SYSTEM APPROACH TO A RASTER-TO-VECTOR CONVERSION: From Research to Commercial System

Eugene BODANSKY
Environmental Systems Research Institute, Inc – ESRI
380 New York St., Redlands, CA 92373-810, USA
ebodansky@esri.com

ABSTRACT

This paper analyzes the main factors that influence the effectiveness of raster-to-vector conversion systems. Since complex systems such as these are more than simply the sum of their parts, we emphasize analysis of the whole system.

The paper shows that currently developing new automatic vectorization methods very often cannot increase the effectiveness of conversion systems. The effectiveness of conversion systems depends to a much greater extent on correct division of tasks between the operator and computer and on the capabilities of the raster and vector editors used for pre- and post-processing. Our analysis revealed some important problems that have escaped scientists' attention. Some of them are developing and analysis of semi-automatic vectorization.

KEY WORDS

Data Capturing, Raster-to-Vector Conversion, Vectorization, GIS.

1. INTRODUCTION

Data capturing from maps, engineering drawings, electrical schematics, and other paper line drawings is the combination of several tasks: manual digitizing, scanning of paper line drawings, heads-up digitizing, binarization and pre-processing of the raster images, automatic and semi-automated vectorization of black and white raster images, post-processing of vector data, text and symbol recognition, and interpretation of vector objects. After conversion, data could be loaded into vector databases where objects of line drawings have to be represented as points, lines, and polygons. This task is very labor-intensive and time-consuming.

Much research has been done in the field of data capturing and many articles are dedicated to text and symbol recognition, automatic vectorization, segmentation, compression, and smoothing the result vectors, geometric shapes recognition, measuring of effectiveness of vectorization, and so on.

A number of papers [1-6] have analyzed the data capture process and different raster-to-vector conversion systems.

[1] discusses the data capturing process and briefly mentions heads-up digitizing, automatic (batch mode), and semi-automated (tracing) vectorization, together with the analysis of the pro and contra arguments.

Papers [2] and [3] are dedicated to the specific conversion systems.

In [2] the authors describe the system that was intended for conversion and interpretation of land register maps, which satisfy the Italian Land Register Authority standards. The standards simplify the problems of vectorization and interpretation of raster objects because they define the guidelines for drawings and rules that form a graphic language and restrict the objects shown on these maps.

In this paper, the authors suggest a new algorithm of vectorization that is based on the processing of raster images in RLE format. To simplify vectorization raster object recognition is executed before vectorization.

They suggest that the efficiency of conversion systems or the conversion systems' performance be assessed on the basis of average elapsed time needed to process maps. This suggestion is in line with the opinion of Adrien Litton who writes in [1]: "Your goal is to produce the highest quality vectors in the shortest amount of time."

The general architecture of one commercially available CAD conversion system, GTX, is described in [3]. The authors discuss not only vectorization algorithms but also pre- and post-processing. They emphasize the importance of the operator by showing that automated CAD conversion is never a completely automatic process. At a minimum, a human must check the results of the conversion. In the worst case, the operator must spend hours correcting the revealed errors. In addition to automatic vectorization, the system suggests semi-automated vectorization (tracing).

There is an analysis of the conversion process in [4]. The author writes about the significance of semantic

interpretation of the result vector objects. The last step of the conversion of components is to use one's knowledge about the type of document to assign a semantic label. The result is a description of a document, as a human would give it. The semantic description makes easier for the operator to control the result and to correct revealed errors but usually itself requires the interference of the operator.

The most common opinion is that conversion systems can be made more efficient only by using additional information (gray scale raster images, libraries of used symbols, and so on) and by using more complex algorithms for automation of the vectorization process.

Only [6] emphasizes the problem of correct distribution of tasks between operator and computer. The authors believe that it can make conversion systems more efficient.

For more than a quarter of a century, there has been research and development of the problem of automatic vectorization all over the world. The main goal of these efforts is to accelerate, reduce the costs, and improve the quality of conversion. Nevertheless, manual digitizing from paper maps and scanning, with further heads-up digitizing, has been used up to the present day. In [7], the authors state that automatic vectorization is the easiest and quickest method of data conversion if a source document is in a very good condition. If not, it will require a lot of editing afterwards to reveal and correct all errors. Time spent on editing such a map could easily exceed time spent on document conversion done by some other method. The authors conclude that manual digitizing from analog maps "is the most basic method of digitizing traditional paper maps."

In this paper, we explain why we do not believe that the new methods of automatic vectorization could significantly increase the efficiency of conversion systems and show how to achieve a significant increase in efficiency.

2. VECTORIZATION

We begin our investigation with the problem of vectorization because it is a mandatory task of the conversion process.

Is it possible to strictly define what is the correct vector description of the raster image and what has to be the result of vectorization? Unfortunately the answer has to be "no," if the trivial cases (horizontal and vertical lines of constant thickness) are to be excluded. Strict and objective opinion about this matter does not exist, and only the user can judge whether the result of vectorization is correct. We illustrate this point with several examples.

Example 1. Figures 1a and 1b show the results of vectorizing the same fragment of a contour map. These vectors and width of the corresponding linear objects, which are usually calculated during vectorization, can be used to restore the source raster image. Which result is better? Is it the result that allows one to restore the raster object? The answer is no.

Any cartographer will tell you that both solutions are incorrect. Contours cannot intersect and intersections in this image can be explained only by scanning error and the noise of the raster image.

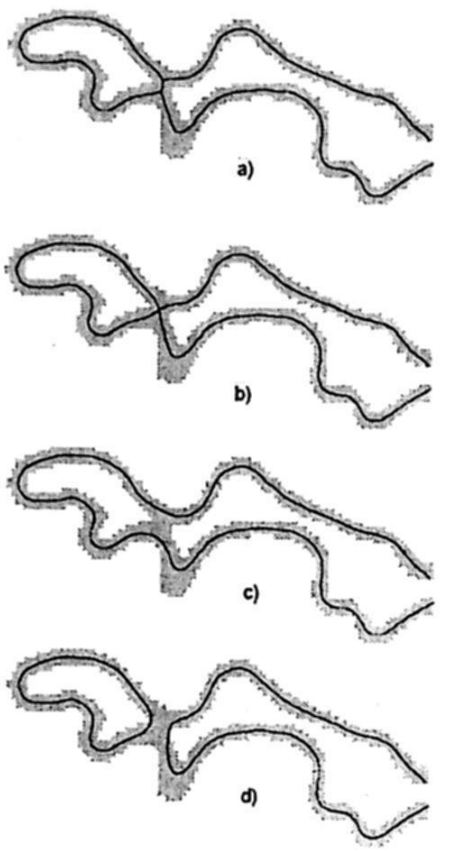


Figure 1. Results of vectorizing a fragment of a contour map.

Correct results are shown on Figure 1c or 1d. To select the correct result, it is necessary to analyze the source line drawing and to study the context. The opinion of the cartographer will be the most important.

Figure [2] shows three different results of vectorizing a fragment of a map. Which one is correct? If we do not know that these are roads, we might think that it's 2a. If we know that these are roads, then the correct result is 2b, if each line represents one side of the road, or 2c, if double lines were used as the line symbol to represent roads.

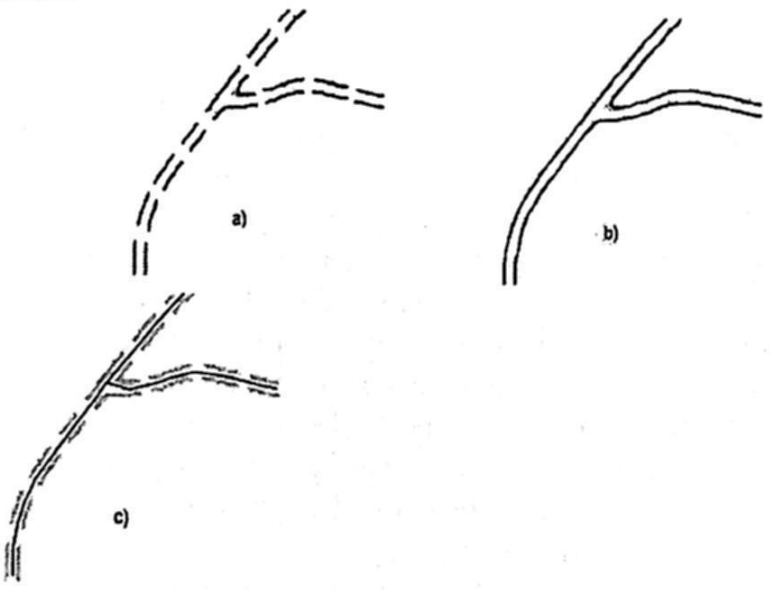
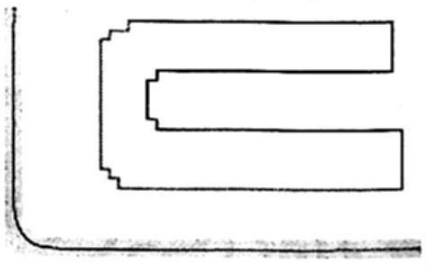


Figure 2. Results of vectorizing a fragment of a road map.

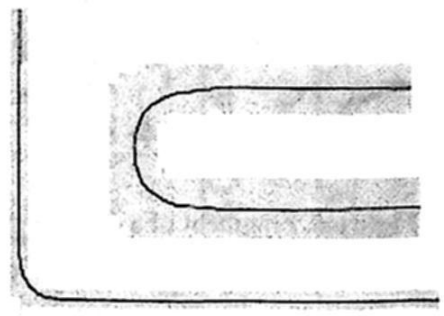
It is possible to find the correct result only if the entire context is taken into account.

Figures 3a and 3b show the results of the vectorizing a fragment of a city map. If the map shows a building, then the correct result of the vectorization is represented by Figure 3a, namely the centerline of the linear object and the border of the building interpreted as a solid. If the map shows two linear objects, the correct result is represented by Figure 3b.

Figure 4 shows a map consisting of several thematic layers. Each thematic layer may require a different vectorization. Texts must be recognized, roads vectorized, buildings outlined. If it is not possible to get hold of the separates, the map should be divided into several sub-maps, each containing one thematic layer, before using automatic vectorization. Dividing a map into thematic layers is complicated, and in the most complex cases, the algorithms that would provide a stable solution in the automatic mode have not yet been found. Sometimes in order to use automatic vectorization methods, the layers are manually copied onto Mylar.



a)



b)

Figure 3. A fragment of a city plan

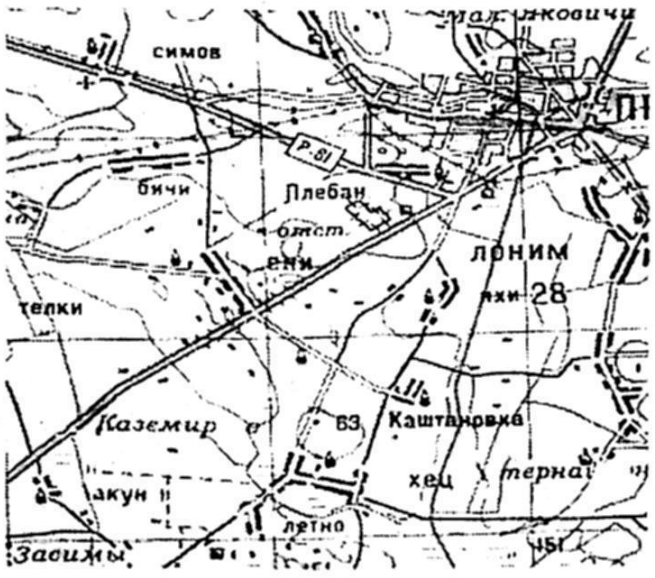


Figure 4. A fragment of a map with several thematic layers

In practice, it is extremely difficult to obtain the correct result by full automatic vectorization of documents. This difficulty stems, in large part, from the fact that the desired solution depends heavily on the following factors:

- noise level
- what linear symbols are used
- quantity and meaning of the thematic layers
- subject area

3. THE EFFICIENCY OF THE CONVERSION SYSTEM

As stated in article [2], efficiency should be measured by the amount of time it takes the operator to convert a document, provided the result satisfies the user's requirements. The cost of this time is equivalent to the cost of labor alone, because if the system is used intensively enough, the cost of the system itself and the cost of the operator's training can be disregarded.

The efficiency of a conversion system depends on what vectorization method is used. If head-up digitizing or manual digitizing is used, vectorization takes up a lot of time, the most of the time of the conversion process. This is why so much emphasis is put on the developing algorithms for automatic vectorization.

There were developed a lot of different methods of automatic vectorization [2, 8, 12, 16,17]. In the references we list only some of the articles dedicated to this problem. Many of them show good results while vectorizing black and white raster images of simple line drawings of good quality. Companies that develop commercial conversion systems also have implemented some good automatic vectorization algorithms.

In addition to errors caused by the imperfections of automatic vectorization algorithms, there are errors caused by noise and other factors listed above. That is why even with good methods of automatic vectorization it is practically impossible to completely avoid errors in automatic vectorization.

Many attempts have been made to compare different systems of automatic vectorization. Some of them offer a quantitative estimation of vectorization errors [8-11], such as an average deviation from the model lines, the number of extra vertices, and the length of incorrectly recognized straight segments, stroke lines, or arcs. It may be useful for the comparison of vectorization algorithms. This estimation is intended for evaluation of the amount of time the operator would have to spend on monitoring and correcting errors. However, this time is affected not only by vectorization errors, but also by the capabilities of the raster and vector editors, by tools that the operator has, as well as by what type of document is being operated on. Here are two simple examples.

Example 1. Because of the noises and errors of the raw vectorization, topological errors may appear. It is difficult to detect them by sight, and eliminating them manually is a difficult task. If a vector editor can

automatically detect and correct such errors, these errors will have little influence on the efficiency of the system.

Example 2. Engineering drawings and electrical schematics contain many straight lines and circle arcs, while contour or hydrographic maps consist mainly of free curves. Without special tools, correcting errors made during vectorization of free curves is more difficult than correcting straight lines.

The fact that manual and head-up digitizing are still being widely used, suggests that frequently the time spent on editing after automatic vectorization is commensurate with the time spent on digitizing.

Our experience with just some algorithms of automatic vectorization leads us to conclude that vectorization of linear objects of constant thickness is quite accurate and in most cases satisfies the user's requirements. The large number of post-vectorization errors and the considerable amount of time necessary for their correction result not from the low geometrical quality of the automatic vectorization but from the complexity of the converted documents and the poor quality of the raster images (see, e.g., figs. 4 and 5).

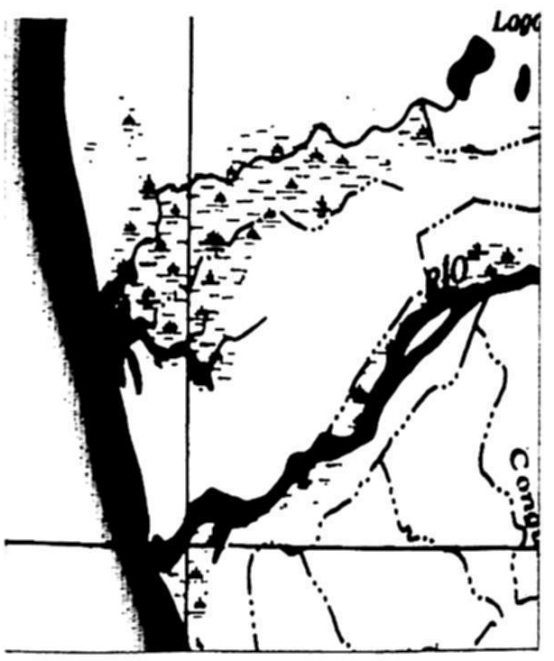


Figure 5. Fragment of a hydrological map

Consequently, in order to increase the efficiency of the conversion system, vectorization methods are needed occupy the middle ground between digitizing and automatic vectorization. Such methods would reduce the number of errors caused by the complexity of the converted documents and their bad quality, as well as mistakes by the operator. Semi-automatic vectorization methods would spare the operator from performing monotonous, labor-intensive routine work and would use the operator only when human judgment is necessary. In addition, these methods would allow one to implement selective vectorization (separate objects or layers) and, in especially complex cases, very easy to switch to heads-up digitizing.

4. INTERACTIVE VECTORIZATION METHODS

Currently, there are two interactive vectorization methods: tracing and raster snapping. Operators who use tracing have to only select a starting point and the direction of the tracing and then the vectorization software will trace the linear object until it comes to an intersection. That is much easier than digitizing each vertex. In addition, it yields computer-generated vectors along the lines, which are usually of higher quality than manual ones [1].

Tracing and raster snapping are indispensable tools for selective digitizing. But they are used for digitizing full documents also.

The fragment of a contour map, shown on Figure6, has more than ten intersections, one of them with six intersecting lines. But contours cannot intersect.

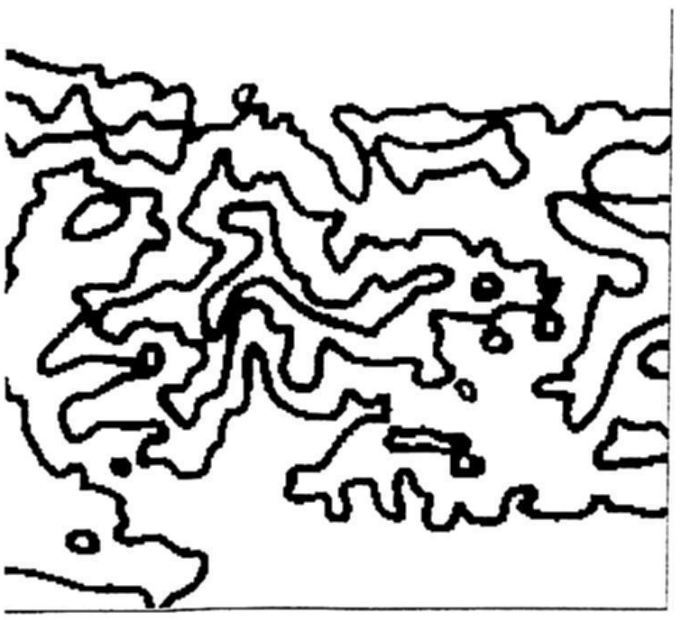


Figure 6. Fragment of a contour map.

What is simpler, to vectorize this fragment automatically and correct all the intersections afterwards, or to use tracing so that the operator could use head up digitizing at the intersections? The answer will depend on the extent of the automatization of the intersection correction procedure in a given system.

Amazingly, it is difficult to find research dedicated to interactive vectorization methods like tracing or raster snapping. If you look for the word "tracing" on the Internet you will receive tons of links to numerous commercial companies, and almost none scientific publications.

Is this so because tracing does not involve any major theoretical problems? No. Some automatic vectorization algorithms essentially perform tracing, that is, they build a vector description of each linear object by moving along this object. The algorithm Sparse Pixel Vectorization is an example of this kind [8]. Nevertheless, in order to achieve a full-fledged, truly interactive vectorization regime, it is first necessary to solve a number of problems that have not yet been adequately addressed. Among these problems are elimination of edge effects; on-line recognition of linear objects, intersections, solids, and

ends; the influence of the starting point on the final solution; smoothing and compression of simultaneously changing lines, and the influence of thresholds.

During tracing it is not important how long it takes vectorization of the full document, but tracing each linear object has to be done in real time. There is no difference if the operator waits for the result 0.1 sec or 0.001 sec. That's why some algorithms and methods that are too time-consuming for automatic vectorization of the whole document, can be successfully used for tracing. It is possible to accelerate tracing with localization of the segment of the image where the vector solution is being built. Segmentation can be performed in a variety of ways, by dividing the image into the tiles of fixed size, dynamic dividing into overlapping areas, selecting parts of the vectorized connected component, etc. However the parts of the image located beyond the borders of the segments can influent on the result vectors. The edge effects have to be suppressed.

Actions of the operator may be required too frequently because of noises. The horizontal line on the Figure7 has more than ten intersections because of connections with digits and symbols. So tracing that ignores intersections may prove extremely useful. Maybe it is possible to develop the program that automatically builds a straight segment or an arc that starts at a given point and extends in a given direction?

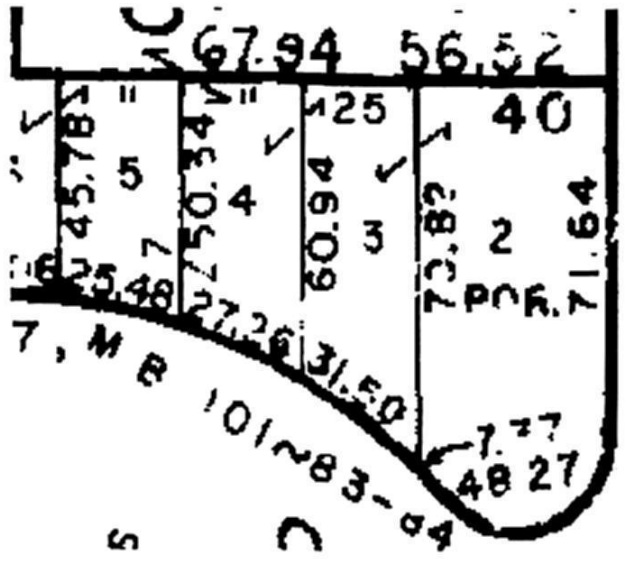


Figure 7. Fragment of a parcel map

Tracing can be used with raster snapping that automatically places a cursor on the nearest point of either centerline, end, intersection, corner, or solid center. This possibility saves a lot of the operator time.

Raster snapping has an independent value when one has to perform vectorization of documents with many straight segments, that contain intensive noise or other layers, making automatic vectorization or tracing too difficult because of the large number of intersections.

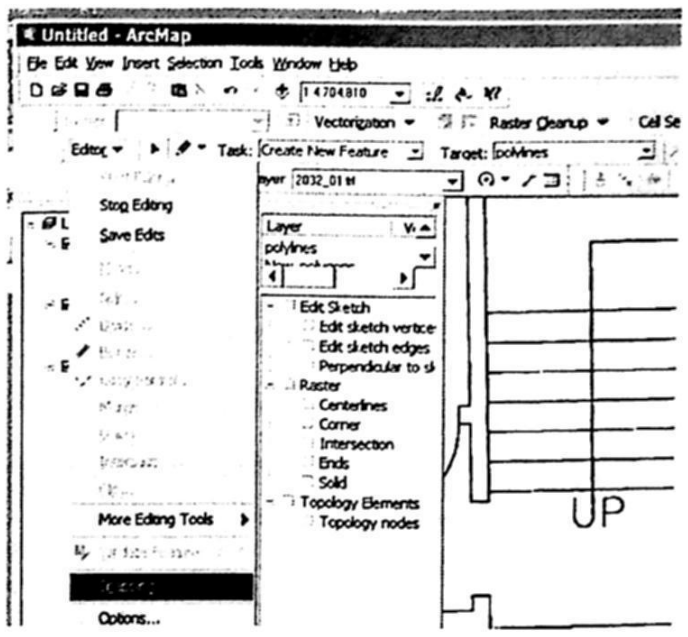


Figure 8. Interface for calling raster snapping in ArcScan

Figure 8 illustrates the procedure for raster snapping in the ArcScan conversion sub-system of ArcGIS. It allows raster snapping on a centerline, nodes intersections, corners (critical points), ends of lines, and centers of solids.

If raster snapping is performed on the ends and corners, then snapping just to two points can vectorize every straight line. At the same time, it is not necessary to zoom in the image to achieve high accuracy.

5. DIVISION OF LABOR BETWEEN OPERATOR AND MACHINE

The efficiency of a conversion system depends largely on how responsibilities are distributed between operator and machine. Humans are peculiarly adept at recognition. They easily identify various thematic layers, isolate and recognize texts and symbols, find critical points, locate the continuation of broken lines, situate the contours of areas filled with symbols, recognize shaded areas and much more, all things that are difficult for a computer program to do automatically.

Let us take the problem of closing gaps as an example [13]. Usually algorithms for solving this problem use two thresholds, search radius and fan angle. However, in some discontinuity points of lines several candidates for continuation can be found in spite of these two thresholds. Even if a more complex algorithm is used or additional parameters are introduced, there is no way to obtain the correct result in all cases. Is it worth it, then, making algorithms more and more complicated? Would not it be better to pass these problems on to the operator and to simplify and stabilize the algorithms instead?

The same issue arises with intersections. Figure 1 showed four possible versions of the solution of one intersection. An operator can easily decide which version is correct, but then it is necessary to redraw the intersection and this is a time-consuming and labor-intensive process if the operator will do it.

Better if the operator's decision can be communicated to the program via templates. Figure9 shows four templates that correspond to the four solutions of this intersection. The program, no longer required to provide a decision, could be simplified and would guarantee obtaining a

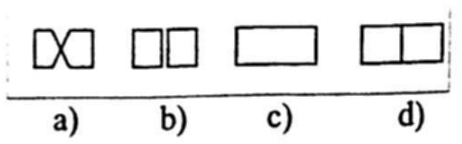


Figure 9. Templates for the intersection in Figure 1.

stable solution. The operator must simply choose the correct template, and thus be spared the lengthy and tedious task of drawing.

Therefore, let us take this into account when designing a conversion system. Let operator do what he can do easily, and let us use the computer and automatic methods to solve those problems that render themselves to formalization and have strict stable solutions. "Render unto Caesar the things that are Caesar's, and to God the things that are God's."

In order to use algorithms of automatic and semi-automatic vectorization, it is often necessary to know the maximum thickness of the linear elements. Frequently this value is not already known. To determine it, one has to measure the line thickness in several places, usually zooming in to an area and pointing the cursor at edge of the line and then the other with the distance tool. This has to be repeated several times.

Can this procedure be simplified? We developed a relatively simple algorithm that will measure line thickness in any point. All the operator has to do is to place cursor next to the line. Figure10 illustrates how ArcScan performs this procedure. Local thickness of the line equals 12 pixels.

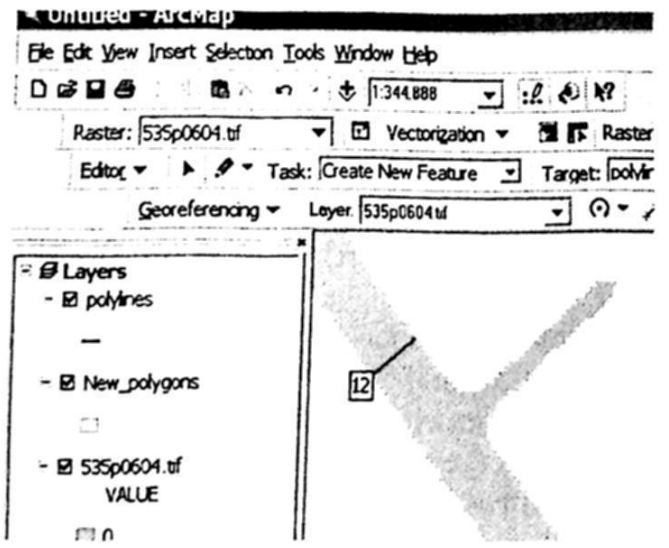


Figure 10. Measuring local thickness

Verification of vectorization results takes a long time. The authors of [14] proposed an algorithm that would identify how much the result of vectorization deviated from centerline. This algorithm can be used to automatically identify spots where deviations exceed some given value and mark them (see Figure11). The deviations can result from either the vectorization algorithm itself (a) or from post-processing of the obtained result, such as compression (b) and smoothing (c).

Automatic verification of the accuracy of vectorization result can significantly increase the efficiency of the conversion system.

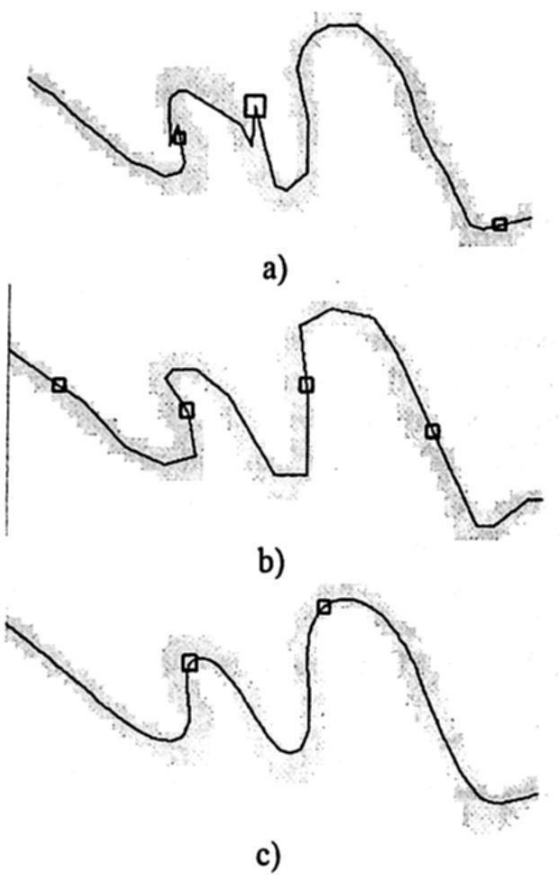


Figure 11. Automatic identification and marking of large local deviations of centerlines

While there are presently a lot of methods and programs for text recognition, there are no effective text recognition programs for graphic documents. Maps, engineering drawings, electrical schematics, and other graphic documents have lots of text on them, text that may touch linear objects or be otherwise difficult to distinguish, automatically, from graphic elements. To recognize text of graphical documents, it is necessary first to separate it from linear objects, solids, and symbols and to define its orientation.

There are no efficient programs that do it automatically. The operator can do it. As proposed in [15], the operator can draw line through the text, marking it as text and revealing its orientation all in one step.

Many conversion systems allow us to clean the raster image of noise before vectorization. To perform this task, it is necessary to find speckles and holes as connected components that meet requirements for size, area, and sometimes more sophisticated characteristics. But small graphic elements (dashes, dots, and others) can be identified as noise too. So it is more efficient to select speckles and holes automatically and highlight them. Then the operator can verify the result of selection and, if

necessary, to correct it by selecting or unselecting connected components before cleaning.

When processing the map of a city, it is often difficult to vectorize rectangular buildings. If it is a relatively small-scale map, contours of buildings can have quite a big noise. The program for an automatic recognition of rectangle buildings with big noise is complex and doesn't always give good and stable results. Manual drawing rectangle contours of arbitrary orientation is time-consuming. But an operator can recognize solids that are rectangular buildings easily and there are simple programs that approximate borders of the solids with rectangles. So it is possible to develop an effective interactive procedure of one click vectorization of rectangular buildings.

One way to increase the effectiveness of conversion systems is by using learning algorithms. The corners between straight-line segments and the boundary points of circle arcs are called critical points. The recognition of critical points is an important component of the conversion process, because critical points will help to correct recognition of geometrical objects. All the algorithms for this task use some thresholds. Often it is difficult to evaluate these thresholds because they depend on so many factors: maximum and minimum curvatures, noise, thickness of lines, and so on. But it is relatively easy to show critical points on the screen (Figure 12). The effectiveness of the conversion system will be increased if an algorithm can be developed that can automatically evaluate necessary thresholds and parameters using information about location of some of the critical points.

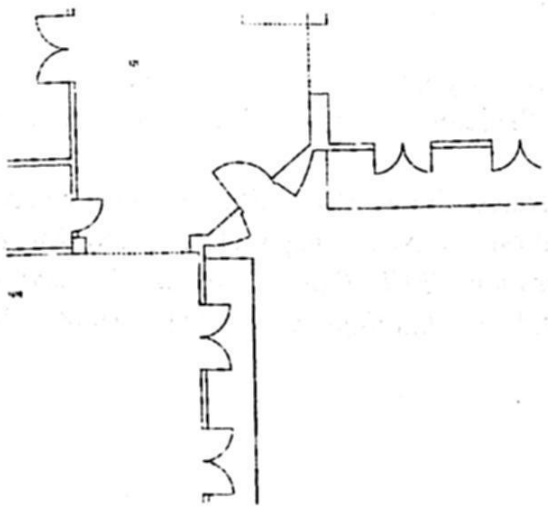


Figure 12. Fragment of architectural design

Sometimes, addition of a simple tool drastically expands a system's capabilities. ArcScan added a seemingly insignificant instrument that allows toggling colors of foreground and background pixels (see Figure13).

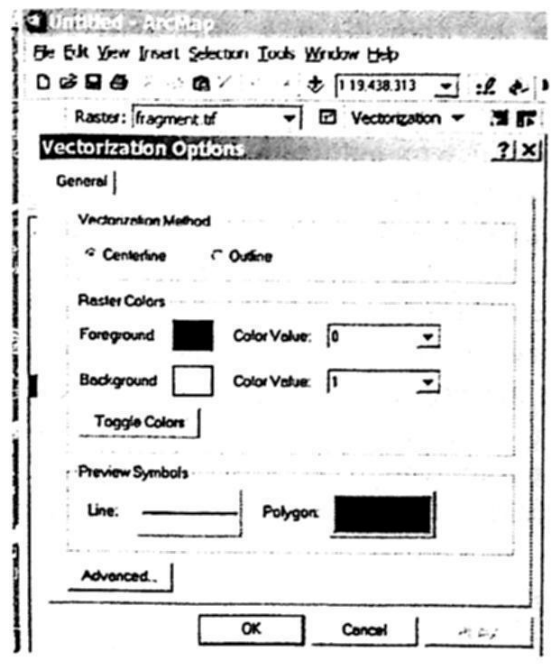


Figure 13. Tool for toggling colors of foreground and background pixels

When combined with semi-automatic vectorization (tracing), however, it allows one to vectorize a non-trivial image, in which linear objects are represented both by solid and double lines (Figure14).

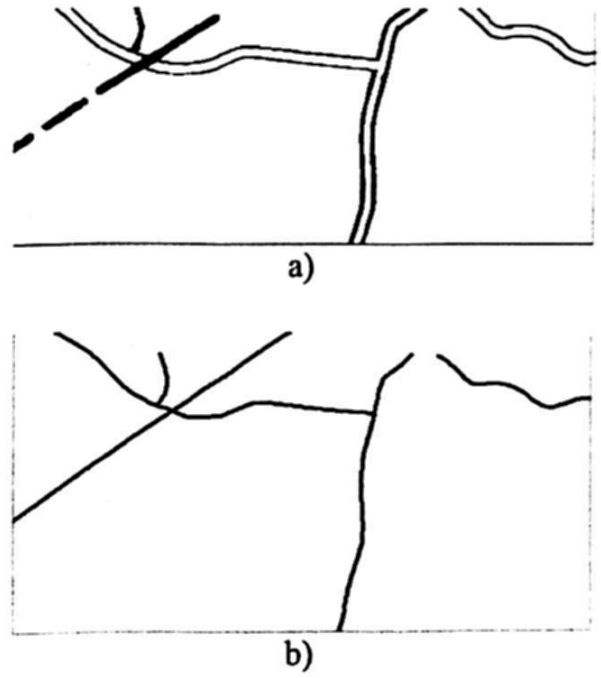


Figure 14. A source raster image (a) and the result of vectorization (b)

6. CONCLUSION

The article analyzes the task of data capturing and the principles of designing conversion systems. It demonstrates with examples how the efficiency of the conversion systems can be greatly increased if the division of labor between operator and machine will be done right, when an operator does what he can do easily, and the computer solves those problems that render

themselves to formalization and have strict stable solutions.

It further highlights the significance of the algorithms and methods of semi-automatic vectorization, which have been largely ignored by the scholarly community. Our approach involves a comprehensive analysis of the system as a whole, rather than looking at it as a mere collection of separate instruments.

The new version of the conversion system, ArcScan, which was developed by ESRI and is a part of ArcGIS, is a prototype of the new generation of conversion systems. In preparing the article, we used ArcScan to illustrate some of our statements, assumptions, and conclusions.

REFERENCES

[1] *GIS Data Conversion: Strategies, Techniques, Management* (Ed.: Pat Hohl 1998).

[2] L.Baotto, V.Consorti, M.Del Buono, S. Di Zenzo, V.Eramo, A.Esposito, F.Melcarne, M.Meucci, A.Morelli, M. Mosciatti, S.Scarci, M.Tucci, An Interpretation System for Land Register Maps, *IEEE Computer*, 25(7), 1992, 25-33.

[3] A.J.Filipsky, R.Flandrena, Automated Conversion of Engineering Drawing to CAD Form, *Proceedings of the IEEE*, 80(7), 1992, 1195-1209.

[4] Lawrence O'Gorman, Basic Techniques and Symbol-Level Recognition – An Overview, *Lecture Notes in Computer Science Vol. 1072*, 1996, 1-12.

[5] Sergey Ablameyko, Tony Pridmore, *Machine Interpretation of Line Drawing Images. Technical Drawings, Maps, and Diagrams* (Ed.: Springer-Verlag, 2000).

[6] Serguei Levachkine, Evgueni Polchkov, Integrated Technique for Automated Digitization of Raster Maps, *On-line Journal: Revista Digital Universitaria*, 1(1), Art. 4, 2000 (www.revista.unam.mx/vol.1/art4)

[7] *Parcel Mapping Using GIS. A guide to Digital Parcel Map Development for Massachusetts Local Governments*. Prepared by University of Massachusetts Office of Geography Information and Analysis for MASSGIS. August 1999. http://umass.edu/tei/ogia/parcelguide

[8] Dori Dov, Wenyin Liu, Sparse Pixel Vectorization: An Algorithm and Its Performance Evaluation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3), 1999, 202-215.

[9] Hori Osamu, Doermann David S, Quantitative Measurement of the Performance of Raster-to-Vector Conversion Algorithms, *Lecture Notes in Computer Science, Vol. 1072*, 1996, 57 – 68.

[10] Jaisimha M.Y., Dori D., Haralick R., A methodology for the characterization of the performance of thinning algorithms, *Proc. of Int. Conf on Document Analysis and Recognition (ICDAR'93)*, Tsukuba, Japan, 1993, 282-286.

[11] Phillips Ihsan, Liang Jisheng, Chhabra Atul K., Haralic Robert, A Performance Evaluation Protocol for Graphics Recognition Systems, *Lecture Notes in Computer Science Vol. 1389*, 1998, 372-389.

[12] Dave Elliman, A Really Useful Vectorization Algorithm, *Lecture Notes in Computer Science Vol. 1941*, 1999, 19-27.

[13] Eugene Bodansky, Alexander Gribov, Closing Gaps of Discontinuous Lines: A New Criterion for Choosing the Best Prolongation, *Lecture Notes in Computer Science Vol. 2423*, 2002, 119-122.

[14] Bodansky Eugene, Pilouk Morakot, Using Local Deviations of Vectorization to Enhance the Performance of Raster-to-Vector Conversion Systems, *International Journal on Document Analysis and Recognition*, No. 3, 2000, 67-72.

[15] Arvind Ganesan, Integration of Surveying and Cadastral GIS: From Field-to-Fabric & Land Records-to-Fabric, *Proc. ESRI User Conference 2002* (http://gis.esri.com/library/userconf/proc02/abstracts/a0868.html)

[16] 16. Ogniewicz R.L., Kubler O, Hierarchic Voronoi Skeletons, *Pattern Recognition*, 28(3), 1995, 343-359.

[17] Desseilligny M.P., Stamon G., Suen Ch.Y., Veinerization: A New Shape Description for Flexoble Sceletonization, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5), 1998, 505-521.