DOI 10.15622/sp.60.7

S.YU. MIROSHNICHENKO, V.S. TITOV, E.N. DREMOV, S.A. MOSIN

HOUGH TRANSFORM APPLICATION TO DIGITIZE RECTANGULAR SPATIAL OBJECTS ON AEROSPACE IMAGERY

Miroshnichenko S.Yu., Titov V.S., Dremov E.N., Mosin S.A. **Hough Transform Application to Digitize Rectangular Spatial Objects on Aerospace Imagery.**

Abstract. The paper describes the method of development for the remote sensing data processing to speed up the digitizing workflow. The method is designed to digitize rectangular objects using their approximate spatial positions and provides an automatic estimation of the orientation and aspect ratio.

The paper contains a formal statement of the problem of digitizing an object with the desired geometric shape using it's apriori known spatial position on a source image. The method creates polygonal representations of rectangular spatial objects from one or a few reference points set by an operator. It is based on source image's pixels clustering using spectral bands as a feature space. The following Hough transform incorporates local direction of intensity gradient to estimate object's orientation and reduce computational complexity together with low-pass filtering within an accumulation process to improve robustness. It is shown that the developed method can be modified to digitize objects of any analytically described shape.

The method is designed to allow easy user interaction without any significant delays and to provide transparent and predictable control of an output object's polygon size.

To investigate the developed method a test dataset with more than 700 rectangular objects was used. The root-mean-square error of object's points positioning, mean rotation error in polar coordinates and a Jaccard index were used to measure a precision of the digitized objects. The experiment results demonstrate that digitizing workflow is accelerated by 25–40% using the software implementing the developed method without a significant precision loss.

Keywords: remote sensing data, automated digitizing, image processing, hough transform.

1. Introduction. Current stage of Earth remote sensing systems' (RSS) development is marked with the continuous growth of their quantity. Modern RSS sensors gain better spatial resolution and a wider range of spectral channels. Together these facts turn to a rapidly growing stream of remote sensing data (RSD) that should be processed and georeferenced to have sufficient precision for further use [1, 2]. RSD processing workflow also should be accelerated to prevent airborne and space imagery lose their actuality. Digital maps are one of the RSD secondary processing main products. Operators create and update digital maps using interactive software instruments (interactive method) that do not imply any assistance to recognize objects or detect their boundaries. An operator digitizes image by finding and recognizing objects using his skills and experience. He defines each object's shape, spatial position, orientation, type and then sequentially places vertexes of a polygon that bounds the object from others and an underlying surface. An operator's actions within the confines of digitizing workflow are guided by editorial and technical requirements that change due to spatial scale and a resulting map's field of application. These requirements regulate object's level of detail, acceptable shape variants, a minimum distance between neighboring objects, etc. for each thematic layer the map contains. A digitizing workflow is laborious; the precision of the result directly depends on operator's skills, experience, concentration and gradually reduces during work time due to his tiredness caused by monotony and diligence of performed actions.

There are some software products for automated RSD processing that provide wizards to create individual rules and step-by-step sequences for spatial objects digitizing and recognition [3-6]. These products are based on object-oriented detection [7] and are aimed to accelerate digital maps creation workflow, but, in practice, in spite of declared features their range of application comes down to greenery and hydrography objects digitizing and calculative and estimative problems solution. The limited range of applications results from the fact that automated RSD processing software has a limited ability to follow editorial and technical requirements and to transform objects' polygonal representations to a demanded geometric shape.

Figure 1 shows the result of space image automated digitizing comparing to the interactive method. Source image on Figure 1a is separated to clusters depending on contour and texture features (Figure 1b) and used further to extract the buildings and structures subset (Figure 1d) with kNN method [8]. Figure 1c displays ground truth polygons of the same objects that were created in the interactive mode according to common editorial and technical requirements. Figure 1c displays ground truth polygons of the same objects created in the interactive mode according to common editorial and technical requirements. The automated result is an object's "draft" (Figure 1e shows a difference from the interactively created ground truth polygons) rather than a digital map's item and is only suitable to estimate the number of objects of the desired type, their spatial positions, occupied area, etc. For now, Jaccard index is a de facto standard quantitative measure of RSD automated recognition. The results of automated digitizing require per-object revision and correction to satisfy the minimum precision demands and lead to even greater operator's efforts than a map's interactive creation from scratch.

Results of investigations and competitions devoted to automated RSD processing demonstrate domination of approach based on deep learning artificial neural networks. Modern neural network architectures combined with GPGPU computing technologies allow to find practical solutions of the following space imagery recognition problems: to detect objects of a priori specified type (for instance roads in [10] and vehicles in [11-12]) or several classes (up to 10 in [9, 13-16]), to find and count certain kinds of animals [17-18]. However, we should mention that all developments in the field of deep learning neural networks are in a stage of experimental investigation far from being incorporated into commercial software products and for now are available only for experts in machine learning.

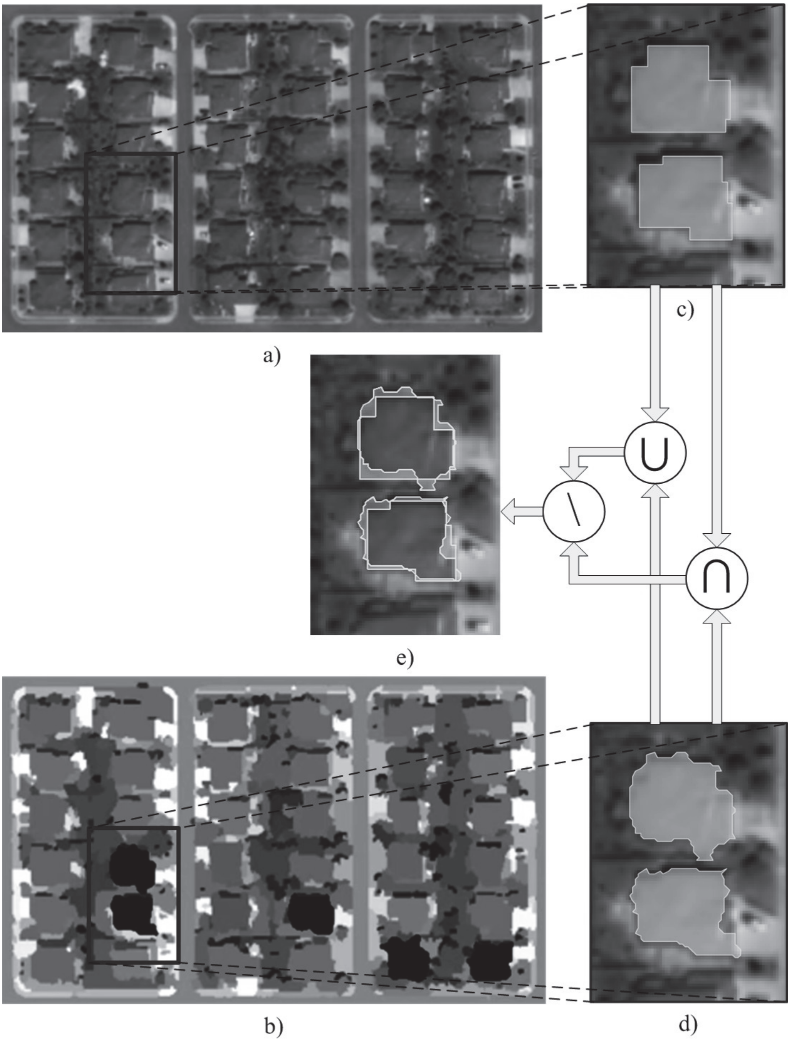


Fig. 1. The result of space image’s (a) automated buildings digitizing (b, d) comparing to the interactive one (c) using Jaccard Index, visually represented on (e)

Another approach to decrease time costs for RSD processing lies in the development of automated software tools designed to be easily integrated into existing digitizing workflow and help users to perform the most laborious and tedious operations. Also, these tools can transform objects' polygons to the desired shape according to editorial and technical requirements.

ERDAS IMAGINE software contains the "Region grow" tool that in combination with a set of vector cleanup operations provide a user-controlled automated digitizing [19]. However, this tool has significant computational complexity (a created polygon appears through at least a couple of seconds after the user action). A vector cleanup process performs after the digitizing was finished for the whole layer and requires more operator's efforts to control output precision compared to the object-to-object approach. Feature Analyst contains a toolset for the assisted extraction that allows a user to digitize building by merely dragging a rectangle that encompasses it [20]. The shortcoming of this tool is that the object must be oriented strictly along the screen axes. Our paper focuses on the RSD digitizing process speedup by development and investigation of a new method designed to create rectangular polygonal representations with automatic estimation of orientation and aspect ratio based on the corresponding objects' approximate position.

2. Problem formalization. The essence of the digitizing process is to create a bounding polygon v for each recognized object on the source image $\mathbf{I}$. The bounding polygonal representation (further — polygon) is the polygon separating the object from others and an underlying surface. A priori information referring to the object is obtained by a preliminary recognition by an operator and includes the following:

– set of reference points (Figure 2a) representing an approximate spatial position of the object to be described by the polygon v;

– object's shape s that guides a selection of the corresponding digitizing method.

Hence, automated digitizing the object with the rectangular shape s_{sq} is described by:

$$v = F_{sq}\left(\mathbf{I}, \mathbf{R}, t\right)\Big|_{s=s_{sq}}, \tag{1}$$

where t — is a threshold value regulating a size of a produced rectangular polygon.

Source image $\mathbf{I}$ contains K spectral bands, and each is described by a discrete brightness field with linear sizes M x N:

$$\mathbf{I} = \mathbf{f}\left(x, y\right) = \left(f_k\left(x, y\right)\right)_{k=1}^{K},\ x = \overline{1, M}, y = \overline{1, N}, \tag{2}$$

where $\mathbf{f}\left(\cdot\right)$ — is a vector of brightness values in the pixel $\left(x, y\right)$.

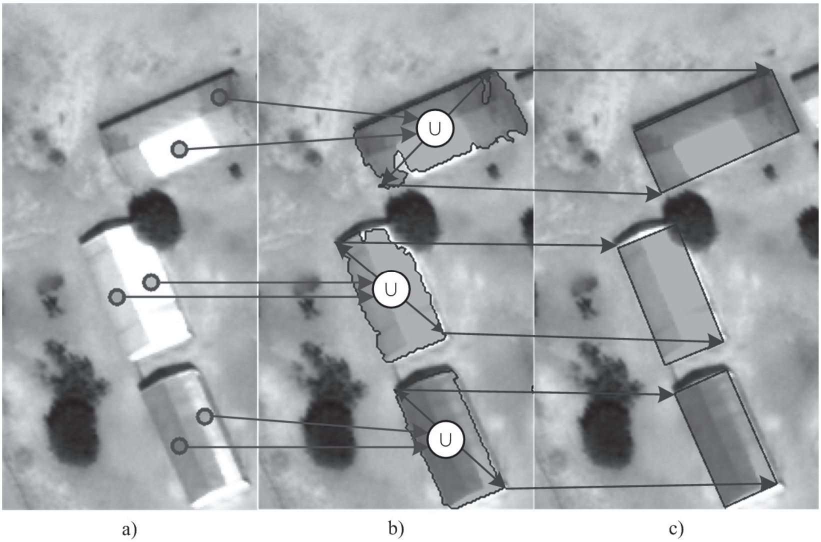


Fig. 2. Automated digitizing process: reference points (a) are used to detect corresponding objects (b) and then to transform them to a required (rectangular in our case) shape (c)

Reference points of **R** are set in the source image local coordinate system and can be readily transformed into geodetic or projected coordinates using appropriate matrix [1]:

$$\mathbf{R} = \left\{ \left(x_j^{(\mathbf{R})}, y_j^{(\mathbf{R})} \right) \right\}_{j=1}^{N_{\mathbf{R}}}. \tag{3}$$

Object polygon v (Figure 2c) is described by a closed sequence of points in the source image local coordinate system:

$$v = \left(x_i^{(v)}, y_i^{(v)} \right)_{i=1}^{N_v}, \tag{4}$$

where N_v — is a number of points in v, which for a rectangular object equals four.

To get a ground truth digital map, we have to make a parallel projection of every terrain object to an underlying surface with a normal vector $\vec{n}$. For a perspective image (created with a sensor whose optical axis is set by a vector $\vec{o}$ having a non-zero angle with $\vec{n}$) without a digital surface model (DSM) of sufficient spatial resolution, we encounter coordinate distortions $\left(\Delta x_i^{(v)}, \Delta y_i^{(v)}\right)$ shown at Figure 3a. For that reason, the paper considers only orthorectified images which have measurement properties [1] and can be used to create digital maps directly (Figure 3b).

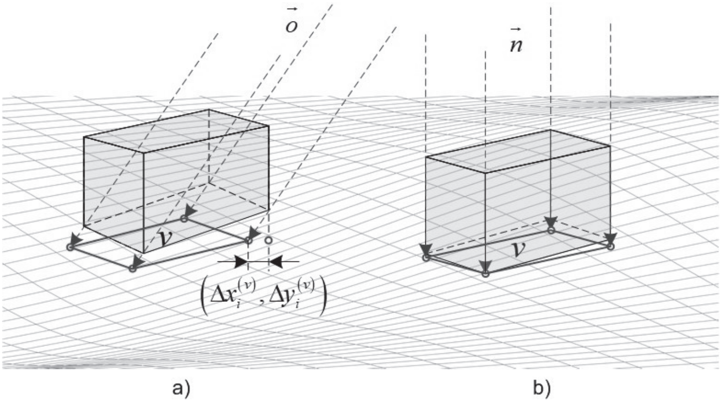


Fig. 3. Creating a polygon from a perspective image without a DSM (a) and from an orthorectified image (b)

Polygon (4) of a rectangular object contains four intersection points of four pairwise parallel and pairwise perpendicular straight lines (Figure 4). Coordinates of intersection points are calculated by a solution of the following system of linear equations:

$$\begin{gathered}
\mathbf{A}\cdot\left[x_1^{(v)}, y_1^{(v)}\right]^T = \left[r_1, r_3\right]^T,\ \mathbf{A}\cdot\left[x_2^{(v)}, y_2^{(v)}\right]^T = \left[r_1, r_4\right]^T, \\
\mathbf{A}\cdot\left[x_3^{(v)}, y_3^{(v)}\right]^T = \left[r_2, r_4\right]^T,\ \mathbf{A}\cdot\left[x_4^{(v)}, y_4^{(v)}\right]^T = \left[r_2, r_3\right]^T, \\
\mathbf{A} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ \cos\left(\alpha - \dfrac{\pi}{2}\right) & \sin\left(\alpha - \dfrac{\pi}{2}\right) \end{bmatrix}.
\end{gathered} \tag{5}$$

3. Developed method. The following requirements guided the method development process:

– the method has to implement functional (1) with the input parameters described by (2), (3) and output has the form of (4);

– software tool based on the developed method should allow comfortable interaction without any significant (clearly visible by an operator) delays. Hence a time of a single execution of (1) should be less than 500 мs on a workstation with Intel Core i3 processor.

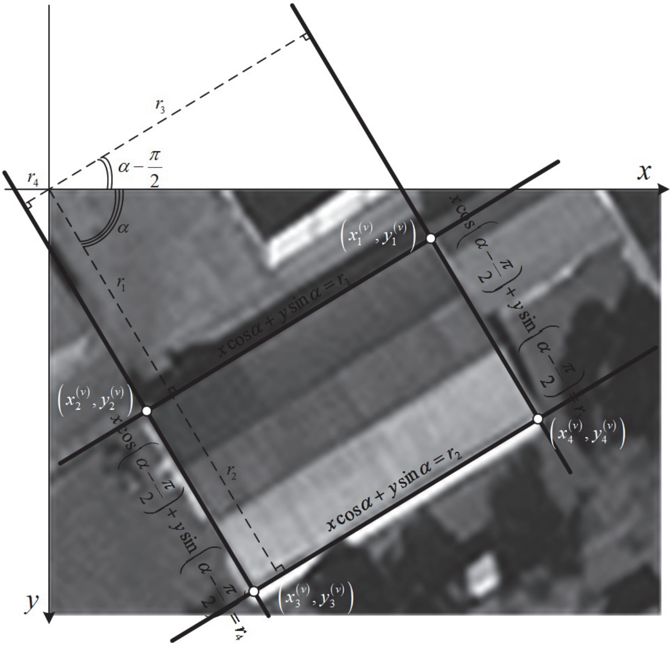

r_3
r_4
$\alpha-\frac{\pi}{2}$
x
α
r_1
$\left(x_1^{(v)}, y_1^{(v)}\right)$
$x\cos\left(\alpha-\frac{\pi}{2}\right)+y\sin\left(\alpha-\frac{\pi}{2}\right)=r_3$
$x\cos\alpha+y\sin\alpha=r_1$
$\left(x_2^{(v)}, y_2^{(v)}\right)$
$\left(x_4^{(v)}, y_4^{(v)}\right)$
r_2
$x\cos\alpha+y\sin\alpha=r_2$
$x\cos\left(\alpha-\frac{\pi}{2}\right)+y\sin\left(\alpha-\frac{\pi}{2}\right)=r_4$
y
$\left(x_3^{(v)}, y_3^{(v)}\right)$

Fig. 4. Polygon of a rectangular object contains four intersection points of four pairwise parallel and pairwise perpendicular straight lines

– the method should provide transparent and predictable control of an output polygon size using a threshold: a higher t value corresponds to a greater area of a produced polygon.

Choice of Hough transform [21] to implement a rectangular objects' digitizing is explained by fusion of its ability to detect straight lines forming an output rectangles' sides with acceptable computational complexity providing an interactive operation mode for a developing software tool. However, a classic Hough transform based straight lines detector combined with edge detection [22] as an object boundaries extractor do not grant to an operator a transparent model to control sizes of output polygon (4) using a threshold value t.

To satisfy the mentioned above requirements, we have chosen a two-stage scheme consisting of the following sequentially performed functions:

1. The function $f_c(\cdot)$ creates a cluster c representing digitized object's boundaries visible on a source image $\mathbf{I}$ (Figure 2b). A function output depends on a reference points set $\mathbf{R}$ and a threshold value t:

$$c = f_c(\mathbf{I}, \mathbf{R}, t). \qquad (6)$$

2. The function $f_{sq}(\cdot)$ converts a previously obtained cluster c to a rectangular polygon v using a Hough transform (figure 2c):

$$v = f_{sq}(\mathbf{I}, c). \tag{7}$$

We have knowingly divided the functional (1) into functions (6) and (7), besides the advantages described above, to gain an ability to convert a detected on an image cluster to any analytically described shape by merely replacing function (7) with the desired type of Hough transform.

3.1. Object detection. A starting point of function (6) is a reflection of a reference points set $\mathbf{R}$ into a set $\mathbf{C}^{(0)}$ of initial cluster values (Figure 5a):

$$\mathbf{R} \to \mathbf{C}^{(0)}, \mathbf{C}^{(0)} = \left\{c_i^{(0)}\right\}_{i=1}^{N_\mathbf{R}}, \; c_j^{(0)} = \left\{\left(x_0^{(c)}, y_0^{(c)}\right)\right\} = \left\{\left(x_j^{(\mathbf{R})}, y_j^{(\mathbf{R})}\right)\right\}. \tag{8}$$

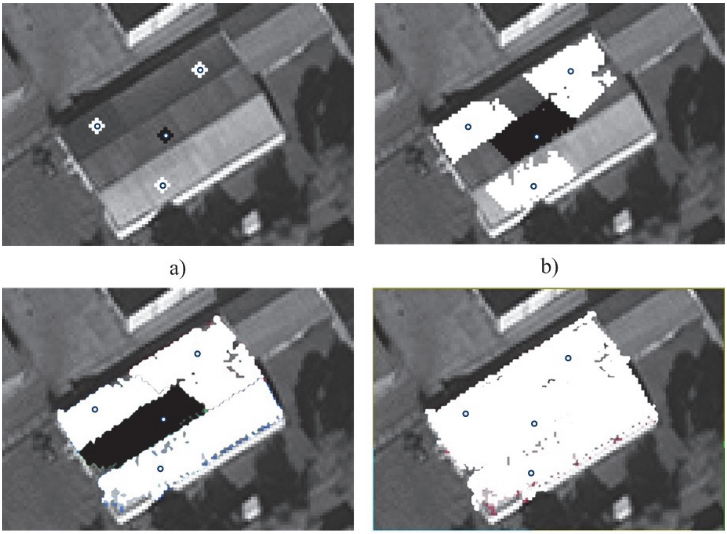


Fig. 5. An illustration of a reference points reflection into the set of initial clusters (a), their growth (b shows the result of 16th iteration, c – is the final 62nd iteration) and merging to the resulting cluster (d)

Next, an iterative cluster-growing function $f_{cg}(\cdot)$ (Figure 5b) on each m-th step to all items of $\mathbf{C}^{(m)}$ adds that pixels meet the following demands:

– do not belong to any cluster on the m-th iteration;

– situated on the current border of a cluster $c_j^{(m)}$;

– a distance value $d(\cdot)$ of a pixel in a feature space to a cluster's $c_j^{(m)}$ center does not exceed a threshold t set by an operator:

$$\mathbf{C}^{(m+1)} = \left\{c_i^{(m+1)}\right\} = \left\{f_{cg}\left(c_i^{(m)}, t\right)\right\}_{j=1}^{N_{\mathbf{C}}^{(m)}},$$

$$f_{cg}\left(c_i^{(m)}, t\right) = c_j^{(m)} \cup \left\{\begin{array}{l} \forall (x,y) \in \mathbf{I} : (x,y) \notin \mathbf{C}^{(m)} \\ \wedge (x \pm 1, y \pm 1) \in c_i^{(m)} \\ \wedge d\left(c_i^{(m)}, (x,y)\right) < t \end{array}\right\}. \tag{9}$$

We used maximum brightness difference to incorporate all K spectral bands of the image (2) as a feature space to calculate distance $d(\cdot)$:

$$d\left(c_i^{(m)}, (x,y)\right) = \max_k \left| \frac{1}{N_i^{(c)}} \sum_i f_k\left(x_i^{(c)}, y_i^{(c)}\right) - f_k(x,y) \right|_{k=1}^{K}. \tag{10}$$

Further feature space extension by edges, textures or spectrums on the one hand increases object detection precision but on the other hand — complicates distance function (10) and requires preliminary computations to obtain the mentioned features that in total do not comply with the interactivity requirement.

Iterations number of cluster-growing function (9) doesn't have predefined limit. Growing process stops in case no new image pixels can be added to any cluster within a current iteration (Figure 5c):

$$\mathbf{C} = \mathbf{C}^{(m)} : \bigcup_{i=1}^{N_{\mathbf{C}}^{(m-1)}} f_{cg}\left(c_i^{(m-1)}, t\right) \setminus c_i^{(m)} = \varnothing. \tag{11}$$

The final operation of function (6) is a merge $f_{cm}(\cdot)$ of clusters, containing in a set $\mathbf{C}$, into a resulting cluster c (Figure 5d). Clusters pair c_i, c_n merging requires them to have joint or adjacent pixels:

$$c_i = f_{cm}\left(c_i, c_n\right) = c_i \cup c_n \left| \forall (x,y) \in c_i : (x \pm 1, y \pm 1) \in c_n \right.. \tag{12}$$

If a resulting set $\mathbf{C}$ after the merge (12) contains more then one item (i.e., the threshold value is not enough to cover feature space distances (10) between all reference points to produce a single cluster), there are three options of action:

1. Accept a cluster of $\mathbf{C}$ with the maximum area as a result of the function (6):

$$c = f_c\left(\mathbf{I},\mathbf{R},t\right) = \arg\left[\max_i \left|c_i\right|\right]. \tag{13}$$

2. Let the result of (6) contain all produced clusters and transform each of them into a rectangular polygon using (7).

3. Continue an iterative execution of the function (9) and increase threshold value t every time the growing process stops due to condition (11) fulfill and a cluster set $\mathbf{C}^{(m)}$ contains more than one item. The developed method uses the first option because it allows an operator to concentrate on single object detection with sufficient visual control of the workflow together with the least computational complexity.

3.2. Object transformation to a rectangular polygon. To construct rectangular polygon (7), we have to find a solution of four linear equations systems (5), which, in turn, requires to obtain parameters $\alpha, r_1, r_2, r_3, r_4$ of straight lines bounding an object represented by a previously detected cluster c. As said before the mathematical basis for function (7) implementation is a Hough transform that converts source image's $\mathbf{I}$ coordinate field to an accumulator plane $h(\cdot)$ whose quantized dimensions represent parameters α, r of a straight line equation in Hesse normal form. Each pixel (x, y) appears as a curve in a plane $h(\cdot)$ by calculating the parameter r from all values α_n of the independent parameter α within a range $\left[\alpha_{\min}, \alpha_{\max}\right]$:

$$\begin{gathered} h\left(\alpha_n, r\right) = \sum_{(x,y)} \sum_{\alpha_n=\alpha_{\min}}^{\alpha_{\max}} H\left(x\cos\alpha_n + y\sin\alpha_n, r\right), \\ H\left(a,b\right) = \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases}. \end{gathered} \tag{14}$$

For interactivity reasons, an accumulator plane $h(\cdot)$ is filled only by pixels (x, y) situated on the external boundary of the cluster c:

$$\exists\left(x,y\right) \in c\ \exists\left(x_b, y_b\right) \notin c : \left|\left(x,y\right) - \left(x_b, y_b\right)\right| \leq 1. \tag{15}$$

Transform (14) has an asymptotic described by $O\left(p_c \cdot N_\alpha\right)$, where p_c — is an external perimeter of the cluster c. To reach a precision of object's spatial position and orientation, sufficient for cartographic products, we should set a quantity N_α of parameter's α quants in proportion to p_c so, that a rotation of radius vector from coordinate system origin to any cluster boundary pixel by an angle $\left(\alpha_{\max} - \alpha_{\min}\right)/N_\alpha$ leads to shifting of its ending to a distance not exceeding one pixel.

Hence N_α is a function of p_c and furthermore an asymptotic of (14) corresponds to $O\left(p_c^2\right)$, that do not meet the interactivity requirement.

To obtain a linear asymptotic function, we reduced the number of accumulator plane $h(\cdot)$ dimensions by calculating the parameter α directly from a source image as a mode $Mo(\cdot)$ of gradient vector angle φ [22, 23] near the cluster's external boundary (15) within a range of $\left[0, \pi/2\right)$:

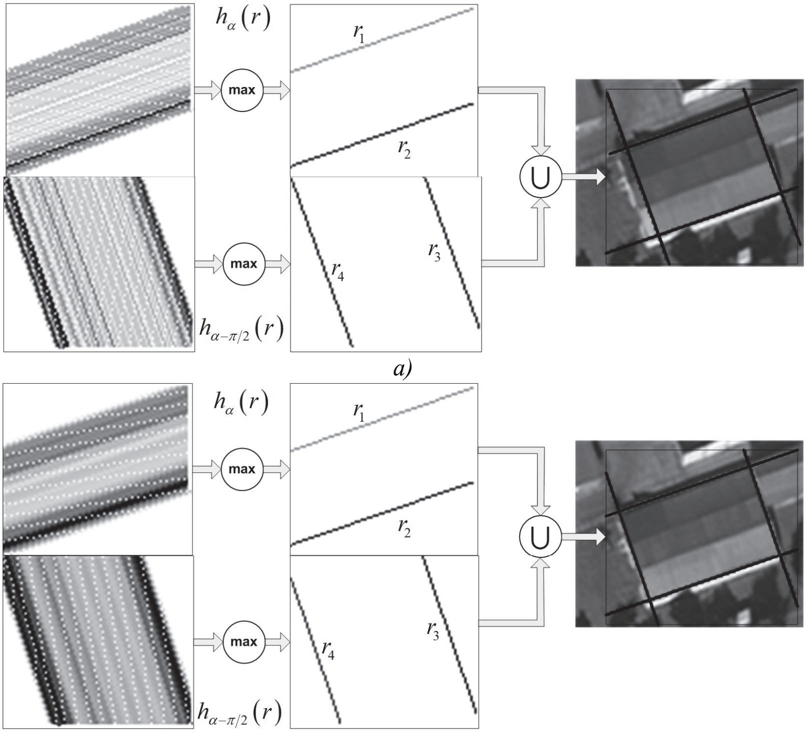
$$\alpha = Mo\left(\begin{cases}\varphi, \ \varphi < \pi/2\\ \varphi - \pi/2, \ \varphi \ge \pi/2\end{cases}\right),$$
$$\varphi = \angle\left(\nabla \overline{f}(x, y), \overrightarrow{Ox}\right). \tag{16}$$

Now when we have a fixed parameter α any pixel (x, y) of cluster's boundary transforms to a single value of r. Values $r_1 - r_4$ are calculated through two one-dimensional accumulators $h_\alpha(r)$ and $h_{\alpha-\pi/2}(r)$, each of what corresponds to a particular pair of straight lines:

$$h_\alpha(r) = \sum_{(x,y)} w_G\left(x\cos\alpha + y\sin\alpha - r\right), \tag{17}$$

where $w_G(\cdot)$ — is a Gauss function [24], applied as a low-pass filter to improve Hough transform robustness, and, to be more specific, to reduce an adverse effect of image's discreteness and α parameter calculation error on an output polygon precision.

Figure 6a illustrates a direct accumulation of r values similarly to (14) that leads to a r_4 parameter calculation error and the following incorrect position of one of the building's walls. Figure 6b shows an accumulation of low-pass blurred values (17) that provides more robust boundaries detection of a digitized object.



a)

b)

Fig. 6. An illustration of direct values accumulation (a) that leads to a parameter calculation error and to the subsequent incorrect position of building's wall compared to an accumulation of low-pass blurred values (b) that provides more robust boundaries detection

Filled up accumulators are used to select two pairs of parameter values $h_\alpha(r)\rightarrow(r_1,r_2)$, $h_{\alpha-\pi/2}(r)\rightarrow(r_3,r_4)$, and each corresponds to a maximum integral weight in view of a distance between straight lines:

$$(r_1,r_2)=\arg\left[\max_{\substack{r_i\in\left[r_{\max},r_{\min}\right]\\ r_j\in\left[r_{\max},r_{\min}\right]}}\left(\frac{\left(h_\alpha(r_i)+h_\alpha(r_j)\right)\cdot\left|r_i-r_j\right|}{r_{\max}-r_{\min}}\right)\right]. \qquad (18)$$

Parameters α,r_1,r_2,r_3,r_4 obtained through (16) and (18) are subsequently used to find a solution of linear equation system (5) and to create a resulting polygon in a form (4).

3.3. Method implementation. The implementation of our method has the following features and particular qualities to comply with requirements to interactivity and control predictability described in Section 3:

1. Cluster creation (6) uses only a part of the source image that is visible to an operator and has a screen zoom level. Thus sizes of a processed image part never exceed a monitor resolution that at the moment in most cases is equal to FullHD. A preliminary pyramid computation [25] provides a fast image overviewing for zoom levels that is less the original. If a working area has a zoom level more then 100%, a source image will not be scaled for processing as standard scaling algorithms would supply no additional data.

2. Only K available spectral bands of the source image $\mathbf{I}$ form a feature space to distinguish an object from its neighbors and the underlying surface. A cluster stores current mean brightness values for (10) as its attributes and updates them on adding new pixels.

3. Mode of gradient vector's angle (16) is computed by Sobel operator combining a sufficient edge detection precision with a low computational cost compared to other operators and Gaussian derivatives [26, 27]. To further maximize performance a module and an angle of gradient vector are obtained only in the neighborhood (15) of current cluster's external boundary. A histogram with a step equal to 1 deg is applied to select a mode value (we chose the step value empirically). Histogram accumulates pairs "gradient angle"→"gradient module" with the following selection of an angle value coheres with the maximum module sum.

4. To reduce an operator's need to interact with a visual interface of software tool implementing our method we transferred a threshold value control to a wheel of a mouse manipulator. Figure 7 illustrates a threshold value effect on a resulting cluster. A further increase of the value over shown on Figure 7d is redundant and leads to cluster growth beyond object's visible boundaries. After one polygon was created, a current threshold value can be applied to digitize another similar object without adjusting to further reduce the number of an operator's actions.

5. During the digitizing, the external boundary is displayed together with corresponding object's polygon as shown on Figure 8 to improve visual perception of the process by an operator and to enhance the comfort of his work.

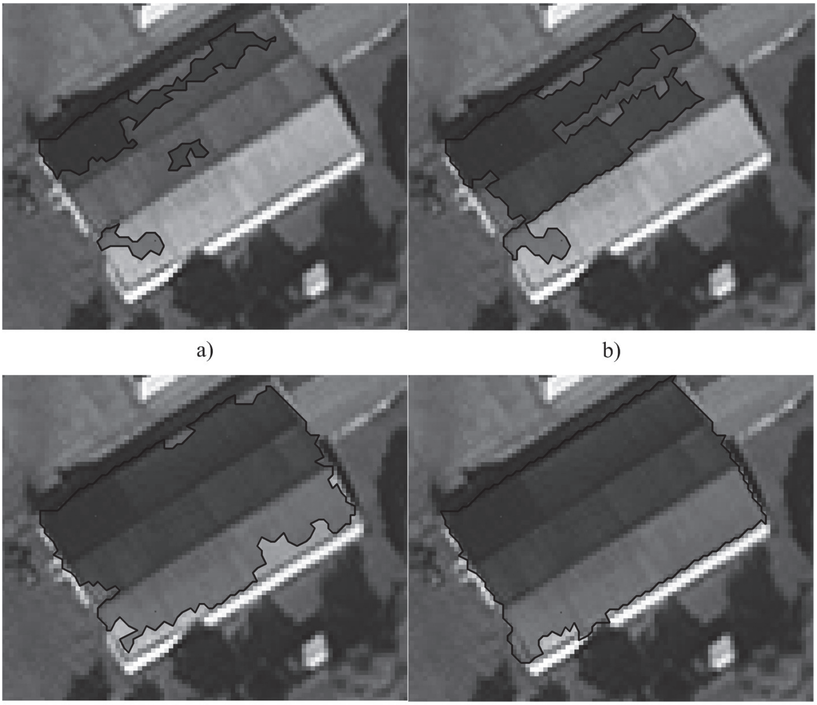
6. The reference points' setup process must be straightforward and transparent for an operator. It is recommended to set a point for every homogeneous region of an object starting from a larger one or a central one (in case regions sizes are close). After a starting polygon has appeared operator chooses one of the following options:

– increases threshold value in case the polygon doesn’t cover the whole object, and there is no clear border between its covered and uncovered regions;

– places addition reference points in case there is a clear border between the polygon and uncovered areas (mostly one point per region);

– decreases threshold value in case the polygon covers some regions that do not belong to a digitizing object.

An operator performs the described actions until the polygon’s position and aspect ratio would match a digitizing object.


a)

b)

c)

d)

Fig. 7. An illustration of threshold value effect on a resulting cluster with $t = 4$ (a), $t = 7$ (b), $t = 10$ (c) и $t = 18$ (d)

For instance, the building with a span-roof contains two regions due to different sunlight reflection angles of its parts (the left object on Figure 8). To digitize that object an operator should set two reference points to each side of the roof.

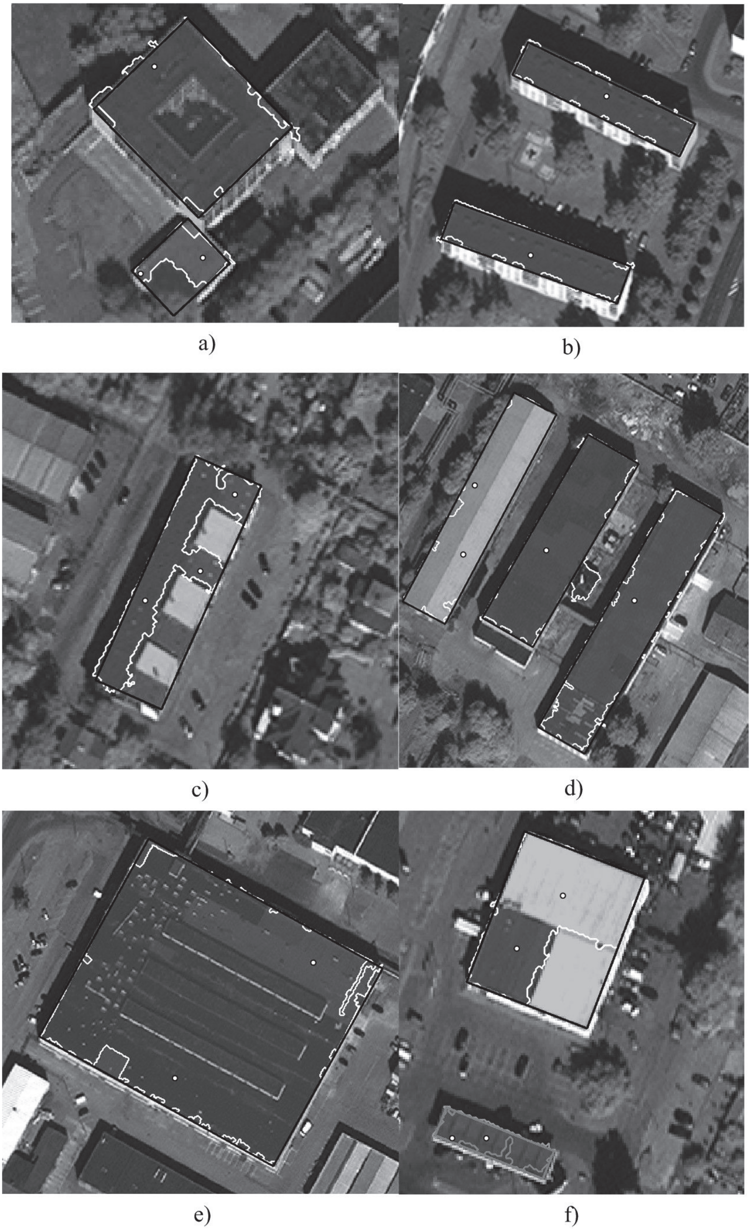


Fig. 8. A visualization of objects' digitizing with the developed method (output polygons' borders are black and the cluster borders are white)

Figure 8 displays samples of different objects' digitizing with the developed method. We set up to three reference points to detect them:

– one point per sample for buildings on Figure 8b, central and right on figure 8d and the larger one on Figure 8a;

– two points per sample were used for instances on Figure 8e, 8f, the left on Figure 8d, and the smaller one on Figure 8a;

– three points were required to digitize just the object on Figure 8c.

The developed method, on the one hand, digitizes objects that consist of several visually distinctive parts (Figure 8c, 8e, 8f) and on the other hand, correctly detects objects that are fractionally hidden or shaded by others with compensation of minor clustering errors.

4. Experiment and results. Within the experiment, we compared an output map precision and time consumption to create it using software based on the developed method and one of the most popular interactive digitizing toolsets of GIS MapInfo [28]. Dataset is represented by urban imagery with a spatial resolution of 0.5 m/pixel where mostly comprises multi-apartment residential development and industrial zones, and was used to prepare a ground truth digital map containing 777 rectangular buildings and structures (38% from all objects on the selected territory according to OpenStreetMap data [29]). Figure 10a shows a histogram of objects' sizes on this map.

We engaged two experts to carry out the experiment whose aim was to create a map similar to the ground truth using alternately the interactive toolset and the software based on the developed method. During the process experts did not have to spend time finding objects to digitize themselves – they used the specific markup layer made from the ground truth map that indicates the desired buildings by crosses as shown on Figure 9.

4.1. Precision criterions. Quantitative measure of precision for digital maps, produced by experts, is represented by the following criterions:

1. Root-mean-square error (RMSE) of object's points positioning which is expressed in pixels and equals to a Euclidian distance to the nearest point of the corresponding ground truth object.

2. Mean rotation error (MRE) expressed in degrees is measured as a minimum angle between straight lines forming the current object and the corresponding ground truth one.

3. Jaccard index (is dimensionless, expressed in percentages) represents the ratio of intersection area of a created object and a corresponding ground truth one to their union's area.

RMSE describes an absolute object positioning error comprising four types of distortions: aspect ratio mismatch, spatial shift, scaling and rotation errors. The last distortion is quantitatively expressed by MRE which was separated from RMSE to estimate a substitution precision of gradient angle mode (16) instead of the rotation angle.

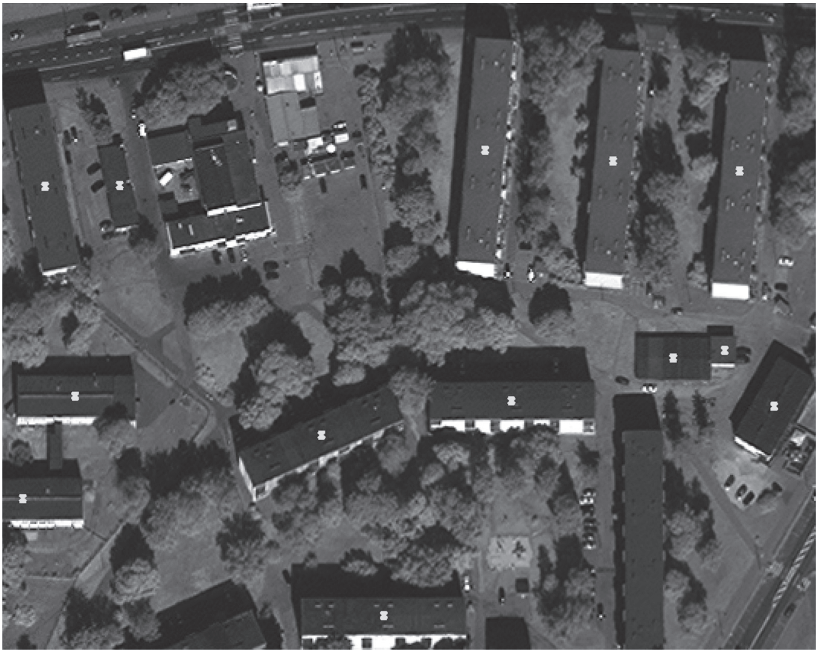


Fig. 9. A specific markup layer that indicates the desired objects by crosses

Jaccard index represents a complete relative similarity measure of a created polygon to a ground truth one.

The algorithm of the created digital map precision estimation contains the following steps:

1. Find a corresponding object on the ground truth map for each one on the created map by a criterion of maximum intersection area (has to be nonzero).

2. If the ground truth object was found — calculate values of RMSE, MRE, and Jaccard index and write them to the created object's semantic fields to have the ability to analyze the results visually.

3. Calculate mean values of precision criterions for the created map in a whole.

4.2. Experiment results. In total experts created four digital maps within the experiment (two — with the interactive toolset, and two — with the developed method). Table 1 demonstrates mean RMSE, MRE, and Jaccard index values together with digitizing workflow duration. Duration of the digitizing workflow decreased by 39.9% and 26.3% for the first and the second expert respectively. Error values of the digital map created by the second expert using the developed method although are slightly greater comparing to the interactive one, but still are within the range of acceptable values. The first expert gained a more significant time economy but exceeded maximum RMSE and MRE values that for our experiment are set to 2 pixels (mean error of object's border visual detection in an original image scale without zooming) and 1 degree (determined by the a quantization of histogram used to calculate a gradient angle mode). Both experts used two reference points per object on the average.

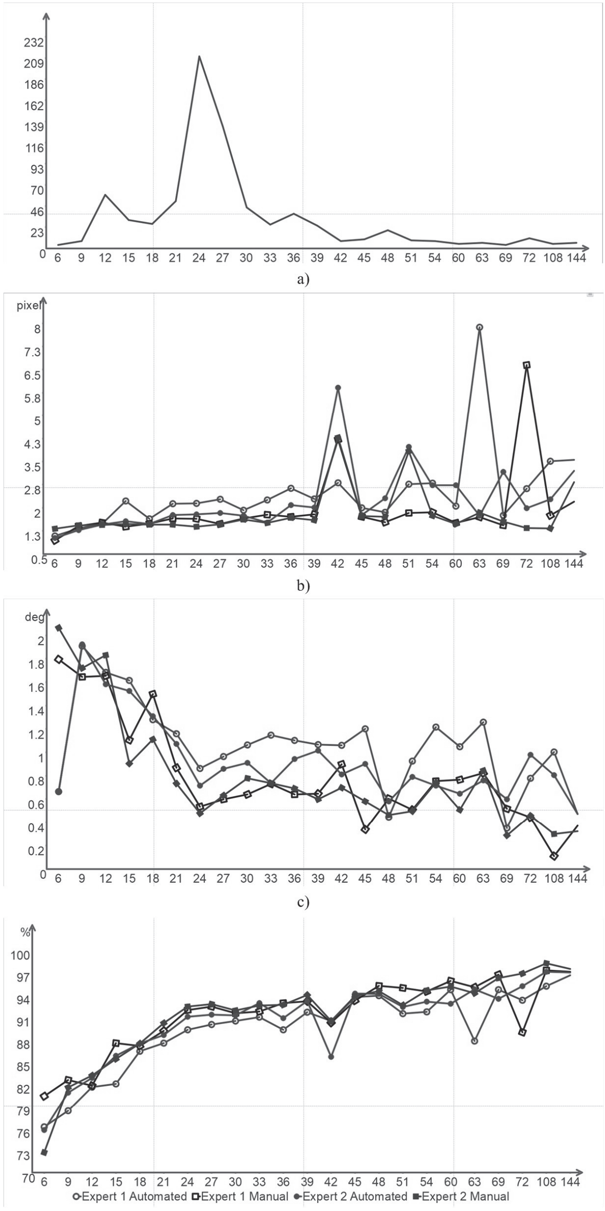
Table 1. Mean RMSE, MRE and Jaccard index values together with digitizing workflow duration for created digital maps

Expert	Method	Objects quantity	Digitizing workflow duration, min	Time per object, sec	RMSE, pixels	MRE, deg	Jaccard index, %
Expert 1	Interactive	775	180	13.94	1.779	0.751	90.93
	Developed	773	*108*	*8.38*	2.251	1.053	88.89
	Difference		–72 (–40%)	–5.56 (–39.9%)	+0.472 (+21%)	+0.302 (+28.7%)	–2.04
Expert 2	Interactive	773	188	14.59	1.567	0.728	91.33
	Developed	770	*138*	*10.75*	1.909	0.925	90.33
	Difference		–50 (–26.8%)	–3.84 (–26.3%)	+0.342 (+17.9%)	+0.197 (+21.3%)	–1

Within the experiment, we also investigated dependencies of RMSE (Figure 10b), MRE (Figure 10c) and Jaccard index (Figure 10d) from a polygon's size to reveal features of the developed method.

RMSE has a direct dependency from a polygon's size. For objects smaller then 15 pixels RMSE value is 25% below the mean both for the interactive method and for the developed one. Similarly, RMSE value is 25% above the mean for objects with sizes higher than 39 pixels because operators subjectively pay less attention to a precision of a more large object. A higher spread of RMSE values for these objects is explained by their less quantity (not enough to get a smooth graph).

A discreetness of the source image explains an MRE reverse dependency from a polygon's size. The positioning error of 1 pixel for an object with the size of 10 pixels causes a rotation error of 6.3 degrees. In contrast, the same positioning error for an object with the size of 50 pixels causes a rotation error of 1.3 degrees. Higher MRE values of the developed method for objects with sizes above 18 pixels appear due to quantization of histogram used to calculate a gradient angle mode (16).



a)

b)

c)

d)

Fig. 10. Histogram of objects polygons’ sizes on the experimental dataset (a), dependencies graphs of RMSE (b), MRE (c) and Jaccard index (d) from a polygon’s size

Mean Jaccard index value directly depends on polygon’s size similar to MRSE that is also caused by a source image’s discreetness. A higher error of the interactive method is specified by a capability to reach a sub-pixel precision by digitizing objects on a source image zoomed above the original scale (an operator magnifies the image in 2-5 times and manually sets points of an output polygon between vertexes of image’s discrete grid using his skills and experience).

5. Conclusion and discussion. We developed a new method for remote sensing data processing designed to digitize rectangular objects using approximate spatial positions providing automatic orientation and aspect ratio estimation. The experimental investigation of the method has shown workflow acceleration by 25–40% facing a slight precision reduction. In a shortcomings list, we should mention 20% higher MRE value comparing to the interactive method and an impossibility to reach a sub-pixel precision of a resulting polygon’s points positioning.

Within a further work, we will improve our method to overcome the mentioned disadvantages, apply Hough transform for round and orthogonal polygons and further to develop a generalized method for digitizing objects of any analytically described shape, investigate a neural networks based methods and an interface design of automated digitizing tools to enhance workflow performance and comfort.

References

1. Schowengerdt R.A. Remote sensing: models and methods for image processing. Elsevier. 2007. 560 p. (Russ. ed.: Schowengerdt R.A. *Distancionnoe zondirovanie. Modeli i metody obrabotki izobrazhenij.* Moscow: Tehnosfera. 2013. 592 p.).
2. Antonushkina S.V. et al. *Sovremennye tehnologii obrabotki dannyh distancionnogo zondirovanija Zemli* [Modern technologies for processing of Earth remote sensing data]. Moscow: Physmathlit. 2015. 460 p. (In Russ.).
3. Blundell J.S., Opitz D.W. Object recognition and feature extraction from imagery: the Feature Analyst® approach. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. 2006. vol. 36. no. 4. pp. 42–47.
4. ENVI EX User’s Guide. Available at: http://www.harrisgeospatial.com/portals/0/pdfs /enviex/ENVI_EX_User_Guide.pdf (accessed: 15.02.2018).
5. Manual for satellite data analysis eCognition developer. Available at: http://open_jicareport.jica.go.jp/pdf/12150314_03.pdf (accessed: 15.02.2018).
6. Kompleks avtomatizirovannogo deshifrirovanija i vektorizacii dannyx DZZ [Complex for automated recognition and digitizing of remote sensing data]. Available at: https://gisinfo.ru/products/automap.htm (accessed: 15.02.2018). (In Russ.).
7. Jebur M.N. et al. Per-pixel and object-oriented classification methods for mapping urban land cover extraction using SPOT 5 imagery. *Geocarto International.* 2014. vol. 29. no. 7. pp. 792–806.
8. Larose D.T. Discovering knowledge in data: an introduction to data mining. John Wiley & Son. 2014. 336 p.
9. Iglovikov V., Mushinskiy S., Osin V. Satellite imagery feature detection using deep convolutional neural network: A Kaggle competition. 2017. arXiv:1706.06169. Available at: https://arxiv.org/abs/1706.06169 (accessed: 15.02.2018).

10. How we participated in SpaceNet three Road Detector challenge. And how we got into top 10. Available at: https://spark-in.me/post/spacenet-three-challenge (accessed:15.02.2018).
11. Safe Passage: Detecting and Classifying Vehicles in Aerial Imagery. Available at: https://www.datasciencechallenge.org/challenges/1/safe-passage (accessed: 15.02.2018).
12. Iglovikov V. Britanskie sputnikovye snimki 2: kak vse bylo na samom dele [British Satellite Images 2: how it was really]. Available at: https://habrahabr.ru/company/ods/blog/330118/ (accessed: 15.02.2018). (In Russ.).
13. Dstl Satellite Imagery Feature Detection. Available at: https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection (accessed: 15.02.2018).
14. Kuzin A. Vtoroe pochetnoe. Zametki uchastnika konkursa Dstl Satellite Imagery Feature Detection [The second honorable place. Notes of the Satellite Imagery Feature Detection Challenge Participant]. Available at: https://habrahabr.ru/company/avito/blog/325632/ (accessed: 15.02.2018). (In Russ.).
15. Lee K. Dstl Satellite Imagery Competition, 1st Place Winner's Interview. Available at: http://blog.kaggle.com/2017/04/26/dstl-satellite-imagery-competition-1st-place-winners-interview-kyle-lee/ (accessed: 15.02.2018).
16. Otkrytyj konkurs na luchshee reshenie v oblasti sozdanija intellektual'nyh tehnologij deshifrirovanija vidovoj ajerokosmicheskoj informacii [The open competition for a best solution in a field of intellectual technologies creation for recognition of aerospace imagery in visible spectrum]. Available at: http://fpi.gov.ru/activities/konkurs/spacemap (accessed: 15.02.2018). (In Russ.).
17. NOAA Fisheries Steller Sea Lion Population Count Available at: https://www.kaggle.com/c/noaa-fisheries-steller-sea-lion-population-count (accessed: 15.02.2018).
18. Kaggle: kak nashi setochki schitali morskih l'vov na Aleutskih ostrovah [Kaggle: how our networks counted the sea lions in the Aleutian Islands]. Available at: https://habrahabr.ru/company/ods/blog/337548/ (accessed: 15.02.2018). (In Russ.).
19. Obusek F. Delineating rooftops in ERDAS IMAGINE. Available at: https://www.youtube.com/watch?v=HBcPB6Agr5c (accessed: 15.02.2018).
20. Blundell S. et al. Feature analyst v5.0. Proceedings of the ASPRS Annual Conference. 2008. pp. 28–37.
21. Ballard D.H. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition.* 1981. vol. 13. no. 2. pp. 111–122.
22. Gonzalez R., Woods R. Digital image processing. Pearson Education. 2002. 793 p. (Russ. ed.: Gonzalez R., Woods R. *Cifrovaja obrabotka izobrazhenij*. Moscow. Tehnosfera. 2006. 1072 p.).
23. Duda R.O., Hart P.E., Stork, D.G. Pattern classification. John Wiley & Sons. 2000. 680 p.
24. van Vliet L.J., Verbeek P.W. Edge localization by MoG filters: Multiple-of-Gaussians. *Pattern Recognition Letters*. 1994. vol. 15. no. 5. pp. 485–496.
25. Burt P., Adelson E. The Laplacian pyramid as a compact image code. *IEEE Transactions on communications*. 1983. vol. 31. no. 4. pp. 532–540.
26. Emelyanov S.G. et al. *Metody i sistemy cifrovoj obrabotki ajerokosmicheskih izobrazhenij* [Methods and systems for digital processing of aerospace imagery]. Novosibirsk. Nauka. 2012. 175 p. (In Russ.).
27. Vatutin E.I., Miroshnichenko S.Yu. Titov V.S. [Software optimization of Sobel operator using SIMD-extensions of the x86 family processors]. *Telecommunikatsii — Telecommunication*. 2006. Issue 6. vol. 12. pp. 12–16. (In Russ.).
28. MapInfo Pro – Desktop GIS. Available at: https://www.pitneybowes.com/us/location-intelligence/geographic-information-systems/mapinfo-pro.html (accessed: 15.02.2018).
29. NextGIS Data. Geodata in ESRI Shape, ESRI Geodatabase, GeoJSON, Mapinfo Tab, PDF formats. Available at: https://data.nextgis.com/en/ (accessed: 15.02.2018).

Miroshnichenko Sergey Yurievich — Ph.D., associate professor of computer sciences department, Southwest State University (SWSU). Research interests: Earth remote sensing data processing, automated recognition and digitizing of geospatial objects, machine learning. The number of publications — 90. oldguy7@rambler.ru; 94, 50 Let Oktyabrya str., Kursk, 305040, Russia; office phone: +79081277657, Fax: +7(4712)222-665.

Titov Vitalii Semenovich — Ph.D., Dr. Sci., professor, head of computer sciences department, Southwest State University (SWSU). Research interests: image analysis, pattern recognition, robotic systems, autonomous robots, machine vision systems, calibration. The number of publications — 630. titov-kstu@rambler.ru; 94, 50 Let Oktyabrya str., Kursk, 305040, Russia; office phone: +7(4712)222-665, Fax: +7(4712)222-665.

Dremov Evgenii Nikolaevich — Ph.D. student of computer sciences department, Southwest State University (SWSU). Research interests: raster topographic maps automated processing and recognition, images preprocessing. The number of publications — 16. evgeni-dremov@yandex.ru; 94, 50 Let Oktyabrya str., Kursk, 305040, Russia; office phone: +7(4712)222-665, Fax: +7(4712)222-665.

Mosin Sergey Aleksanrovich — Ph.D. student of computer sciences department, Southwest State University (SWSU). Research interests: automated remote sensing data recognition, artificial neural networks. The number of publications — 8. 22mosin@gmail.com; 94, 50 Let Oktyabrya str., Kursk, 305040, Russia; office phone: +7(4712)222-665, Fax: +7(4712)222-665.

Acknowledgements. This research is supported by the state assignment (project №2.9102.2017/BCh).

УДК 004.92 **DOI** 10.15622/sp.61.7

С.Ю. Мирошниченко, В.С. Титов, Е.Н. Дремов, С.А. Мосин

ВЕКТОРИЗАЦИЯ ПРЯМОУГОЛЬНЫХ ПРОСТРАНСТВЕННЫХ ОБЪЕКТОВ НА АЭРОКОСМИЧЕСКИХ ИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ ПРЕОБРАЗОВАНИЯ ХАФА

Мирошниченко С.Ю., Титов В.С., Дремов Е.Н., Мосин С.А. **Векторизация прямоугольных пространственных объектов на аэрокосмических изображениях с использованием преобразования Хафа.**

Аннотация. Рассматривается метод обработки данных дистанционного зондирования, предназначенный для векторизации прямоугольных объектов по их ориентировочным положениям с автоматическим определением ориентации и соотношения сторон.

Приведена формальная постановка задачи векторизации объекта заданной геометрической формы на изображении с использованием априорной информации о его пространственном положении. Метод позволяет создавать векторные представления прямоугольных геопространственных объектов по одной или нескольким устанавливаемым оператором опорным точкам и основан на кластеризации исходного изображения с использованием имеющихся спектральных каналов в качестве пространства признаков. Последующее преобразование Хафа использует локальное направление градиента яркости для оценки пространственной ориентации объекта и снижения вычислительной сложности преобразования, а также низкочастотную фильтрацию в процессе накопления значений для повышения робастности. Показана возможность модификации метода для обеспечения возможности векторизовать объекты любой аналитически задаваемой формы.

При разработке метода учтены требования по минимизации времени векторизации для повышения комфорта работы оператора и обеспечения возможности предсказуемого контроля размера создаваемого векторного представления.

Для экспериментального исследования разработанного метода использована тестовая выборка, содержащая более 700 объектов прямоугольной формы. В качестве критериев точности векторизации использованы средняя квадратическая ошибка позиционирования точек объекта, среднее угловое отклонение объекта в полярных координатах и коэффициент схожести площадных объектов Жаккара. Результаты эксперимента показывают снижение временных затрат на векторизацию на 25–40% при использовании программной реализации разработанного метода без существенного снижения точности создаваемой картографической продукции.

Ключевые слова: данные дистанционного зондирования, автоматизированная векторизация, обработка изображений, преобразование Хафа.

Мирошниченко Сергей Юрьевич — к-т техн. наук, доцент кафедры вычислительной техники, Юго-Западный государственный университет (ЮЗГУ). Область научных интересов: обработка данных дистанционного зондирования Земли, автоматизированное дешифри-рование и векторизация геопространственных объектов, машинное обучение. Число научных публикаций — 90. oldguy7@rambler.ru; ул. 50 Лет Октября, 94, Курск, 305040; р.т.: +79081277657, Факс: +7(4712)222-665.

Титов Виталий Семенович — д-р техн. наук, профессор, заведующий кафедрой вычислительной техники, Юго-Западный государственный университет (ЮЗГУ). Область научных интересов: обработка изображений, распознавание образов, робототехнические системы, автономные роботы, системы технического зрения, калибровка. Число научных публикаций — 630. titov-kstu@rambler.ru; ул. 50 Лет Октября, 94, Курск, 305040; р.т.: +7(4712)222-665, Факс: +7(4712)222-665.

Дремов Евгений Николаевич — аспирант кафедры вычислительной техники, Юго-Западный государственный университет (ЮЗГУ). Область научных интересов: обработка и распознавание растровых топографических карт, предварительная обработка изображений. Число научных публикаций — 16. evgeni-dremov@yandex.ru; ул. 50 Лет Октября, 94, Курск, 305040; р.т.: +7(4712)222-665, Факс: +7(4712)222-665.

Мосин Сергей Александрович — аспирант кафедры вычислительной техники, Юго-Западный государственный университет (ЮЗГУ). Область научных интересов: автоматизированное дешифрирование данных дистанционного зондирования Земли, искусственные нейронные сети. Число научных публикаций — 8. 22mosin@gmail.com; ул. 50 Лет Октября, 94, Курск, 305040; р.т.: +7(4712)222-665, Факс: +7(4712)222-665.

Поддержка исследований. Работа выполнена при финансовой поддержке государственного задания (проект №2.9102.2017/БЧ).

Литература

1. *Шовенгердт Р.А.* Дистанционное зондирование. Модели и методы обработки изображений // М.: Техносфера. 2013. 592 с.
2. *Антонушкина С.В. и др.* Современные технологии обработки данных дистанционного зондирования Земли // М.: Физматлит. 2015. 460 с.
3. *Blundell J.S., Opitz D.W.* Object recognition and feature extraction from imagery: the Feature Analyst® approach // International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. 2006. vol. 36. no. 4. pp. 42–47.
4. ENVI EX User’s Guide. URL: http://www.harrisgeospatial.com/portals/0/pdfs /enviex/ENVI_EX_User_Guide.pdf (дата обращения: 15.02.2018).
5. Manual for satellite data analysis eCognition developer. URL: http://open_jicareport.jica.go.jp/pdf/12150314_03.pdf (дата обращения: 15.02.2018).
6. Комплекс автоматизированного дешифрирования и векторизации по данным ДЗЗ. URL: https://gisinfo.ru/products/automap.htm (дата обращения: 15.02.2018).
7. *Jebur M.N. et al.* Per-pixel and object-oriented classification methods for mapping urban land cover extraction using SPOT 5 imagery // Geocarto International. 2014. vol. 29. no. 7. pp. 792–806.
8. *Larose D.T.* Discovering knowledge in data: an introduction to data mining // John Wiley & Sons. 2014. 336 p.
9. *Iglovikov V., Mushinskiy S., Osin V.* Satellite imagery feature detection using deep convolutional neural network: A Kaggle competition. 2017. arXiv:1706.06169. URL: https://arxiv.org/abs/1706.06169 (дата обращения: 15.02.2018).
10. How we participated in SpaceNet three Road Detector challenge. And how we got into top 10. URL: https://spark-in.me/post/spacenet-three-challenge/ (дата обращения: 15.02.2018).
11. Safe Passage: Detecting and Classifying Vehicles in Aerial Imagery. URL: https://www.datasciencechallenge.org/challenges/1/safe-passage (дата обращения: 15.02.2018).
12. *Игловиков В.* Британские спутниковые снимки 2: как все было на самом деле. URL: https://habrahabr.ru/company/ods/blog/330118/ (дата обращения: 15.02.2018).
13. Dstl Satellite Imagery Feature Detection. URL: https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection (дата обращения: 15.02.2018).

14. *Кузин А.* Второе почетное. Заметки участника конкурса Dstl Satellite Imagery Feature Detection. URL: https://habrahabr.ru/company/avito/blog/325632/ (дата обращения: 15.02.2018).
15. *Lee K.* Dstl Satellite Imagery Competition, 1st Place Winner's Interview. URL: http://blog.kaggle.com/2017/04/26/dstl-satellite-imagery-competition-1st-place-winners-interview-kyle-lee/ (дата обращения: 15.02.2018).
16. Открытый конкурс на лучшее решение в области создания интеллектуальных технологий дешифрирования видовой аэрокосмической информации. URL: http://fpi.gov.ru/activities/konkurs/spacemap (дата обращения: 15.02.2018).
17. NOAA Fisheries Steller Sea Lion Population Count URL: https://www.kaggle.com/c/noaa-fisheries-steller-sea-lion-population-count (дата обращения: 15.02.2018).
18. Kaggle: как наши сеточки считали морских львов на Алеутских островах. URL: https://habrahabr.ru/company/ods/blog/337548/ (дата обращения: 15.02.2018).
19. *Obusek F.* Delineating rooftops in ERDAS IMAGINE. URL: https://www.youtube.com/watch?v=HBcPB6Agr5c (дата обращения: 15.02.2018).
20. *Blundell S. et al.* Feature analyst v5.0 // Proceedings of the ASPRS Annual Conference. 2008. pp. 28–37.
21. *Ballard D.H.* Generalizing the Hough transform to detect arbitrary shapes // Pattern recognition. 1981. vol. 13. no. 2. pp. 111–122.
22. *Гонсалес Р., Вудс Р.* Цифровая обработка изображений // М.: Техносфера. 2005. 1072 с.
23. *Duda R.O., Hart P.E., Stork D.G.* Pattern classification // John Wiley & Sons. 2000. 680 p.
24. *van Vliet L.J., Verbeek P.W.* Edge localization by MoG filters: Multiple-of-Gaussians // Pattern Recognition Letters. 1994. vol. 15. no. 5. pp. 485–496.
25. *Burt P., Adelson E.* The Laplacian pyramid as a compact image code // IEEE Transactions on communications. 1983. vol. 31. no. 4. pp. 532–540.
26. *Емельянов С.Г. и др.* Методы и системы цифровой обработки аэрокосмических изображений // Новосибирск. Наука. 2012. 175 с.
27. *Ватутин Э.И., Мирошниченко С.Ю., Титов В.С.* Программная оптимизация оператора Собела с использованием SIMD-расширений процессоров семейства x86 // Телекоммуникации. 2006. Т. 6. №. 12. С. 12–16.
28. MapInfo Pro – Desktop GIS. URL: https://www.pitneybowes.com/us/location-intelligence/geographic-information-systems/mapinfo-pro.html (дата обращения: 15.02.2018).
29. NextGIS Data. Геоданные в форматах ESRI Shape, ESRI Geodatabase, GeoJSON, Mapinfo Tab, PDF. URL: https://data.nextgis.com/ru/ (дата обращения: 15.02.2018).