Objects Segmentation in Augmented Reality Environment

Ievgen Sidenko, Tetiana Misiuk, Galyna Kondratenko and Yuriy Kondratenko

Petro Mohyla Black Sea National University, 68th Desantnykiv Str., 10, Mykolaiv, 54003, Ukraine

Abstract
The paper discusses objects (images) segmentation algorithms that are applicable in mobile applications with augmented reality. An example of image processing with virtual objects by different algorithms (the MeanShift algorithm, the GrabCut algorithm, the k-means algorithm) is considered. Different libraries, tools, and environments to implement segmentation algorithms were analyzed, such as Scikit-image, Pixellib, OpenCV, Point Cloud Library. The application was created for mobile devices running iOS 10 and higher. The GrabCut algorithm turned out to be the best algorithm for image processing. The processing result was the closest to the expected one. Although the algorithm has some errors. Despite the fact that the area that was contoured turned out to be the clearest and most complete in comparison with other algorithms, this area also includes areas of the image that do not belong to the objects under study.

Keywords 1
Augmented reality, mobile applications, image segmentation, virtual scene

1. Introduction

The creation of applications using augmented reality today does not require a lot of resources. All the necessary materials and tools can be found in the public domain. However, the question arises about the quality of such projects and their practical application.

At the moment, the most common use of augmented reality is in game development. However, the use of augmented reality in education is gaining more and more popularity. This means that there is a need to create applications that will provide rich capabilities for working with virtual objects using a small amount of device resources [1, 2].

Image processing allows you to get information from the scene that the user sees. Segmentation is the technique of dividing an image into specific segments, which are called objects. It can be used for object recognition, estimation of occlusion boundaries in images with dynamic objects or in stereos, image compression, editing them, or to search for similar images in databases [1].

The purpose of segmentation is to modify the representation of an image (selection of objects) for further analysis, in particular the level of brightness, noise, detail when zooming, blur, the presence of artifacts and defects. The result of segmentation is a certain number of objects in the image.

The problem in this case is the number of tasks that must be processed by the system of the mobile device to display the desired result by the user. In this case, there are two ways of development of events. The first option is the loss of the quality of the results [2, 3]. That is, the position of virtual objects on the scene may be inaccurate, offset from the target position. The second option is to obtain sufficiently accurate results of image processing from the camera, but with a large amount of time required to process the image [4, 5, 6]. Many image processing methods have been proposed to obtain a satisfactory result for an optimal period of time. The question arises about improving the existing methods of image segmentation by combining image segmentation algorithms [7, 8, 9].

2. Related works and problem statement

The integration of cameras and multicolor displays has made mobile phones an ideal platform for augmented reality (AR). Today, the use of mobile phones in all spheres of life can lead to the fact that AR will be used in even more applications to interact effectively with the environment.

The first mobile augmented reality systems, such as the Feiners Touring Machine, which was launched in 1997, included a laptop and display (HMD). Using the Rekimotos Transvision system, the use of a handheld display for viewing simple objects in AR has been investigated [1, 3].

The first phones did not have enough processing power, so researchers also looked at thin client approaches. For example, the AR-Phone project used Bluetooth wireless technology to transfer image data from a phone camera to a server to overlay additional graphic elements, which took a few seconds per image.

Image segmentation can be a powerful technique in the early stages of the diagnostic and treatment pipeline for many conditions requiring medical imaging, such as CT or MRI [8, 9]. In fact, segmentation can effectively separate homogeneous areas, which may include critical organ pixels, lesions, and the like. However, there are significant problems including low contrast, noise, and various other imaging inaccuracies.

Mobile Visual Search (MVS) is a specialized search system for mobile devices. In Mobile Image Search (MIS), any information can be found on the Internet using an image of a query captured with a mobile phone or using specific keywords. The main characteristics and parameters of the image are used to search and compare with existing images in the database, and then in response to the user are sent the results of the query [3].

Object recognition applications have become widespread in mobile stores due to the simple interface and intuitive operation of such applications. So, for example, many applications have been created to recognize the types of dishes the user has consumed; some applications were created to recognize text data; systems for tracking a user's face and emotions based on an extensive database of human facial expressions [1, 4, 7].

Image segmentation is widely used in the search for anomalies in medical images, in the selection of objects on satellite images, in traffic control systems and in preparatory work for analyzing text on an image [10, 11, 12]. Therefore, the creation of a faster and more efficient image segmentation algorithm is one of the key issues in the field of digital image processing.

It is quite difficult to find popular mobile applications that use image segmentation algorithms. These include applications that are used in medicine. For example, SkinsCanApp uses additional accessories such as a microscope for mobile phones. You also need a phone with good zoom to use this app. This application is aimed at a narrow area of research and the application is also used to classify objects, rather than highlighting these objects in the image on a mobile device. Several more studies were found that considered the possibility of creating applications using image segmentation algorithms, but the research data was aimed at the possibility of classifying objects on the input images without rendering the results of the algorithms.

The purpose of this paper is to describe and consider image segmentation algorithms in the context of their application for mobile devices, highlight the key principles of their operation and describe the process of creating an application with augmented reality, in which the image of a virtual scene is segmented with real and virtual objects. The novelty is to develop an AR application for objects segmentation with the implementation of an algorithm that is efficient in terms of energy efficiency, speed, quality of segmentation, accuracy, etc.

3. Object segmentation algorithms

One of the problems that often arises when analyzing an image is the division of the entire set of pixels into groups according to a certain characteristic. This process of dividing a set of points is commonly called image segmentation. The most widespread are two segmentation methods: by brightness and by color coordinates [13]. Segmentation by brightness is used for grayscale images. Color segmentation is used for color images. It is customary to consider the segmentation problem as a formalization of the problem of highlighting some object in the image from the background [14, 15].

The quality of the clustering algorithms significantly depends on the properties of the original image, such as brightness distribution, object shape, blurring of boundaries, etc.

The MeanShift algorithm combines objects that have the same or similar properties. Pixels with similar characteristics are grouped into one segment, and the result of segmentation is an image with homogeneous areas [16, 17, 18].

It is easier to describe the thickened points, the function of the thickness is introduced:

$$f(\vec{x}) = \frac{1}{Nh^d} \sum_{i=1}^{N} K(\frac{(\vec{x} - \vec{x}_i)}{h}),$$
(1)

where $\vec{x}$ is the vector of features of the i pixel; d is the number of properties; N is number of points; h is a parameter that is responsible for smoothness; $K(\vec{x})$ is a core.

The function maxima are located at the points of condensation of image points in the feature space. Pixels belonging to the same local maximum are combined into one segment [19].

When choosing coordinates of points and intensities by colors as features, pixels with similar colors and located close to each other will be combined into one segment. Accordingly, if you choose another feature vector, then the merging of pixels into segments will already follow it. For example, if we remove coordinates from features, then the sky and the lake will be considered one segment, since the pixels of these objects in the feature space would fall into one local maximum [20, 21].

If the object to be selected consists of areas that differ greatly in color, then MeanShift will not be able to combine these regions into one, and our object will consist of several parts. But the algorithm works well with a uniformly colored subject on a colored background. Also, MeanShift is used in the implementation of the tracking algorithm for moving objects.

The GrabCut segmentation algorithm is an interactive object selection algorithm [22]. It is based on the GraphCut interactive segmentation algorithm, where the user needs to put markers on the background and on the object. The image is treated as an array $z(z_1, \ldots, z_n, \ldots, z_N)$, z is a pixel intensity value, N is the total number of pixels. To separate the object from the background, the algorithm determines the value of the elements of the transparency array $a(a_1, \ldots, a_n, \ldots, a_N)$ wherein a_n can take two values if $a_n = 0$, means the pixel belongs to the background, if $a_n = 1$, means an object. The internal parameter contains a histogram of the foreground intensity distribution and a histogram of the background:

$$\theta = \{h(z; a), a = 0, 1\}.$$
(2)

The task of segmentation is to find unknowns a_n. The energy function is considered:

$$E(a, \theta, z) = U(a, \theta, z) + V(a, z).$$
(3)

Moreover, the minimum energy corresponds to the best segmentation:

$$U(a, \theta, z) = -\sum_n \log h(z_n, a_n).$$
(4)

$$V(a, z) = \sum_{(m,n) \in C} \frac{1}{dis(m, n)} [a_n \neq a_m] \exp(-\beta(z_m - z_n)^2),$$
(5)

where $V(a, z)$ is the term responsible for the connection between pixels. The sum goes over all pairs of points that are neighbors, $dis(m, n)$ is an Euclidean distance; $[a_n \neq a_m]$ is responsible for the participation of pairs of pixels in the sum, if $a_n = a_m$, then this pair will not be counted; $U(a, \theta, z)$ is responsible for the quality of segmentation, that is, the separation of the object from the background.

Having found the global minimum of the energy function E, we obtain the transparency array $\hat{a} = argmin_a E(a, \theta)$. To minimize the energy function, the image is described as a graph and a search is made for the minimum cut of the graph. Unlike GraphCut, the GrabCut algorithm considers pixels in RGB space, therefore, to describe color statistics, a Gaussian Mixture Model (GMM) is used [23, 24].

The k-means algorithm (method) is a widely used clustering method. The purpose of the algorithm is to split the input image with N pixels into K clusters, where K is given by the user. Clusters represent some grouping of pixels, which depends on the values of pixels in the picture, but does not necessarily depend on their location in the original image [25, 26].

Let it go $X = \{x1, \ldots, xN\}$ lots of N image points, $V(x_i)$ is the property vector associated with the pixel x_i. Consider the main steps of the k-means method.

Step 1. Initialization of parameters. The center of each of the K clusters is initialized with the values of potential vectors of properties. In the classical method, k means values of each element of the property vector are selected at random from the set of all possible values for this element. For example, if the property vector is (R, G, B) and represents the intensity of corresponding colors, the first element of R will be randomly selected from all possible intensities of red.

Step 2. Rigid distribution of properties across clusters. When each of the K clusters C_k has a center μ_k. All pixels are added to the cluster with the nearest center, based on the distance function, determines the distance between two property vectors. After that, each pixel belongs to only one cluster C_k.

Step 3. Recalculation of the parameter. The cluster centers are recalculated using the values of the property vectors of all points in each cluster. Thus, the center μ_k calculated by the formula:

$$\{V\ (x_i)\ |\ x_i \in C_k\}. \tag{6}$$

Steps 2 and 3 are repeated until the cluster centers stop changing [26]. This will happen when not a single pixel from one cluster is moved to another on the next iteration. Fig. 1 illustrates k-means clustering for a color image.

Figure 1: Clustering by the k-means method on the example of a building

As a result of executing the clustering algorithm, 4 clusters were identified in the picture. It is worth noting that the roof of the building and the surface on which people walk are roughly the same color in the image, so they belong to the same cluster.

You can use different libraries, tools, and environments to implement segmentation algorithms. For example, a cross-platform environment that is used to develop segmentation and image registration programs. The Insight Segmentation and Registration Toolkit (ITK) was designed with the financial support of the National Medical Library (USA) as an open source algorithm for image analysis of the Visible Human Project. This toolkit provides advanced segmentation algorithms, and provides control of the configuration process based on the CMake system. It generates control files.

Scikit-image is a Python package designed for image processing and uses NumPy arrays as image objects. It implements algorithms and utilities for use in research, educational and industrial applications. The library is well documented with many practical examples. The package is imported as a skimage, and most functions are inside submodules [1, 12, 18, 27, 28].

Pixellib is a Python programming language library for segmentation problems. The library allows segmentation not only on images but also on videos [4].

The library allows to implement segmentation models without theoretical knowledge of neural networks. One of the tasks that PixelLib can solve is to edit the background of an image or video. The functionality of the library allows you to create a virtual background for images and videos; erase the background on images and videos; paint the background in a certain color; make a black and white background, etc. PixelLib supports semantic segmentation of 20 unique objects (Fig. 2). R2P2 Medical Laboratory uses PixelLib to analyze medical images in the neonatal (neonatal) intensive care unit. PixelLib is integrated into unmanned cameras to segment instances of real-time video streams. On iOS, the library is used to perform semantic and instance image segmentation [22, 24].

Open Source Computer Vision Library is a library of algorithms for different tasks of image processing. Developed for widely used programming languages and platforms. OpenCV is licensed by BSD and is free for both academic and commercial use. OpenCV has interfaces for C ++, Python and Java, can run on widely used desktop and mobile platforms [5, 7, 29, 30, 31].

(a)   (b)

Figure 2: Images: before (a) and after (b) segmentation using the Pixellib library

Point Cloud Library (PCL) is a famous open source library for processing images using point cloud technology. This library contains many modern techniques, in particular, for performance evaluation, image segmentation, visualize elements, filter noise and combine objects. Appropriate algorithms can be used for better modeling of production parts of high complexity and accuracy, for visualization of organs in medicine, for smooth animation of graphic objects in game applications, for rendering and creating models of architectural objects, for landscape designers. PCL is an open source library with BSD license. It can run on widely used desktop and mobile platforms [7, 9].

The Perception package provides a set of tools for creating large-scale datasets for learning and testing computer vision. At the moment, it focuses on several options for using the camera (Fig. 3) and will eventually be extended to other forms of sensors and machine learning tasks [1, 5, 7, 9].
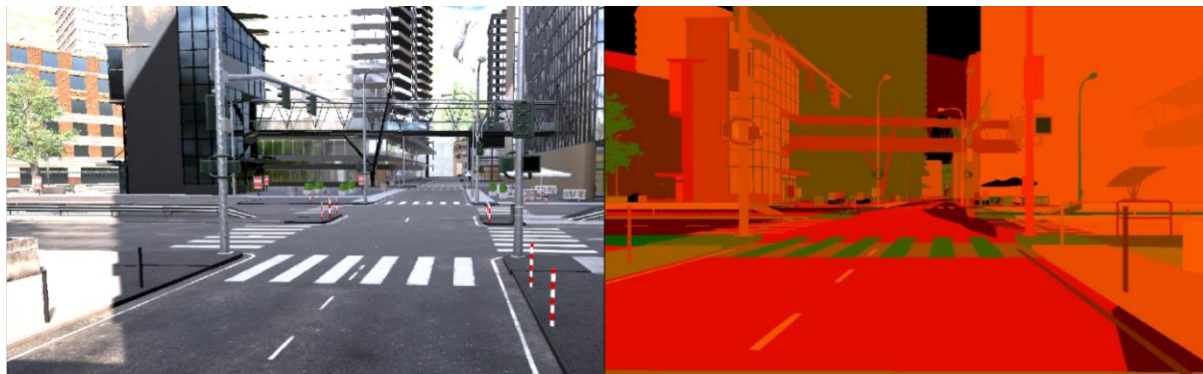

Figure 3: Example of Perception Package

After researching these libraries, it was decided to use the OpenCV library [32, 33, 34, 35] for Unity 3D. This library is imported into the project through the package import option. The list of imported files displays test scenes for viewing examples from OpenCV plugin developers.

4. Implementation of object segmentation algorithms

The OpenCV plugin for Unity was used to create the application. Due to some features of the plugin that was used, it was not possible to create an apk file for the mobile device. Therefore, it was decided to create a mobile application with augmented reality, which creates a snapshot from a virtual camera and saves it in Texture2D format. This file is processed in the editor in runtime and the result is displayed on the screen [36].

The application was created for mobile devices running iOS 10 and higher. Test launches of the application were carried out on just such a device. It is assumed that with the expansion of the functionality and quality of the application, it is possible to create applications for other mobile operating systems.

Image processing by the K-Means algorithm is shown in Fig. 4. To perform the actual segmentation of the image, an array is created containing all the clusters created during the initialization phase. In addition, a loop is implemented to iterate over the next array, and at each iteration of the next loop, it goes through the superpixel array. A linear search is performed for each superpixel. Next, a new image is created from the selected points for a new cluster. The color of each pixel is replaced with the color of the centroid superpixel. The new image in this case will contain pixels of the same color that define a specific area of the original image. Finally, a new cluster is added to the array of clusters [27, 28, 29].

After that, the coordinates of the center point are calculated and a check is made to see if the centroid superpixel of the newly built cluster has been moved relative to the center point of the parent cluster. In this case, the newly built cluster is added to the array of clusters, otherwise it is simply skipped based on the hypothesis that the new and parent clusters are identical.

The above process is repeated for each specific cluster in the target array of clusters until there are no more clusters to process.

To get a finished image that will be segmented, it is necessary to combine images from all these clusters embedded in the whole image. Obviously, the image associated with each of these clusters will contain an area of the original image that has a specific color. To calculate the distance between two 3D color vectors (R; G; B), a 3D version of the Euclidean distance formula is used. At the input, we receive an image from a camera with a 3D object on a virtual scene, and at the output, the result of processing the image from the camera using an algorithm (Fig. 4).

Figure 4: The result of image processing by the K-Means algorithm

In the image, not only objects are highlighted, but also certain areas on these objects into separate clusters. For example, for the world on the object from the illumination lamp it is allocated into a separate cluster, it outputs unreliable information about the current object on the screen. As a result, further work with the data may be performed incorrectly.

When using the Grabcut algorithm in the developed application, the user needs to enter a frame as the target segmentation location in order to achieve separation/segmentation of target and background. This is a feature of this method that distinguishes it from other image segmentation methods such as K-Means and MeanShift [29, 30, 31].

The Grabcut algorithm turned out to be the best algorithm for image processing. The processing result was the closest to the expected one. Although the algorithm has some errors. Despite the fact that the area that was contoured turned out to be the clearest and most complete in comparison with other algorithms, this area also includes areas of the image that do not belong to the objects under study.

In some cases, the segmentation will not work well, for example, it may notice the scenes incorrectly. In the developed application, handling of such cases is not provided. In other implementations, the user needs to perform minor retouching. It is necessary to paint some strokes on images where there are some erroneous results. Then the next iteration will produce better results [32, 33, 34].

Anything outside the rectangle the user enters will be considered a safe background (i.e., a background that is guaranteed to be removed). Everything else that is inside the rectangle is unknown. Then the algorithm works. Initial marking is carried out, the pixels of the foreground and background are noticed. The Gaussian mixture model (GMM) is designed to simulate and evaluate the truth of objects. GMM examines and creates a new pixel section according to the input data. This determines whether the newly created pixel affects the foreground or the background. This is achieved through the ratio of colors. Based on the distribution of points, a graph is built (nodes on the graph are pixels).

The probability of the ratio of a pixel to the foreground or background is determined on the basis of the weights of the boundaries that connect the pixels. Thus, with a large difference in color between the pixels, they will have a small boundary weighting coefficient. The Mincut algorithm is designed to divide the graph into two nodes, in particular, the source and destination with a minimum cost function. As a result of the algorithm, all pixels connected to the source node come to the foreground, and the pixels connected to the destination node become the background. It should be borne in mind that the accuracy of processing objects in the foreground can vary from many parameters, as well as from the area that the user enters independently [30, 31]. The result of image processing by the Grabcut algorithm is shown in Fig. 5a.
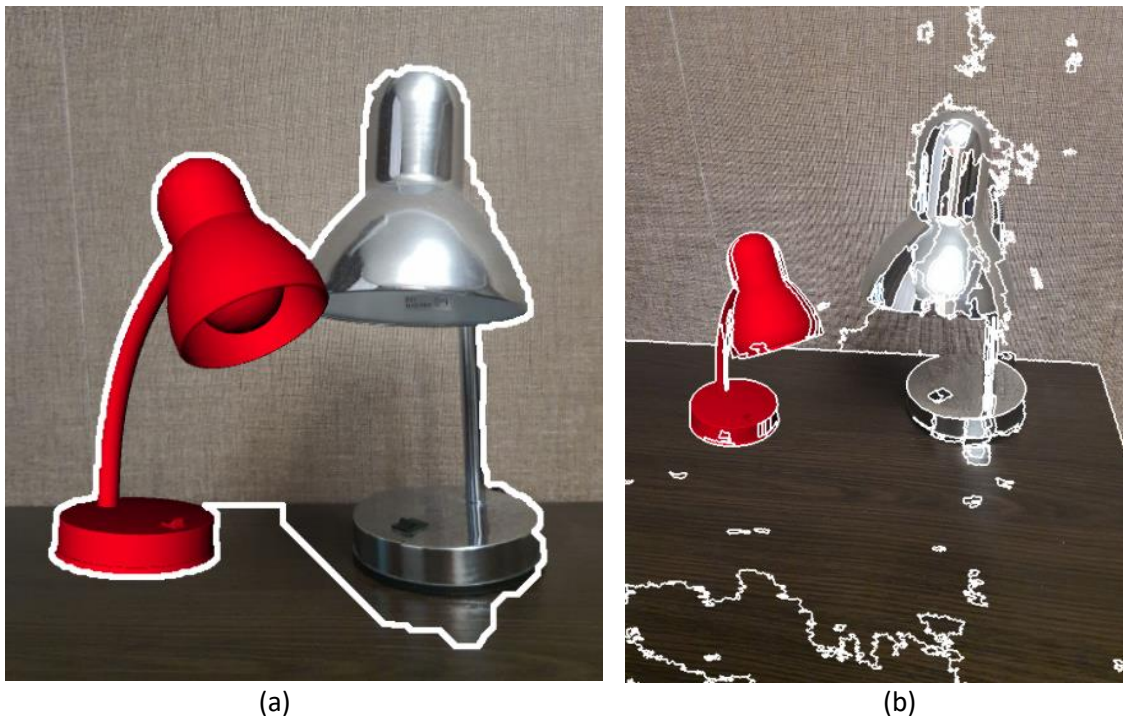

(a)   (b)

Figure 5: The results of object segmentation by: (a) the Grabcut algorithm, (b) the MeanShift algorithm

MeanShift is positioned as one of the clustering algorithms that iteratively group points into specific clusters, constantly finding the mode and shifting the points to it. This algorithm is used to increase the contrast of objects in images that have clear boundaries, identify and remove noise or highlighted areas, segmentation of brightly colored elements, detection of moving objects in video [29, 35, 38, 39]. However, computations are computationally expensive. An example of image segmentation using the MeanShift algorithm is shown in Fig. 5b. Among the advantages of the MeanShift algorithm is that the algorithm does not take any preliminary form (for example, elliptical) of data clusters and can process arbitrary feature spaces. The weaknesses of the algorithm include the need to use an adaptive window size, since an incorrect window size can lead to cluster merging and, as a result, to poor clustering. You can see areas in the resulting image that do not belong to the objects under study, but they are also selected. Thus, the algorithm selects not only objects, but also areas that may be similar to objects, but are not.

These algorithms were used to segment images on a mobile device. By improving these algorithms, you can achieve good image processing speed on portable devices. By using an app for mobile devices that segments images in real time, technology can be more involved in the learning process of schoolchildren and students by creating special apps for these groups. For example, an application that will segment the image from the camera of a mobile device to recognize the shape of objects, and later on the classes of objects themselves, and to report data about such objects to the user. Medical students can use this version of the application to identify certain forms of skin diseases that are clearly manifested on the body of the sick.

5. Conclusions

As a result of the study of object segmentation algorithms, an AR application for mobile devices was created using the OpenCV library for image processing. The following image processing algorithms were tested: K-Means, MeanShift and GrabCut. The best result was obtained using the GrabCut algorithm. As you can see, algorithms do not always select not only specific objects in the image, but also those areas of the image that differ from the background in certain characteristics, such as color, brightness, and so on.

The OpenCV library uses scripts written in C++. For such cases in the Unity 3D environment there is a support of plug-ins that allows to use in the code functions from the given libraries.

Analyzing the obtained results, we can conclude that for image processing on mobile devices it is necessary to use algorithms that do not require large resources and can quickly display the result on the screen.

The use of image processing algorithms in augmented reality mobile applications makes it possible to create virtual scenes with great detail and better interaction between real objects and virtual ones. The Unity 3D development environment is a simple and affordable way to develop not only gaming applications, but also applications that can be used in the educational field of human activity.

6. References

[1] R.M. Thanki, A.M. Kothari, Image Segmentation. In: Digital Image Processing using SCILAB, Springer, Cham, 2019. doi:10.1007/978-3-319-89533-8_6.

[2] K.K. Tseng, R. Zhang, C.M. Chen, et al, DNetUnet: a semi-supervised CNN of medical image segmentation for super-computing AI service, J Supercomput 77, 3594–3615, 2021. doi:10.1007/s11227-020-03407-7.

[3] F.Y. Shih, Image Segmentation. In: Liu L., Özsu M.T. (eds) Encyclopedia of Database Systems, Springer, New York, 2018. doi:10.1007/978-1-4614-8265-9_1011.

[4] M. Li, D. Chen, S. Liu, et al, Online learning method based on support vector machine for metallographic image segmentation, SIViP 15, 571–578, 2021. doi:10.1007/s11760-020-01778-1.

[5] A. Jindal, S. Joshi, R. Jangwal, A. Rathi, R. Jain, Image Segmentation. In: Goyal D., Chaturvedi P., Nagar A.K., Purohit S. (eds) Proceedings of ICSEC. Springer, Singapore, 2021 doi:10.1007/978-981-15-6707-0_23.

[6] F.Y. Shih, Image Segmentation. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA, 2009. doi:10.1007/978-0-387-39940-9_1011.

[7] L. Caponetti, G. Castellano, Image Segmentation. In: Fuzzy Logic for Image Processing. SpringerBriefs in Electrical and Computer Engineering. Springer, Cham, 2017. doi:10.1007/978-3-319-44130-6_7.

[8] Y. Ma, K. Zhan, Z. Wang, Image Segmentation. In: Applications of Pulse-Coupled Neural Networks. Springer, Berlin, Heidelberg, 2010. doi:10.1007/978-3-642-13745-7_3.

[9] A. Distante, C. Distante, Image Segmentation. In: Handbook of Image Processing and Computer Vision. Springer, Cham, 2020. doi:10.1007/978-3-030-42374-2_5.

[10] Medical Image Segmentation. In: Furht B. (eds) Encyclopedia of Multimedia. Springer, Boston, MA, 2008. doi:10.1007/978-0-387-78414-4_108.

[11] S. Biswas, B.C. Lovell, Image Segmentation. In: Biswas S., Lovell B.C. (eds) Bézier and Splines in Image Processing and Machine Vision. Springer, London, 2008. doi:10.1007/978-1-84628-957-6_2.

[12] F.Y. Shih, Image Segmentation. In: Liu L., Özsu M. (eds) Encyclopedia of Database Systems. Springer, New York, 2016. doi:10.1007/978-1-4899-7993-3_1011-2.

[13] R. Vidal, Y. Ma, S.S. Sastry, Image Segmentation. In: Generalized Principal Component Analysis. Interdisciplinary Applied Mathematics, vol 40. Springer, New York, 2016. doi:10.1007/978-0-387-87811-9_10.

[14] N. Shusharina, M. P. Heinrich, R. Huang, Segmentation, Classification, and Registration of Multi-modality Medical Imaging Data, MICCAI 2020 Challenges, Lima, Peru, October 4–8, Springer, Cham, 2020. https://doi.org/10.1007/978-3-030-71827-5

[15] R. Klette, Image Segmentation. In: Concise Computer Vision. Undergraduate Topics in Computer Science. Springer, London, 2014. doi:10.1007/978-1-4471-6320-6_5.

[16] S. Chabrier, C. Rosenberger, B. Emile, et al, Optimization-Based Image Segmentation by Genetic Algorithms. J Image Video Proc, 842029, 2008. doi:10.1155/2008/842029.

[17] S. Basu, Selecting the optimal image segmentation strategy in the era of multitracer multimodality imaging: a critical step for image-guided radiation therapy. Eur J Nucl Med Mol Imaging 36, 180–181, 2009. doi:10.1007/s00259-008-1033-5.

[18] A. Campilho, F. Karray, Z. Wang, Image Analysis and Recognition, 17th International Conference, ICIAR 2020, Póvoa de Varzim, Portugal, June 24–26, Springer, Cham, 2020. doi:10.1007/978-3-030-50516-5

[19] A. Morales-González, E. García-Reyes, L.E. Sucar, Improving Image Segmentation for Boosting Image Annotation with Irregular Pyramids. In: Ruiz-Shulcloper J., Sanniti di Baja G. (eds) Congress CIARP 2013. Lecture Notes in Computer Science, vol 8258. Springer, Berlin, Heidelberg, 2013. doi:10.1007/978-3-642-41822-8_50.

[20] K. Suresh, P. Srinivasa rao, Various Image Segmentation Algorithms: A Survey. In: Satapathy S., Bhateja V., Das S. (eds) Smart Intelligent Computing and Applications. Smart Innovation, Systems and Technologies, vol 105. Springer, Singapore, 2019. doi:10.1007/978-981-13-1927-3_24.

[21] D. Oliva, M. Abd. Elaziz, S. Hinojosa, Metaheuristic Algorithms for Image Segmentation: Theory and Applications, Springer, Cham, 2019. doi:10.1007/978-3-030-12931-6.

[22] G. Windisch, M. Kozlovszky, Framework for Comparison and Evaluation of Image Segmentation Algorithms for Medical Imaging. In: Braidot A., Hadad A. (eds) VI Latin American Congress on Biomedical Engineering CLAIB, vol 49. Springer, Cham, 2015. doi:10.1007/978-3-319-13117-7_123.

[23] C. Jun, S. Liping, Z. Dongyan, et al, Application study of image segmentation methods on pattern recognition in the course of wood across-compression. Journal of Forestry Research 11, 57–59, 2000. doi:10.1007/BF02855499.

[24] N. Ikonomakis, K.N. Plataniotis, A.N. Venetsanopoulos, Color Image Segmentation for Multimedia Applications. In: Tzafestas S.G. (eds) Advances in Intelligent Systems. International Series on Microprocessor-Based and Intelligent Systems Engineering, vol 21. Springer, Dordrecht, 1999. doi:10.1007/978-94-011-4840-5_26.

[25] K. Ohkura, H. Nishizawa, T. Obi, et al, Unsupervised Image Segmentation Using Hierarchical Clustering. OPT REV 7, 193–198, 2000. doi:10.1007/s10043-000-0193-8.

[26] B. Roy, R.K. Chatterjee, Historical Handwritten Document Image Segmentation Using Morphology. In: Sengupta S., Das K., Khan G. (eds) Emerging Trends in Computing and Communication. Lecture Notes in Electrical Engineering, vol 298. Springer, New Delhi, 2014. doi:10.1007/978-81-322-1817-3_14.

[27] D. Murashov, Application of Information Redundancy Measure to Image Segmentation. In: Strijov V., Ignatov D., Vorontsov K. (eds) Intelligent Data Processing. Communications in Computer and Information Science, vol 794. Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-35400-8_9.

[28] I. Sova, I. Sidenko, Y. Kondratenko, Machine learning technology for neoplasm segmentation on brain MRI scans, CEUR Workshop Proceedings, ICTERI-PhD 2020, volume 2791, pp. 50-59, 2020.

[29] K. Ivanova, G. Kondratenko, I. Sidenko, Y. Kondratenko, Artificial intelligence in automated system for web-interfaces visual testing, CEUR Workshop Proceedings, COLINS 2020, volume 2604, pp. 1019-1031, 2020.

[30] V. Zinchenko, G. Kondratenko, I. Sidenko, Y. Kondratenko, Computer Vision in Control and Optimization of Road Traffic, IEEE Third International Conference, DSMP, Lviv, Ukraine, pp. 249-254, 2020, doi: 10.1109/DSMP47368.2020.9204329.

[31] M. Benyoussef, N. Idrissi, D. Aboutajdine, A Distributed Approach to Color Image Segmentation. In: Choraś R.S. (eds) Image Processing and Communications Challenges 3. Advances in Intelligent and Soft Computing, vol 102. Springer, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-23154-4_10.

[32] V. Lytvyn, A. Gozhyj, I. Kalinina, V. Vysotska, V. Shatskykh, L. Chyrun, Y. Borzov, An intelligent system of the content relevance at the example of films according to user needs, CEUR Workshop Proceedings, ICT & ES, volume 2516, pp. 1-23, 2019.

[33] Q. Lou, J. Peng, F. Wu, D. Kong, Variational Model for Image Segmentation. In: Bebis G. et al. (eds) Advances in Visual Computing. ISVC 2013. Lecture Notes in Computer Science, vol 8034. Springer, Berlin, Heidelberg, 2013. doi:10.1007/978-3-642-41939-3_64.

[34] J. Shotton, P. Kohli, Semantic Image Segmentation. In: Ikeuchi K. (eds) Computer Vision. Springer, Boston, MA, 2014. https://doi.org/10.1007/978-0-387-31439-6_251.

[35] D. Mikhov, Y. Kondratenko, G. Kondratenko, I. Sidenko, Fuzzy Logic Approach to Improving the Digital Images Contrast, IEEE 2nd Ukraine Conference, UKRCON, Lviv, Ukraine, pp. 1183-1188, 2019, doi: 10.1109/UKRCON.2019.8879961.

[36] Y. Pomanysochka, Y. Kondratenko, I. Sidenko, Noise filtration in the digital images using fuzzy sets and fuzzy logic, CEUR Workshop Proceedings, ICTERI 2019: PhD Symposium, volume 2403, pp. 63-72, 2019.

[37] I. Sidenko, K. Filina, G. Kondratenko, D. Chabanovskyi, Y. Kondratenko, Eye-tracking technology for the analysis of dynamic data, IEEE 9th International Conference, DESSERT, Kyiv, UKraine, pp. 479-484, 2018, doi: 10.1109/DESSERT.2018.8409181.

[38] M. Ramadas, A. Abraham. Metaheuristics for Data Clustering and Image Segmentation, Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-04097-0.

[39] S. Bhattacharyya, P. Dutta, S. De, G. Klepac. Hybrid Soft Computing for Image Segmentation, Springer, Cham, 2016. https://doi.org/10.1007/978-3-319-47223-2