

NEW IMAGE DESCRIPTOR FROM EDGE DETECTOR AND BLOB EXTRACTOR

Rafał Grycuk, Rafał Scherer, Marcin Gabryel

*Institute of Computational Intelligence, Czestochowa University of Technology
Czestochowa, Poland
rafal.grycuk@iisi.pcz.pl, rafal.scherer@iisi.pcz.pl, marcin.gabryel@iisi.pcz.pl*

Abstract. In this paper we present a novel approach for image description. The method is based on two well-known algorithms: edge detection and blob extraction. In the edge detection step we use the Canny detector. Our method provides a mathematical description of each object in the input image. On the output of the presented algorithm we obtain a histogram, which can be used in various fields of computer vision. In this paper we applied it in the content-based image retrieval system. The simulations proved the effectiveness of our method.

Keywords: *image description, content-based image retrieval, CBIR, edge detection, blob extraction, blob detection*

1. Introduction

Along with the development of the Internet and a possibility of capturing images, a new area of expertise was created. Accurate image content description [1, 2] is one of the greatest challenges of computer science. Effective browsing or retrieving images is used in various fields of life, e.g. medicine, crime prevention, face recognition, robotics, military and many others. Precise mathematical description of visual objects is difficult because we need to overwhelm the semantic gap between human recognition and recognition performed by computers. Image feature extraction can be based on various attributes, i.e. color [3], texture [4], shape [5, 6], local features [7], multiresolution wavelet analysis [8-11] and their improvements [12, 13]. It seems to be interesting to develop a novel method for object description. Such mathematical features can be used in further processing of various systems. The following subsections briefly describe the most important algorithms used in our method.

1.1. Edge detection

In the literature we can find many methods for edge detection. The Canny edge detector [14-16] is one of the most commonly used image processing methods

for detecting edges. It takes as input a gray scale image, and produces as output an image showing the positions of tracked intensity discontinuities. The algorithm runs in four separate steps [17]:

1. Noise reduction. The image is smoothed by applying an appropriate Gaussian filter
2. Finding the intensity gradient of the image. During this step the edges should be marked where gradients of the image have large magnitudes
3. Non-maxima suppression. If the gradient magnitude at a pixel is larger than those at its two neighbors in the gradient direction, mark the pixel as an edge. Otherwise, mark the pixel as the background
4. Edge tracking by hysteresis. Final edges are determined by suppressing all edges that are not connected to genuine edges.

The effect of the Canny operator is determined by parameters:

- The width of the Gaussian filter used in the first stage directly affects the results of the Canny algorithm
- The thresholds used during edge tracking by hysteresis. It is difficult to provide a generic threshold that works well on all images.

The Canny detector basically finds edges where the pixel intensity changes (image gradient). Before edge detection the non-important edges need to be removed. Thus the Gaussian smoothing method is applied. To approximate gradient for both directions (x, y) the following formula was used [17]:

$$K_{Gx} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad (1)$$

$$K_{Gy} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (2)$$

The edge strengths can be calculated by the Euclidean distance measure (3) or Manhattan distance measure (4) [17]:

$$|G| = \sqrt{G_x^2 + G_y^2}, \quad (3)$$

$$|G| = |G_x| + |G_y|, \quad (4)$$

where G_x is gradient in horizontal direction, G_y is gradient in vertical direction. The edge direction can be described by the following formula [17]:

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right). \quad (5)$$



Fig. 1. Edge detection

Figure 1A shows an input image and Figure 1B represents the edge detected image. As can be seen, the edges were detected correctly, because the image gradient is low.

1.2. Blob detection

Blob detection is one of the basic methods of image processing. It allows to detect a list of blobs (objects) in the image. Unfortunately, obtaining homogeneous objects from an image as a list of pixels is a challenging task. Especially, when we deal with a heterogeneous background, i.e. the objects containing multicolored background. There are many methods for extracting objects (blobs) from images [18-22]. In this paper we use methods implemented in the AForge.NET library. These algorithms are described by Andrew Kirillov [23] and exist in four variants: Convex full, Left/Right Edges, Top/Bottom Edges, Quadrilateral, shown in Figure 2.

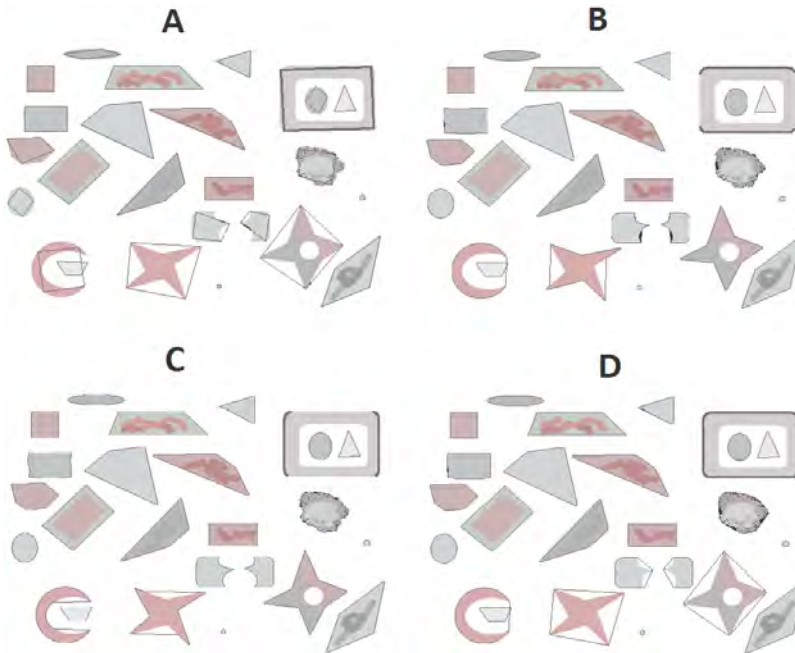


Fig. 2. Comparison of methods for blob detection used in the AForge.NET library [23]

Figure 2A illustrates Quadrilateral detection method. As can be seen, round edges of the objects are not detected correctly. Much better results are obtained by the Top/Bottom Edges algorithm (Fig. 2C). Edges of objects are detected mostly correctly, with individual exceptions. The Left/Right Edges method behaves similarly (Fig. 2B). The last method has a problem with the detection of vertices inside figures, e.g. star-shaped objects (Fig. 2D).

2. Proposed method for image description

In this section we present the novel method for image description. Our approach is based on two well-known algorithms: edge detection and blob extraction. The first step is to perform the edge detection method. This step is crucial because the correctness of detected edges will have effect on the blob extraction step. Thus, the selection on two input parameters of Canny detector is a key stage. In our experiments we select these parameters empirically for each image class (dinosaurs, car cards, mountains, etc.). The second stage of our method is blob detection and blob extraction. Both of them are tightly connected. The first one allows to locate the blob (object), and the second one performs the object extraction. We use the Quadrilateral algorithm for this step. It allows to determine blob properties such as: ROI (Region of Interest), center of gravity, edge points. The last property is extremely important in the next step of our method. From four edge points extracted by the Quadrilateral we calculate the following distances: Top, Bottom, Left, Right. For example, the top is distance between two most distant points in top image direction. The remaining distances are calculated in the same way. Last stage of our algorithm creates the histogram for each based on calculated distances. Histograms are often used in Content-Based Image Retrieval (CBIR) for comparing images (e.g. with Euclidean or Manhattan distance), thus such histograms can be used as feature extraction stage of CBIR. These steps can be described by the following pseudo-code:

INPUT: Input image

OUTPUT: Histograms

1. `edgeDetectedImage := DetectEdgesByCanny(thresh, threshlinking);`
2. `DetectBlobs(edgeDetectedImage);`
3. `blobs := ExtractBlobs(edgeDetectedImage);`
4. **ForEach** $blob_i \in blobs$ **do**
 - {
 - `topi := CalculateDistance(blobi.QuadrilateralPoints, topi);`
 - `bottomi := CalculateDistance(blobi.QuadrilateralPoints, bottomi);`
 - `lefti := CalculateDistance(blobi.QuadrilateralPoints, lefti);`
 - `righti := CalculateDistance(blobi.QuadrilateralPoints, righti);`
 - `histogrami := CreateHistogram(topi, bottomi, lefti, righti);`
 - }

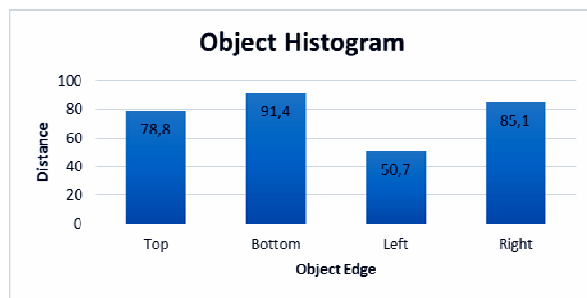


Fig. 3. Object histogram which provides a mathematical description of a visual object

In Figure 3 we can distinguish four bins which describe four edges of the detected object (blob). Such mathematical description allows to compare content of visual objects (and images) by measuring distance between them. The horizontal axis contains the edges labeled by direction (top, bottom, left, right). The vertical axis provide the calculated distance of the edge.

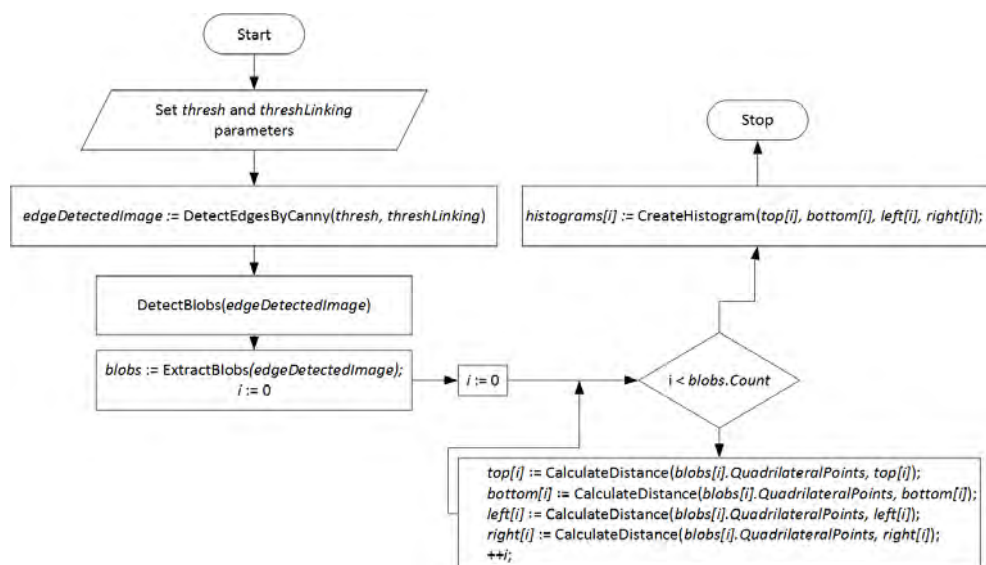


Fig. 4. Block diagram of the proposed method

Figure 4 shows a block diagram of the proposed method. The output histograms is a mathematical description of the input image, thus our method can be used as a preprocessing stage for a more complex image retrieval system.

3. Experimental results

In this section we present the results of the experiments. The simulation environment were written by the authors in .NET Framework with AForge.NET library

and C# as programming language. Test images for simulations were taken from the Corel database. We selected images from various classes and divided each class in two sets. First set (90% of image class) is used for indexing and second one (10%) serves as query images.

To evaluate performance of our method we based our experiments on two established measures: Precision and Recall. In order to calculate these measures we need to use the following sets of images (see Fig. 5):

- AI - set of appropriate images,
- RI - set of returned images,
- Rai - properly returned images,
- Iri - improperly returned images,
- Anr - proper not returned,
- Inr - improper not returned images.

Precision and recall can be calculated by the following formulas:

$$precision = \frac{|rai|}{|rai + iri|} \quad (6)$$

$$recall = \frac{|rai|}{|rai + anr|} \quad (7)$$

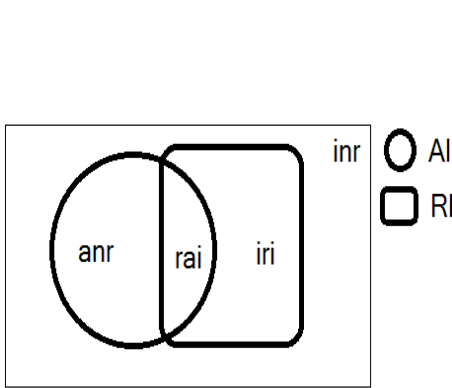


Fig. 5. Performance measures diagram



Fig. 6. Query results

During our experiments we obtained the following results, presented in Table 1. The measures were normalized and presented as percentage values. As can be seen, the algorithm proved to be effective and the *rai* factor is high and the *anr* significantly low.

The presented method seems to be effective in image retrieval purposes. We have implemented the results evaluation system and we also examine our result empirically. Figure 6 shows the result of a single query image. Image with border is the query image. Our method was compared with multiresolution wavelet analysis and provided better results in image retrieval. Methods based on wavelet analysis

(such as SURF, see [24, 25], or Haar wavelets) provide image description based on keypoints. Such description is effective when we need to compare images and find corresponding keypoints on both images (determine similarity). Nevertheless, when facing the image retrieval problem, we need to describe object, not entire image (object + background). Unfortunately, the keypoints are often located in the background, hence provide irrelevant data in the indexation phase.

Table 1

Simulation results for multi query [%]

Image Id	AI	<i>rai</i>	<i>iri</i>	<i>anr</i>	Precision	Recall
1 (1).jpg	33	21	12	7	13	7
1 (2).jpg	33	31	3	18	3	19
1 (20).jpg	33	21	12	7	13	7
1 (21).jpg	33	22	12	8	14	7
2 (1).jpg	33	27	7	7	7	8
2 (10).jpg	33	28	6	6	6	7
2 (11).jpg	33	24	9	9	10	10
2 (17).jpg	33	28	6	6	6	7
3 (1).jpg	33	23	11	12	11	12
3 (10).jpg	33	23	10	11	11	12
3 (11).jpg	33	27	7	11	7	12
3 (15).jpg	33	29	4	17	4	17

4. Conclusions

The presented algorithm is a contribution to image description and content-based image retrieval. The experiments proved the effectiveness of our method. The algorithm requires two input parameters, which describe the intensity of the object edges. The method is based on four main steps: edge detection, blob extraction, factors calculation and histogram creation. The presented approach can be used for content-based image retrieval tasks and image description generally. For the purposes of this paper we applied this method in a CBIR system with satisfactory results. Our approach allowed to describe each object separately instead of the entire image thus, the resulting precision increased significantly.

Presented method was compared with wavelet analysis and provide better results. Multiresolution wavelet analysis can be added (in the future work) to our method after the object extraction stage is completed, because keypoints will be detected only on the extracted object. Such approach will provide two histograms (two comparing attributes), which would significantly improve the results.

Acknowledgment

The work presented in this paper was supported by a grant BS/MN-1-109-301/14/P "Clustering algorithms for data stream - in reference to the Content-Based Image Retrieval methods (CBIR)".

References

- [1] Bazarganigilani M., Optimized image feature selection using pairwise classifiers, *Journal of Artificial Intelligence and Soft Computing Research* 2011, 1, 147-153.
- [2] Drozda P., Sopyła K., Górecki P., Online crowdsourcing system supporting ground truth datasets creation, *Computing 12th Conference, ICAISC2013, Zakopane, June 9-13, 2013*, eds. L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, J.M. Zurada, 532-539.
- [3] Meskaldji K., Boucherkha S., Chikhi S., Color quantization and its impact on color histogram based image retrieval accuracy, *Networked Digital Technologies 2009*, 515-517.
- [4] Huang J., Kumar S.R., Mitra M., Zhu W.J., Zabih R., Image indexing using color correlograms, *Computer Vision and Pattern Recognition 1997*, 762-768.
- [5] Śmietański J., Tadeusiewicz R., Łuczyńska E., Texture analysis in perfusion images of prostate cancer - A case study, *International Journal of Applied Mathematics and Computer Science* 2010, 20(1), 149-156.
- [6] Veltkamp R.C., Hagedoorn M., State of the art in shape matching, [in:] *Principles of Visual Information Retrieval*, ed. M.S. Lew, Springer, London - Berlin - Heidelberg 2001, 87-119.
- [7] Zalaśiński M., Cpałka K., New approach for the on-line signature verification based on method of horizontal partitioning, *Artificial Intelligence and Soft Computing* 2013, 342-350.
- [8] Grossmann A., Wavelet transforms and edge detection, *Stochastic Processes in Physics and Engineering, Series Mathematics and Its Applications* 1988, 42, 149-157.
- [9] Mallat S., Wen Liang Hwang, Singularity detection and processing with wavelets, *IEEE Transactions on Information Theory* 1992, 38(2), 617-643.
- [10] Krim H., Tucker D., Mallat S., Donoho D., On denoising and best signal representation, *IEEE Transactions on Information Theory* 1999, 45(7), 2225-2238.
- [11] Mallat S.G., A theory for multiresolution signal decomposition: the wavelet representation, *Pattern Analysis and Machine Intelligence, IEEE Transactions* 1989, 11, 7, 674-693.
- [12] Bay H., Tuytelaars T., Van Gool L., Surf: Speeded up robust features, *Computer Vision-ECCV 2006*, 404-417.
- [13] Evans C., Notes on the OpenSURF Library, University of Bristol, Tech. Rep., 2009.
- [14] Canny J., A computational approach to edge detection, *Pattern Analysis and Machine Intelligence, IEEE Transactions* 1986, 8(6), 679-698.
- [15] Bao P., Zhang D., Wu X., Canny edge detection enhancement by scale multiplication, *Pattern Analysis and Machine Intelligence* 2005, 27(9), 1485-1490.
- [16] Wang B., Fan S., An improved Canny edge detection algorithm, *Computer Science and Engineering* 2009, 1, 497-500.
- [17] Luo Y., Duraiswami R., Canny edge detection on NVIDIA CUDA, *Computer Vision and Pattern Recognition Workshops* 2008, 1-8.
- [18] Grycuk R., Gabryel M., Korytkowski M., Scherer R., Voloshynovskiy S., From single image to list of objects based on edge and blob detection, *Artificial Intelligence and Soft Computing* 2014, 8468, 605-615.
- [19] Damiand G., Resch P., Split-and-merge algorithms defined on topological maps for 3D image segmentation, *Graphical Models* 2003, 65(1), 149-167.

-
- [20] Grycuk R., Gabryel M., Korytkowski M., Romanowski J., Scherer R., Improved digital image segmentation based on stereo vision and mean shift algorithm, *Parallel Processing and Applied Mathematics* 2014, 8384, 433-443.
 - [21] Grycuk R., Gabryel M., Korytkowski M., Scherer R. Content-based image indexing by data clustering and inverse document frequency, *Beyond Databases, Architectures, and Structures* 2014, 424, 374-383.
 - [22] Liu Y., Zhang D., Lu G., Ma W.Y., A survey of content-based image retrieval with high-level semantics, *Pattern Recognition* 2007, 40(1), 262-282.
 - [23] Kirillov A., Detecting some simple shapes in images, *AForge .NET*, 2010.
 - [24] Bay H., Speeded-up robust features (SURF), *Computer Vision and Image Understanding* 2008, 110, 3, 346-359.
 - [25] Terriberry T., French L., Helmsen J., GPU accelerating speeded-up robust features, *Proceedings of 3DPVT* 2008, 355-362.