*УДК 621.391*

# A FULLY SEEDED REGION GROWING ALGORITHM BY WAVE PROPAGATION

A.T. NGUYEN, T.H. DOAN, V.Yu. TSVIATKOU

*Belarusian State University of Informatics and Radioelectronics, Republic of Belarus*

*Submitted30 March 2020*

**Abstract.** A fully automatic seeded region growing algorithm for image segmentation based on quasi-parallel wave growing of regions around local extrema with a gradual change in the brightness threshold from the extremum value was proposed. Unlike well-known segmentation algorithms, the proposed algorithm allows to separate areas with smooth differences in brightness, and control the number of segments sufficient to approximate images and compactly describe their segments.

*Keywords:* local extrema; image segmentation; region growing; level set based region growing; wave region growing.

## Introduction

Image segmentation aspires to gather pixels into prominent image regions, i.e., regions equivalent to individual surfaces, items or accepted parts of objects. Segmentation is the course of action in which an image is partitioned into constituent objects or parts. It is often the primary and most imperative step in an image analysis task. The outcome of image segmentation is a set of segments that cooperatively cover the intact image, or a set of contours extract from the image. Segmentation is one of the basic digital image processing procedures underlying their analysis, visualization and object-oriented coding [1].

Segmentation accuracy determines the quality of the subsequent processing results. In some cases, the segmentation time may be limited, or it is necessary to control the number of image segments [2]. The main characteristics of image segmentation methods include time, errors, and compact presentation of segmentation results.

Segmentation time is the main factor determining the effectiveness of segmentation methods. Depending on the size, complexity of the image, the chosen method of segmentation and computing resources, it ranges from fractions of a second to several minutes.

Segmentation errors are manifested in the accuracy and stability of the localization of regions when changing the conditions of video recording, leading to a change in brightness, contrast, image shift and rotation [3]. The main cause of errors in segmentation methods in real conditions is the uneven illumination of the scene, which arises due to the instability of the light source, uneven distribution of light over the surface of the object (especially large), and the inability to optically isolate the object from the shadow of other objects [4].

Taking into account the compactness of the presentation of segmentation results [5] allows us to evaluate the effectiveness of methods in terms of requirements for computing resources.

Histogram quantization segmentation does not provide an accurate division of areas due to the assignment of identical numbers to segments with the same brightness. In addition, well-known segmentation methods based on the formation of areas using a watershed [6–9], quantization by a histogram [10], region splitting and merging using the quad-trees [11–13], region growing [14–20], are not effective for separating regions with smooth differences in brightness. All the methods considered do not provide adaptation to the restrictions on the segmentation time and do not allow controlling the number of segments. In this paper, the urgent task is developing a method of image segmentation, taking into account the above disadvantages.

## Research method

***Automatic region growing method***

Segmentation is a process of extracting required features or Region of Interest from an image for future purpose like compression. The given or input image is sliced into multiple regions based on some properties like pixel intensity, texture, position or some local (or) global statistical parameter. The algorithm used in this proposed system for segmentation is Region Growing (RG) method and it consists of two methods. The first method is Seeded Region Growing (SRG) method, it takes a set of seeds as input along with the image and it requires seeds as additional input. The seeds mark each of the objects to be segmented and compare with pixel value. The pixel with the smallest difference measured is allocated to the respective region the difference between a pixel's intensity value and the region's mean, is used as a measures of similarity, this process continues until all pixels are allocated to a region. The second method is Unseeded Region Growing (URG). Simply it starts off with a single region. It differs from seeded region growing as if the minimum mean pixel value is less than a threshold $T$ then it is added to the respective region If not, then the pixel is considered as it is different from all current regions and a new region is created with this pixel [16].

The automatic seeded region growing algorithm is one ofthe simplest region-based segmentation methods. Itperforms a segmentation on an image with examine theneighboring pixels of a set of points, acknowledged as seedpoints, and conclude whether the pixels could be classifiedto the cluster of seed point or not [16, 19]. The algorithmprocedure is as follows

Step1. We start with a number of seed points which havebeen clustered into $N$ clusters, called $C_1, C_2...C_N$. And theposition of initial seed points is set as $P_1, P_2...P_N$.

Step2. To compute the difference of pixel value of theinitial seed point $P_i$ and its neighboring points, if thedifference is smaller than the threshold (criterion) wedefine, the neighboring point could be classified into $C_i$ ,where $i$=1, 2...$N$.

Step3. Recomputed the boundary of $C_i$ and set thoseboundary points as new seed points $P_i$. In addition, themean pixel values of $C_i$ have to be recomputedcorrespondingly.

Step4. Repeat Step2 and 3 until all pixels in image havebeen allocated to a suitable cluster.

In [21] a Level Set based Region Growing (LSRG) method for automatic partitioning of color images into segments was developed. Waves from the selected base points are iteratively emitted. At a base point, the local variance of the data reaches a minimum, which indicates the base point is a suitable representative of its local neighborhood. The local variance is determined by applying a hierarchical gradient operator. The speed of the wave is determined by the color similarity of the point on the front to the current coverage of the wave, and by edge information. The wave front propagation-based image segmentation method was proposed to overcome the limitations of the existing region growing techniques. The segmentation algorithm is summarized as

Step1. Select a base point, initialize a wave front and a segment.

Step2. Keep moving the wave front using the speed function until

(a) Eulerian method: a stopping criteria is fulfilled, go to Step3

(b) non-Eulerian method: all points covered, go to Step4

Step3. Mark the current segment, and remove it from a set of available points. If no point remains then stop, otherwise continue to Step1.

Step4. Find the maximum of the point-wise minimum arrival times. If the maximum is smaller than a threshold then to stop it, otherwise continue to Step1.

***Base points selection***

In [21] base points are iteratively selected. A point is assigned as a base point $b$ if it represents the color distribution within its local neighborhood as relevant as possible. The local color variance $\sigma^2$ and color gradient $\nabla I$ are indicators of the color homogeneity. The likelihood of being a base point is inversely proportional to the gradient and variance. A point has a higher likelihood of being a base point if it has smaller (with respect to a global maximum) gradient and variance values, which shows the color of the point is consistent with its neighbors and color distribution is uniform around the point. This prevents from starting a wave on an edge which will disturb the color homogeneity of the

segment, and enable us to effectively center the initial wave front for more accurate segment boundary localization. Ideally, a base point should be at the center of the segment that it belongs to.

Let $I(x, c)$ be the color value of the point in the corresponding color channel $c$. Assuming the color space is orthogonal,e.g. $RGB$, we define the local color variance within a local window $W$ around point as

$$\sigma^2 = \sum_c \sum_{x \in W} \omega \left[ I(x, c) - \mu(x, c) \right]^2 \quad (1)$$

where $\mu(x, c)$ is the color mean within the window and $\omega$ is an envelope function which has a Gaussian form. For theresults presented in this paper, we assumed the contribution of all points within the window is same, i.e. $\omega = 1$. The colorgradient is given as

$$\left| \nabla I(x) \right| = \sum_c \left| I(x+1, y, c) - I(x-1, y, c) \right| + \left| I(x, y+1, c) - I(x, y-1, c) \right| \quad (2)$$

To compute the variance within a bigger window withoutincreasing the computational load, we applied (1) using the same window size of $5 \times 5$ pixels.

In [22–24] the local maxima or local minima are selected as base points using mathematical morphology for automatic seeded region growing algorithm. Finding Local Extrema (LE) is often solved by mathematical morphology using dilation and erosion operations, respectively. It gives accurate results compared to block algorithms. However, the morphological algorithm has high computational complexity, which is associated with separate processing of maxima and minima, as well as iterative processing of the neighborhoods of all pixels. In this proposed system we developed two algorithms for extracting local extrema in grayscale images with low computational complexity, high accuracy and less memory [25, 26].

***Proposed segmentation method***

To overcome the limitations of the existing region growing techniques a Quasi-Parallel Wave Growing (QPWG) algorithm for automatic partitioning of regions around local extrema is proposed. The essence of the algorithm consists of extracting local extrema in the image, gradually adding new elements to them, taking into account their location around the extrema, the threshold value that is stepwise changed from the extremum value in the opposite direction to avoid blocking the wave growth process. The process of oncoming wave propagation continues until all areas are segmented. The segmentation algorithm is summarized as

Step1. Find all local extrema as seed points, start with a number of local extrema which havebeen clustered into $N$ regions, called $R_1, R_2 ... R_N$. And theposition of local extrema is set as $E_1, E_2 ... E_N$. Initialize all wave fronts and all segments around the local extrema.

Step2. Keep quasi-parallel moving the wave fronts around all extrema using the extreme values to compare to neighbor pixels.To compute the difference of pixel value of theinitial seed point $E_i$ and its neighboring points, if thedifference is smaller than the current threshold (initial threshold $T = 1$), the neighboring point could be classified into $R_i$ ,set thoseboundary points as new seed points for the next loop, otherwise save current seed points into a memory stack $STACK$, where $i$=1, 2...$N$.

Step3. If the $STACK$ is not empty then the current threshold is added to 1($T \leftarrow T + 1$) value and set seed points in the stack as new seed points for the next loop using the current threshold.

Step4. Mark all segments. If no point remains then stop, otherwise continue to Step 2 and 3.

To reduce redundancy and increase the stability of segmentation to noise, low-pass filtering can be used as pre-processing.

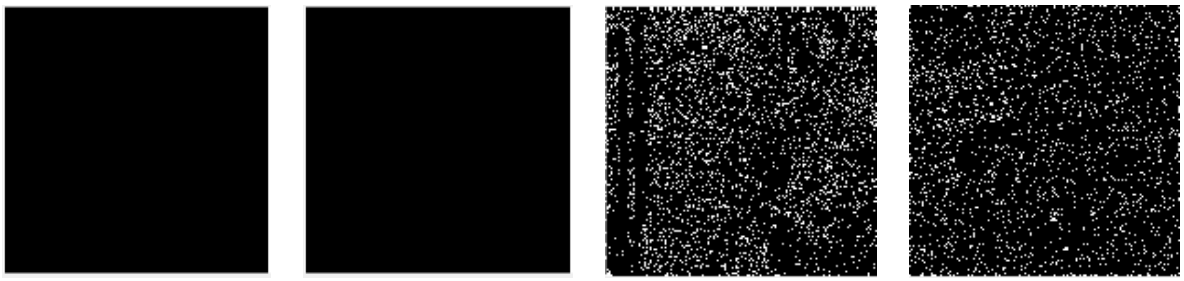
As a result of the proposed algorithm, a segmentation matrix is formed, the value of each element of which indicates the number of the segment to which the pixel of the segmented image belongs to the corresponding coordinates. The number of segments obtained coincides with the number of local extrema, which allows controlling the number of segments, as well as the location and segment size.

**Results and analysis**

To assess the effectiveness of segmentation methods four test grayscale images are shown in Fig. 1: Best and Worst with smooth differences in brightness, a low-frequency image Lena and a high-frequency image City in $256 \times 256$ pixels. The local extrema of these images are shown in Fig. 2.

*a* *b* *c* *d*

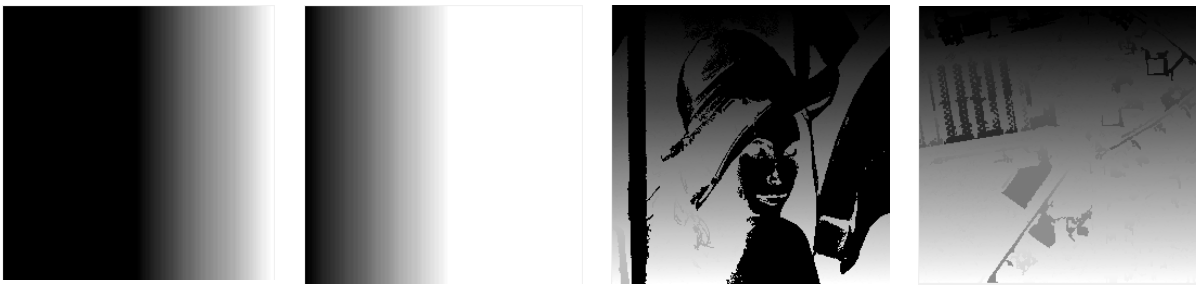
Fig. 1. Test grayscale images: *a* – Best; *b* – Worst; *c* – Lena; *d* – City



*a* *b* *c* *d*

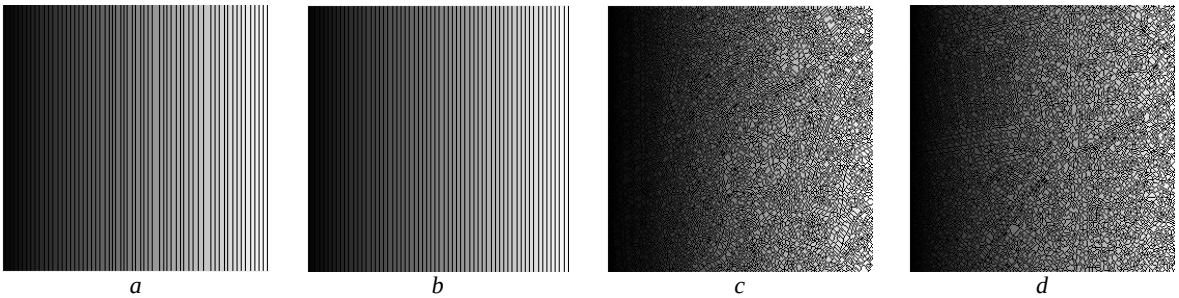
Fig. 2. Local extrema of test images: *a* – 2 extrema;
*b* – 2 extrema; *c* – 7505 extrema; *d* – 4830 extrema

In Fig. 3–8 the results of segmentation of the above grayscale images are shown using region splitting and merging using quad-trees (Splitting&Merging), gradient-based watershed algorithm, unseeded region growing (URG), local extrema based seeded region growing usingmorphology (SRG+LE(M)), level set based region growing (LSRG) and the proposed segmentation method usingmorphology (QPWG+LE(M)) or using new algorithms to find local extrema [25, 26] (QPWG+LE).



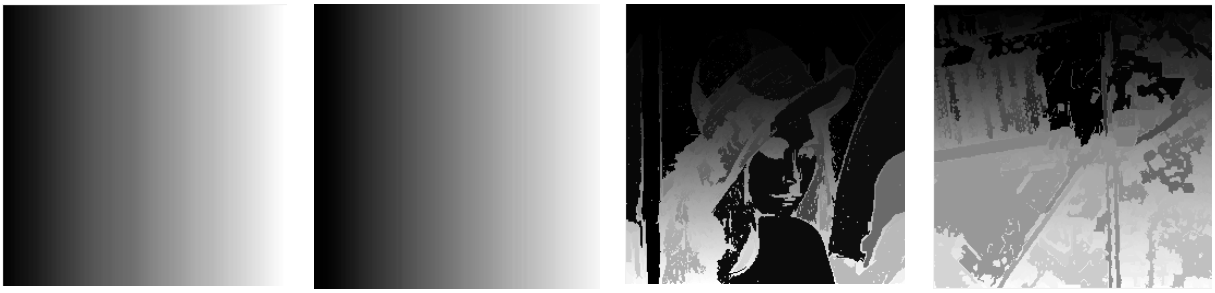
*a* *b* *c* *d*

Fig. 3. The result of applying the algorithm Splitting&Merging: *a* – 79 segments( $T = 0$ );
*b* – 77 segments( $T = 0$ ); *c* – 25753 segments ( $T = 0$ ); *d* – 40081 segments ( $T = 0$ )



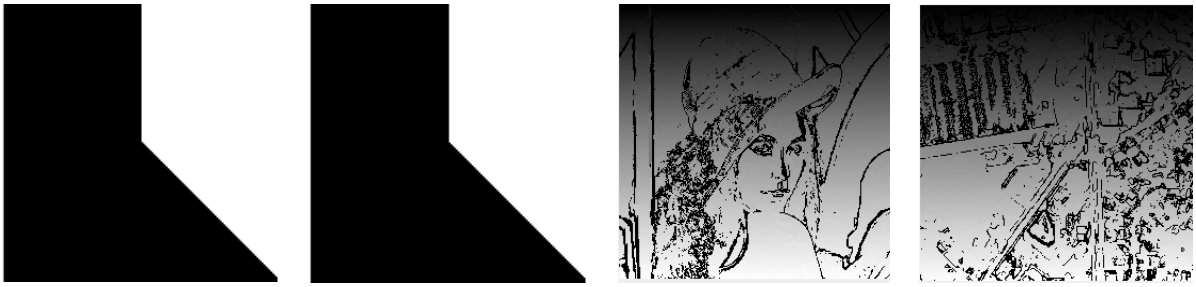
*a* *b* *c* *d*

Fig. 4. The result of applying gradient-based watershed algorithm:
*a* – 62 segments; *b* – 62 segments; *c* – 8728 segments; *d* – 8003 segments



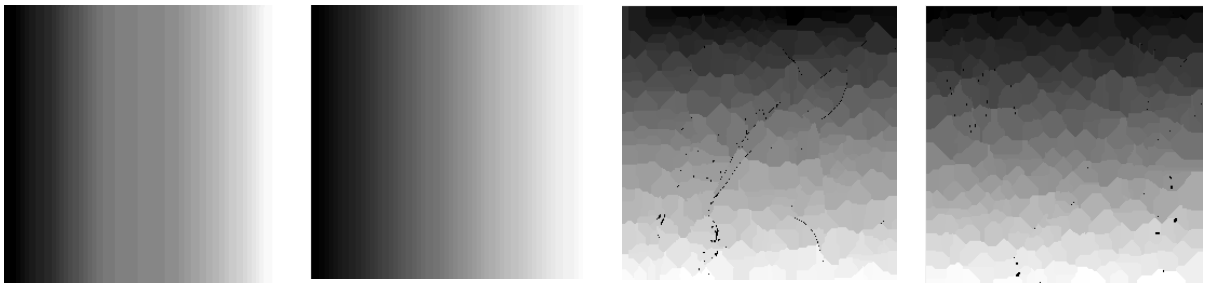
*a* *b* *c* *d*

Fig. 5. The result of applying unseeded region growing algorithm:*a* – 152 segments ($T = 0$);
*b* – 145 segments ($T = 0$); *c* – 6988 segments ($T = 5$); *d* – 9323 segments ($T = 5$)



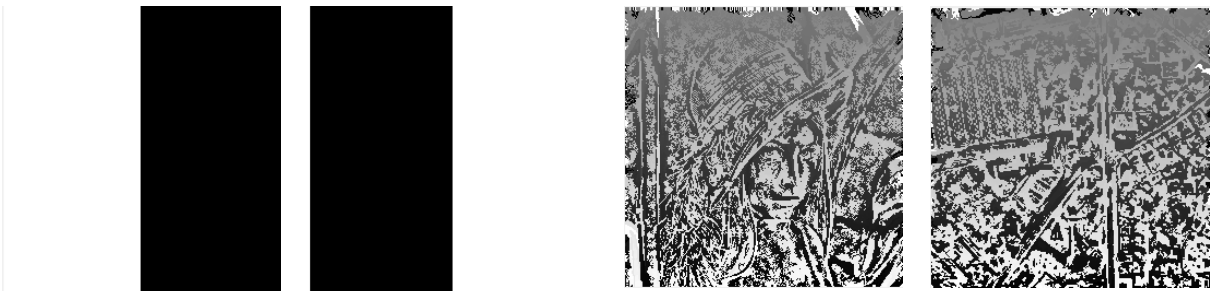
*a* *b* *c* *d*

Fig. 6. The result of applying local extrema based seeded region growing algorithm using morphological operations:*a* – 2 segments ($T = 5$); *b* – 2 segments ($T = 5$);
*c* – 7507 segments ($T = 5$); *d* – 4830 segments ($T = 5$)



*a* *b* *c* *d*

Fig. 7. The result of applying level set based region growing algorithm:*a* – 39 segments ($T = 20$);
*b* – 54 segments ($T = 20$); *c* – 305 segments ($T = 40$); *d* – 269 segments ($T = 40$)



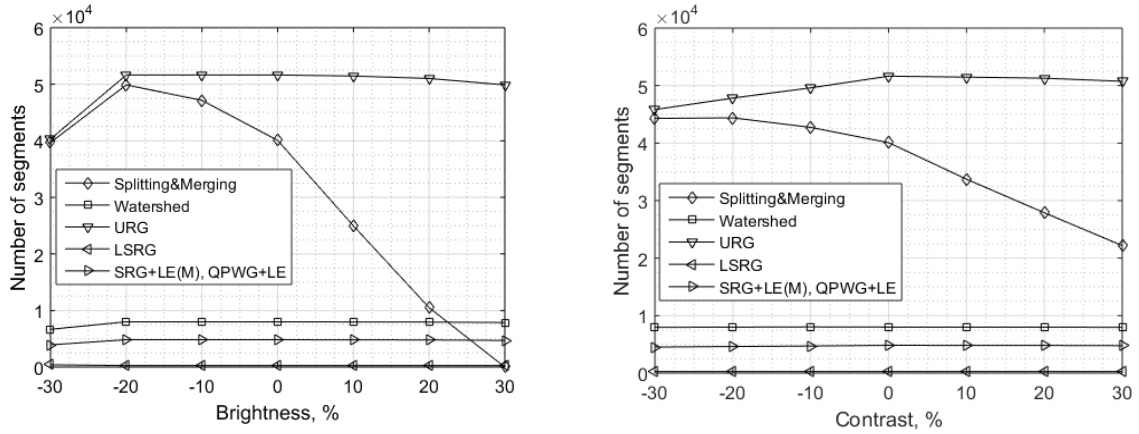
*a* *b* *c* *d*

Fig. 8. Theresultofapplyingtheproposedalgorithm QPWG+LE:
*a* – 2 segments; *b* –2 segments; *c* – 7505 segments; *d* – 4830 segments

Fig. 3–8 show that the proposed algorithm ensures a stable border position and the allocation of two regions for any value of the brightness drop (Fig. 8*a*, *b*). Other algorithms for some values of the brightness difference show an error or redundancy in segmentation, extracting a lot of redundant areas in the image. Thus, the proposed algorithm provides an increase in the sensitivity of segmentation to differences in brightness, the shapes of the obtained segments of which are not complex and depend on the locations, shapes, and distances between the extrema in the image, which allows them to be compactly described for subsequent processing. Such a compact description of segments is of interest for image recognition and compression.

The effectiveness of the proposed algorithm and known segmentation algorithms are evaluated. As for indicators of effectiveness, we used the time of segmentation, the stability of borders and the number of segments.

Fig. 9 shows dependences of the number of segments on brightness and contrast changes, characterizing the stability of the segmentation results. Stability is estimated by the ratio of the number of segments for the base image to the number of segments for the modified image subjected to a change in brightness and contrast. It was found that the proposed algorithm wins in the stability of the number of segments when changing brightness, contrast compared to well-known algorithms




*a* *b*

Fig. 9. Dependences of the number of segments on brightness and contrast changes of image City (Fig. 1*d*):*a* – Brightness change; *b* – Contrast change

In Table 1 the results of estimating the segmentation time for the considered algorithms are implemented in MATLAB and MATLAB ®Image Processing Toolbox (R2015b) using Intel Core i3 3.1 GHz system with 6 GB of RAM. For this experiment, we used four grayscale images: Lena, Barbara, City, and Man.

Table 1. **Comparison of segmentation performance**

| Algorithms | | Programming languages | Segmentation time, s | | | |
|---|---|---|---|---|---|---|
| | | | Lena 128×128 | Barbara 256×256 | City 512×512 | Man 1024×1024 |
| 1 | Splitting&Merging | Visual C++/MATLAB | 0,029 | 0,077 | 0,311 | 1,246 |
| 2 | Watershed | Visual C++/MATLAB | 0,017 | 0,043 | 0,162 | 0,917 |
| 3 | URG | MATLAB | 0,027 | 0,068 | 0,208 | 0,808 |
| 4 | SRG+LE(M) | Visual C++/MATLAB | 0,036 | 0,105 | 0,478 | 2,058 |
| 5 | LSRG | Visual C++/MATLAB | 0,178 | 0,691 | 2,821 | 11,465 |
| 6 | QPWG+LE(M) | Visual C++/MATLAB | 0,094 | 0,293 | 0,998 | 4,117 |
| 7 | QPWG+LE | MATLAB | 0,067 | 0,247 | 0,937 | 3,374 |

Table 1 follows that the proposed algorithm wins in segmentation speed compared to QPWG+LE(M) and LSRG in any image size, but loses in segmentation speed compared to URG, SRG+LE(M), Splitting&Merging, and gradient-based watershed algorithms written in Visual C++/MATLAB programming languages. The processing speed of the proposed algorithm is faster when implemented in C++ programming language.

## Conclusion

In this paper aquasi-parallel wave growing algorithm of regions around local extrema for image segmentation is proposed. The essence of the algorithm consists of the quasi-parallel growing of regions around local minima and maxima selected as the initial seed points. This provides automatic separation of areas with a smooth difference in brightness, which well-known algorithms segment with errors. It is shown that the proposed algorithm makes it possible to clearly distinguish segments and control their number in comparison with the known segmentation algorithms. To reduce redundancy and increase the stability of segmentation to noise, low-pass filtering can be used as pre-processing.

## References

1. Milan S. [et al.] // Thomson press, west. 2008. P. 175–240.
2. Gonzales R., Woods R., Eddins S. // Technosphere. 2006. P. 396–443.
3. Fabijańska A., Strzecha K., Sankowski D. // CADSM'2007, Polyana, Ukraine, 20–24 February. 2007. P. 439–441.
4. M. Chandrakala and P.D. Devi // International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE). 2016. Vol. 5. P. 163–168.
5. Gladureva A.Yu. // Automatics, Lviv, Ucraina. 2011. P. 348–349.
6. Lalitha M., Kiruthiga M., Loganathan C. // International Journal of Science and Research (IJSR). 2013. Vol. 2. P. 348–358.
7. Gauch J. M. // IEEE transactions on image processing. 1999. Vol. 8. P. 69–79.
8. Khiyal M.S.H., Khan A., Bibi A. // Informing Science and Information Technology. 2009. Vol. 6. P. 876–886.
9. Arindrajit Seal, Arunava Das, Prasad Sen // International Journal of Computer Science and Information Technologies (IJCSIT). 2015. Vol. 6. P. 2295–2297.
10. Chang J.H., Fan K.Ch., Chang Y.L. // Image and Vision Computing. 2002. Vol. 20. P. 203–216.
11. Muhsin Z.F. [et al.] // The Imaging Science Journal. 2014. Vol. 62. P. 56–62.
12. Xiaolin Wu // IEEE transactions on pattern analysis and machine intelligence. 1993. Vol. 15. P. 808–815.
13. Dass R., Priyanka, Devi S. // International Journal of Electronics & Communication Technology (IJECT). 2012. Vol. 3. P. 66–70.
14. Adams R., Bischof L // IEEE Trans. Pattern Anal. Mach. Intelligence. 1994. 16 (6). P. 641–647.
15. Lin Z., Jin J.S., Talbot H // Conferences in Research and Practice in Information Technology, Sydney, Australia. 2000. P. 31–37.
16. Mancas M., Gossellin B., Macq // Proceedings of twenty-six conference on Image Processing: Algorithms and Systems. 2006. P. 388–398.
17. Singh K.K., Singh A. // International Journal of Computer Science Issues. 2010. Vol. 7. P. 414–417.
18. Shih F.Y., Cheng S. // Image and Vision Computing. 2005. P. 877–886.
19. Sharma Ritu, Sharma Rajesh // International Journal of Innovative Research in Computer and Communication Engineering. 2014. Vol. 2. P. 5686–5692.
20. Mohd Saad N. [et al.] // Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, 14-16 March. 2012. P. 674–677.
21. Porikli F.M. // International Society for Optics and Photonics. 2004. Vol. 5298. P. 536–543.
22. Malek A.A. [et al.] // Procedia-Social and Behavioral Sciences. 2010. Vol. 8. P. 634–639.
23. Han X. P., Fu D. M. // Application Research of Computers. 2007. Vol. 24. P. 158–160.
24. Vartak A.P., Mankar V. // International Journal of Computer Science and Applications. 2013. Vol. 6(2). P. 161–165.
25. Nguyen A.T., Tsviatkou V.Yu. // International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE). 2019. Vol. 8. P. 1–10.
26. Nguyen A.T., Tsviatkou V.Yu. // Informatics. 2019. 16(3). P. 23–36.