Localization and Normalization of Isolated Direct Contour Lines Based On Masks and Form-Factor

[1]Oday J. Al-Furaiji [2]Aksana G. Shauchuk, [3]Viktar Yu. Tsviatkou,
1 Head of Computer Technologies Engineering, Kinouze University College
2. Department of networks and devices BSUIR, Minsk, Belarus
3.Department of networks and devices BSUIR, Minsk, Belarus

Abstract
The method for localization of isolated direct contour lines based on form- factor is proposed in this paper. The essence of this method consists in comparison to unit of value of form-factor – the relations of the size of the contour line (distance between end points) to its length (number of the forming contour pixels). It is found that the proposed method provides a gain in speed of allocating direct contour lines in 4 times in comparison with the method of LSD (Line Segment Detector) and in 14.8 times in comparison with the method based on the Hough transform.
The effective method of normalization of the contour line on thickness on the basis of the analysis of orientation of the connected pixels is proposed in this paper. This method is specially developed for deleting excess pixels of a dedicated line.
Also developed algorithm of normalization of contour lines on thickness promotes the best calculation of a form-factor.

Key words: allocation of direct lines, contour image processing, key elements, identification, parameterization.

الخلاصة: تحديد وتطبيع الخطوط الكنتورية المستقيمة المنعزلة على اساس الاقنعة وعامل الشكل في هذا البحث تم اقتراح طريقة لتحديد الخطوط الكنتورية المستقيمة المنعزلة على اساس عامل الشكل. اساس هذه الطريقة يكمن بالمقارنة بوحدة قيمة عامل الشكل علاقة حجم الخط الكنتوري (البعد بين نقاط النهاية) والنسبة الى طولها (عدد البكسلات الكنتورية المكوّنة). وقد وجد بان الطريقة المقترحة توفر ربح في سرعة تحديد الخطوط الكنتورية المستقيمة بـ(٤ مرات) مقارنة بطريقة مكتشف المقطع الخطي LSD (Line Segment Detector) و بـ(١٤,٨ مرة) مقارنة بطريقة تحويل هوف Hough transform.

Introduction

Methods of processing of the overlapped images (imposing, comparison, creation of panoramas, etc.) are based on localization the key elements. The most effective methods of localization of key elements, such as SIFT [1] and SURF [2], use keypoints. It allows to reduce computing complexity of comparison of images in comparison with correlation methods [3], however results in low resistance of results of localization of reference points to change of brightness, contrast and noise of the image. Direct lines possess considerably bigger stability in comparison with points. For their allocation on images the methods based on transformation of the Half [4], mask search [5], calculation of a gradient [6] and quantization on orientation [7] are widely used. However these methods have high computing complexity. Elimination of this shortcoming possibly at the expense of the accounting of a ratio of lengths of contour lines and distances between their end points, that is at the accounting of a form factor of the contour line.

In case of parametrization and identification of lines with use of a form factor thin boundaries yield considerably the best result. Therefore after application of operators of separation of boundaries (Canny, Roberts and others) it is necessary to subject dedicated lines to additional processing – a thinning of line. Algorithm of thinning is resulted by deleting all excess contour pixels and contour lines one pixel thick is created.

Widely used algorithm of thinning, such as algorithm Zong-Suen [8], the algorithm of thinning by sample, a wave method, Shchepin algorithm, provide the maximum thinning of contour lines [9]. However these approaches aren't effective for a algorithm of thinning of the contour lines having thickness some pixels as are oriented on processing of areal objects of any size and therefore use iterative algorithms of deleting excess contour pixels before achievement of the required line thickness that leads to growth of time of contour processing.

For abbreviation of time of processing of the contour lines having thickness some pixels it is possible to use algorithm of normalization of the line on thickness which is based on processing of the form of lines and not iterative algorithms.

Method of localization of lines on the basis of a form factor and of a fast algorithm of normalization of contour lines on thickness is proposed in this paper.

1. Method of localization of lines based on form-factor

The method for localization of isolated contour lines based on the analysis of value of a form factor – the ratio of the size of the contour line calculated on known coordinates of her ending points to length of the contour line determined by the amount of the contour points which form it - is offered. For direct contour lines this relation shall be close to 1. By this criterion the method allows to select only isolated direct lines which don't have intersections with other lines. The method provides reduction of computing complexity of localization of contour lines due to of rather small number of processed contour pixels.

The flowchart of a method for localization of isolated lines based on form-factor is provided in figure 1.

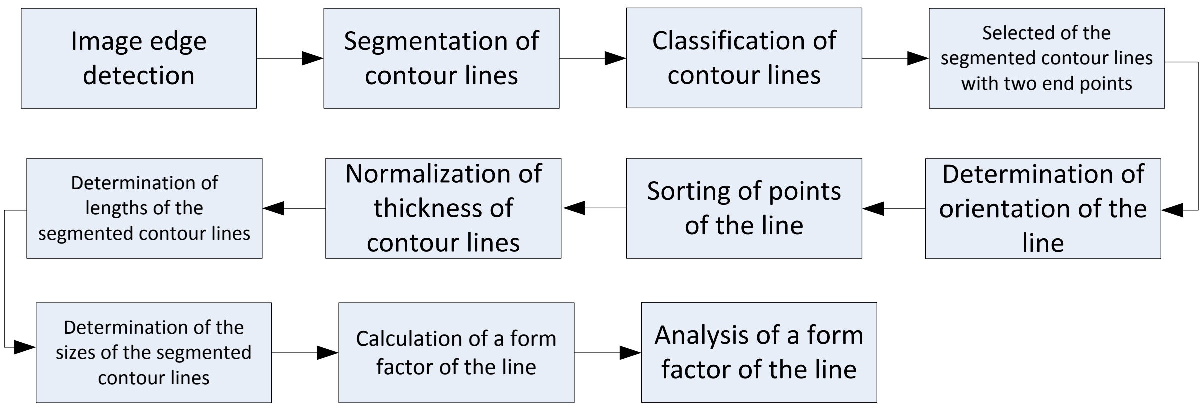


Figure 1 – flowchart of a method for localization of isolated lines based on form-factor.

The algorithm of localization of isolated direct contour lines based on form-factor consists of the following steps.

1) Image edge detection. As a result of contour edge detection of the source image $I = \left\| i(y,x) \right\|_{\left(y=\overline{0,Y-1},x=\overline{0,X-1}\right)}$ the binary contour image $I_B = \left\| i_B(y,x) \right\|_{\left(y=\overline{0,Y-1},x=\overline{0,X-1}\right)}$ which single pixels define contour points where Y, X – image sizes on a vertical and a horizontal is created.
2) Segmentation of contour lines. To each contour pixel $i_B(y,x)$ is assigned the number of the line which it belongs. The segmentation matrix $I_S = \left\| i_S(y,x) \right\|_{\left(y=\overline{0,Y-1},x=\overline{0,X-1}\right)}$ is created. Values of array elements are concluded in an interval $[1, N_s]$, where N_s – number of contour lines.
3) Classification of contour lines. Classification of contour lines is carried out on number of end points. For each segmented the contour line $l(n)$ formed by pixels $i_B(y,x)=1$, for which the appropriate pixels $i_S(y,x)=n$ where $n \in (1, N_S)$, the number of end points is defined. An end point is the single pixel $i_B(y,x)$ of a binary contour image which has only one adjacent single pixel $i_B(y+p, x+q)$ where $p = \overline{-1,1}$, $q = \overline{-1,1}$ and $p+q \neq 0$.
4) Selected of the segmented contour lines with two end points. From the set of the segmented contour lines $L_S = \left\{ l_S(n) \right\}_{\left(n=\overline{1,N_S}\right)}$ including the shorted, crossed and open circuit isolated lines the subset of contour lines $L_2 = \left\{ l_2(n) \right\}_{\left(n=\overline{1,N_2}\right)}$ which have two end points where N_2 – number of the segmented contour lines with two end points is selected. As a rule, the crossed contour lines have more than two end points. Therefore on this step of algorithm contour lines are selected generally isolated (not having intersections). An exception is intersection of closed and open-ended contour lines. Such exception is processed on the following steps of algorithm.
5) Determination of orientation of the line. For the selected contour line on the basis of the found end points orientation is calculated: $O = \begin{cases} \dfrac{x_1 - x_2}{y_1 - y_2}, & y_1 \neq y_2 \\ 60, & y_1 = y_2 \end{cases}$, where x_1, y_1, x_2, y_2 – coordinates of end points of the line.
6) Sorting of points of the line. The stack of pixels of the selected contour line is sorted by the bubble sort method on x if the line is drawn out on a x axis, or on y if it is drawn out on a y axis. The stack is sorted by increase, if orientation of the line the negative, and by decrease, if orientation – the positive.
7) Normalization of thickness of contour lines. For each contour line excess pixels which are deleted are defined. Excess pixels decide on the help of masks of

2×2 pixel and orientation of the line. As a result of geometrical normalization all a contour lines have thickness in one pixel.

8) Determination of lengths of the segmented contour lines. For each contour line having two end points $l_2(n)$ the number of the contour pixels $s_2(n)$ forming this line $i_B(y,x)$ by means of expression is defined

$$s_2(n)=\sum_{y=0}^{Y-1}\sum_{x=0}^{X-1}t(m)i_B(y,x)k(n,y,x), \quad (1)$$

where $k(n,y,x)=\begin{cases}0 \ ,i_S(y,x)\neq n,\\ 1 \ ,i_S(y,x)=n;\end{cases}$, $t(m)$ - the coefficient depending on a mask to which belongs pixel (figure 2), $t(m)=1$ in case of pixel belonging to a direct mask, $t(m)=1{,}2$ – to a broken mask and $t(m)=1{,}5$ – to an oblique mask.

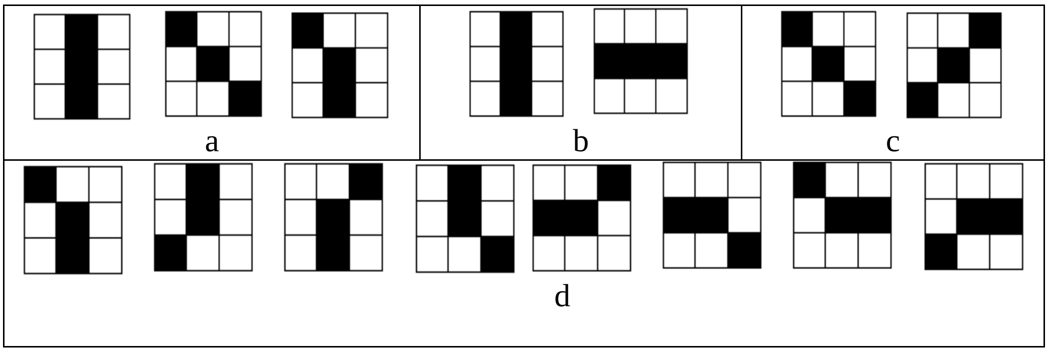


Figure2 - Classification of generatrixs:
a – types of generators (direct, oblique, broken);
b – kind of a direct line; c – kind of oblique;
d – kind of a broken line.

The number $s(n)$ is interpreted as length of the contour line.

9) Determination of the sizes of the segmented contour lines. For each contour line $l_2(n)$ having two end points the distance between end points $r_2(n)$ (the size of the contour line) by means of expression is calculated

$$r_2(n)=\sqrt{(y_1(n)-y_2(n))^2+(x_1(n)-x_2(n))^2} \quad (2)$$

where $(y_1(n),x_1(n)),(y_2(n),x_2(n))$ – coordinates of end points of the contour line $l_2(n)$.

10) Calculation of a form factor of the line. For each contour line $l_2(n)$ having two end points value of a form factor $f_2(n)$ of the line by means of expression is calculated

$$f_2(n) = r_2(n)/s_2(n) \quad (3)$$

11) Analysis of a form factor of the line. For each contour line $l_2(n)$ having two end points comparing of value of its form factor $f_2(n)$ with 1 is carried out. If value of a form factor $f_2(n)$ closely to 1, the contour line $l_2(n)$ is a direct line. Otherwise $f_2(n)<1$ the line is a curve or a broken line. Closeness of value of a form factor $f_2(n)$ of the line to 1 is estimated as a result of comparison of the given threshold T_c defining curvature of the line to the difference module calculated by means of expression $|1 - f_2(n)|$. The contour line $l_2(n)$ is a direct line if $|1 - f_2(n)| \le T_C$. As a result of execution of this step of algorithm from a subset L_2 the subset $L_D = \{l_D(n)\}_{(n=\overline{1,N_D})}$ of isolated direct contour lines is selected, where N_D – number of isolated direct contour lines.

Are fixed information on the found isolated direct contour lines (coordinates of end points $(y_1(n), x_1(n)), (y_2(n), x_2(n))$, length $s_2(n)$, value of a form factor $f_2(n)$).

2. Segmentation of contour lines.

Segmentation usually is understood as process of search of uniform areas on the image. This stage very difficult and in a general view not algoritmization up to the end for arbitrary images. The segmentation methods based on determination of the uniform brightness (colors) or homogeneities like textures are most widespread.

In case of existence of stable distinctions in brightness of separate areas of a field of vision threshold methods are applied. Methods of building of areas are effective with steady connectivity in separate segments. It is good to apply a method of edge detection, if edge accurate and stable. The listed methods serve for separation of segments by criterion of uniform brightness. We will note that one of the most effective methods of building of areas assumes a choice of start points or by means of the operator (algorithm of centroid binding), or automatically. Here the method of watersheds based on search of local minima with the subsequent group of areas round them on connectivity is represented effective.

All methods are very acceptable from the point of view of computing expenses, however, for each of them ambiguity of a marking of points in real situations because of need of application эвристик is characteristic (a choice of thresholds of coincidence of brightness, a choice of digital masks, etc.). The algorithms of an acceleration of process of a marking allowing parallel processing on the basis of the logical analysis of adjacent elements have important practical value.

For the description and segmentation of properties of images, exactly, of homogeneity, roughness's, regulars, apply the textural methods sharing conditionally on two categories: statistical and structural. An example of

statistical approach is use of matrixes of the coincidence created of source images to the subsequent count of the statistical moments and entropies. In case of structural approach, for example, on the basis of a Voronoi mosaic, the set of polygons is built. Polygons with the general properties integrate in areas. For research of the general properties often use signs - the moments of polygons.
After segmentation there are noises in a look as separate changes of isolated image elements, and in the form of distortions of some downlink areas. Thus in case of segmentation by separation of boundaries use of the averaging filters masks is impossible as boundaries thus aren't emphasized, and are blurred. Special operators of integral type are applied to underlining of circuits.
The considered methods of segmentation can be reduced to the diagram in figure 3.

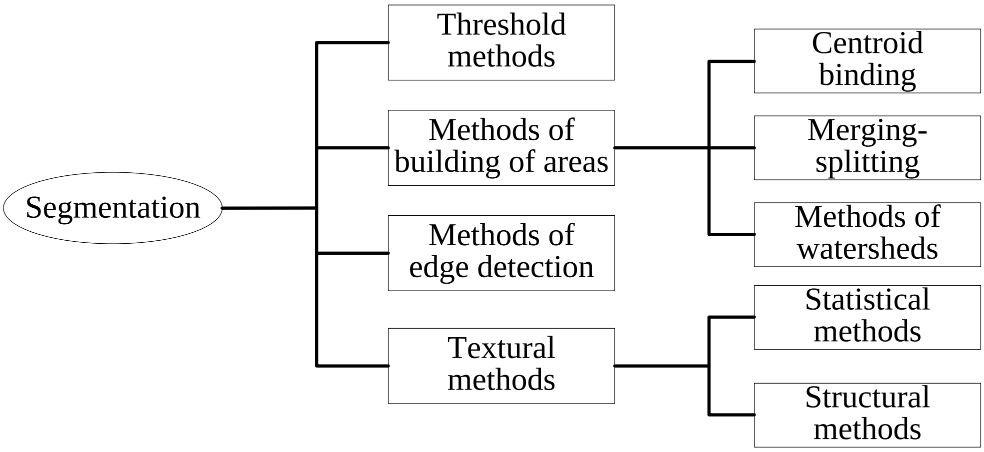


Figure 3. Classification of methods of segmentation.

Building of areas - one of the simplest approaches to segmentation for understanding on brightness: adjacent elements with an identical brightness are grouped together and form area.
Brice and Fennem [10] developed the method of building of areas based on a choice of simple rules of growth and a brightness mark. This provides rather exact segmentation of simple scenes with a small number of objects and without texture [10, 11].
In the developed method of localization of lines on the basis of a shape factor the modified Bruce-Fennem's method is used. This method is based not on a brightness sign, and on binary value of pixel, equal 1.

3. Normalization of contour lines on thickness

Among the existing thinning methods the most productive is the algorithm Zong-Suen, however it has the linear dependence from number of the pixels making boundaries [8].

In this algorithm the matrix 3*3 [12] is entered:

$$\begin{bmatrix} P9 & P2 & P3 \\ P8 & P1 & P4 \\ P7 & P6 & P5 \end{bmatrix}$$

This matrix is superimposed on the image, combining the interesting pixel with P1. Each iteration consists of two subiterations: one aimed at deleting the south-east boundary points and the north-west corner points while the other one is aimed at deleting the north-west boundary points and the south-east corner points. End points and pixel connectivity are preserved.

Iterations are executed until the quantity of the points deleted for iteration doesn't become equal to zero.

This algorithm when processing the contour lines having thickness of 2-3 pixels, a minimum two times processes all pixels from which only 10-20% will be deleted. I.e. the algorithm executes excess iterations that significantly influences processing rate.

The method of normalization of contour lines on thickness based on the shadow-mask analysis of orientation connected and deleting's excess contour pixels is offered. The method allows to increase the speed of contour processing at the expense of the one-fold analysis of each pixel in comparison with the known methods of a thinning using repeated processing of pixels. The entity of a method consists in classification of fragments of the contour line by means of masks, determination of excess contour pixels in these fragments and their deleting. The flowchart of algorithm is provided in figure 4.

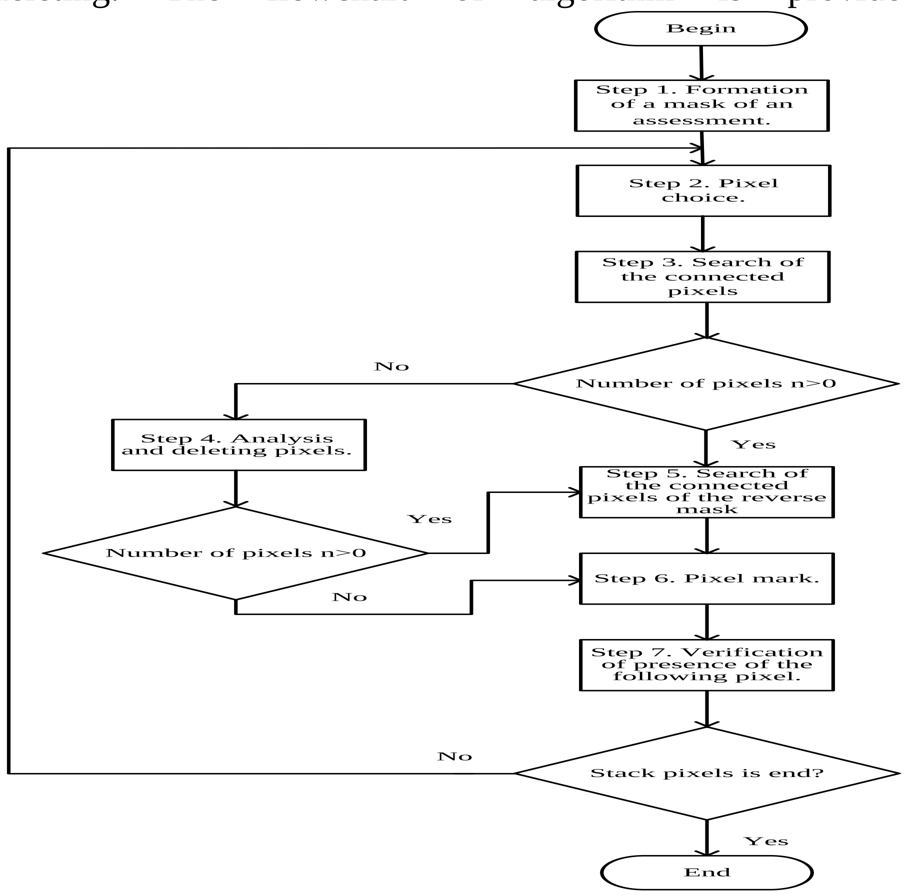
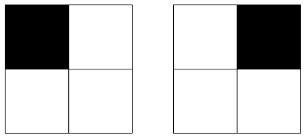


Figure 4. Flowchart of algorithm of normalization of the line on width.

Description of algorithm:

Step 1. Formation of a mask of an assessment. Masks for the analysis are created based on of the general orientation of the line of size 2x2 pixels (figure 5) relatively checked pixel. The checked pixel of the line is in the upper left corner, if orientation of the line $O<0$ (figure 5a), and in the upper right corner, if $O\geq 0$ (figure 5b). The provided masks are reversed each other.



a) b)

Figure 5. The Mask of detection of the connected pixels of the line:
a) for the line with orientation $O<0$; b) for the line with orientation $O\geq 0$.

Step 2. Pixel choice. From a stack of coordinates of the line not processed and not remote pixel (further reference pixel) is undertakes.

Step 3. Search of the connected pixels. All connected pixels with reference within 1 mask selected on a step are looked for (figure 10). If their quantity from 1 to 3, is carried out transition to a step 4 if pixels aren't found – that go on a step 5.

Step 4. Analysis and deleting pixels. All found pixels are checked for compliance of orientation of the line. For this purpose the deviation $r=\left|d(n)-O\right|$ is calculated, where $d(n)=\begin{cases}0, & y_1=y_2, x_1\neq x_2\\ 1, & y_1\neq y_2\ x_1\neq x_2\\ 60 & y_1\neq y_2\ x_1=x_2\end{cases}$ - orientation of the found pixel in a mask rather checked, O – orientation of the line. All pixels not the appropriate orientations, i.e. with $r>0.1$ are deleted if they aren't marked as checked. If the found pixels was 2 and all of them are deleted, for elimination of discontinuity of the line in a consequence of normalization there is an artificial determination of the connected pixel depending on orientation of the line. If $\left|O\right|$ is ranging from 0.5 to 2, the pixel with orientation is artificially decides on. If $\left|O\right|<0.5$ – the pixel with orientation $d(n)=0$ is selected. If $\left|O\right|>2$ – the pixel with orientation $d(n)=60$ is selected. Transition to a step 6 is carried out.

Step 5. Search of the connected pixels of the reverse mask. If within a mask of a pixel aren't found, all connected pixels within the reverse mask are looked for (figure 5). If pixels are found, transition to a step 4 if isn't present – on a step 6 is carried out.

Step 6. Pixel mark. The reference pixel is marked as processed.

Step 7. Verification of presence of the following pixel. If not processed and not remote pixel is present at a stack, transition to a step 3 if isn't present is carried out – that comes an output from algorithm.

Operation of algorithm is resulted by search and deleting excess pixels. Pixels visually and physically thickens contour are superfluous, pixels without which the contour will become thinner, but won't change the parameters of weight.

4. The experimental part.

For an assessment of efficiency of the offered method for localization of isolated contour lines based on form-factor comparing with the LSD method (Line Segment Detector) [6] and method based on Hough transform is carried out it [4]. As criteria of efficiency time and stability of results of localization of direct contour lines on the image are used.

In table 1, the time of localization of direct contour lines is given in the image size 600×600 pixels (figure 6a) for the offered method (FFLD – Form-Factor Line Detection), and also the LSD method and method based on Hough transform (HT). The computer on which experiments were made, has the following characteristics: number of cores – 2; processor frequency – 2.0 GHz; bus frequency – 250 MHz; cache memory size: 2.0 GB. For contour processing in the offered method and a method based on Hough transform Canni's filter is used. Follows from table 1 that the offered FFLD method provides a scoring in the speed of localization of direct contour lines by 4 times in comparison with the LSD method and by 14.8 times in comparison with method based on Hough transform.

Table 1 - Time of localization direct contour lines

	FFLD	LSD	HT
Time, c	0,1	0,4	1,48

In figure 6b – 7d, shows the results of localization of direct contour lines in the image by means of the FFLD, LSD methods and method based on Hough transform. In figure 7 and 8 characteristics of stability (S) of results of localization of direct contour lines in case of change of brightness (ΔY) and contrast (ΔC) of the image are provided. Comparison of number of pixels, which saved the location after change of brightness and contrast of the image, on the direct contour lines with total number of pixels, which was of localization on the source image, on the direct contour lines is made for the quantitative assessment of stability.

It is set that the offered FFLD method benefits in stability of position fix of the localization direct contour lines in comparison with method based on Hough transform to 30% and 20%, but loses to the LSD method to 10% and 30% in case of increase and reduction of image brightness respectively (figure 7). In case of increase in contrast the FFLD method benefits in stability against a method based on Hough transform to 10%, but loses to the LSD method to 20% (figure 13). In case of contrast reduction the FFLD method loses in stability to methods based on Hough transform and LSD to 30% and 15% respectively.

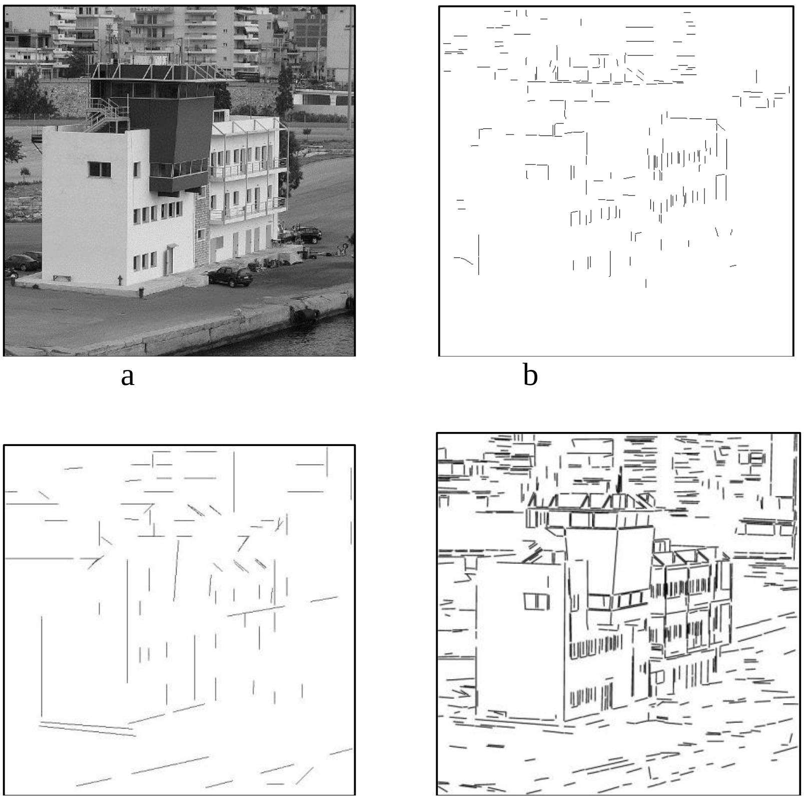


Figure 6. The localization of direct contour lines:
a) source image; b) results of localization for the FFLD method; c) results of localization for method based on Hough transform; d) results of localization for the LSD method

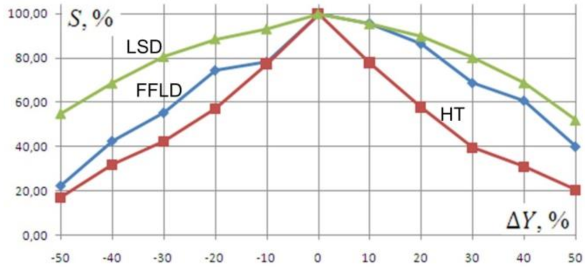


Figure7. Characteristics of stability of results of localization of direct contour lines in case of change of brightness for 50%

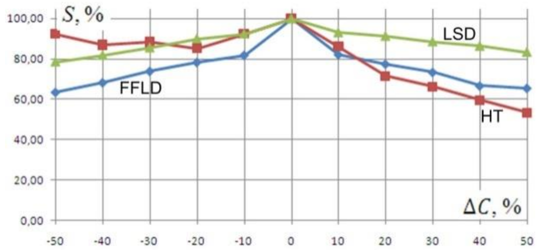


Figure 8. Characteristics of stability of results of localization of direct contour lines in case of change of contrast for 50%

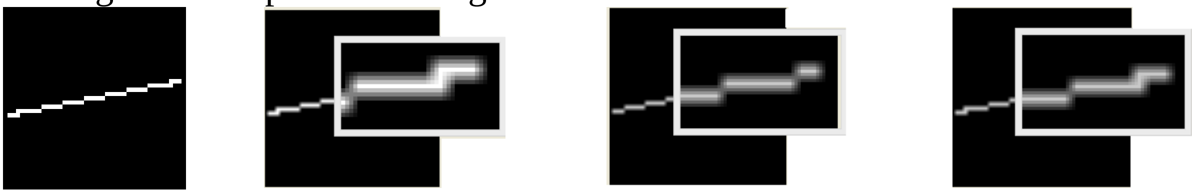
The algorithm of normalization of contour lines on thickness is developed for more effective operation of a method of localization of contour lines based on form-factor. This algorithm is realized in language C++ with use of OpenCV 2.4.10 library. For comparison purposes operations of algorithm the most known algorithm of thinning - algorithm Zong-Suen was realized. Experiment is made on a computer with the following technical characteristics: the processor -

Intel(R) Core(TM) i7-4700HQ CPU @ 2.40 GHz; The RAM – 6 GB; type of system – a 64-bit operating system, the processor x64; an operating system – Windows 8.1.

For comparing of operation of algorithms artificially the created line and the line selected on the image was used.

Artificially the created line was used different size (5, 11, 15, 25, 41, 65 and 101 pixels) with curvature of a deviation from 0 to 3 pixels of rather ideal direct line. Each line was round on an angle from 0 to 180 degrees concerning a y axis. Examples of source and processed by the developed method and algorithm Zong-Suen lines are provided in figures 9-12.

Source image and filtered by the Canny method image are provided in figure 13. The line processed by the developed algorithm of normalization and algorithm of Zong-Suen are provided in figure 14.



a b c d

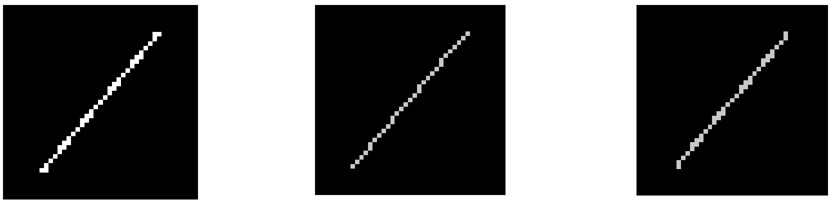
Fig. 9. Results of normalization of a direct line on thickness of long 41 pixels and round on 11 degrees:

a– the source line;

b – the source line with the selected problem section;

c – method of normalization of the contour line on thickness with the selected problem section;

d –Zong-Suen algorithm with the selected problem section.



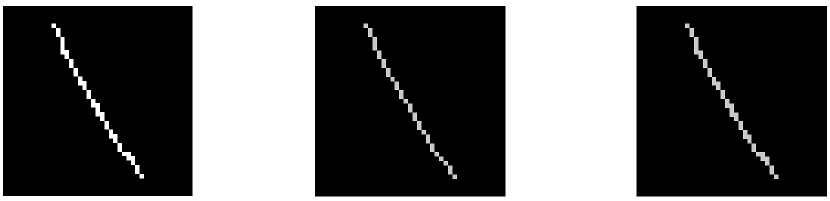
a B c

Figure 10. Results of normalization of a direct line on thickness of long 41 pixels and the curvature 1 pixel, round on 50 degrees:

a– the source line;

b –normalization of the contour line;

c – Zong-Suen algorithm.



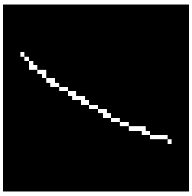
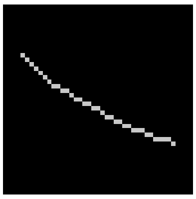
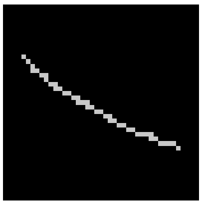
a B c

Figure 11. Results of normalization of a direct line on thickness of long 41 pixels and the curvature 2 pixel, round on 120 degrees:

a– the source line;

b – method of normalization of the contour line on;

c –Zong-Suen algorithm.

a B c

Fig. 12. Results of normalization of a direct line on thickness of long 41 pixels and the curvature 3 pixel, round on 150 degrees:

a– the source line;

b –normalization of the contour line;

c –Zong-Suen algorithm.

a

b

Figure 13. The tested image: a – the source image;
b – the image filtered by Canny's method.

A

B

Figure 14. Results of normalization of the line of the image on thickness:
a – method of normalization of the contour line;
b –Zong-Suen algorithm.

Proceeding from figures 9-12 it is visible that the algorithm of Zong-Suen leaves more "excess" pixels that influences computation of a form-factor.

The assessment of runtime of the developed algorithm of normalization in comparison with algorithm of Zong-Suen is given in table 2.

Table 2 - Comparative assessment of runtime of algorithms

Length of the line, pixels	Algorithm of normalization of the line on thickness, Msec	Algorithm of Zong-Suen, msec
5	0.0570065	0.110769
11	0.101445	0.35529033
15	0.1306015	0.593886
25	0.19512325	1.560834
41	0.29332025	4.1196445
65	0.44578175	10.0513325
101	0.6689235	24.203958

The algorithm of normalization of the contour line on thickness in the developed method of localization of lines based on form-factor is used after segmentation, and algorithm of Zong-Suen – after. The comparative assessment of runtime of segmentation when using each of algorithms is given in table 3.

Table 3 - A comparative assessment of runtime of segmentation when using different algorithms of normalization.

Length of the line, pixels	Algorithm of normalization of the line on thickness, Msec	Algorithm of Zong-Suen, msec
5	0.0387545	0.0395215
11	0.062543667	0.063474
15	0.078612	0.0808365
25	0.11609525	0.118144
41	0.179705	0.1839628
65	0.290381	0.2928783
101	0.4771175	0.4819805

From tables 2 and 3 it is visible that in case of identical costs of segmentation, the provided algorithm works in 1.9..36 times, depending on line length, quicker than algorithm Zong-Suen. It occurs due to smaller number of iterations of the developed method, the one-fold analysis of each pixel of the contour line.
Impact assessments of algorithms of normalization and thinning of the line on determination of a form-factor it is based on calculation of dispersion of the received values of a form-factor for artificial lines and the real image.
Dispersion of a form-factor is calculated on a formula:

$$D = \frac{\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}}{n}, \qquad (4)$$

where n – number of the considered lines of one length for which the form-factor was calculated; x_i – value of a form-factor of the line.

The diagram of dependence of dispersion of a form-factor on curvature of the line of different length it is given in figures 15-21.

The artificial line long 5(figure 15), 11 (figure 16), 15 (figure 17), 25 (figure 18), 41 (figure 19), 65 (figure 20), 101 (figure 21) pixels was round from 0 to 180 degrees with a step 1. Curvature of the line 5 pixels long makes 0 and 1 pixels, curvature of the line 11 pixels long – 0-2 pixels, curvature of lines 15, 25, 41, 65, 101 pixel long – 0-3 pixels.

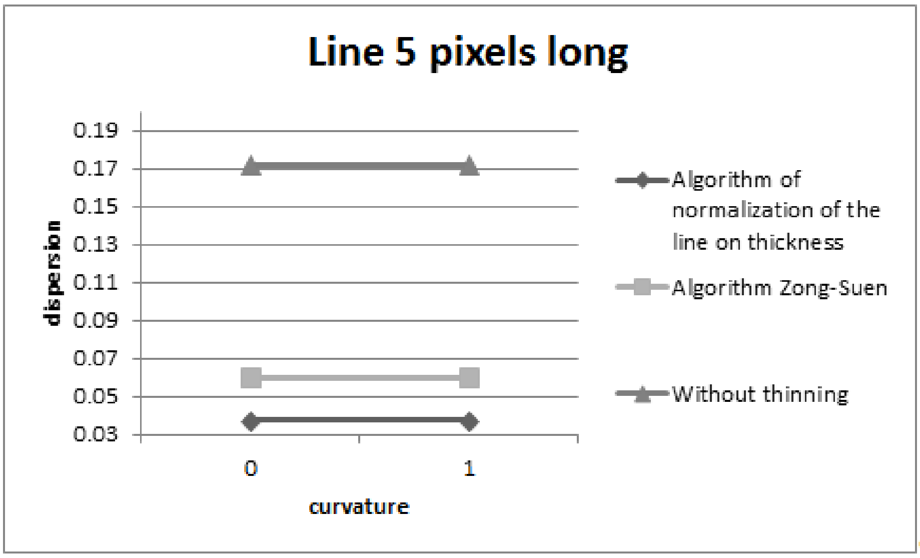


Figure 15 - Dependence of dispersion of a form-factor on curvature of the line 5 pixels long

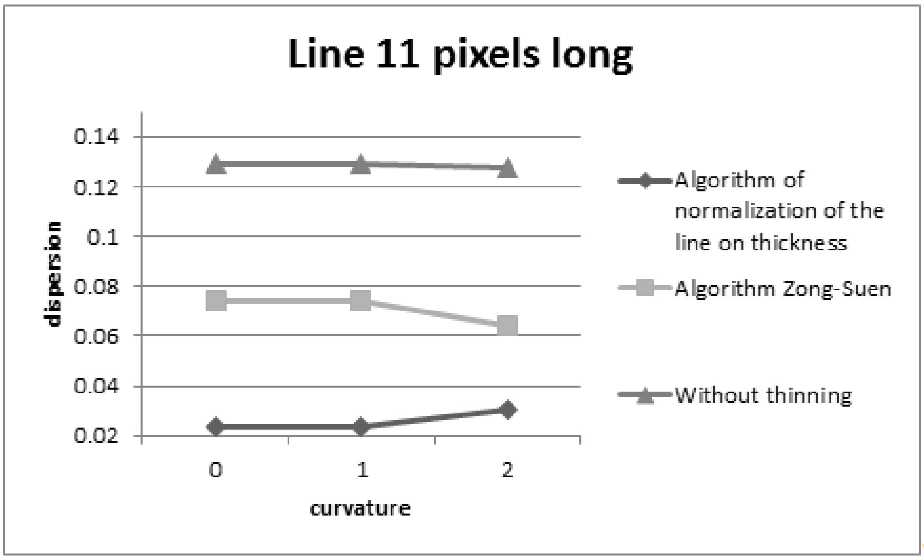


Figure 16 - Dependence of dispersion of a form-factor on curvature of the line 11 pixels long

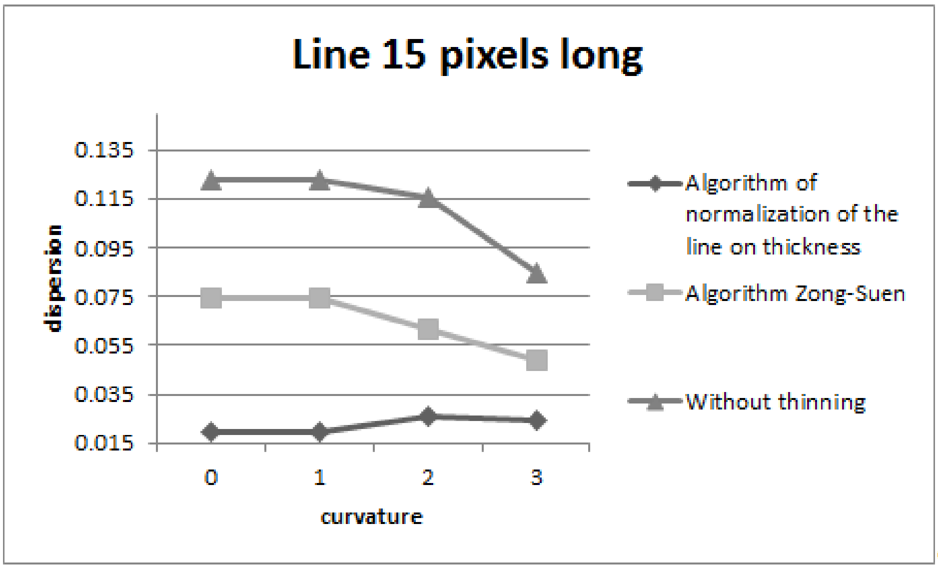


Figure 17 - Dependence of dispersion of a form-factor on curvature of the line 15 pixels long

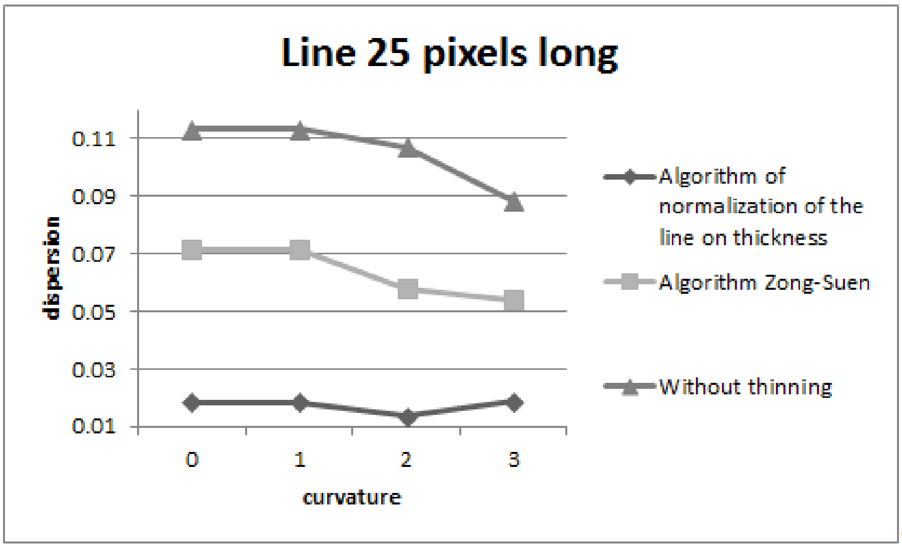


Figure 18 - Dependence of dispersion of a form-factor on curvature of the line 25 pixels long

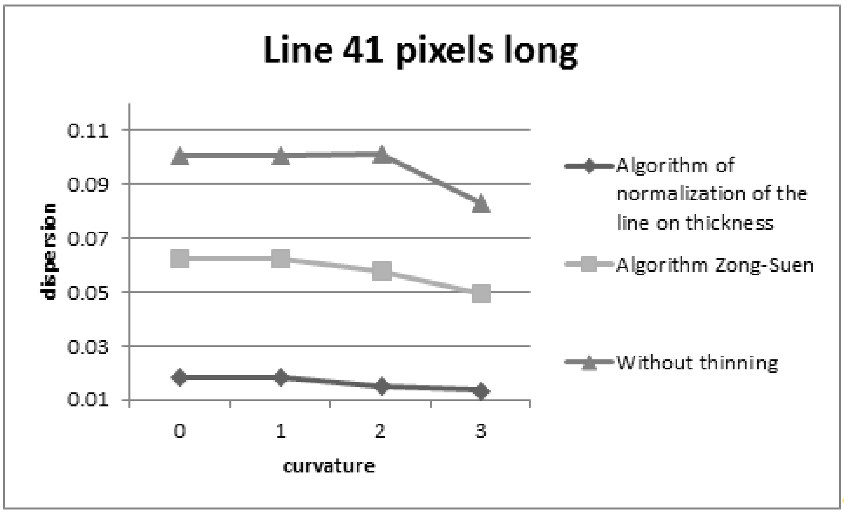


Figure 19 - Dependence of dispersion of a form-factor on curvature of the line 41 pixels long

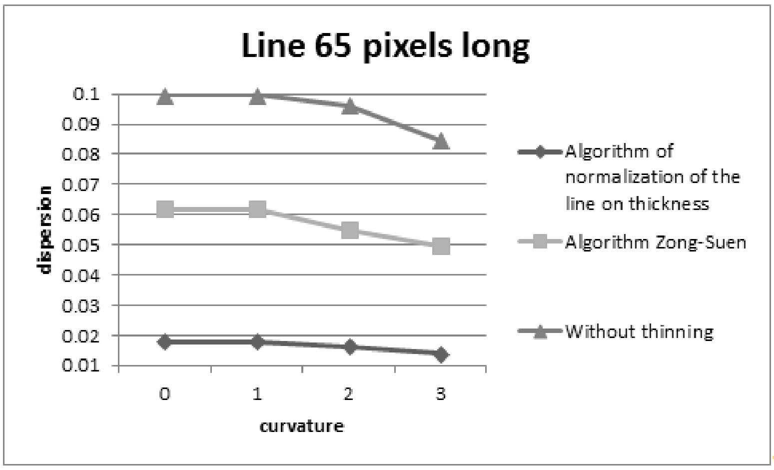


Figure 20 - Dependence of dispersion of a form-factor on curvature of the line 65 pixels long

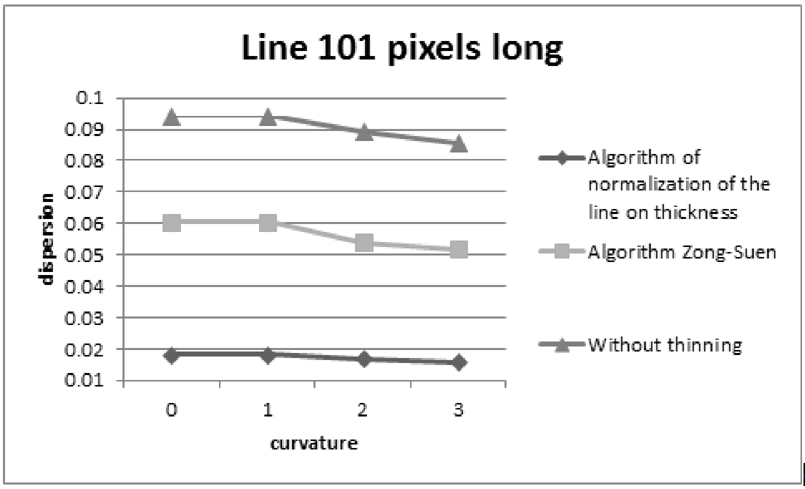


Figure 21 - Dependence of dispersion of a form-factor on curvature of the line 101 pixels long

From figures 15-21 it is visible that use of a method of normalization or thinning in case of calculation of a form-factor improves dispersion last in case of round of the line. It is also shown that when using of the offered normalization method dispersion on average on 0,05 points is lower, than when using algorithm of Zong-Suen. With insignificant increase in curvature of the line dispersion changes slightly.

In table 4 are provided dispersion of a form-factor depending on the selected method of normalization of the line or without its use in case of round of the image (figure 13a) from 0 to 90 degrees with a step to 15 degrees.

Table 4 – An assessment of dispersion of a form-factor for the line of the image.

	Algorithm of normalization of the line on thickness, мsec	Algorithm Zong-Suen, мsec	Without thinning, мsec
dispersion	0.019484	0.103612	0.166342

Comparing of algorithms of normalization and segmentation on time for the line of the real image is given in table 5.

Table 5 - A comparative assessment of runtime of normalization of the line on thickness and segmentation.

	Algorithm of normalization of the line on thickness, мsec	Algorithm Zong-Suen, мsec
normalization time	3.203058	92.42205
segmentation time	1.756553	1.980582

From tables 4-5 it is visible that in case of identical time of segmentation the developed algorithm is nearly 30 times faster than algorithm of Zong-Suen. Also developed algorithm of normalization of contour lines on thickness promotes the best calculation of a form-factor by 10 times that is shown in table 4.

Conclusion

The method of localization of isolated direct contour lines based on form- factor is offered. The entity of a method consists in comparison to unit of value of form-factor – the relations of the size of the contour line (distance between end points) to its length (number of the forming contour pixels). It is set that the offered method provides a scoring in the speed of localization of direct contour lines by 4 times in comparison with the LSD method and by 14,8 times in comparison with method based on Hough transform. It is shown that the offered method benefits in stability of position fix of the localization direct contour lines in comparison with method based on Hough transform to 30%, but loses to the LSD method to 30% in case of image brightness change. In case of increase in contrast the offered method benefits in stability against a method based on Hough transform to 10%, but loses to the LSD method to 20%. In case of contrast reduction the offered method loses in stability to methods based on Hough transform and LSD to 30% and 15% respectively.

The effective method of normalization of the contour line on thickness on the basis of the analysis of orientation of the connected pixels is developed. It is shown that the method works in 1.9..36 times quicker than algorithm Zong-Suen. To shortcomings of a method the impossibility of its application belongs to lines more than 3 pixels thick, on what the main methods of a thinning are directed. However this method is specially developed for deleting excess pixels of a dedicated line.

References

1. Lowe, D. Distinctive image features from scale invariant keypoints/ D. Lowe// International J. of Computer Vision. – 2004. – Vol. 60, № 2. – P. 91–110.

2. Bay, H. SURF: Speeded up robust features/ H. Bay// Proc. of the 9th European Conference on Computer Vision. – Graz, Austria,2006. – Vol. 3951. – P. 404–410.

3. Hirschmuller, H. Realtime correlation-based stereo vision with reduced border errors/ H. Hirschmuller, J.M. Garibaldi// International J. of Computer Vision. – 2002. – Vol. 47, № 1–3. – P. 229–246.

4. Duda, R.O. Use of the Hough Transformation to Detect Lines and Curves in Pictures/ R.O. Duda// Communication of the ACM. – 1972. – Vol. 15, № 1. – P. 229–246.

5. Anver, M.M. Fuzzy edge detection using competition between multiple masks/ M.M. Anver, R.J. Stonier// Proc. of the 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems, CIRAS 2003. – Singapore, 2003. – P. 344–348.

6. LSD: A Fast Line Segment Detector with a False Detection Control/ R. Grompone von Gioi [et al.]// IEEE Transactions on Pattern. Analysis and Machine Intelligence. – 2010. – Vol. 32, № 4. – P. 722–732.

7. Chan, T.S. Line detection algorithm/ T.S. Chan, K.K. Raymond// Proc. of 13th Int. Conference on Pattern Recognition, ICPR 1996. – Vienna, 1996. – P. 126–130.

8. T. Y. Zhang, C. Y. Suen. A fast parallel algorithm for thinning digital patterns// Commun. ACM, vol. 27, no. 3, pp. 236--239, 1984.

9. Молчанова, Грунский. Решение задачи топологического утончения объектов бинарного растра с использованием специализированных агентов.// Информационные управляющие системы и компьютерный мониторинг (ИУС КМ 2013)

10. Brice C. R., Fenema C. L., Scene Analysis Using Regions, Artificial Intelligence, 1, 205-226 (1970).

11. Barrow H. G., Popplestone R. J., Relational Descriptions in Picture Processing, in: Machine Intelligence, Vol. 6, Meltzer B., Michie D., Eds., University Press, Edinburgh, 1971, pp. 377-396.

12. Р. Гонсалес, Р. Вудс Цифровая обработка изображений — М: Техносфера, 2005 – 1007с