Automatic urbanity cluster detection in street vector databases with a raster-based algorithm

Volker Walter, Steffen Volz
University of Stuttgart
Institute for Photogrammetry
Geschwister-Scholl-Str. 24D
D-70174 Stuttgart
Germany
Email*: firstname.lastname@ifp.uni-stuttgart.de*

Abstract

Clustering spatial data can be interpreted as a segmentation or classification process for finding meaningful patterns in geospatial databases. Such patterns can be used to adapt further algorithms to the individual characteristics of the detected classes or clusters. For example, cartographic generalisation works differently in urban and in rural areas. If this kind of information is not explicitly stored in the database, it can be derived automatically with map interpretation. In this paper it will be discussed, how this interpretation can be done automatically by a computer.

1. Introduction

Geodata contain much implicit information that can be derived from human persons by visual inspection. For example, it is often possible to distinguish between inner city areas, rural areas or industrial areas only by looking at the object geometries without considering any attributes. Further examples are the differentiation between main and side streets, flat and hilly areas, dense and thin populated areas, etc. Furthermore, a human person can easily identify and group objects that belong together (spatial clustering). Even the type of a map (street map, cadastral map, topographic map, etc.) can be identified only by looking at the object geometries. All these examples can be subsumed under the term “map interpretation”.

Automatic map interpretation can be used to support other applications, like automatic map generalisation, matching of spatial datasets, data fusion or data update. Furthermore, automatic map interpretation can support data retrieval processes. In conjunction with search engines and digital globes complete new applications are thinkable. For example, the company Google explores techniques for the automatic indexing of audio files with speech recognition software. With such indexing techniques it would be possible to retrieve audio files automatically in the same way as normal web pages. The same idea can be transferred to digital maps. With automatic indexing techniques it would be possible to assign key words to maps or spatial parts of maps. Furthermore, the support of local searches can be improved. For example, if a user wants to get all maps that contain a golf course with ocean view, the corresponding maps can be found, even if this information is contained only implicitly.

The automatic derivation of unknown information from databases is also known under the term Data Mining or Knowledge Discovery [Frawley et. al. 1991]. Data mining techniques are used to derive unknown information from huge data sets that are not visible for a human person. This applies only partly to this work, because we want to derive information that are very well visible for human persons but which are not modelled and stored explicitly in the database. Automatic map interpretation is a mixture between data mining and automatic image interpretation.

Automatic map interpretation has already been discussed in other works. The different approaches can be differentiated whether they use raster or vector data as an input. An approach for the automatic interpretation of raster maps with query languages can be found for example in [Graeff and Carosio 2002]. The interpretation here is done with pattern recognition algorithms. The detected objects are implicitly contained in raster maps. However, the objects were explicitly modelled when the map was digitized. Therefore the objects are already visible but can not be queried because of the raster representation. A vector-based approach for automatic map interpretation is discussed in [Viglinio and Pierrot-Deseilligny 2003]. The input for this process is also a raster map that is first converted into a vector representation. Different object classes (for example buildings, hangars or parcels) are reconstructed with low level primitive extraction and classification. Vector-based approaches very often are based on techniques from the field of Artificial Intelligence. For example, [Sester 2000] presents an approach for the semi-automatic interpretation of unstructured vector data based on machine learning techniques. A graph-based approach for clustering unstructured point data can be found in [Anders 2001]. To the best of our knowledge, none of the existing techniques has tried to perform the map interpretation of vector data in the raster domain.

2. Approach

In our study we use vector data from the Geographic Data Files (GDF) in order to derive raster-based clusters of different degrees of urbanity. GDF is an international standard for the modelling and exchange of road network data. GDF data are captured in a scale of approximately 1:25.000. An area of 4 square kilometres in the inner city area of Stuttgart in the southern part of Germany was selected that shows different characteristics in terms of urbanity. The test area is shown in Figure 1a.

2.1 Process chain

At the beginning of the process, an operator can define two different parameters for generating the clusters: the grid cell size of the resulting raster map and the radius around the centre of each grid cell (cluster radius), so that the area for which the cluster indicators have to be observed can be calculated (area of influence). After the operator has chosen the clustering parameters, the area is subdivided into equally sized, square-shaped grid cells. Figure 1b) shows the grid cells superimposed on the vector data.

Then, using the centre point of each cell, the area of influence is determined and the indicators are calculated for each grid cell. The result is a raster layer for each indicator. As indicators for recognizing different levels of urbanity we used *node density* and *rectangularity* of streets, since we assumed that (at least in Germany) in city centres are more topological nodes and more irregular, non-orthogonal streets than in suburbs or rural areas. Figure 1c) shows the result of calculating the rectangularity. Dark grey values stand for high rectangularity and bright grey values for low rectangularity. Figure 1d) shows the result of the calculation of the node density. Bright grey values indicate high node density whereas dark grey values indicate low node density.

In the next step, the grey value matrices of each indicator are categorized with thresholds into three classes:

$$\forall_n : Class_{Indicator_n} = \begin{cases} 1 \text{ if } value\_{Indicator_n} < tresholdlow \\ 3 \text{ if } value\_{Indicator_n} > tresholdhigh \\ 2 \text{ else} \end{cases} \tag{1.1}$$

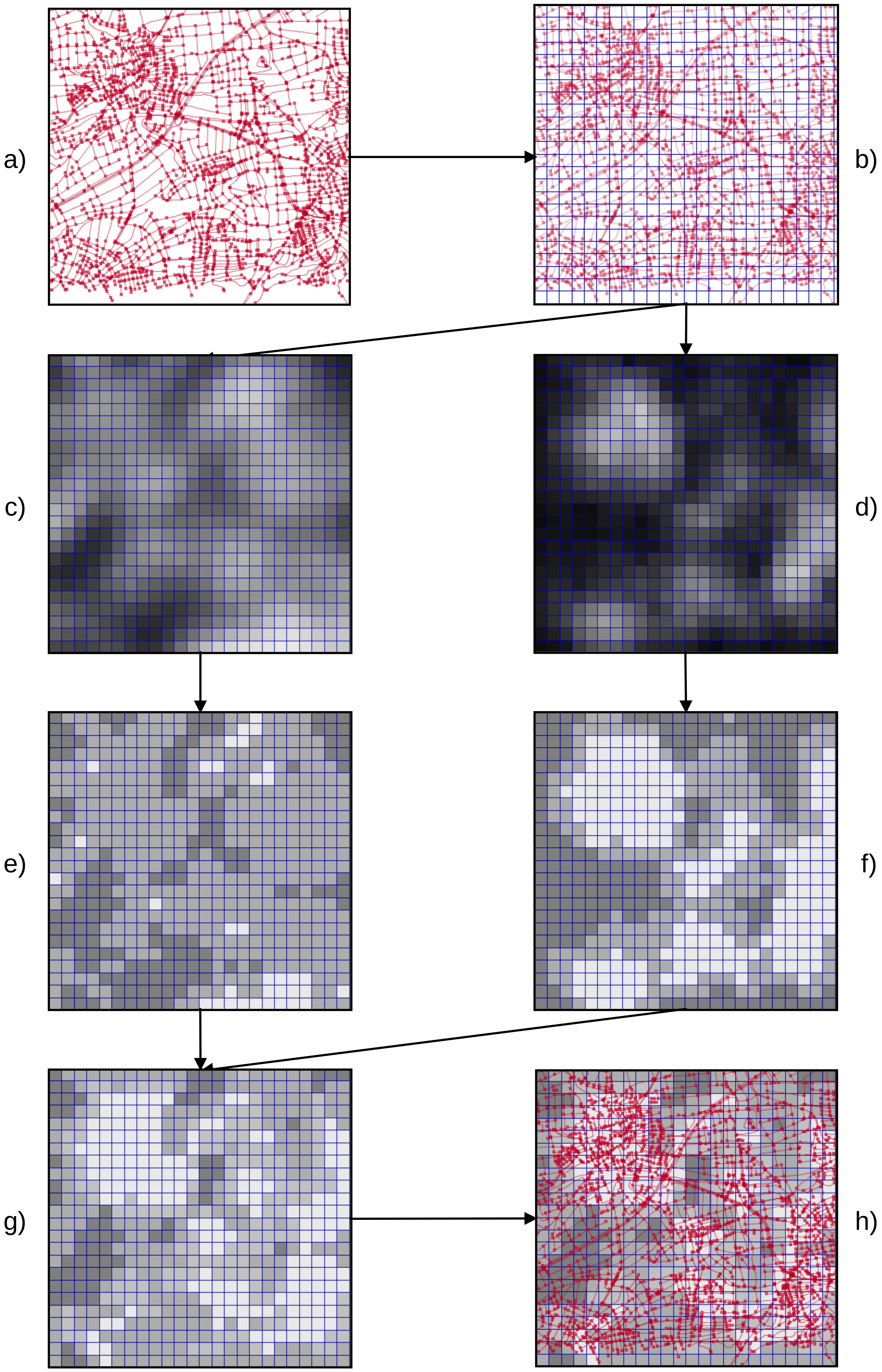


Figure 1: Process chain: a) input data b) grid cells c) rectangularity d) node density e) categorisation of rectangularity f) categorisation of node density g) combination of the measures h) final result

The result of this categorisation is shown in Figure 1e) for the rectangularity and in Figure 1f) for the node density. Bright grey values stand for high urbanity and dark grey values for low urbanity. The evaluation of the point density leads to very good results whereas the result of the evaluation of the rectangularity is more diffuse. The reason for this is that the assumption, that in inner-city areas more irregular, non-orthogonal roads can be found, is only partly true. In fact, in some inner-city areas we can find large areas with orthogonal roads and in some rural areas we can find large areas with non-rectangular roads. However, we use this layer for the further cluster detection, in order to show how different indicators can be combined very easily. These indicators must not come necessarily from the evaluation of grid cells. For example they also can be the result of a multispectral classification or from a GIS layer.

In order to join the different layers and to achieve a final categorization of each individual grid cell, a function has to be defined enabling the combination of the different raster layers. This can easily be done with the weighted sum of all class indicators for each grid cell:

$$Final\_Categorisation = \sum_{n} w_n \, Class\_Indicator_n \qquad (1.2)$$

In the example in Figure 1 we combined the two layers with w_1 = 0.5 and w_2 = 0.5. Since we have three classes for each input layer, the result is a final layer with five different classes, which is represented in Figure 1g). Figure 1h) shows the final result superimposed with the vector input data.

2.2 Calculation of the indicators

The calculation of the node density indicator can easily be done by counting all topological nodes which are within the cluster radius. The calculation of the rectangularity indicator is somewhat more difficult to obtain. For the calculation we only use nodes with more than two incident edges that are within the cluster radius. Nodes with only two incident edges do not represent intersections. Therefore we do not use them for the calculation of the rectangularity. Typically nodes like that are used to model an attribute change of a road element between two intersections (for example change of the maximum speed).

For nodes with more than two incident edges we calculate the *n* - 1 smallest angles between these incident edges. The values of these angles are then normalized onto an interval between 0 and 90:

$$normalized\_angle = \begin{cases} angle & \text{if } angle \text{ between}(0,90) \\ 180 - angle & \text{if } angle \text{ between}(90,180) \end{cases} \tag{1.3}$$

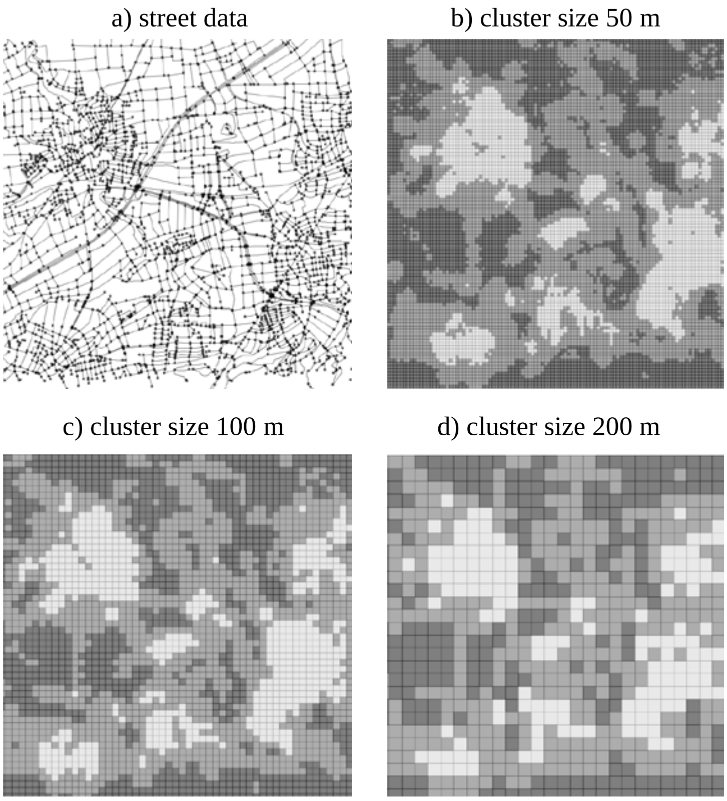
Angles that are larger than 180 degrees must not be considered because we evaluate only the *n* – 1 smallest angles between the *n* incident edges of a node. Finally, the rectangularity of each node is calculated by the arithmetic mean of all normalized angles. The closer this value is to 90, the higher is the degree of rectangularity of the corresponding node. The rectangularity indicator is finally calculated by the arithmetic mean of the rectangularity of all nodes which are within the cluster radius.

3 Results

In this section it is shown on examples how the cluster size and the cluster radius influence the classification result. In the following examples we use only the node density indicator for the classification because it leads to clearer results as the combination of the node density and the rectangularity indicator as already discussed above.

Figure 2 shows the influence of the cluster size to the classification result. Figure 2a) shows the vector input data and Figure 2b) to 2d) show the classification results with cluster size 50m, 100m and 150m. In all examples the cluster radius is 150m and the thresholds for the categorisation are *threshold_low* = 5 and *threshold_high* = 20. The cluster size has only an effect to the resolution of the result but not to the segmentation into different clusters. Therefore this is a very robust parameter that can be set to a wide range of values and does not require a carefully selection.

Figure 3 shows the influence of the cluster radius to the classification result. Figure 3b) to 3d) show the classification results with cluster radius 150m, 300m and 450m. In all examples the cluster size is 100m. The thresholds for the categorisation have to be adapted to the different cluster radii because the number of topological nodes in a small area of influence is naturally smaller as in a large area of influence. Therefore we adapt the thresholds depending on the size of the area of influence and round them to the next integer value, as it is shown in Table 1.



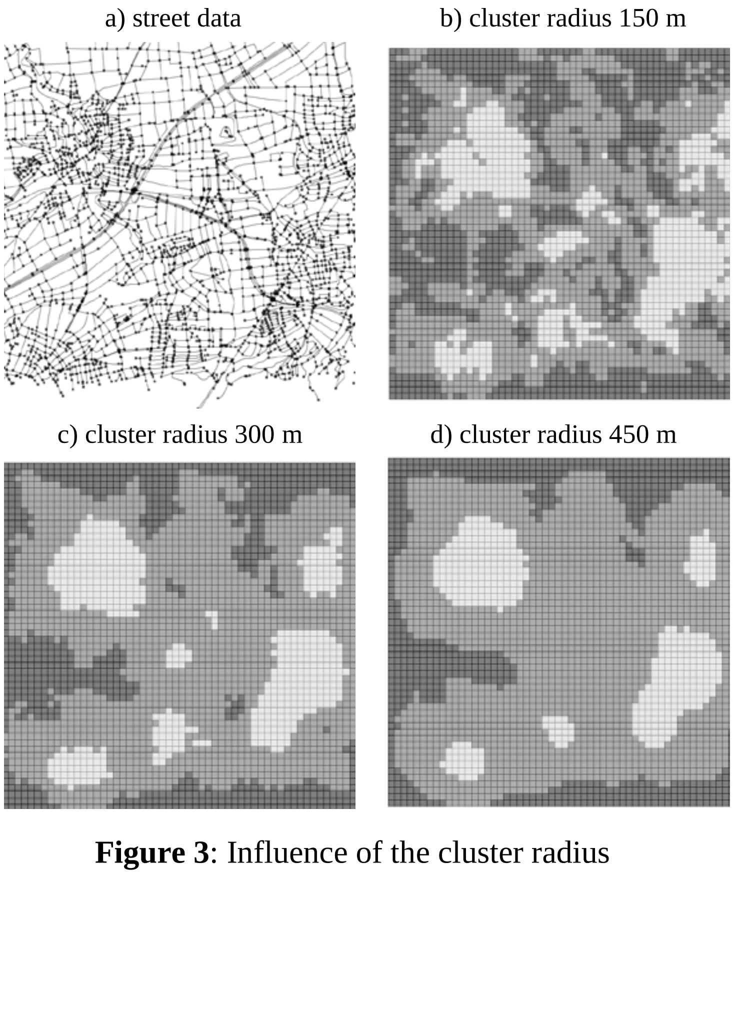
a) street data

b) cluster size 50 m

c) cluster size 100 m

d) cluster size 200 m

Figure 2: Influence of the cluster size



a) street data

b) cluster radius 150 m

c) cluster radius 300 m

d) cluster radius 450 m

Figure 3: Influence of the cluster radius

cluster_ radius	*area of influence =* p * *cluster_radius*^{2}	*threshold_low*	*threshold_high*
150	*70685*	*5*	*20*
300	282743 ≈ **4** * *70685*	20 = **4** * *5*	80 = **4** * *20*
450	636172 ≈ **9** * *70685*	45 = **9** * *5*	180 = **9** * *20*

Table 1: Adapting of the thresholds for different cluster radii

The adaptation of the thresholds has only to be done by indicators which are sensitive to the area of influence. For example the rectangularity indicator is always a value between 0 and 90 and therefore must not be adapted to different cluster radii.

Increasing the cluster radius leads to a smoothing of the result. Therefore the cluster radius influences the scale of the result. The larger the cluster radius the smaller is the scale. Another effect is that if the cluster radius is much larger than the cluster size, the form of the clusters become more roundly, because the area of influence is calculated as a circle around the centre of each grid cell. Again, this is robust parameter and does also not require a carefully selection.

The influence of the thresholds to the classification result is shown in Figure 4. The optimal thresholds were determined by a visual inspection of different test runs. Figure 4b) shows the classification result with optimal thresholds, Figure 4c) with higher thresholds and Figure 4d) with lower thresholds. A small increase or decrease of these thresholds has already significant effects to the result of the classification. Therefore the thresholds are the most sensitive parameters in the classification process.

5 Discussion

The proposed algorithm is a simple and straightforward approach that allows the fast computing of clusters. For the cluster detection we examined the use of two indicators: node density and rectangularity. The evaluation of the indicator node density leads already to very good results whereas the indicator rectangularity is not useable in all areas. But rectangularity is an indicator which is important for other applications like for example automatic generalisation approaches. The determination of the thresholds for the categorisation has to be done carefully, because these are the most sensitive parameters. Alternatively, these parameters can be determined semi-automatically with a supervised classification. In contrast the other two parameters node density and rectangularity are very robust.

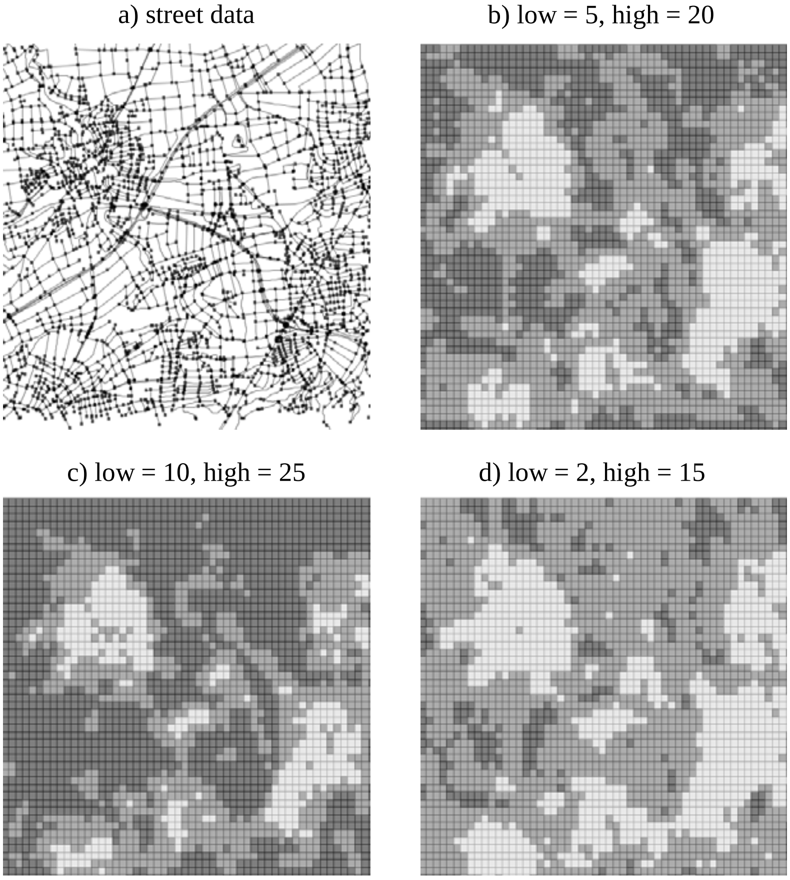


Figure 4: Influence of the thresholds

The proposed approach is very powerful and can be implemented very easily. In order to segment vector data regarding other characteristics, other indicators can be defined and combined with the same techniques as described in this approach.

6 Literature

Anders, K.H. (2001): Data Mining for Automated GIS Data Collection. In: Fritsch, D. and Spiller, R. (eds): Photogrammetric Week ’01, Wichmann Verlag, Heidelberg, 263 - 272.

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

Graeff, B. and Carosio, A. (2002): Automatic Interpretation of Raster-Based Topographic Maps by Means of Queries. FIG XXII International Congress Washington, D. C., published on CD-ROM, 12 pages.

Sester, M. (2000): Knowledge acquisition for the automatic interpretation of spatial data. International Journal for Geographical Information Science, Vol. 14, No. 1, 1 – 24.

Viglinio, J.-M. and Pierrot-Deseilligny, M. (2003): A Vector Approach for Automatic Interpretation of the French Cadastral Map. In: Proceedings of the Seventeenth International Conference on Document Analysis and Recognition (ICDAR ’03), published on CD-ROM, 5 pages.