A Pixon-based Image Segmentation Method Considering Textural Characteristics of Image

M.- H. Khosravi[*]

Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran.

Abstract

Image segmentation is an essential and critical process in image processing and pattern recognition. In this paper, we proposed a textured-based method to segment an input image into regions. In our developed method, an entropy-based textured map of image is extracted, followed by a histogram equalization step to discriminate different regions. Then with the aim of eliminating unnecessary details and achieving more robustness against unwanted noises, a low-pass filtering technique is successfully used to smooth the image. As the next step, the appropriate pixons are extracted and delivered to a fuzzy c-mean clustering stage to obtain the final image segments. The results of applying the proposed method on several different images indicate its better performance in image segmentation compared to the other pixon-based methods, especially for the images with textured regions.

Keywords: *Image Segmentation, Image Texture, Pixon.*

1. Introduction

Image segmentation is one of the most important operations in image processing and computer vision, with numerous applications in image understanding and content analysis. The goal of segmentation is to partition an image into multiple uniform regions, which are more meaningful and easier to analyze than the raw set of pixels. These regions might be some tumor tissues in medical imaging [1-3], color regions in artistic cartoons [4], or some human faces, pedestrian, roads, forests, and so on in object detection applications. Formally, suppose $Uni(A)$ to be a uniformity predicate for all elements in A, in which when $Uni(A)$ is true for some region A, then $Uni(B)$ is also true for any $B \subset A$. Having this uniformity predicate, the image segmentation is the process of partitioning an image I into disjoint non-empty regions $I_i (i = 1, 2, ..., n)$ with the following conditions [5, 6]:

$$
\begin{cases}
U_{i=1}^{1} I_i = I, \\
I_i \cap I_j = \varnothing & \text{for all } i \neq j, \\
Uni(I_i) = TRUE & \text{for } i = 1, L, n \\
Uni(I_i \cup I_j) = FALSE & \text{for all } i \neq j,
\end{cases}
\tag{1}
$$

The first two conditions illustrate the partitioning behavior of segmentation, and the last two conditions emphasize on the uniformity of the extracted regions.

In the recent decades, a considerable amount of literature have been published on image segmentation, and various techniques have been presented for it. These techniques can be classified into two major categories: (1) region-based, and (2) contour-based approaches. Both of these approaches rely on the pixel relationships to their local neighborhood. Region-based approaches try to find partitions of the image based on some similarity property of the pixels, with respect to the texture, brightness, and color. On the other hand, contour-based approaches perform based on some discontinuity property of the pixels. These methods usually start with an edge detection phase, followed by a linking process [7]. One of the most important segmentation methods that belongs to the first aforementioned category is pixon-based image segmentation [8]. The concept of pixon was first introduced by Pina and Puetter for astrophysical image reconstruction [9, 10].
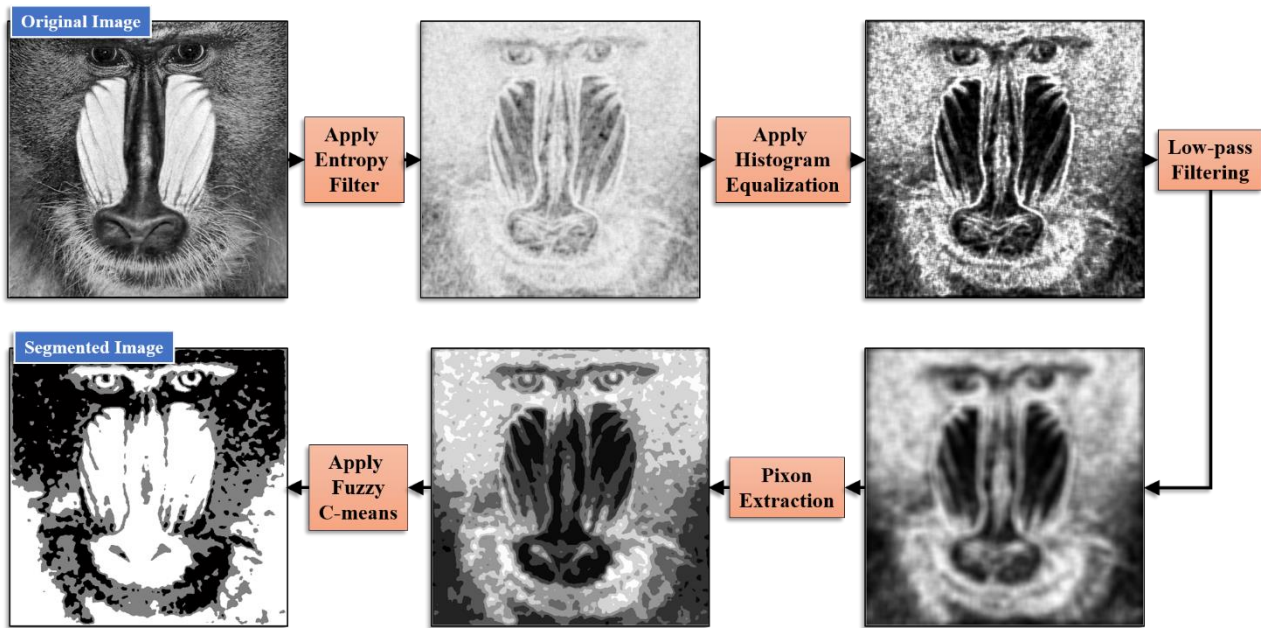
Figure 1. Flowchart of proposed segmentation method.

The pixons are cells that locally define the resolution of the data. The local resolution means that at each image pixel there is the finest spatial scale that has the ultimate information of that pixel or its surroundings, and that there is no information content below this scale. More easier, we need more spatial resolution to capture the information of a non-smooth detailed region of image, while less resolution is required for the portions of image with coarse structure.

In the pixon definition of [9], the image is modeled by a local convolution of a pseudo-image and a kernel function, which is commonly a circularly symmetric pixon with variable size but fixed shape. A modified definition of pixon was introduced by Yang *et al.* in [11]. In Yang's definition of pixon, the shape and size of pixons can vary simultaneously, and so it is more convenient for image segmentation. They use the anisotropic diffusion equation to form the pixons.

Lin *et al.* have proposed an image segmentation method based on the Markov random field (MRF) model, which is applied on a pixon-based image representation [12]. They suggested a Fast QuadTree Combination (FQTC) algorithm to extract the good pixon-representation.

Later, Hassanpour *et al.* [13] proposed another pixon-based image segmentation method, which has two major differences with the Yang's and Lin's methods. Firstly, with the aim of image smoothing, they used the wavelet thresholding instead of the diffusion equation, and secondly, they replaced the MRF algorithm with the fuzzy c-mean one. In this paper, we propose a segmentation approach based on the concept of

pixons. We noticed that the performance of the existing pixon-based methods could be improved by incorporating the texture characteristics of the image regions. In the proposed method, firstly, we extracted the texture regions by employing the entropy, as a well-known statistical measure. The texture map obtained passes from the histogram equalization and low-pass filtering stages, as two pre-processing stages, before the pixon extraction. After pixon extraction, the final segments were formed using the fuzzy c-mean algorithm. These stages are shown in the flowchart of figure 1. Our experimental results show that these improvements lead to more robust segmented regions, and remarkably reduce the computational cost of segmentation.The rest of the paper is organized as what follows. Section 2 introduces and discusses different stages of the proposed method. In this section, we give a brief description of the pixon concept and also the fuzzy c-mean algorithm. Section 3 is devoted to the introduction of two quantitative evaluation measures that gauge the performance of the segmentation methods. The experimental results are reported in Section 4, and conclusions are derived in Section 5.

2. Proposed method

This section describes the proposed textured-based image segmentation technique. In particular, we will divide this section into five sub-sections, which deal with successive stages of the proposed method, illustrated in the flowchart of figure 1.
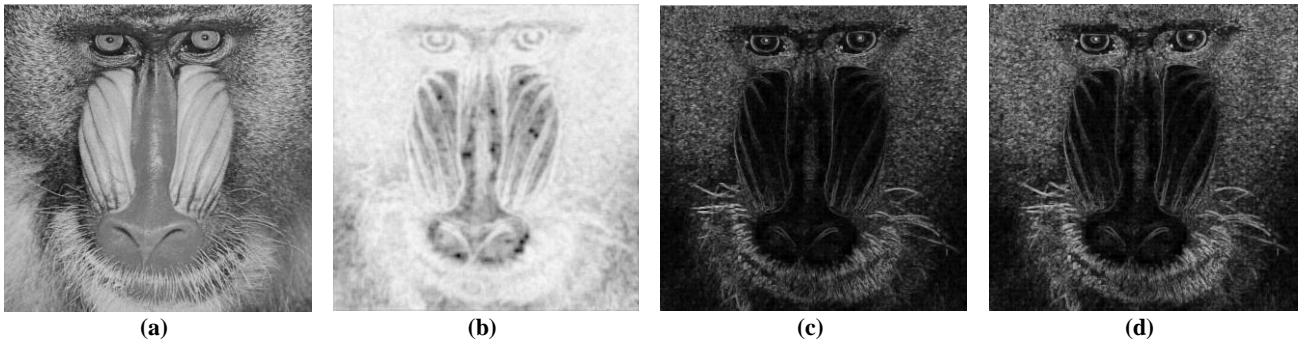
Figure 2. Comparison of maps obtained by three different statistical texture descriptors. (a) Original image of baboon, (b) entropy-based texture map, (c) standard deviation-based texture map, and (d) range filter texture map.

2.1. Texture extraction

Texture is a local-neighborhood property of image homogeneous regions, which has a vital role in the task of region segmentation in the human visual system (HVS). It has been shown that there exists a pre-attentive visual system in HVS with the ability to identify the basic primitives of image textures [14], and consequently, discriminate the regions with different textural appearances. Therefore, the texture-based image segmentation is entirely based upon the natural process of HVS segmentation.

Mainly, the image texture regions are analyzed in two different ways: structural approach and statistical approach. In the structural approach, the goal is to find the fundamental units of a texture map (i.e. texels) in some regular or repeated arrangements, which mostly exist in synthesized artificial textures. Differently, in the statistical approach, the arrangement of intensities in a region is investigated as a quantitative measure. This approach has advantages over the first one. First, it is less sensitive to the spatial arrangement of texture elements, and so is more appropriate for analysis of natural images. Secondly, the statistical approach is easier to compute and has a lower computational complexity.

Considering these advantages, we followed the statistical approach to distinguish the texture regions of an image. Here, we tested some well-known statistical quantities like the local standard deviation of a region, dynamic range that determines the difference between the maximum and minimum values in a specified neighborhood, and the local entropy of a patch. Figure 2 shows the three maps corresponding to these three statistical quantities for the image of baboon. Here, we chose the entropy measure due to its better performance obtained during our experiments. It is well known that the most common interpretation of image entropy is as a measure of randomness or uncertainty of a set pixel intensities. On the other hand, the rate of intensity changes in a texture patch (i.e. its

activity level) is obviously higher than the ones in a flat and non-texture patch. Hence, the patch entropy is a reasonable choice for quantifying the amount of patch texturedness.

The local entropy H of an image patch B is defined as:

$$H = -\sum_{k=0}^{M} P_k(B)\log_2(P_k(B)) \qquad (2)$$

where, M is the number of gray levels and P_k is the probability associated with the kth grayscale in patch B. Here, we constructed an entropy map with the same size of the original image by calculating the entropy of overlapped patches of size 9×9 (one patch centered at every pixel). It should be noted that the higher-value elements in entropy map correspond to the pixels with high dynamicity in their neighborhood.

2.2. Histogram equalization

The value of each entropy map element lies in the range of $[0, \log_2(M)]$. This range of dynamicity leads to a low-contrast map, in which the difference between the textured and non-textures regions is not noticeable. In order to have more discriminated texture regions, we had to increase the global contrast of the map. To do so, we rescaled the values of entropy map to the range of $[0-255]$, and stretch its dynamic range as much as possible by applying the histogram equalization algorithm on the rescaled map. The resulting map is a matrix, in which the highest values correspond to the textured regions and the lowest values indicate the smooth non-texture regions in the original image.

2.3. Low-pass filtering

High-frequency details of the histogram equalized entropy map are annoying elements during the pixon extraction in the next step. In addition, the existence of high-frequency noises such as Gaussian additive noise or salt and pepper noise can potentially affect the results of pixon

extraction, and subsequently, the final results of image segmentation. Hence, we performed a Gaussian low-pass filter on the histogram equalized entropy map at this stage.

2.4. Pixon extraction

In this work, we employed the Yang's model of pixons [11]. In this model, an image is a collection of disjoint pixons, which completely cover the entire image (i.e. $I = U_{\{i=1\}}^n P_i$, in which I is the pixon-based image model, P_i is a pixon, and n is the number of pixons). A pixon, by itself, is a set of connected pixels, a single pixel or even a sub-pixel, and hence both the shape and size of each pixon can vary. The pixons of this model, are extracted in the following three steps [11]:

1) Prepare a pseudo-image with at least the same resolution as the input image. A pseudo-image is obtained by increasing the resolution through interpolation, with the aim of describing the image regions with a lot of details. More formally, if the original image has the dimension $M \times N$, then the dimension of the pseudo-image is $2^m M \times 2^m N$, where m is the algorithm parameter. For $m = 0$, the original image and the pseudo-image are identical, and for $m > 1$, the pseudo-image is build out iteratively by applying the bilinear interpolation on the rescaled pseudo-image of the previous iteration. It is worthy to note that in the case of $m > 1$, the finally pixons formed are probable to be a sub-pixel.

2) Form the pixons using an anisotropic diffusion filter [15]. The anisotropic diffusion is an extension to isotropic diffusion, which is a blurring process motivated from the scale space concept. The isotropic diffusion is a space-invariant transformation, and removes edges and other details of image contents during smoothing. In contrast, the anisotropic diffusion is a space-variant and non-linear transformation of the original image, which behaves locally at different image regions. In regions close to the edges, this method diffuses along the edges but not across them. Unlikely, in smooth areas, the method performs standard isotropic diffusion. Thus, this filter smooths the image more in homogenous regions than in texture regions. This behavior is desirable for our application. In the output image of this step, the regions with less information (having fewer edges) will tend to be uniform, and hence, can be regarded as the pixons.

3) Extract the final pixons using a simple segmentation algorithm based on hierarchical clustering. The final output is a planar graph of pixons.

2.5. Fuzzy C-Means

Fuzzy c-means (FCM) algorithm, introduced by Dunn [16] and extended by Bezdek [17], is one of the most widely used clustering algorithms in image segmentation. This method employs fuzzy membership to assign pixels to different categories. Let $X = (x_1, x_2, ..., x_N)$ denotes an image with N pixels, in which x_i represents the gray-scale value of the ith pixel. The FCM algorithm, with the aim of partitioning X into c clusters, performs an iterative optimization that minimizes the following cost function:

$$J = \sum_{j=0}^{N} \sum_{i=1}^{c} u_{ij}^m \left\| x_j - v_i \right\|^2, \qquad (3)$$

where, $\left\| \bullet \right\|$ is a norm metric and $\{v_i\}_{i=1}^c$ stands for the centers of the clusters. The array $U = \{u_{ij}\}$ includes the fuzzy membership factors, each denoting the membership of pixel x_j to the ith cluster, satisfying:

$$U = \left\{ u_{ij} \in [0,1] \mid \sum_{i=1}^{c} u_{ij} = 1, \forall j, \text{ and } 0 < \sum_{j=1}^{N} u_{ij} < N, \forall i \right\} (4)$$

In equation 3, m is a constant, which controls the fuzziness of the results. The membership factor (i.e. u_{ij}) indicates the probability that a pixel belongs to a specific cluster. The cost function minimization is happened when pixels far from the clusters' centroid are assigned low membership values, and high membership values are devoted to the pixels close to the centroid of their clusters. This is done by iteratively updating the values of membership factors and the centers of clusters by the following:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\left\| x_j - v_i \right\|}{\left\| x_j - v_k \right\|} \right)^{\frac{2}{m-1}}} \qquad (5)$$

and

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^m x_j}{\sum_{j=1}^{N} u_{ij}^m} \qquad (6)$$

The FCM algorithm starts with initial guesses for each cluster center, and ends with convergence to the solution values for these cluster centers, which lead to a minimum value for cost function of (3).

3. Segmentation evaluation metrics

We performed both the visual and quantitative comparisons between the proposed approach and the methods suggested by Yang [11], Lin [12],

and [13]. For a quantitative comparison, we employed the following metrics:

1) *Pixon to Pixel Ratio (PPR)*: A good segmentation process must obtains as large as possible meaningful segments, with low details [18]. In the pixon-based methods, the number of pixons can be regarded as a suitable measure, evaluating the size of segments. In other words, lower pixons means larger segments. We used the ratio between the number of pixons and the number of pixels of image [13] to obtain a normalized measure, which can be used for comparing the images with different sizes.

2) *Normalized sum of segment variances (NSSV)*: NSSV is one of the most important quantitative measures employed to perform evaluation of image segmentation approaches [13]. It is well known that a suitable segmentation method should produce segments with a maximum amount of homogeneity. In other words, the variance of pixel intensities among each segment must be adequately low. The NSSV measure is used to gauge this phenomenon. Assume that the original image with size $M \times N$ is partitioned into c segments during the segmentation process. NSSV is defined as below:

$$
\begin{aligned}
NSSV \quad &= \quad \frac{\text{Sum of partial variances}}{\text{Image variance}} \\
&= \quad \frac{\sum_{k=1}^{c} \frac{N_k \times V_k}{M \times N}}{\frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(I(i,j) - \mu \right)^2} \qquad (7) \\
&= \quad \frac{\sum_{k=1}^{c} N_k \times V_k}{\sum_{i=1}^{M} \sum_{j=1}^{N} \left(I(i,j) - \mu \right)^2}
\end{aligned}
$$

in which N_k and V_k denote the number of pixels and the variance of segment k, respectively, and μ is the intensity mean of the whole image. Smaller values for NSSV imply more homogeneity of the regions and, consequently, better segmentation results.

4. Experimental results

To demonstrate the segmentation performance of our method, we tested it on three standard natural images, from [19], namely, the baboon, the pirate, and the pepper images, shown in figure 4(a), figure 5(a), and figure 6(a), respectively, to produce the segmented images, which can be compared with the results of the Yang's, Lin's, and Hassanpour's methods. To have a fair comparison, we set the FCM algorithm to produce three clusters (i.e. segments) for each image. As we mentioned in the previous section, the comparisons were made in two visual and quantitative schemes.

4.1. Visual comparison

Figure 3 illustrates the output of the stages of the proposed segmentation method on a test image containing both the dominant texture and the smooth regions. It can be seen that the proposed method performs a reasonable segmentation with suitable results. The smooth regions include the entire sky, and a small patch on the right bottom corner of the image are segmented well from other regions that contain non-smooth texture patches.

We also verified our image segmentation method by the visual comparison of the results of the proposed method with the results of the Yang's, the Lin's, and the Hassanpour's methods, applied on the baboon, pirate, and pepper images shown in figures 4, 5, and 6, respectively. It can be seen that our proposed method is working well for images with dominant textures. For example, in the baboon image, the nose part of the baboon's face and also the surrounding regions of its eyes are completely different from the other parts in terms of smoothness, and hence, our method assigns different segment labels to these parts. Similarly, in the pirate image, the smooth and non-smooth texture regions are segmented into different classes, successfully. In this image, black segments correspond to the textured regions with high dynamicity, gray segments show the smooth regions, and white segments indicate moderately textured regions. For these two images, the superiority of our method against other three approaches is clear. However, in pepper image shown in figure 6, there are not dominant texture regions and almost all of the image regions are smooth. Our proposed method is not suitable for this kind of image, as predicted.

4.2. Quantitative comparison

As mentioned earlier, we performed the quantitative comparison by assessing and comparing the PPR and NSSV measures, introduced in Section 3. Table 1 shows the number of pixels, number of pixons, and the ratio between the number of pixons and pixels (PPR) for three equal-size 512×512 images: baboon, pirate, and pepper. In addition, table 2 compares the value of PPR for different competitive methods, for these images. In each row of this table, the best value of PPR is bolded for a better comparison. It can be seen that the proposed method has the best PPR for the images of baboon and pirate, but its PPR for the pepper image is not as good as other three methods. As mentioned in the previous sub-section, it is due to the lack of texture regions in the pepper image.

(a)

(b)

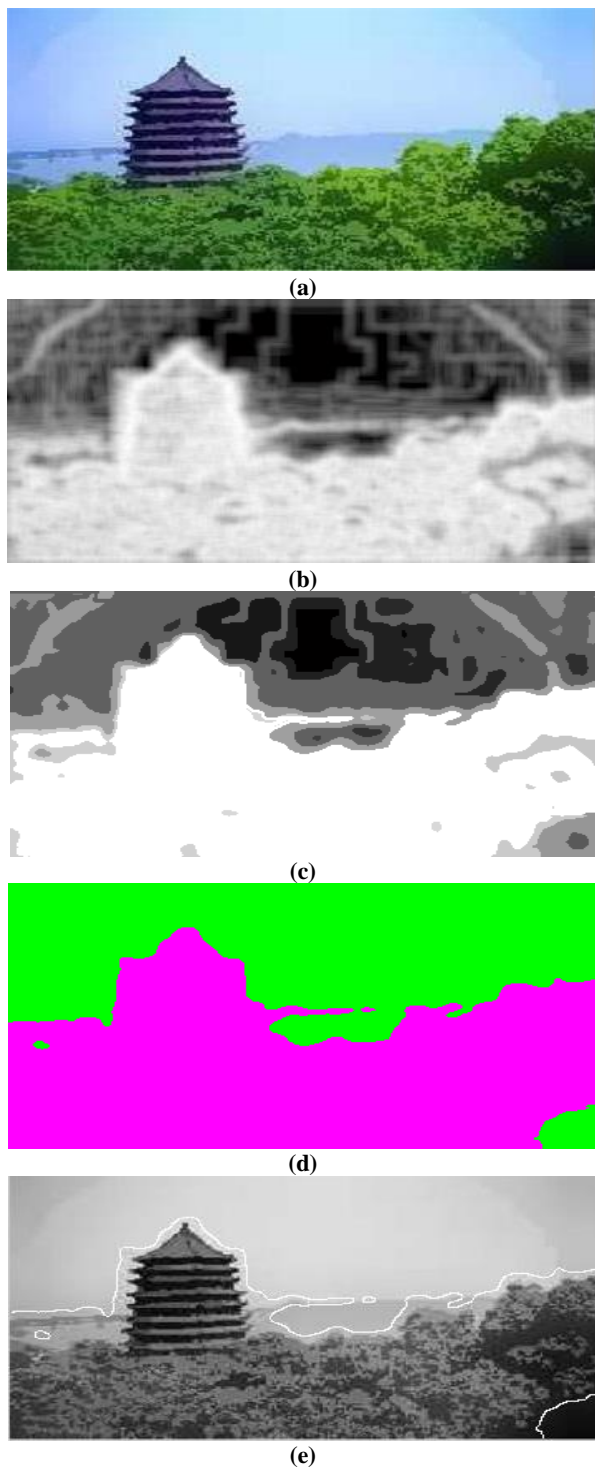**(c)**

(d)

(e)

Figure 3. Segmentation results of a test image. (a) Original image, (b) texture map after histogram equalization and low-pass filtering, (c) pixon map obtained from texture map, (d) clustered image after FCM, and (e) clusters' boundaries shown on original image.

Table 3 indicates the values of averages and variances of intensities for each class (i.e. segment) obtained by the four competing methods, on three pre-mentioned images. In addition, the values of NSSV, which is defined in (7), are also reported in this table, for the three images. For each image, the best value of NSSV

is bolded for a simple comparison. It can be inferred clearly from table 3 that the proposed method has the minimum value of NSSV measure for the baboon and pirate images, which contain texture regions.

Table 1. Number of pixons and pixels of images after applying the proposed method.

Images	The number of pixons	The number of pixels	Pixon to Pixel Ratio (PPR)
Baboon	20129	262144	7.68%
Pirate	19176	262144	7.31%
Pepper	38827	262144	14.8%

Table 2. Comparison of ratio between number of pixons and pixels, among four methods.

Images	Yang's method	Lin's method	Hassanpour's method	Proposed method
Baboon	31.8%	23.39%	9.79%	**7.68%**
Pirate	36.52%	28.44%	12.24%	**7.31%**
Pepper	12.2%	9.43%	**5.04%**	14.8%

4.3. Computational complexity

Table 4 indicates the computational time required by the competing methods, for the segmentation of each image. Obviously, the proposed segmentation method is the fastest, compared to the other three competitors. This low computational complexity of the proposed model comes from two major aspects: (1) we extracted the texture map using the patch entropy of image, which is relatively a low cost function, and (2) we eliminated the annoying details of texture map by applying a Gaussian high-pass filter, which has a lower complexity than the wavelet thresholding of the Hassanpour's model. The most CPU consuming stages of the proposed method are the pixon extraction and the fuzzy c-mean clustering. Figure 7 shows the actual computational times (in seconds) for different stages of our method for segmentation of the baboon image. It can be seen that the highest percentage of the overall time belongs to the FCM stage, followed by the pixon extraction one.

5. Conclusion

In this paper, we have proposed an image segmentation algorithm based on the pixon concept. In our algorithm, we have noticed the role of texture characteristics of the image regions and employed them to enhance the results of image segmentation. We used successfully the local entropy of image patches as an efficient, yet effective texture descriptor. Histogram equalization and low-pass filtering were the two successive stages that improved the quality of the texture map. Employing the pixon extraction algorithm, followed by fuzzy c-mean, yields the final segmented image.
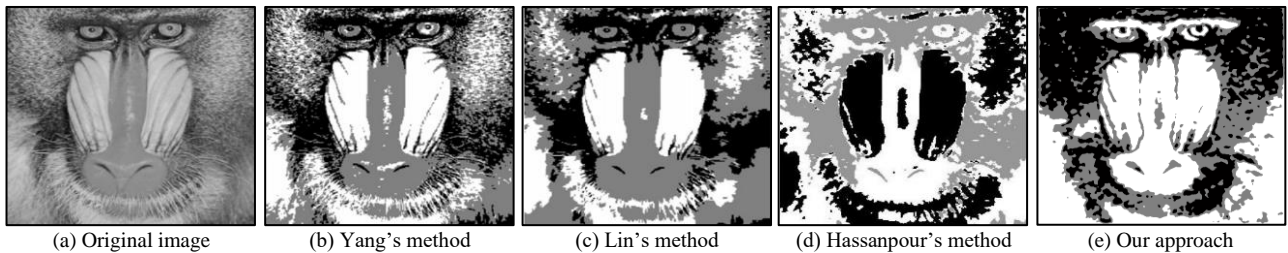
| (a) Original image | (b) Yang's method | (c) Lin's method | (d) Hassanpour's method | (e) Our approach |

Figure 4. Segmentation results of baboon image.

| (a) Original image | (b) Yang's method | (c) Lin's method | (d) Hassanpour's method | (e) Our approach |
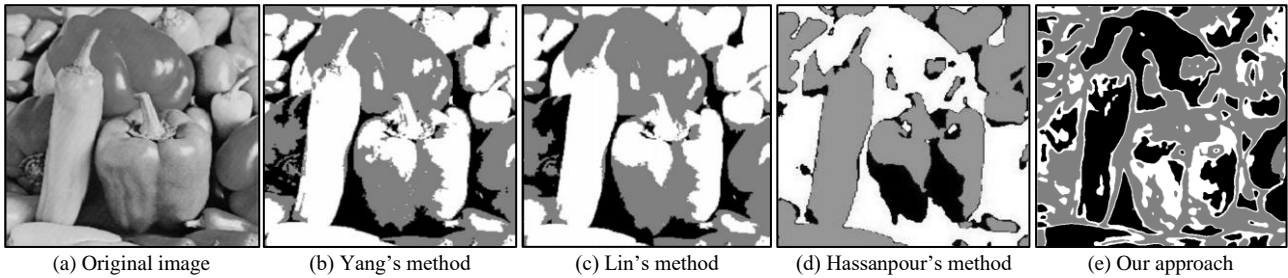
Figure 5. Segmentation results of pirate image.

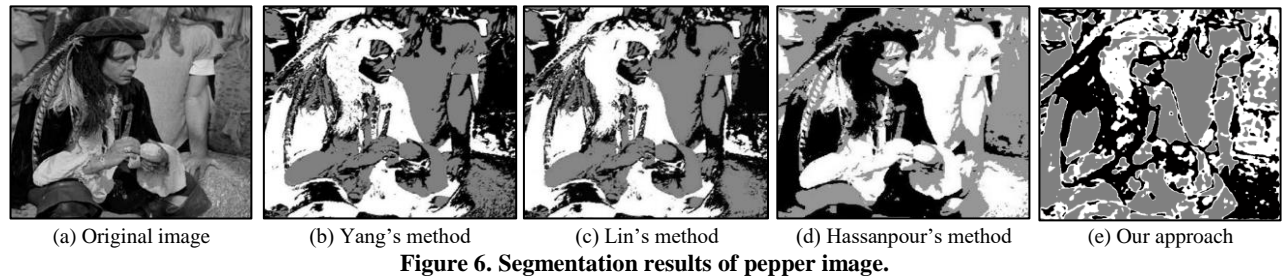| (a) Original image | (b) Yang's method | (c) Lin's method | (d) Hassanpour's method | (e) Our approach |

Figure 6. Segmentation results of pepper image.

Table 3. Comparison of averages and variances of each class, and overall NSSV obtained by four competing methods.

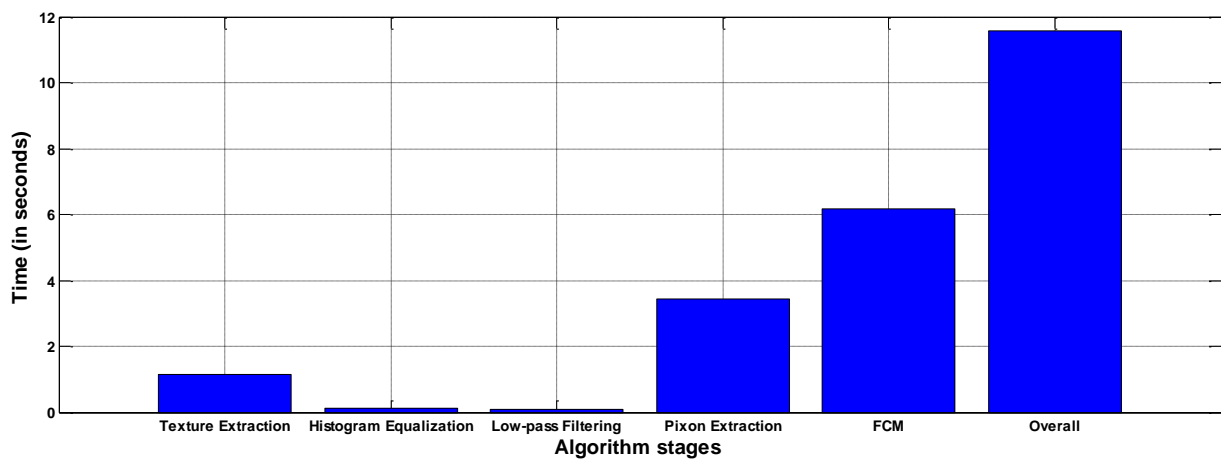| Images	Classes	Yang's method			Lin's method			Hassanpour's method			Proposed method		
		Average	Variance	NSSV	Average	Variance	NSSV	Average	Variance	NSSV	Average	Variance	NSSV
Baboon	Class 1	203.13	12.18		217.35	12.05		198.29	11.35		180.63	12.17	
	Class 2	130.43	11.06	0.1203	114.54	11.56	0.1113	108.74	11.46	0.0917	140.93	10.43	**0.0821**
	Class 3	48.2	17.37		56.36	16.68		44.16	16.96		98.76	15.27	
Pirate	Class 1	177.87	21.31		181.92	20.67		183.85	19.37		168.73	18.17	
	Class 2	168.28	18.91	0.1117	152.72	18.84	0.1102	170.05	16.83	0.1014	128.05	19.24	**0.0941**
	Class 3	23.82	17.68		31.90	16.18		68.84	16.52		25.82	16.35	
Pepper	Class 1	122.38	25.97		123.76	16.28		124.79	24.32		122.13	32.65	
	Class 2	196.37	21.35	0.0911	195.87	22.66	0.0856	192.78	18.36	**0.0822**	191.21	27.32	0.1412
	Class 3	32.70	22.86		33.6	22.30		35.06	22.41		32.07	27.88	

Figure 7. Actual computational times (in seconds) for different stages of the proposed method for image of baboon.

Table 4. Comparison of the computational time (in milliseconds) between four methods.

Images	Yang's method	Lin's method	Hassanpour's method	Proposed method
Baboon	18549	19326	15316	11469
Pirate	25651	22910	17378	10215
Pepper	16143	17034	13066	8835

There are two major differences between our method and the pre-mentioned Hassanpour's method. First, in our method, the input of the pixon extraction stage is the enhanced texture map of the image instead of the original image. Secondly, we employed the low-cost Gaussian low-pass filter instead of the wavelet thresholding. By incorporating these two modifications, the computational cost was decreased compared to the Hassanpour's image segmentation algorithm. In addition, the experimental results demonstrate that our algorithm performs fairly well, especially for images with dominant texture regions.

References

[1] Ain, Q., Jaffar, M. A., & Choi, T.-S.. (2014). Fuzzy anisotropic diffusion based segmentation and texture based ensemble classification of brain tumor, Applied Soft Computing, vol. 21, no. 1, pp. 330-340.

[2] Gordillo, N., Montseny, E., & Sobrevilla, P. (2013). State of the art survey on mri brain tumor segmentation, Magnetic Resonance Imaging, vol. 31, no. 8, pp. 1426-1438.

[3] Noble, J. A. & Boukerroui, D. (2006). Ultrasound image segmentation: A survey, IEEE Transactions on Medical Imaging, vol. 25, no. 8, pp. 987-1010.

[4] Fateh, M. & Kabir, E. (2018). Color reduction in hand-drawn persian carpet cartoons before discretization using image segmentation and finding edgy regions, Journal of AI and Data Mining, vol. 6, no. 1, pp. 47-58.

[5] Zhang, Y.-J. (2006). Advances in image and video segmentation, Pennsylvania: IGI Global.

[6] Felzenszwalb, P. F. & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation, International journal of computer vision, vol. 59, no. 2, pp. 167-181.

[7] Malik, J., et al. (2001). Contour and texture analysis for image segmentation, International Journal of Computer Vision, vol. 43, no. 1, pp. 7-27.

[8] Hassanpour, H., Yousefian, H. & Zehtabian, A. (2011). Pixon-based image segmentation, in Image segmentation. InTech.

[9] Pina, R .K. & Puetter, R. C. (1993). Bayesian image reconstruction: The pixon and optimal image modeling, Publications of the Astronomical Society of the Pacific, vol. 105, no. 688, pp. 630.

[10] Puetter, R. (1995). Pixon-based multi-resolution image reconstruction and the quantification of picture information content, International Journal of Imaging Systems and Technology, vol. 6, no. 4, pp. 314-331.

[11] Yang, F. & Jiang, T. (2003). Pixon-based image segmentation with markov random fields, IEEE Transactions on Image Processing, vol. 12, no. 12, pp. 1552-1559.

[12] Lin, L., et al. (2008). A novel pixon-representation for image segmentation based on markov random field, Image and Vision Computing, vol. 26, no. 11, pp. 1507-1514.

[13] Hassanpour, H., et al. (2009). A novel pixon-based approach for image segmentation using wavelet thresholding method. in International Conference Image Analysis and Recognition. Berlin, Heidelberg, 2009.

[14] Julesz, B. & Bergen, J. R. (1983). Textons, the fundamental elements in preattentive vision and perception of textures, The Bell System Technical Journal, vol. 62, no. 6, pp. 1619-1645.

[15] Perona, P. & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 629-639.

[16] Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters, Journal of Cybernetics, vol. 3, no. 3, pp. 32-57.

[17] Bezdek, J. C., Ehrlich, R. & Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm, Computers & Geosciences, vol. 10, no. 2, pp. 191-203.

[18] Fu, K. S. & Mui, J. K.. (1981). A survey on image segmentation, Pattern Recognition, vol. 13, no. 1, pp. 3-16.

[19] Image databases. (2017). Available from: http://imageprocessingplace.com/root_files_V3/image_databases.htm.