



## Automatic Color Edge Detection with Similarity Transformation

M. Ozan INCETAS<sup>1,\*</sup> , Recep DEMIRCI<sup>2</sup> , H. Guclu YAVUZCAN<sup>3</sup> 

<sup>1</sup>Alanya Alaaddin Keykubat University, Department of Electric and Energy, 07450, Alanya, Turkey

<sup>2</sup>Gazi University, Department of Computer Engineering, 06500, Ankara, Turkey

<sup>3</sup>Gazi University, Department of Industrial Design, 06570, Ankara, Turkey

### Article Info

Received: 06/06/2018

Accepted: 24/12/2018

### Keywords

Edge detection  
Similarity Transform  
Thresholding

### Abstract

Edge detection is an important step in image processing. As edge is intensity variation with spatial coordinates, the similarities between neighboring pixels could be used for edge detection. It has been observed that the effective results could be attained by thresholding the homogeneity images generated by means of the similarity transformation. Nevertheless, the user-defined normalization coefficient in similarity transform stage seriously effects edge detection performance and it needs to be automatically selected for every particular image. In this study, a new approach in which the normalization coefficient is automatically determined has been presented. The automating process of the similarity transform has been performed according to the gray level values of the neighboring pixels. The gray level differences of the central pixel and other neighboring pixels have been used to determine the similarity coefficient. Subsequently, the binarization process of the homogeneity images obtained with proposed algorithm have been completed with different thresholding techniques. Additionally, the F-score of the proposed edge detection has been obtained with 200 images in the BSDS training dataset. The achieved F-score values have showed that the performance of automatic approach is quite high.

## 1. INTRODUCTION

Edge detection is one of the most important steps in image processing. It is used in many processes from object recognition to image retrieval systems. The edge could be generally expressed as the sudden change between two neighboring pixels. Therefore, initially it is necessary to use a derivative or similar technique and then the thresholding must be employed for the edge detection on a gray-level images [1,2]. Nevertheless, determination of the thresholds is one of the most significant steps and many procedures have been established. On the other hand, edge detection on color images is a bit more complicated than the gray level images as there are three different channels. In 2007, a similarity-based approach was developed by Demirci in order to reduce the complexity of edge detection in color images [3]. Subsequently, one-dimensional similarity images were obtained by using three-dimensional color images. The developed technique has been used for edge detection or segmentation by properly selecting various parameters. For example, a new segmentation technique was developed by using pixel similarity in 2008 [4] and a similarity-based segmentation approach was developed in 2014 [5]. Both studies are based on similarity images obtained with different normalization coefficients. The fast edge detection algorithm was also developed by using similarity matrix [6] and different normalization coefficients has been tested in 2017 [7]. The same similarity technique was used in neonatal jaundice detection of newborn infants [8]. The values obtained from the similarity image were also used for image enhancement. In 2016, a new diffusion filter using similarity values was introduced [9].

In all of the studies mentioned above, the user determined or fixed similarity coefficient was used in the calculation of similarity value or similarity transformation. However, the determination of the optimal coefficient value for each image can be time consuming. In addition, the selection coefficient value makes the existence of a user mandatory. In this study, a new similarity based edge detection approach in which

\*Corresponding author, e-mail: ozan.incetas@alanya.edu.tr

the normalization coefficient is automatically selected and user intervention is completely eliminated has been presented to eliminate all these requirements. Automatically obtained similarity images are thresholded with the center of gravity of histogram (CoG), Otsu and Kapur techniques, and consequently edge pixels in image are detected. The similarity images obtained with various constant normalization coefficients have been compared with the suggested automatic method. In order to measure the success of the proposed technique, F-scores have been obtained by using the edge information given in BSDS database.

## 2. IMAGE THRESHOLDING

The use of thresholding techniques to obtain binary images is quite common in edge detection. Otsu, Kapur and CoG algorithms are widely used in image processing area [10-12]. The Otsu approach is based on the finding of the threshold value that maximizes the intra-class variance and minimizes the inter-class variance. The threshold value that maximizes the sum of local entropy at each gray level is determined in Kapur method. The weighted average of the histogram is calculated with the CoG method. The objective function proposed by Kapur is defined as

$$J(t) = H_0 + H_1 \quad (1)$$

where  $H_0$  and  $H_1$  represent the entropies of the classes. The entropies of the classes are calculated as follows:

$$H_0 = -\sum_{i=0}^{t-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0} \quad (2)$$

$$H_1 = -\sum_{i=t}^{L-1} \frac{p_i}{\omega_1} \ln \frac{p_i}{\omega_1} \quad (3)$$

where the partial probabilities of the classes are given by

$$\omega_0 = \sum_{i=0}^{t-1} p_i \quad \text{and} \quad \omega_1 = \sum_{i=t}^{L-1} p_i \quad (4)$$

The probability of the  $i^{\text{th}}$  gray level is computed as

$$p_i = \frac{h_i}{M N} \quad (5)$$

The objective function of the Otsu approach is defined as

$$J(t) = \sigma_0 + \sigma_1 \quad (6)$$

where the  $\sigma_0$  and  $\sigma_1$  represent the variances of two different classes. They are expressed as

$$\sigma_0 = \omega_0 (\mu_0 - \mu_T)^2 \quad \text{and} \quad \sigma_1 = \omega_1 (\mu_1 - \mu_T)^2 \quad (7)$$

where  $\mu_T$  is the mean of the histogram which is calculated as

$$\mu_T = \sum_{i=0}^{L-1} i p_i \quad (8)$$

The means of the classes are given by

$$\mu_0 = \sum_{i=0}^{L-1} \frac{i P_i}{\omega_0} \text{ and } \mu_1 = \sum_{i=1}^{L-1} \frac{i P_i}{\omega_1} \quad (9)$$

The mathematical expression of the CoG method is given

$$T = \frac{1}{M N} \sum_{i=0}^L i \cdot z_i \quad (10)$$

where  $i$  represents the gray level,  $z_i$  represents the number of the pixels in gray level  $i$ ,  $M$  and  $N$  represent the width and height of image, respectively.  $L$  represents the maximum gray level value which is 255.

### 3. AUTOMATIC SIMILARITY TRANSFORMATION

Edge detection based on color similarity of pixels in image was first defined by Demirci [3]. The most significant advantage of Demirci's approach is that it does not require the complex processes such as gradient, Laplace or statistical calculations. The similarity transform is based on the similarity between two pixels in a color image. A central pixel and its neighbors are shown in Figure 1. In similarity transform algorithm, initially, the similarities between central pixel  $P_0$  and its neighbors are calculated as follows:

$$S(P_0, P_i) = \exp\left(-\frac{\|P_0 - P_i\|}{D_n}\right) \quad (11)$$

$$\|P_0 - P_i\| = \frac{1}{\sqrt{3}} (\Delta R^2 + \Delta G^2 + \Delta B^2)^{1/2} \quad (12)$$

$$\Delta R = |R_0 - R_i|$$

$$\Delta G = |G_0 - G_i|$$

$$\Delta B = |B_0 - B_i|$$

(13)

where  $\|P_0 - P_i\|$  is the normalized Euclidean distance in color space and  $D_n$  is the normalization coefficient which is determined by the user. The average similarity of mask is substituted in output image as

$$S_{P_0} = \frac{1}{9} \sum_{i=0}^8 S(P_0, P_i) \quad (14)$$

and

$$S(P_0, P_i) = 255 \cdot \exp\left(-\frac{\|P_0 - P_i\|}{D_n}\right) \quad (15)$$

$P_1$	$P_2$	$P_3$
$P_8$	$P_0$	$P_4$
$P_7$	$P_6$	$P_5$

**Figure 1.** 3x3 neighborhoods of a  $P_0$  pixel

Nevertheless, as the normalization coefficient is user dependent, the edge detection performance will be depending on user's selections as well as Euclidean color distance in RGB space. Typical values for the normalization coefficient were suggested as 1, 16, 32, 64, 128, 196, and 256 in original study. Variation of similarity with as the normalization coefficient is shown in Figure 2. As could be seen, although the lower values are sensitive small variations, the high intensity changes could not be detected. So an approach which is independent from user and image types must be developed.

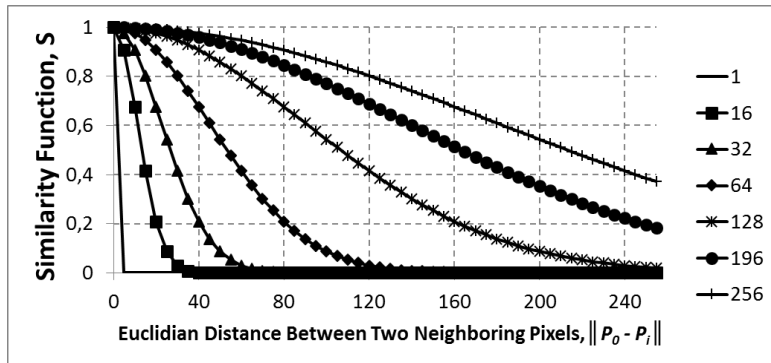


Figure 2. The similarity function,  $S$

In this study, a new approach in which the  $D_n$  normalization coefficient automatically is determined [13], has been proposed as follows:

$$D_n = \left( \frac{255}{d_a^2 + 1} \right) + 1 \quad (16)$$

where  $d_a$  is the average of the normalized Euclidean distances between the center pixel and the neighbors in the  $3 \times 3$  mask. It is calculated as

$$d_a = \frac{1}{9} \sum_{i=0}^8 d_{0,i} \quad (i=0, 1, \dots, 8) \quad (17)$$

where  $d_{0,i} = ||P_0 - P_i||$  is the normalized Euclidean distances between the center pixel,  $P_0$  and its neighbor,  $P_i$ . Consequently, the similarity transformation is automatically completed without any user intervention. Furthermore, the similarity transformation is locally performed as every unique pixel will have particular normalization coefficient whereas the original methodology uses the single global parameter.

#### 4. EXPERIMENTAL RESULTS AND DISCUSSION

200 images from the BSDS training dataset have been used in the experiments [14]. Typical two examples from BSDS are shown in Figure 3(a). There are also reference images called ground truth in the dataset. The ground truth images were marked manually by different users for each image in the BSDS dataset. They are used for evaluation of the success of edge detection procedures. Figure 3(b) and 3(c) show two examples of the grand truths boundaries of the images in Figure 3(a). Figure 4 and Figure 5 show similarity transform results of images in Figure 3(a). The  $D_n$  normalization coefficients such as 16, 32, 64, 128, 192 and 256 are used to obtain the similarity images in Figure 4(a- f) and Figure 5(a- f), respectively. As can be seen from Figure 4 and Figure 5, the increase of the  $D_n$  coefficient also increases the homogeneity. On the other hand, Figure (6) shows the results of automatic similarity transform. The variation of the homogeneous areas and edges with fixed  $D_n$  values are clearly seen. Nevertheless, they are not real edge information. The similarity images need to be thresholded to attain edges. Therefore, the similarity images have been thresholded by known techniques. Subsequently, F-score was used to measure the success of the edge detection process. The F-score value is calculated by means of the harmonic mean as

$$Fscore = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (18)$$

where the recall indicates the proportion of the edge pixels that is selected by the algorithm and the precision indicates the accuracy of the selected edge pixels by the algorithm. Both quantities were estimated as follows:

$$Precision = \frac{TP}{TP+FP} \tag{19}$$

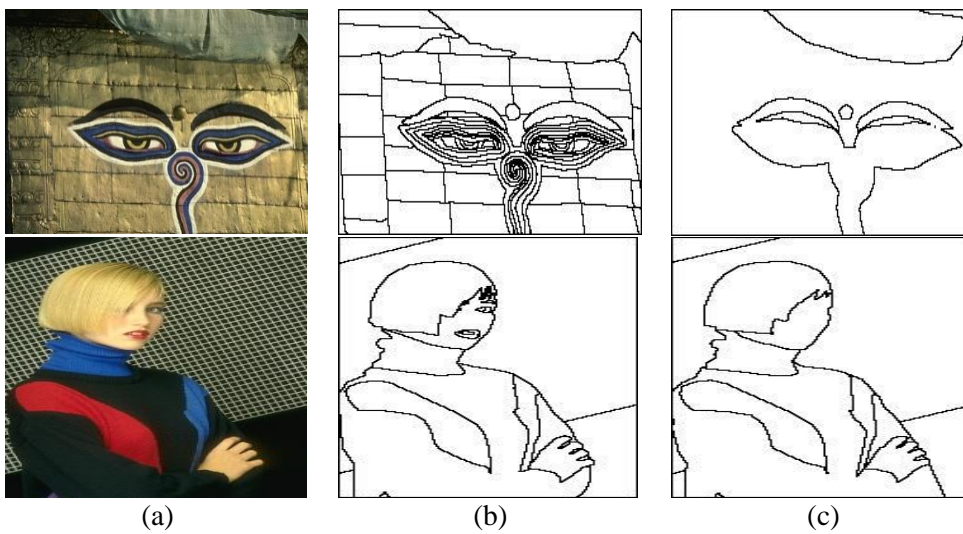
$$Recall = \frac{TP}{TP+FN} \tag{20}$$

The terms used in the calculations for F-score are given in Table 1.

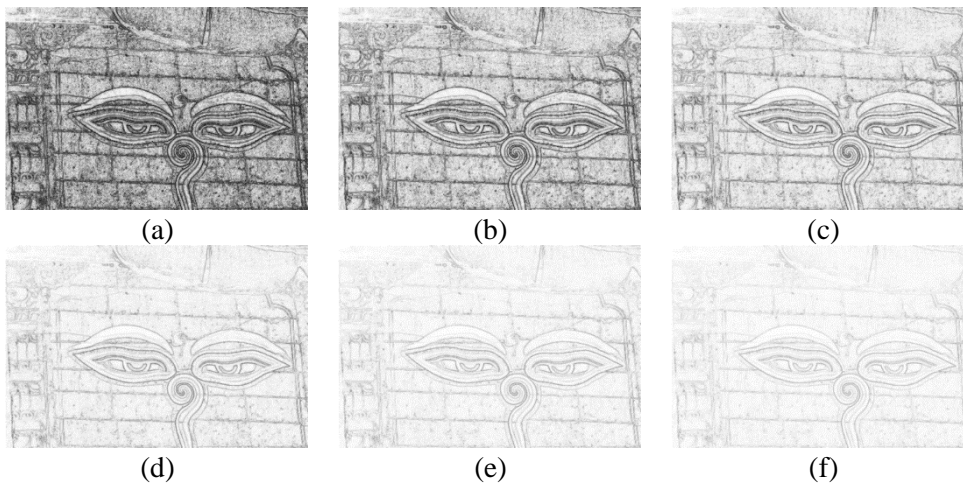
**Table 1.** F-Score Calculation Terms

		Algorithm Results	
		Edge Pixels	Non-Edge
Ground Truth	Edge Pixels	TP	FN
	Non-Edge	FP	TN

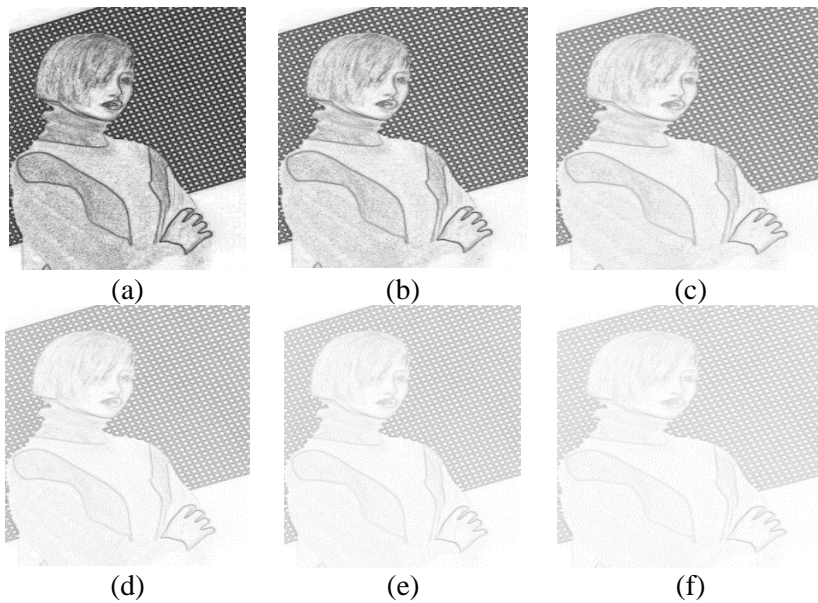
TP: True Positive, FN: False Negative, FP: False Positive, TN: True Negative



**Figure 3.** Sample images: BSDS (a) original images, (b)-(c) grand truth images



**Figure 4.** Similarity images: Eye a)  $D_n = 16$ , b)  $D_n = 32$ , c)  $D_n = 64$  d)  $D_n = 128$ , e)  $D_n = 192$ , f)  $D_n = 256$



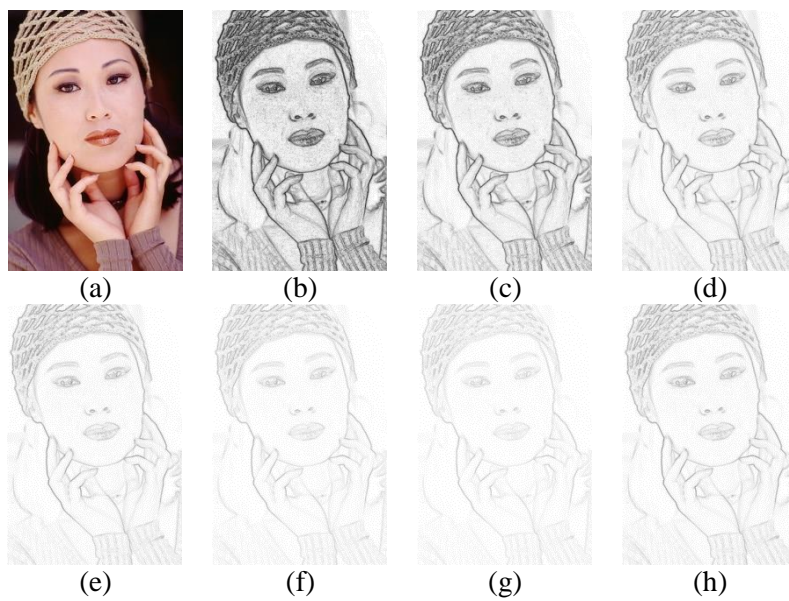
**Figure 5.** Similarity images: Lady a)  $D_n = 16$ , b)  $D_n = 32$ , c)  $D_n = 64$  d)  $D_n = 128$ , e)  $D_n = 192$ , f)  $D_n = 256$



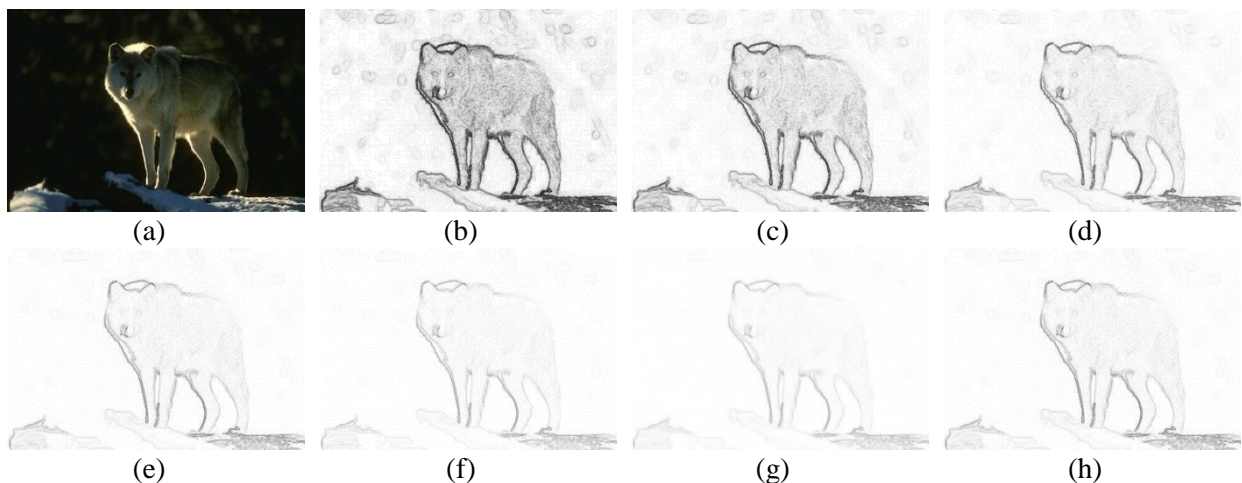
**Figure 6.** Similarity image obtained with proposed method

Similarity transformation results with different  $D_n$  values (16, 32, 64, 128, 192 and 256) including automatic calculation in Eq. (16) have been obtained for each of the 200 images in the BSDS training dataset at the beginning of the experiments. Three different thresholding methods have been applied to each of 1400 similarity image in order to calculate the F-score value for a total of 4200 binary images. The results of two typical images have been given below. Figure 7(b-g) show the similarity images obtained with fixed  $D_n$  coefficients for the images shown in Figure 7(a) whereas the similarity image obtained automatically are shown in Figure 7(h). Moreover, Figure 8(b-g) show the similarity images obtained with fixed  $D_n$  coefficients for the images shown in Figure 8(a) whereas the similarity image obtained automatically are shown in Figure 8(h). It could be observed that as the  $D_n$  coefficient increases, the homogeneous areas consisting of the pixels with similar gray levels increases. Nevertheless, the real edge pixels are considered as the member of homogeneous area and in fact, it is a kind of information losses. This kind of misclassification could be seen when Figure 7(b) and Figure 7(g) are compared. The similar cases could also be observed in Figure 8(b) and Figure 8(g) by means of the body of wolf and background mixture.

In second stage of experiments, the similarity images obtained have been thresholded in order to achieve edge information. Otsu, Kapur and CoG techniques have been employed for both images. As could be seen in Figure 9, the number of pixels marked as edges decreases with the high constant  $D_n$  coefficients whereas the non-edge pixels may be marked as edges for low values of  $D_n$ . Additionally, the higher  $D_n$  values increased the edge loss with Kapur and Otsu methods. On the other hand, edge losses are very low with the proposed automatic similarity transform approach. The edges in the face and hand area have been successfully obtained as could be seen in Figure 9. The similar success has been attained for wolf in Figure 10.



**Figure 7.** Similarity images: Girl a) original b)  $D_n=16$  c)  $D_n=32$ , d)  $D_n=64$ , e)  $D_n=128$ , f)  $D_n=192$ , g)  $D_n=256$ , h) proposed



**Figure 8.** Similarity images: Wolf a) original b)  $D_n=16$  c)  $D_n=32$ , d)  $D_n=64$ , e)  $D_n=128$ , f)  $D_n=192$ , g)  $D_n=256$ , h) proposed

Apart from the experiments given, the F-score performance evolution criteria have also been calculated for both images. Therefore, different normalization constants and different thresholding methods have been tested for both images. The F-score results for the edge results for Figure 9 are given in Table 2 while the results for Figure 10 are shown in Table 3. The highest F-score values for CoG, Kapur and Otsu for Girl image have been obtained with  $D_n$  coefficients of 192, 64 and 192 respectively. However, the scores obtained with proposed algorithm are very close to the most successful results and the differences are less than 1% for Figure 9. The similar success was also observed for Figure 10. The highest F-score values obtained for CoG, Kapur and Otsu are obtained with  $D_n$  coefficients of 192, 128 and 192 respectively. Nevertheless, performance of proposed techniques is already very close to the most successful results and the differences are also less than 1%.














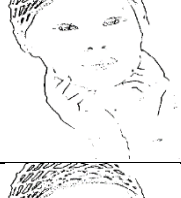







$D_n$	CoG	Kapur	Otsu
16			
32			
64			
128			
192			
256			
Proposed			

Figure 9. Edges: Girl






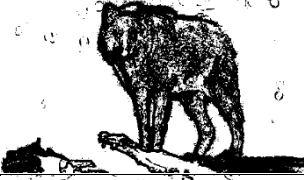
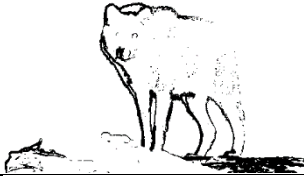
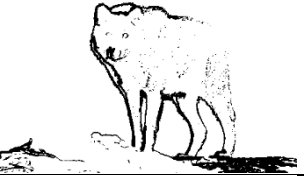





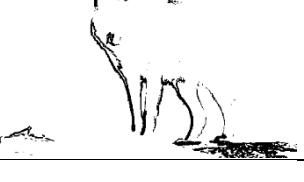
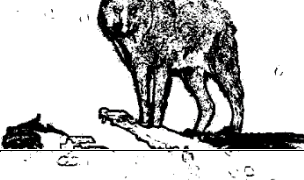

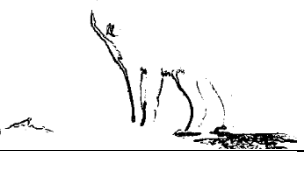


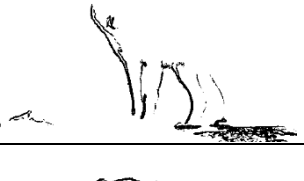



$D_n$	CoG	Kapur	Otsu
16			
32			
64			
128			
192			
256			
Proposed			

Figure 10. Edges: Wolf

**Table 2.** F-score results of Figure 9

$D_n$	F-score Values								
	CoG			Kapur			Otsu		
	Recall	Precision	F-score	Recall	Precision	F-score	Recall	Precision	F-score
16	0.4233	0.2214	0.2907	0.5766	0.3515	0.4367	0.4767	0.3005	0.3686
32	0.4409	0.2468	0.3164	0.6070	0.3678	0.4581	0.5272	0.3463	0.4180
64	0.4372	0.2755	0.3380	0.6218	0.3896	0.4790	0.5500	0.3720	0.4438
128	0.4469	0.2956	0.3559	0.5522	0.4106	0.4710	0.5622	0.3688	0.4454
192	0.4538	0.2986	0.3602	0.5455	0.4101	0.4682	0.5699	0.3779	0.4544
256	0.4507	0.2988	0.3594	0.5131	0.4091	0.4552	0.0095	0.3727	0.0185
<b>Proposed</b>	0.4469	0.2956	0.3559	0.5523	0.4100	0.4706	0.5621	0.3684	0.4451

**Table 3.** F-score results of Figure10

$D_n$	F-score Values								
	CoG			Kapur			Otsu		
	Recall	Precision	F-score	Recall	Precision	F-score	Recall	Precision	F-score
16	0.3867	0.1135	0.1755	0.4915	0.1823	0.2659	0.4148	0.1310	0.1992
32	0.4555	0.1210	0.1912	0.5000	0.2127	0.2985	0.4738	0.2016	0.2828
64	0.4584	0.1233	0.1944	0.4773	0.2345	0.3145	0.4773	0.2350	0.3149
128	0.4577	0.1238	0.1949	0.4692	0.2338	0.3121	0.4675	0.2360	0.3137
192	0.4518	0.1235	0.1940	0.4777	0.2302	0.3107	0.4777	0.2304	0.3109
256	0.4459	0.1233	0.1932	0.4748	0.2304	0.3103	0.4681	0.2255	0.3044
<b>Proposed</b>	0.4578	0.1235	0.1945	0.4692	0.2333	0.3116	0.4675	0.2360	0.3137

Although the experimental results of only two example images have been given in pervious figures and tables, the proposed algorithm has been also tested with the all 200 images from BSDS training dataset. The experiments were repeated for different values of  $D_n$  and each one of tresholding methods. The average values of recall, precision and F-score were calculated. Initially, the CoG technique was evaluated and its performance was given in Table 4. The most successful coefficient,  $D_n$  was observed as 256 with 0.3126 F-score while the average F-score value of the proposed method is 0.3090. Furthermore, it was observed that as the  $D_n$  increases, both recall and precision values are increased.

According to the results given in Table 5, the Otsu method has achieved its most effective results with automatic similarity approach and  $D_n=128$ . It could be seen that the recall reaches the highest value for  $D_n=64$  and the precision reaches the highest value for  $D_n=256$ . Also the results in Table 6 show that the highest success is obtained for  $D_n=192$  as 0.3812 F-score with Kapur thresholding technique. The average F-score value is obtained 0.3800 with the proposed method for Kapur. The recall reaches the highest value for  $D_n=32$  and the precision reaches the highest value for  $D_n=256$ .

The edge detection success in terms of F-score at the same  $D_n$  values is ordered as Kapur, Otsu and CoG methods. Otsu has the same F-score performance for the most successful  $D_n$  and automatic transform. On the other hand, the F-score difference between the best one and proposed procedure is 0.0036 with CoG, and 0.0012 with Kapur. Therefore, it could be concluded that there is no significant performance difference between the automatic approach and the most successful  $D_n$  coefficient.

**Table 4.** Average edge detection results: CoG

$D_n$	Recall	Precision	F-score
16	0.6548	0.1864	0.2902
32	0.6732	0.1916	0.2983
64	0.6872	0.1959	0.3049
128	0.6972	0.1985	0.3090
192	0.7030	0.2005	0.3121
256	0.7067	0.2006	0.3126
<b>Proposed</b>	<b>0.6972</b>	<b>0.1985</b>	<b>0.3090</b>

**Table 5.** Average edge detection results: Otsu

$D_n$	Recall	Precision	F-score
16	0.6772	0.2033	0.3127
32	0.7180	0.2098	0.3247
64	0.7363	0.2143	0.3319
128	0.7263	0.2196	0.3373
192	0.6651	0.2256	0.3369
256	0.4569	0.2277	0.3039
<b>Proposed</b>	<b>0.7263</b>	<b>0.2196</b>	<b>0.3373</b>

**Table 6.** Average edge detection results: Kapur

$D_n$	Recall	Precision	F-score
16	0.7491	0.2139	0.3328
32	0.7645	0.2271	0.3502
64	0.7249	0.2489	0.3705
128	0.6659	0.2659	0.3800
192	0.6309	0.2731	0.3812
256	0.6033	0.2776	0.3803
<b>Proposed</b>	<b>0.6659</b>	<b>0.2659</b>	<b>0.3800</b>

## 5. CONCLUSIONS

In this study, a novel automatic color edge detection technique based on similarity transformation has presented and its success has been quantitatively justified. The similarity images of 200 images from BSDS database have been obtained with different  $D_n$  normalization coefficients and subsequently, edge detection has been performed by using three different thresholding approaches. The achievement of edge detection processes is evaluated by means of F-score. It was observed that the F-score values of the proposed automatic algorithm are very close to the best results produced by the user-determined  $D_n$  coefficient. Additionally, it was observed that the real edge pixels in some images could not be detected with high values of  $D_n$  and the Otsu and Kapur thresholding techniques. However, the edge detection process for all test images have been successful with the proposed automatic approach. Consequently, the automatic determination of the  $D_n$  value with the proposed method removes the requirement of user supplied parameters. So an autonomous edge detection scheme has been developed. The edge thinning operations in order to obtain more successful results is the future direction of study.

## CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

**REFERENCES**

- [1] Gonzalez, R.C. and Woods, R.E., Digital Image Processing 3rd Ed., Pearson/Prentice Hall, New Jersey, (2008).
- [2] Canny, J., "A Computational Approach to Edge-Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6): 679-698, (1986).
- [3] Demirci, R., "Similarity relation matrix-based color edge detection", AEU-International Journal of Electronics and Communications, 61(7): 469-477, (2007).
- [4] Incetas, M.O., Demirci, R. and Yavuzcan, H.G., "Automatic segmentation of color images with transitive closure", AEU-International Journal of Electronics and Communications, 68(3): 260-269, (2014).
- [5] Güvenç U., Elmas Ç. and Demirci R., "Automatic Segmentation of Color Images", Journal of Polytechnic, 11(1): 9-12, (2008).
- [6] Ali, H.I., "Fast Color Edge Detection Algorithm Based on Similarity Relation Matrix", International Journal of Computer Science Issues, 10(5): 108-113, (2013).
- [7] Incetaş, M.O., Tanyeri, U., Kılıçaslan, M., Girgin, B.Y. and Demirci, R., "Eşik Seçiminin Benzerliğe Dayalı Kenar Belirlemeye Etkisi", ISMSIT2017 - International Symposium on Multidisciplinary Studies and Innovative Technologies, Tokat, Turkey, 102-106, (2017).
- [8] Aydın, M., Hardalac, F., Ural, B. and Karap, S., "Neonatal Jaundice Detection System", Journal of Medical Systems, 40(7): 1-11, (2016).
- [9] Tanyeri, U., Incetas, M.O. and Demirci, R., "Similarity based Anisotropic Diffusion Filter", IEEE 24th Signal Processing and Communication Application Conference (SIU 2016), Zonguldak, Turkey, 1401-1404, (2016).
- [10] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man, and Cybernetics, 9(1): 62-66, (1979).
- [11] Kapur, J.N., Sahoo, P.K. and Wong, A.K.C., "A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram", Computer Vision Graphics and Image Processing, 29(3): 273-285, (1985).
- [12] Demirci, R., "Adaptive threshold selection for edge detection in colour images", IEEE 18th Signal Processing and Communications Applications Conference (SIU 2010), Diyarbakır, Turkey, 677-679, (2010).
- [13] Incetaş, M.O., "Analysis Of Medical Images With Adaptive Region Growing Algorithm", Phd. Thesis, Graduate School Of Natural And Applied Sciences, Ankara, 57-62 (2014).
- [14] Martin, D., Fowlkes, C., Tal, D. and Malik, J., "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics", Eighth IEEE International Conference on Computer Vision, Vol II, Proceedings: 416-423, (2001).