Vectorization of Raster Images

A. N. Kolesnikov*, V. V. Belekhov, and I. O. Chalenko****

** Institute of Automation and Electrometry, Siberian Division, Russian Academy of Sciences, Universitetskii pr. 1, Novosibirsk, 630090 Russia*
*** AO VMK-Optoelektronika, Universitetskii pr. 1, Novosibirsk, 630090 Russia*

Abstract—In this paper, we present the principles of design and discuss algorithms and software of the system for automatic transformation of raster graphic images into the vector format that is oriented toward enriching data bases of geoinformation systems. The developed system complements the available interactive and semiautomatic vectorization means and makes it possible to considerably reduce the number of routine manual operations required for the inputting of maps, leaving the editing and augmenting of the vector representation for an operator.

1. INTRODUCTION

A broad spread of computers in the last decade and considerable improvement of their parameters (the speed, the capacity of the random-access and external memory, and the quality of monitors) made it possible to efficiently employ computer means in various computer-aided design and geoinformation systems and in data bases of textual and graphic information. In this context, we encounter a rather urgent problem of translating of the available graphic information (maps, plans, drawings, and schemes) from paper carriers into the form that would allow compact storage and efficient use of these data in information and computer systems.

The existing technology of inputting graphic information into a computer is reduced to the manual tracking of map or plan lines with the use of various digitizers. In certain cases, this problem is solved by tracking the lines in a raster image of a map on a computer display, which implies that the map should be preliminarily inputted into a computer by means of a scanner. Available computer programs facilitate manual line tracking (automatic capture of a line, an object, etc.). Certain special-purpose systems employ, as a compromise, hybrid data representation, when the information concerning, for example, water-supply and sewerage is represented in the vector format, and the city map is represented in the raster form.

In one way or another, the inputting of graphic data into a computer requires an abundance of monotonous manual operations, especially when maps with a large number of winding lines are to be inputted. In certain cases, the maps to be inputted are so complicated that the possibility of their manual inputting within reasonable time casts some doubt (see Fig. 1).

The application of automatic transformation of raster images into the vector format could reduce the number of routine manual operations required for the inputting of graphic data into a computer, leaving the editing of the resulting vector representation for an operator. Currently, several automatic systems are available for this purpose. Some of such systems are described in [1–3].

In this paper, we present the principles of design and discuss algorithms and software of the system for automatic transformation of raster graphic images into the vector format that is oriented toward the augmenting of data bases of geoinformation systems (inputting of maps, city plans, and schemes).

2. PROBLEMS OF THE AUTOMATIC VECTORIZATION OF RASTER IMAGES

It is unlikely that the problem of totally automated transformation of raster images into the vector format can be completely solved. Therefore, in developing the system under consideration, we pursued less general but more easily achievable goal—to reduce the number of routine operations required for the inputting of graphic images as much as possible and facilitate the subsequent augmenting and editing of the resulting vector image performed by an operator. The developed automatic vectorizer is intended to complement the available interactive and semiautomatic vectorization means rather than to replace them.

Solving a problem concerning the expediency of applying an automatic vectorizer, it is important to adequately estimate its capabilities, as well as its advantages and disadvantages, in order to implement the procedure of the inputting of graphic information in the optimal manner. The part of an image that can be correctly automatically vectorized is determined by several factors, including the quality, the complexity, and the character of the raster image. For example, it is unlikely that the application of an automatic vectorizer could be advantageous in the inputting of plans consisting of straight lines with a large number of intersections and a low quality of the original. An operator would cope with this task much more successfully. At the

Received November 13, 1995

Pattern Recognition and Image Analysis, Vol. 6, No. 4, 1996, pp. 786–794.

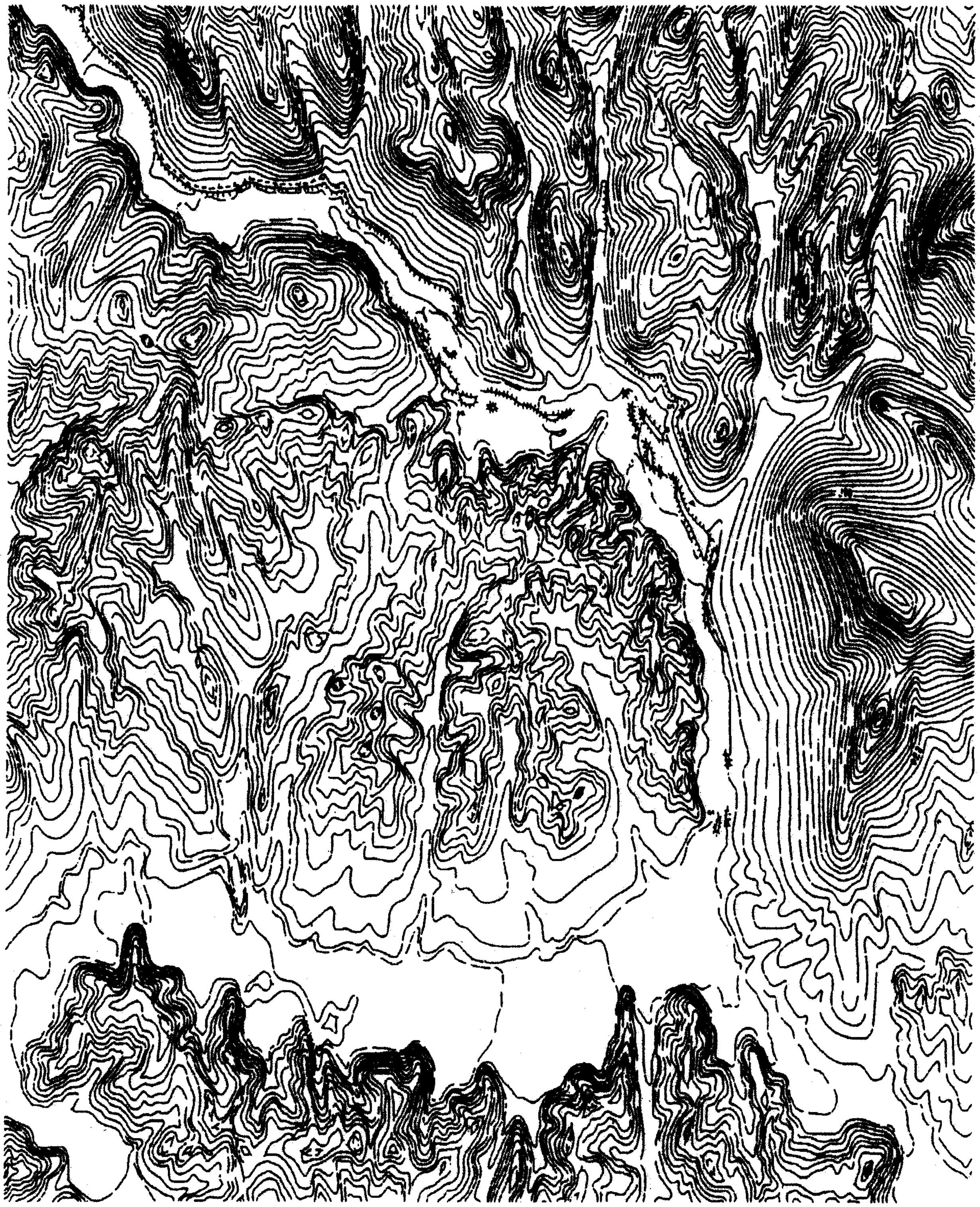


Fig. 1. Vectorized image of a map fragment.

same time, there is a rather broad class of images where the use of an automatic vectorizer is quite appropriate.

To achieve this goal, we developed algorithms and software that allow one to vectorize a raster image and analyze the result of vectorization in order to select the part of the image that can be correctly interpreted at the computer level and to eliminate those vectors from the image that do not belong to certain ordered structures or that are inconsistent with the adopted image model.

In designing a system for the automatic vectorization of raster images, we should also take into account the scale factor. Transformation of the images of maps and drawings is associated with the necessity to process a very large body of information. In addition, the number of maps is also usually very large. Therefore, all the algorithms should be optimized in their speed and in the requirements imposed on computer resources (random-access memory, the capacity of the hard disk, etc.).

In connection with these problems, we focused a particular attention to the development of such algorithms that do not require much space in random-access memory and do not imply too intense data exchange between the computer and the external memory. All the developed algorithms ensure correct processing of an image in the course of a single sequential reading of the file.

To reduce the load on the disk memory, all the information that is to be written on the disk is stored in a compressed form. The employed methods of data compression provide an opportunity to store the compressed information in a sufficiently compact form. However, compression does not interfere with fast random access to these data.

3. VECTORIZATION ALGORITHM

The developed algorithm for the transformation of a raster image into the vector format can be divided into four main stages:

- binarization of a gray-scale image;
- skeletonization of the binary image;
- primary vectorization of the skeleton image;
- analysis of the vector representation.

As can be easily seen, each stage of the algorithm implies the transformation of the data into a form with a higher level of structuring: raster points are transformed into marked skeleton points, skeleton points are transformed into vectors, vectors are transformed into vector chains, and the list of chains is transformed into a graph. Such an approach makes it possible to reduce the size of data bases without the loss of significant information.

3.1. Locally Adaptive Binarization

Obviously, the quality of the output vector representation is determined, to a great extent, by the quality of the input binary image. The problem of binarization is complicated by the fact that, usually, large-size images are to be processed. Because of nonuniform illumination of the document in scanners of certain types or because of inhomogeneities of the document itself, it is difficult and sometimes impossible to choose a single binarization threshold for such images.

To improve the quality of binary images, we implemented locally adaptive binarization of the input gray-scale image [4]. This procedure was performed in two stages:

- analysis of the image;
- binarization proper.

Analysis of the image is performed in the following manner. The entire image is divided into nonoverlapping rectangular fragments, and an intensity histogram is computed for each fragment. In accordance with this histogram, we evaluate the local threshold or make a decision that this fragment does not contain any objects. In the latter case, the threshold function for this fragment is determined from the neighboring fragments that contain objects.

Binarization of the input gray-scale image implies that a local threshold function is computed for each point from four adjacent fragments and processing is performed with this threshold (see Fig. 2). In addition, to reduce noise on the boundaries of objects, provision is made for the filtration of the binary image.

3.2. Skeletonization of Images

To obtain the skeleton representation, we employed an algorithm based on distance transform [5]. The specific features of algorithms of this type is associated with the fact that a skeleton can be obtained with the use of a fixed number of passes through an image, regardless of the thickness of objects. In addition, each point of the resulting skeleton contains information concerning its local thickness. Thus, replacing a binary image by its skeleton representation, we are able to save all the significant topological and metric information concerning the image, which is subsequently used in the vectorization and analysis of this image (see Figs. 3 and 4).

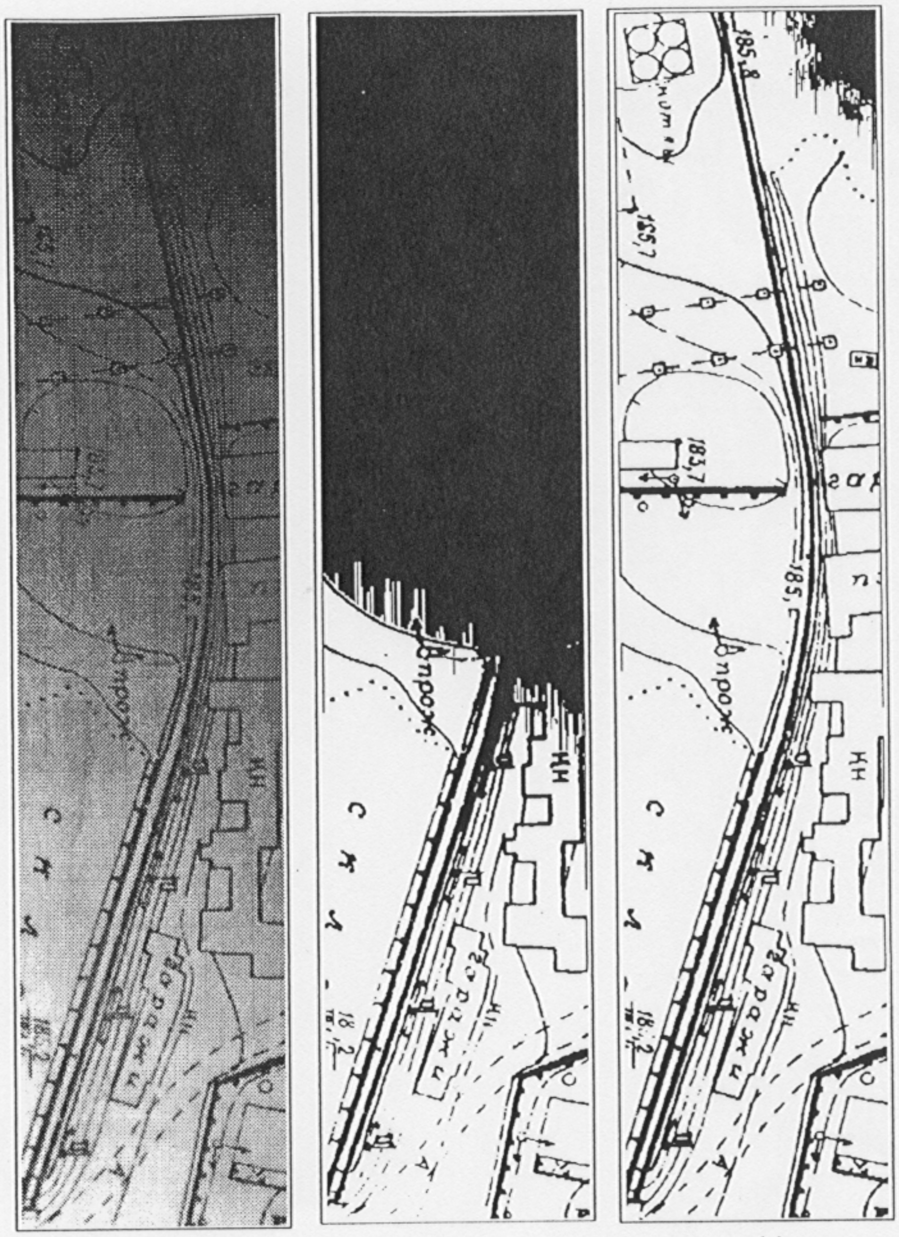
One of possible ways to reduce the processing time is to apply multiprocessors. With this aim in mind, we developed a parallel skeletonization algorithm intended to operate in multiprocessor systems with distributed memory [6, 7].

3.3. Primary Vectorization

Primary vectorization implies the transformation of the skeleton representation into a set of vectors. At this stage, we track all the branches of the skeleton, analyze the obtained information on the behavior of these lines, and determine parameters of straight lines defined on a discrete grid that approximate skeleton branches [8]. The results of primary vectorization are written as a set of vectors with the corresponding attributes (see Fig. 5).

3.4. Analysis of Vectors

Experience shows that it is rather difficult to ensure an acceptable quality of vector representation by means of line tracking and approximation of these lines by broken (or some other) lines without additional analysis



(*a*) (*b*) (*c*)

Fig. 2. (a) Nonuniformly illuminated input gray-scale image. (b) Image obtained by means of binarization of the input image with a single threshold. (c) Image obtained by means of binarization with a locally adaptive threshold.

in accordance with a model of an image being analyzed and on the basis of more global information concerning the image [1–3].

For this purpose, we developed a data base intended to be used as part of the considered system, where the obtained vectorized image is represented in the form of an attributed graph, which allows a fast and convenient access to the entire collection of vectors and permits processing of these vectors. Using this data base, we implemented several algorithms for the analysis and processing of vector images to reduce the size of the vector description of an image and to enchance the image.

4. ANALYSIS OF THE VECTOR REPRESENTATION

Let us consider in greater detail the stage of vector analysis, because it is this stage that determines, to a great extent, the quality of the output vector representation.

Once processing has been completed, the vectorizer outputs a vector file where all the vectors reproduce the

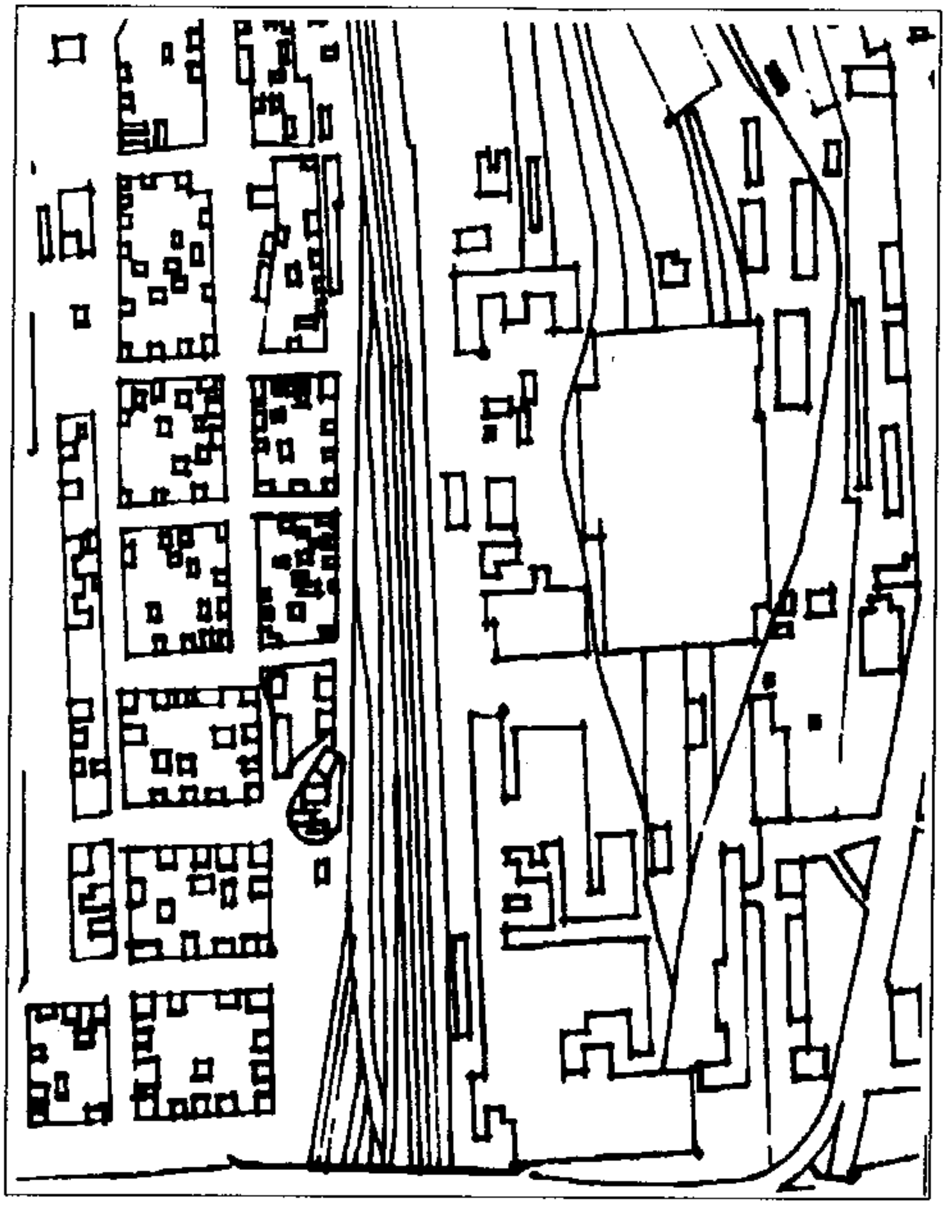


Fig. 3. Binary image of a fragment of a 1 : 10 000 map.

lines in the skeleton image with a high accuracy. The vectors in this file are stored in the same order as they were detected by the vectorizer. Analysis of possible variants of vector storage shows that the use of a linear structure for data storage is associated with a time-consuming procedure of searching for elements on the order of $O(n)$, where n is the number of elements, in such a structure. In the case of a treelike structure, the search time can be estimated as $O(\log_2 n)$. Therefore, it was decided to store the vectors using one of the modifications of a binary directory, namely, the so-called AVL trees. The specific feature of an AVL tree is that the depths of two subtrees at each vertex in such a tree can differ by no more than unity. The search time for an arbitrary element in an AVL tree does not exceed a quantity on the order of $O(\log_2 n)$. Thus, the vector analyzer starts to operate with extracting an ordered data base from the vector file obtained after primary vectorization.

4.1. Transformation of a Vector Set into a Treelike Data Base

To form a data base within the set of vector vertices, we introduce two relations: order relation and incidence relation.

The order relation ">" implies that two vertices A_1 and A_2 with coordinates (x_1, y_1) and (x_2, y_2), respectively, satisfy the condition $A_1 > A_2$ if either $x_1 > x_2$ or $x_1 = x_2$ and $y_1 > y_2$. The relation ">" is defined within the

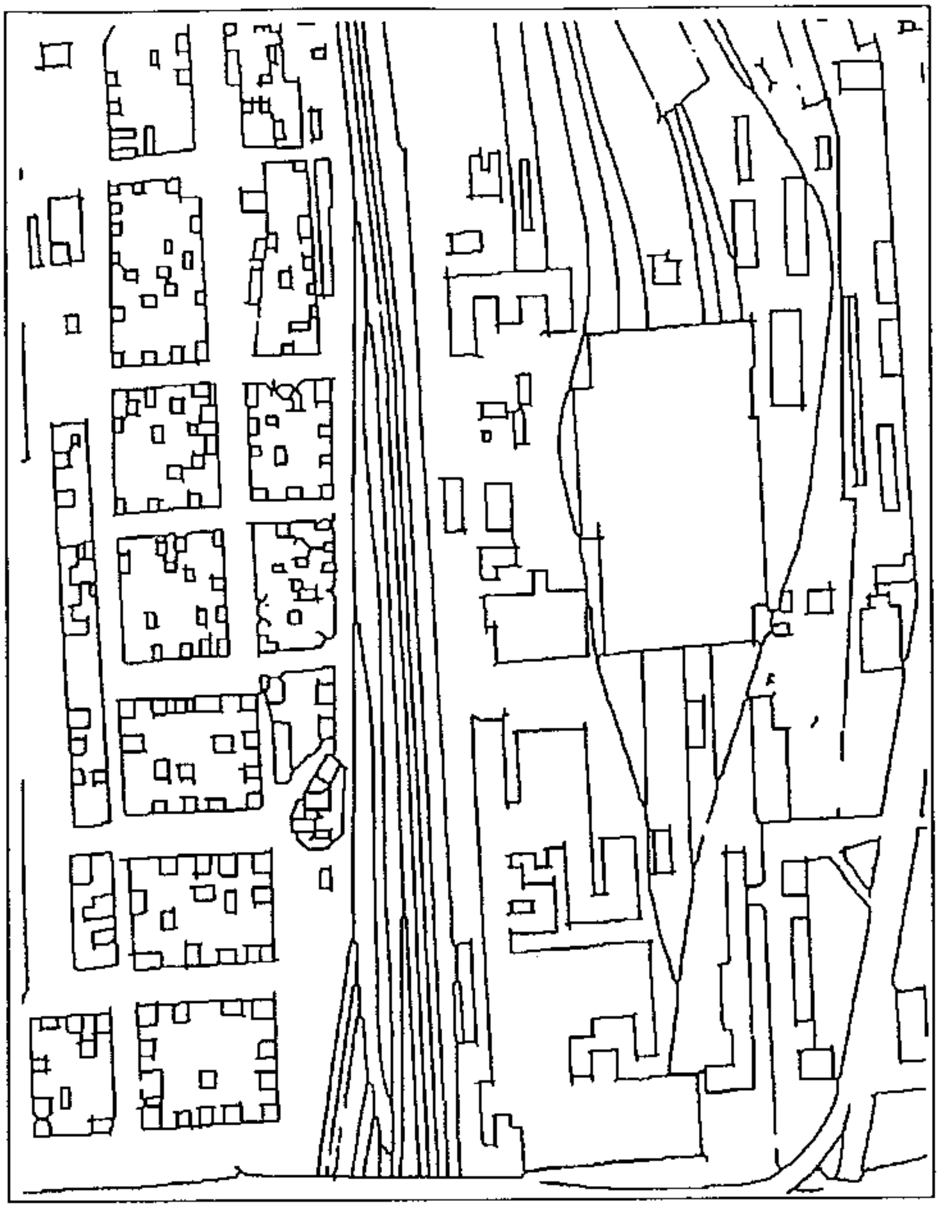


Fig. 4. Skeleton of the binary image of the map fragment shown in Fig. 3.

set $M = V \times V$, where V is the set of all the vertices defined as a set of ordered pairs of integers. In addition, the relation ">" is the equivalence relation. Now, let us introduce the incidence relation for vertices. A vertex A_1 is incident to a vertex A_2 if the set $I \subseteq M$ that consists of all the image vectors includes an undirected segment $\{A_1, A_2\}$.

Having introduced order and incidence relations, we can propose the following procedure for the formation of the data base:

(1) The vertices of all the image vectors are inserted into an AVL tree in accordance with the order relation;

(2) Each vertex element in the obtained AVL tree is attached to the list of the corresponding incident vertices with a certain additional information. The vertex element and the corresponding incident vertex together with additional information represent a vector in the initial image.

The employed model of data for the representation of a vector image ensures a fast and simple access to the vector data required at the subsequent stages of image analysis.

4.2. Chain Extraction from the Treelike Data Base

Formation of the data base with the use of an AVL tree allows us to represent the vector image in the form convenient for processing. To solve applied problems associated with the processing of vector images (to simplify the description of such images, to extract certain

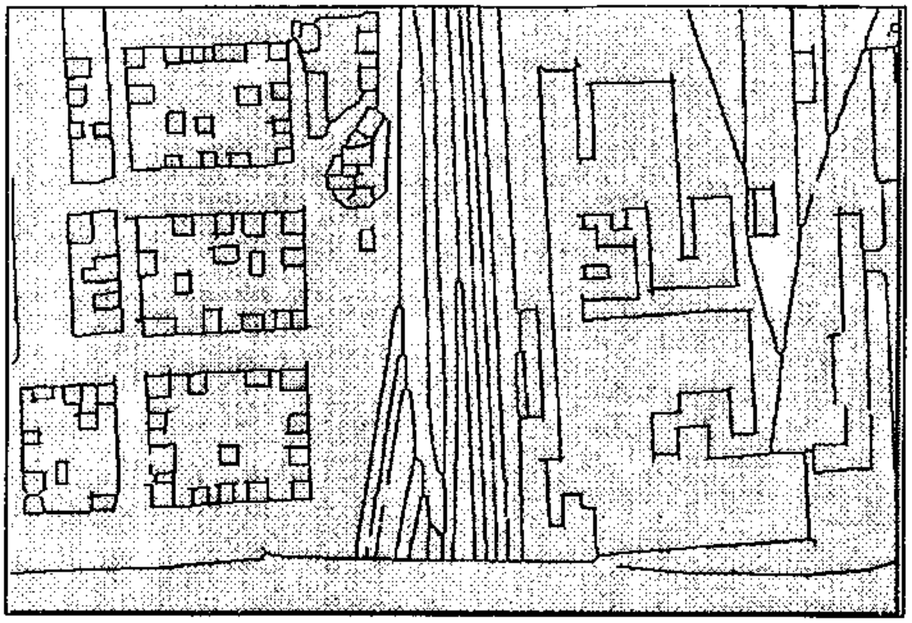


Fig. 5. Fragment of the vector image obtained after the primary vectorization of the skeleton image shown in Fig. 4.

structures, etc.), we should consider a set of objects at a higher level of abstraction than vectors. At the moment, the analyzer employs objects of one of possible classes—chains of vectors.

A chain of vectors is defined as a set of directed vectors $C_1, C_2, \ldots, C_n$ such that the initial vertex of the vector C_{m+1} coincides with the final vertex of the vector C_m, where $1 \leq m \leq n - 1$. A chain of vectors may be closed and may form a contour. Then, the initial vertex of the vector C_1 coincides with the final vertex of the vector C_n. The ends of a chain are defined in the following manner: a vertex is referred to as the end of a chain if one or more than two other vertices in the vector image are incident to this vertex. Using chains, we can describe the majority of objects in an image (houses, roads, isolines, etc.).

4.3. Analysis of Chains in a Vector Image

At the next stages, the analyzer processes the extracted chains in order to identify the best structured, i.e., the most informative, part and analyzes these chains in order to remedy various defects (to eliminate false branches and approximate chains with broken lines within the limits of the preset error).

4.3.1. Partitioning of chains into classes. Once the analyzer has extracted all the chains in the vector image, the chains are classified in accordance with a certain set of features. The necessity to perform such a classification is due to the fact that, in most cases, an image includes several details that cannot be recognized by the considered analyzer (text and notation). Furthermore, an image usually contains a considerable number of noise components. At the same time, a certain part of the image can be successfully processed. To reduce the subsequent work of an operator on eliminating defects of automatic vectorization, the analyzer filters out a part of information that would be definitely impossible to satisfactorily process.

The classification process is governed by several parameters, which allows one to adjust the analyzer to a definite class of images and problems. As chain-classification parameters, we chose the length of a chain, its curvature, the number of intersections, and some other characteristics, including parameters that specify the relative arrangement of chains in the image. All in all, analysis of a vector image takes into account more than ten various parameters characterizing the chain structure of the image.

Based on the results of classification, we can divide chains into two classes: "good" chains, i.e., chains chosen by the analyzer for further processing, and "bad" chains, i.e., chains that are excluded from subsequent processing but that can be saved, if required, in a certain layer of the output vector file.

4.3.2. Glueing and joining of chains. Because of a low contrast of the input gray-scale image, binarization of a gray-scale raster image may be accompanied by the appearance of defects in a binary image, i.e., discontinuities at certain points of lines in the binary image (and, consequently, in the vector image). The analyzer is intended to eliminate such discontinuities if the length of these defects is not too large.

To eliminate discontinuities, the analyzer checks whether or not the considered pair of chains formed a single continuous line in the original image that could be represented as a union of this pair of chains (the so-called

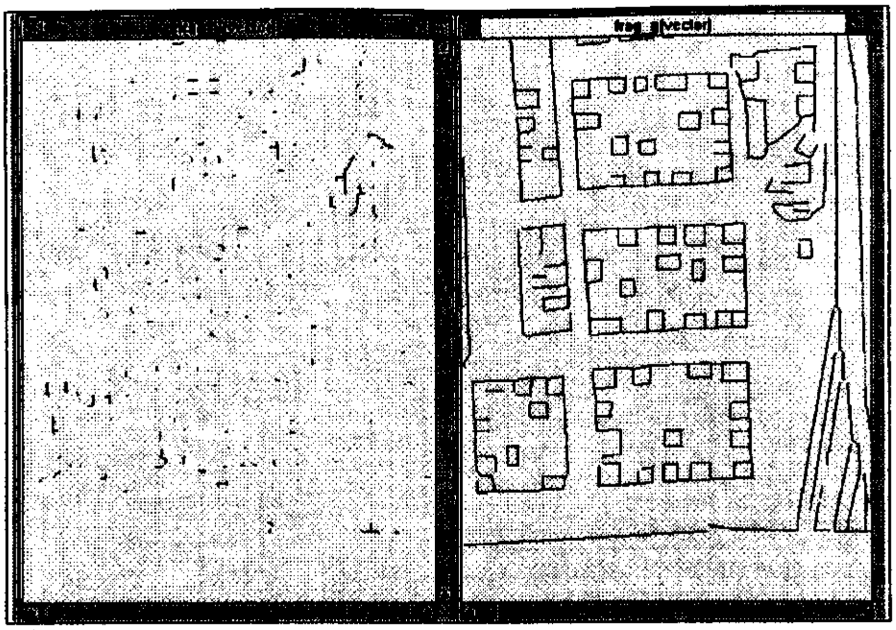


Fig. 6. Two layers of the vector image obtained after processing the fragment of the vector image of the map shown in Fig. 3 with the use of the analyzer. The layer of "bad" vectors is on the left, and the layer of "good" vectors is on the right.

glueing of chains). Such a procedure, which is performed for all chains in the image, implies that an appropriate weight is determined for each configuration, whereupon the analyzer sorts the revealed candidates for glueing in accordance with their weights and excludes the candidates that should not be subject to glueing for given parameters of processing. Once sorting and the elimination of inappropriate candidates have been completed, the remaining chains are glued together in pairs.

Simultaneously with glueing, the chains are joined at intersection points. The latter procedure implies that, for each pair of chains having a common vertex at the intersection point (i.e., a vertex where more than two vectors meet each other), we consider the possibility that these chains can form a single continuous chain of a greater length. The glueing and joining of chains are performed in accordance with the same algorithm.

With the use of the above-described procedures, the analyzer, in fact, extracts image objects with larger sizes than lines passing from one intersection point to another. In practice, such an approach allows us to extract polygons and long lines (isolines, roads, etc.) in a vector image in spite of the fact that some irrelevant lines may adjoin these objects.

4.3.3. Elimination of false branches. Skeletonization of a binary raster image is often accompanied by the appearance of false skeleton lines caused by noise on the boundaries of objects. Naturally, in the vectorization of the skeleton, these false branches are tracked and transformed into vectors. There are certain features that make it possible to recognize such vectors. In particular, the length of such vectors is comparable with their thickness at the initial point, and the thickness decreases virtually linearly as we approach the final vertex. The analyzer eliminates such vectors from the chain or attempts to correct them at the stage when the chains are glued and joined together at intersection points. The vectors that should be eliminated from chains are marked as bad vectors and are removed into the corresponding layer (see Fig. 6).

The above-described procedure provides an opportunity to eliminate a considerable number of noise components from a vector image. Manual elimination of these components would be time consuming.

4.3.4. Elimination of defects in right angles. In gray-scale raster images, interior angles are often padded and exterior angles are blurred, which leads, in particular, to the smoothing of right angles in a vector image. To eliminate such defects in right angles, we developed an algorithm that corrects these roundings. Correction of this type is performed in chains upon the elimination of false skeleton branches. If necessary, this algorithm can be employed to correct arbitrary angles.

4.3.5. Rectification of chains. As mentioned above, a skeleton image is transformed into a vector image with a minimum approximation error. Since both the input gray-scale image and the intermediate binary image contain noise, the skeleton image also contains a certain number of noise vectors. In addition, the process of skeletonization is accompanied by the appearance of certain artifacts associated with the discrete character of the image, the type of the spatial metric, and other circumstances. Therefore, the extracted chains consist of a large number of small-size vectors, which increases the memory space required to store the vector image and causes considerable difficulties in subsequent manual editing.

To approximate the extracted curves by broken lines with a preset accuracy, the analyzer employs a modified skeletonization algorithm [6], which approximates (with preset restrictions on the approximation error) a broken line by another broken line with a smaller number of vectors. This stage of processing of a vector image (rectification of chains) completes the analysis of the vector representation.

The application of this algorithm makes it possible to reduce the number of vectors in chains, on the average, by a factor of two to seven without causing considerable deviations of chains from the binary image.

The analyzer outputs the resulting file (see Figs. 1 and 6) in the DXF format. This file can be edited or supplemented in Autocad or in a geoinformation system compatible with this format.

5. CONCLUSION

Thus, we developed algorithms and software that make it possible to preprocess gray-scale raster images, automatically vectorize them, and perform subsequent image analysis at the vector level. The developed software package is intended to be used in geoinformation systems for the enrichment of the graphic data base and translation of graphic information from paper carriers into a computer.

The developed software package provides an opportunity to considerably reduce the number of routine manual operations required to input graphic information into a computer and improves the accuracy of this procedure due to the elimination of the subjective factor. The application of the developed software package to the solution of practical problems demonstrated its high efficiency.

All the developed algorithms were implemented as a software for an IBM PC. In addition, some of the processing algorithms were implemented as a software for an IBM RS/6000 (Model 590) workstation and a PARAStation transputer workstation (the Transtech firm) controlled by a SPARCStation 10 workstation (the Sun firm).

Currently, the work is in progress on improving the algorithms and the software. The objective of these studies is to improve the quality of the output vector representation and to simplify the procedure of choosing optimal parameters in the adjustment of the system to a specific type of images with allowance for characteristic features of the problem under consideration. In addition, we intend to extend the developed algorithms to the analysis of objects contours in those cases when such a description of objects is appropriate.

REFERENCES

1. Sakauchi, M. and Ohsawa, Yu., A New Type of Image Processing Using a Dynamic Graphic Data Structure, *Proc. SPIE, Application of Digital Image Processing XI*, 1988, vol. 974, pp. 135–142.
2. Darwish, A.M. and Jain, A.K., A Rule Based Approach for Visual Pattern Inspection, *IEEE Trans. Pattern Analysis Machine Intelligence*, 1988, vol. 10, no. 1, pp. 56–68.
3. Suzuki, S. and Yamada, T., MARIS: Map Recognition Input System, *Pattern Recognition*, 1990, vol. 23, pp. 919–933.
4. Sahoo, P.K., Soltani, S., and Wong, A.K., A Survey of Thresholding Techniques, *Computer Vision, Graphics and Image Processing*, 1988, vol. 41, pp. 233–260.
5. Arcelli, C. and di Baja, G.S., A One-Pass Two-Operation Process to Detect the Skeletal Pixels on the 4-Distance Transform, *IEEE Trans. Pattern Analysis Machine Intelligence*, 1989, vol. 11, no. 4, pp. 411–414.
6. Kolesnikov, A. and Trichina, E., Parallel Algorithm for Image Skeletonization, in *Proceedings of the Fifth International Workshop on Distributed Data Processing*, Akademgorodok, Novosibirsk, 1995, pp. 276–281.
7. Kolesnikov, A. and Trichina, E., Parallel Algorithm for Binary Image Thinning, *Optoelectron. Instrum. Data Processing*, 1995, no. 6.
8. Aoyama, H. and Kawagoe, M., A Piecewise Linear Approximation Method Preserving Visual Feature Points of Original Figures, *Graphical Models and Image Processing*, 1991, vol. 52, no. 5, pp. 446–453.

Aleksandr N. Kolesnikov. Born 1954. Graduated from Novosibirsk State University in 1976. Researcher at the Institute of Automation and Electrometry, Siberian Division, Russian Academy of Sciences. Scientific interests: compression and analysis of signals and images. Author of 25 scientific publications.

Vitalii V. Belekhov. Born 1975. Student of the third year of study at the Faculty of Mechanics and Mathematics, Novosibirsk State University. Programmer and engineer at AO VMK-Optoelektronika. Scientific interests: image processing, pattern recognition, and artificial intelligence. Author of one scientific publication.

Igor' O. Chalenko. Born 1975. Student of the third year of study at the Faculty of Mechanics and Mathematics, Novosibirsk State University. Programmer and engineer at AO VMK-Optoelektronika. Scientific interests: image processing and analysis. Author of one scientific publication.