Methods for automated vectorization of point objects on cartographic images

S. Rychazhkov[1], V. Fedoseev[1,2], R. Yuzkiv[1,2]

[1]*Samara National Research University, 34 Moskovskoe Shosse, 443086, Samara, Russia*
[2]*Image Processing Systems Institute – Branch of the Federal Scientific Research Centre "Crystallography and Photonics" of Russian Academy of Sciences, 151 Molodogvardeyskaya st., 443001, Samara, Russia*

Abstract

In this paper, we propose methods designed to automate vectorization of point objects at cartographic image digitizing. These methods are based on the analysis of the skeleton-contour image representation and allow to detect point objects, estimate their parameters, and restore a regular grid of points. The experiments show high accuracy of the developed methods, which results in the possibility of using them in order to accelerate the process of digitizing cartographic images.

Keywords: vectorization; digitizing; cartographic image; point object; skeletonization; geoinformation system; GIS; skeleton-contour representation

1. Introduction

At present, the problem of digitizing important information from paper carriers is of great urgency [1]. These include old books, drawings, engravings important for the preservation of cultural heritage. Also, we can note plans, charts, and maps printed 20 or more years ago, which digital sources did not exist or were not preserved. Such documents in addition to historical value can have great practical importance. Thus, topographic maps and plans are an important data source for geoinformation systems (GIS) and can be used to solve a lot of practical problems of territory analysis and development: digital terrain model creation, maintenance of forest and water registers, identification of flooding areas and others [2]. So the subject of this paper is digitizing such cartographic images.

The simplest way to digitize paper maps is to scan them in bitmaps with subsequent coordinate binding in GIS. In such case, these data can be used as an underlying layer supplementing the information on a particular territory. However, they can not be used for solving any analytical tasks in GIS. For that, raster maps must be vectorized and become a part of the GIS database. In the past 20 years, a couple of methods and software tools were developed to simplify the process of digitizing bitmaps. For instance, we can mention such tools as ArcScan (in ArcGIS software), PowerTRACE (in CorelDRAW), EasyTrace, LineTracer, Spotlight, etc. However, cartographic images often contain various figures (topographic signs), and for their fast digitizing it is often necessary to develop new specialized methods and software tools.

The current paper is devoted to the development of methods for solving one of the particular problems in the described area: vectorization of point objects. In general, all the objects plotted on the maps can be divided into two main groups. The first one includes linear objects, defined by their axial line, drawing method, and thickness. At their digitizing, the most important task is to restore the axial line with high precision. For that, there are used methods based on either raster skeletonization [3, 4] or vector skeleton-contour representation [5]. For the first group, the contour line is less significant than the axial line. The second group consists of all other objects, which are topographic signs, letters, and objects of complex shape. For them, unlike the first group, the contour line is more important than the axial line. Point objects, formally related to the second group, are a special case, and for their exact digitizing, as will be shown later, both the contour and the axial line are of high importance.

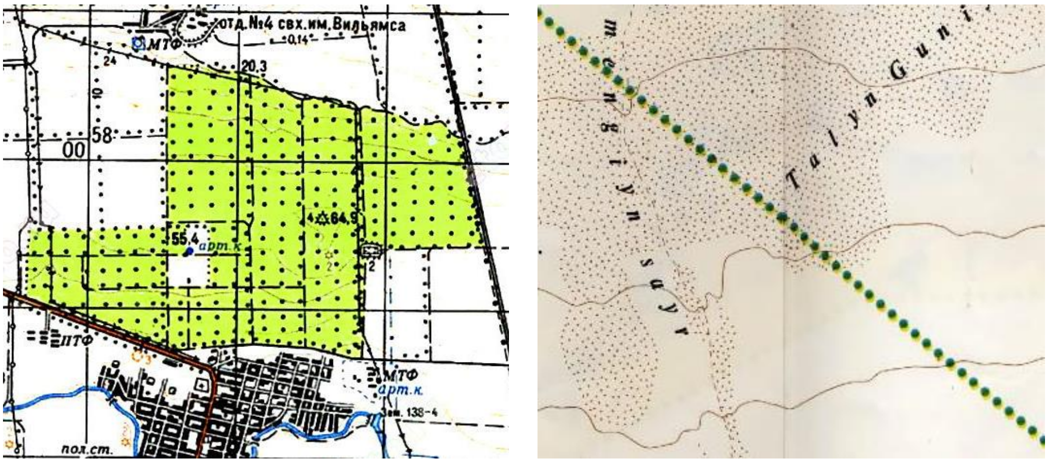
Point objects are quite common on paper maps. They are used, for example, for designation of plantations, bushes, elevations, sand, orchards, etc. (see Figure 1) [6]. When vectorizing in a GIS, each such object is associated with a point object, defined by its coordinates (x, y) and radius R. In this case, depending on the type of data source, the radius can be either arbitrary (for example, if the point objects depict sands) or constant (most other objects). Often the exact point radius is unknown or is not of great significance. In this case, point objects of the same type should be justified in size after the vectorization. Another common case is the strictly ordered placement of point objects on maps. This is typical, for example, for the designation of orchards or forest plantations (see Figure 1). In this case, vector objects in GIS should be placed on the terrain according to the same rule. Thus, we see some special cases in the problem of digitizing point objects. This fact necessitates the development of specialized algorithms for automated vectorization of point objects that is the goal of this paper.

The paper is organized as follows. The second section describes the general technology of cartographic image digitizing that we use and also denotes its stages requiring the development of specialized methods for point objects. Section 3 contains a description of the methods developed to detect such objects, restore their radius, and localize them on a continuous or discrete grid. The results of experimental studies of the methods developed are presented in Section 4. The paper ends with the conclusion and acknowledgments.

2. General technology of cartographic image digitizing

In our work, we used a typical technology of cartographic image digitizing that includes the following stages:

1. Preprocessing a bitmap.
2. Transformation of the image into the skeleton-contour representation.
3. Classification of objects to be vectorized.
4. Type-adapted object vectorization.
5. Correction of vector results based on analysis of object groups.



a) b)

Fig. 1. Examples of cartographic images with point objects.

At the first stage, the original color scanned image is converted into one or more binary images prepared for subsequent vectorization. Thus, the mandatory procedures used at this stage are color segmentation and binarization. The need for color segmentation is due to the fact that most maps contain several thematic layers displayed by predetermined colors (for example, blue is used for hydrography, red or brown for relief, green for vegetation, etc.) [7, 8]. Thus, the separation of the original image into thematic layers by the color feature will avoid the overlay of dissimilar objects and improve the digitizing quality. Binarization is necessary because of the need for subsequent skeletonization, which usually requires binary input data [5]. In addition to the two mentioned procedures, at the first stage, additional processing operations can be carried out to improve image quality: linear or nonlinear filtering for noise reduction, morphological processing for gluing lines etc.

The binary image obtained at the first stage can be considered as a set of figures. At the second stage, we estimate contours and skeletons for each of these figures. According to [5], the contour of a figure is an ordered set of vertices $\left\{\left(x^{(i)}, \mathrm{y}^{(i)}\right)\right\}_{i=1}^{n}$ defining a polygon approximating the boundary representation of the figure. Such a polygon can not have intersections with other contours and self-intersections. For a formal definition of a figure skeleton, we use a definition based on the concept of a maximal empty circle [5]. An empty circle of a figure A is a closed set of points $\overset{o}{S}_r(p)=\{q:\ q\,\mathrm{O}\,R^2,\ \ d(p,q)\,\mathrm{J}\ r\}$ such that $\overset{o}{S}_r(p)\,\mathrm{M}\,A$. This set is a circle of the radius $r\ \mathrm{i}\ \ 0$ with the center at a point $p\,\mathrm{O}\,R^2$. The maximum empty circle is an empty circle, which is not contained in any other empty circle. The skeleton of a figure is the set of centers of all its maximal empty circles.

The skeleton can be described by a flat graph [5], whose vertices are the centers of maximal empty circles having either one common point with the boundary of the figure or three and more points. In this case, the edges of a graph are lines that consist of the centers of those empty circles that touch the boundary exactly at two points. As shown in [5], the construction of a skeleton-contour representation for a raster image is associated with a number of difficulties. Therefore, in this paper, we used algorithms [9] developed by Leonid Mestetskiy and widely used in the papers of his school.

Since different methods of processing the skeleton-contour data are required for different object types (linear, point, area), at the third stage it is necessary to classify each object (figure) for correct processing performed at the fourth stage. The result of the fourth stage is the vector data, which can be further corrected in the fifth stage based on a priori known constraints on the geometry of similar objects. For example, many conventional topographic signs should have a constant size, and some groups of symbols must also be located at the same distance from each other [6]. The methods used at the stages 3-5 for vectorizing point objects are described in the next section.

3. Methods for point object vectorizing

3.1. Estimation of a circle radius

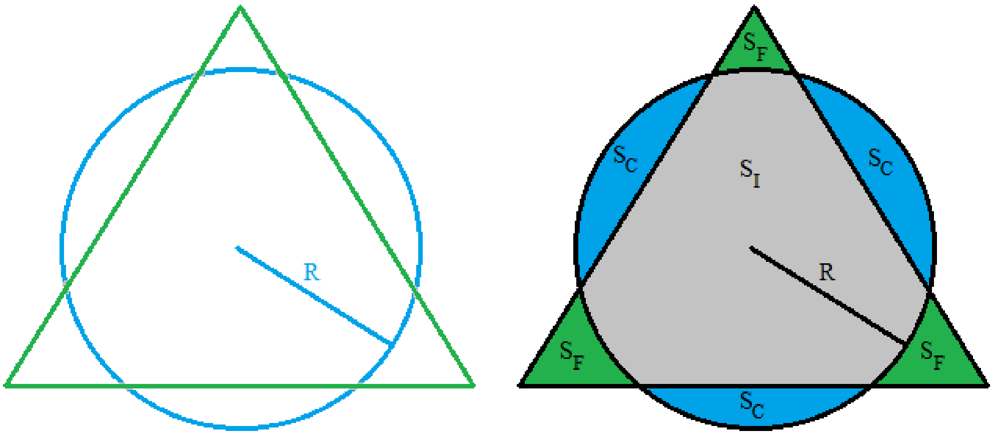
As studies have shown, to classify point objects (i.e. to check whether an object is point-like), it is useful to estimate the object radius. The initial approximation of the circle radius R_0 can be calculated based on the skeleton edge lengths $l_j,\ j=1..N$ and radiuses $r_{j,1}$ and $r_{j,2}$ of maximal empty inscribed circles for corresponding vertices. Let L be the length of the entire figure skeleton:

$$L = \mathop{\mathrm{e}}_{j=1}^{N} l_j .$$

Then

$$R_0 = \mathop{\mathrm{e}}_{i=1}^{N} \frac{r_{i,1} + r_{i,2}}{2} \frac{l_i}{L}.$$

Further, two different methods can be used to refine the radius of the circle: the method based on contour length, and the one based on figure area.



a) minimizing the perimeter deviation

b) minimizing the area deviation

Fig. 2. An illustration of two methods for the radius refinement. The figure to be analyzed is a triangle.

Method 1 is based on minimizing the deviation of the figure perimeter from the theoretical circumference based on the value of the radius r :

$$R = \arg\min_{\rho} \left(\left| \frac{D}{2\pi\rho} - 1 \right| \right), \tag{1}$$

where D is the length of the figure contour.

The problem (1) can be solved numerically (for example, using the golden section search) with the initial approximation $r = R_0$. Method 1 is illustrated in Fig. 2a, where the figure contour (shown as a triangle) and the theoretical circumference are highlighted in color.

Method 2 is based on minimizing the deviation of the figure area from the theoretical value of circle area, determined by the radius r. Let S be the figure area, S_I – the area of intersection of a circle with a figure, S_F – the figure area that is outside of the circle, S_C – the circle area that is outside of the figure (see Fig. 2b). We used the following accuracy characteristics:

$$a = \frac{S_I}{S_I + S_F} = \frac{S_I}{S}, \quad b = \frac{S_I}{S_I + S_C} = \frac{S_I}{pr^2}.$$

Let g be the harmonic mean of a and b :

$$g = \frac{2}{\frac{1}{a} + \frac{1}{b}} = \frac{2S_I}{S + pr^2}.$$

Then the radius R can be calculated by minimizing the deviation of g from unity:

$$R = \arg\min_{\rho} \left(\left| \frac{2S_{\Pi}}{S + \pi\rho^2} - 1 \right| \right). \tag{2}$$

The problem (2) can be solved by the same method as the problem (1).

3.2. Detection of point objects

To detect point objects on a map, we need to calculate some features and classify whether the object is from the point class. Our studies have shown that a set of only two features is sufficient and the linear separating function can be applied for classification. The first feature characterizes the deviation of the functional (1) (or (2)) from zero, thus, it can have one of two forms:

$$f_1^1 = \left|\frac{D}{2pR} - 1\right|,$$

$$f_1^2 = \left|\frac{2S_{\Pi}}{S + pr^2} - 1\right|.$$

To reduce the computational complexity, it is preferable to use the feature f_1^1 for the first method of a circle radius refinement and the feature f_1^2 for the second one.

The second feature is defined as the ratio of the skeleton length to the radius estimation:

$$f_2 = \frac{L}{R}.$$

For point objects, such ratio should be small, as shown in Fig. 3.

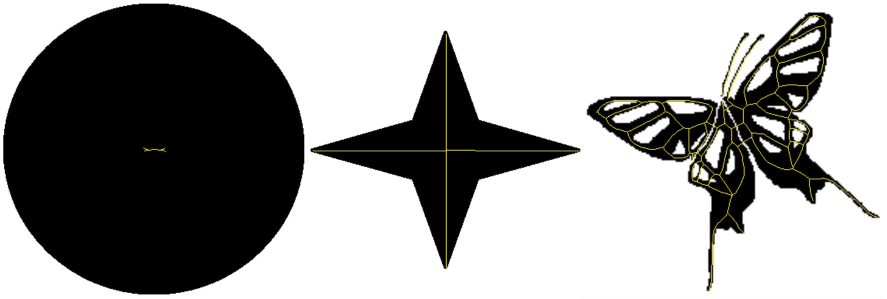


Fig. 3. Skeletons of figures with various shapes (figure shows that the skeleton length of a point object should be much less than the radius).

3.3. Reconstruction of a regular grid of objects

In Section 1 it was noted that some point symbols (for example, orchards, wooded areas) on maps follow the same distance from each other. Thus, the resulting vector objects must also keep a regular layout. However, this requirement is not guaranteed when objects are processed individually. To restore a regular object grid, the following method is proposed.

Let one of the grid nodes is located in the origin of coordinates. Then a two-dimensional grid can be defined by two vectors $\vec{V_1}$and $\vec{V_2}$ (which are often orthogonal to each other) (see Fig. 4). Then the coordinates $\left(x^{(i)}, y^{(i)}\right)$ of a particular grid node can be found as follows:

$$\left(x^{(i)}, y^{(i)}\right) = k_1\vec{V}_1 + k_2\vec{V}_2\,, \tag{3}$$

where $k_1, k_2 \in \mathbb{Z}$ are the indices of the grid node.

The initial approximation of the grid can be based on three adjacent support objects, one of which is central, and the other two specify the direction of the vectors $\vec{V_1}$and $\vec{V_2}$ (see Fig. 4). In practice, these support objects can be selected by the user. When reconstructing a regular grid of points, it is necessary to solve next three problems:

- refine the vectors $\vec{V_1}$and $\vec{V_2}$;
- detect all point objects lying on the grid

$$S = \left\{\left(x^{(i)}, y^{(i)}\right) : \left(x^{(i)}, y^{(i)}\right) = k_1\vec{V}_1 + k_2\vec{V}_2 \quad \forall k_1, k_2 \in \mathbf{Z}\right\}_i .$$

- correct the center coordinates of the found objects taking into account the refined vectors $\vec{V_1}$and $\vec{V_2}$

Here is the proposed algorithm:

<u>Step 0</u>. Set the initial approximations of vectors $\vec{V_1}$and $\vec{V_2}$; add 3 reference objects to the set S .

<u>Step 1</u>. Find the theoretical coordinates of all grid nodes not yet added to the set S and having at least one neighbor in the set S by the rule of four neighbors.

<u>Step 2</u>. Among all the point objects of the map, find objects that are located at a distance no more than the threshold value d from the theoretical coordinates found in Step 1. If at least one object is found, go to Step 3, otherwise – to Step 5.

A loop for all found objects (grid nodes). After the loop go to Step 1.

<u>Step 3</u>. Refine vectors $\vec{V_1}$and $\vec{V_2}$:

$$\vec{V}_1 = \frac{N_1\vec{V}_1 + \vec{A}_1}{N_1 + 1}, \quad \vec{V}_2 = \frac{N_2\vec{V}_2 + \vec{A}_2}{N_2 + 1}\,,$$

where $N_1, N_2 \in \mathbb{Z}$ are the counts of vectors used to estimate the vectors; $\overrightarrow{A_1}, \overrightarrow{A_2}$ are the estimations of vectors $\overrightarrow{V_1}$and $\overrightarrow{V_2}$ for current grid node by Eq. (3) and known indices k_1, k_2. If one of these indices is zero, then it is obvious that only one of the vectors $\overrightarrow{V_1}$and $\overrightarrow{V_2}$ is recalculated.

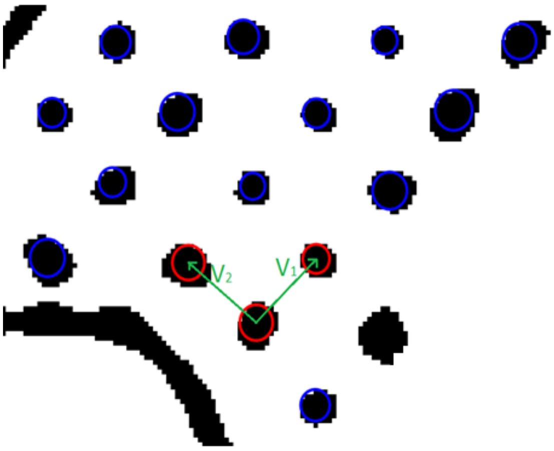


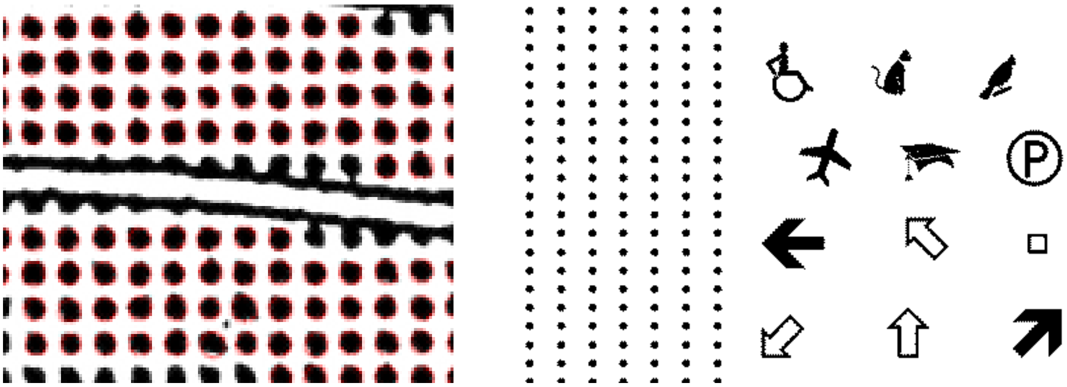
Fig. 4. An initial approximation of a grid of point objects.

Step 4. Add the current object to the set S .

Step 5. Recalculate the coordinates of the initial reference object (which corresponds to the indices $k_1 = k_2 = 0$). Go to Step 1. Finish the algorithm if no objects have been added at step 4.

4. Experimental research

For the experiments, we primarily used a scanned image containing many regularly located point objects with some linear ones (see Fig. 5a). For this image, a manual vectorization was performed, as well as automatic vectorization using the proposed methods. The results of manual vectorization were considered as reference ones. Also, during the experiments, a synthesized image was used (see Fig. 5b), for which the exact number of point objects and their locations are known.



a) scanned image b) synthesized image

Fig. 5. Fragments of the test images.

4.1. Detection and localization of point objects

Since the average point size influence on the parameters of a point classifier and we had only two test images of different origin, we decided not to train a particular classifier (which would have to be trained on a fragment of the same image). Instead, several fixed threshold values Δ_1^1 (or Δ_1^2) and Δ_2 for features f_1^1 (or f_1^2) and f_2 were considered, and the classification was performed according to the rule

$$f_1^1 < \Delta_1^1 \left(f_1^2 < \Delta_1^2\right)$$

The value of F-measure was used as a quality measure.

During the experiment on the real image, for the refinement method 1, the largest value of F-measure was achieved at Δ_1^1= 0.35, Δ_2= 3.5 and was equal to 0.9943. For the refinement method 2, the largest value was achieved at Δ_2^1= 0.2, Δ_2= 4.5 and reached 0.9957. Fig. 6 shows how Δ_1^1 (for the fixed) Δ_2= 3.5 and Δ_1^2 (for fixed) Δ_2= 4.5 influence on F-measure. As can be seen from the graphs, for a wide range of values, the F-measure has very high values, which confirms the possibility of separation of point objects in this image.

In addition, Table 1 specifies some other statistics. As can be seen, the second method showed a somewhat higher accuracy both in detecting the points and in estimating their location.

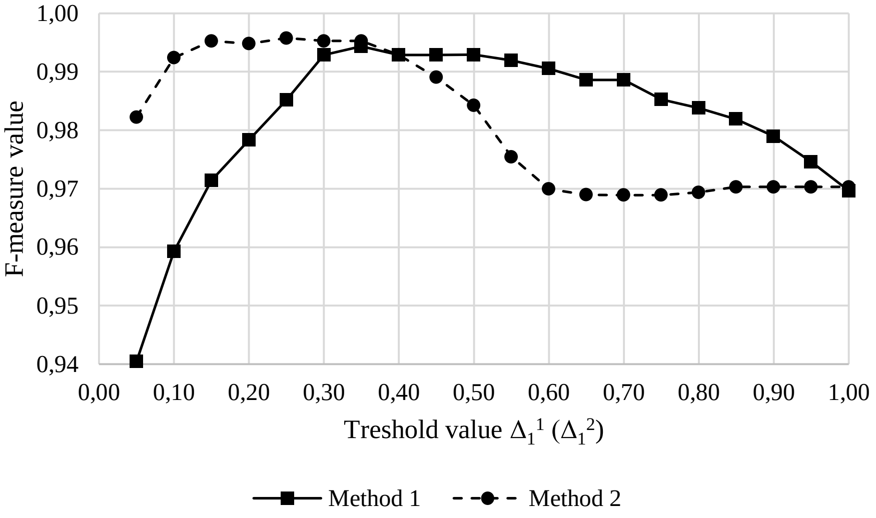


Fig. 6. Dependencies of F-measure on features f_1^1 (f_1^2).

Table 1. Comparison of the best results for two methods of radius refinement.

Characteristic	Method 1	Method 2
Number of detected objects	1056	1059
Number of false positives	6	6
Number of missed objects	6	3
F-measure for hits of the center of a point inside the true contour of a point	0.9953	0.9957
Median deviation of the centers of circles, pixels	1.065	0.7908
Median radius deviation, pixels	2.3146	1.1011

For the synthesized image (Fig. 5b), in a sufficiently large range of threshold values, it is possible to separate all 2243 point objects from all 326 figures having other shapes by both methods.

4.2. Reconstruction of a regular grid

Testing of the grid reconstruction method was carried out using a fragment of the image from Fig. 5a, containing 147 circles, with $d = 10$ pixels. As the measure of the method effectiveness, the number of objects matched with the grid nodes and the total deviation of their centers from the grid nodes were used.

During the experiment, at the first iteration (steps 0-4, until the coordinates of the support objects were refined), all circles were found which should lie on the grid, and the total deviation was 11.02 pixels (for 147 objects). At the second iteration (after changing the coordinates of the reference object), the deviation was 6.47 pixels, and the total deviation was reduced to 6.41 pixels.

Figure 7 shows the results of processing after the first iteration (Fig. 7a) and the final result (Fig. 7b). For comparison, both figures show the initial contours of points before the procedure starts. The result in Fig. 7b has a smaller deviation than initial approximation on Fig. 7a.

5. Conclusion

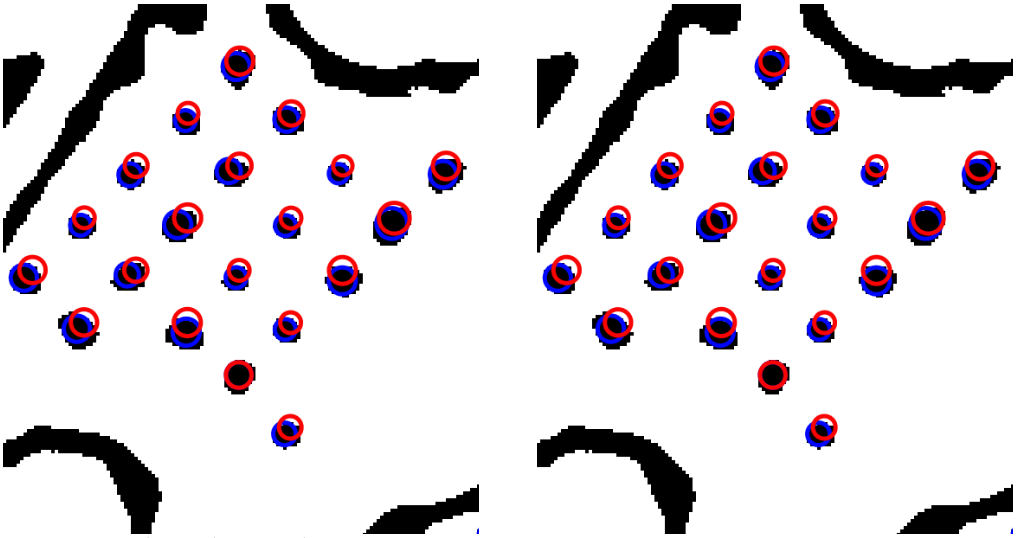
In this paper we have proposed a number of algorithms which use the skeleton-contour image representation:

1. the algorithm for estimating the radius of a point object;
2. the algorithm for point objects classification;
3. The algorithm for reconstructing a regular grid of objects.

The experimental study has shown, that the proposed methods have high accuracy and allow significantly accelerate the process of raster map digitizing.

Acknowledgements

Research has been supported by the RFBR grants (projects No. 15-07-05576 and 16-41-630676). During the research, BSTransLib library by Leonid Mestetskiy was used [9]. The authors express their gratitude to the colleague Daria Terentyeva, who performed manual vectorization of the test images, which allowed to estimate the accuracy of the developed methods.



a) after the first step of processing b) the result

Fig. 7. Reconstruction of a regular grid.

References

[1] Conway P. Overview: Rationale for digitization and preservation. Handbook for digital projects. Andover, Massachusetts: NEDCC, 2000: 5–20.

[2] Awange JL, Kyalo Kieam JB. Environmental geoinformatics. Springer-Verlag Berlin Heidelberg, 2013; 541 p. DOI: 10.1007/978-3-642-34085-7.

[3] Zhang TY, Suen CY. A fast parallel algorithm for thinning digital patterns. Communications of ACM 1984; 27(3): 236–239. DOI: 10.1145/357994. 358023.

[4] Oka S, Garg A, Varghese K. Vectorization of contour lines from scanned topographic maps. Automation in Construction 2012; 22: 192–202. DOI: 10.1016/j.autcon.2011.06.017.

[5] Mestetskiy LM. Continuous morphology of the binary images: the figures, skeletons, circulars. Moscow: Fizmatlit, 2009; 288 p.

[6] Topographic map symbols for the scales 1:5000, 1:2000, 1:1000, 1:500. Moscow: KartGeoCenter, 2005; 287 p. (in Russian)

[7] Chiang Y-Y, Leyk S, Knoblock CA. A survey of digital map processing techniques. ACM Computing Surveys 2014; 47(1): 1–44. DOI: 10.1145/2557423.

[8] Liu T, Miao Q, Tian K, Song J, Yang Y, Qi Y. SCTMS: Superpixel based color topographic map segmentation method. Journal of Visual Communcation and Image Representation 2016; 35: 78–90. DOI: 10.1016/j.jvcir.2015.12.004.

[9] Continuous morphological models and algorithms (a lecture course by Mestetskiy LM). URL: http://www.machinelearning.ru/wiki/index.php?title=Morphmodels (16.05.2017).