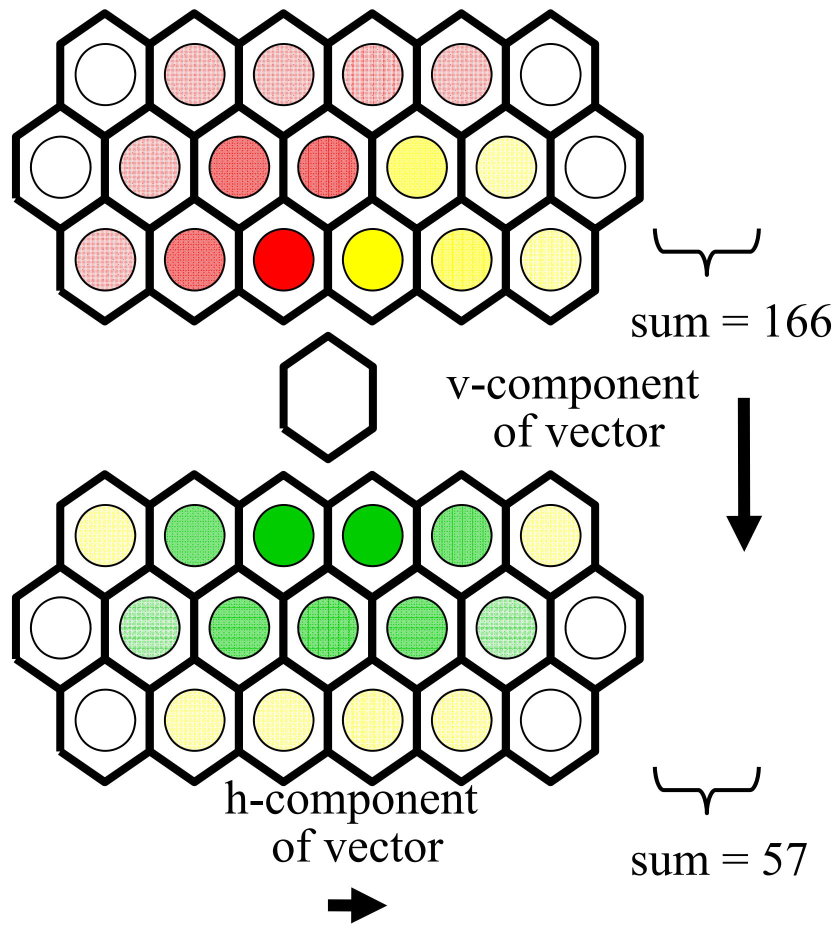
Flow Computation

1. calculate distances
2. weight with kernel function
3. divide into positive and negative h/v directions
4. calculate sums of all contributions



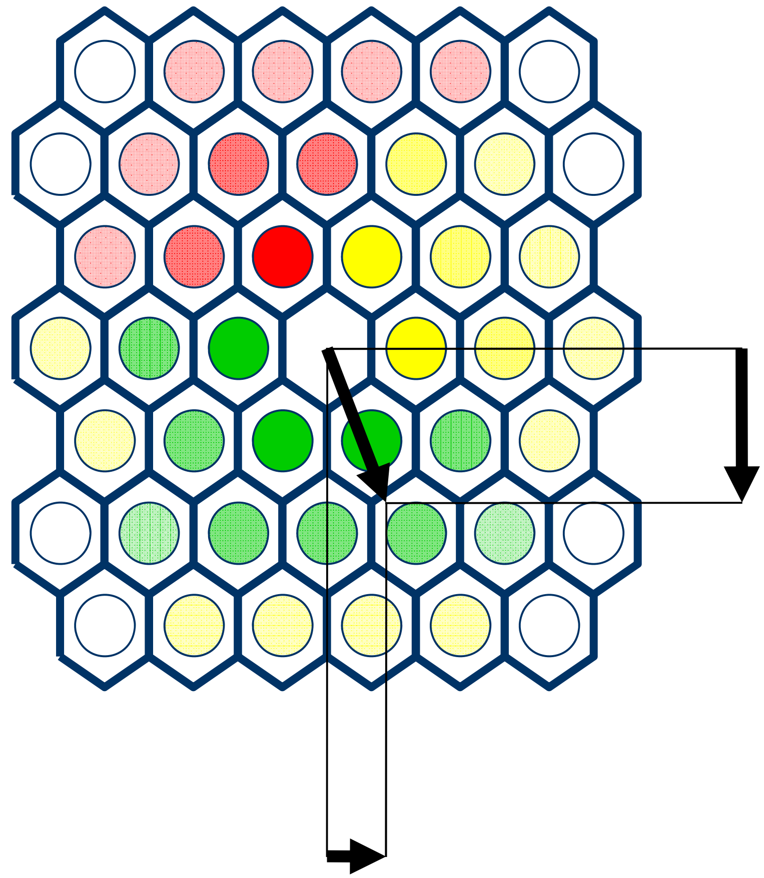
Flow Computation

1. calculate distances
2. weight with kernel function
3. divide into positive and negative h/v directions
4. calculate sums of all contributions



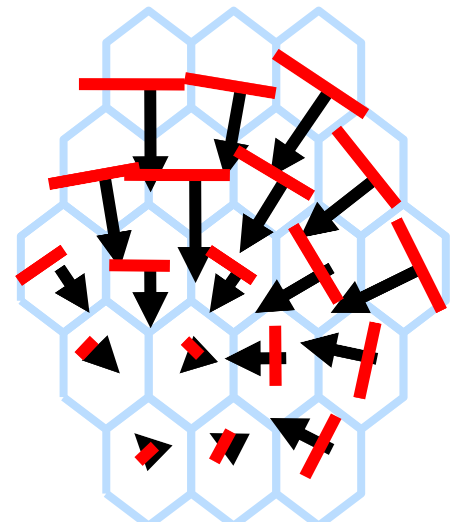
Flow Computation

1. calculate distances
2. weight with kernel function
3. divide into positive and negative h/v directions
4. calculate sums of all contributions
5. aggregate h/v components



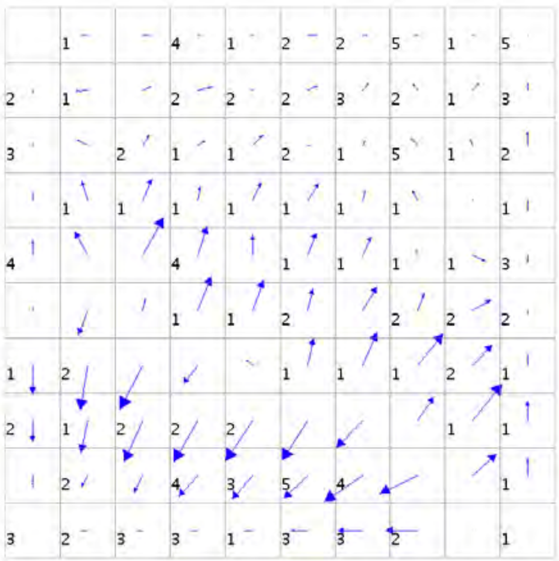
Flow vs. Borderlines

- Flow
  - arrows point to cluster centers
  - length shows intensity
- Borderlines
  - dual representation
  - rotate arrows by 90 degrees
  - show cluster boundaries

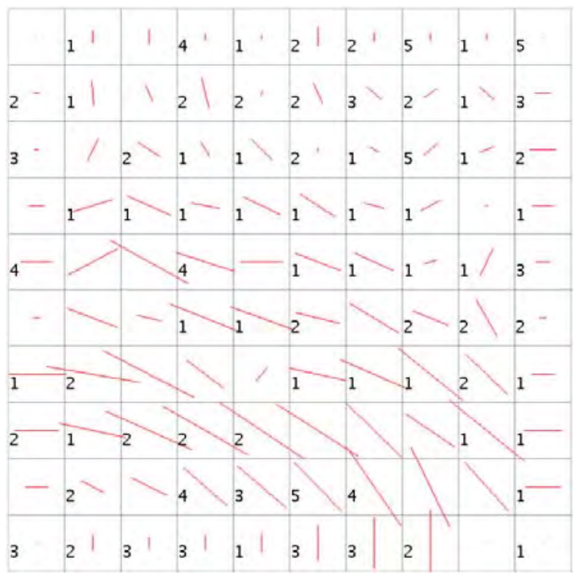


Flow/Borderlines

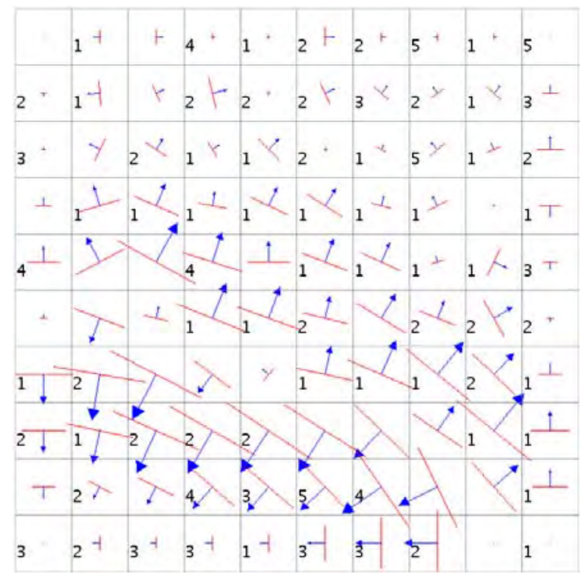
- Iris Dataset:



Flow

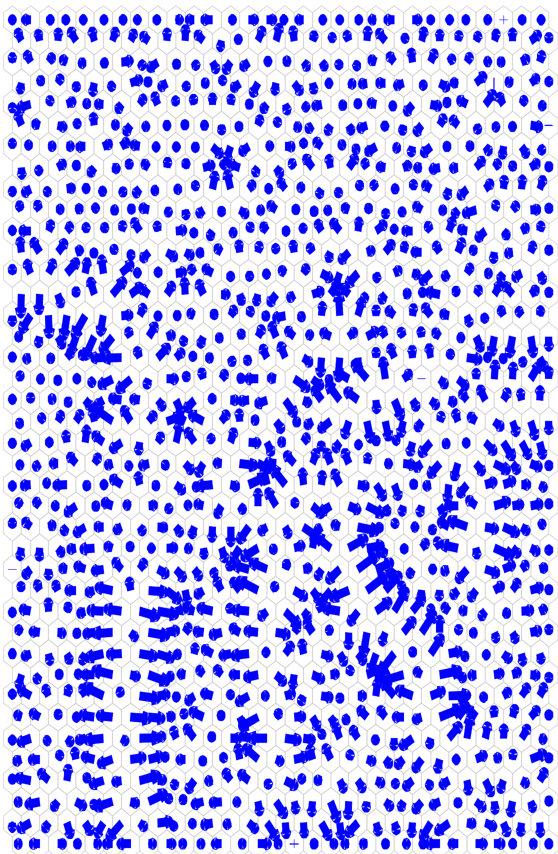


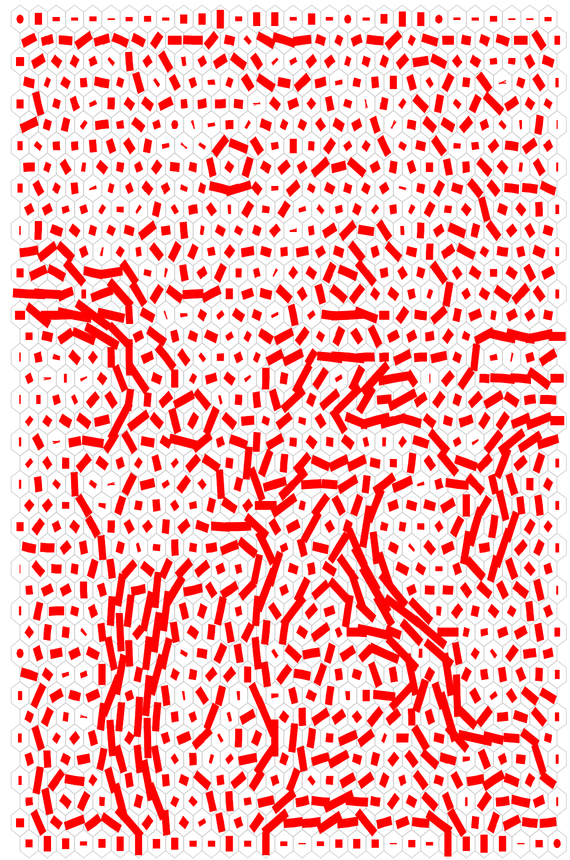
Borderlines

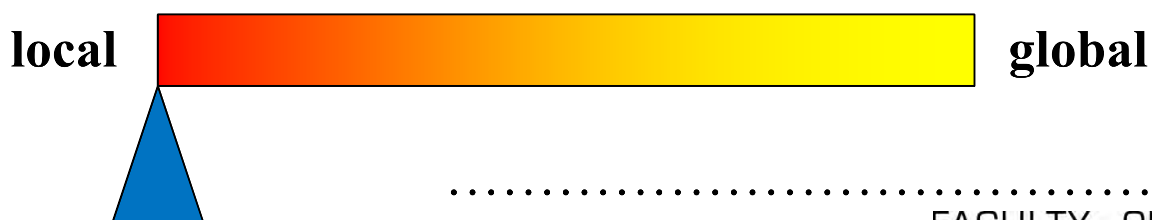


Flow+Borderlines

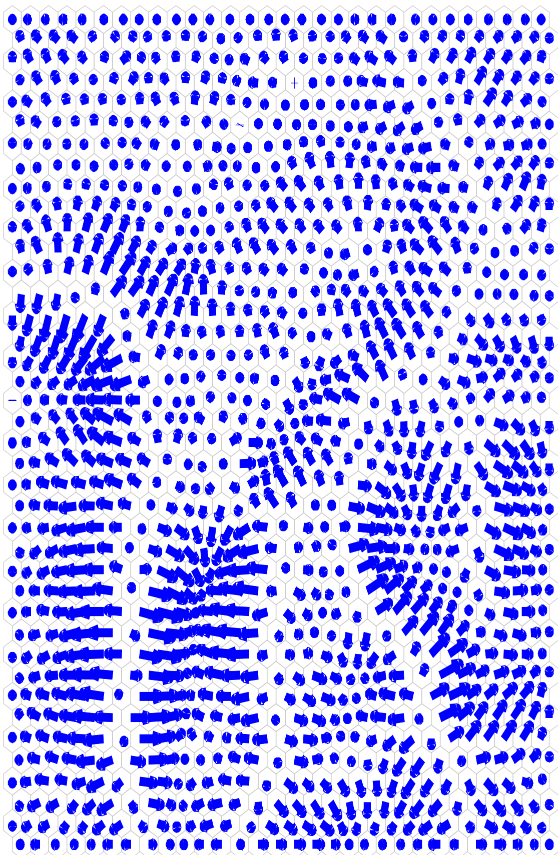
Flow/Borderline: sigma = 1

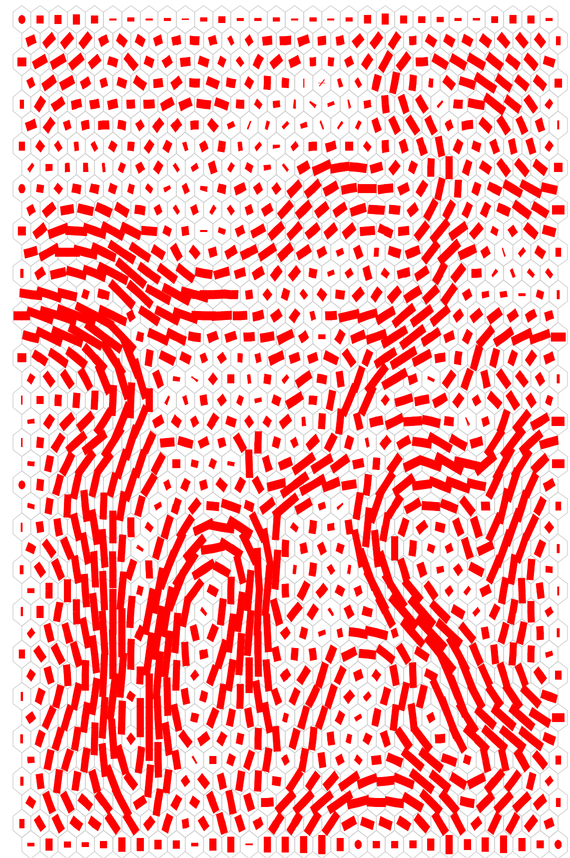


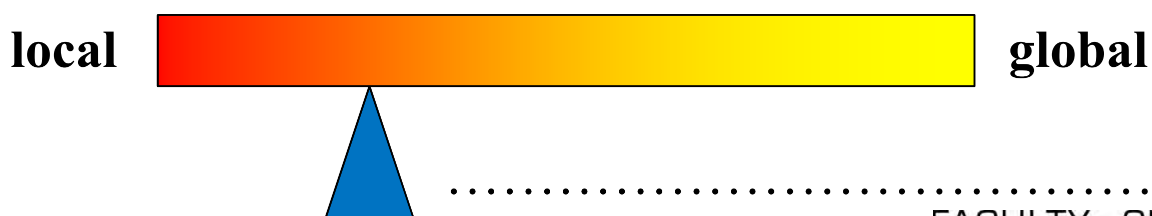




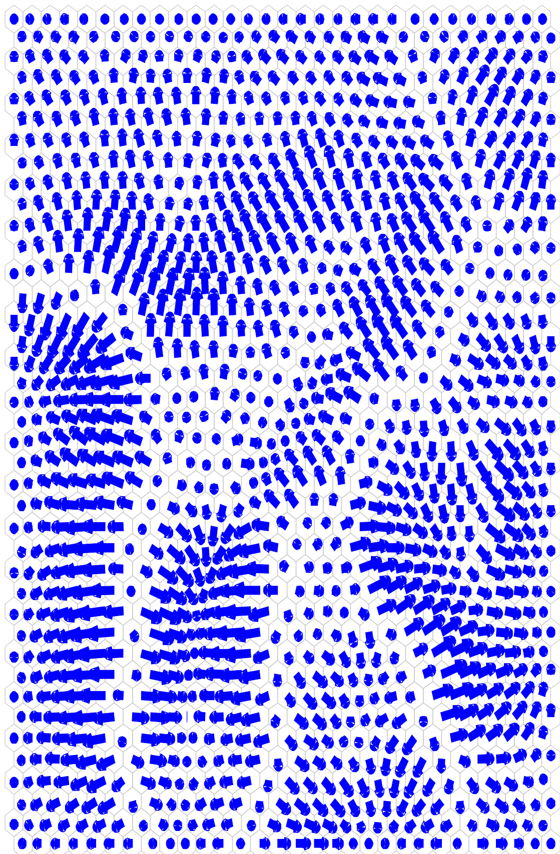
Flow/Borderline: sigma = 3

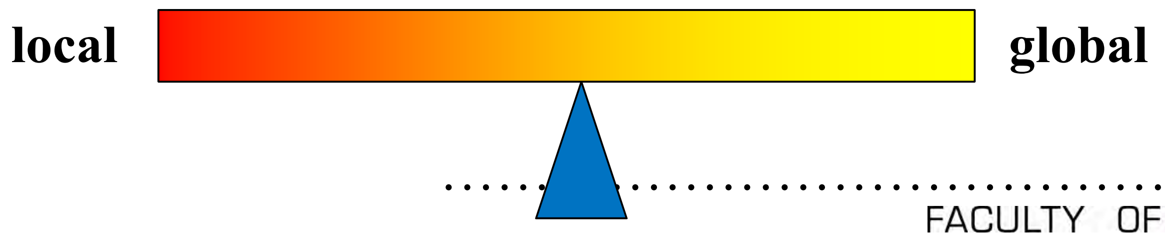




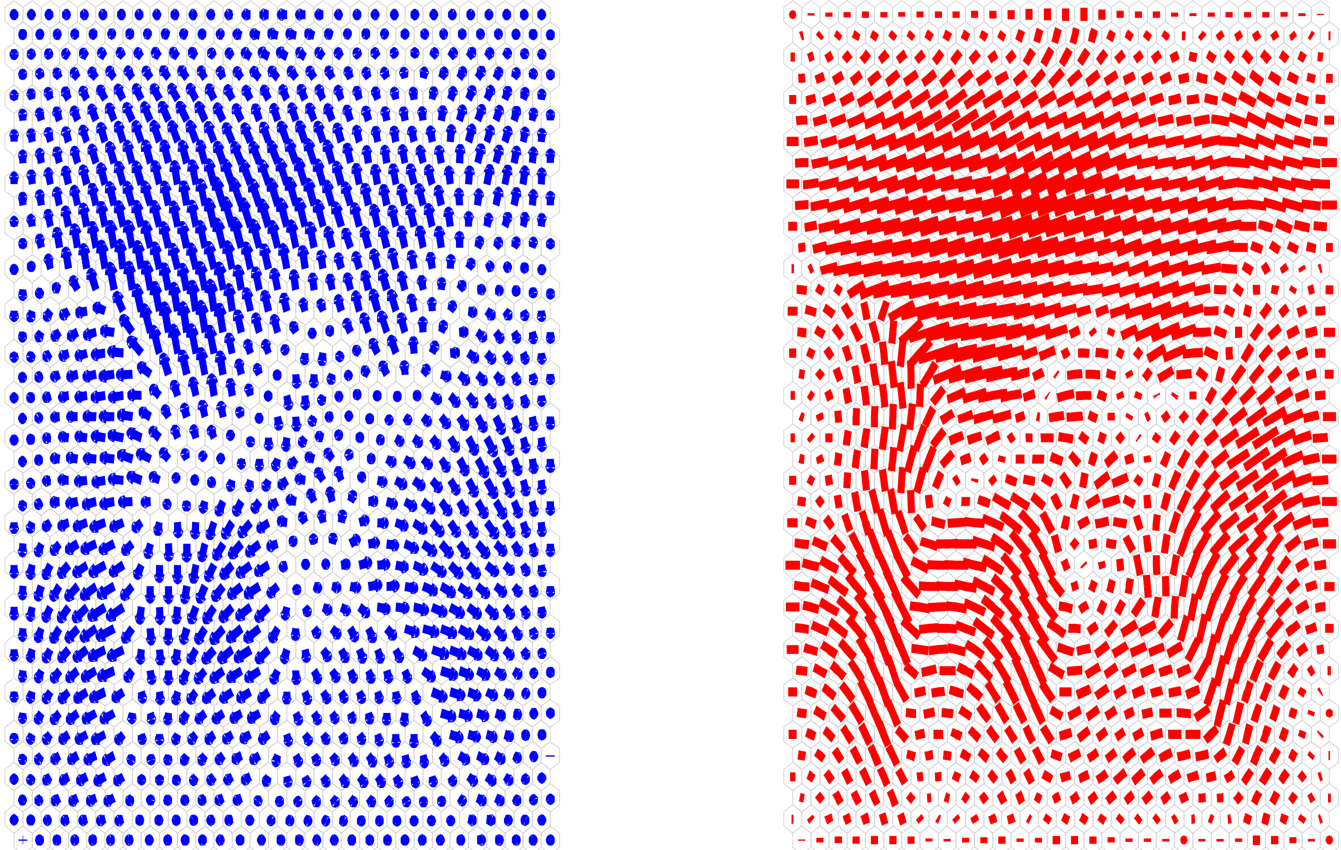


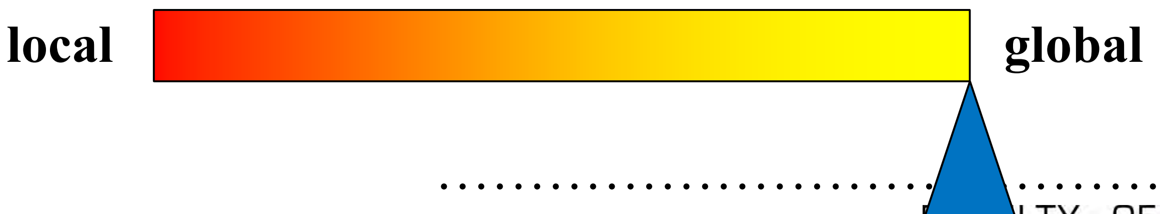
Flow/Borderline: sigma = 5



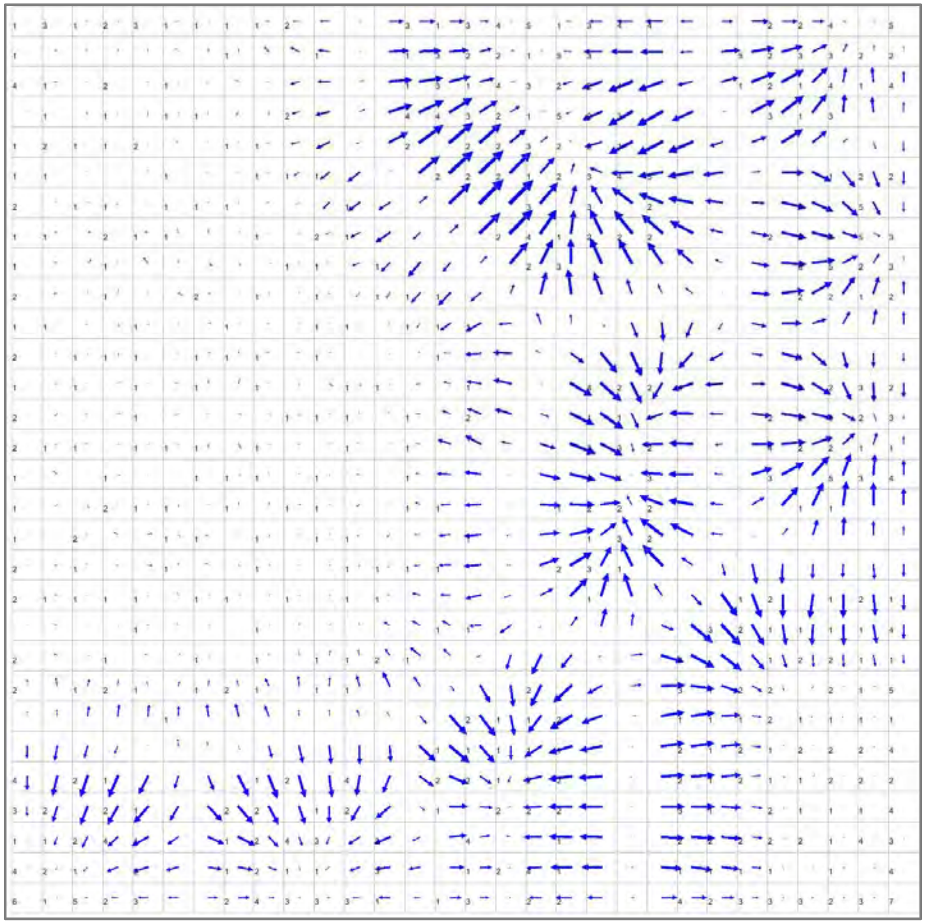


Flow/Borderline: sigma = 15

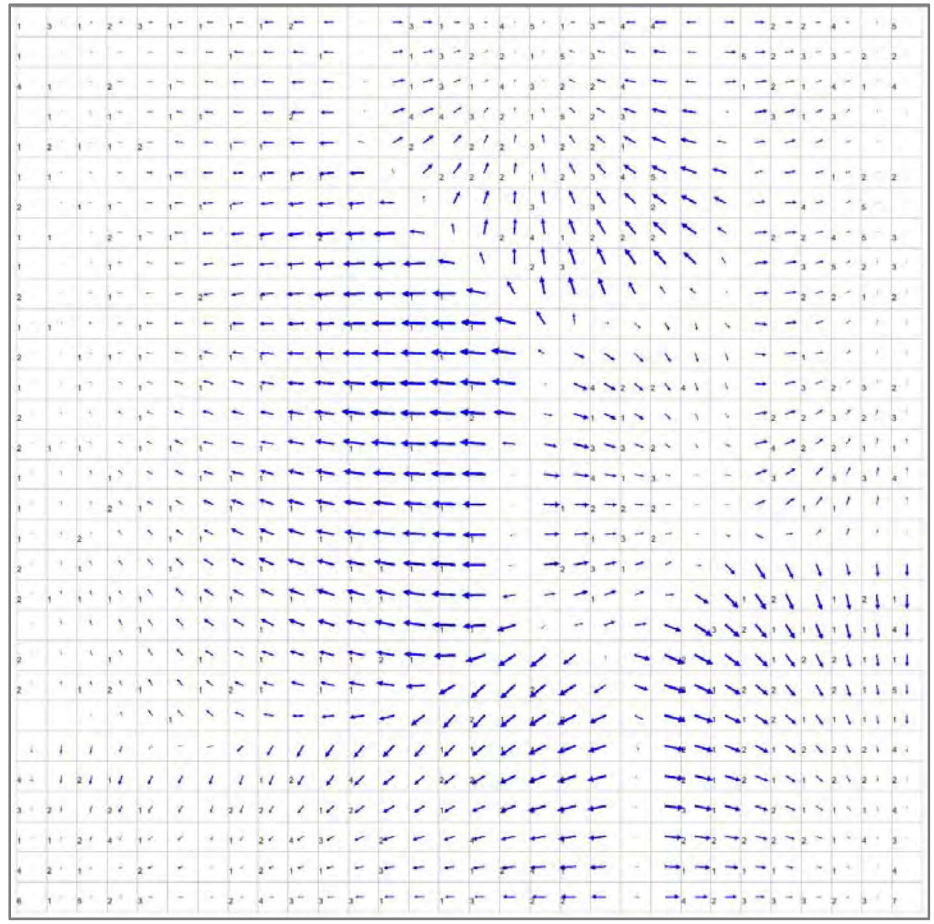




Flow: 10 Clusters Dataset

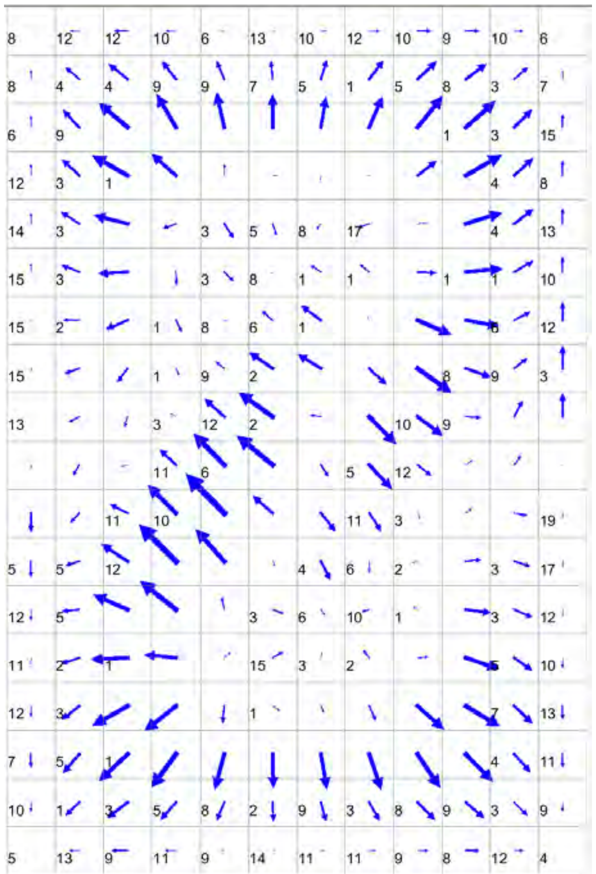


R = 1.7

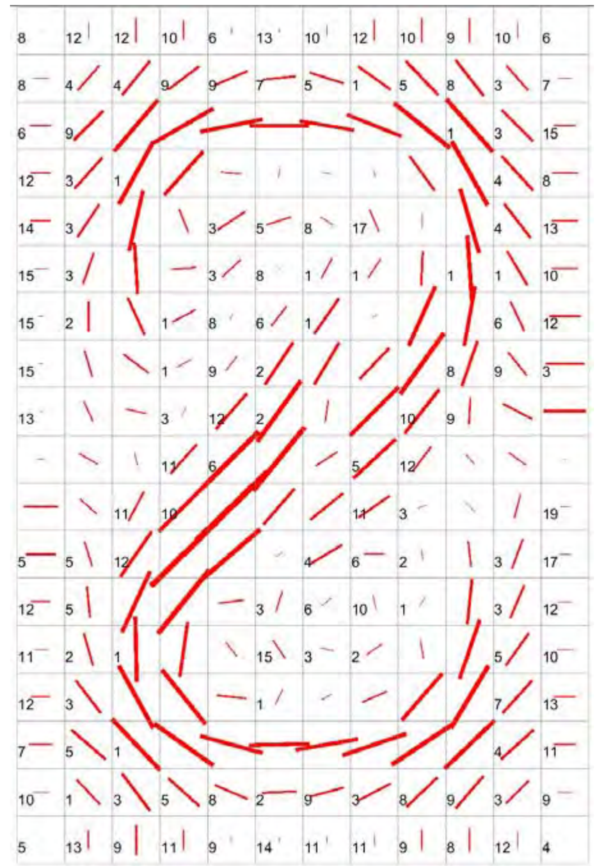


R = 8.0

Chainlink Dataset

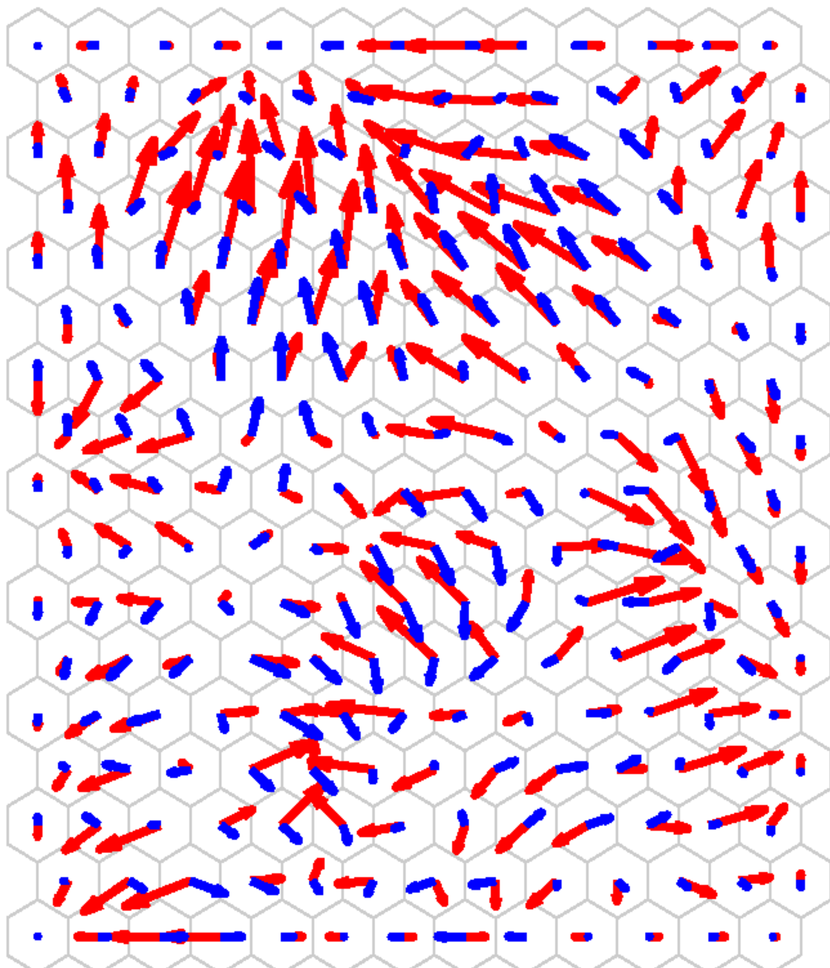


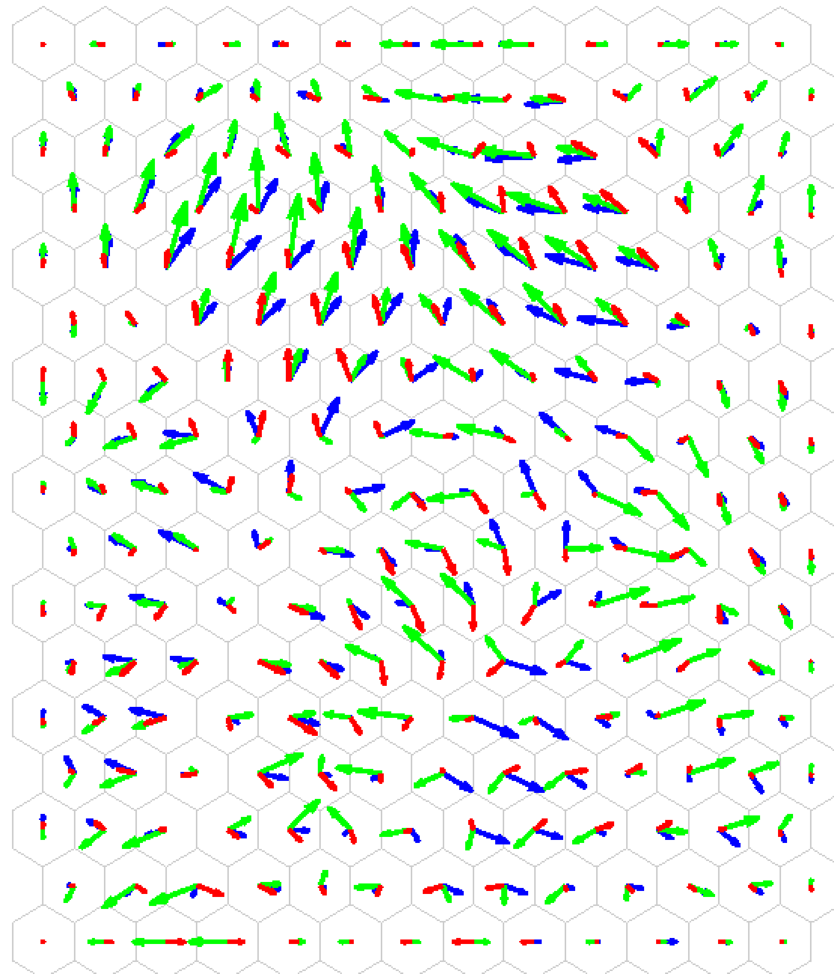
Flow
R = 4.3



Border
R = 4.3

Flow: Groups of Attributes





Outline

Visualizations on the SOM

- Textual information
- Density
- Distances
  - Activity Histograms
  - Minimum Spanning Trees
  - Cluster Connections (CC)
  - D-Matrix, U-Matrix
  - U* Matrix: U-Matrix + P-Matrix
  - Vectorfields: Flow / Borderline
- Class info
- Attributes
- Clustering of the SOM

Outline

- Overview of visualization types
- Visualizing the SOM
  - Codebook projection
  - Adaptive Coordinates
- Visualizations on the SOM
  - Textual information
  - Density
  - Distances
  - Class info
  - Attributes
  - Clustering of the SOM

Outline

Visualizations on the SOM

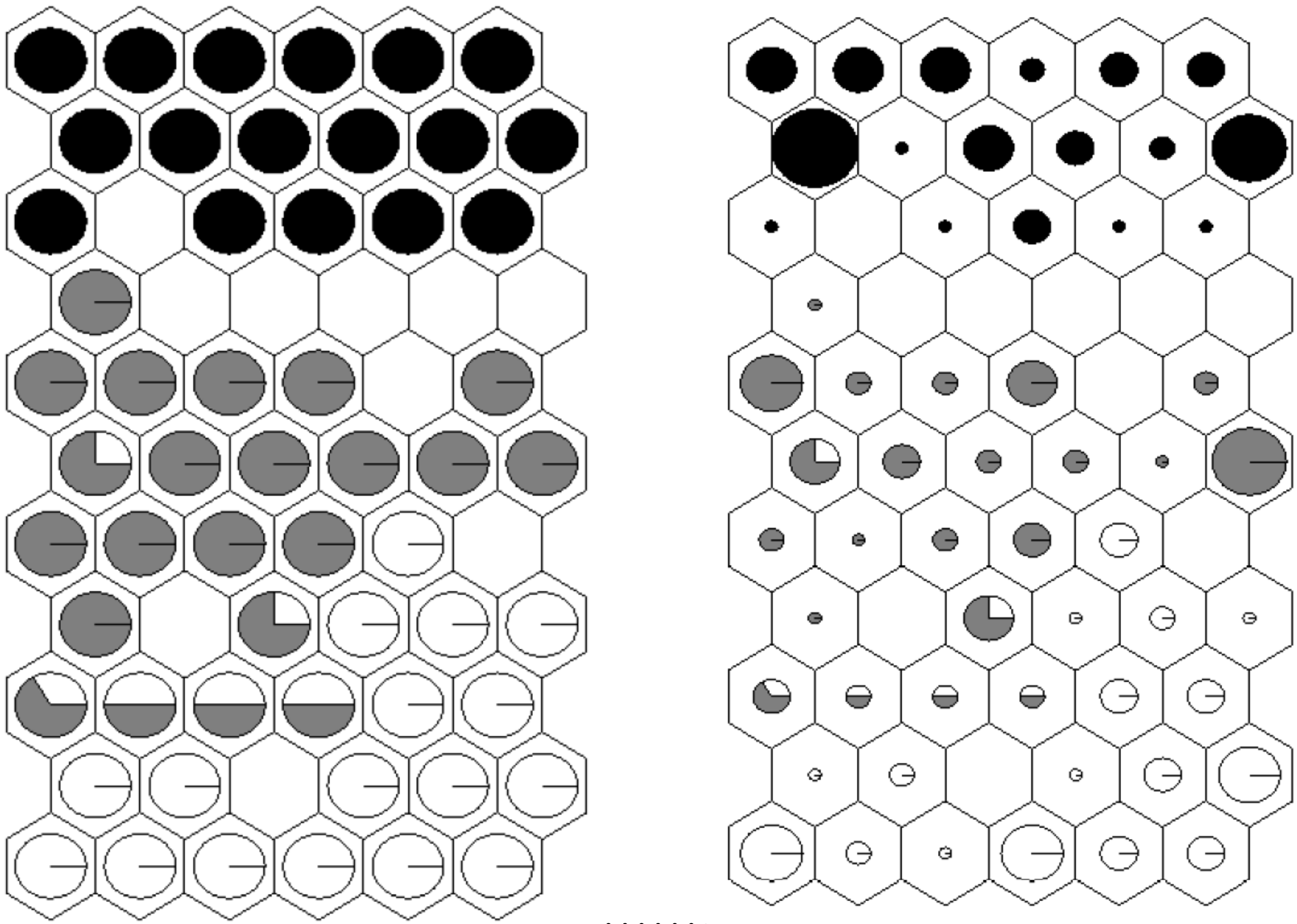
- Textual information
- Density
- Distances
- Class info
  - Pie Charts / Patches
  - Class Coloring:
    - Chessboard
    - Color Filling with Attractor
- Attributes
- Clustering of the SOM

Class Distributions

- Class information: distribution of classes onto units
- Shows homogeneity of class distribution, sub-classes, split classes, class overlap, …
- Standard approaches:
  - Discrete visualization on units
  - Pie Charts: with / without size indicating density
  - Overlapping patches
- Class-coloring, Chessboard-Visualization
  - Coloring the entire map
  - Better visual representation of distribution and mixtures

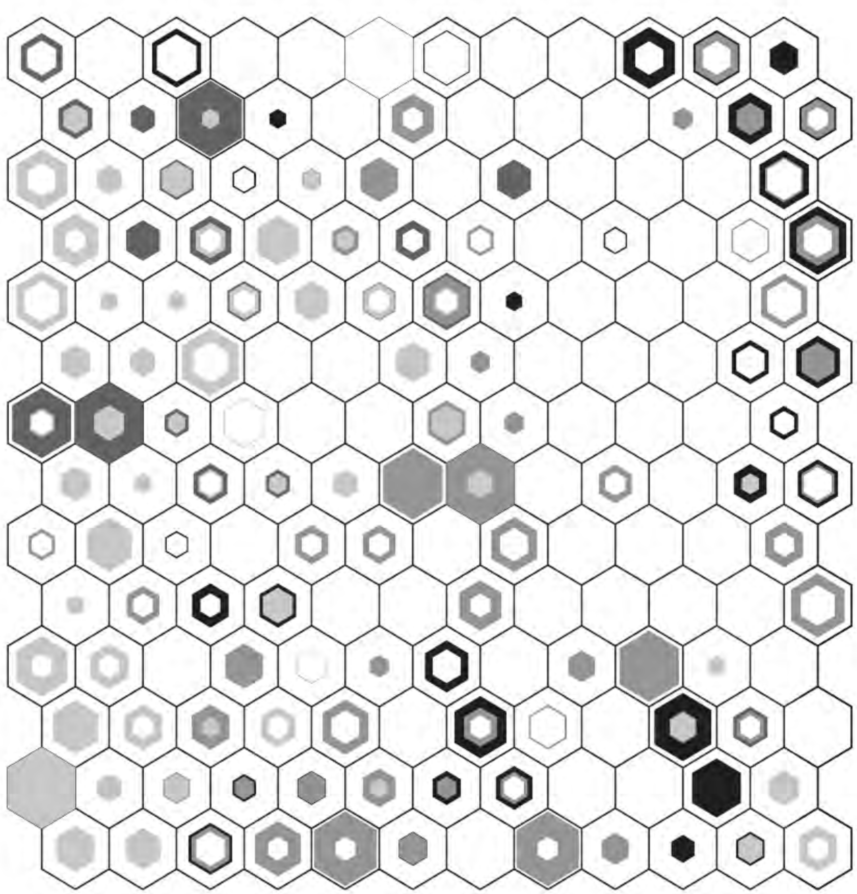
Class Distributions

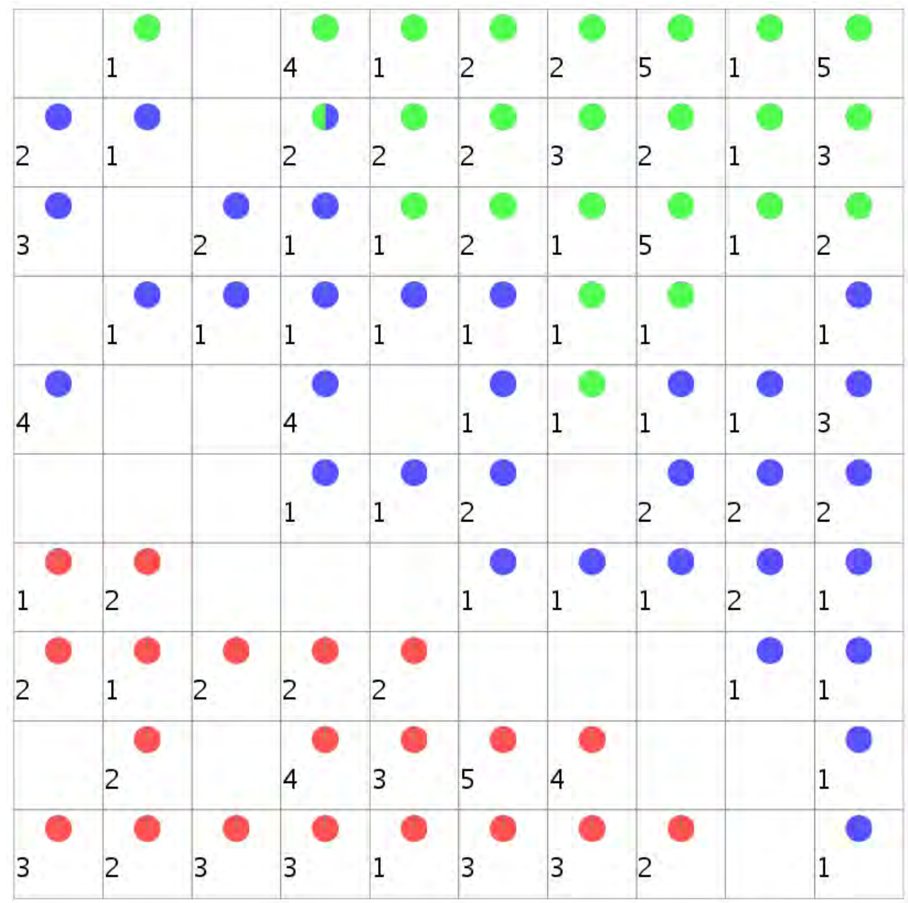
- Pie Charts: without / with hit histogramm



Class Distributions

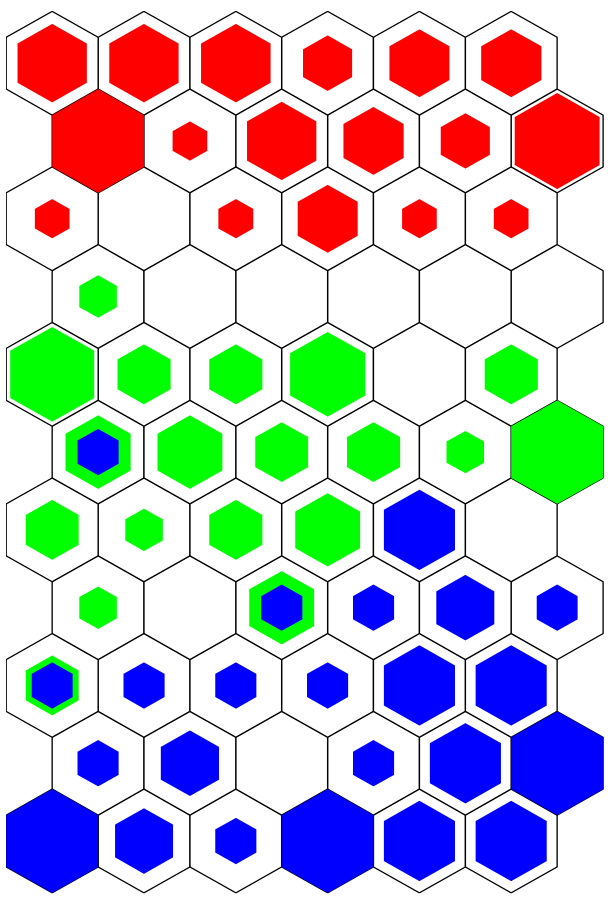
- Patches with Hit Histogramm, Pie Charts



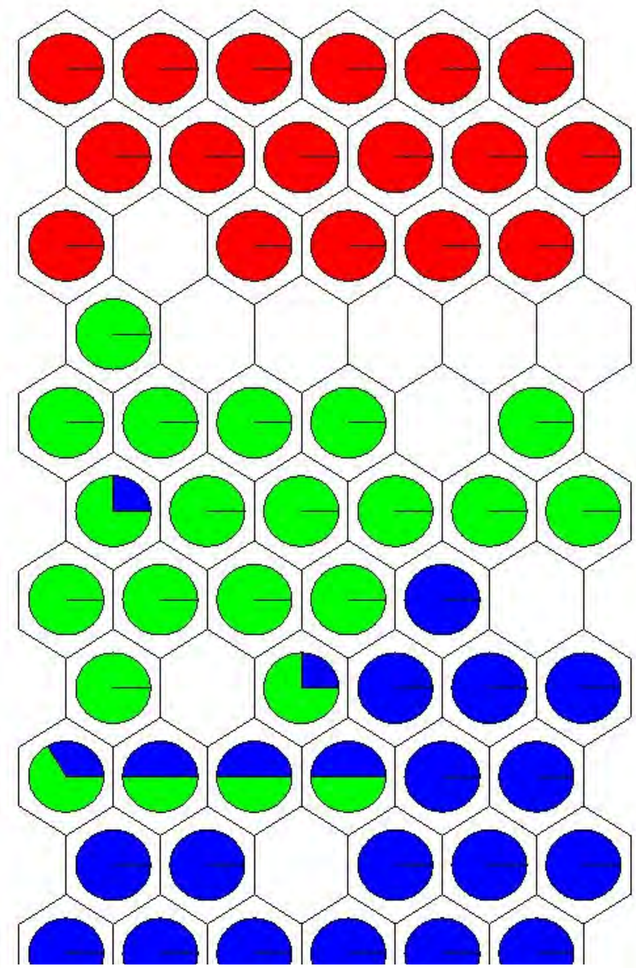


Class Distributions

- Patches with Hit Histogramm, Pie-Charts



Iris (small),
3 classes



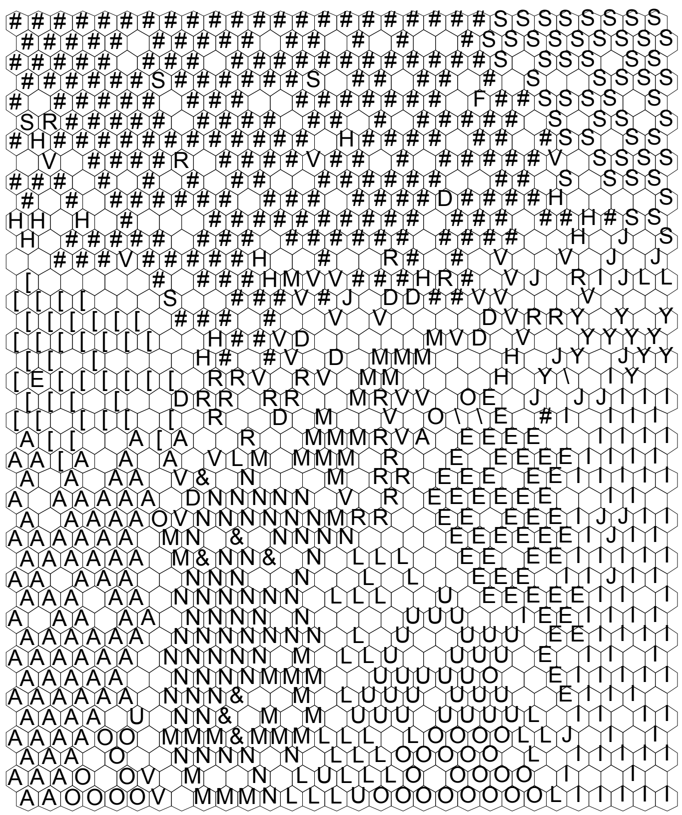
Iris (small),
3 classes, Pie Charts

GP9

Class Distributions

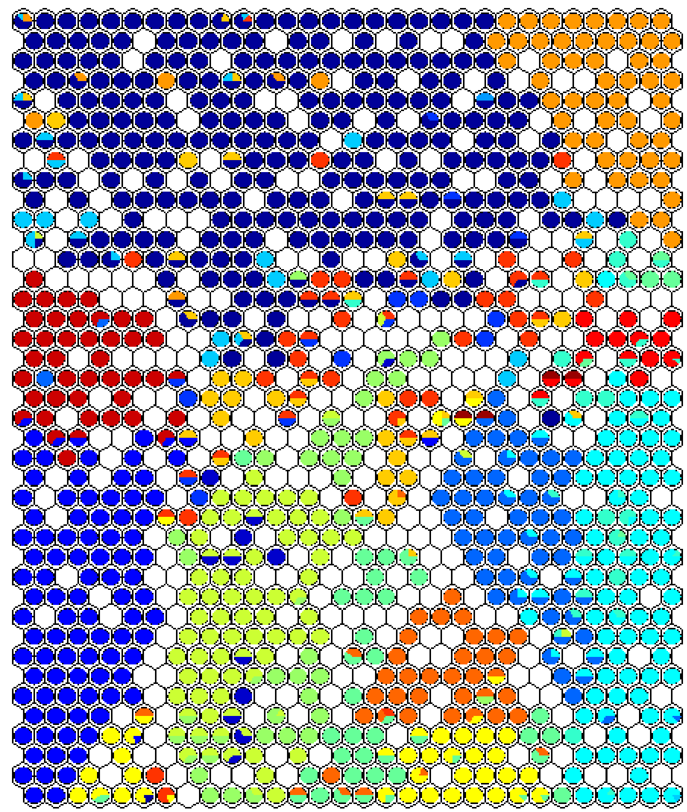
- Textual Class-IDs, Pie-Charts

Labels (only most frequent label)



class IDs

Pie Chart of Categoriy (unscaled)



pie charts

GP9 Label: Majority voting von Hit Histogramm, nach Klassenlabel
Georg Pölzlbauer, 7/26/2005

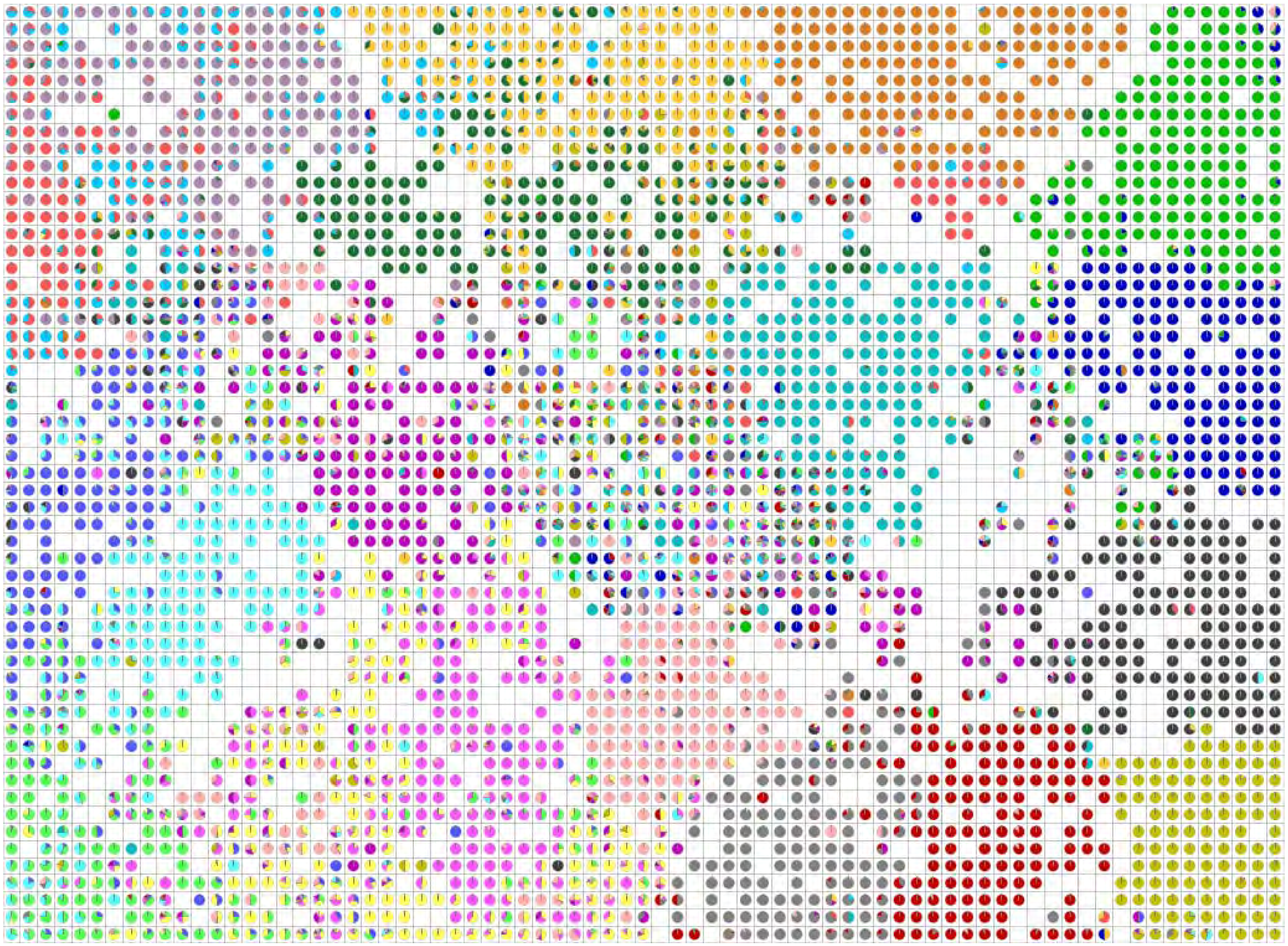
Class Distributions

- 20-Newsgroups Benchmark dataset
- 1000 postings per newsgroup
- Hierarchy of newsgroups

- Full-term indexing
- Stemming
- Note: class coloring might be used to reflect hierarchy!

alt.atheism	
comp.graphics	
comp.os.ms-windows.misc	
comp.sys.ibm.pc.hardware	
comp.sys.mac.hardware	
comp.windows.x	
misc.forsale	
rec.autos	
rec.motorcycles	
rec.sport.baseball	
rec.sport.hockey	
sci.crypt	
sci.electronics	
sci.med	
sci.space	
soc.religion.christian	
talk.politics.guns	
talk.politics.mideast	
talk.politics.misc	
talk.religion.misc	

Class Distributions: 20 Newsgroups



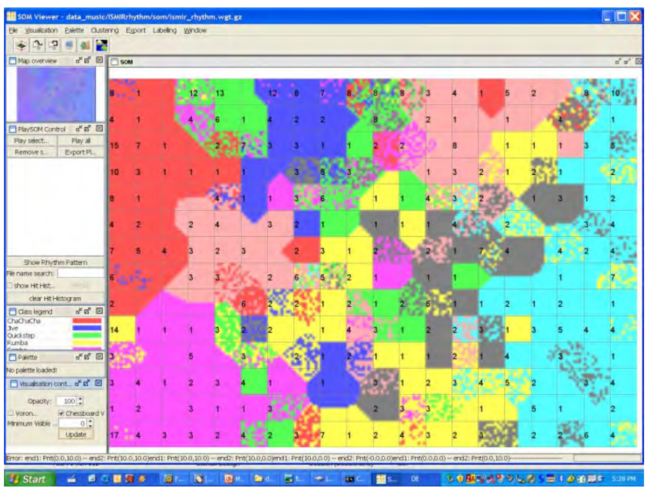
Outline

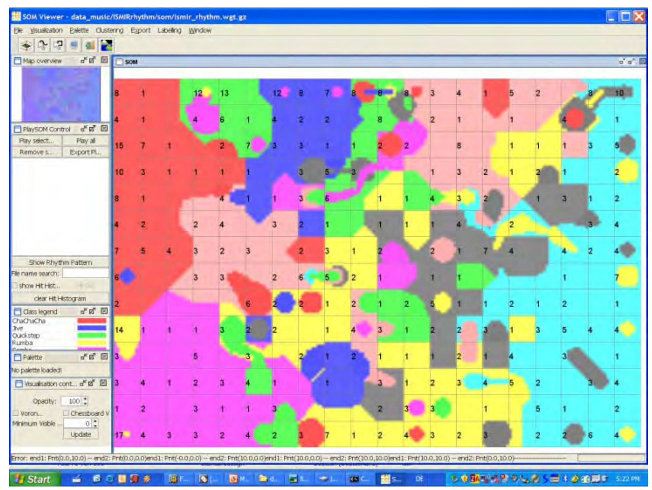
Visualizations on the SOM

- Textual information
- Density
- Distances
- Class info
  - Pie Charts / Patches
  - Class Coloring:
    - Chessboard
    - Color Filling with Attractor
- Attributes
- Clustering of the SOM

Class Distribution: Class Coloring

- Color SOM with class information
- Similar to pie chart representation
- 2 visualization types:
  - chessboard visualization
  - color flooding with attractor





Class Distribution : Class Coloring

- Taha Abdel-Aziz: **Coloring of the SOM based on Class Labels**. Master Thesis, Department of Software Technology and Interactive Systems, Vienna University of Technology, October 2006.
- Rudolf Mayer, Taha Abdel Aziz, and Andreas Rauber. **Visualising Class Distribution on Self-Organising Maps (accepted for publication)**.
  In Proceedings of the International Conference on Artificial Neural Networks (ICANN'07), Porto, Portugal, September 9 - 13 2007. Springer Verlag.

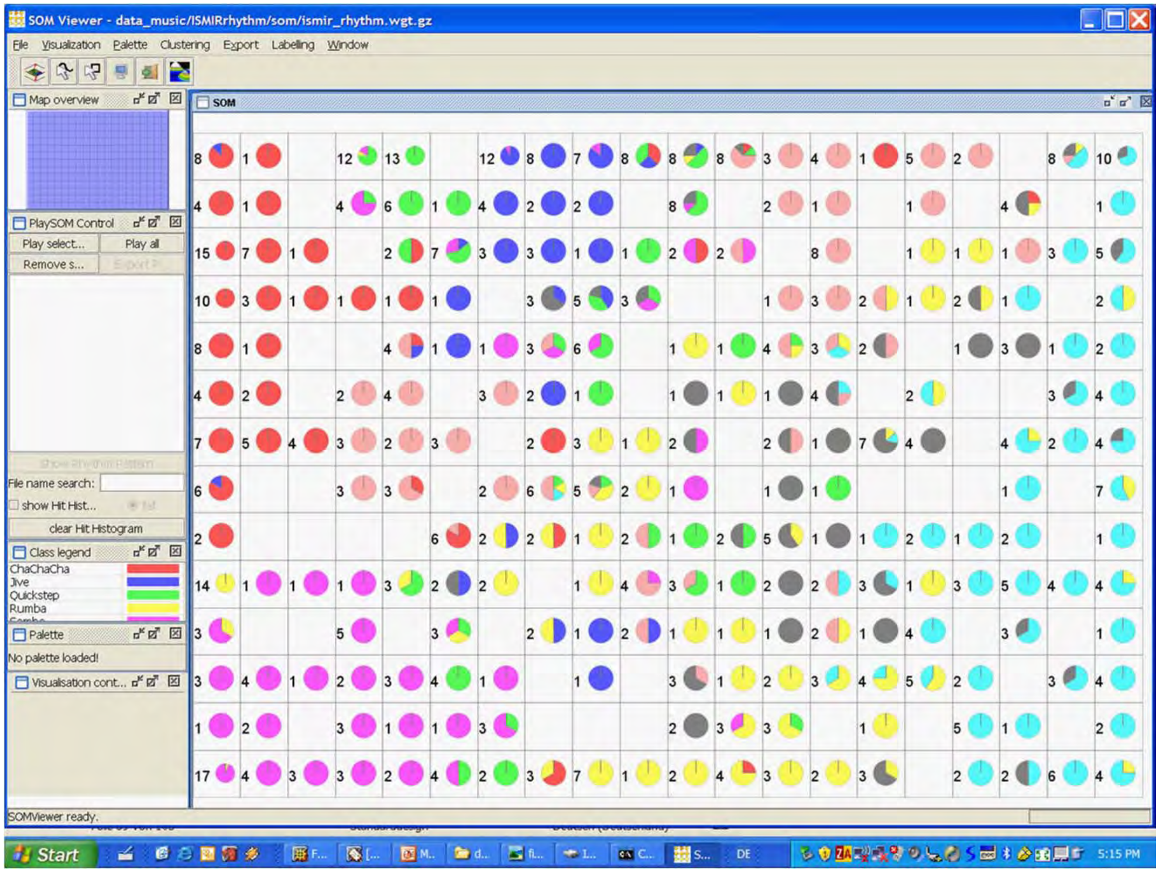
Class Coloring: Attractor Flooding

- Chessboard Visualization:
  - Voronoi Tesselation
  - color voronoi cells with patches accoridng to percentual share of class
  - opt.: set frequency threshold for small classes

Class Coloring: Chessboard

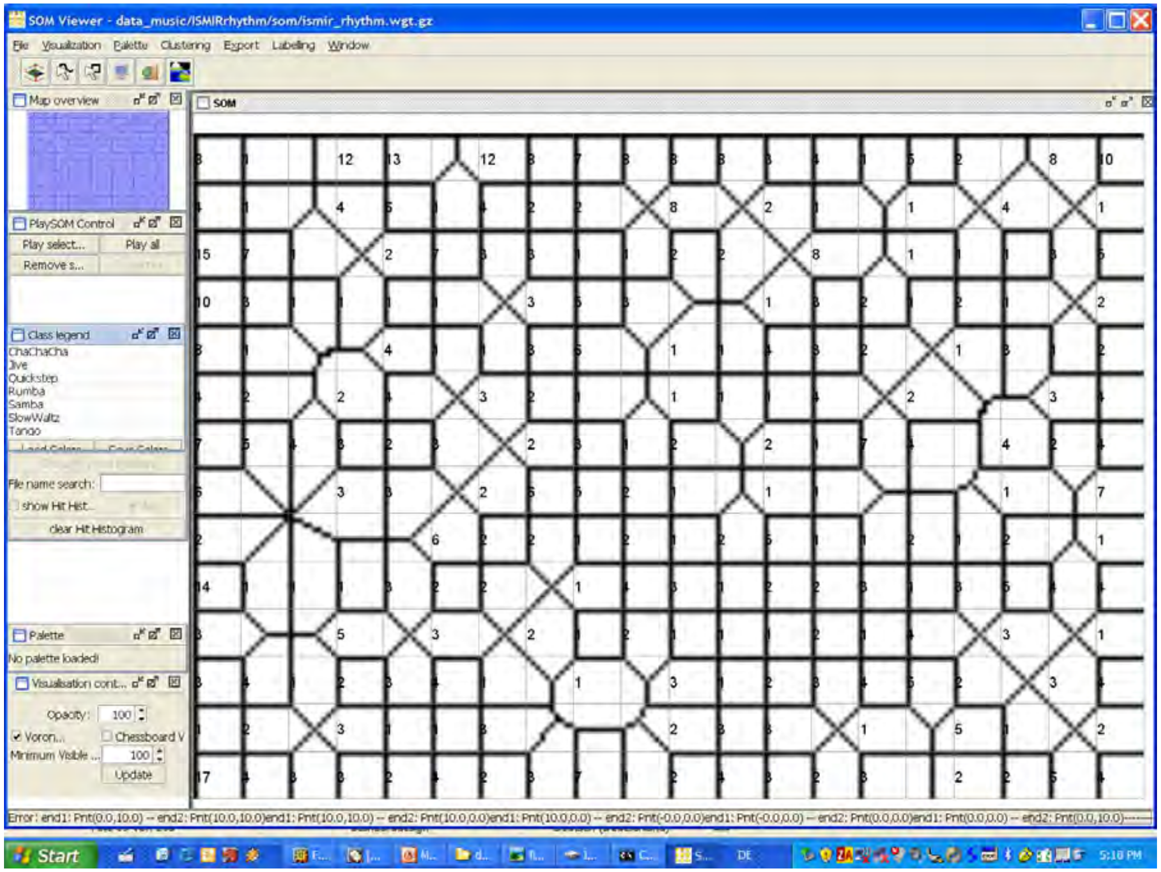
Chessboard:

- Step 1: Class information pie charts



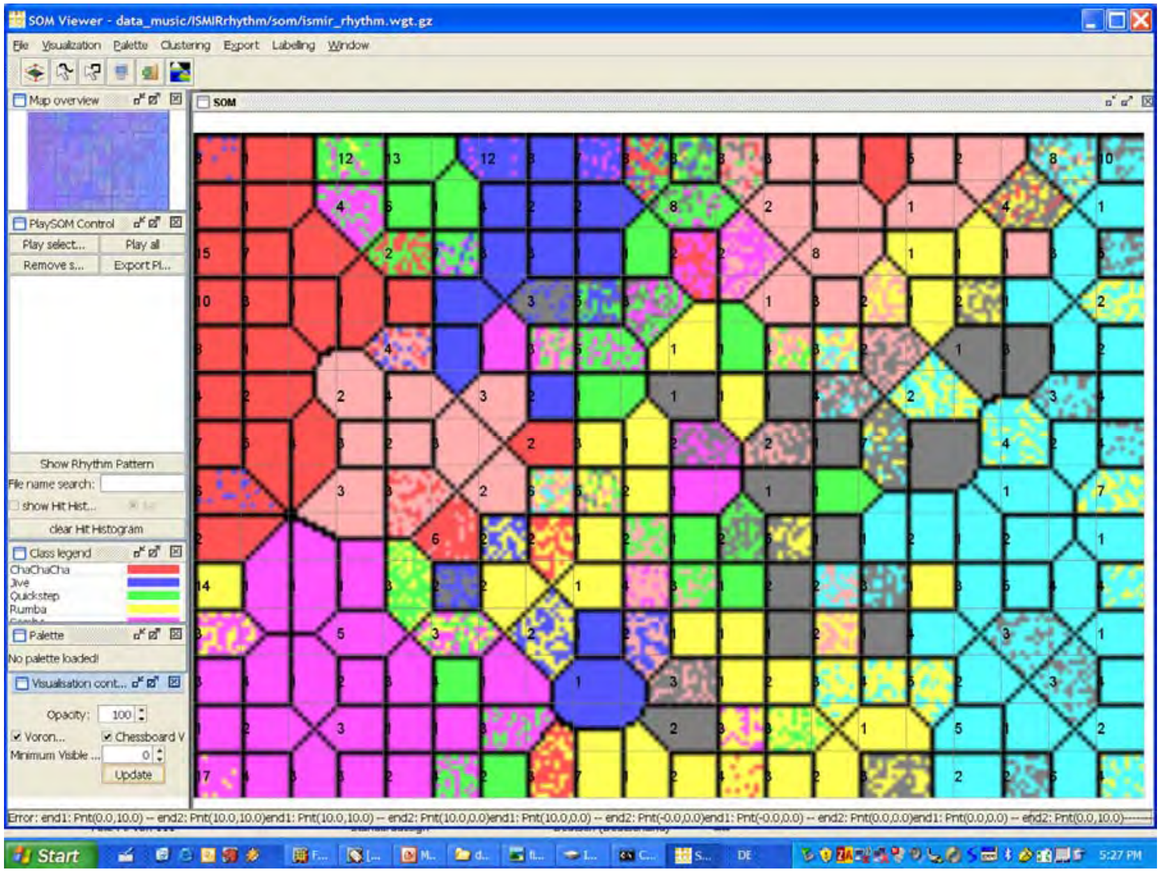
Class Coloring: Chessboard

- Step 2: Voronoi Tesselation



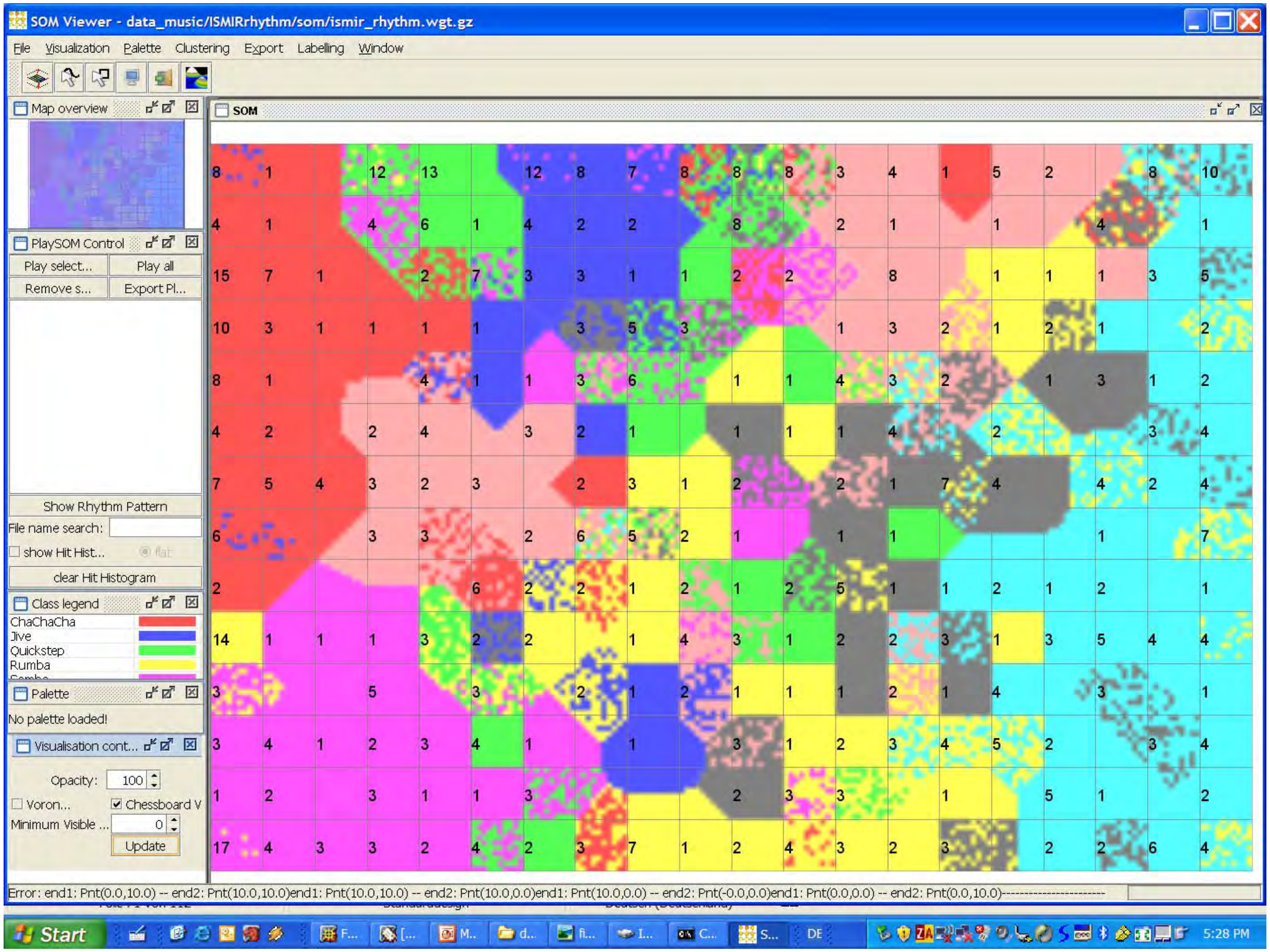
Class Coloring: Chessboard

- Step 3: Chessboard-style pixel coloring according to class frequency



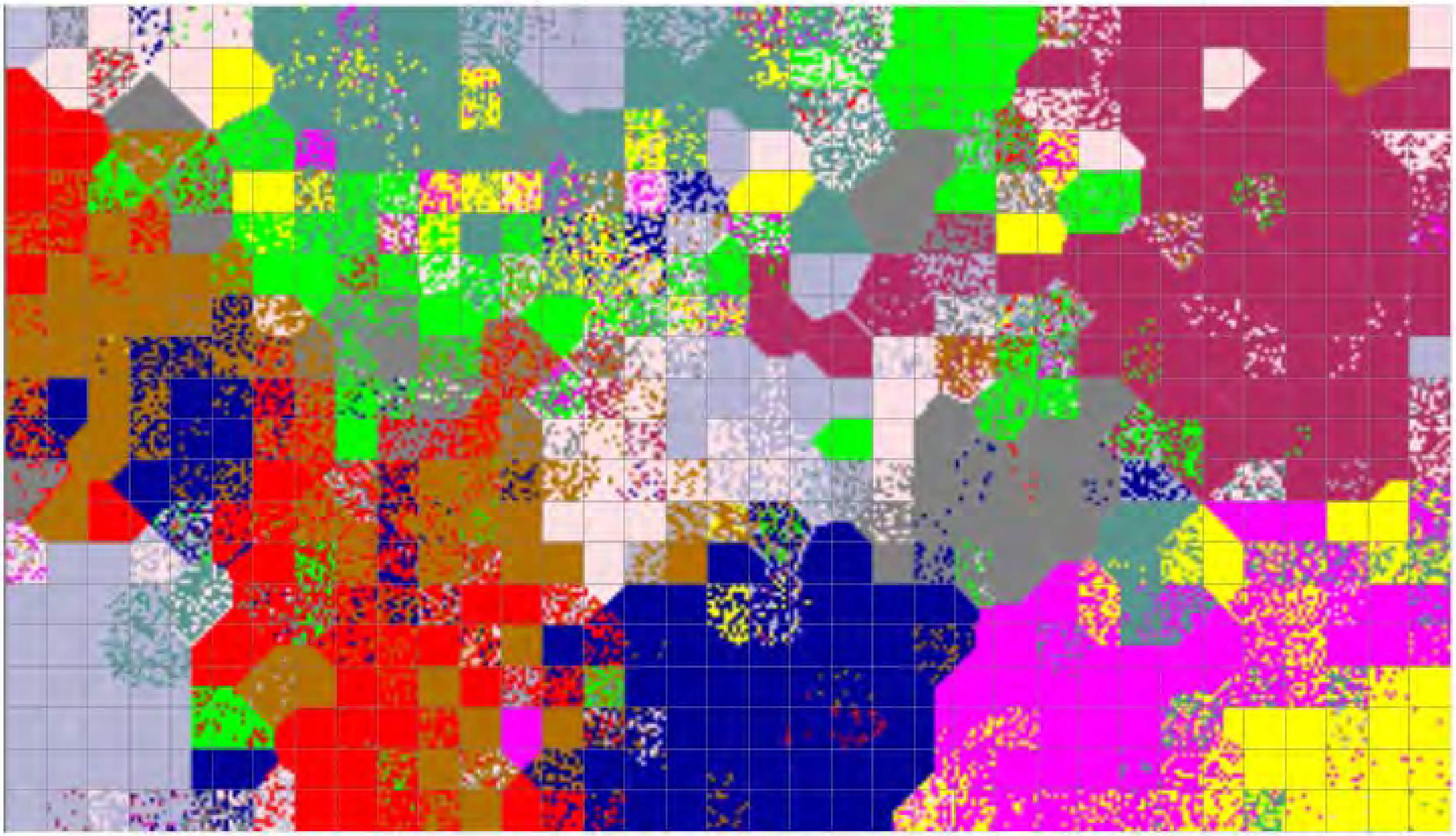
Class Coloring: Chessboard

- final:



Class Coloring: Chessboard

Chessboard visualization of Banksearch Data SOM



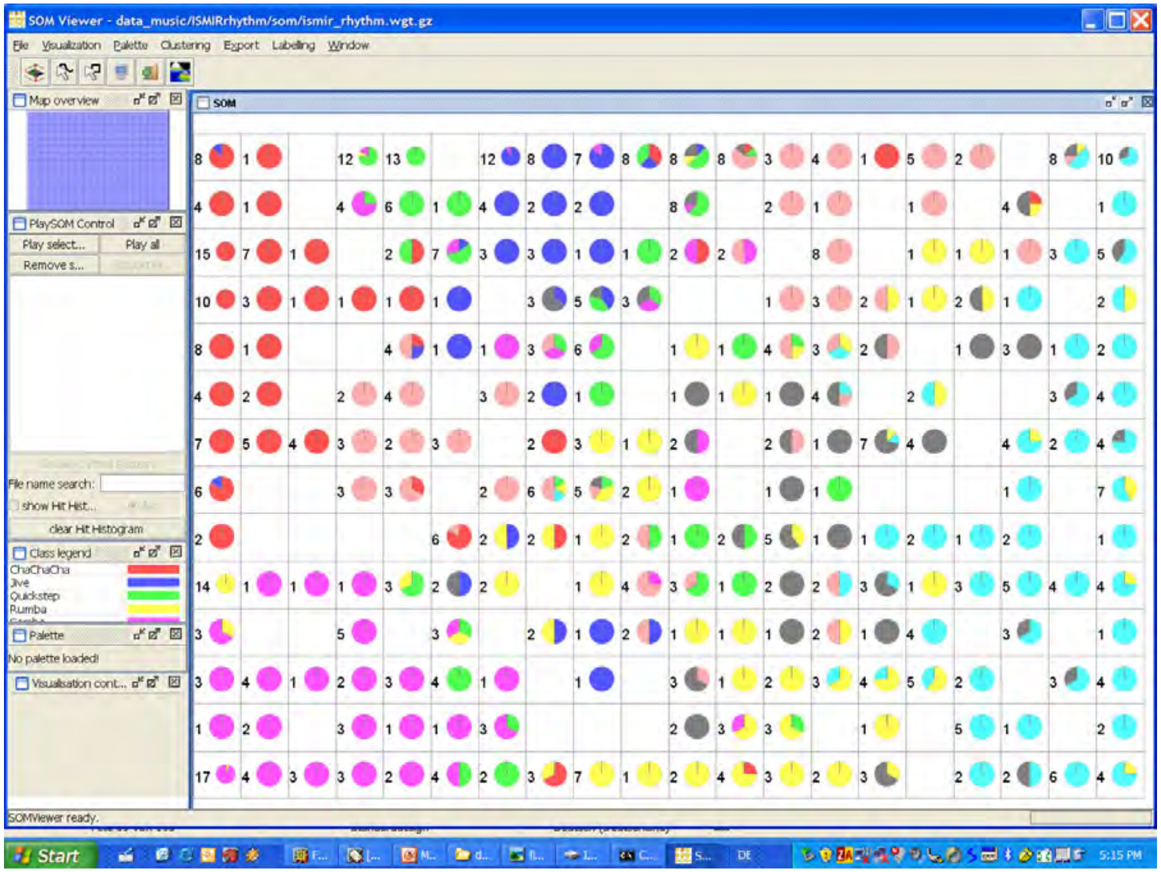
Class Coloring: Attractor Flooding

- Attractor Flooding:
  - Voronoi Tesselation
  - Fill with dominant class color
  - Identify neighboring class distributions
  - Identify attractors
  - Flood-fill style coloring along attractors according to frequency
  - opt.: set frequency threshold for small classes

Class Coloring: Attractor Flooding

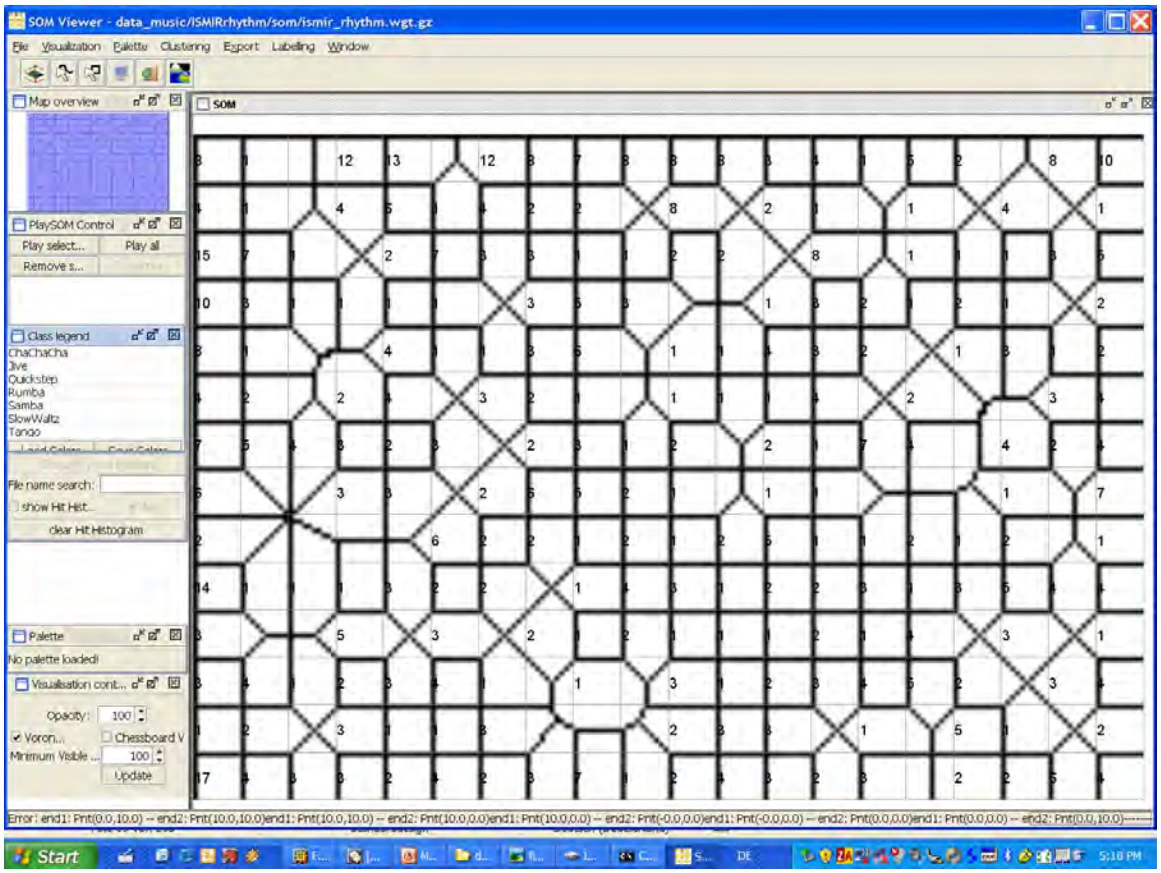
Attractor Flooding:

- Step 1: Class information pie charts



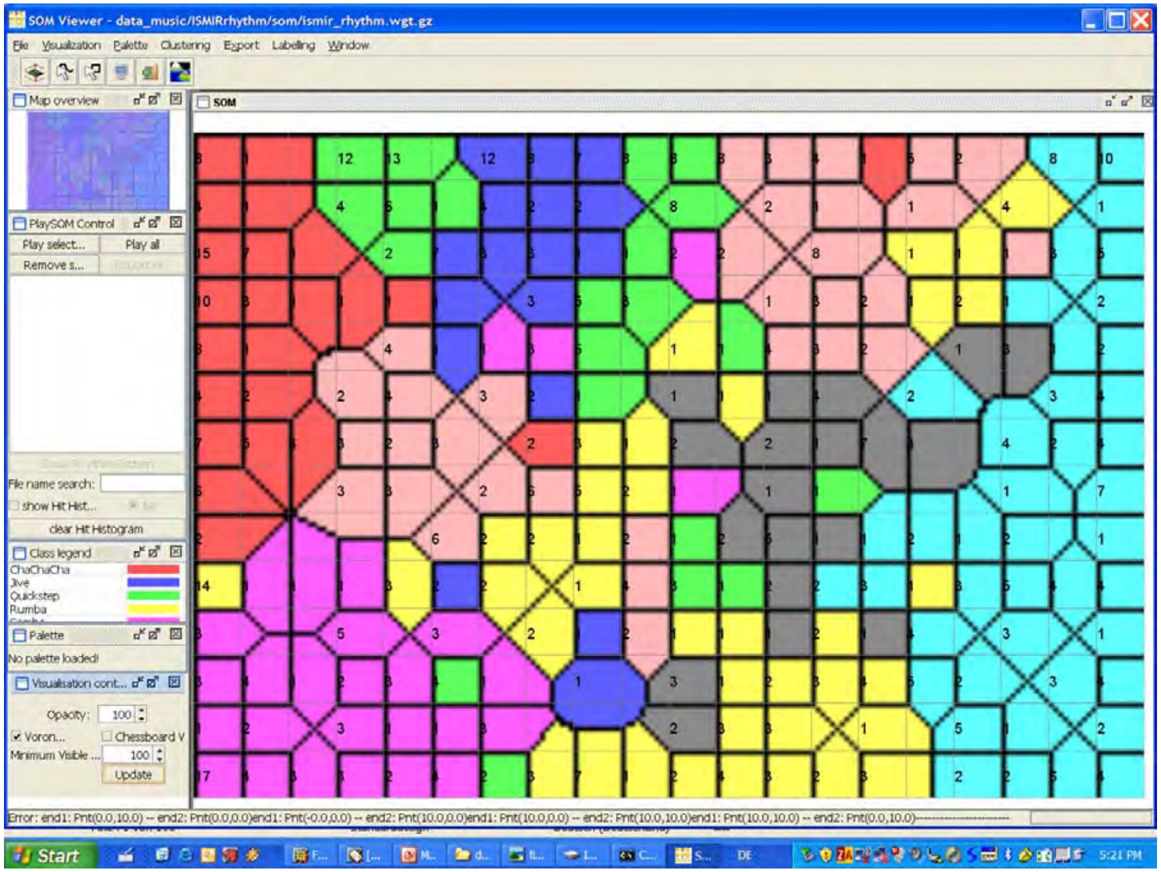
Class Coloring: Attractor Flooding

- Step 2: Voronoi Tesselation



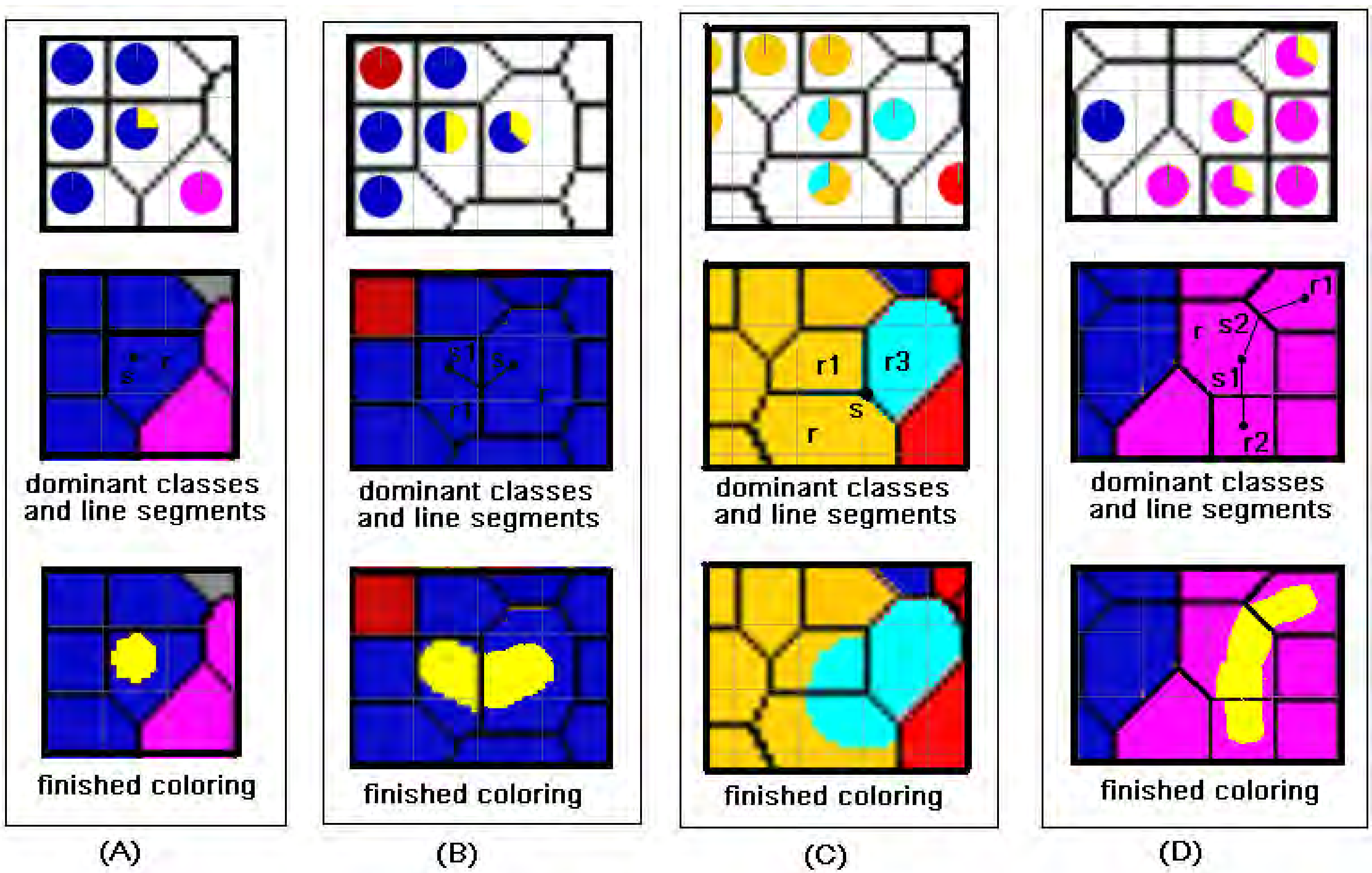
Class Coloring: Attractor Flooding

- Step 3: fill with dominant class color



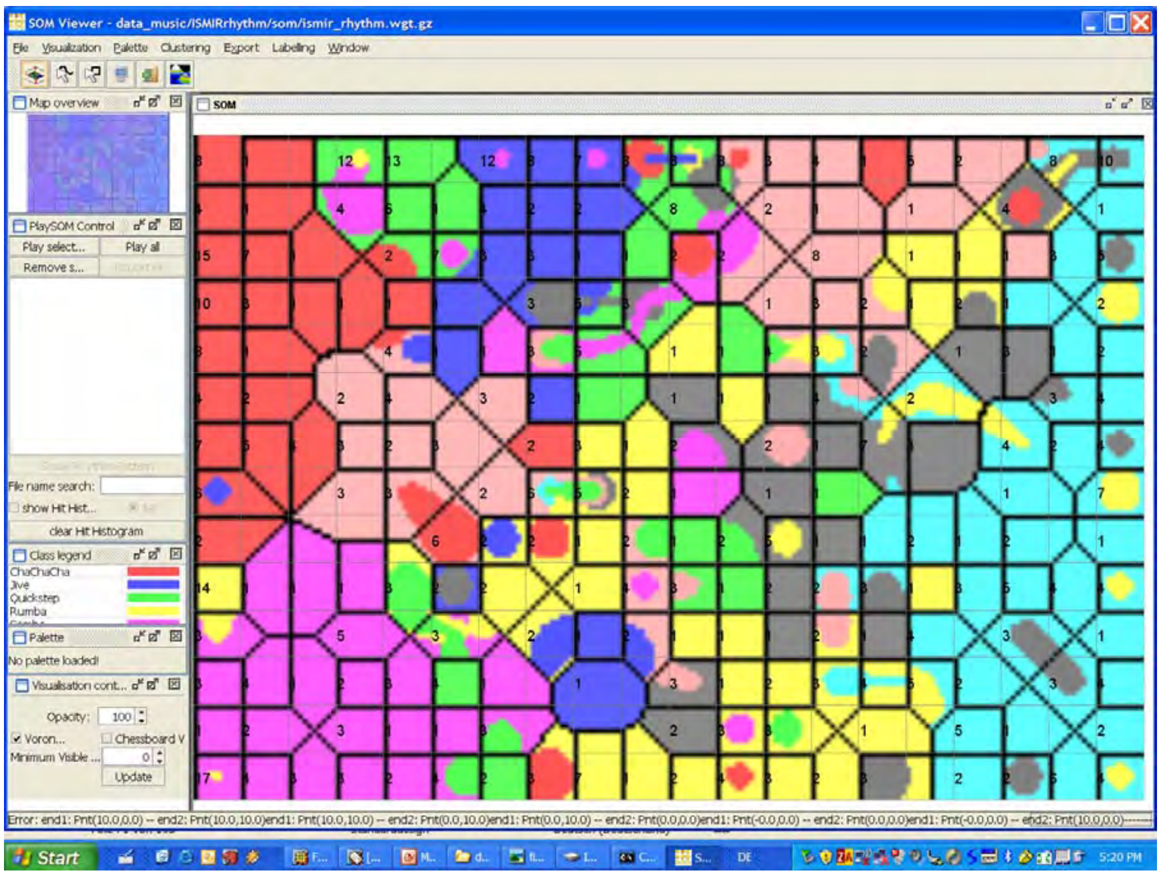
Class Coloring: Attractor Flooding

- Step 4: identify regions and attractors



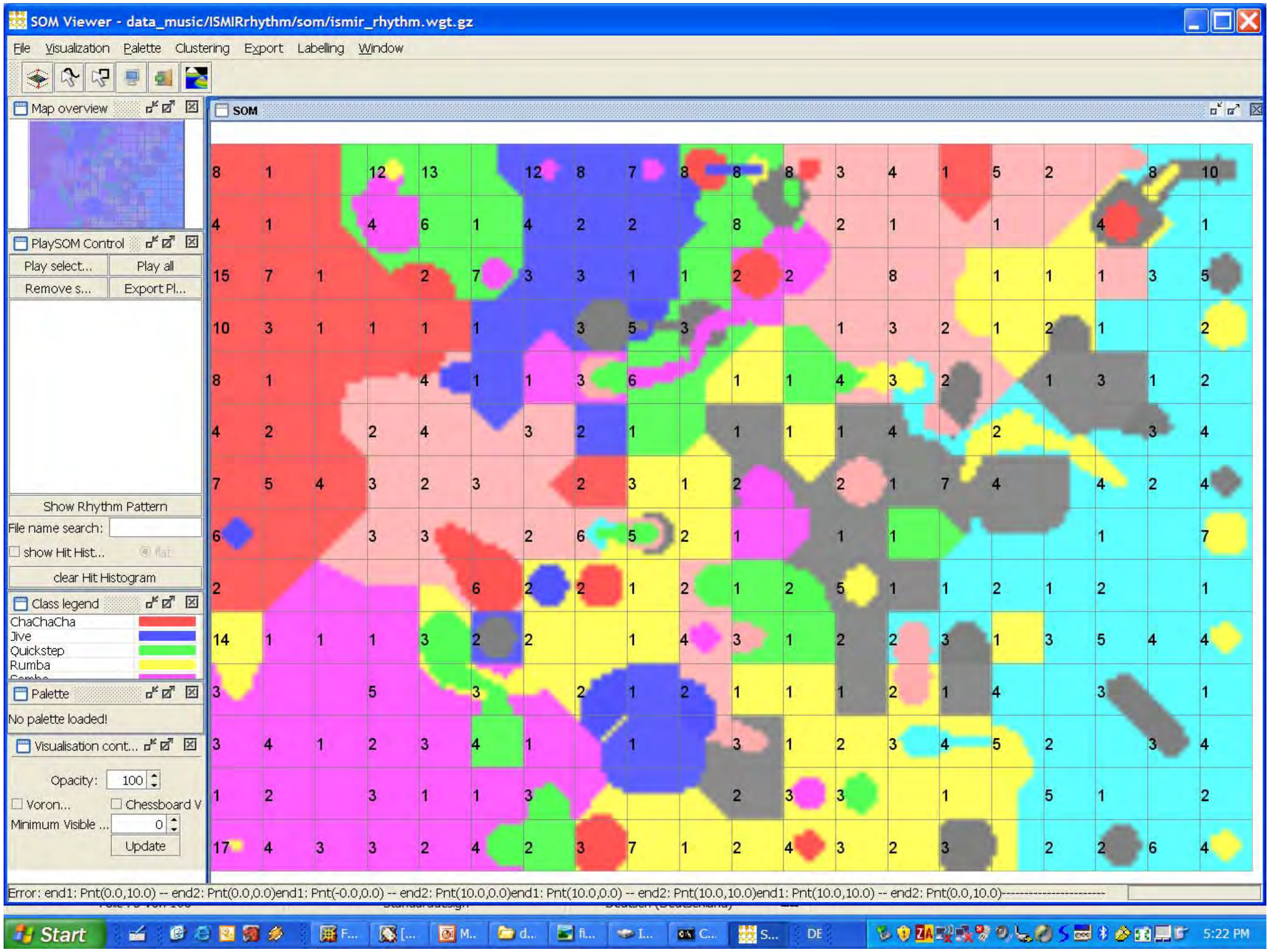
Class Coloring: Attractor Flooding

- Step 5: flood-fill along attractors according to class frequency



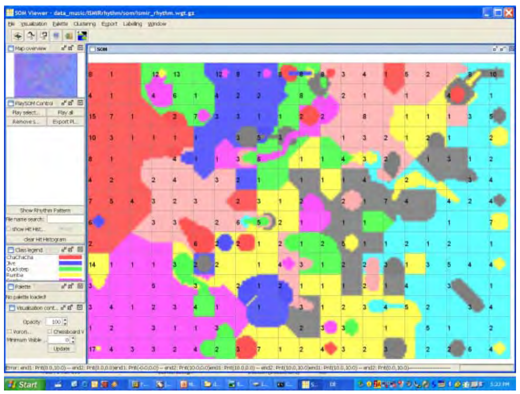
Class Coloring: Attractor Flooding

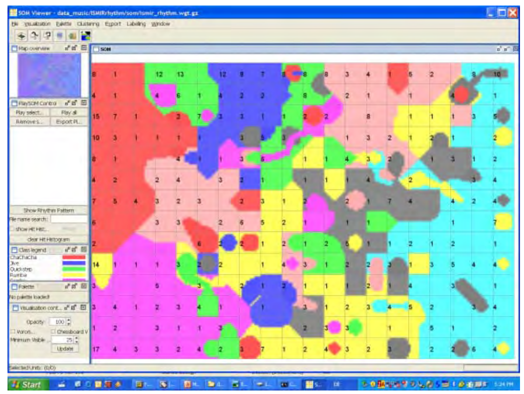
- final:

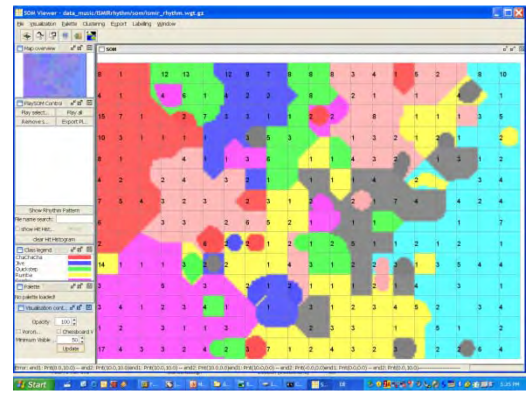


Class Coloring: Attractor Flooding

- Class frequency thresholding:





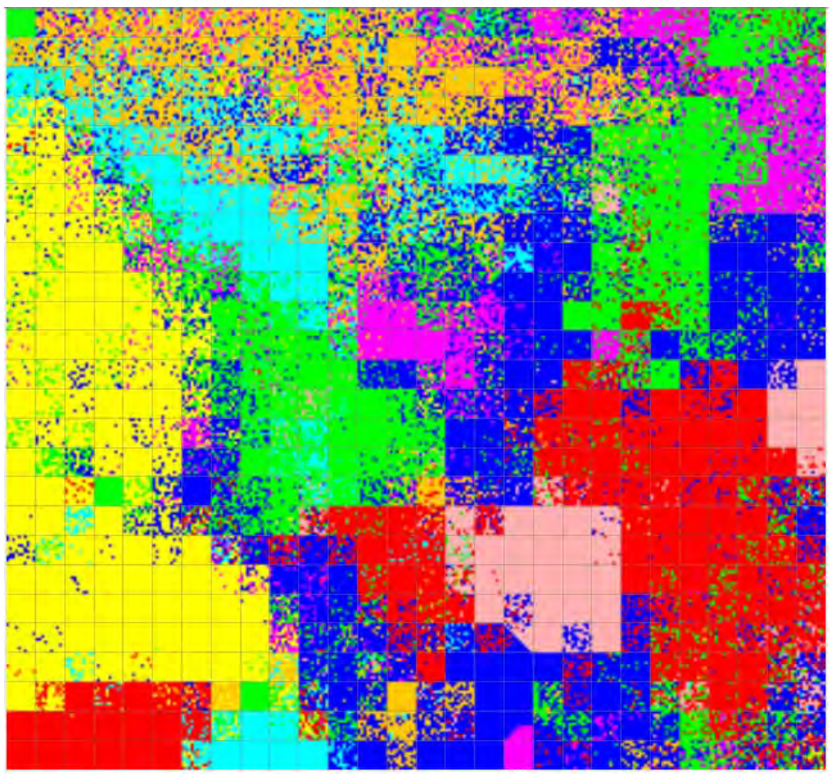


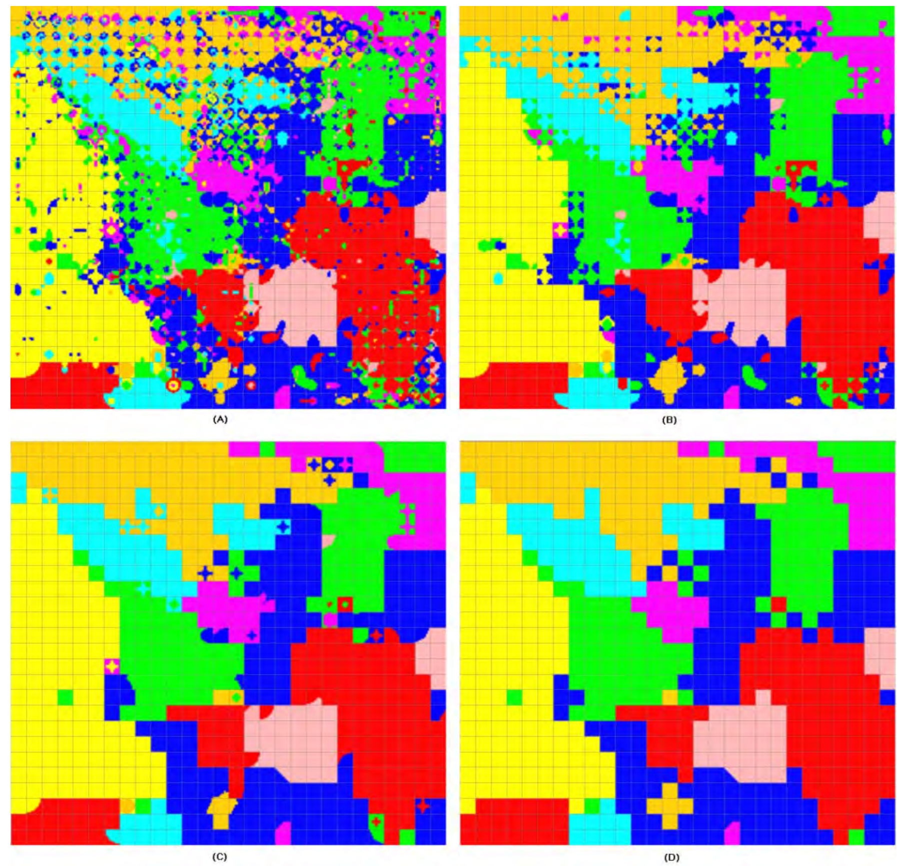
100% 50% 25%

Class Coloring

Radio Search data set:

chessboard and attractor flooding





Outline

- Overview of visualization types
- Visualizing the SOM
  - Codebook projection
  - Adaptive Coordinates
- Visualizations on the SOM
  - Textual information
  - Density
  - Distances
  - Class info
  - Attributes
  - Clustering of the SOM

Outline

Visualizations on the SOM

- Textual information
- Density
- Distances
- Class info
- Attributes
  - Component Planes
  - Clustering of Component Planes
  - Metro Maps
  - Vectorfields: grouped Flow
- Clustering of the SOM

Attributes

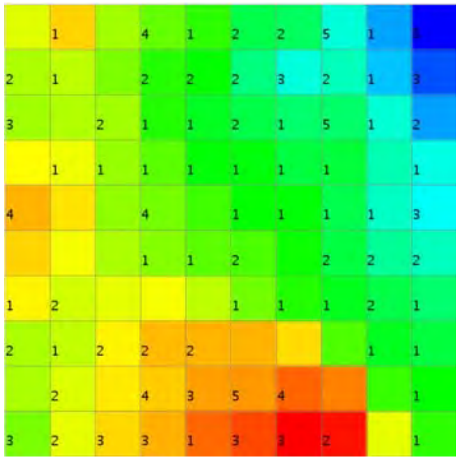
- Analysis of individual attributes or groups of attributes
- Distribution of attribute values
- Correlation between attribute values
- different visualizations
  - component planes
  - clustering of component planes
  - metro maps
  - (component-based flow)

Attributes: Component Planes

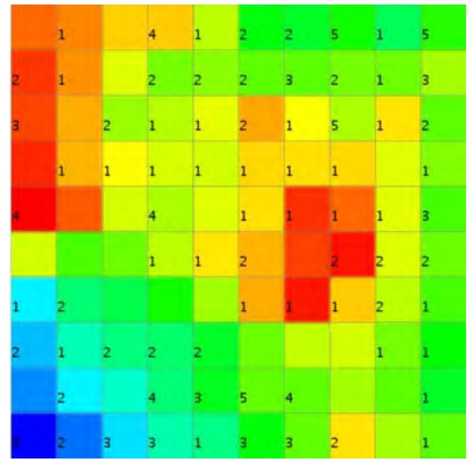
- Component Planes
- „Horizontal slice“: color each unit according to the value of a given attribute
- Analyze regularity of distribution:
  - clear gradient
  - islands with high/low value
  - quasi-random, no structure
  - analyze correlations

Attributes: Component Planes

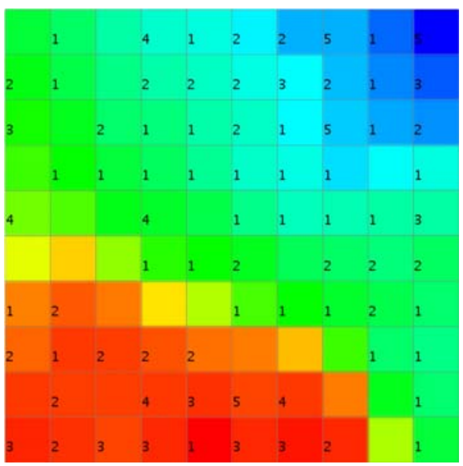
- Iris Dataset – Component Planes



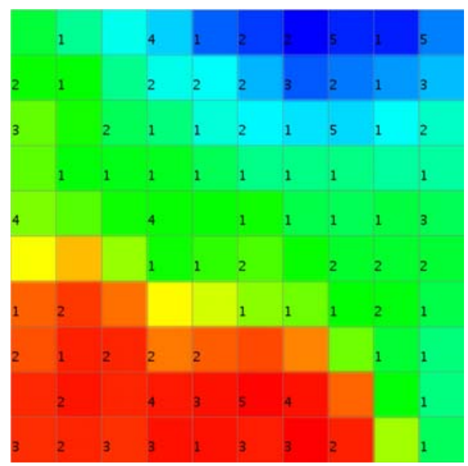
sepal length



sepal width



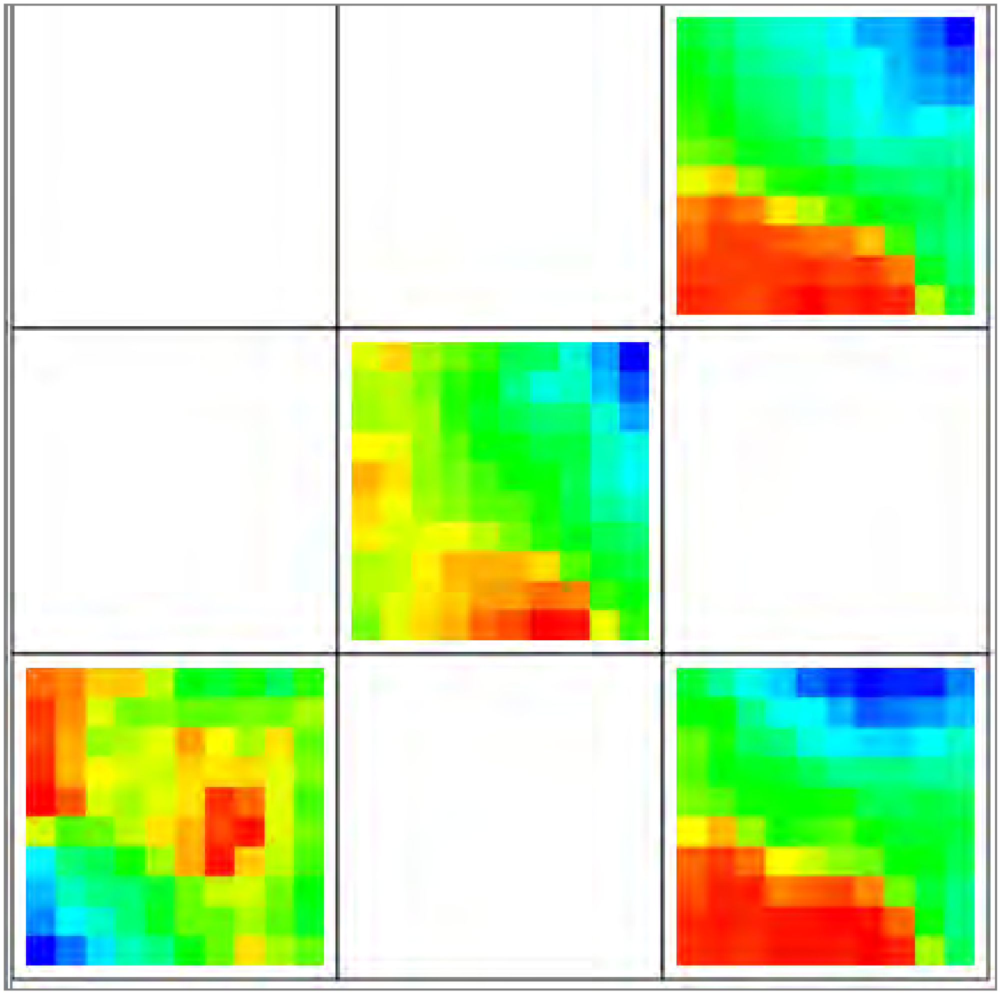
petal length



petal width

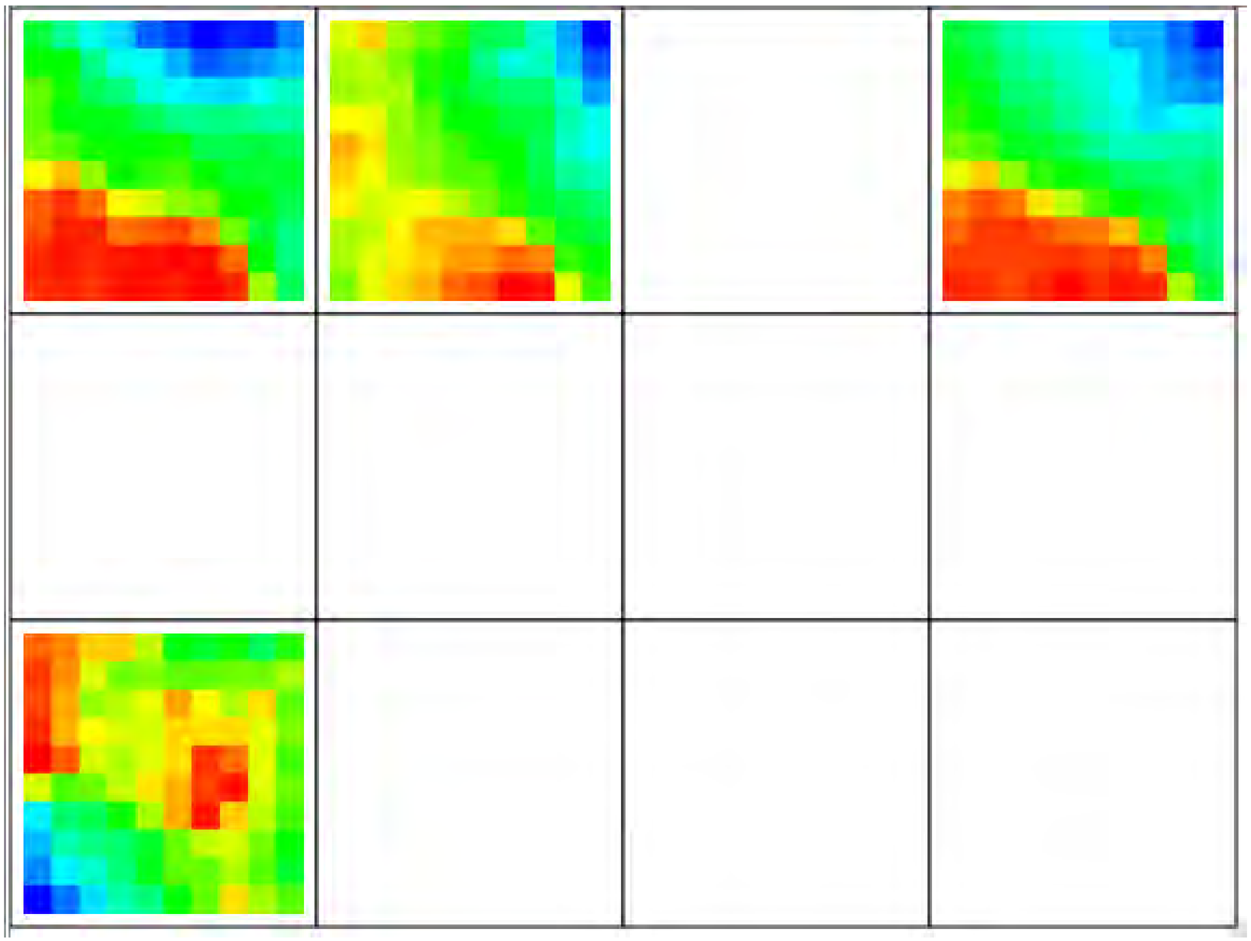
Attributes: Component Planes

- Iris Dataset – Clustered Component Planes



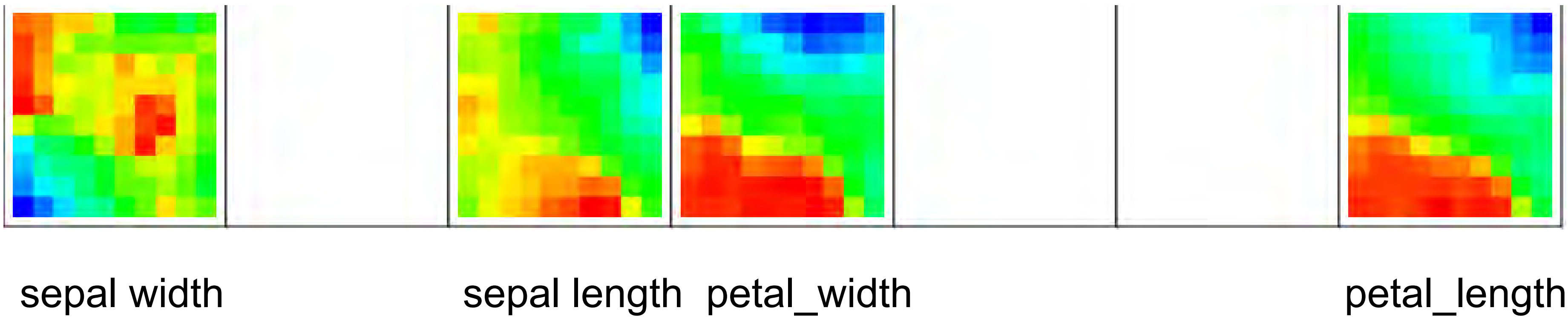
Attributes: Component Planes

- Iris Dataset – Clustered Component Planes



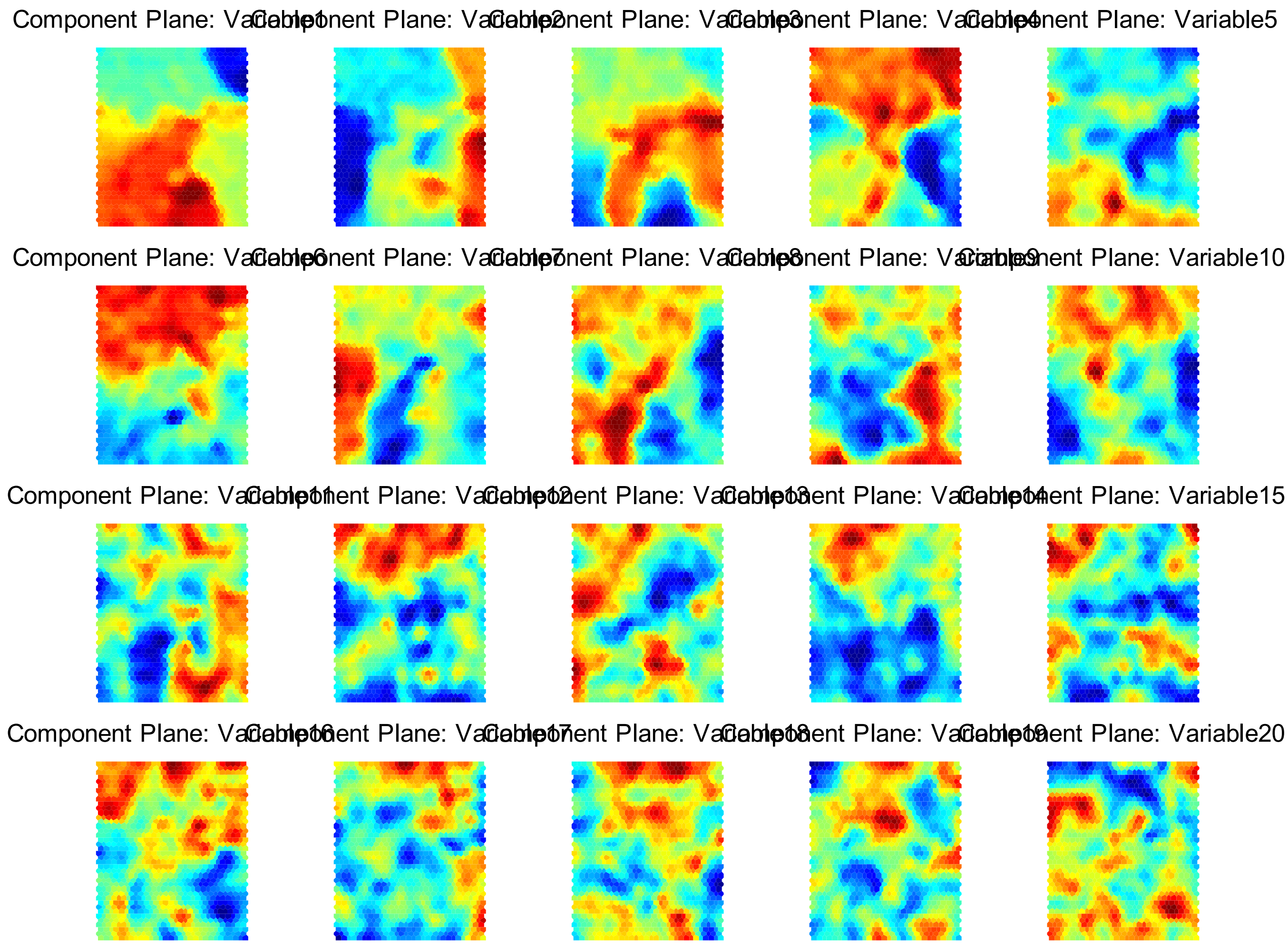
Attributes: Component Planes

- Iris Dataset – Clustered Component Planes
  - linear 7x1 SOM



Attributes: Component Planes

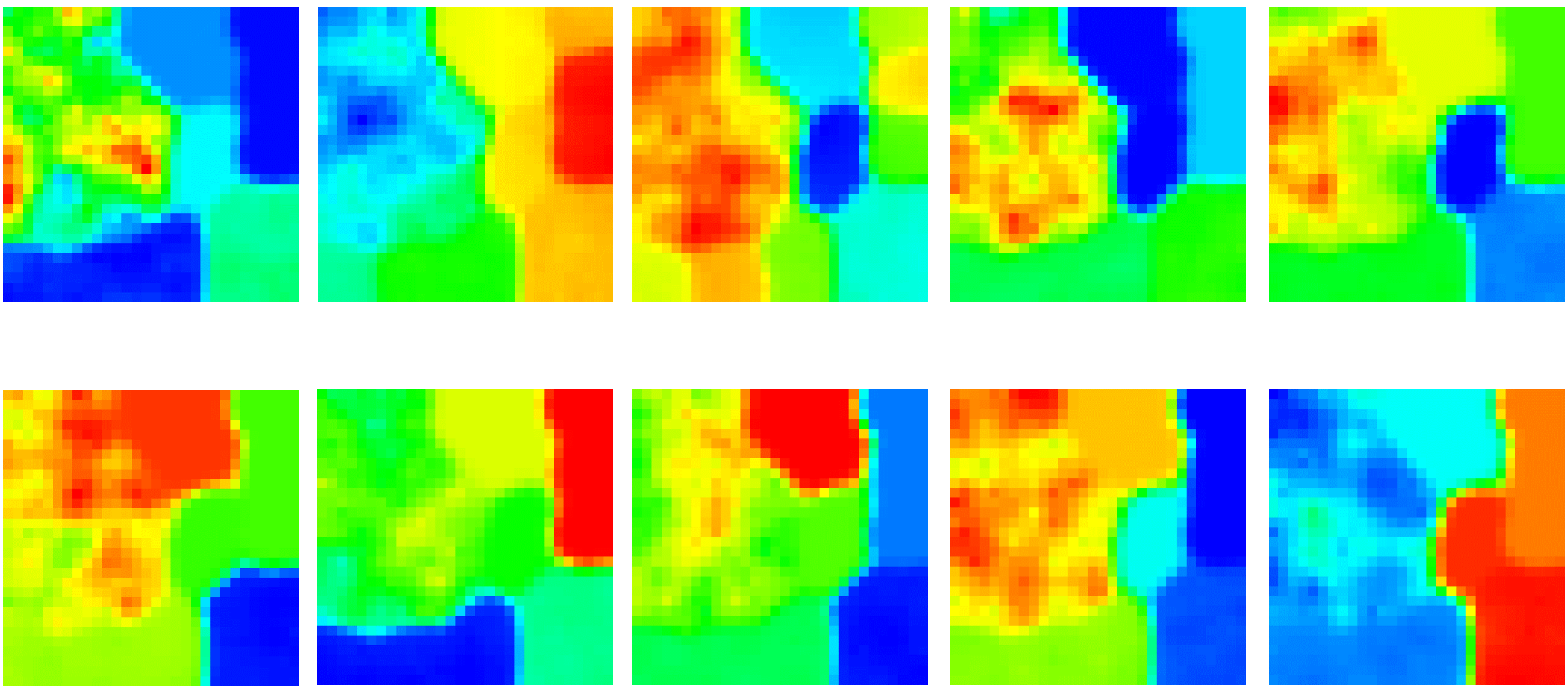
GP10



GP10 Component Planes
Georg Pölzlbauer, 7/26/2005

Attributes: Component Planes

- 10-Clusters Dataset
- 10 clusters in 10-d space



GP10 Component Planes

Georg Pölzlbauer, 7/26/2005

Outline

Visualizations on the SOM

- Textual information
- Density
- Distances
- Class info
- Attributes
  - Component Planes
  - Clustering of Component Planes
  - Metro Maps
  - Vectorfields: grouped Flow
- Clustering of the SOM

Attributes: MetroMaps

- Component Planes provide overview of attribute value distribution across SOM
- Multiple images (1 per attribute)
- Difficult to comprehend
- Hard to understand correlations between attributes
- MetroMaps
  - concept of skewed distances
  - simplified structure
  - aggregation of correlated component planes
  - overlay to any colored SOM visualization

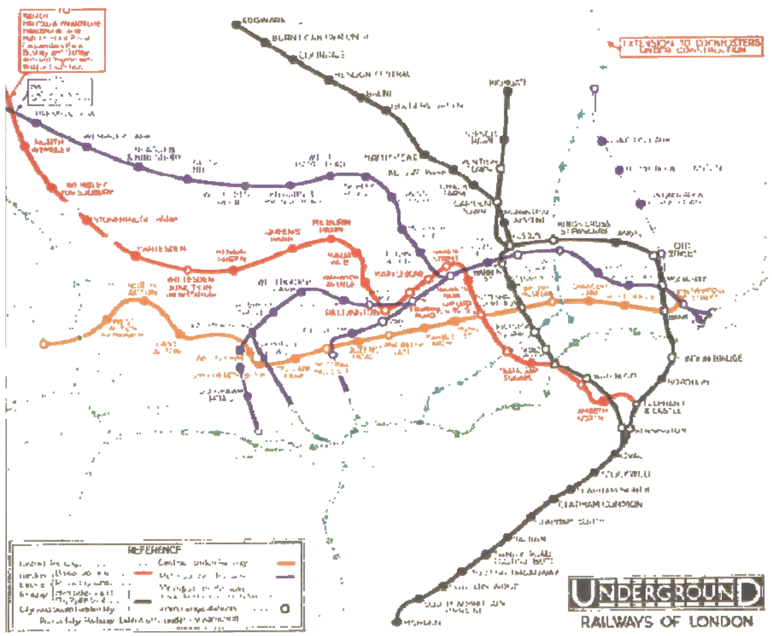
Attributes: MetroMaps

- Robert Neumayer, Rudolf Mayer, Georg Pölzlbauer, and Andreas Rauber. The metro visualisation of component planes for self-organising maps. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'07)*, Orlando, FL, USA, August 12-17 2007. IEEE Computer Society.
- Robert Neumayer, Rudolf Mayer, and Andreas Rauber. Component selection for the metro visualisation of the SOM. In *Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM'07)*, Bielefeld, Germany, September 3-6 2007.

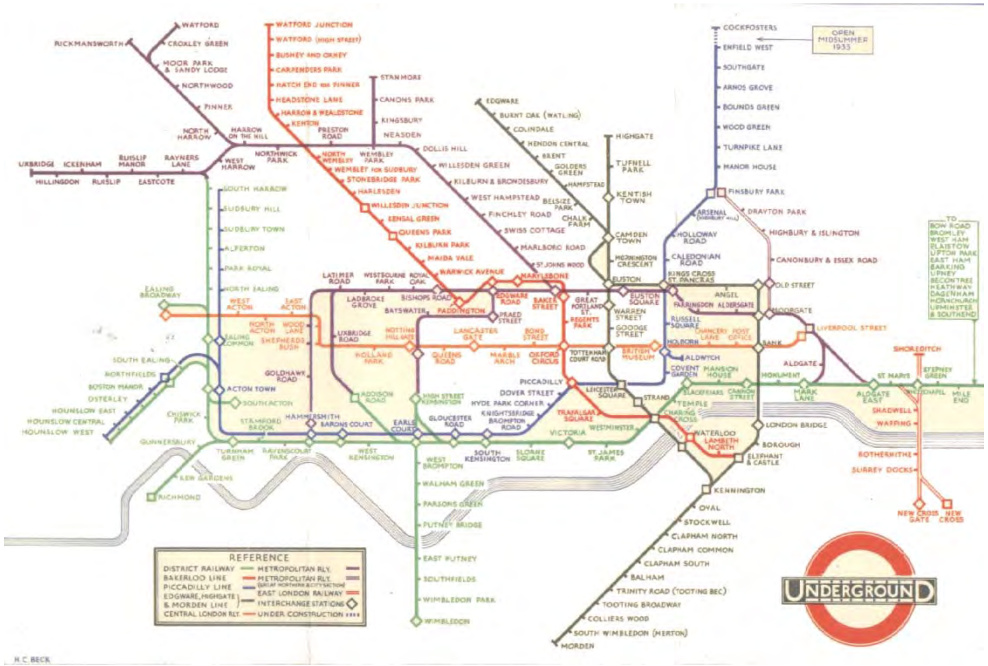
Attributes: MetroMaps

- London Metro Map

1932



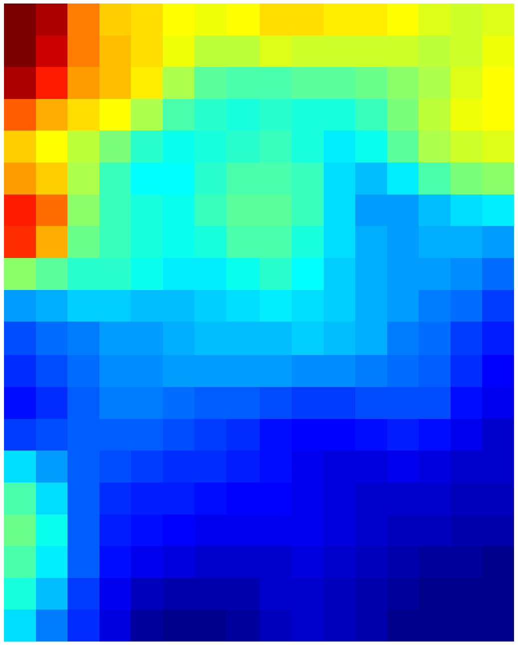
1933



Attributes: MetroMaps

Component Planes

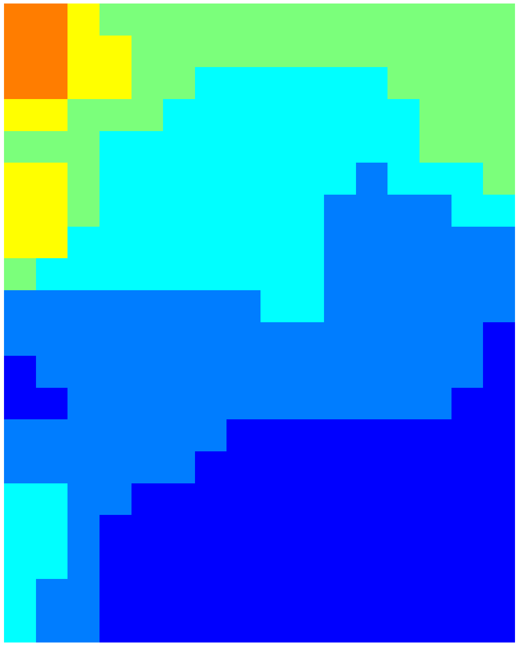
- Single component plane: visualization of 1 model vector component across map
- continuous gradients



Attributes: MetroMaps

Discretization

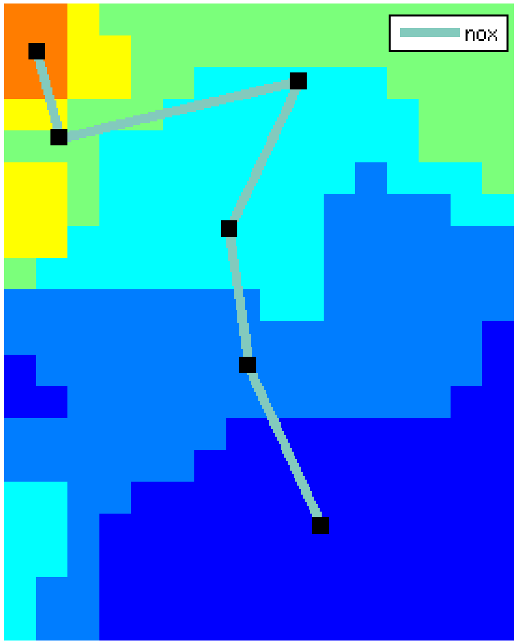
- Discretization of values
- Binning into n bins



Attributes: MetroMaps

Component Lines

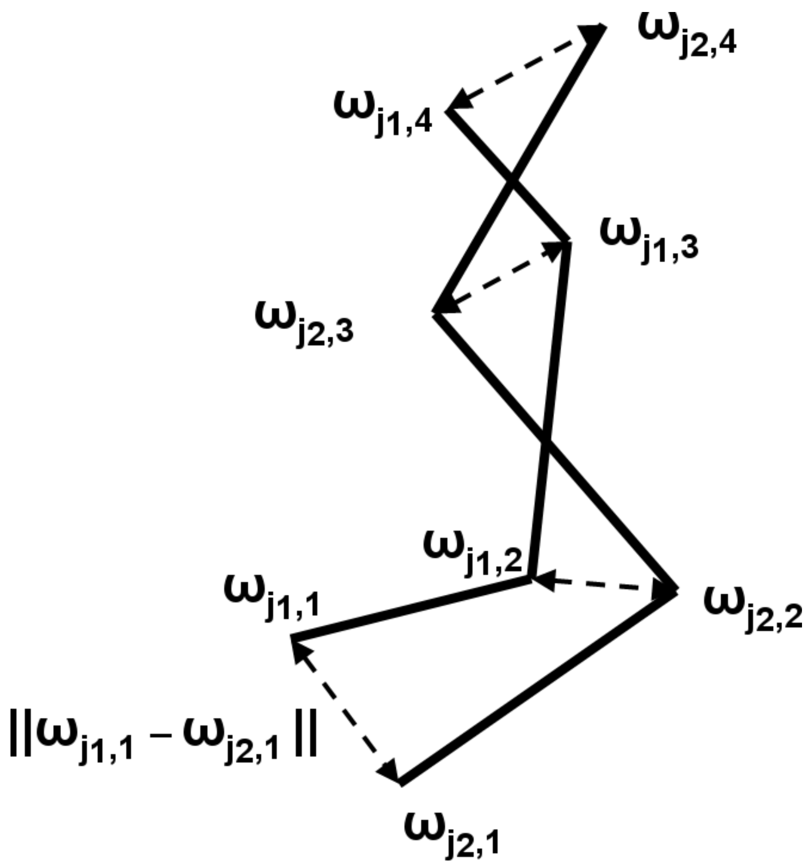
- Computation of centers of gravity
- interconnecting lines
- revealing the gradient of single components



Attributes: MetroMaps

Aggregation

- Calculate distance between component lines
- based on minimum pairwise distances
- Cluster component lines
- Visualization of aggregated subset



Attributes: MetroMaps

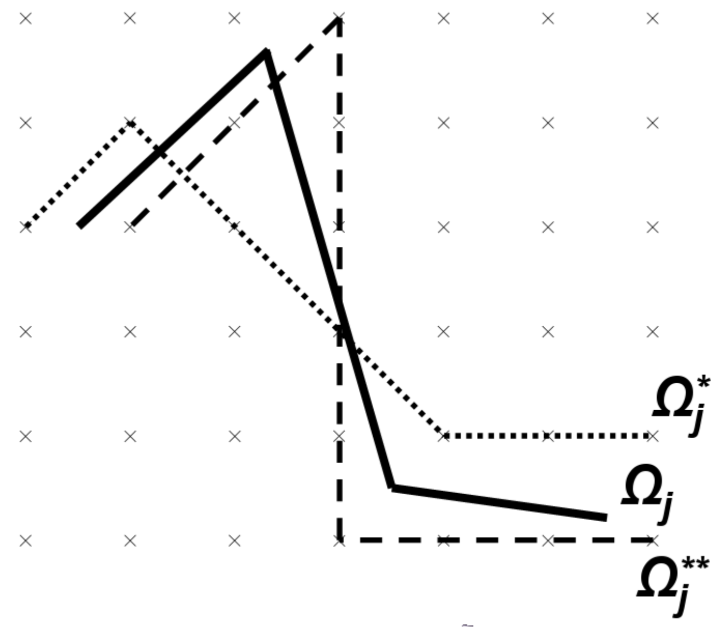
Selection

- use only components with consistent scattering
- ratio of number of bins divided by number of closed regions
- select component lines that have a high ration, i.e. little to no scattering /local minima

Attributes: MetroMaps

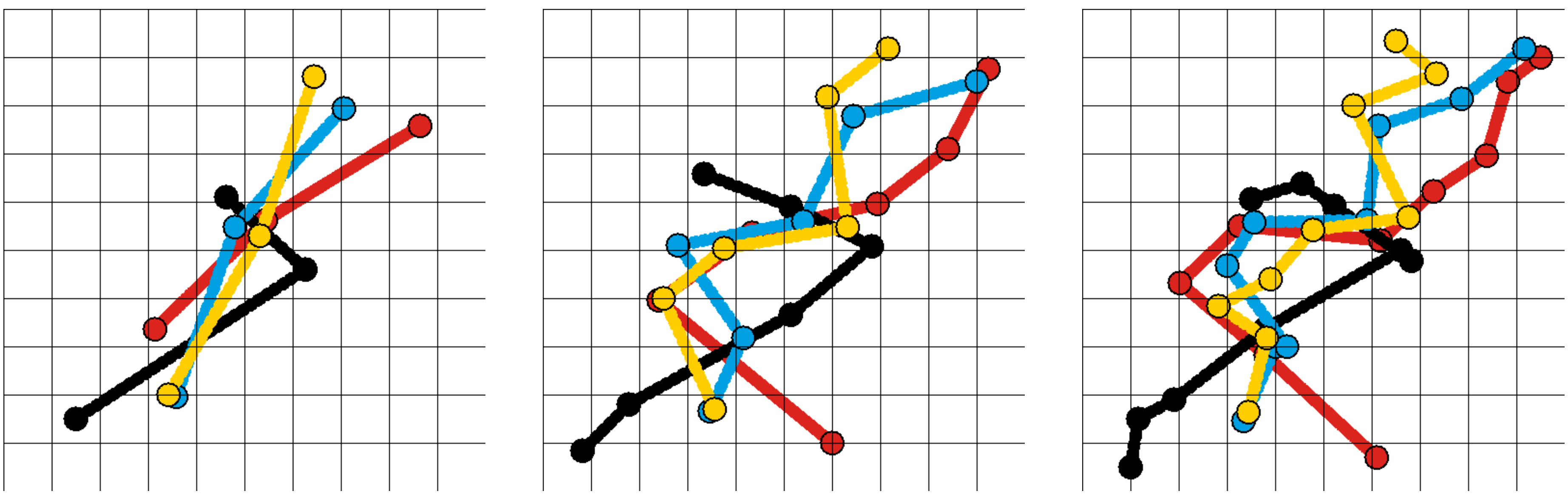
Snapping

- Visual enhancement
- Snapping component lines onto SOM grid
- Allowing only horizontal, vertical, and 45 degree lines
- Clearer structure



Attributes: MetroMaps

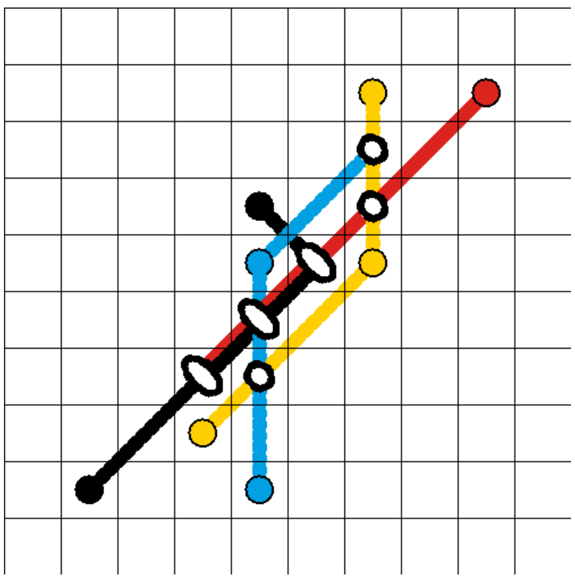
- Iris Data Set
  - Metro Maps, unsnapped



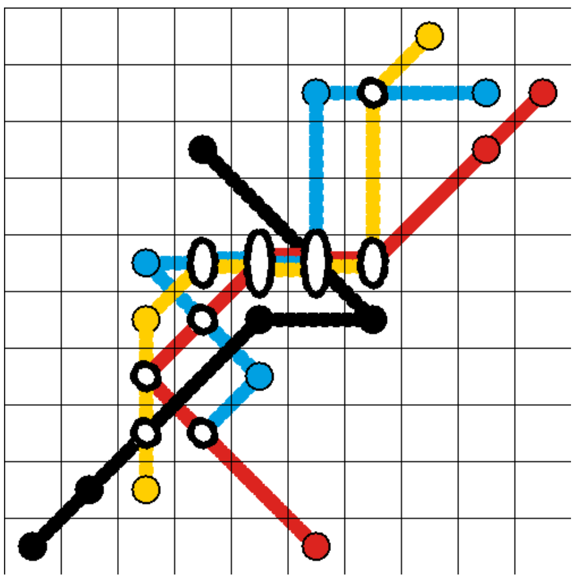
3 bins 6 bins 9 bins

Attributes: MetroMaps

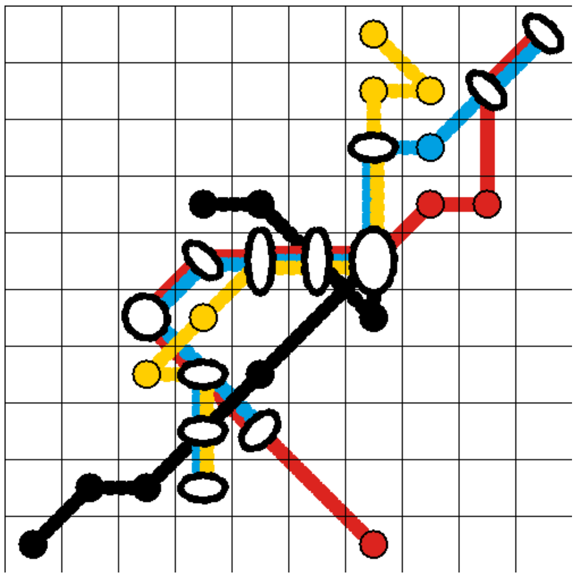
- Iris Data Set
  - Metro Maps, snapped



3 bins



6 bins



9 bins

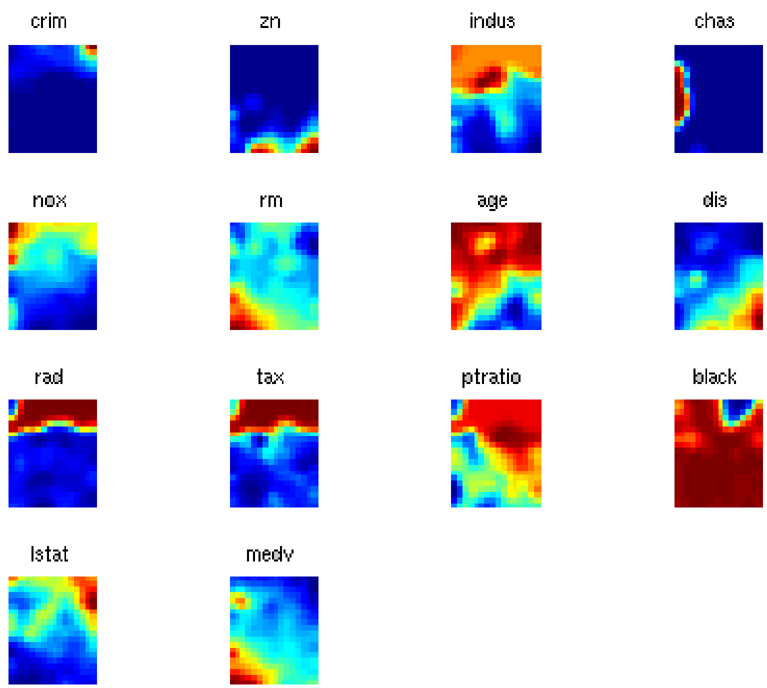
Attributes: MetroMaps

- Boston Housing Data
  - UCI Machine Learning Repository
  - describes households in Boston
  - 506 instances
  - 14 attributes
  - 20x16 SOM
  - discretization based on 6 bins
  - U-matrix as background visualization

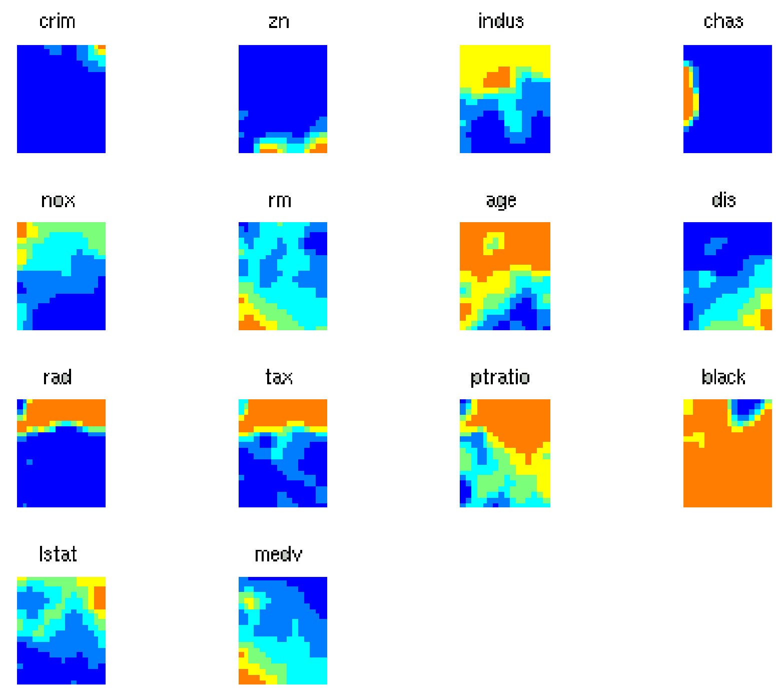
Attributes: MetroMaps

- Boston Housing - Discretization

component planes

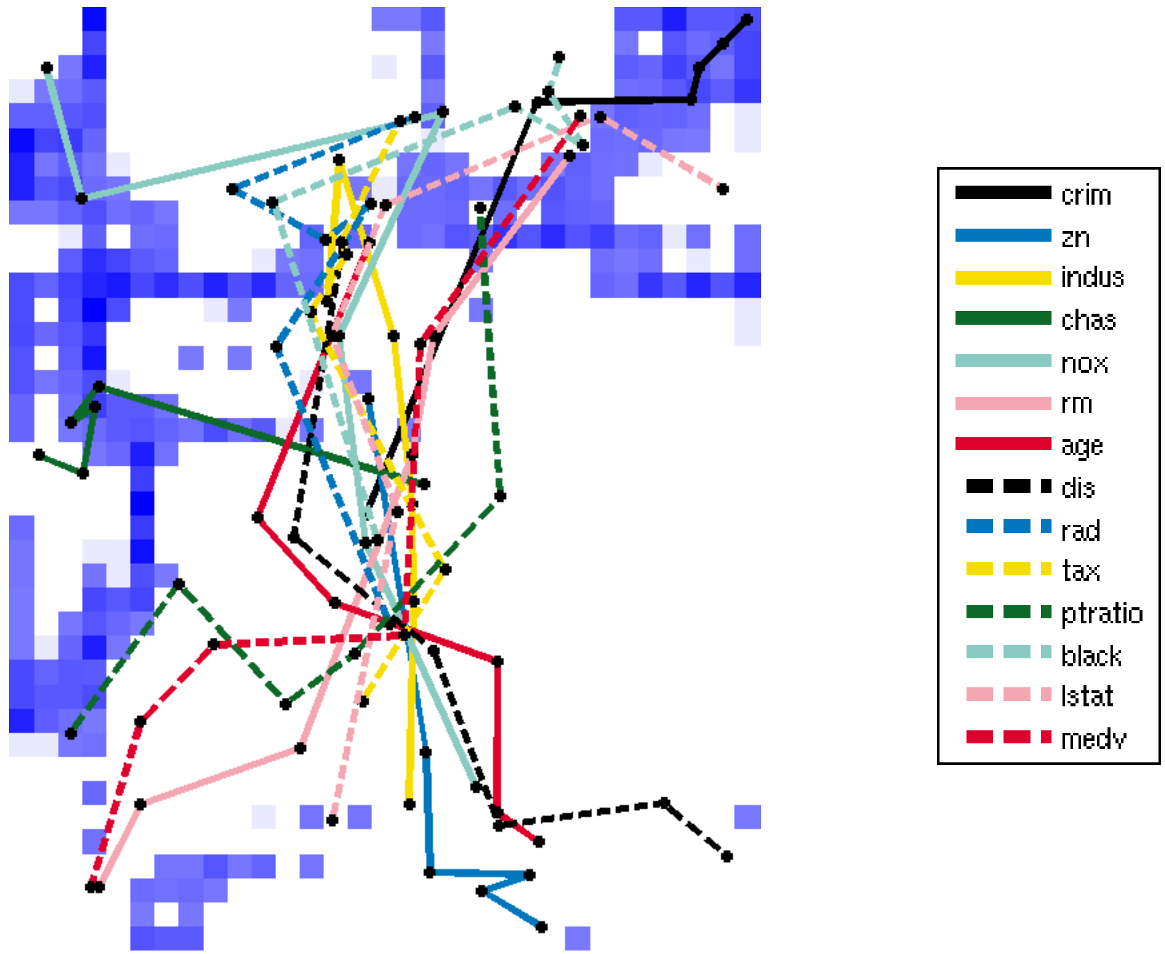


binned



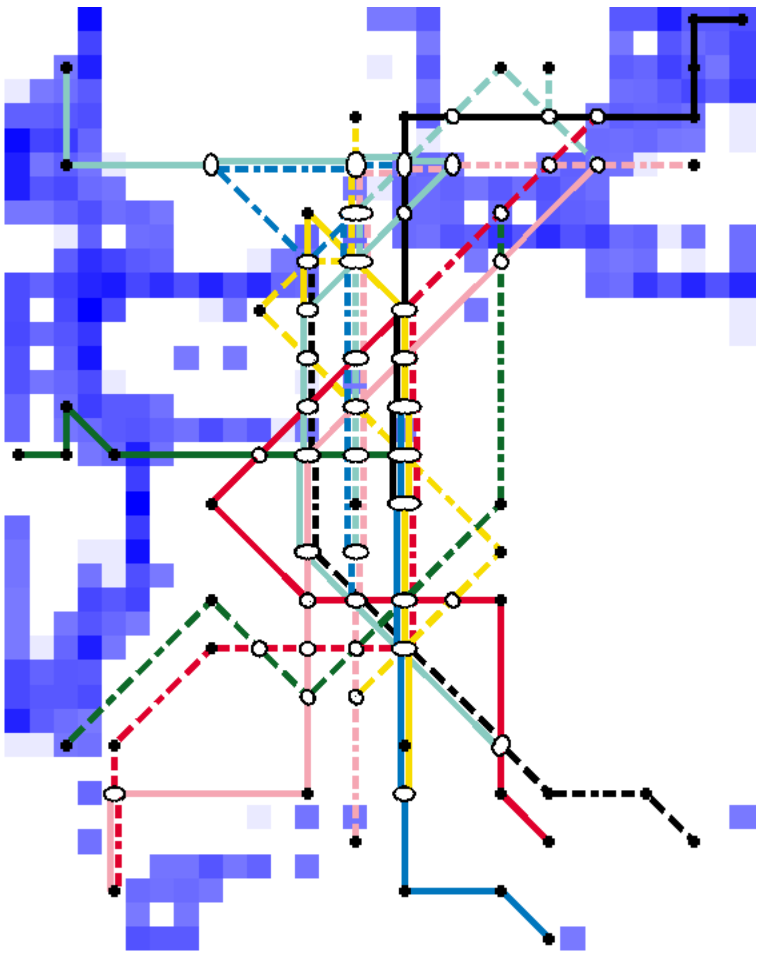
Attributes: MetroMaps

- Boston Housing - Component Lines



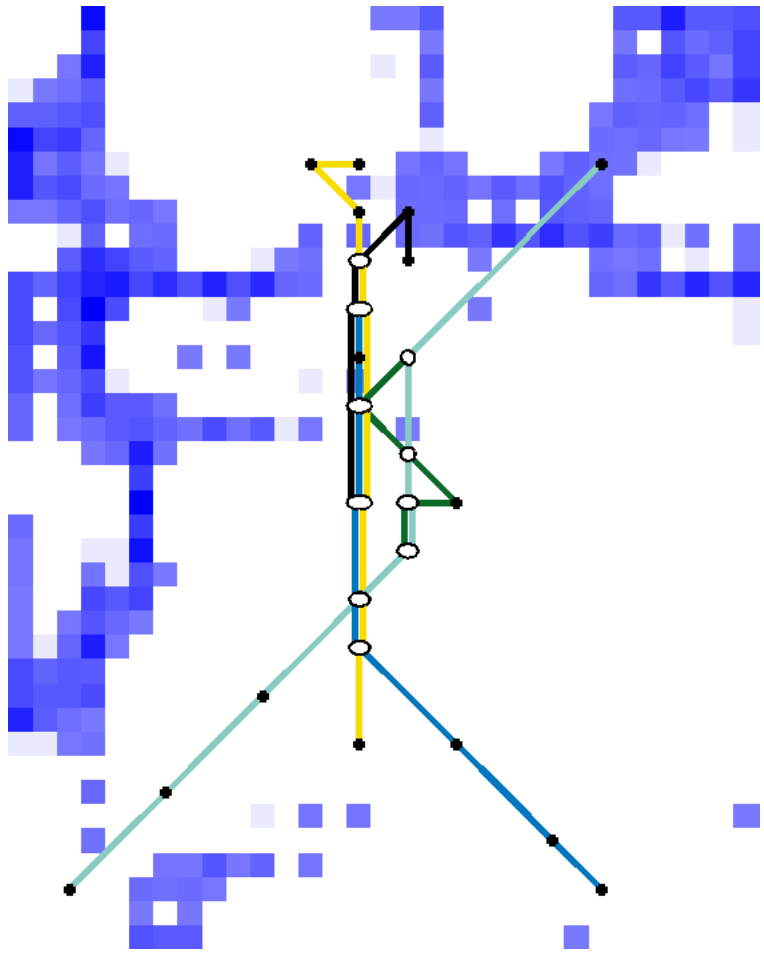
Attributes: MetroMaps

- Boston Housing - Snapped



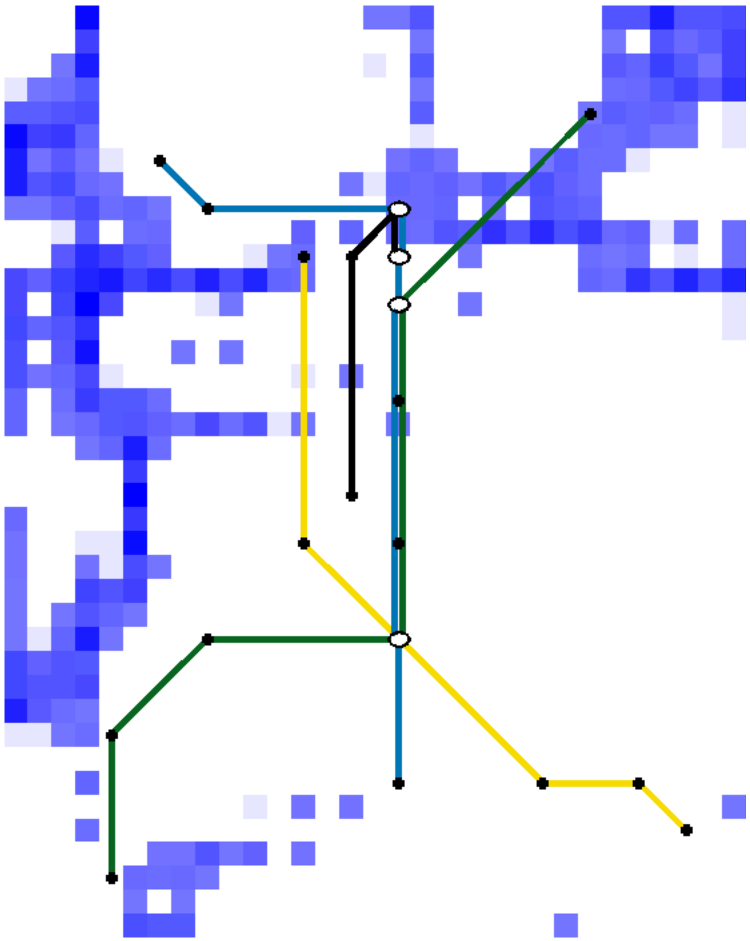
Attributes: MetroMaps

- Boston Housing - Aggregated



Attributes: MetroMaps

- Boston Housing - Selected



Outline

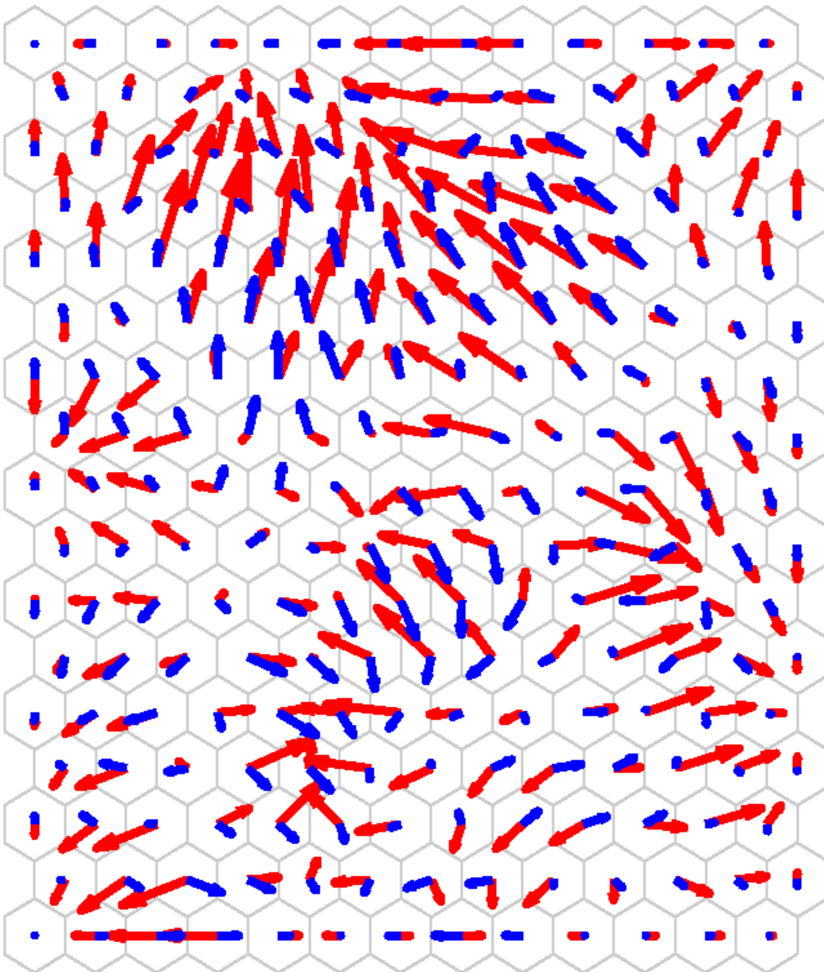
Visualisierungen der SOM

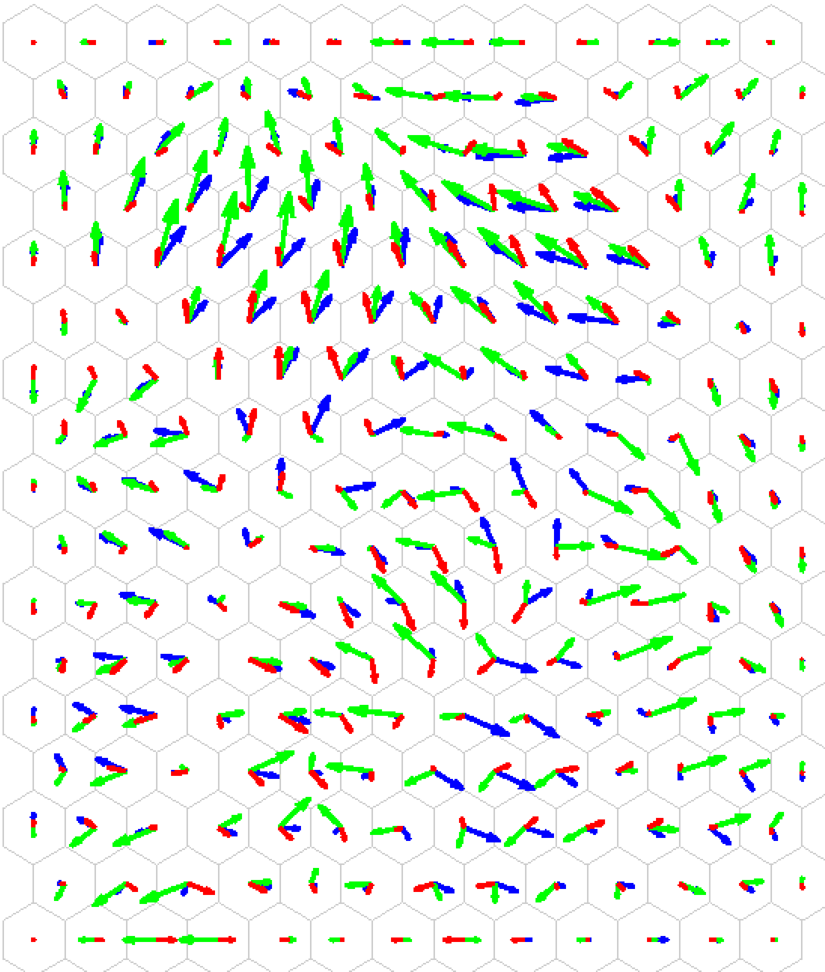
- Textuelle Informationen
- Dichte
- Distanzen
- Klasseninformation
- Attribute
  - Component Planes
  - Clustering of Component Planes
  - Metro Maps
  - Vectorfields: grouped Flow
- Clustering der SOM

Flow: Groups of Attributes

- Flow-based visualization for groups of attributes
- Attributes may be grouped by
  - clustering: data correlation
  - semantics: source, type of information:
- Up to 3 groups can be meaningfully interpretable
- Extremely powerful for hypothesis generation and validation
  - e.g. splitting control parameters and fixed measures: direction of movement from same “fixed” characteristics depending on control parameters, leading to which SOM area

Flow: Groups of Attributes





Outline

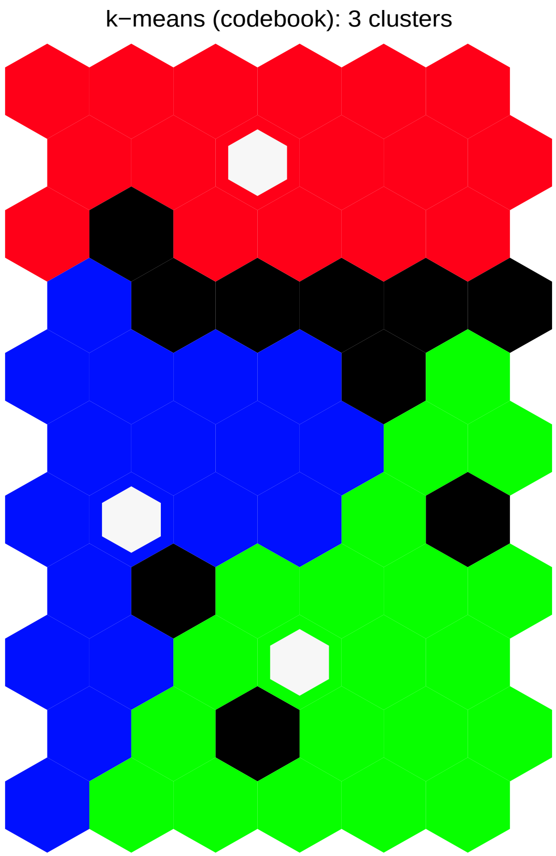
Visualizations on the SOM

- Textual information
- Density
- Distances
- Class info
- Attributes
- Clustering of the SOM
  - flat: k-means
  - hierarchical: single/complete linkage, WARD,...

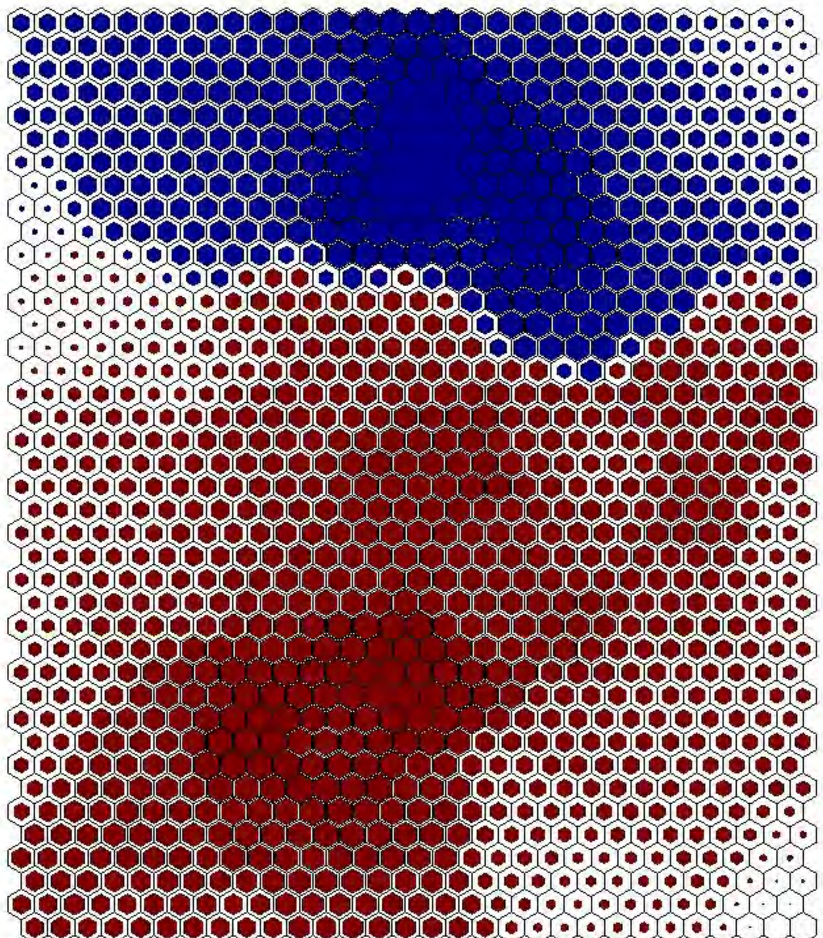
Clustering the SOM

- Clustering: Subdividing the map into regions
- k-means: generates *k* clusters
- Calculation similar to SOM
- non-deterministic: results not necessarily identical if appliedmultiple times to same map
- Juha Vesanto, Esa Alhoniemi:Clustering of the Self-Organizing Map. IEEE Transactions on Neural Networks 11(3):586-600. 2000. IEEE.
- Angela Roiger: **Analyzing, Labeling, and Interacting with SOMs for Knowledge Management**. Master Thesis, Department of Software Technology and Interactive Systems, Vienna University of Technology, March 2007.

Clustering: k-means



Iris (small)
k = 3

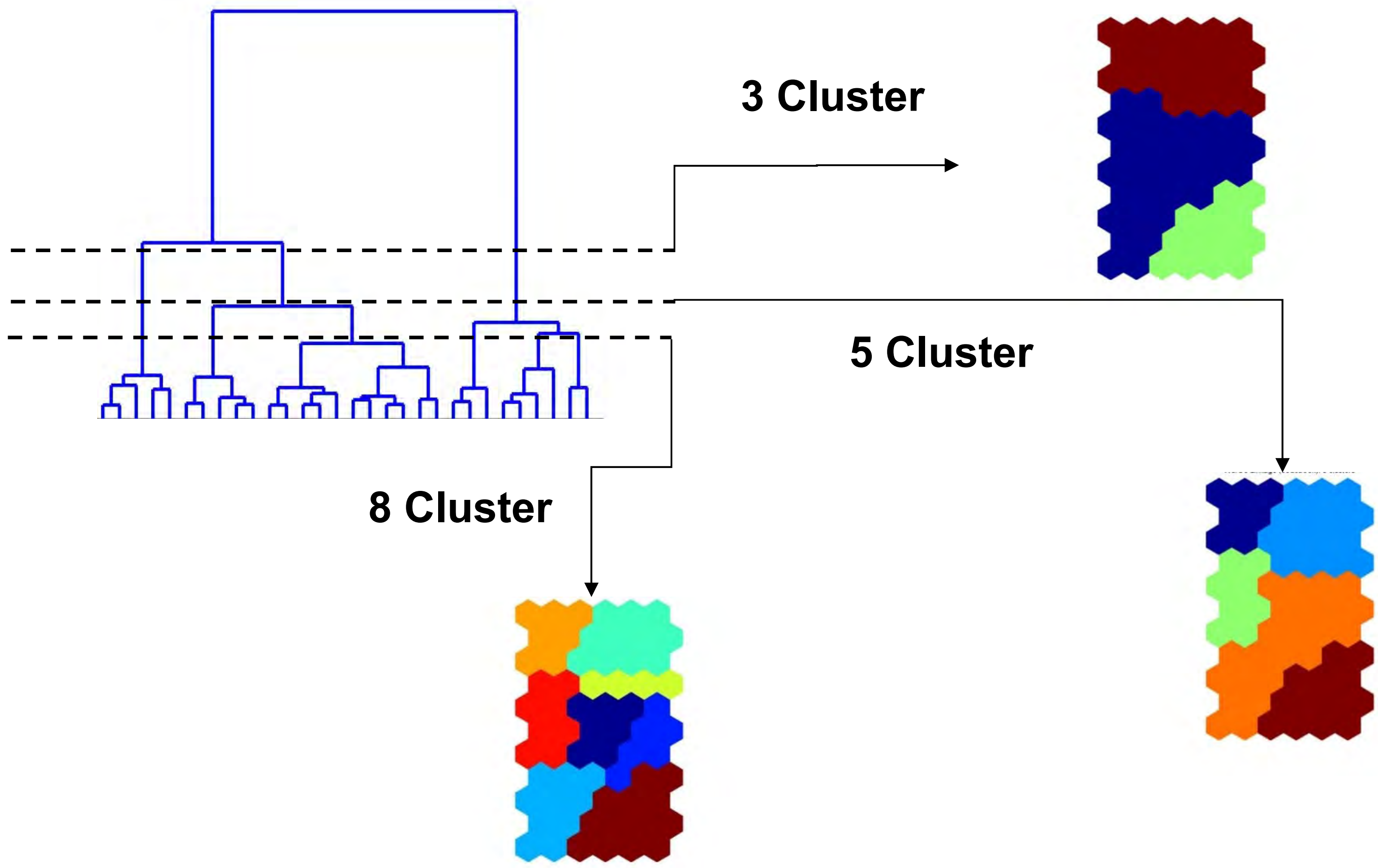


Iris (large)
k = 2

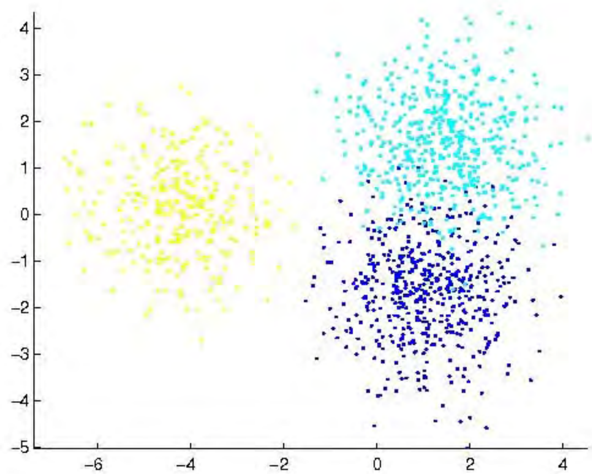
Clustering: Hierarchical

- Tree structure
- 2 clusters are merged to form higher-level aggregation
- Hierarchy can be visualized as dendrogram
- Different approaches
  - single linkage
  - complete linkage
  - WARDs clustering

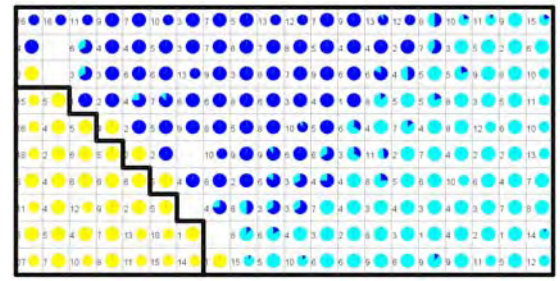
Clustering: Hierarchical



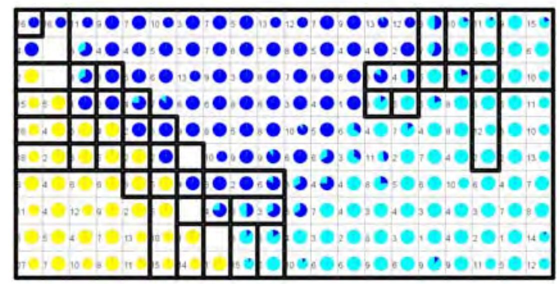
Clustering: Hierarchical



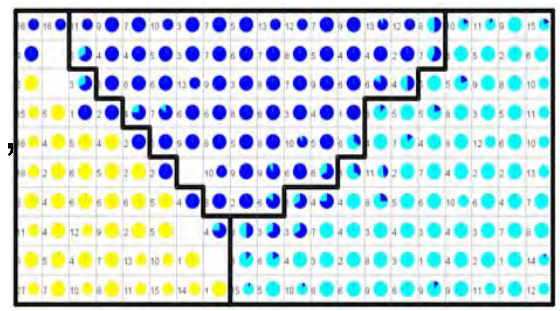
single-linkage,
2 clusters



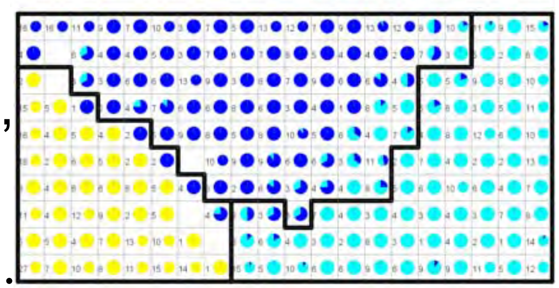
single-linkage,
40 clusters



Ward's clustering,
3 clusters

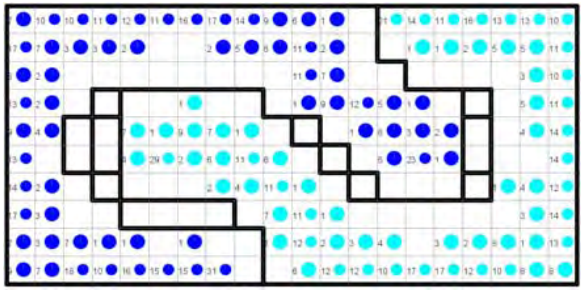


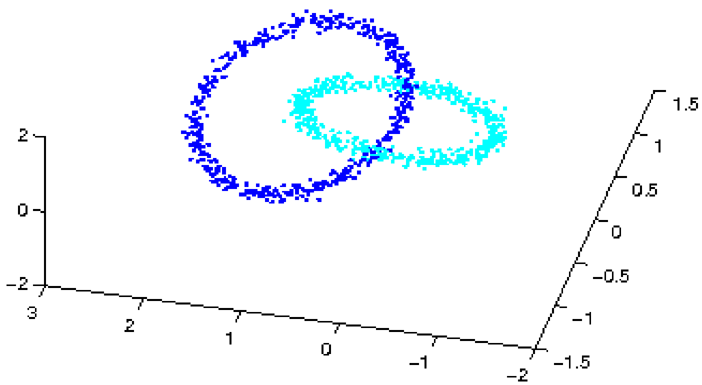
complete-linkage,
3 clusters



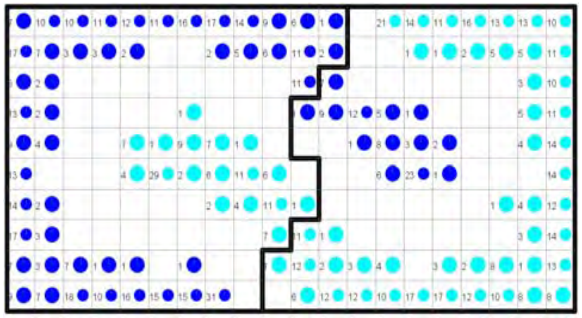
Clustering: Hierarchical

single-linkage,
11 clusters

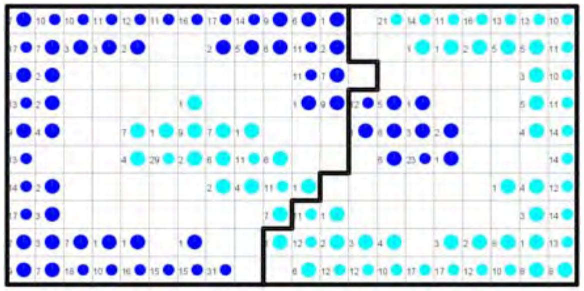




Ward clustering,
2 clusters



complete-linkage,
2 clusters



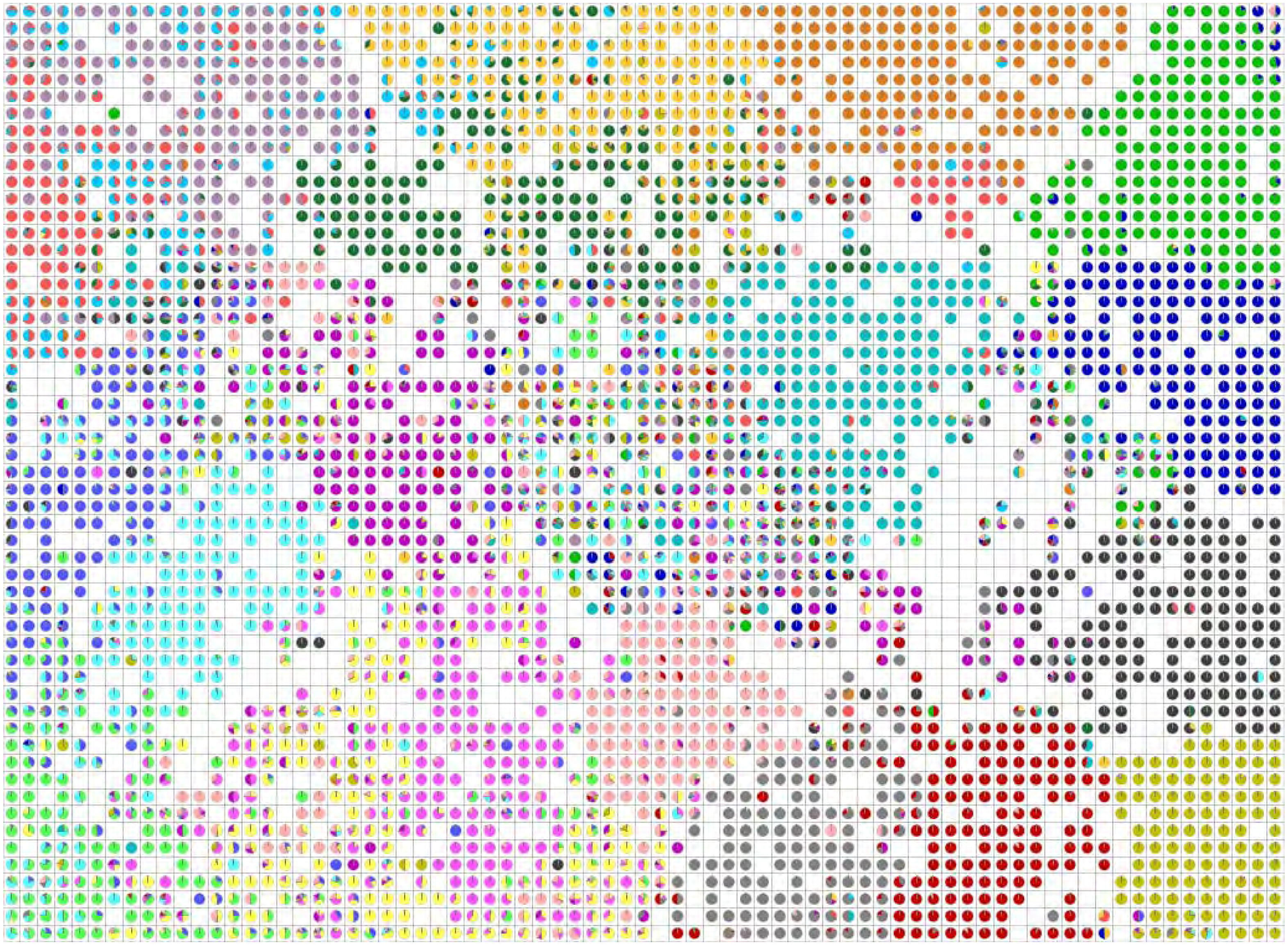
Clustering: Hierarchical

- Example: 20 Newsgroups
- Benchmark Dataset
- 1000 postings per newsgroup
- Hierarchy of newsgroups

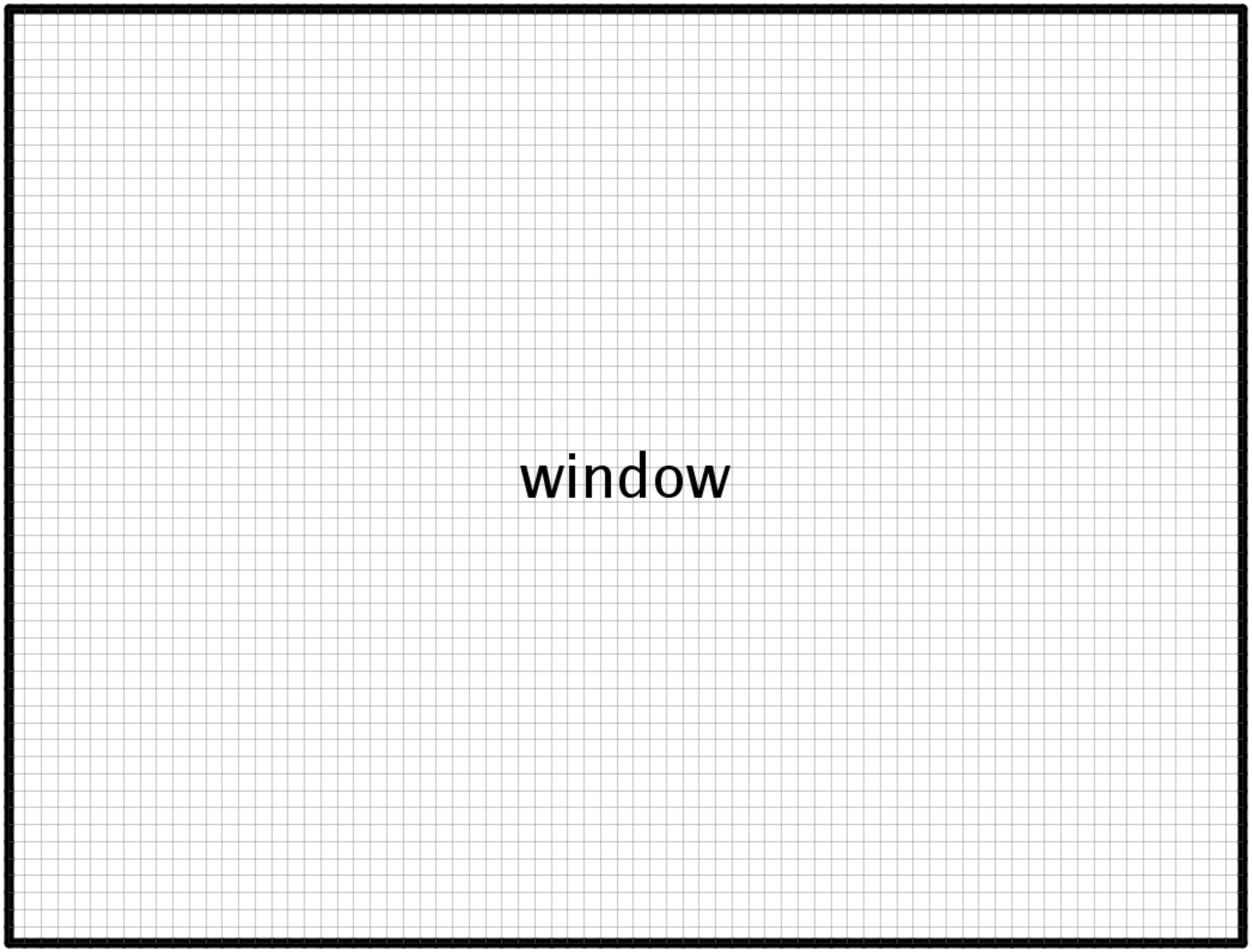
- Full-term indexing
- Stemming

alt.atheism	
comp.graphics	
comp.os.ms-windows.misc	
comp.sys.ibm.pc.hardware	
comp.sys.mac.hardware	
comp.windows.x	
misc.forsale	
rec.autos	
rec.motorcycles	
rec.sport.baseball	
rec.sport.hockey	
sci.crypt	
sci.electronics	
sci.med	
sci.space	
soc.religion.christian	
talk.politics.guns	
talk.politics.mideast	
talk.politics.misc	
talk.religion.misc	

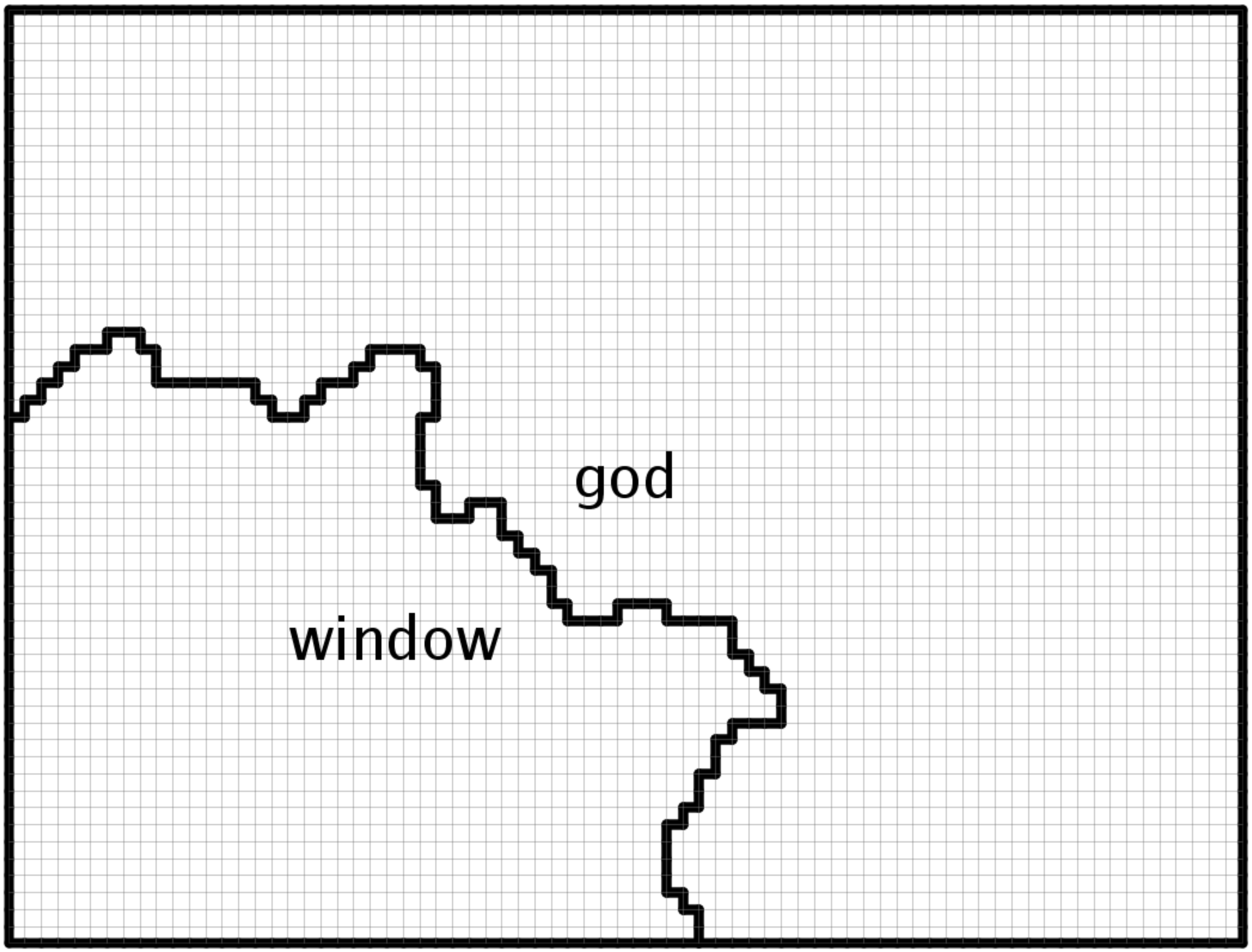
Example: 20 Newsgroups



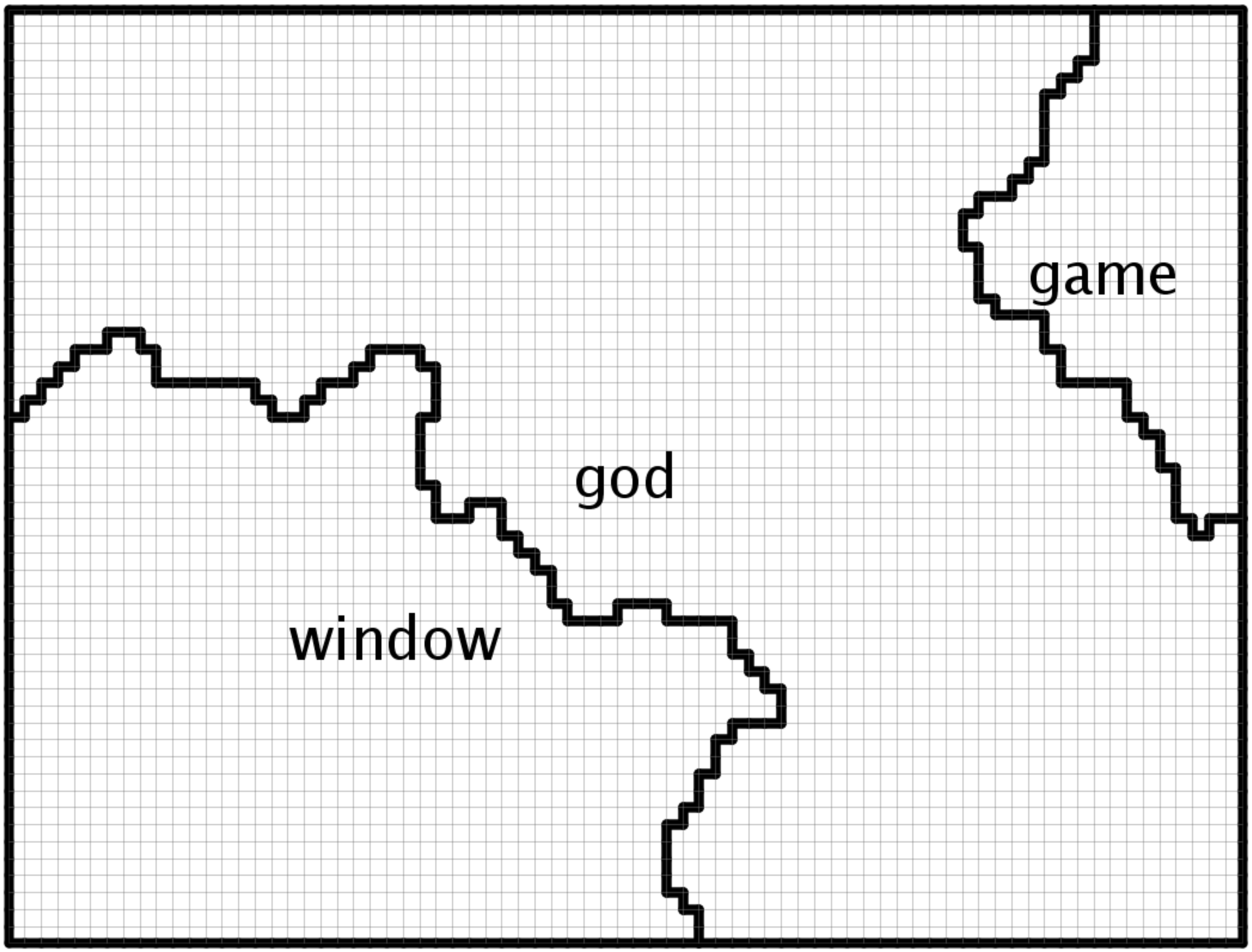
20 Newsgroups: Ward+Labels



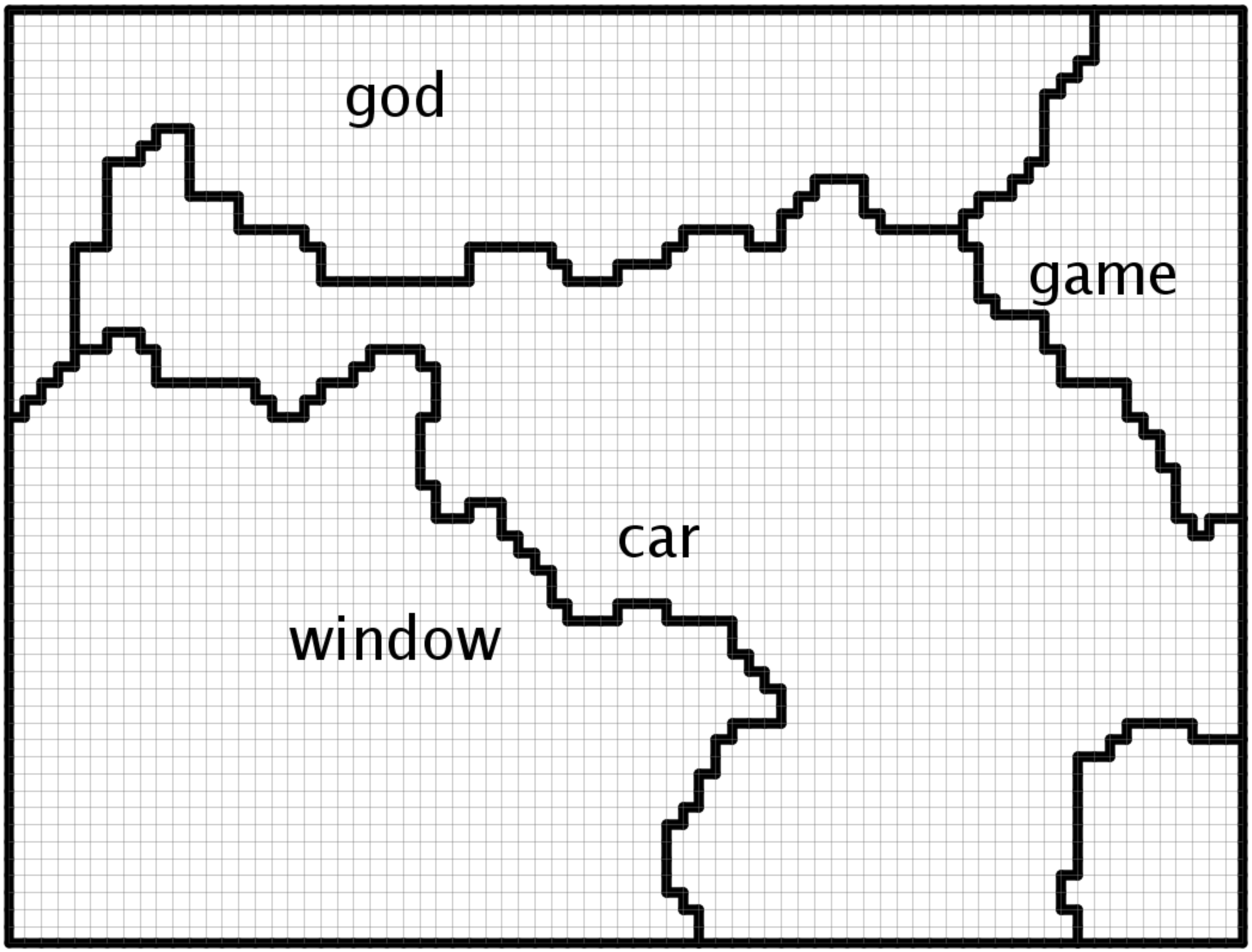
20 Newsgroups: Ward+Labels



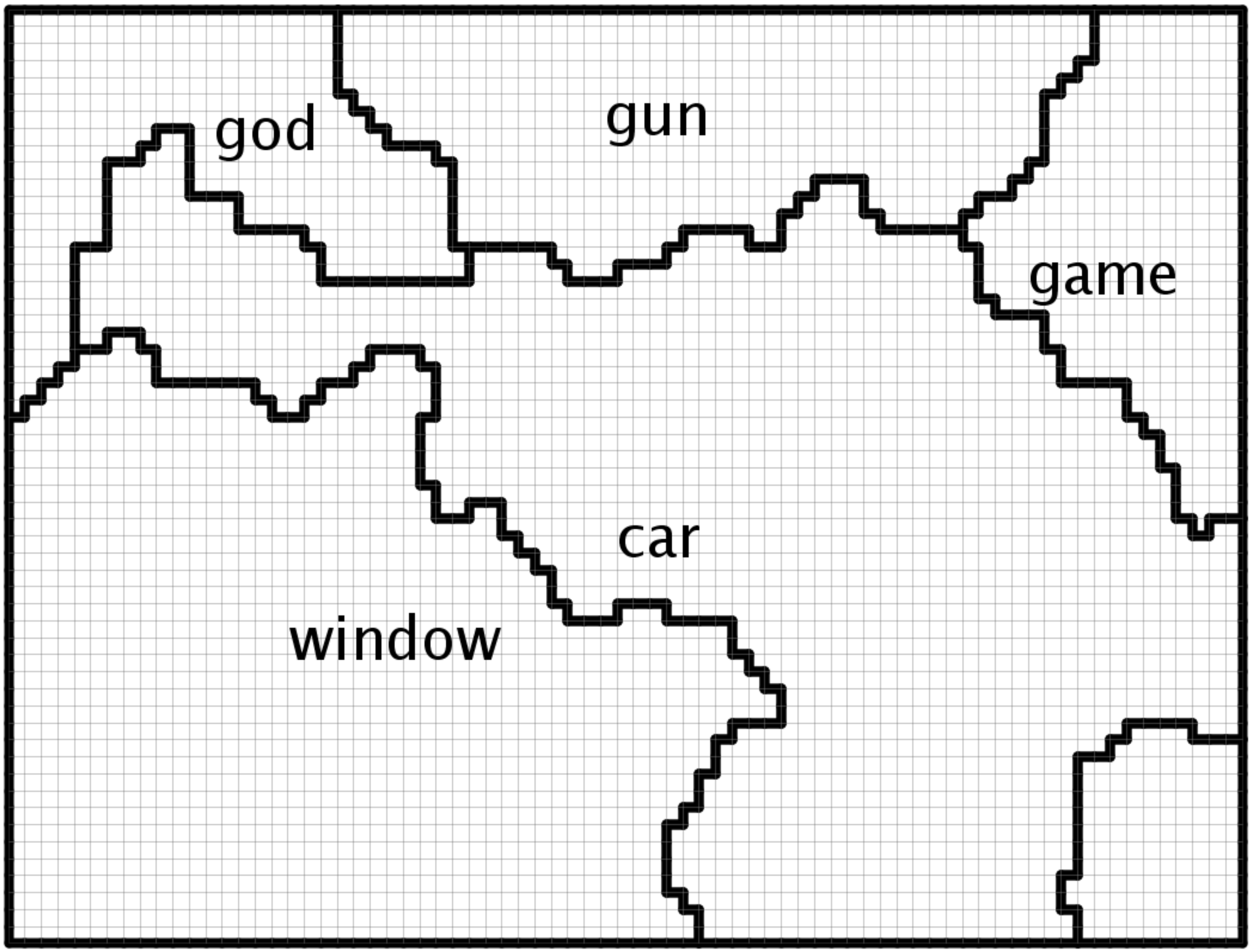
20 Newsgroups: Ward+Labels



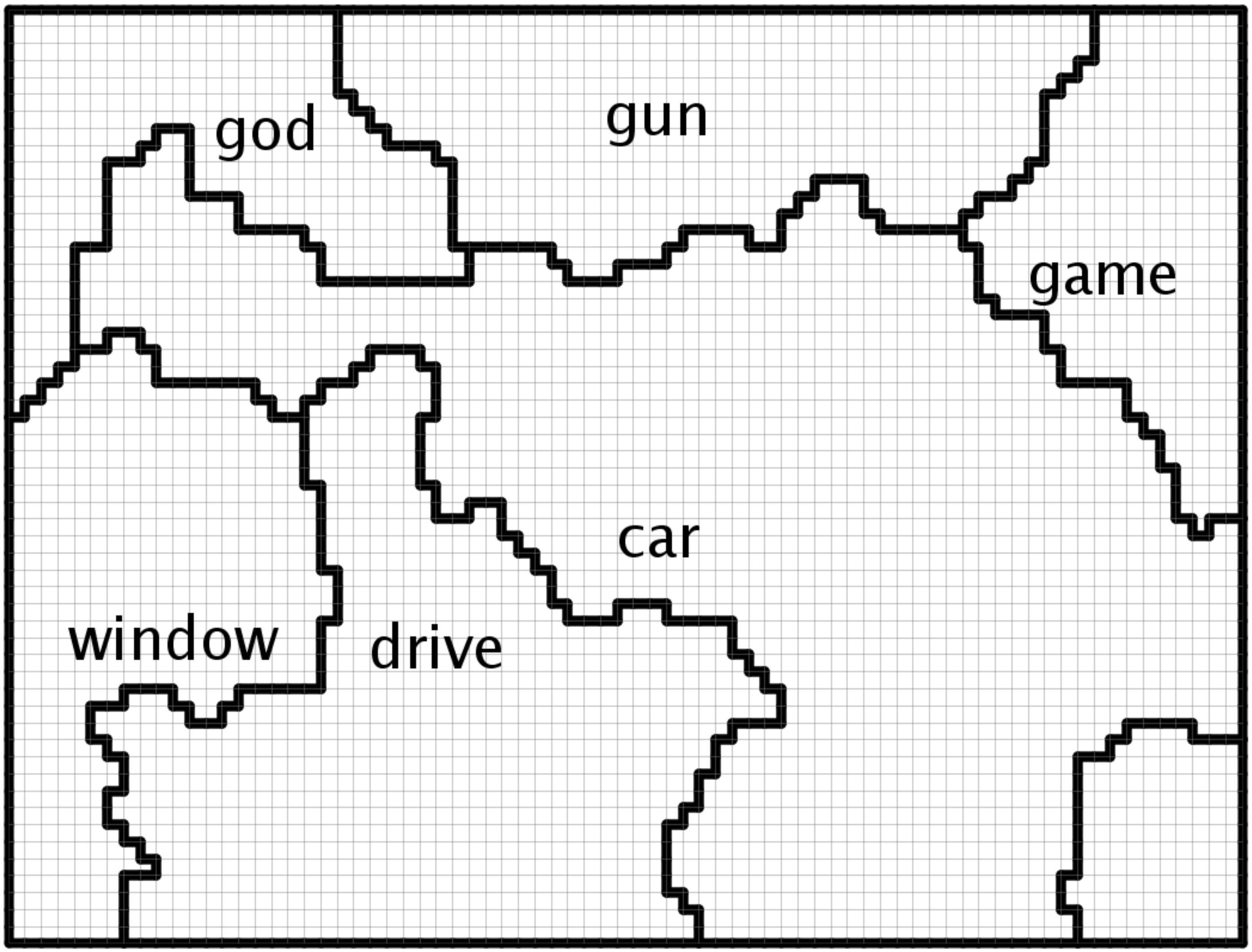
20 Newsgroups: Ward+Labels



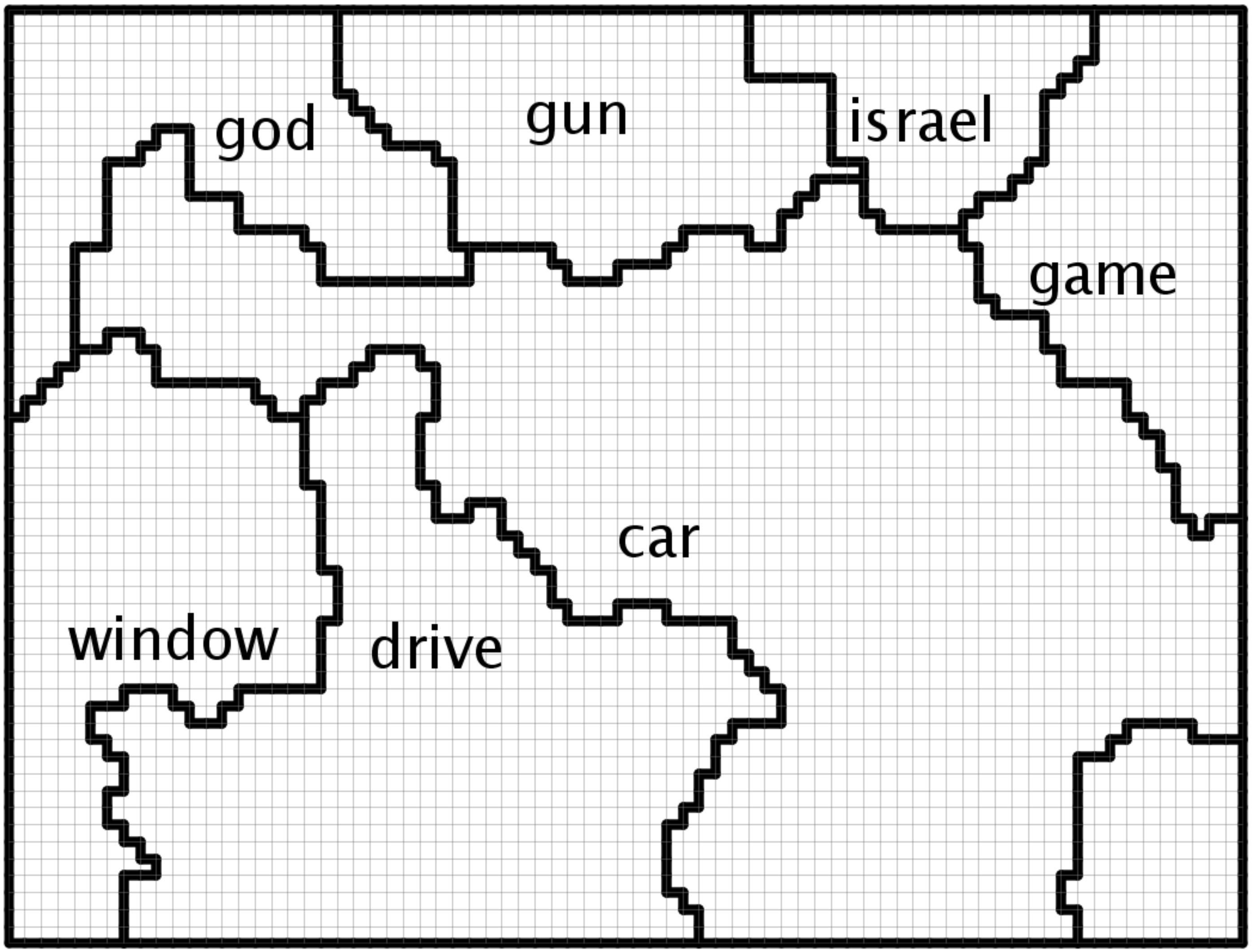
20 Newsgroups: Ward+Labels



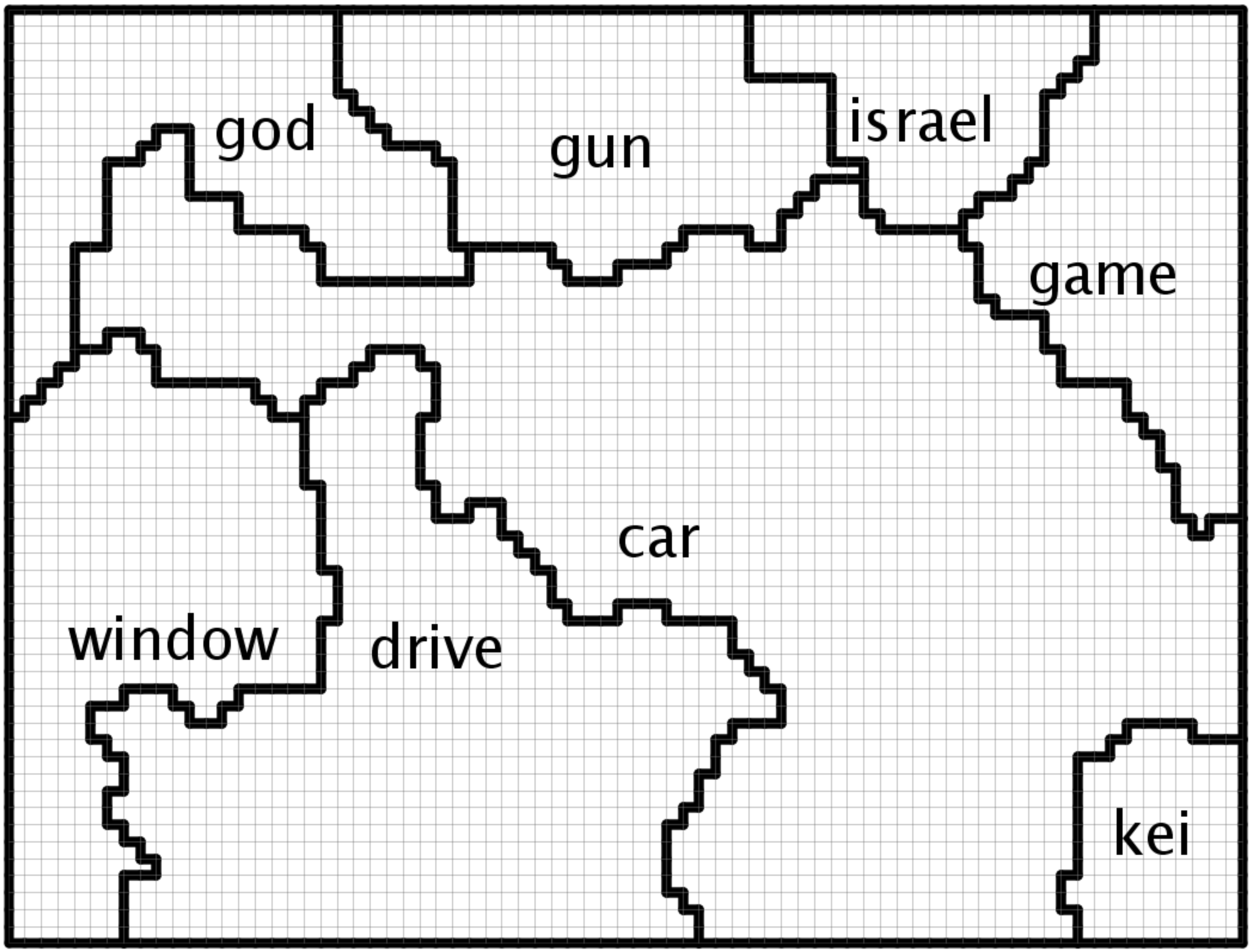
20 Newsgroups: Ward+Labels



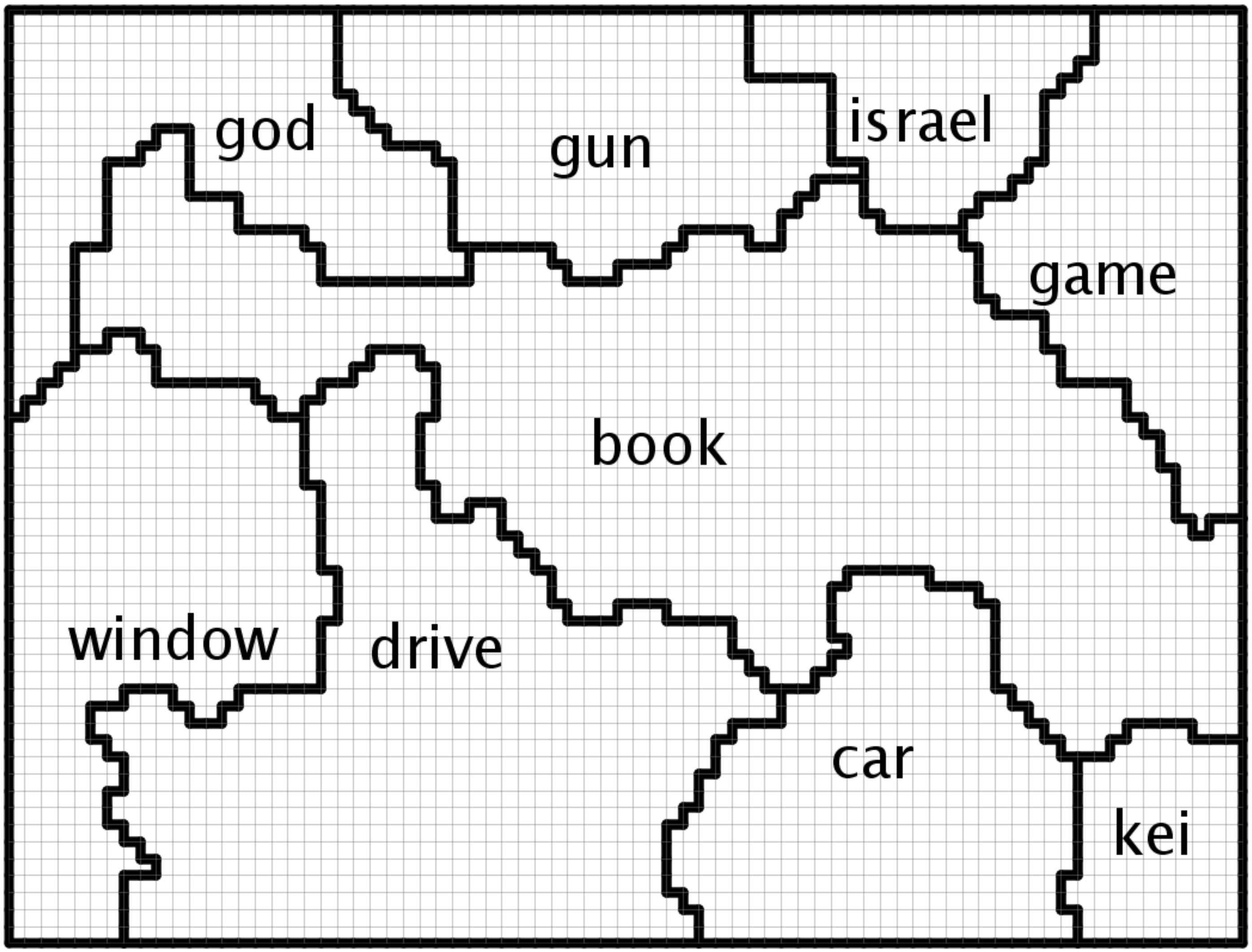
20 Newsgroups: Ward+Labels



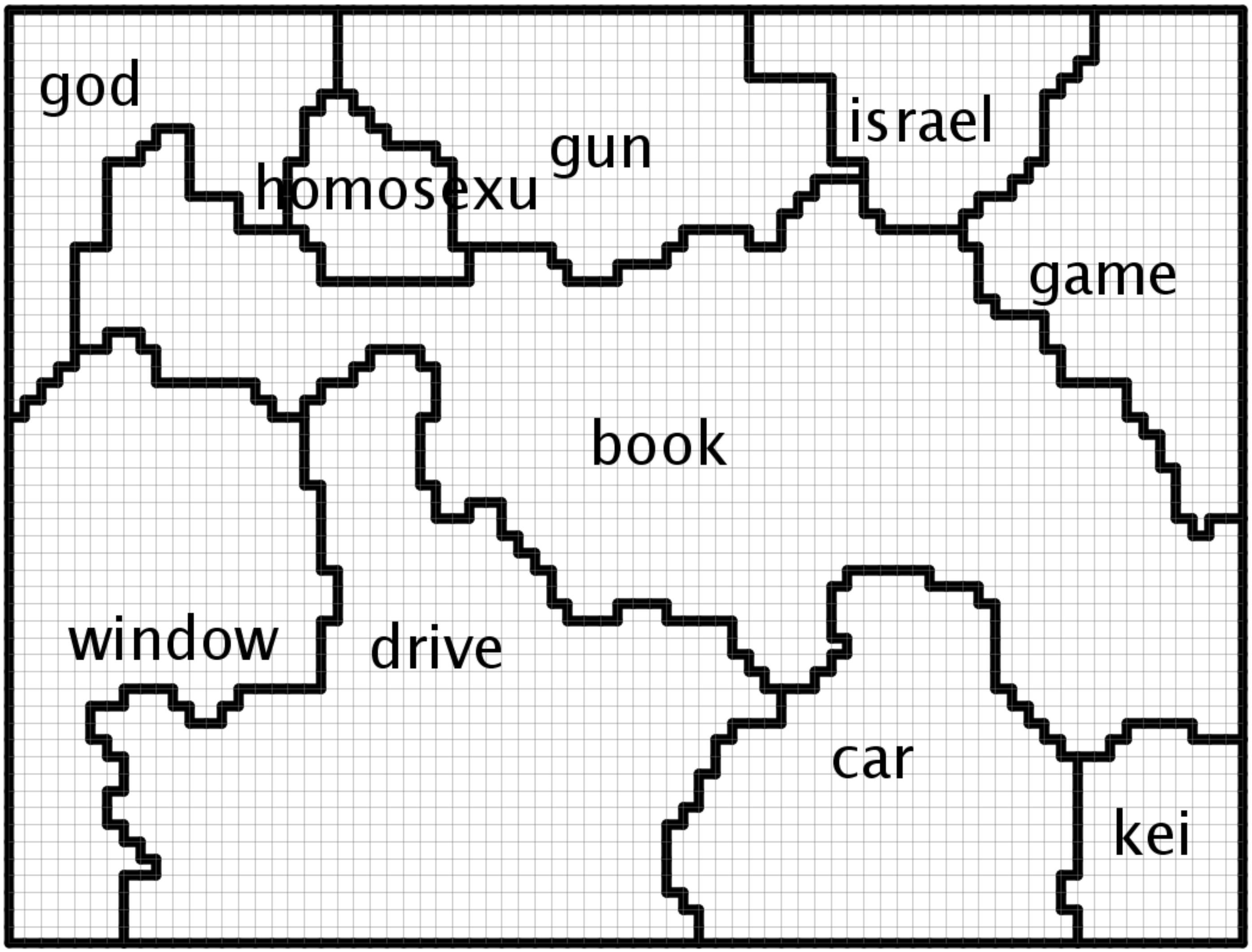
20 Newsgroups: Ward+Labels



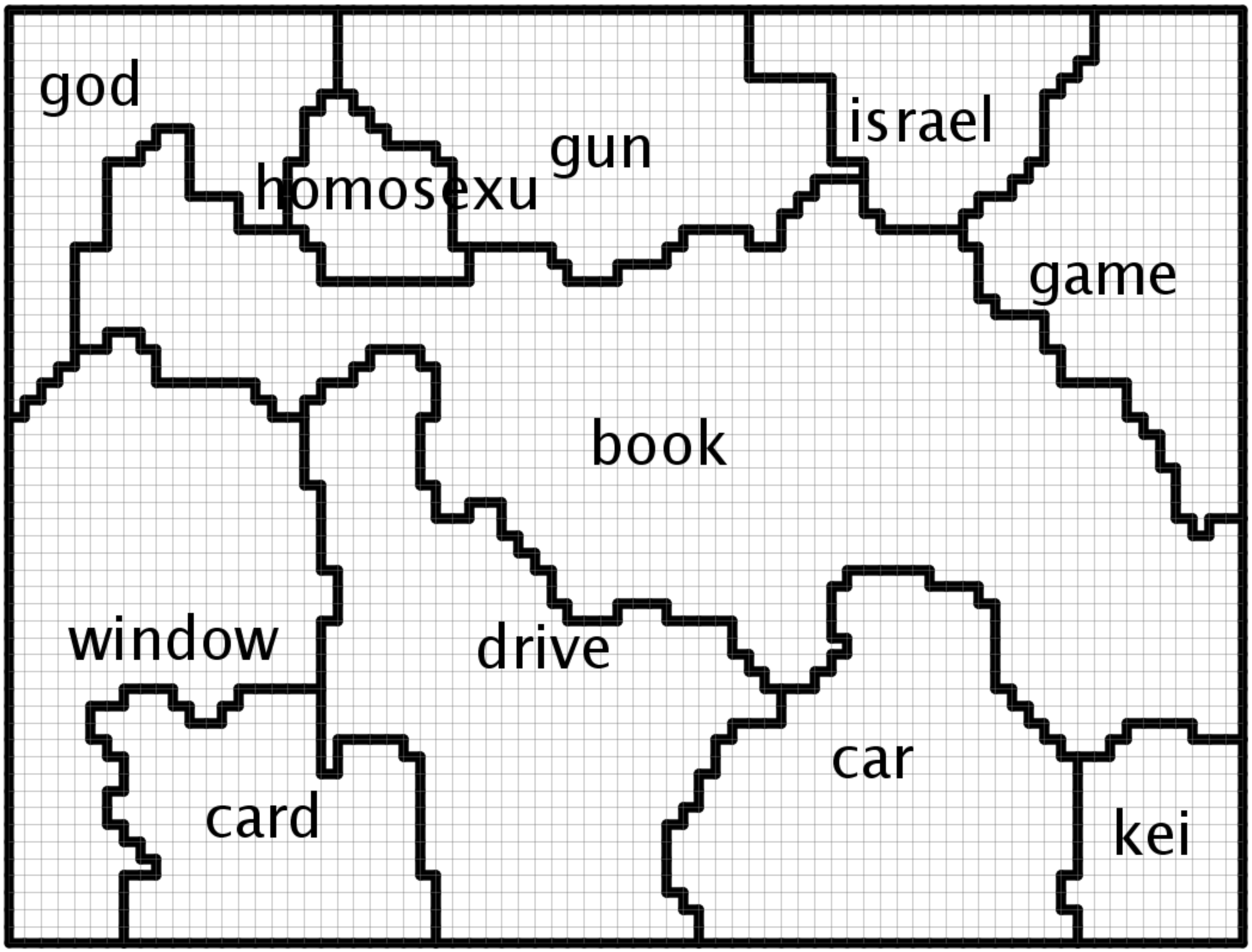
20 Newsgroups: Ward+Labels



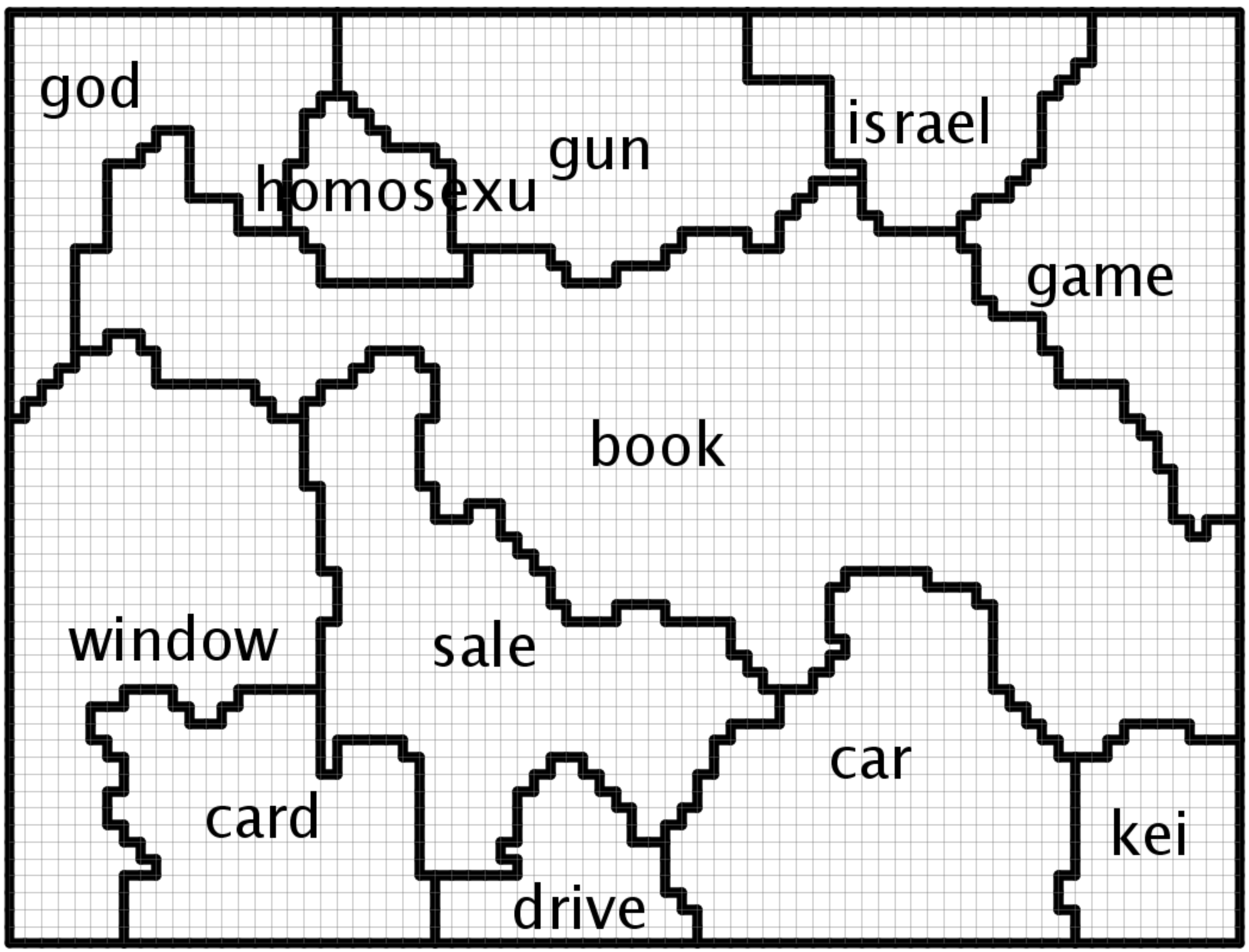
20 Newsgroups: Ward+Labels



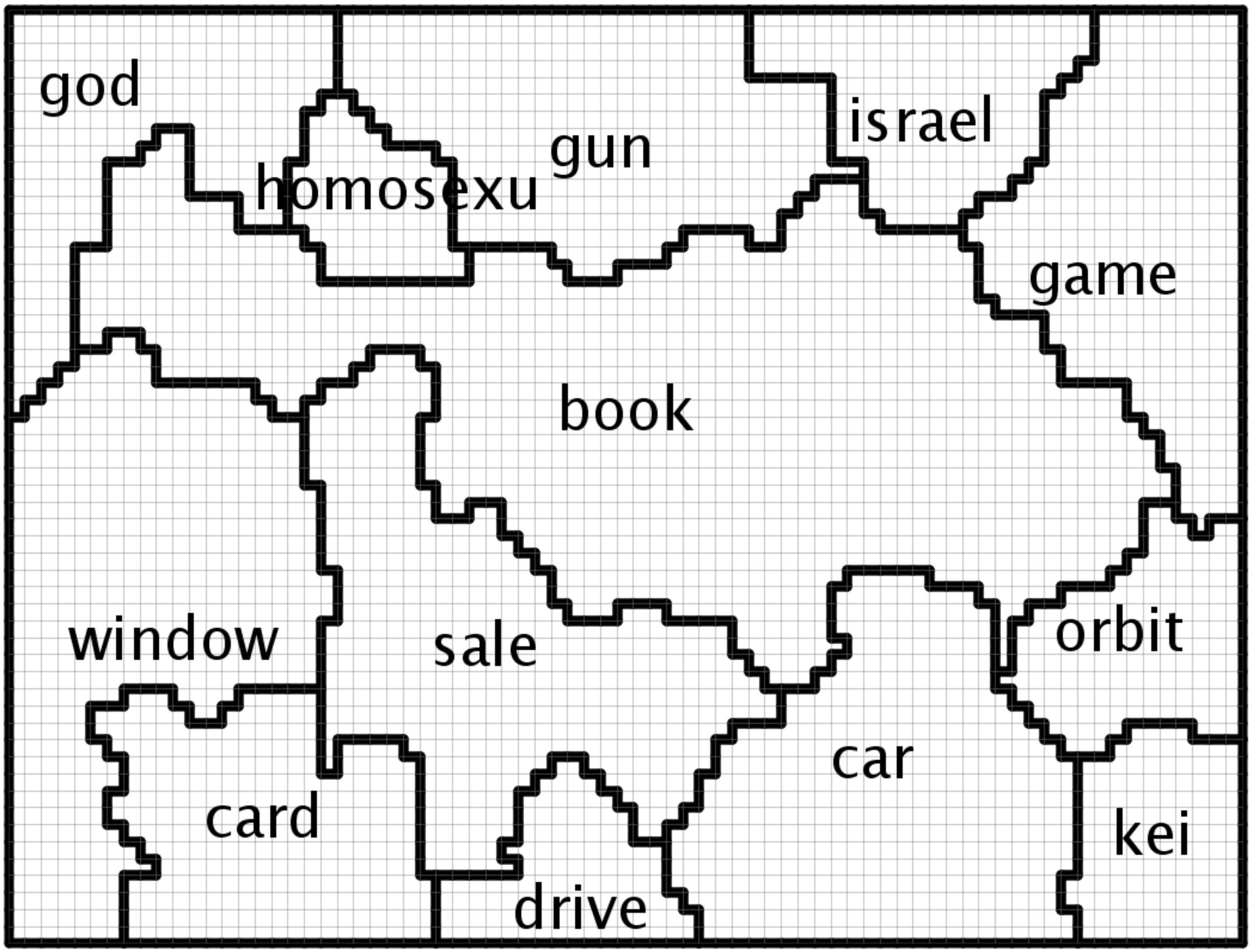
20 Newsgroups: Ward+Labels



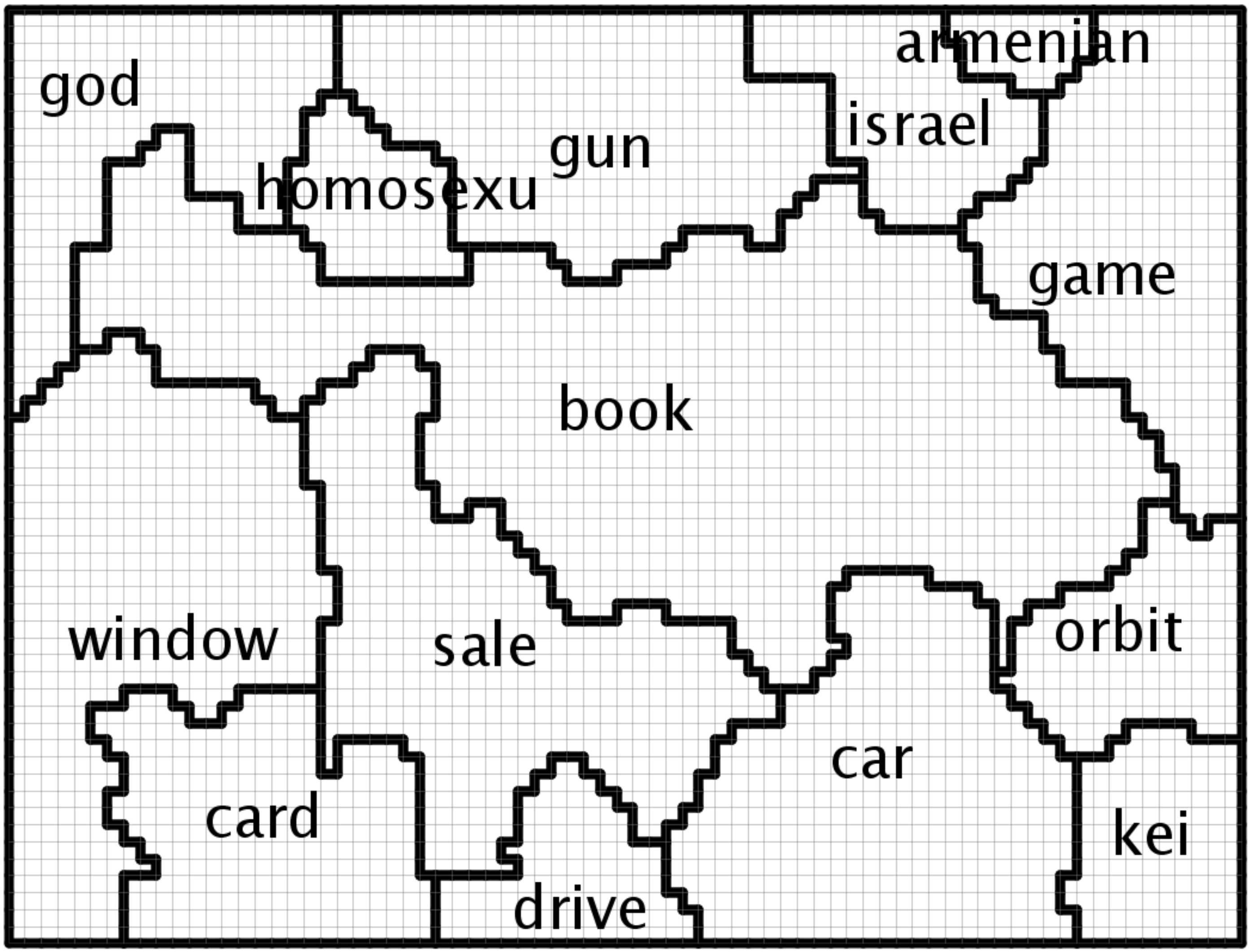
20 Newsgroups: Ward+Labels



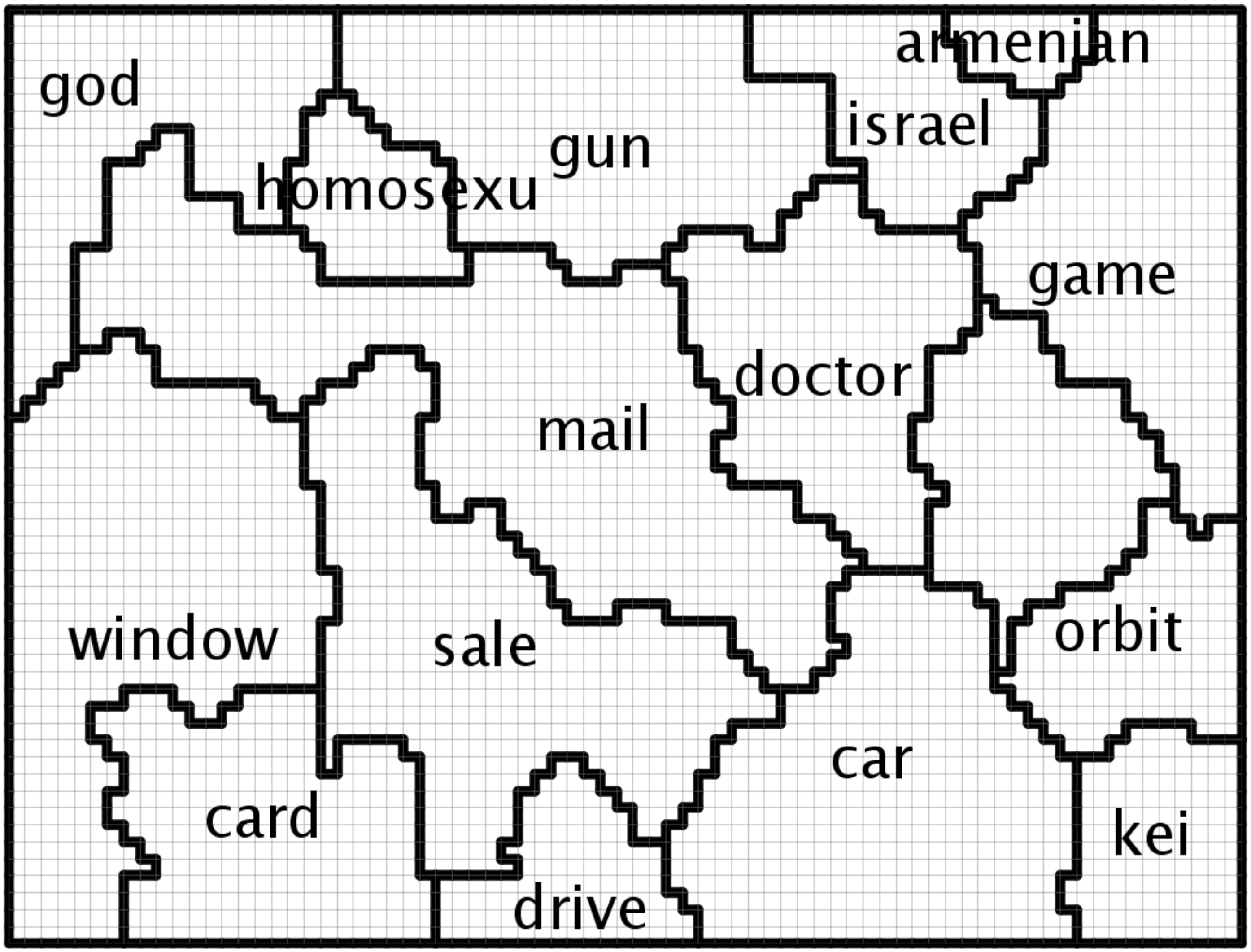
20 Newsgroups: Ward+Labels



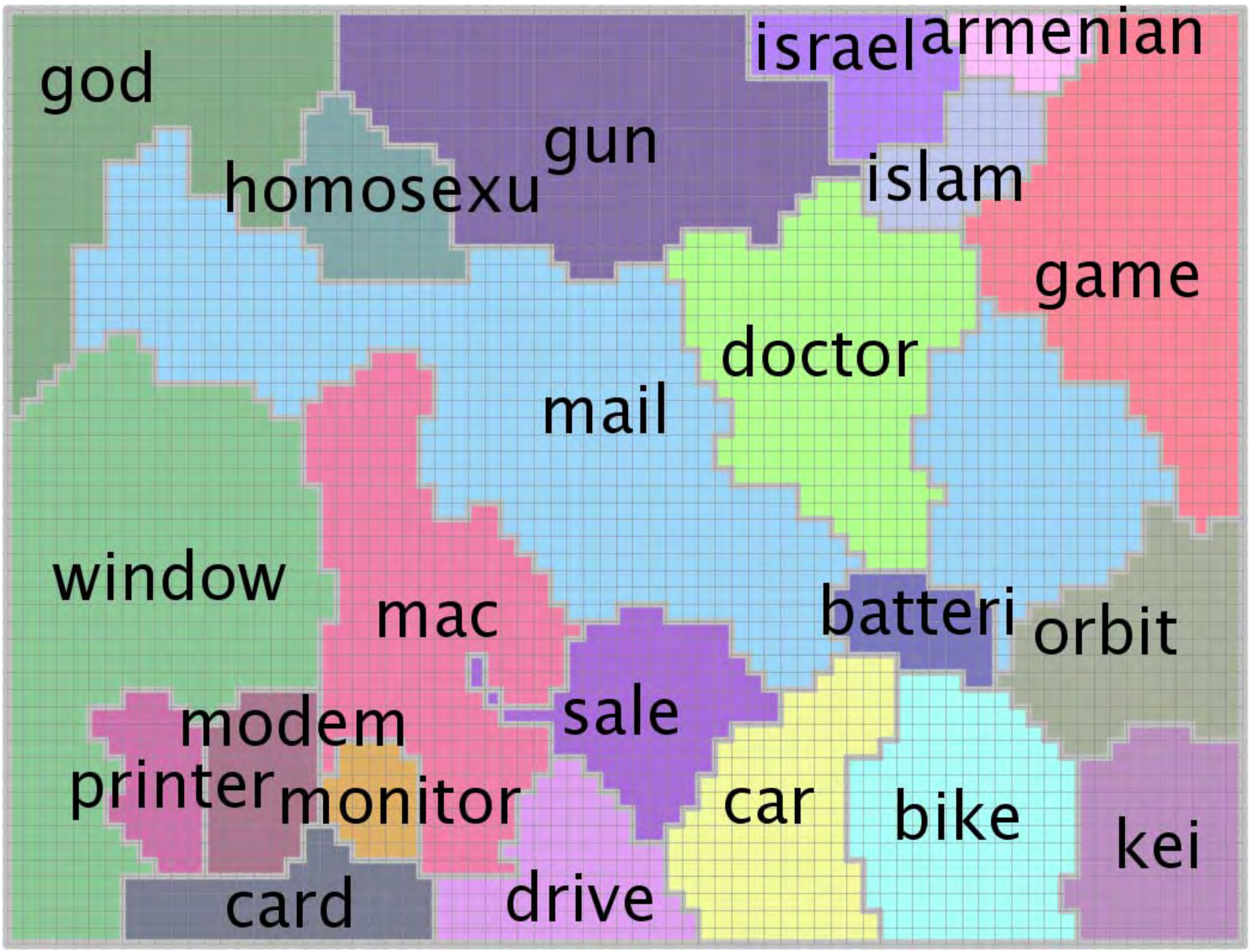
20 Newsgroups: Ward+Labels



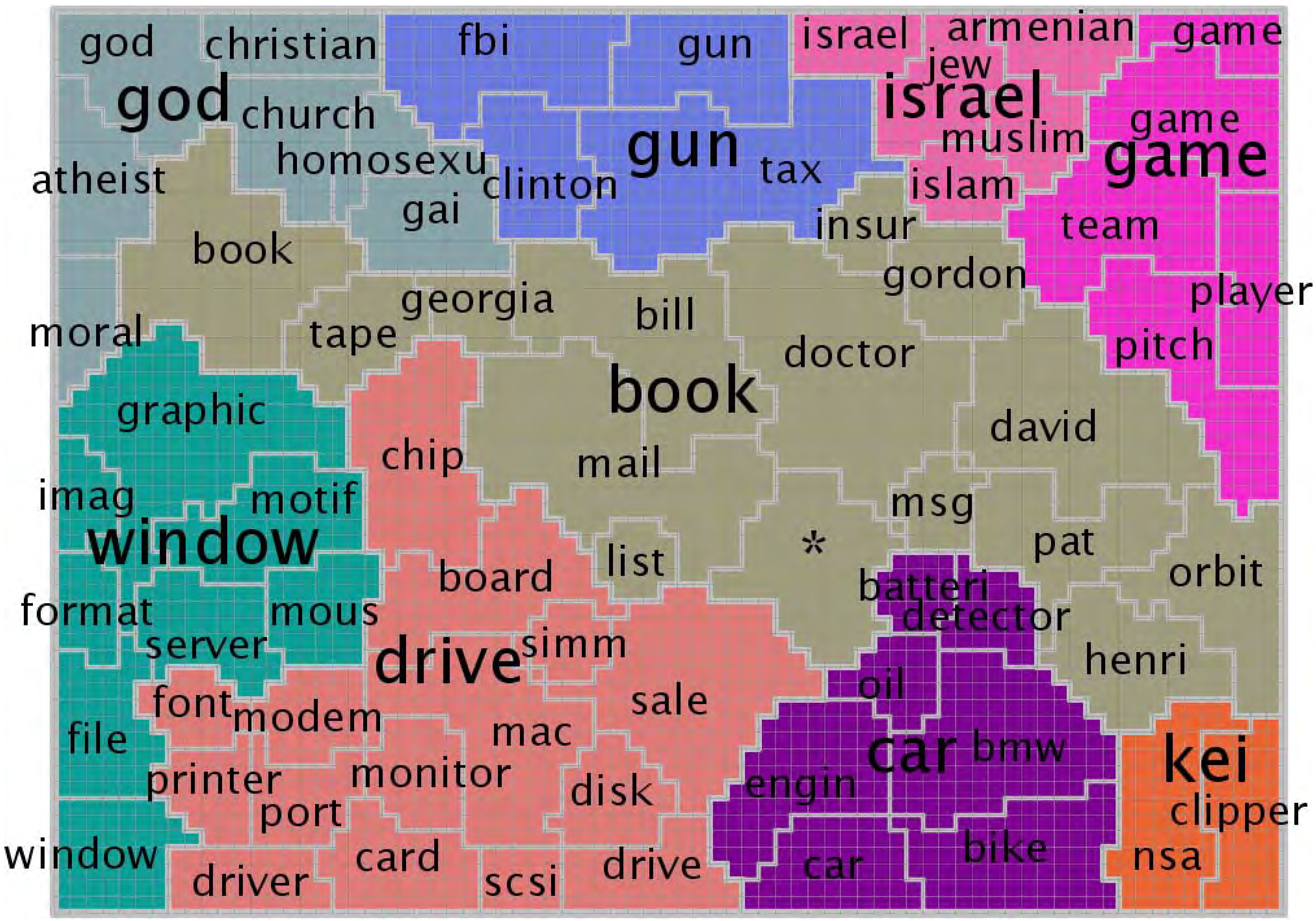
20 Newsgroups: Ward+Labels



20 Newsgroups: Ward+Labels

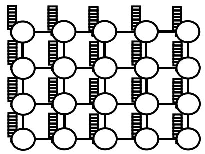


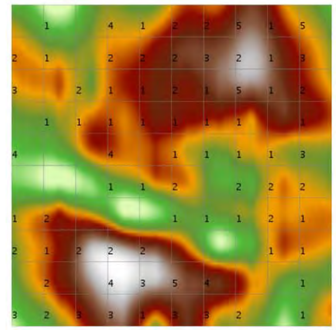
20 Newsgroups: Ward+Labels

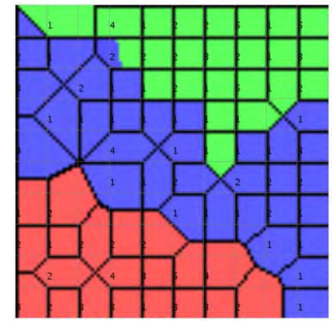


Outline

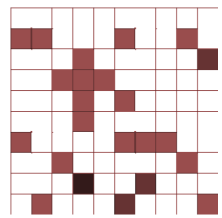
- SOM Basics

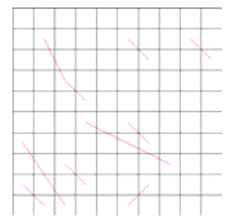




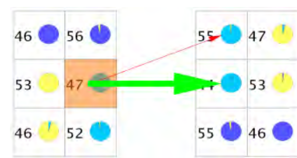


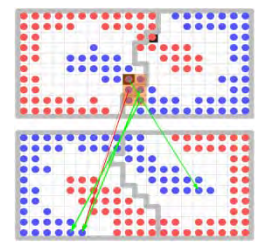
- Visualizations



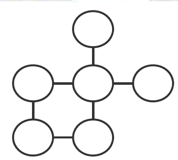


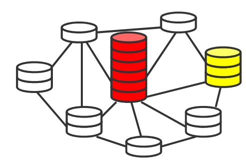
- SOM Comparison



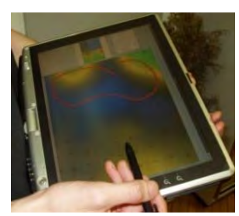


- Related Architectures and Methods





- Applications



SOM Comparison

- SOM not deterministic
  - Initialization
  - Random order of vector presentation
  - Equal / different map sizes
- How to compare 2 SOMs?
  - Quality measures
  - Comparing visualizations
- Process for comparing/aligning 2 SOMs
  - understanding relative shifts of clusters
  - stability of mappings

SOM Comparison

- Test effect of Som training parameters
  - Map size & aspect ration, learning rate, neighbourhood radius, number of iterations, initialization of SOM, random seed value, vector scaling, etc.
- Find topology violations and understand stability of mapping
- 4 approaches:
  - data / cluster comparison
  - data / cluster shifts
- Rudolf Mayer, Robert Neumayer, Doris Baum, and Andreas Rauber.

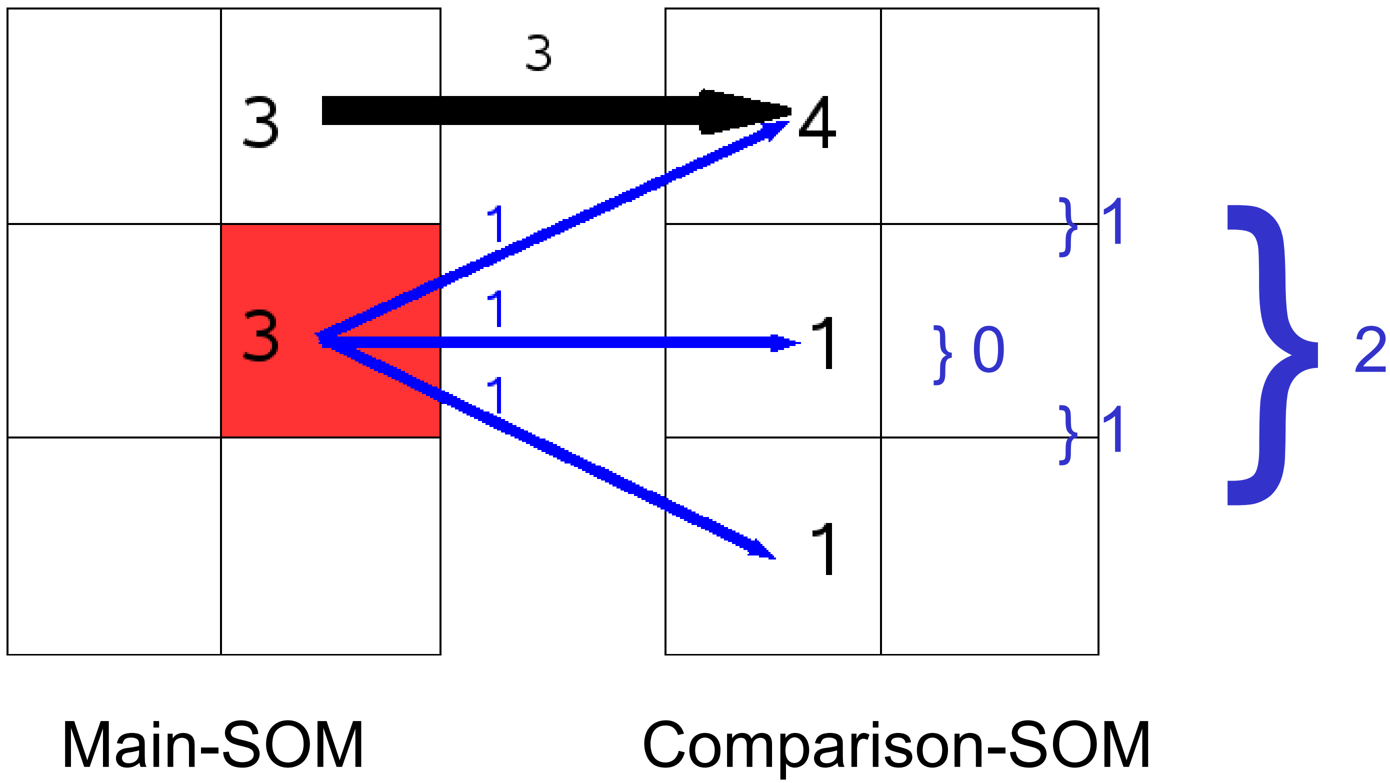
  Analytic Comparison of Self-Organising Maps.

  In *Proceedings of the* *7th International Workshop on Self-Organizing Maps (WSOM'09)*, St. Augustine, FL, USA, June 8 - 10 2009. LNCS 5629, pp 182-190, Springer.

SOM Comparison Visualization

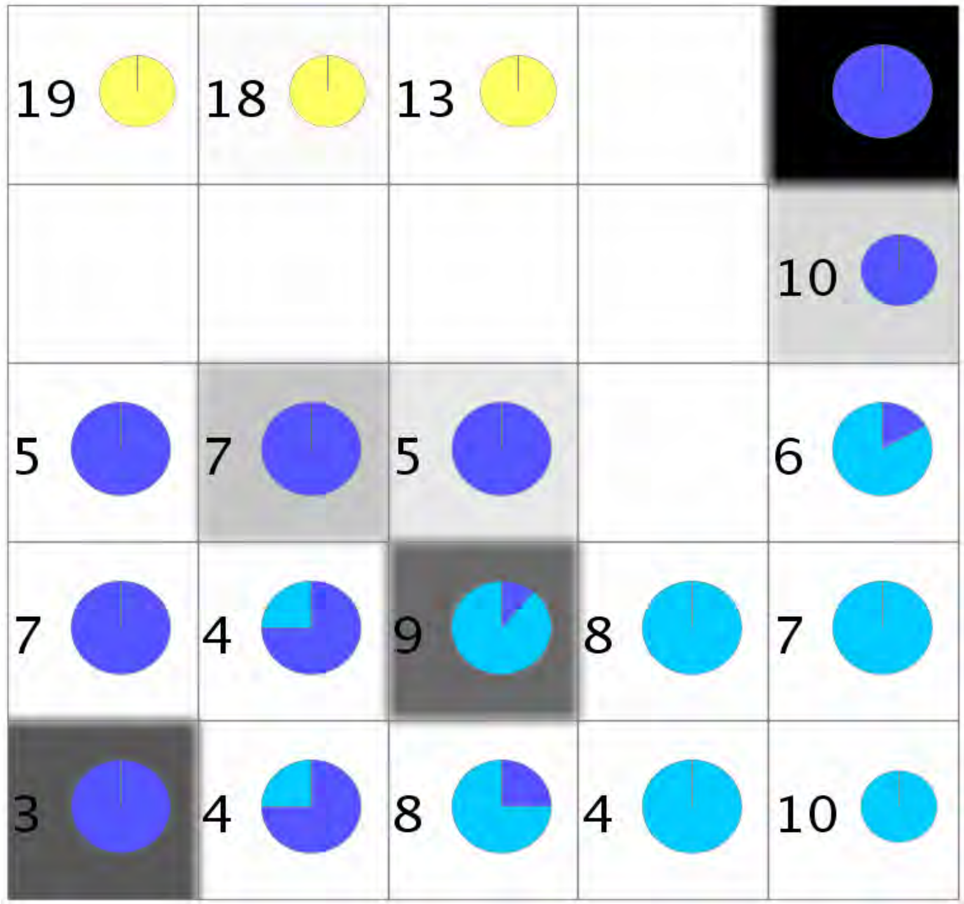
- Compares “arbitrary number” of SOMs:
  - 1 “Main-SOM”, which is presented (viusalized)
  - One or more “Comparison SOMs” over which averages are computed
- For each unit on the Main SOM:
  color unit according to the average pairwise distance of data vectors in output space on comparison SOMs

Comparison Visualization



Distances: 0 and (1+1+2)/3=4/3 = 1.3

Comparison Visualization



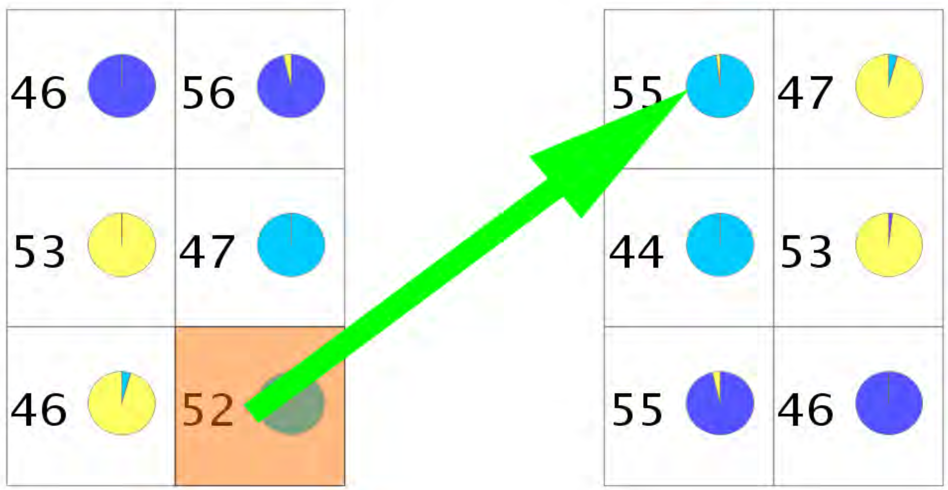
Comparison Visualization

Variations:

- Cluster distance instead of Euclidean Distance:
  - Distance between vectors is replaced by distance between clusters in which they are placed
  - Cluster distance computed via, e.g. Single Linkage (minimal distance between two closest units of clusters)
- Variance instead of mean distance
- Threshold: pairwise distances smaller than threshold are not used in average distance calculations (avoid effect of minor variations, shift to neighboring units in dense areas)

Data Shifts Visualization

- Compares two SOMs to each other
- Shows for data instances on SOM 1 where they are on SOM 2 : „Shift“
- E.g. after training determine where data has moved to



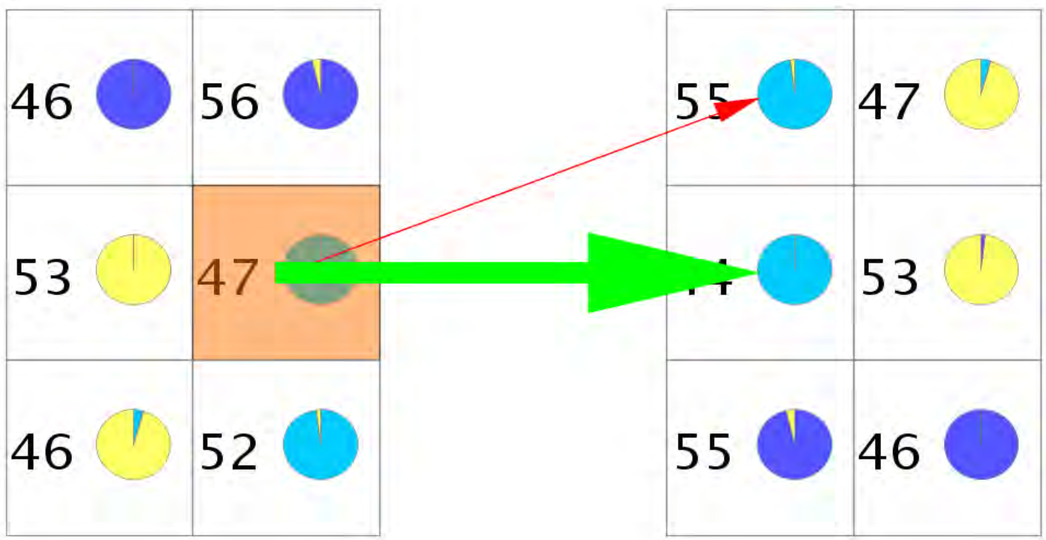
SOM 1　　　　SOM 2

Data Shifts Visualization

- When comparing the position of an instance vector also compare the position of its neighbors :
  - Many neighboring data points from SOM 1 are also neighboring on SOM 2: stable shift
  - If a vector from SOM1 ends up completely dislocated from its former neighbors on SOM 2 : outlier shift
- Seize of “neighbourhood” and number of instances to be considered for stable/outlier are parameters (depending on e.g. different sizes of SOM1 and SOM2)
- Allows statements on the stability of the clustering

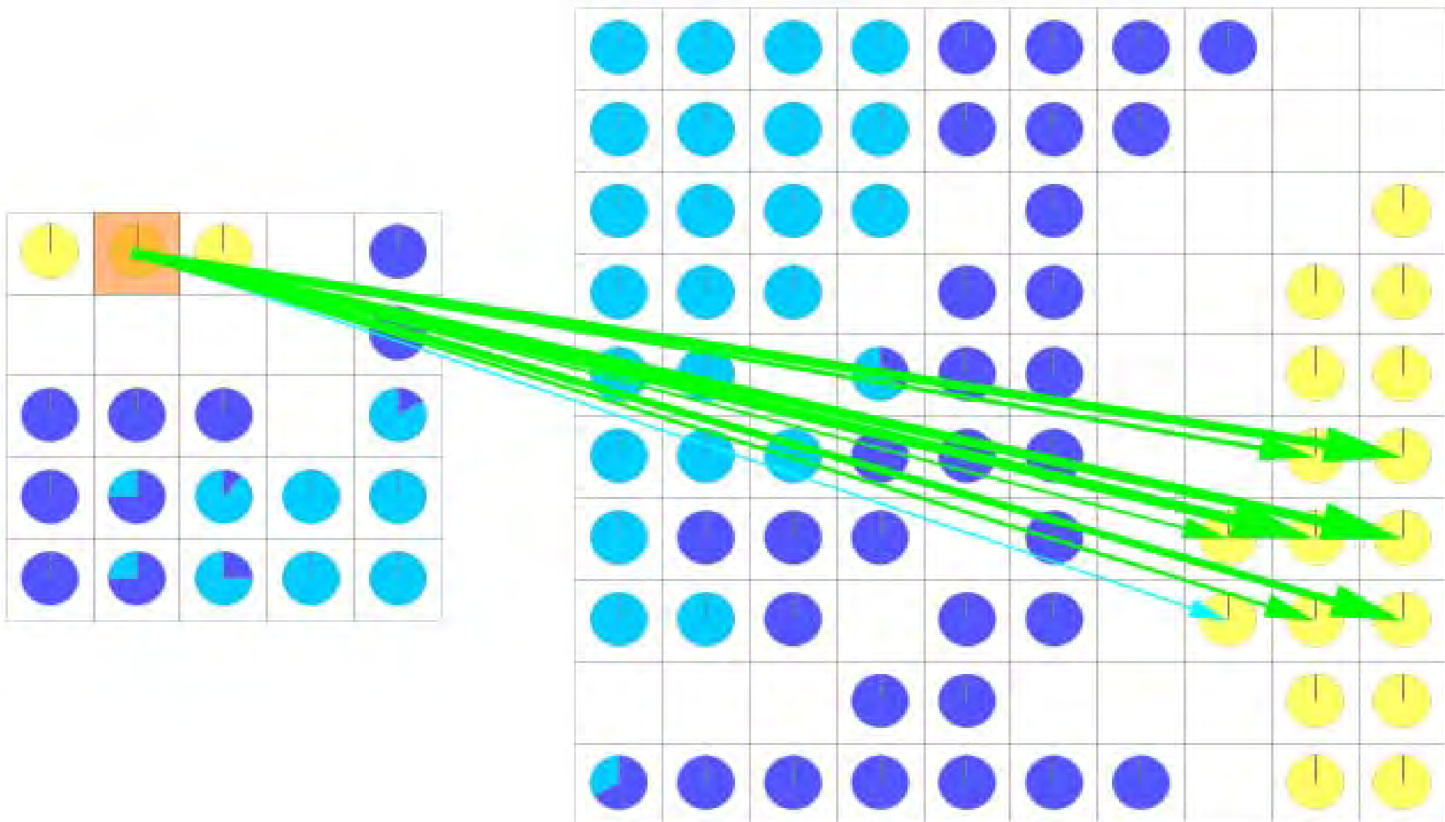
Data Shifts Visualization

- Green: stable; Red: outlier; Neighborhood size: 0
- Line thickness proportional to number of neighbors that stay the same / are different



Data Shifts Visualization

- Data Shifts: small SOM compared to large SOM

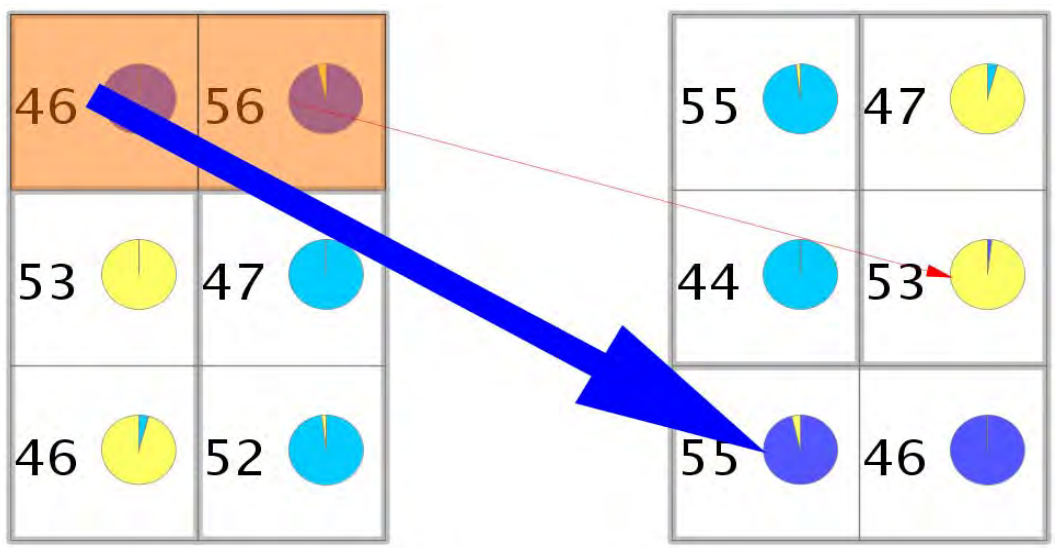


Cluster Shifts Visualization

- Is an instance on SOM 2 in the same cluster as on SOM 1?
- Clustering of both SOMs into specific number of clusters Clustern
- Cluster on SOM 1 are matched to clusters on SOM 2 (based on majority vote in data instances)
- Data instances that end up in the “same” clusters: stable shift; otherwise: outlier

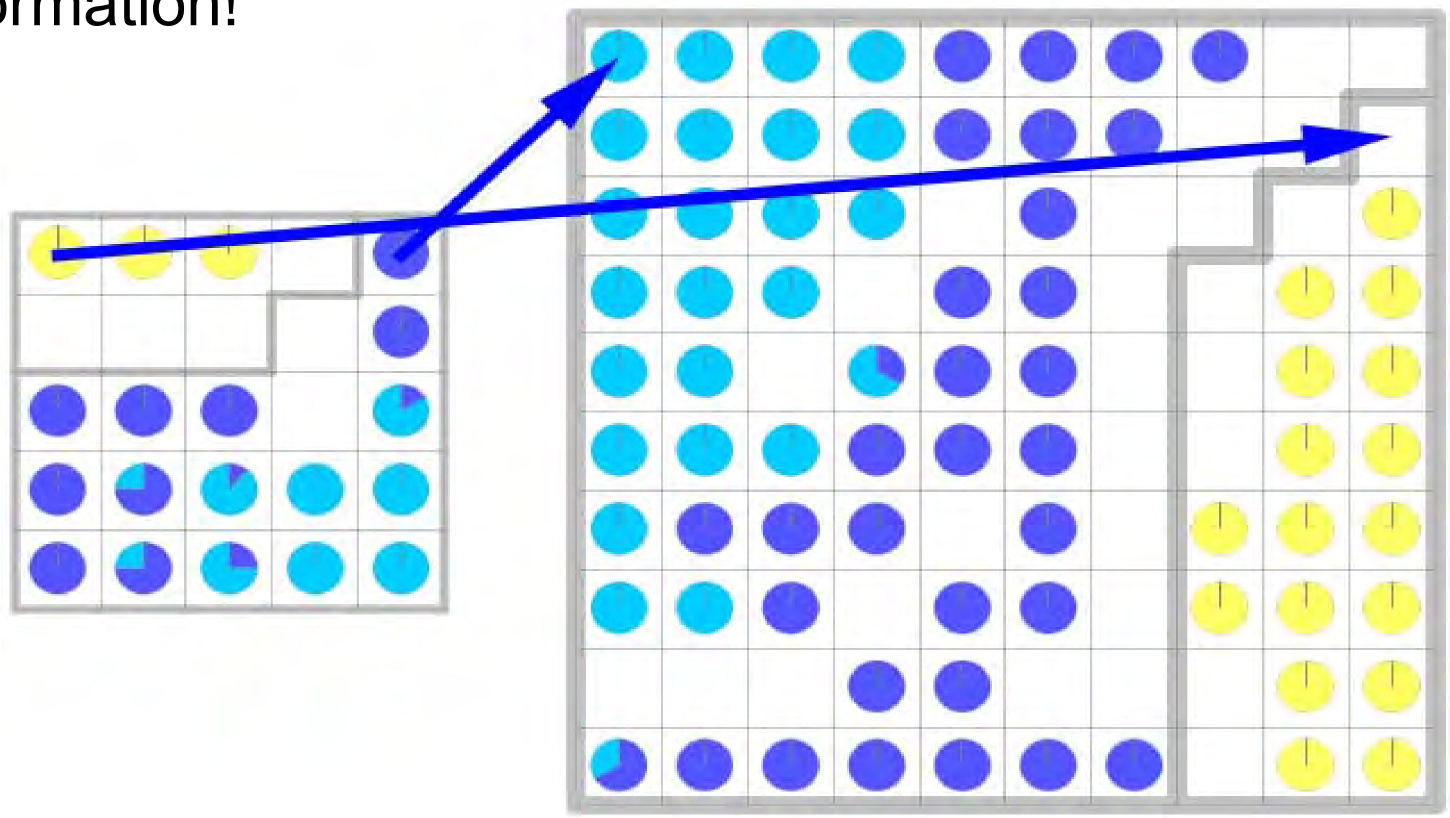
Cluster Shifts Visualization

- Blue: matched Clusters
- Line thickness proportional to match between clusters



Cluster Shifts Visualization

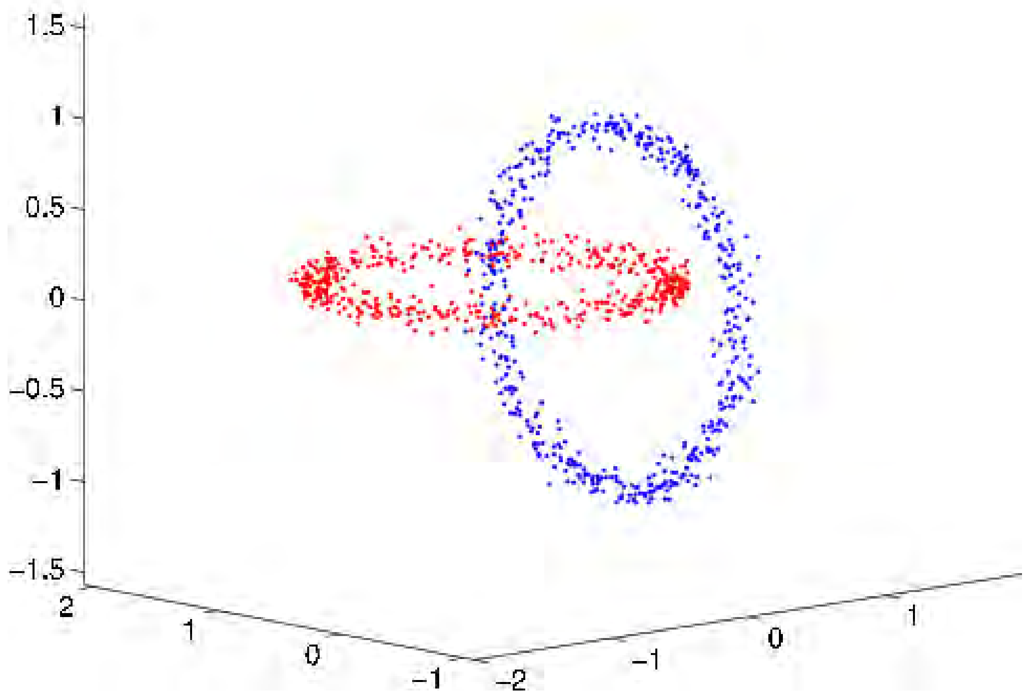
- Cluster Shifts: small SOM to large SOM
- Determine, where the clusters are locate don the new map
- Obviously only necessary when you don't have class information!



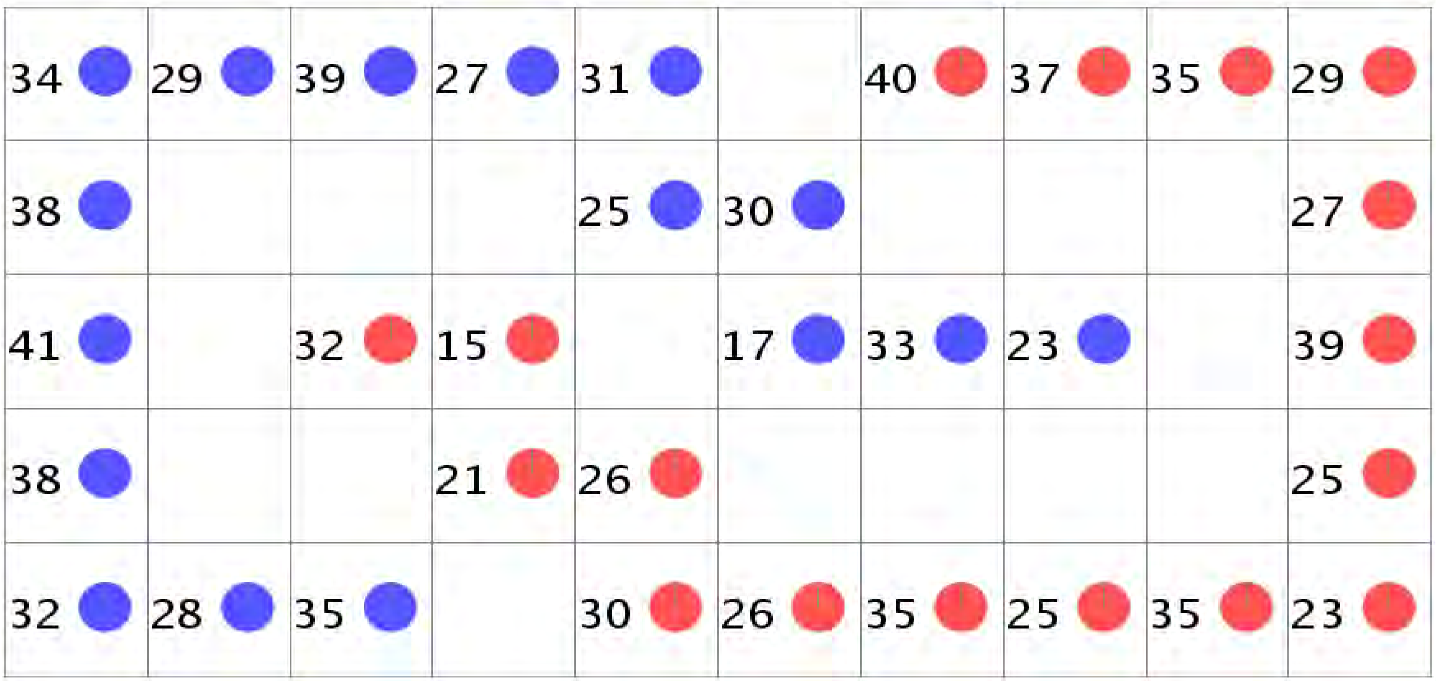
SOM Comparison

Intertwined rings

- 3-dimensional data
- Projection on 2D-grid
- Topology violation!

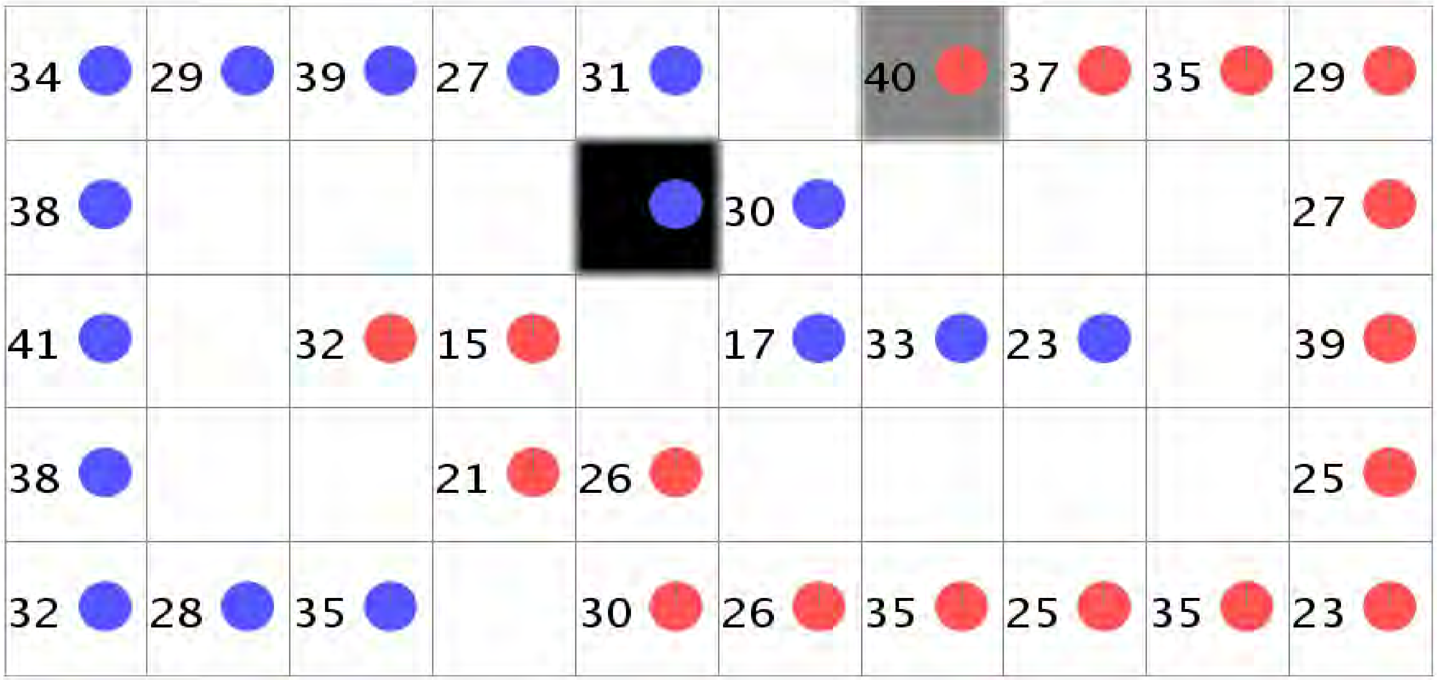


SOM Comparison



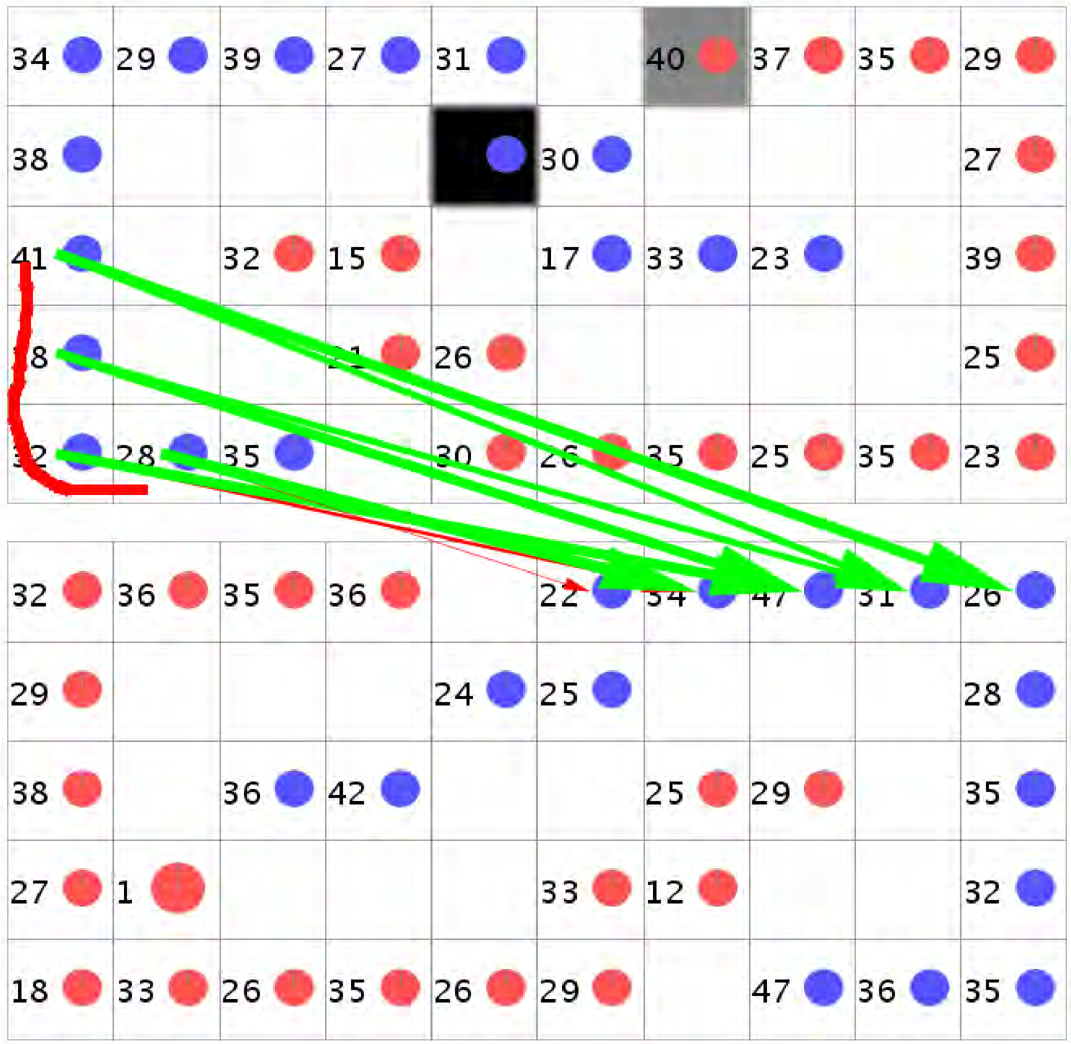
Any SOM breaks rings somewhere

Comparison Visualization

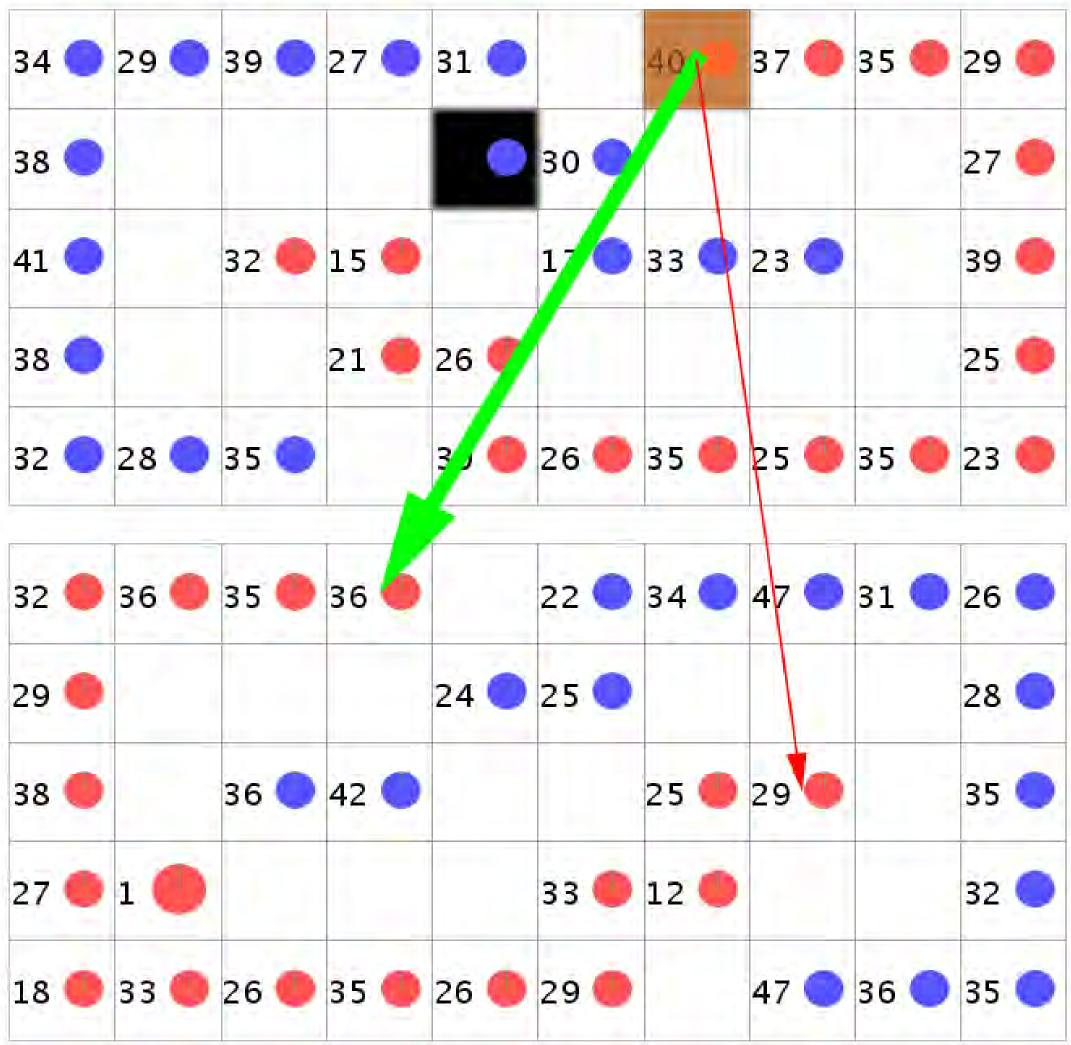


Threshold of 3: only pairwise distances > 3 counted → larger distances have strong impact, minor ones none

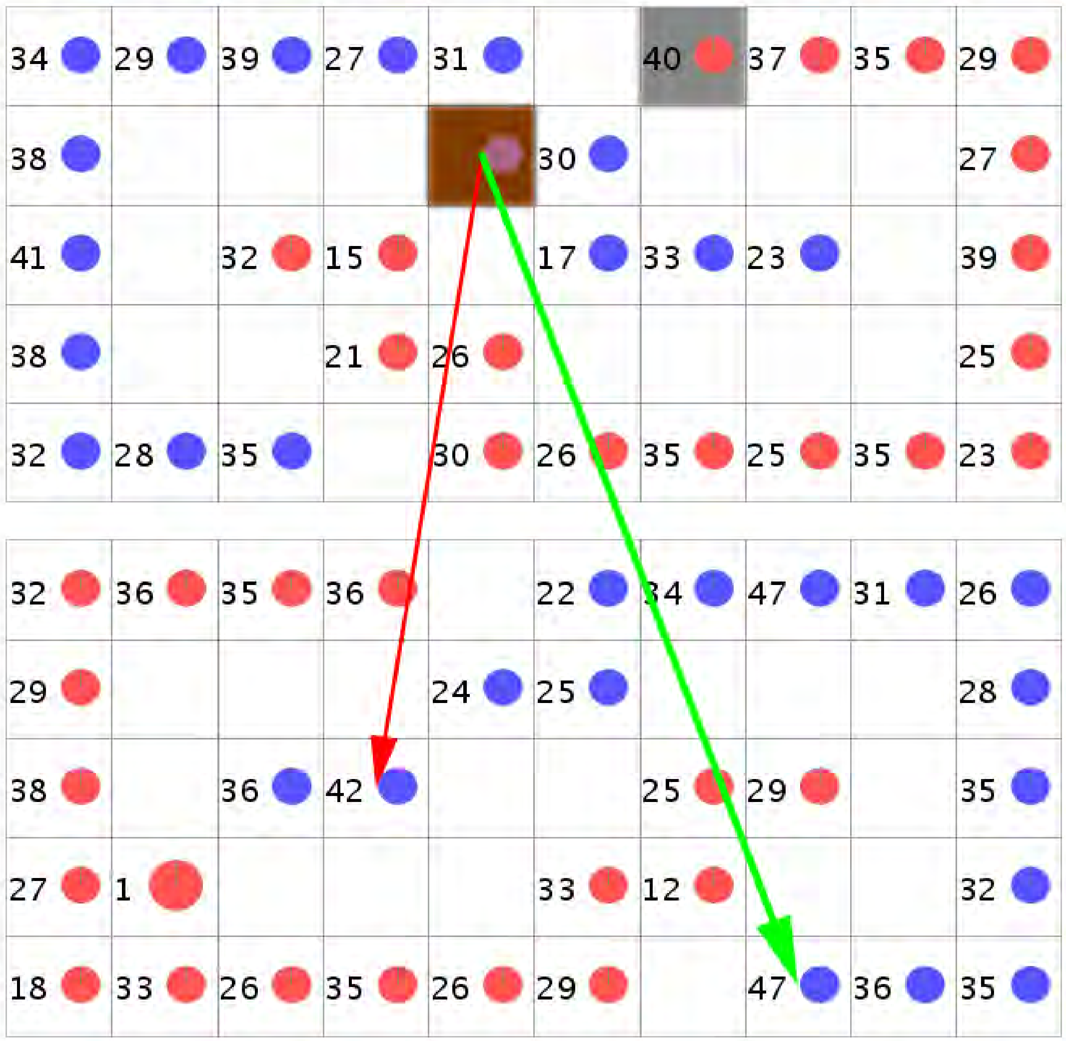
Data Shifts Visualization



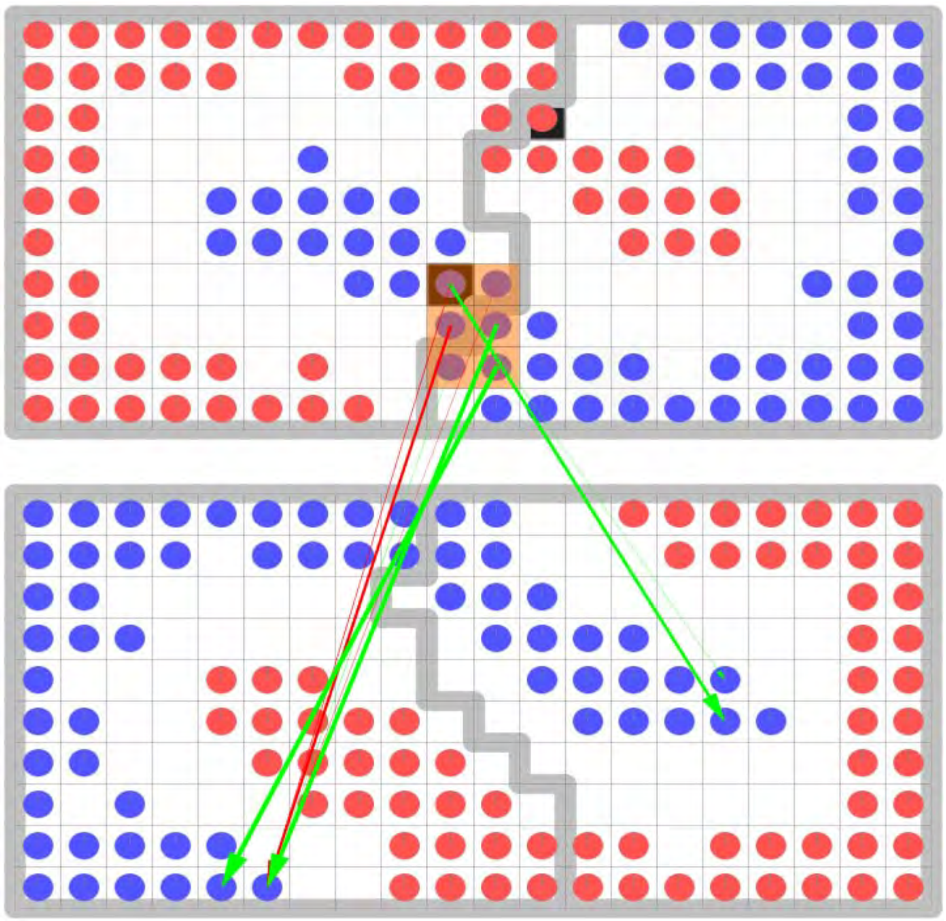
Data Shifts Visualization



Data Shifts Visualization

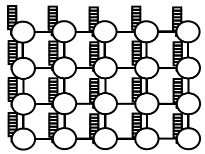


Cluster Shifts Visualization

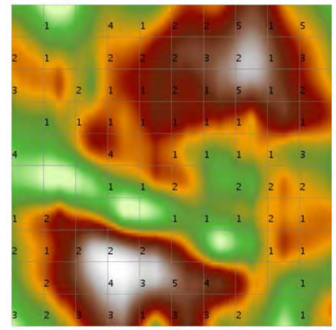


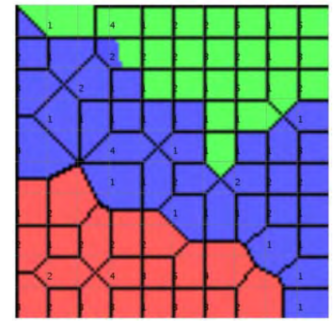
Outline

- SOM Basics

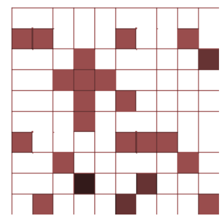


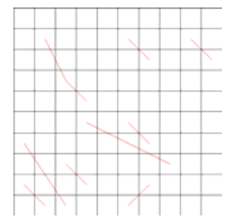
- Visualizations



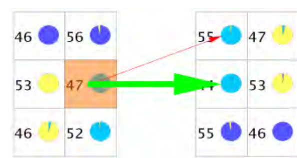


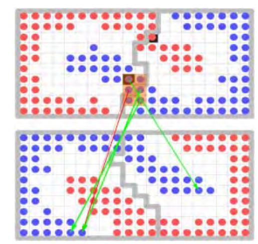
- SOM Quality Measures



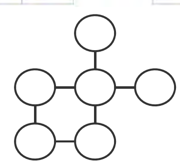


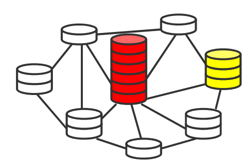
- SOM Comparison



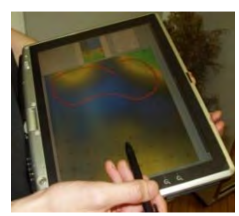


- Related Architectures and Methods





- Applications



Applications

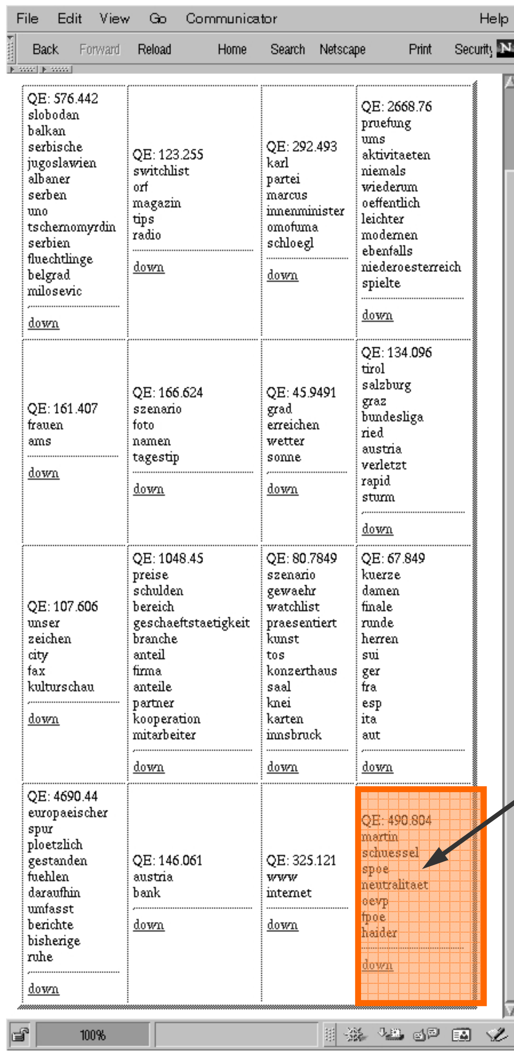
- Prozess-Monitoring
- Explorative Datenanalyse
- Text-Mining: SOMLib
- Musikanalyse: SOMeJB

Growing Hierarchical SOM

- **Dokumentensammlung**
  - Nachrichtenartikel aus dem “Standard”
    (2. Quartal 1999)
  - 11.627 Artikel
  - 3.799 Worte dienen zur Beschreibung der Artikel

Growing Hierarchical SOM

- oberste Ebene



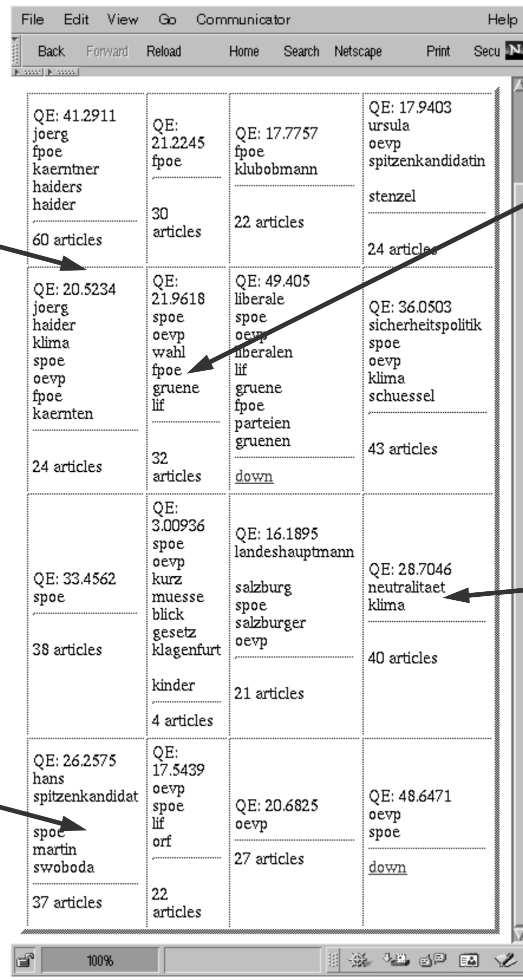
Innenpolitik

Growing Hierarchical SOM

- Karte der 2. Ebene

Freiheitliche

EU Wahlen



Neutralität

Sozial-
demokraten

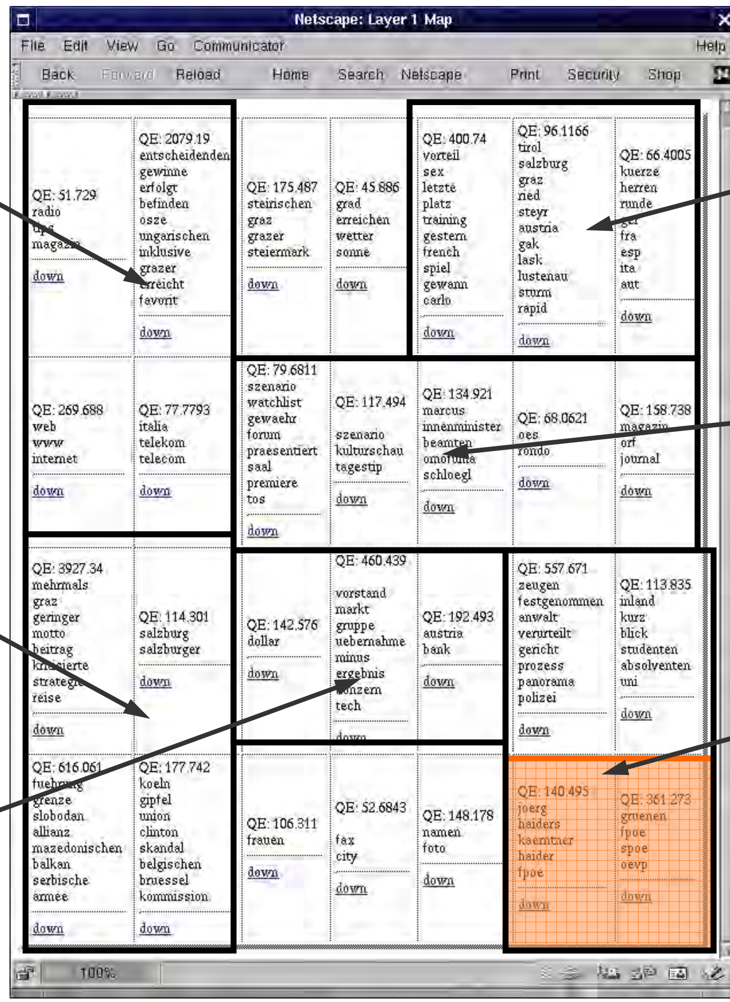
Growing Hierarchical SOM - 2

- oberste Ebene

Medien, www

internationale News

Wirtschaft



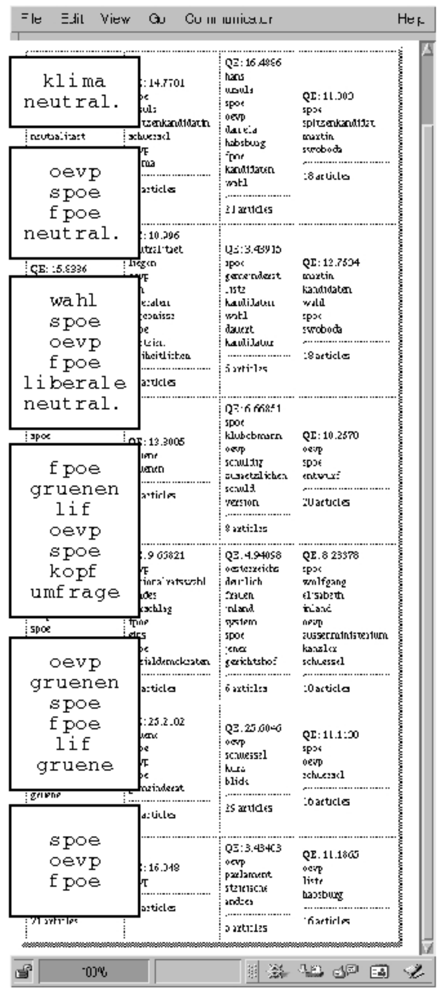
Sport

Kultur

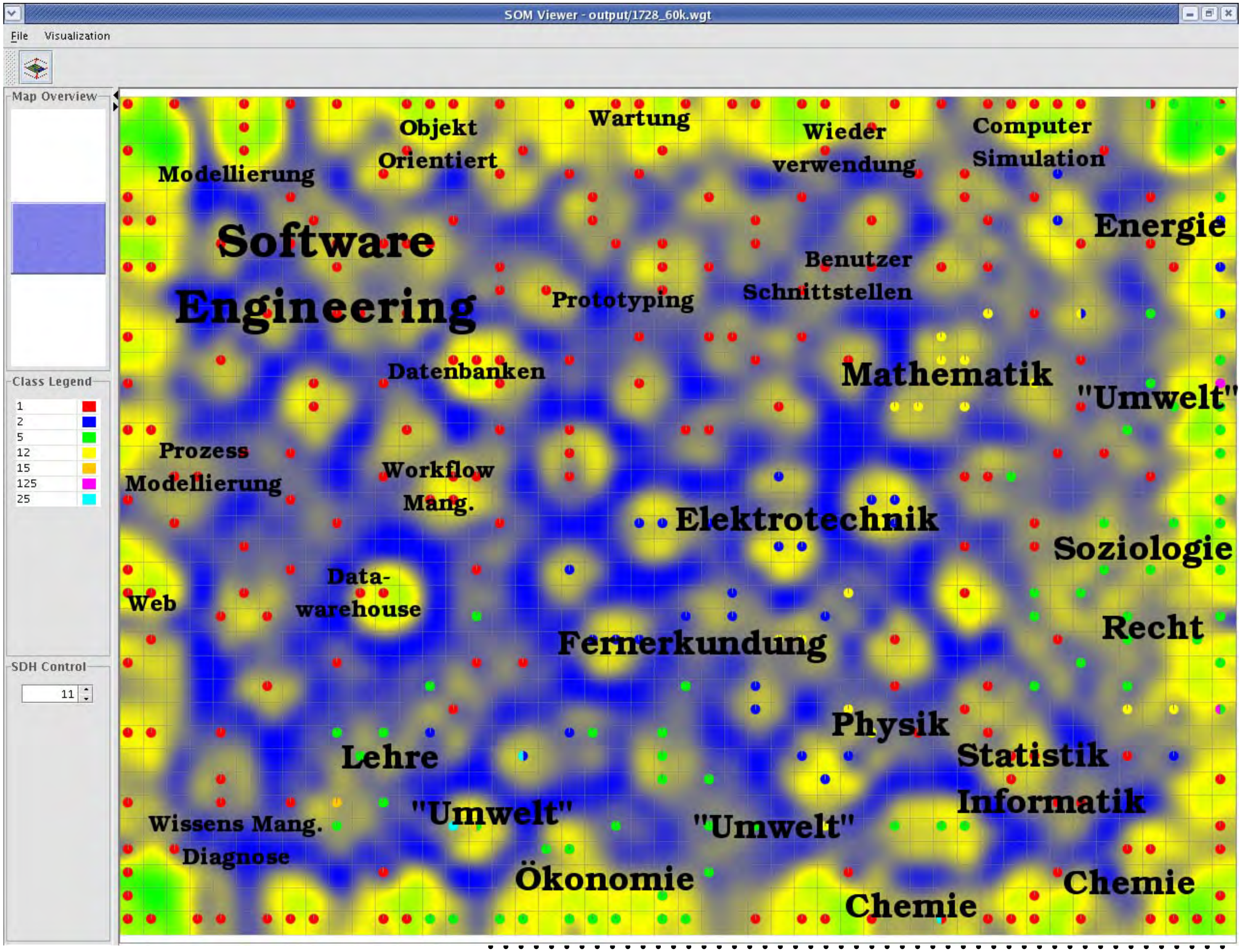
Innenpolitik

Growing Hierarchical SOM

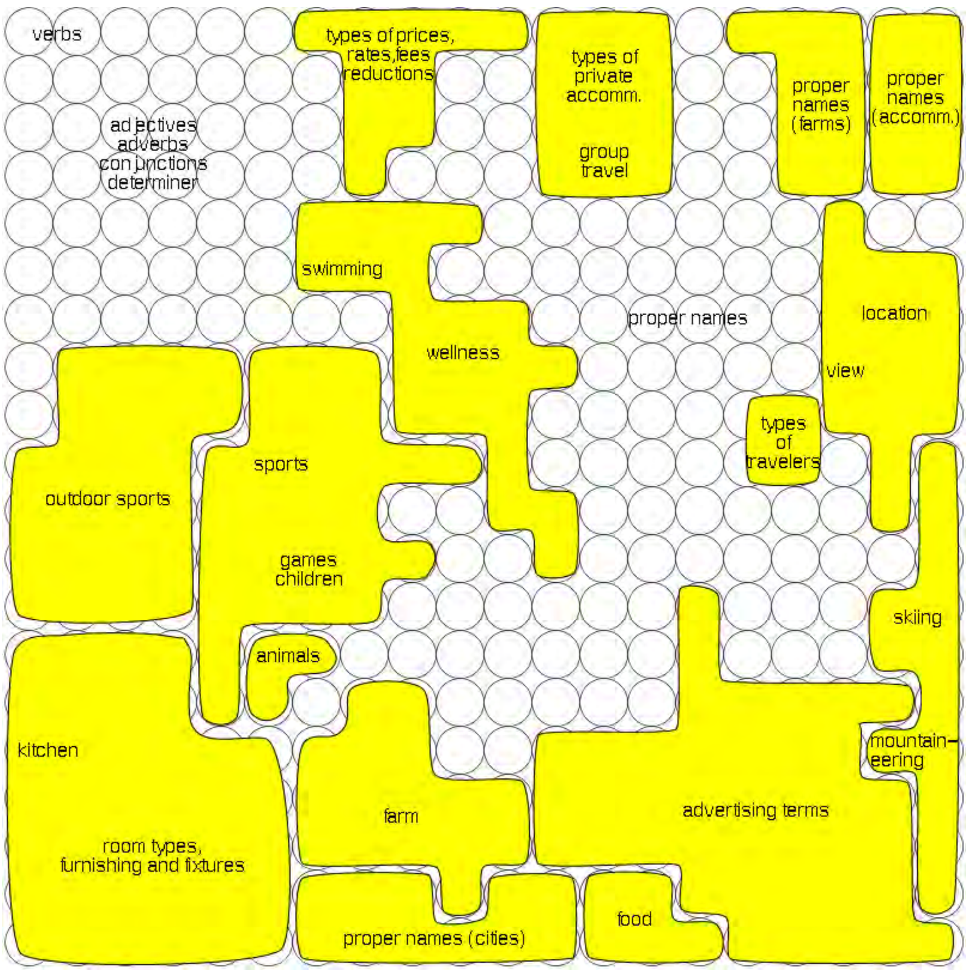
- benachbarte Karten der 2. Ebene



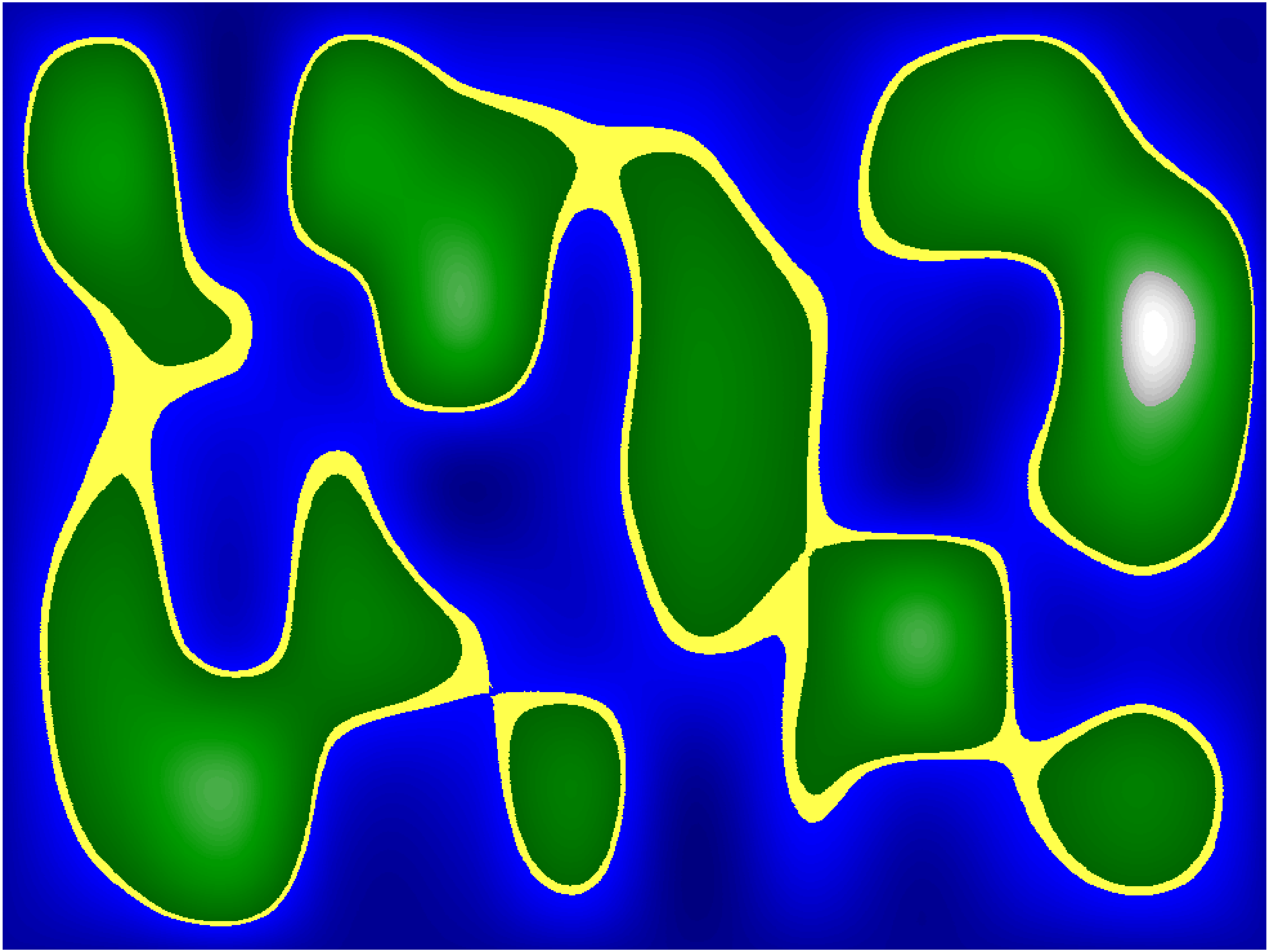
FODOK Pilot Study

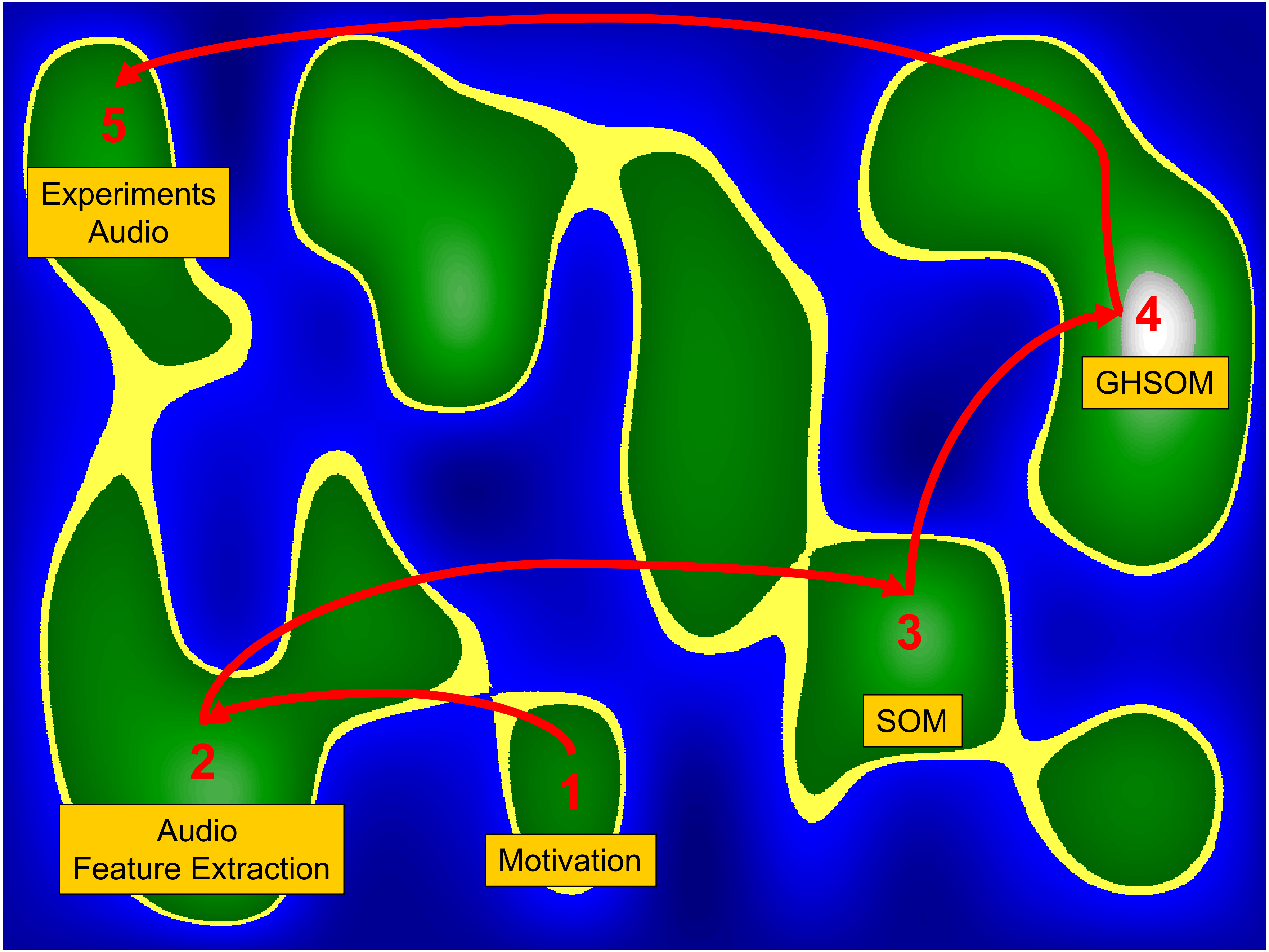


TISCOVER



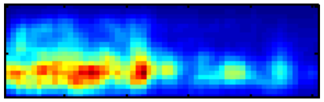
kochnische	bad
wanne	stockbett
sofa	doppelzimmern
badewanne	dusche
waschraum	schlafraeume
doppelbett	zimmerausstattung
schlafmoeglichkeiten	dreibettzimmer
hotelzimmer	wohnschlafraum
essraum	schlafzimmer
kochecke	zimmer
duschen	fliesswasser
kinderzimmer	einbettzimmer
schlafraum	komfortzimmer
wohnschlafzimmer	doppelschlafzimmer

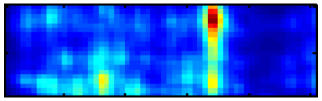



5
Experiments
Audio
4
GHSOM
3
SOM
2
Audio
Feature Extraction
1
Motivation

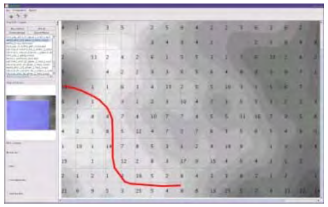
Music Case Studies

- Cluster music by perceived similarity ("genre")
- Music Features:
  - analyzing frequency spectra
  - Rhythm Patterns:
    amplitude modulation in different frequency bands
    psycho-acoustic transformations
    1.440-dimensional vectors per song

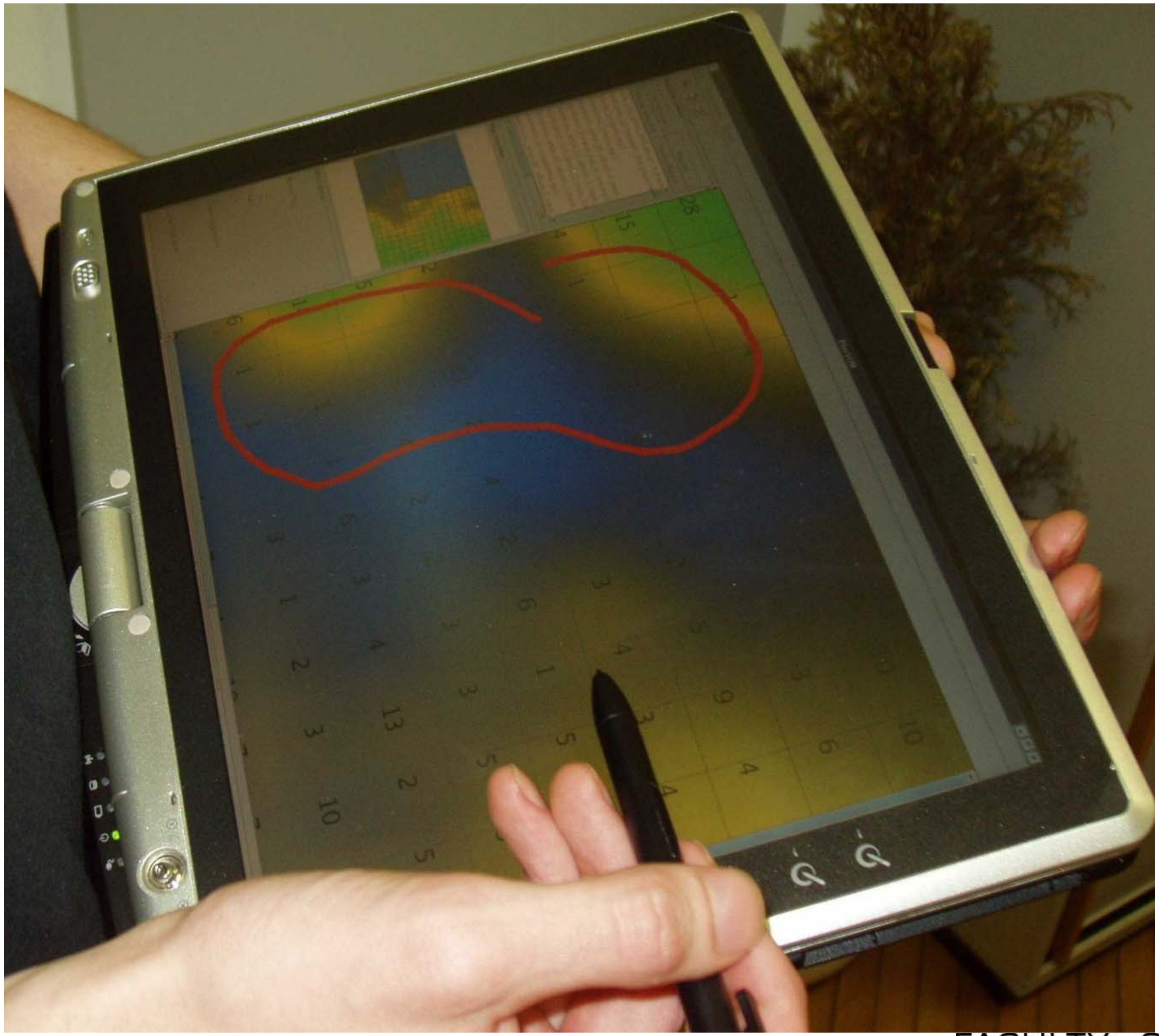




  - statistical spectrum descriptors (SDD),
    Marsyas features,...
- Prototypes:
  - PlaySOM for Desktop PC's
  - PocketSOMPlayer for PDA's

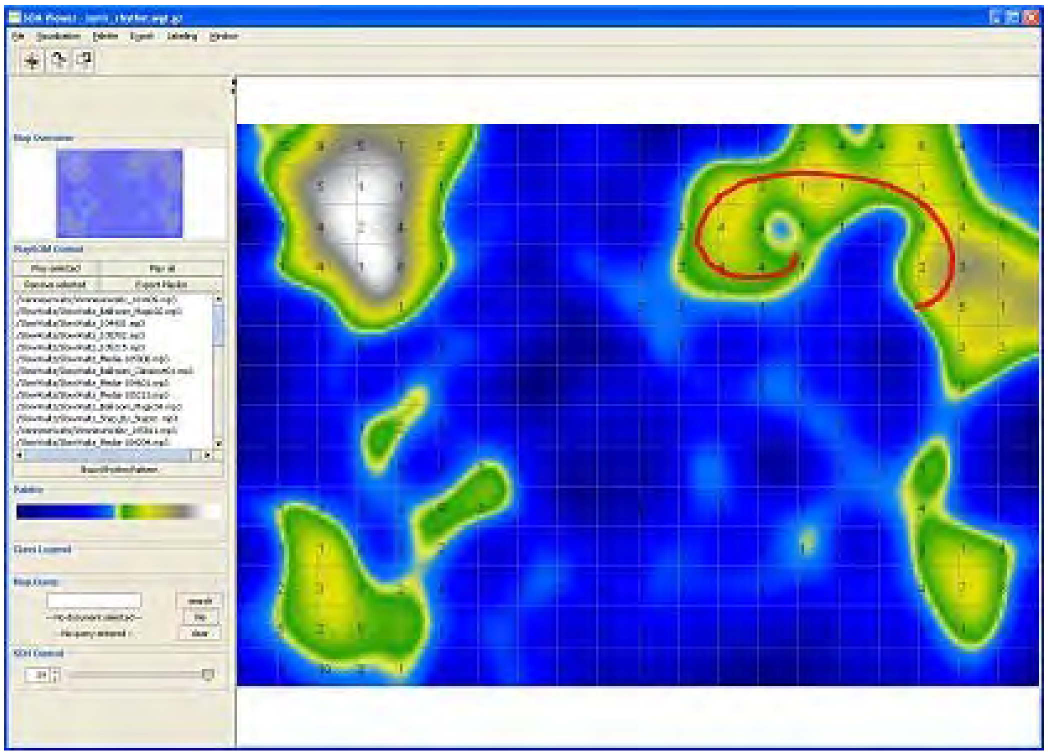


PlaySOM - Playlist Selection



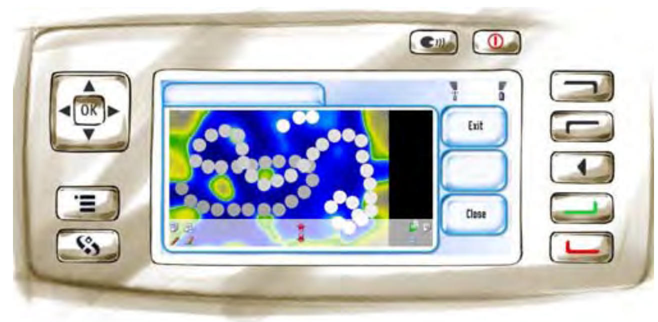
PlaySOM

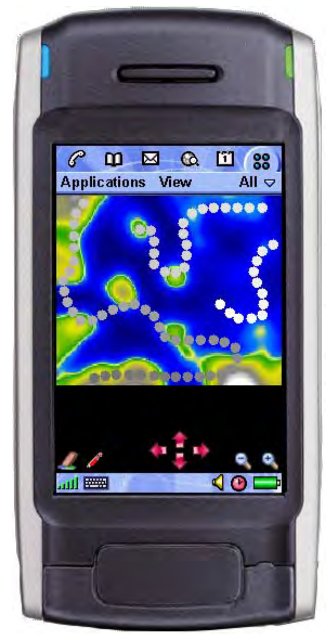
- Organizing Music
- Creating Playlists



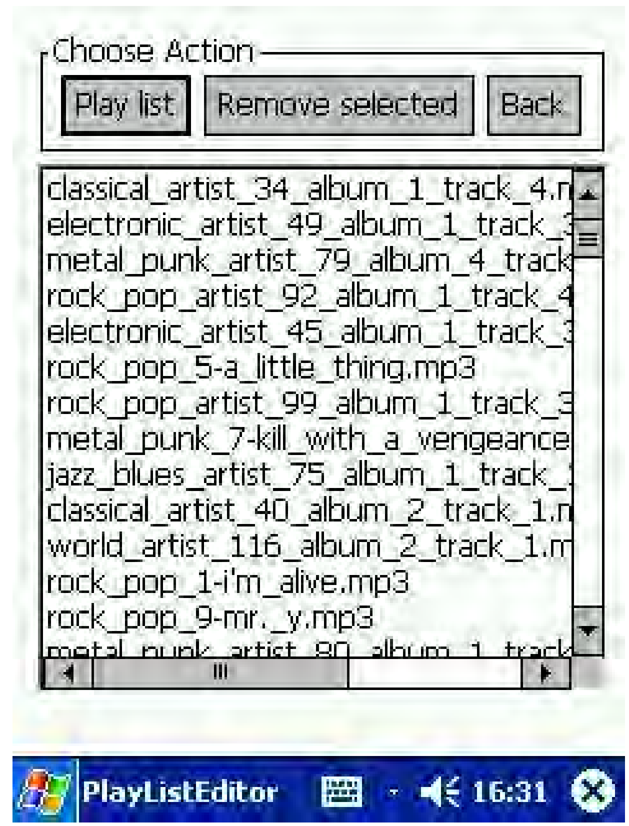
PocketSOM-Player

- Application for mobile devices
- Streaming audio
- Remote control



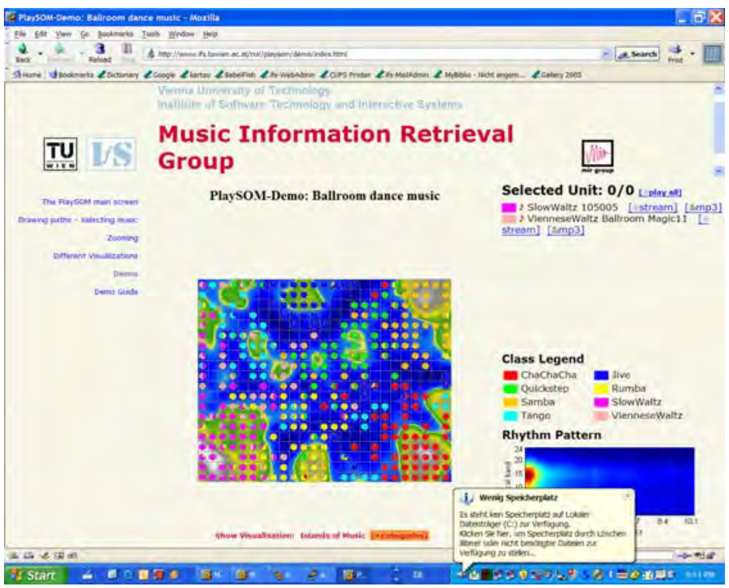


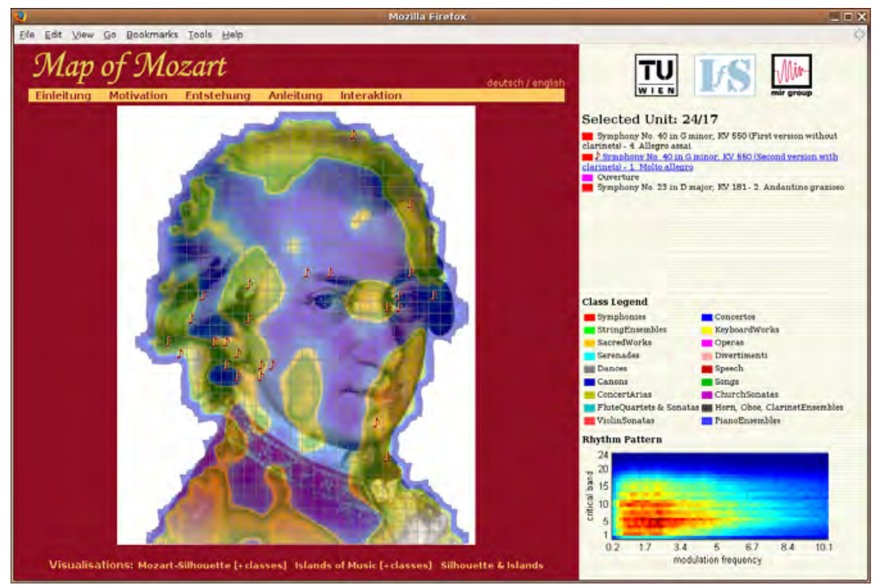
PocketSOMPlayer



Web-based Browsing

- Web-based interface
- Reduced functionality

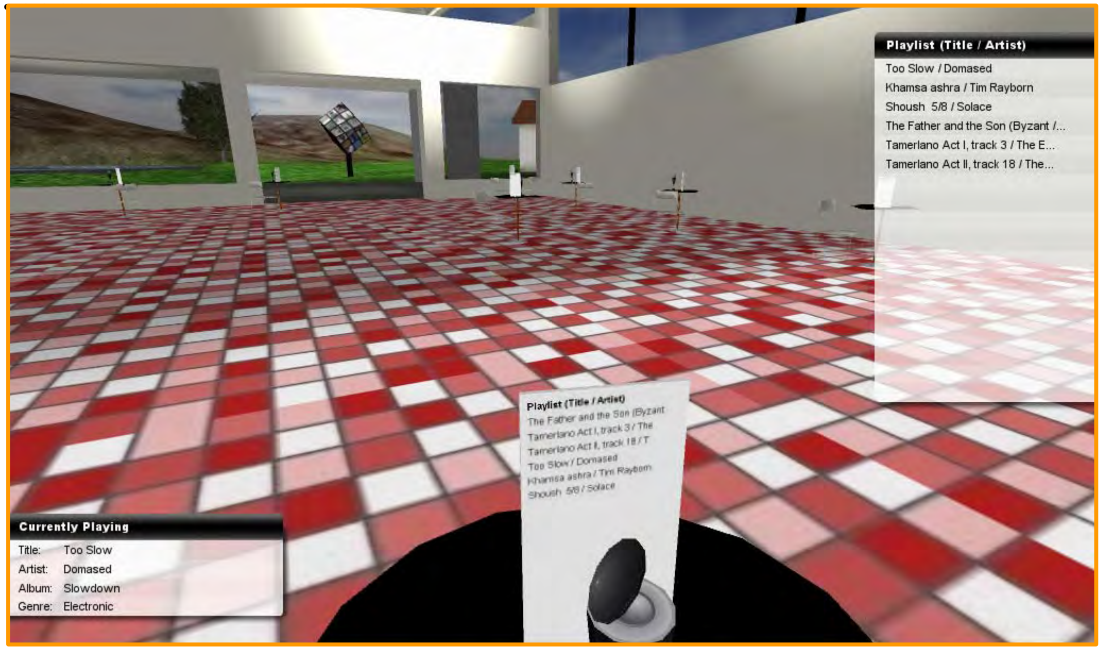




3D Music Worlds

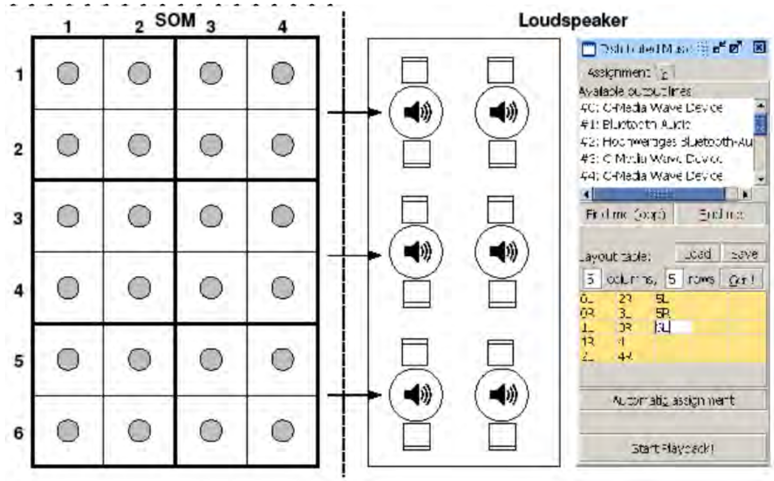
- SOM organizes music by sound similarity
- forms baseline for room set-up
  - real-life & virtual
- Coffee shop, tables, each table plays its music
  tables in a zone play similar music
- Get your coffee and choose a table where the music is to your liking (if there's one free there...)

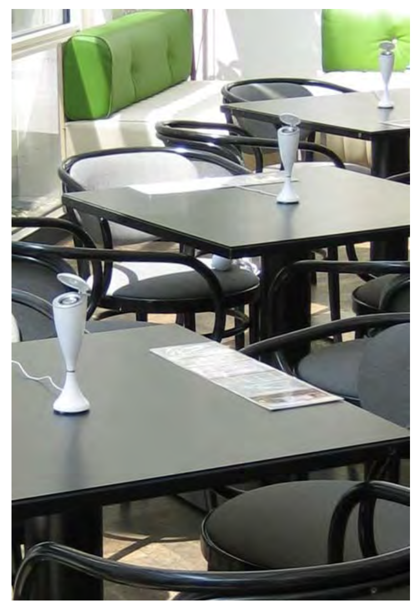
3D Music Worlds



http://ispaces.ec3.at/muscle.php

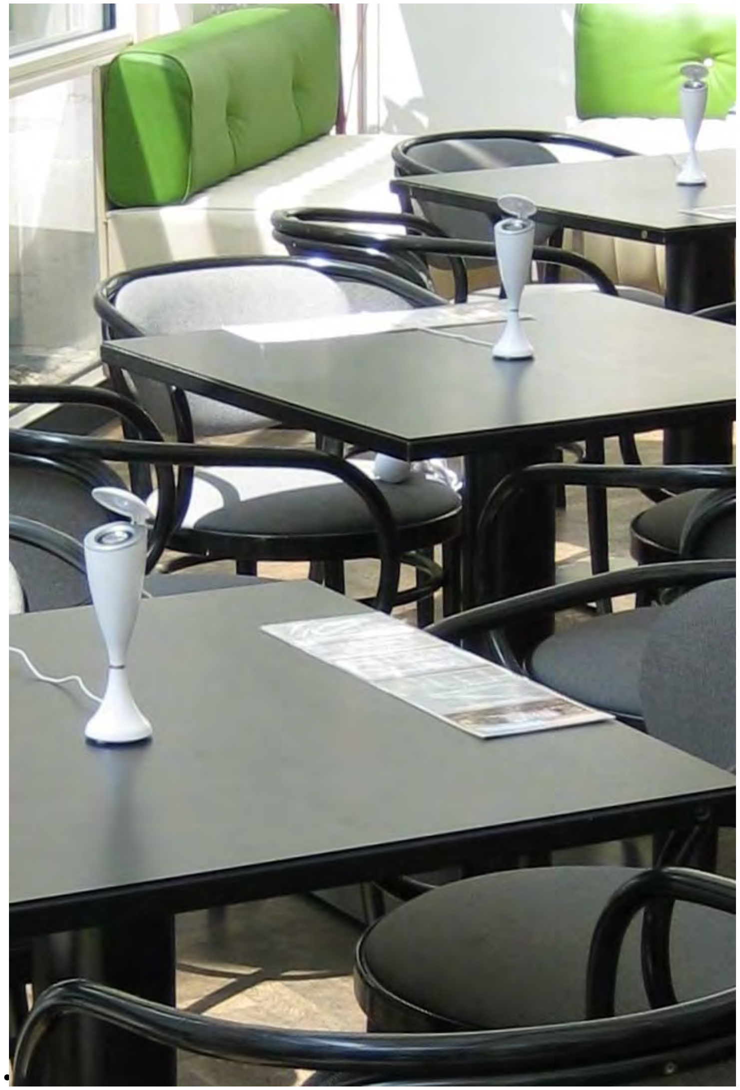
3D Music Worlds





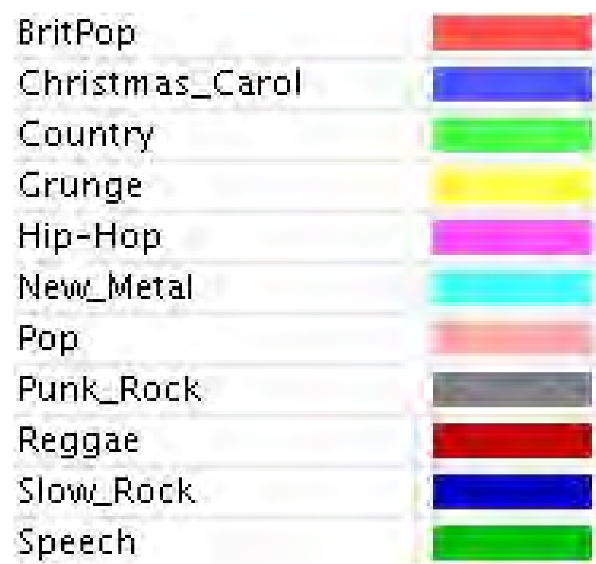
MusicCafe: Real Musical Spaces

- Each table represents a region on the SOM
- Every speaker plays music according to its position
- "Grab your coffee and choose your table"



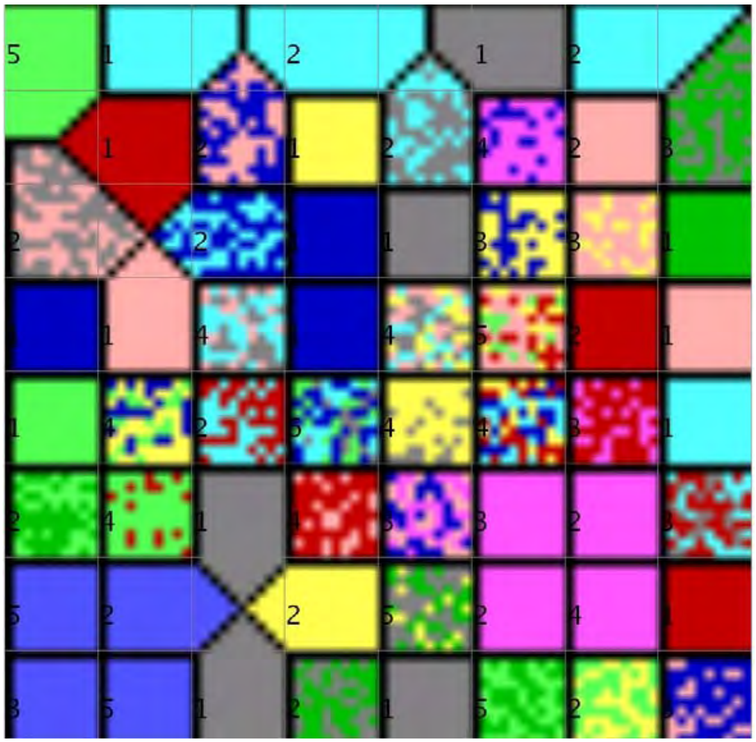
Comparing Multiple SOM Views

- Parallel corpus, indexed by song lyrics and music
- Clustering on a SOM for analysis
  - Lyrics SOM
  - Music SOM
- Analysis of cluster structure on both
- Class visualization based on genre labels

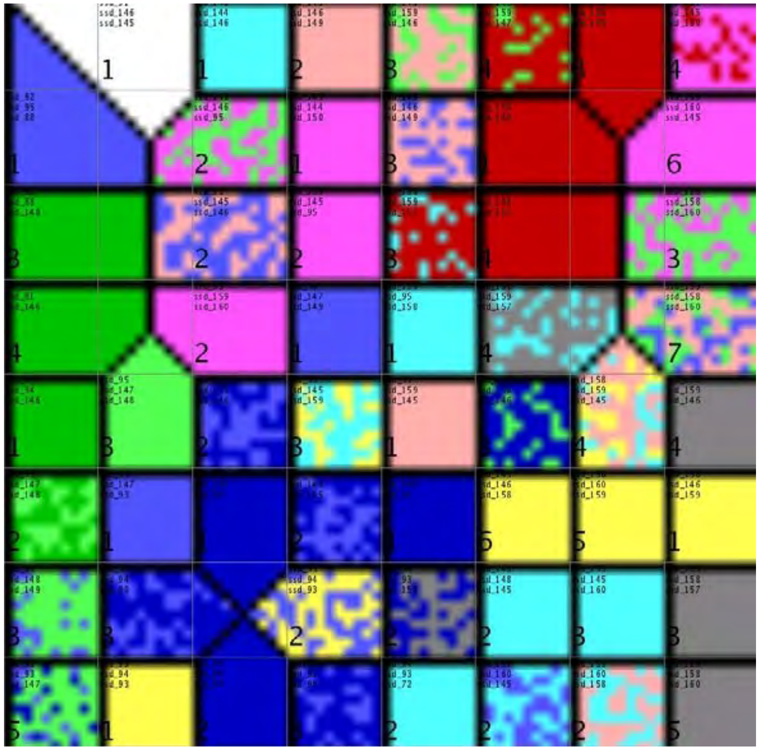


Text and Audio

BritPop
Christmas_Carol
Country
Grunge
Hip-Hop
New_Metal
Pop
Punk_Rock
Reggae
Slow_Rock
Speech

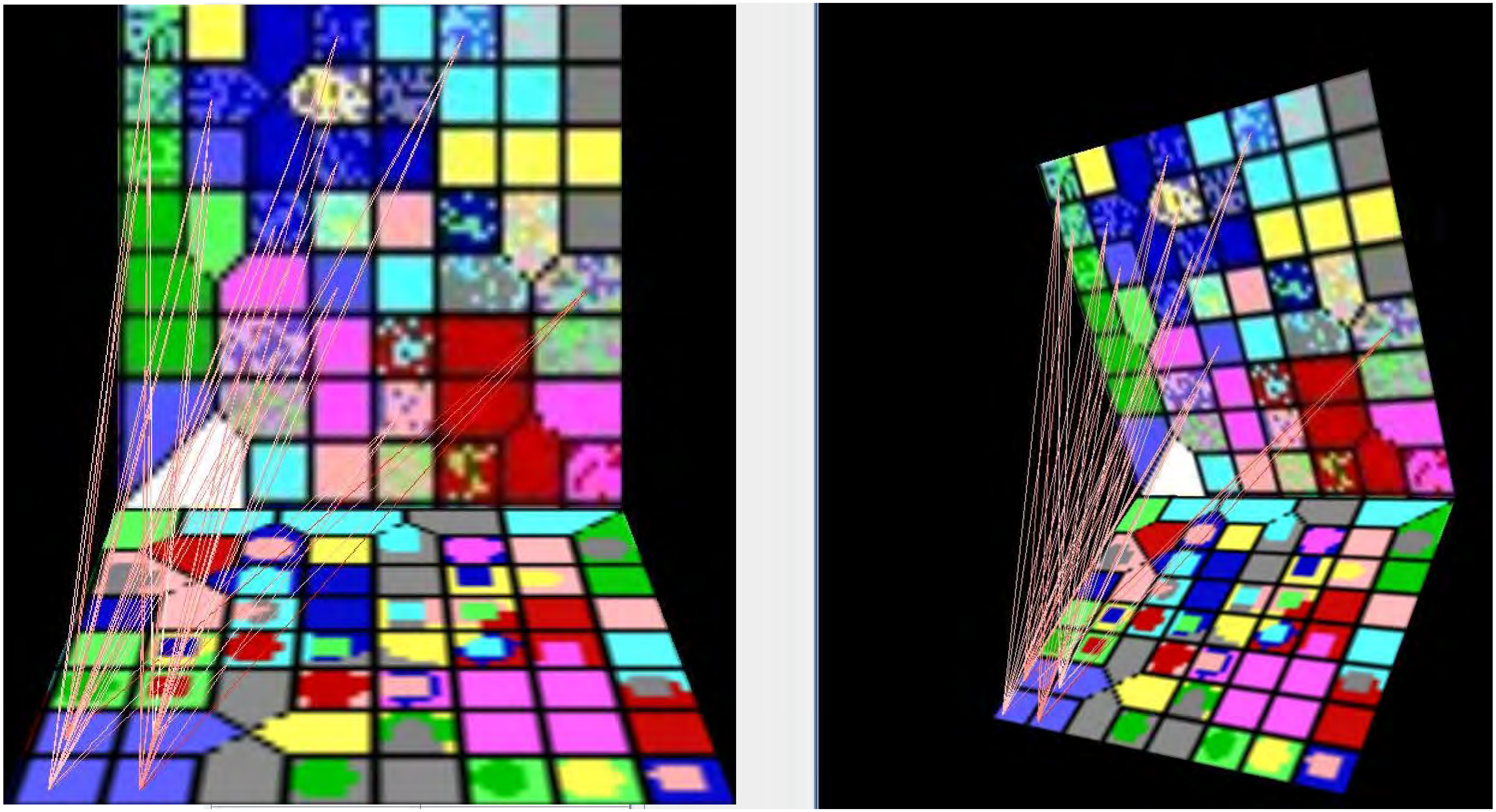


Lyrics SOM



Music SOM

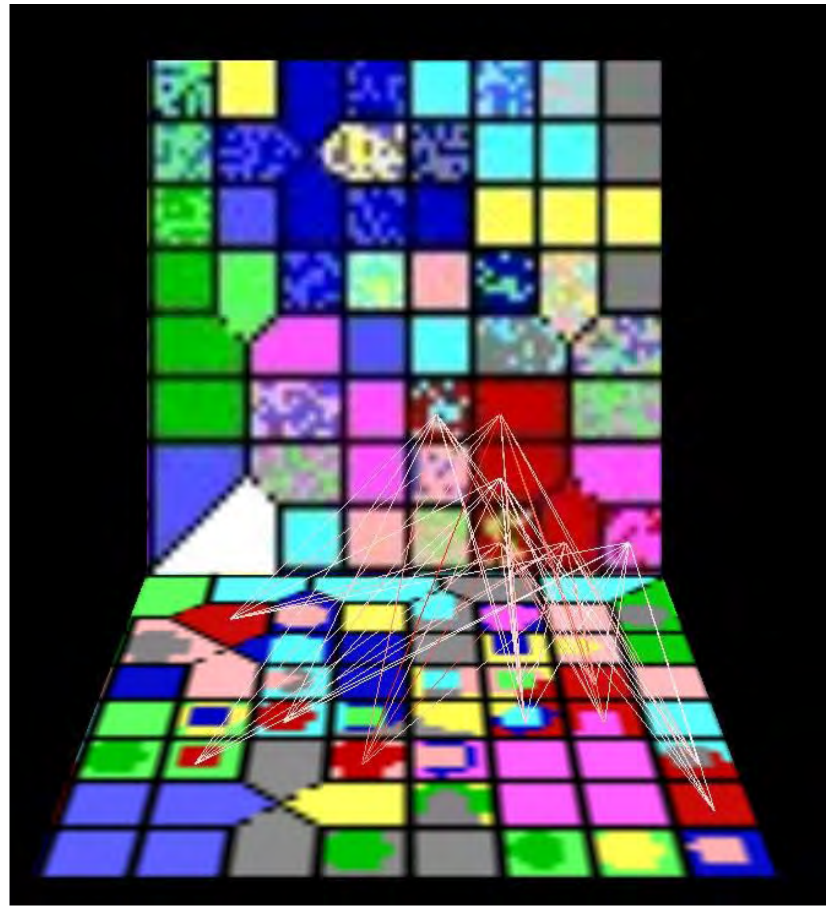
Text and Audio



Christmas songs

Text and Audio

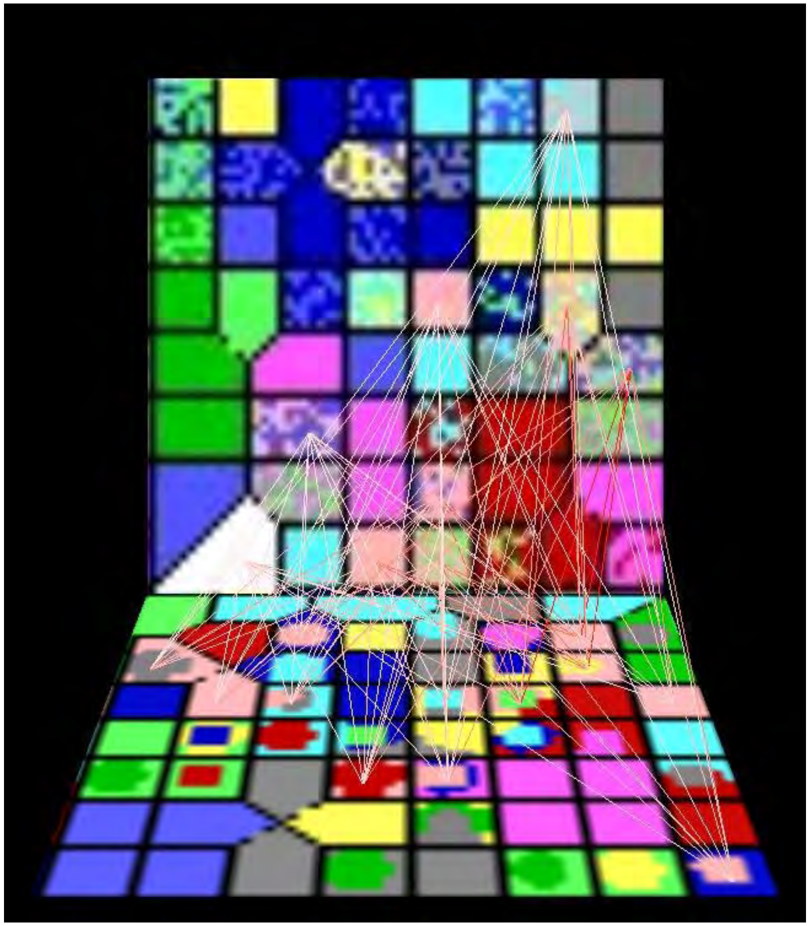
Speech



Reggae

Text and Audio

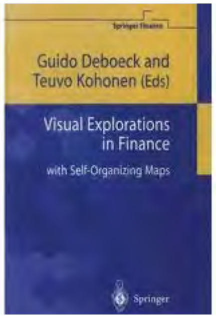
Hip-Hop



Pop

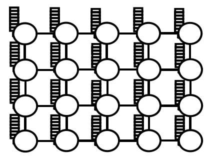
Applications

- Myriads of other applications
- From pure data anlysis to control loops
- Literature:
  - Samuel Kaski, Jari Kangas, and Teuvo Kohonen. Bibliography of self-organizing map (SOM) papers: 1981–1997. *Neural Computing Surveys, 1(3&4):1–176, 1998.*
  - Merja Oja, Samuel Kaski, Teuvo Kohonen. Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum. Neural Computing Surveys, 3, 1-156, 2002
  - Guide Deboeck, Teuvo Kohonen. Visual Explorations in finance with Self-Organizing Maps, Springer, 1998

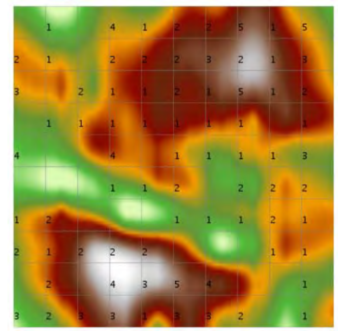


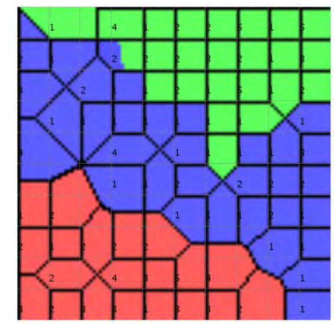
Outline

- SOM Basics

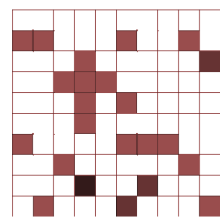


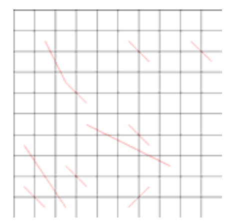
- Visualizations



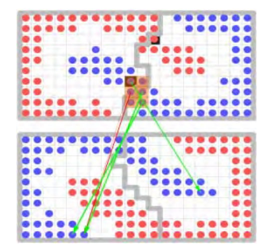


- SOM Quality Measures

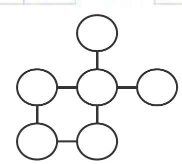


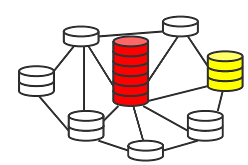


- SOM Comparison



- Related Architectures and Methods





- Applications