Data Mining for Automated GIS Data Collection

KARL-HEINRICH ANDERS, Oberkochen

ABSTRACT

The automatic analysis of spatial data sets presumes to have techniques for interpretation and structure recognition. Such procedures are especially needed in GIS and digital cartography in order to automate the time-consuming data update and to generate multi-scale representations of the data. In order to infer higher level information from a more detailed data set, coherent, homogeneous structures in a data set have to be delineated. There are different approaches to tackle this problem, e.g. model based interpretation, rule based aggregation or clustering procedures, which are part of the main topic called data mining. In the paper, a short introduction to data mining will be given and a parameter-free graph-based clustering approach is presented

1. INTRODUCTION

The ever increasing amount of data and information available demands for an automation of its use. Users need adequate search tools in order to quickly access and filter relevant information. Data Mining has evolved as a branch of computer science, which tries to structure data and find inherent, possibly important, relations in the data. In general, it deals with finding facts by inference; finding information in unstructured data, or in data which is not structured explicitly for the required purpose.

In GIS and digital cartography, respectively, there is a growing demand for such techniques: huge spatial data sets are being acquired and have to be kept up to date at ever increasing cycles; furthermore, information of different levels of detail is required in order to compensate for the requirements of different applications. One important application is the scale dependent data representation for quick visualization on a computer screen. In cartography, typically the data of different scales are acquired, managed and updated separately (a highly time consuming and labor intensive task). In order to accelerate update cycles and deliver actual information on-the-fly, tools and techniques for automation of initial data capture and update are required. In the following there will be a short introduction to the topic *Data Mining*, and a more detailed description of the Data Mining problem *Cluster Analysis*. At last a parameter-free graph-based clustering approach is presented.

2. DATA MINING

Data mining or *knowledge discovery in databases* (Piatetsky-Shapiro & Frawly, 1991), (Holsheimer & Siebes, 1994), (Fayyad et al., 1996) can be defined as the discovery of interesting, implicit, and previously unknown knowledge from large databases (Frawley et al., 1991). The subject spatial data mining is the extension of data mining from relational and transactional databases to spatial databases. Nowadays huge amounts of spatial data have been captured in various applications, ranging from remote sensing, to geographic information systems, environmental and planning. The human ability to analyze this large spatial databases manually is far exceeded. That makes it necessary to automate the information (knowledge) discovery to support a human operator.

The subject spatial data mining represents the integration of several fields, including machine learning (Michalski et al., 1984), database systems, data visualization, statistics (Shaw, 1994), information theory and computational geometry. Spatial data mining techniques have wide applications in geographic information systems and remote sensing (Koperski, 1996). These methods can be used for understanding spatial data, discovering relationships between spatial and nonspatial data, construction of spatial knowledge-bases, query optimization, characterization of spatial data. Possible applications for spatial data mining are described in (Anders, 1996, 1997), (Haala & Anders, 1996, 1997), and (Sester, 1998, 1999).

2.1. Data Mining Problems

Data mining problems can be classified as follows:

- **Classification methods:** Data items are mapped into predefined categorical classes.
- **Cluster analysis:** Cluster analysis is a branch of statistics that has been studied for many years. The goal of this method is to cluster objects into classes, based on their features, which maximize the similarity of class objects and minimize the similarity of objects from different classes. There are *probability-based* and *distance-based* clustering methods. Cluster analysis represents a type of *unsupervised learning*. The advantage of this method is that interesting structures or clusters can be found directly from the data without any background knowledge. Modified clustering techniques are described by (Ng & Han, 1994), (Ester et al., 1995) and (Ester et al., 1996).
- **Search for associations rules:** These methods look for characteristic rules that link one or more objects to other objects. Asociation rules are of the form $X \rightarrow Y$ *(s%,c%)*, where X and Y are spatial or nonspatial predicates, *s%* the support and *c%* the confidence of the association rule ((Koperski & Han, 1995), (Bollinger, 1996)). There are various kinds of spatial predicates that could constitute a spatial association rule, e.g. neighborhood (distance) information, topologic relations (*intersection, overlap,* etc.) or spatial orientation (*right_of, north_of,* etc.).
- **Aggregation-, approximation-methods :** These methods use spatial relations (topologic or geometric) to create new complex objects (Knorr & Ng, 1995) or to find patterns, structures (Regnauld, 1996) in the spatial database. The geometry of complex objects is described by an appropriate approximation of the aggregated geometries (e.g. convex hull). This type of spatial data mining methods often use concepts of computational geometry (Preparata & Shamos, 1985). An example for that kind of information is a created rule as the following: *80% of all houses in that cluster have a saddle roof.*
- **Time-series analysis:** Find similarities in sequential data. The analysis of the stock market trends is a typical time-series processing.
- **Dependency modeling:** Describes significant dependencies between variables.
- **Deviation analysis:** These methods looks for deviations from the expected values such as outliers in a class of objects. For example, finding unexpected credit charge operations can be performed by deviation analysis.
- **Summarization (Characterization):** Find a compact description of the data. These methods can include data visualization, statistical functions, generalized rules, and tables.
- **Prediction methods:** These methods maps a data item onto a numerical variable. Often used are the linear or nonlinear regression models.

3. CLUSTER ANALYSIS

In the context of data aggregation, there are many approaches in GIS and in digital cartography, namely in model or database generalization. (Richardson, 1996) and (Smaalen, 1996) present approaches to come from one detailed scale to the next based on a set of rules. If such rules are known or models of the situation are available, good results can be achieved (cf. Sester et al, 1998). However, the main problem being the definition of the rules and the control strategy to infer new data from it (Ruas, 1995). Current concepts try to integrate learning techniques for the derivation of the necessary knowledge (Plazanet et al:1998), (Sester, 1999).

Clustering is a well established technique for data interpretation. It usually requires prior information, e.g. about the statistical distribution of the data or the number of clusters to detect. Existing clustering algorithms, such as k-means (Jain:1988), PAM (Kaufman, 1990), CLARANS (Ng & Han, 1994), DBSCAN (Ester et al., 1996), CURE (Gua et al.,1998), and ROCK (Gua et al., 1999) are designed to find clusters that fit some static models. For example, k-means, PAM, and CLARANS assume that clusters are hyper-ellipsoidal or hyper-spherical and are of similar sizes. The DBSCAN algorithm assumes that all points of a cluster are *density reachable* (Ester et al., 1996) and points belonging to different clusters are not. All these algorithms can breakdown if the choice of parameters in the static model is incorrect with regarding to the data set being clustered, or the model did not capture the characteristics of the clusters (e.g. shapes, sizes, densities). In the following, we give a brief overview of existing clustering algorithms.

3.1. Non-hierarchical Schemes

Non-hierarchical clustering techniques are also called partitional clustering techniques. These approaches attempt to construct a simple partitioning of a data set into a set of k non-overlapping clusters such that the partitions optimize a given criterion. Each cluster must contain at least one data element, and each data element must belong to exactly one group. In most of the partitional methods an initial partitioning is chosen and then the cluster membership is changed in order to obtain a better partitioning. *Centroid based* methods like the k-means method (MacQueen, 1967), (Jain:1988) and the ISODATA (Ball, 1965) method try to assign data elements to clusters such that the mean square distance of data elements to the centroid of the assigned cluster is minimized. These techniques are suitable only for data in metric spaces, because they have to compute a centroid of a given set of data elements. *Medoid based* approaches as CLARANS (Ng & Han, 1994) and PAM (Kaufman, 1990) try to find a so called medoid which is a representative data element that minimize the sum of the distances between the medoid and the data elements assigned to this medoid.

One disadvantage of centroid and medoid based methods is that not all values of k lead to natural cluster so it is useful to run the algorithm several times with different values for k to select the best partition. With a given optimization criterion this decision can be automated. The main drawback of both methods is that they will fail for data sets in which data elements belonging to a cluster are closer to the representative of another cluster than to the representative of their own cluster. This case is typical for many natural clusters if the cluster shapes are concave or their sizes vary largely.

3.2. Hierarchical Schemes

Hierarchical cluster schemes constructs a dendrogram is a tree structure which represents a sequence of nested clusters. This sequence represents multiple levels of partitioning. On the top is a single cluster which includes all other clusters. At the bottom are the data elements representing single element clusters. Dendrograms can be constructed top-down or bottom-up. The bottom-up method is known as the agglomerative approach, where each data element starts out as a separate cluster. In each step of an agglomerative algorithm the two most similar clusters are grouped together based on similarity measures in subsequent steps and the total number of clusters is decreased by one. These steps can be repeated until one large cluster remain or a given number of clusters is obtained or the distance between two closest clusters is above a certain threshold. The top-down method known as the divisive approach works in the reverse direction. Agglomerative methods seems to be the most popular in the literature.

In the literature one can find many different variations of hierarchical algorithms. Basically, these algorithms can be distinguished by their definition of similarity and how they update the similarity between existing clusters and the merged clusters. In general, the approaches described are alternative formulations or minor variations of *centroid* (*medoid)* based concepts, *linkage* based concepts, and *variance* or *error sum of squares error* concepts.

The centroid or medoid based approaches also fail on clusters of arbitrary shapes and different sizes like non-hierarchical methods, such as k-means and k-medoid. The oldest linkage based method is the *single linkage* algorithm, sometimes referred to as the nearest neighbor approach. In the single linkage method, no representative exists. The cluster is represented by all data elements in the cluster and the similarity between two clusters is the distance between the closest pair of data elements belonging to different clusters. The single linkage method is able to find clusters of arbitrary shape and different sizes, but it will fail at poorly separated clusters and is susceptible to noise and outliers.

In order to avoid these drawbacks algorithms like the *shared near neighbors method* (Jarvis & Patrick, 1973), *CURE* (Gua et al., 1998) or *ROCK* (Gua et al., 1999) were proposed. Instead of using a single centroid to represent a cluster, CURE choose a constant number of representative points to describe a cluster. The ROCK algorithm operates on a derived similarity graph and scales the aggregate *inter-connectivity* with respect to a predefined inter-connectivity model. The shared near neighbors method use a k-nearest-neighbor graph to determine the similarity between two clusters. The advantage of this clustering method over most other alternatives is that it is independent of absolute scale.

A major limitation of existing agglomerative hierarchical schemes such as the *Group Averaging Method* (Jain, 1988), CURE, and ROCK is that the merging decisions are based on static modeling of the clusters to be merged. More information about the limitations of existing hierarchical methods can be found in (Karypis, 1999).

4. GRAPH-BASED CLUSTERING

The most powerful methods of clustering in difficult problems, which give results having the best agreement with human performance, are the graph-based methods (Jaromczyk, 1992). The idea is extremely simple: Compute a neighborhood graph (such as the minimal spanning tree) of the original points, then delete any edge in the graph that is much longer (according to some criterion) than its neighbors. The result is a forest and each tree in the forest represents a cluster.

In general, hierarchical cluster algorithms work implicitly or explicitly on a similarity matrix such that every element of the matrix represents the similarity between two elements. In each step of the algorithm the similarity matrix is updated to reflect the revised similarities. Basically, all these algorithms can be distinguished based on their definition of similarity and how they update the similarity matrix. In spatial clustering algorithms one can discriminate between *spatial similarity* and *semantic similarity* which means the similarity of non-spatial attributes.

Spatial similarity implies the definition of a neighborhood concept which can be defined on geometric attributes, such as coordinate, distance, density, and shape. The computation of a spatial similarity matrix can be seen as the construction of a weighted graph, so called *neighborhood graph*, where each element is represented by a node and each neighborhood relationship (similarity) is an edge. There are efficient algorithms to compute neighborhood graphs (Jaromczyk, 1992) which can be used to compute a spatial similarity matrix.

4.1. Neighborhood Graphs

A general introduction to the subject of Neighborhood graphs is given in (Jaromczyk, 1992). Neighborhood graphs also called *proximity graphs* (Toussaint, 1991), are used as tools in disciplines where shape and structure of point sets are of primary interest. These include for example visual perception, computer vision and pattern recognition, cartography and geography, and biology.

Neighborhood graphs capture proximity between points by connecting nearby points with a graph edge. The many possible notions of *nearby* (in several metrics) lead to a variety of related graphs. It is easiest to view the graphs as connecting points only when certain regions of space are empty. In the following definitions of proximity graphs we will use these notations:

- V: A set of n points in R^d.
- $Edge(p,q)$: The points p and q have a common edge.
- $NN(p)$: The nearest neighbor of p.
- $\delta(p,q)$: The distance between two points p and q using a given metric.
- Ball: The open ball $B(p,r) = \{q \mid \delta(p,q) < r\}$.
- Lune: $L(p,q) = B(p,\delta(p,q)) \cap B(q,\delta(p,q))$.
- β - Lune: $L_\beta(p,q) = B(p(1-\frac{\beta}{2}) + q\frac{\beta}{2}, \frac{\beta}{2}\delta(p,q)) \cap B(q(1-\frac{\beta}{2}) + p\frac{\beta}{2}, \frac{\beta}{2}\delta(p,q))$.

Some well known proximity graphs are:

- The delaunay triangulation (*DT(V)*)
- The nearest neighbor graph (Jarvis, 1973)
  $NNG(V) = \{Edge(p,q) \mid p,q \in V \wedge B(p,\delta(p,q)) \cap V = \varnothing\}$
- The minimum spanning tree (*MST(V)*).
- The relative neighborhood graph (figure 1a)) (Toussaint, 1980)
  $RNG(V) = \{Edge(p,q) \mid p,q \in V \wedge L_2(p,q) \cap V = \varnothing\}$
- The gabriel graph (Gabriel, 1969)
  $GG(V) = \{Edge(p,q) \mid p,q \in V \wedge L_1(p,q) \cap V = \varnothing\}$
- The β-skeleton (Kirkpatrick, 1985)
  $G_\beta(V) = \{Edge(p,q) \mid p,q \in V \wedge L_\beta(p,q) \cap V = \varnothing\}$
- The sphere of influence graph (Toussaint, 1988)
  $SIG(V) = \{Edge(p,q) \mid p,q \in V \wedge B(p,\delta(p,NN(p))) \cap B(q,\delta(q,NN(q))) \neq \varnothing\}$

The important relationship between some proximity graphs is that they build a part of hierarchy. Given a point set V and a metric, then for any $\beta \in [1,2]$ the following hierarchy is valid:

$$NNG \subseteq MST \subseteq RNG \subseteq G_\beta \subseteq GG \subseteq DT\ .$$

In figure 1b) the hierarchical relationship between the Nearest Neighbor Graph, the Relative Neighborhood Graph, the Gabriel Graph, and the Delaunay Triangulation of a point set is shown.

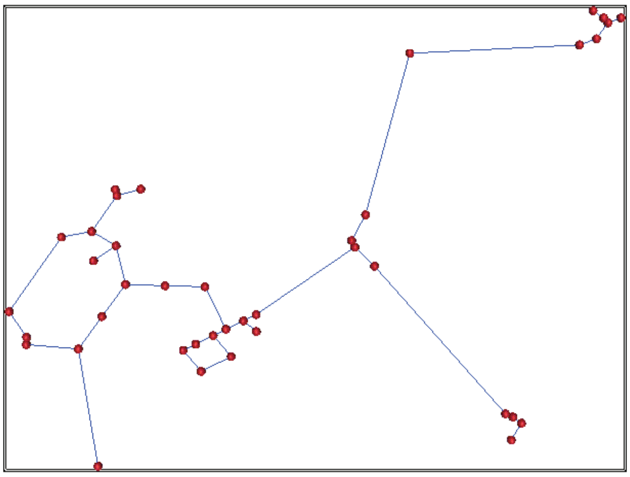


Figure 1a): RNG of a point set.

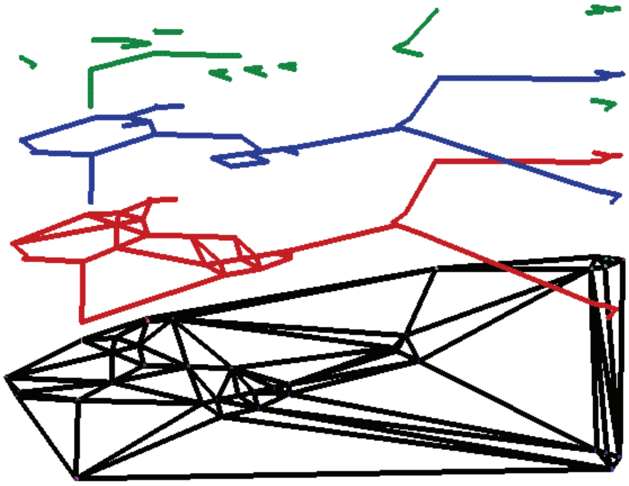


Figure 1b): Hierarchy of proximity graphs.

The computation of such a hierarchy needs *O(n log n)* time, because the computation of the Delaunay Triangulation needs *O(n log n)* time and any sub graph can be computed from its super graph in *O(n)* time. For example, an algorithm for the RNG in the Euclidean metric using the Delaunay Triangulation was developed by (Supowit, 1983).

4.2. Hierarchical Graphbased Clustering

In our approach we use the hierarchical relationship between proximity graphs to represent a near to a far neighborhood model. Our algorithm can be described as follows:
The first basic step is the computation of the Delaunay Triangulation (DT) from a given set of points. In the next step we compute first the Gabriel Graph (GG) from the DT, second the Relative Neighborhood Graph (RNG) from the GG, and third the Nearest Neighbor Graph (NNG) from the RNG (figure 1a)). Then we activate the edges of the NNG to start with the nearest neighbor model. Then all given points (graph nodes) are initialized as a single cluster. Every cluster contains a set of *inner* edges and a set of *outer* edges. The inner edges connect nodes which belongs to the same cluster and the outer edges connect nodes which belongs to different clusters. Every cluster is characterized by the median of the inner edge sizes (*cluster density*) and the *cluster variance*. The cluster variance is the median deviation of all inner and outer edge sizes from the cluster density. Using the inner and outer edges to compute the variance introduce an uncertainty factor to our model. At the beginning every initial cluster has no inner edges and therefore a density of zero, but the variance will be none zero, because every node in the NNG belongs at least to one edge. All initial clusters are put into a priority queue, ordered by their density and variance values. The first cluster in the priority queue is selected (cluster with the highest density) and merged with all of his *valid* neighbor clusters. Valid neighbor clusters are clusters which are connected by an outer edge and meet constraints, but first some used notations:

- $\tilde{C}_X = Median\{\delta(x,y) \mid Edge(x,y) \in X\}$: Median of the edge sizes in cluster X.
- $\tilde{\tilde{C}}_X = Median\left\{ \left|\delta(x,y) - \tilde{C}_X\right| \mid Edge(x,y) \in X\right\}$: Median absolute deviation of the edge sizes in cluster X.
- $\Lambda C_X = \left[\tilde{C}_X - \tilde{\tilde{C}}_X, \tilde{C}_X + \tilde{\tilde{C}}_X\right]$: Confidence interval of the edge size in cluster X.
- $\Delta C_{X,Y} = Median\{\delta(x,y) \mid Edge(x,y) \in X \cap Y\}$: Median distance of two clusters X and Y.

Two clusters X and Y can be merged if the following three constrains are valid:

- **Density compatibility:** $\tilde{C}_X \in \Lambda C_Y \wedge \tilde{C}_Y \in \Lambda C_X$.
- **Distance compatibility:** $\Delta C_{X,Y} \in \Lambda C_X \wedge \Delta C_{X,Y} \in \Lambda C_Y$.
- **Variance compatibility:** $\tilde{\tilde{C}}_{X,Y} \leq \min\left\{\tilde{\tilde{C}}_X, \tilde{\tilde{C}}_Y\right\}$.

After the merging all valid neighbor clusters are removed from the priority queue. Then repeat the selecting and merging step until no more clusters with valid neighbors can be found. The result are the clusters based on the NNG. In the next step the RNG edges are activated an the same procedure as for the NNG is repeated. Then the GG edges are activated and finally the edges of the DT are processed. Figure 2a) and 2b) shows the segmentation of an artificial test set with and without noise.

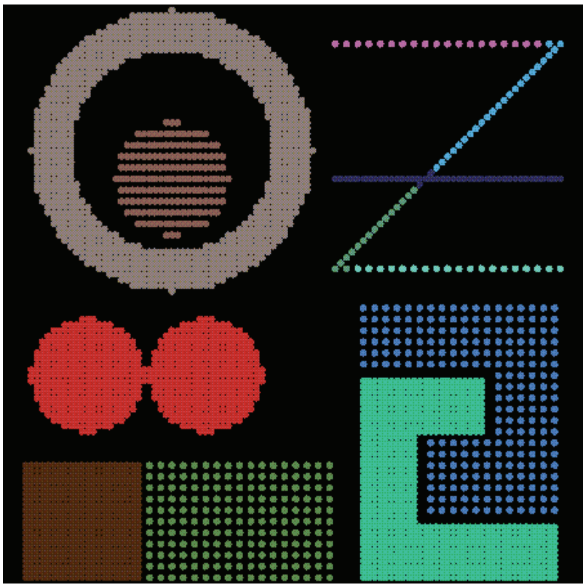


Figure 2a): Cluster in Test set without noise.

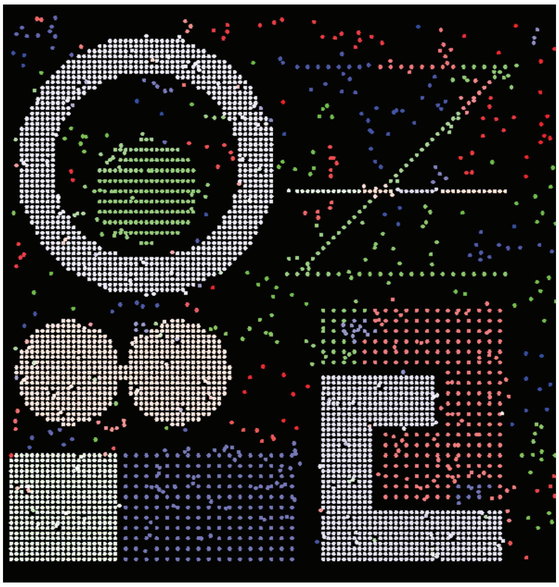
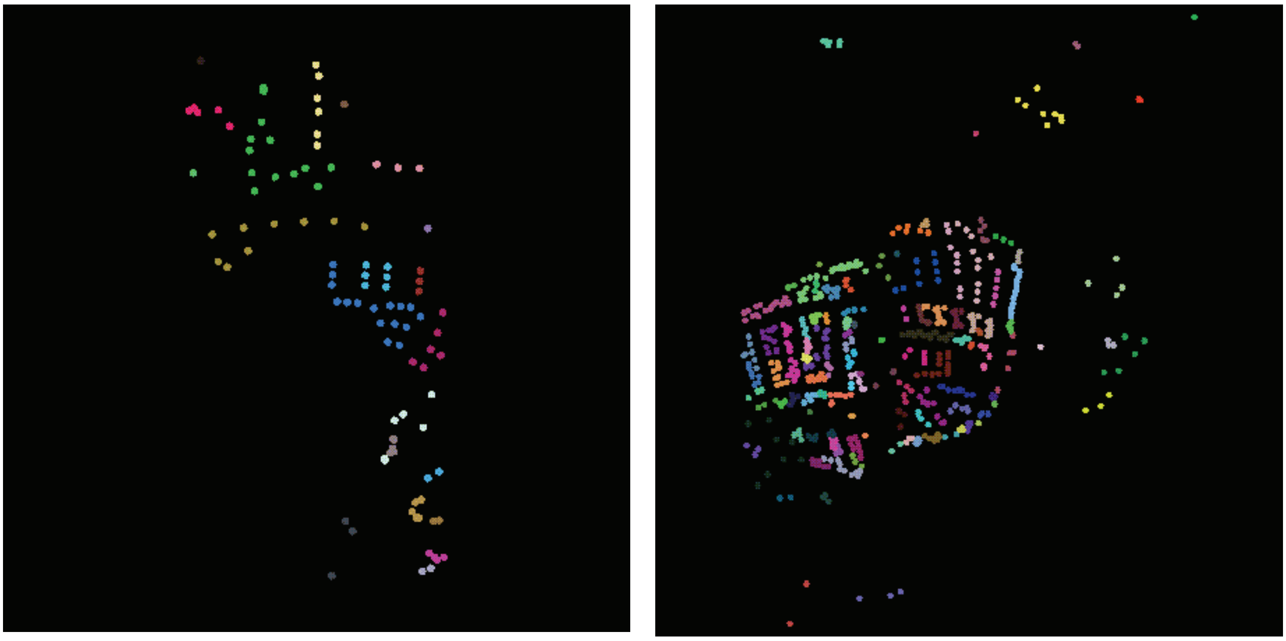


Figure 2b): Cluster in Test set with noise.

One basic aim of our approach was to detected building clusters for map generalization (Sester, 2000). Figure 3a) and 3b) shows the clustering result of two 2D point sets (centroids) derived from 2D building ground plans. We applied our clustering method also to a measured 3D object point cloud. Figure 4a) shows the result of a special segmentation method using a surface model, the surface curvature, and requires some user-defined parameters. Figure 4b) shows the result of our clustering process without any user-defined parameters using only the given 3D points.

5. CONCLUSION

The scope of the paper was to motivate the usefulness and the need for techniques of spatial data base interpretation. Because of the increasing amount of spatial information in digital form and the importance of data actuality and data quality for economy, industry and commerce it will become more important to automate the interpretation and revision of digital landscape models. Spatial data mining techniques are one tool to make the data reuse possible and also allow for a utilization of the data beyond their original purpose.



a): b):

Figure 3: Examples for detected building clusters.

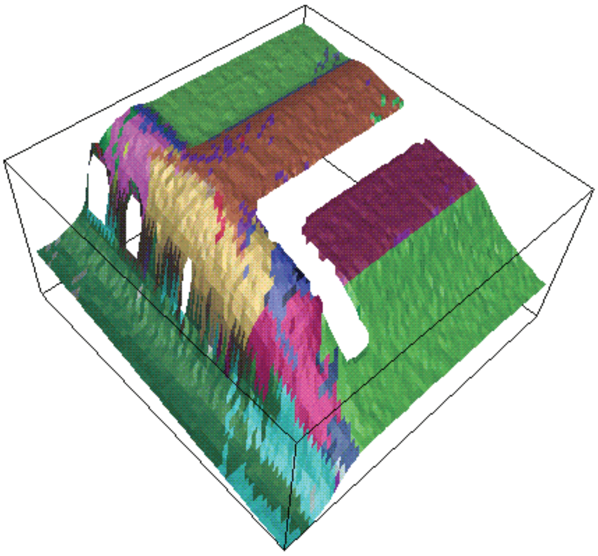


Figure 4a): 3D reference segmentation.

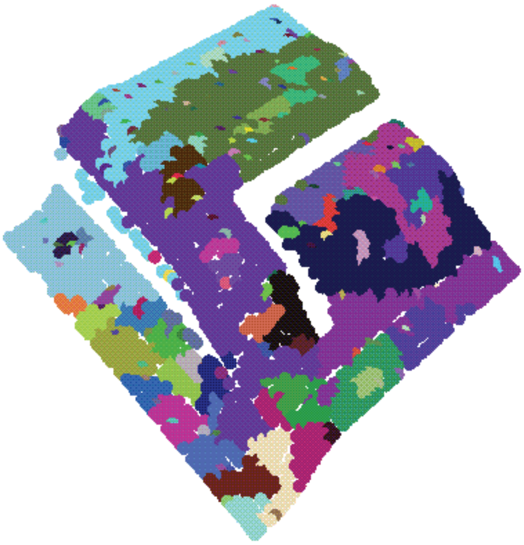


Figure 4b): 3D graph based segmentation.

6. REFERENCES

Anders, K.-H., D. Fritsch (1996): Automatic interpretation of digital maps for data revision, in 'International Archives of Photogrammetry and Remote Sensing', Vol. 31/4, ISPRS, Vienna, Austria, pp. 90-94.

Anders, K.-H., Sester, M. (1997), Methods of Data Base Interpretation – Applied to Model Generalization from Large to Medium Scale, Semantic modeling for the acquisition of topographic information from images and maps In: SMATI 97, Ed. By W. Förstner; L. Plümer, Birkhäuser Verlag, Basel, pp.89-103.

Ball, G., Hall, D., (1965): Isodata: A novel method of data analysis and pattern classification. Technical Report AD 699616, Stanford Research Institute.

Bollinger, T. (1996): 'Assoziationsregeln - Analyse eines Data Mining Verfahrens', Informatik Spektrum **19**(5), 257.

Ester, M., Kriegel, H.-P., Sander, J. (1997), Spatial Data Mining: A Database Approach, in: Proc. of the 5th Int. Symposium on Large Spatial Databases (SSD 97), Berlin, Germany, Lecture Notes in Computer Science, Springer-Verlag, Berlin.

Ester, M., H.-P. Kriegel, J. Sander, X. Xu (1996): A density-based algorithm for discovering clusters in large spatial databases with noise, in 'Proceedings of 2nd. International Conference on Knowledge Discovery and Data Mining (KDD-96)'.

Ester, M., H.-P. Kriegel, X. Xu (1995): Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification, in 'Advances in Spatial Databases (Proc. 4th Symp. SSD-95)', Portland, ME, pp. 67-82.

Fayyad, U., G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (1996): Advances in Knowledge Discovery and data Mining, AAAI/MIT, Menlo Park, CA.

Fotheringham, S., P. Rogerson (1994): Spatial Analysis and GIS, Taylor and Francis.

Frawley, W., G. Piatetsky-Shapiro, C. Matheus (1991): Knowledge discovery in databases: An overview, in G. Piatetsky-Shapiro & W. Frawley, eds, 'Knowledge Discovery in Databases', AAAI/MIT Press, Menlo Park, CA, pp. 1-27.

Gabriel, K., Sokal, R., (1969): A new statistical approach to geographic variation analysis. In Systematic Zoology, Vol.18, pp. 259-278.

Gua, S., Rastogi, R., Shim, K., (1998): Cure: An efficient clustering algorithm for large databases. In: Proc. of 1998 ACM-SIGMOD International Conference on Management of Data.

Gua, S., Rastogi, R., Shim, K., (1999): Rock: A robust clustering algorithm for categorical attributes. In: Proc. of the 15th International Conference on Data Engineering.

Haala, N., K.-H. Anders (1996): Fusion of 2d-gis and image data for 3d building reconstruction, in 'International Archives of Photogrammetry and Remote Sensing', Vol. 31/3, ISPRS, Vienna, Austria, pp. 285-290.

Haala, N., K.-H. Anders (1997): Acquisition of 3D urban models by analysis of aerial images, digital surface models and existing 2D building information, in 'SPIE Conference on Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision III', Orlando, Florida, April 1997, pp. 212-222.

Holsheimer, M., A. Siebes (1994): Data mining: The search for knowledge in databases, Technical Report CS-R9406, CWI, Amsterdam, The Netherlands.

Jain, A., Dubes, R., (1988): Algorithms for Clustering Data. Prentice Hall.

Jaromczyk, J., Toussaint, G., (1992): Relative neighborhood graphs and their relatives. In: Proc. IEEE, Vol. 80, number 9, pp. 1502-1517.

Jarvis, R., Patrick, E., (1973): Clustering using a similarity measure based on shared near neighbors. In: IEEE Transactions on Computers, Vol. 22, number 11, pp. 1025-1034.

Karypis, G., Han, E.-H. S., Kumar, V., (1999): Chameleon: A hierarchical clustering algorithm using dynamical modeling. To appear in the IEEE Computer or via internet at http://winter.cs.umn.edu/karypis/publications/data-mining.html.

Kaufman, L. Rousseeuw, P., (1990): Finding Groups in Data: An introduction to Cluster Analysis. John Wiley & Sons.

Kirkpatrick, D.G., Radke, J.D, (1985): A framework for computational morphology, in: G.T. Toussaint, ed., Computational Geometry, North-Holland, pp. 217-248.

Knorr, E., R. Ng (1995): Applying computational geometry concepts to discovering spatial aggregate proximity relationships, Technical report, University of British Columbia.

Koperski, K., J. Han (1995): Discovery of spatial association rules in geographic information databases, in 'Advances in Spatial Databases (Proc. 4th Symp. SSD'95)', Portland, ME, pp. 47-66.

Koperski, K., Adhikary, J. & Han, J. (1996), Knowledge Discovery in Spatial Databases: Progress and Challenges, in: Proceedings of Workshop on Research Issues on Data Mining and Knowledge Discovery, Montreal, QB.

Lu, W., J. Han, B. Ooi (1993): Discovery of general knowledge in large spatial databases, in ' Proc. of 1993 Far East Workshop on Geographic Information Systems (FEGIS'93)', Singapore, pp. 275-289.

MacQueen, J. (1967): Some methods for classification and analysis of multivariate observations, in: Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, pp. 281-297.

Michalski, R., J. Carbonell, T. Mitchell (1984): Machine Learning - An Artificial Intelligence Approach, Springer-Verlag, Berlin.

Ng, R., J. Han (1994): Efficient and effective clustering method for spatial data mining, in 'Proc. of 1994 Int. Conf. on Very Large Data Bases (VLDB'94)', Santiago, Chile, pp. 144-155.

Piatetsky-Shapiro, G., W. Frawley (1991): Knowledge Discovery in Databases, AAAI/MIT Press, Menlo Park, CA.

Plazanet, C., Bigolin, N., Ruas, A., (1998), Experiments with Learning Techniques for Spatial Model Enrichment and Line Generalization, GeoInformatica, **2**(4), pp. 315-334.

Preparata, F. & Shamos, M. (1985), Computational Geometry: An Introduction, Springer-Verlag, New York.

Quinlan, J. R. (1984), Learning Efficient Classification Procedures and their Application to Chess End Games, in: R. Michalski, J. Carbonell & T. Mitchell, Hrsg., Machine Learning, Springer-Verlag, Berlin, Seiten 463-482.

Regnauld, N. (1996): Recognition of building clusters for generalization, in M. Kraak M. Molenaar, 'Advances in: GIS Research, Proc. of 7th Int. Symposium on Spatial Data Handling (SDH)', Vol. 2, Faculty of Geod. Engineering, Delft, The Netherlands, P. Session 4B.

Richardson, D. (1996), Automatic processes in database building and subsequent automatic abstractions, Cartographica, Monograph 47, **33**(1), pp 41-54.

Ruas, A., Lagrange, J., (1995), Data and knowledge modeling for generalization, in: J.-C. Müller, J.-P. Lagrange, R. Weibel, eds., GIS and Generalization – Methodology and Practice, Taylor & Francis, pp.73-90.

Sester, M., Anders, K.-H., Walter, V., (1998), Linking Objects of Different Spatial Data Sets by Integration and Aggregation, GeoInformatica, **2**(4), pp. 335-358.

Sester, M., (1999), Knowledge Acquisition for the Automatic Interpretation of Spatial Data, Accepted for Publication in: International Journal of Geographical Information Science.

Sester, M., 2000, Generalization based on least squares adjustment, in: IAPRS, Vol. 33, ISPRS, Amsterdam, Holland.

Shaw, G., Wheeler, D. (1994): Statistical Techniques in Geographical Analysis, David Fulton, London.

van Smaalen, J. (1996), Spatial abstraction based on hierarchical re-classification, Cartographica, Monograph 47, **33**(1), pp 65--74.

Supowit, K., J. (1983), The relative neighborhood graph, with an application to minimum spanning trees, J.Assoc.Comput.Mach, **30**, pp. 428-448.

Toussaint, G.T. (1980), The relative neighborhood graph of a finite planar set, in: Pattern Recognition, Vol. 12, pp. 261-268.

Toussaint, G.T. (1988), A graph-theoretical primal sketch, in: G.T. Toussaint, ed., Computational Morphology, North-Holland, pp. 229-260.

Toussaint, G.T. (1991), Some unsolved problems on proximity graphs, in: D.W. Dearholt and F. Harary, ed., Proceedings of the First Workshop on Proximity Graphs. Memoranda in Computer and Cognitive Science MCCS-91-224, Computing research laboratory, New Mexico State University,La Cruces.