Evaluation Method for an Automatic Map Interpretation System for Cadastral Maps

Rik D.T. Janssen Robert P.W. Duin Albert M. Vossepoel

Delft University of Technology
Faculty of Applied Physics, Pattern Recognition Group
Lorentzweg 1, 2628 CJ Delft, the Netherlands
email: janssen@ph.tn.tudelft.nl

Abstract

This paper discusses a base line system for automatic interpretation of Dutch Cadastral maps. Knowledge of the rules for drawing these maps is incorporated in the system. Also, a method for evaluating this system is presented. This method consists of different parts which subsequently evaluate the vectorization, the performance on finding parcels, and on finding parcel numbers. Parts of this evaluation method may also be applicable for other line drawing interpretation systems.

Keywords: Automatic Map Interpretation, Evaluation, Vectorization.

1 Introduction

Automatic map interpretation is a relatively new field in image processing [1, 2]. Until recently, maps like cadastral maps were mainly drawn by hand on paper or film, and then stored in large cabinets. The disadvantage of this compared to computer accessible maps are the large costs for drawing, storing and maintaining them, and their fragility since paper maps are easily torn. Also, after some updates a completely new map must be made to incorporate these changes.

By contrast, computer stored maps are readily accessible, changes are very easy to incorporate, and they can be accessed from different locations using graphic terminals. Information can be processed using e.g. Geographic Information Systems (GIS).

Nowadays, new maps are often made with the computer. Old maps (drawn on paper or film) can be acquired by scanning them and storing them on disk or tape. However, for meaningful queries some kind of interpretation is necessary. This can be done in several ways, of which human interpretation is the most obvious, and, because of the time required, also the most expensive.

In this paper, a base line system for cadastral map interpretation (BMIS) is presented. Knowledge of rules cartographers use for drawing is incorporated in the system. This high level knowledge enables us to anticipate on the objects to recognize, and to improve and correct the interpretation if necessary. This is different from the approach followed in e.g. the map recognition system of [1], in which a human operator is needed for improvement of the vectorization generated by the computer.

Another contribution of this paper is an evaluation method for the BMIS presented, of which parts may be useful for other line drawing interpretation systems. In evaluation, the emphasis is on vectorization because we believe this is an essential first step in interpretation of line drawings. Vectorization gives a large reduction in storage and processing time of data. Its result is scale independent and allows easy computation of e.g. intersection points or angles between lines.

Research as described in this paper is the base line system in one of the on-going projects in Model Based Image Processing in the Netherlands. Interpretation of cadastral maps is one of the applications chosen, because maps are made with a model and cadastral maps are relatively simple compared to other kinds of maps.

The model cartographers have of a map consists of 'expectations', and these expectations guide the interpretation. Conflicts between reality and the model should be easier to detect and to correct because the model specifies what is allowed and what not. Also, switching to different kinds of maps or line drawings should be easier because only the model has to be changed (and often only a small part of it). The research concentrates on the formulation and implementation of such a model based processing methodology.

In the next sections, first a description of the BMIS is given, followed by our evaluation strategy. Maps used, results and a discussion of the results are reported. The paper is concluded with a summary and conclusion. For image processing terms like skeleton, propagation, etc., the reader is referred to a general image processing textbook (e.g. [3]).

2 The base line map interpretation system

The system is depicted in figure 1. In the following paragraphs the different parts are described.

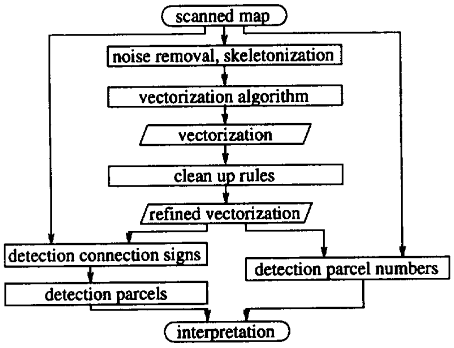
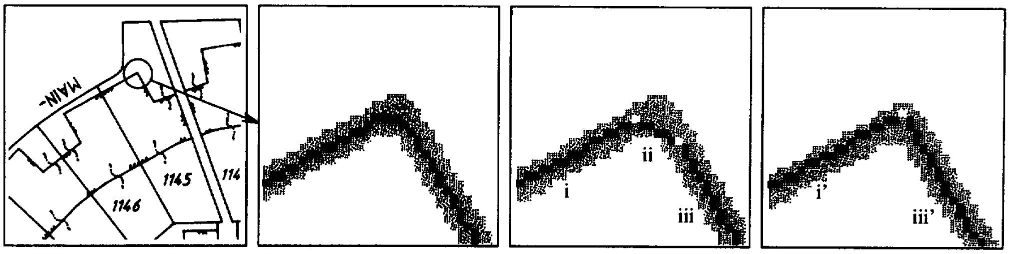


Figure 1: the base line map interpretation system (BMIS).



(a) (b) (c) (d)

Figure 2: (a) a small part of map A. (b), (c) and (d): the effect of the corner clean-up rule. For explanation see the text.

Input. Input are cadastral maps, scanned with high resolution, high quality binary scanners. We call these scanned maps ***bitmaps***. For examples, see figures 2(a) and 3.

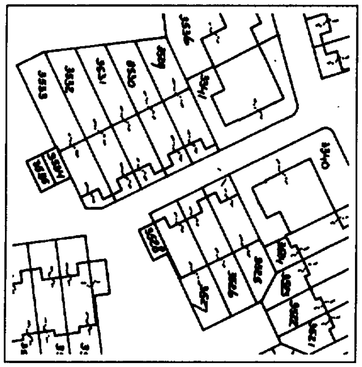


Figure 3: a small part of map B.

On cadastral maps *parcels* are drawn. A parcel is owned by one (legal) person and has one *parcel number* (no more, no less). The cadastral map specifies where this area can be found in the real world. The number, which is usually in the middle of one of the polygons of the parcels, is used for reference. Different polygons which belong to the same parcel are connected using *connection signs* (the small dotted S-like shapes).

Output. The output of the system are parcel descriptions consisting of nodes, edges and numbers.

Noise removal and skeletonization. Because our scanned maps are almost noise free, a median filter suffices to close small irregularities in the bitmap caused by e.g. missing pixels. Small gaps caused by e.g. a non-continuous flow of ink from the drawing pen will be closed by our vectorization algorithm.

Then, a pseudo Euclidean skeleton is used [4]. This is an improvement of the Hilditch skeleton [5] in which a better (almost Euclidean) metric is used. This ensures that the skeletonization is less sensitive to rotations compared with the Hilditch skeleton.

Vectorization algorithm. Following the skeletonization, the resulting image is vectorized using a modified version of L.R. Moore's CARV (Cartographic Raster-to-Vector) software described in [6]. He uses the well-known Douglas-Peucker vectorization algorithm [7]. The results of the vectorization (*nodes* and *edges*) are stored in a special Graph Representation Datastructure.

Clean up rules for the vectorization. Due to the use of skeletons, the vectorization has short unnecessary or spurious edges. Rules were designed to refine the vectorization, e.g. to remove nodes whose two edges make an angle of almost 180 degrees, to correct corners in the vectorization, and to repair T-junctions which will often consist of two or three close lying nodes. The results of these refinements are a reduction in the number of nodes and edges, to the cost of a small increase in vectorization error (see the 'results and discussion' section).

An example of the corner clean up rule is depicted in figure 2. In figure 2(b), the skeletonization of a very small part of the original image is plotted in black on it. In (c), the resulting vectorization is shown. Nodes in the vectorization are plotted white, the edges black. As can be seen, it consists of three edges (i), (ii) and (iii) instead of the desired two (edge (ii) is unnecessary). By setting constraints on the angles of the edges involved and on the length of edge (ii), we are able to correct the vectorization as shown in figure 2(d).

Detection of the connection signs. Connection signs consist of small blobs which are located near to each other (cf. figures 2(a) and 3). If we dilate the original image, the connection signs will become connected to the other lines, and after skeletonization the short lines are part of the connection signs. Next, they can be isolated.

Detection of the parcels. The parcels are detected with a propagation of the connection signs in the original bitmap.

Detection of the parcel numbers. As can be seen in figure 1, the parcels are not used for detecting the parcel numbers. Doing this, the parcel numbers found may be used for correcting errors in finding the parcels, or vice versa. This has not been implemented yet.

To find the parcel numbers, only the small objects in the bitmap (i.e. numbers and characters) are AND-ed with the street image to remove the street names. The parcel numbers and the street names can be sent to a character recognition system.

3 Evaluation strategy

The system described in the previous section is evaluated in different ways, described in this section. Parts of this method may also be useful for other line drawing interpretation systems.

3.1 Evaluating the vectorization

Because in our opinion the vectorization is very important for both data reduction and for reasoning, it should be checked thoroughly. To see the effects of the clean up rules both the raw (not cleaned) vectorization and the refined result are evaluated. To prevent border-effects all edges which have an endpoint are removed.

map	# pixels bitmap	raw, # pixels in vect	wrong	%	refined, # pixels in vect	wrong	%
A	364 198	45 020	36	0.08	45 351	368	0.81
B	571 297	89 709	69	0.08	90 574	1119	1.24

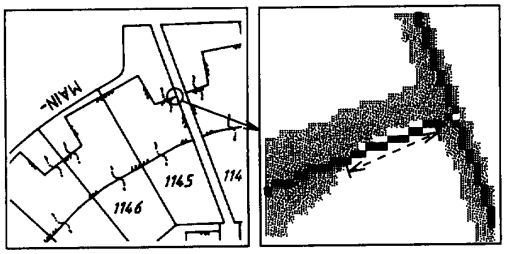
Table 1: the number of pixels wrong in the vectorization.

Only polygons will remain. Desired are both long edges and high accuracy of the vectorization.

At the base of the scoring mechanism is our *vectorization evaluation criterion*:

> We consider the vectorization satisfactory if, when it is plotted in the original map, the original map overlaps the plotted vectorization.

Examples of this can be seen in the figures 2(c), 2(d) and 4(b), where the original bitmap is displayed in grey and the plotted vectorization in black. According to our criterion, figures 2(c) and (d) do not have errors, while the error in figure 4(b) is next to the dashed arrow.



(a) (b)

Figure 4: an example of a wrong vectorization, next to the dashed arrow.

We report on the number of nodes and edges in the vectorization, and the number of pixels wrong and the number of lines wrong.

Number of pixels wrong. This is the number of pixels in the plotted vectorization which is not covered by the original bitmap, relative to the total number of pixels in the plotted vectorization. In figure 4(b), there are 11 pixels wrong.

Number of lines wrong. For this, we form lines from the connected wrong pixels and count the number of lines of each length, and the size of these lines in pixels (these lines are always part of an edge). This is given relative to the total number of edges in the vectorization. Note that due to this scoring method, some lines may be counted twice because they cross the bitmap. In figure 4(b), there is one line wrong with a size of 11 pixels.

3.2 Evaluating the interpretation

Parcels. The correctly found parcels are given relative to the number of parcels completely in the field of view (this last count is called the total number of parcels). Some parcels consist of more than one polygon. For counting a parcel as correctly found, all its polygons have to be found.

Parcel numbers. Only the parcel numbers of the parcels completely in the field of view are evaluated. Computed is this total number of parcel numbers minus the parcel numbers which are found at positions where they should not be (insertions, false positives), minus the parcel numbers which are not found where they should be (deletions, false negatives). The remainder is given relative to the total number of parcel numbers.

4 Results and discussion

4.1 Description of the maps used

Map A. For evaluation, two different maps were used. From map A (resolution 508 dpi, 4935 × 6601 pixels), a part of size 1970 × 2400 pixels is used for evaluation. Of this part, a much smaller part of 700 × 700 pixels (shown in figure 2(a)) was used for development. The map is machine drawn, which is reflected in the small variability of the characters.

Map B. The second map is map X1 of Leiden, the Netherlands, on scale 1:1000. This map has been drawn by hand in 1955. From this map (resolution 400 dpi, 10584 × 15957 pixels), a part of size 2800 × 2700 pixels is used for evaluation. This map has not been used for development, so evaluation gives an indication of the performance of our algorithm on new cadastral maps. A small part of this map (1000 × 1000 pixels) can be seen in figure 3.

4.2 Evaluation of the base line map interpretation system

Evaluation of the vectorization. The number of pixels wrong is shown in table 1. Columns 2 to 5 indicate respectively the number of pixels in the bitmap, number of pixels of the raw (not cleaned) vectorization, number of pixels which are wrong according to our criterion, and the percentage of pixels wrong. Columns 6 to 8 are the same as columns 3 to 5 but then for the refined vectorization.

As can be seen, the error increases after refinement of the vectorization. This is caused for instance by cut-off corners or curves where many short edges are substituted by one or two longer ones. The benefit of the refinements is mainly a substantial reduction in the number of nodes and edges (see table 2), at the cost of a small increase in error.

map	raw # nodes	# edges	refined # nodes	# edges
A	1764	1852	1412	1499
B	2400	2692	1653	1945

Table 2: the number of nodes and edges in the vectorization.

The number of lines wrong is given in table 3. Note that because such a line may cross a line of the bitmap, some of these vectorized lines may be counted twice. Subsequently, this table gives an upper bound.

From this table it follows that the lines which are formed by the pixels that are wrong according to our criterion, are generally very short. Of the lines wrong, 90% is shorter than 10 pixels (for map B 10 pixels is approximately 0.6 mm).

Evaluation of the parcels found. Only parcels which are completely in the field of view are evaluated. See table 4. For map B we decided to use two different measurement methods, i.e. method I and method II. This has been done because our development map (figure 2(a)) did not contain parcels without connection signs. Map B, however, did contain 18 of these parcels (see e.g. parcels numbered 3534, 3535 and 3528 in figure 3). Of course, our parcel finding algorithm did not find these. Method I is counting all the parcels, while method II does not count parcels without

map	total # edges	size ≥ 1 #	 %	size ≥ 2 #	 %	size ≥ 3 #	 %	size ≥ 5 #	 %	size ≥ 10 #	 %
A raw	1852	20	1.08	12	0.65	4	0.22	0	0.00	0	0.00
A refined	1499	55	3.67	32	2.13	20	1.33	12	0.80	5	0.33
B raw	2692	41	1.52	14	0.52	7	0.26	2	0.07	0	0.00
B refined	1945	315	16.20	156	8.02	85	4.37	49	2.52	17	0.87

Table 3: the cumulative size distribution (in pixels) of the lines wrong.

map	total #	correct #	 %
A	40	29	72.5
B method I	104	71	68.3
B method II	86	69	80.2

Table 4: the evaluation of the parcels found.

a connection sign.

Errors in finding the parcels have two different causes. The first is that our algorithm sometimes was not able to find a connection sign. Since a connection sign is necessary for finding a parcel, the method fails immediately. The other cause is that propagation is used in finding the edges of the polygons which belong to the same parcel. If the polygon is not closed when it is supposed to (this can be caused e.g. by a digitizing error), the propagation proceeds forward into other parcels, and subsequently causes errors in the detection of these parcels.

Note that the number of correctly found parcels is two less for map B method II compared to method I. This is caused by two parcels without connection signs which were accidently detected correctly (by e.g. erroneously interpreting part of a parcel number or noise for a connection sign).

Evaluation of the parcel numbers found. See table 5. In evaluating, we consider the two methods of the previous paragraph.

map	total #	correct #	ins #	del #	correct %
A	40	40	1	0	97.5
B method I	103	86	3	17	80.6
B method II	85	70	3	15	78.8

Table 5: the evaluation of the parcel numbers found.

For map B, the total number of parcel numbers is one smaller than the number of parcels. This is due to a (human) drawing error in the cadastral map: one parcel does not have a number. In fact, this is a violation of the drawing rules.

Errors in not finding the parcel numbers are caused by parcel numbers attached to the parcel boundary they belong to. The method for finding parcel numbers presumes them being isolated, and it will fail to find them if they are not.

Our algorithm may be improved by isolating single digits followed by location of whole parcel numbers, but this has not been implemented yet.

5 Conclusions and summary

This paper discusses a base line automatic map interpretation system. Also, it presents a method for evaluating the performance of this system. Parts of this method may also be useful in evaluating other line drawing interpretation systems.

Different performance methods are presented: for the vectorization we report on the number of nodes and edges in the vectorization, and we measure the number of pixels wrong, and the size distribution of the lines wrong. Next, the number of correctly classified parcels and correctly classified parcel numbers is reported. We think that the performance of an automatic map interpretation system can be estimated reliably from these parameters.

In the near future, our system will be extended e.g. to allow more context checking. Also, a better integration of top-down and bottom-up processing will be implemented.

6 Acknowledgements

This research was supported by the Foundation for Computer Science in the Netherlands (SION) and the Netherlands Organization for Scientific Research (NWO). The map in figure 3 is reprinted with permission of the Dutch Cadastre. Research has been conducted at the Delft University of Technology and the TNO Institute of Applied Physics, both in Delft, the Netherlands.

References

[1] M. Ejiri, S. Kakumoto, T. Miyatake, S. Shimada, and K. Iwamura. Automatic recognition of engineering drawings and maps. In R. Kasturi and M. M. Trivedi, editors, *Image analysis applications*, chapter 3, pages 73–126. Marcel Dekker Inc., 1990.

[2] S. Suzuki and T. Yamada. MARIS: map recognition input system. *Pattern Recognition*, 23(8):919–933, 1990.

[3] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison Wesley, Reading, MA, 1992.

[4] B.J.H. Verwer. Improved metrics in image processing applied to the Hilditch skeleton. In *Proc. 9th Int. Conf. on Pattern Recognition (Rome, Nov. 14–17, 1988)*, pages 137–142. IEEE Computer Society Press, 1988.

[5] C.J. Hilditch. Linear skeletons from square cupboards. In B. Meltzer and D. Mitch, editors, *Machine Intelligence 4*, chapter 22, pages 403–420. Univ. Press Edinburgh, 1969.

[6] L.R. Moore. Software for cartographic raster-to-vector conversion. In *International Archives of Photogrammetry and Remote Sensing, Proc. 17th ISPRS Congress (Washington, Aug. 2-14, 1992)*, 1992.

[7] D.H. Douglas and T.K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122, December 1973.