A Novel Pixon-Based Image Segmentation Process Using Fuzzy Filtering and Fuzzy C-mean Algorithm

Ehsan Nadernejad and Amin Barari

Abstract

Image segmentation, which is an important stage of many image processing algorithms, is the process of partitioning an image into nonintersecting regions, such that each region is homogeneous and the union of no two adjacent regions is homogeneous. This paper presents a novel pixon-based algorithm for image segmentation. The key idea is to create a pixon model by combining fuzzy filtering as a kernel function and a fuzzy c-means clustering algorithm for image segmentation. Use of fuzzy filters reduces noise and slightly smoothes the image. Use of the proposed pixon model prevented image over-segmentation and produced better experimental results than those obtained with other pixon-based algorithms.

Keywords: Image segmentation, clustering, fuzzy c-mean, fuzzy filtering, pixonal image.

1. Introduction

Image segmentation refers to the process of partitioning an image into multiple segments, with the goal of simplifying and/or changing the image representation into something more meaningful and analyzable. Image segmentation is typically used to locate objects and boundaries in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image, such that pixels with the same label share certain visual characteristics. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. All of the pixels in a given region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s).

Depending on the image acquisition model, images can be classified into various types: e.g., light intensity (visual), range/depth, magnetic resonance, or thermal images. Light intensity images, which are the most commonly encountered image type, represent the variation of light intensity on the screen. A range image is a map of depth information at different points on the screen. Popular segmentation approaches include threshold [1], edge-based [2], region-based [3], split-and-merge [4], connectivity-preserving relaxation [5], and clustering-based methods [6-11].

Threshold techniques [1] make decisions based on local pixel information. These techniques are effective when the intensity levels of objects fall squarely outside the range of background levels. Because spatial information is ignored, however, blurred region boundaries can create havoc. Edge-based methods are centered on contour detection. Because their weakness is in connecting broken contour lines, edge-based methods are also prone to failure in the presence of blurring [2]. In region-based techniques, image pixels are classified into groups and used to separate the image into areas or subareas. The subareas are united or separate parts of the image that include the whole pixels of the image. In this method, a seed is selected and, by aggregation of adjacent pixels with similar features, larger areas are obtained.

In split-and-merge methods [4], the initial image is segmented into four symmetrical areas. According to specified conditions of the image, these areas are merged together or segmented into smaller sub-areas again. (Each area is segmented into four smaller symmetrical areas). Connectivity-preserving relaxation-based segmentation methods, usually referred to as the active contour model, start with some initial boundary shape represented in the form of spline curves. This shape is iteratively modified by applying various shrink/expansion operations according to some energy function [5]. Although the energy-minimizing model is not new, coupling it with the maintenance of an “elastic” contour model gives it an interesting new twist. A common and difficult to avoid problem with such methods is getting trapped into a local minimum.

The k-means algorithm is an unsupervised method that does not require any initial data [9]. Each image pixel is initially attributed to the class with the nearest center to the pixel. The new center of each class is calculated and the algorithm is repeated. The segmentation...
process is continued until there is no considerable difference between the centers of the new and old classes. The fuzzy c-means algorithm (FCM) is similar to the k-means algorithm [6], except that the absolute dependence scheme is replaced by a relative dependence scheme. Although it is probable that a pixel belongs to any class, it is assigned to the class with the maximum probability.

Fuzzy filters are improved median filters or a rank-condition rank-selection filters [12], in which the binary spatial-rank relation is replaced by a real-valued relation. This process permits the filter to adapt to the signal spread by averaging the flat areas, while the isolated pixels in the edge areas remain.

The pixon concept, introduced by Pina and Puettter [13, 14], involves a set of disjointed regions with constant shapes and variable sizes. Their pixon scheme used a local convolution between a kernel function and a pseudo image. The drawback of this scheme is that the shape of the pixons cannot change once the kernel function is selected. However, both the shape and size of the pixons could change if the definition scheme introduced in [15] were used. This scheme utilizes the anisotropic diffusion equation to form the pixons, and the MRF concept is considered when segmenting the images. The advantage of using pixons is that the decision level is changed from pixels to pixons, which reduces the computational time due to the smaller number of pixons compared to pixels.

In the method proposed in the present paper, a fuzzy filter is used as a kernel function to form the pixons. This process preserves the edges and texture, while averaging out the noise. Because the proposed method is not based on iteration, the computation time of the segmentation algorithm is reduced. After the pixons are formed, the FCM algorithm is applied to image the segments [6]. Application of the proposed method to several standard images revealed that, by incorporating the fuzzy filter followed by the pixon concept, the computational time was decreased significantly. The proposed method excels in some criteria, such as the variance and pixon to pixel ratio, compared to other existing approaches [15, 16].

This paper is organized as follows. The pixon concept is described in Section 2. Section 3 provides background on fuzzy filtering. Section 4 introduces two different clustering algorithms. The proposed method is discussed in Section 5. Simulation results comparing the proposed algorithm to existing approaches are presented in Section 6. Finally, Section 7 gives concluding remarks.

2. Pixonal Image

Pixons are variable-sized building blocks comprised of one or more similar pixels, which locally represent the image resolution. The shape and position of all of the pixons over an image are collected in a pixon map, which gives a multiresolution description of the image with various spatial scales. Because different parts of an image often exhibit different spatial resolutions, the use of a pixon map as an adaptive scale representation is justified. The goal is to select the best spatial scale for each part of the image [13-15].

The fuzzy pixon definition scheme for astronomical image restoration and reconstruction was proposed by Pina and Puettter [13, 14]. To construct a pixonal image, we need a base image from which pixons can be extracted. This image is built with a kernel function and a pseudo image. A pseudo image is a high-resolution image obtained through interpolation of the pixels of the input image. A kernel function is applied to each pixel of the pseudo image to increase the weight of nearby pixels.

We used a two-dimensional fuzzy filter as the kernel function. Each pixel of the base image was modeled by a local convolution of the pseudo image and the kernel function:

$$I(x, y) = \sum_{\sigma_x=1}^{M} \sum_{\sigma_y=1}^{N} K(\sigma_x, \sigma_y)(r, c)I_{\text{pseudo}}(x-r, y-c)$$  (1)

where $I(x, y)$ is the row and column index of each pixel in the spatial domain, $\sigma_x$ and $\sigma_y$ are the standard deviation of the fuzzy kernel, $K$ (here the spread parameter), and $w$ is the width of the fuzzy filter.

Following Yang and Jiang [15], a pixon can be defined as follows:

$$I = \bigcup_{i=1}^{s} P_i$$  (2)

where $I$ is the pixonal image from the base image; $s$ is the number of pixons; and $P_i$ is a pixon, comprised of a set of connected pixels or even one pixel or subpixel in the original image. A subpixel is part of a pixel, [15] reminding that the original image has been interpolated to form the pseudo image. We can consider a pixel in the pixonal image to be a subpixel of the original image. The shape of each pixon may vary depending on the observed image. The construction of a pixonal image transfers the image segmentation problem from the pixel domain to the pixon domain. Therefore, $P$ pixons are evaluated, instead of $(M \times N)$ pixels ($P \ll M \times N$). These P pixons are characterized by the mean values of their corresponding pixels, such that a pixonal image consists of P pixons. The shape and position of the pixons are also collected in the pixon map.

A pixonal image could be represented by a graph-like structure, in which the pixons form a set of graph vertices, and the link between the vertices indicates a common border between the pixons. Hence, the pixonal im-
age $G$ can be represented as:

$$G=(Q,E),$$

where $Q$ is the finite set of the vertices in the graph representing the pixons, and $E$ is the set of the edges in the graph, indicating that the two connected pixons are neighbors. In the example shown in Figure 1, P1, P2, ..., P5 represent 5 pixons that form the pixonal model of an image. The pixonal model can be expressed by the graph structure illustrated in Figure 1b.

![Figure 1. Pixon structure of an image (a) and corresponding graph structure (b).](image)

The most novel use of pixon-based image segmentation was introduced by Hassanpour et al. [16]. In their method, during preprocessing, the wavelet thresholding technique is applied to smooth the image and reduce noise. The threshold value must be assigned properly to avoid over-smoothing. Then, the pixon-based algorithm is used to form and extract the pixons. Finally, the FCM algorithm is applied to segment the image.

### 3. Fuzzy Filtering

Fuzzy filters improve the median filters or rank-condition rank-selection filters [17] by replacing the binary spatial-rank relation by a real-valued relation. This filter is derived from fuzzy transformation theory [12]. In [18], a fuzzy filter was defined by generalizing the binary spatial-rank relation. Assuming that filter $h$ is applied to a set of neighboring samples $f[i+i', j+j']$ around the input $f[i,j]$, we can formulate the output:

$$g[i,j] = \sum_{i',j'} h(f[i+i', j+j'], f[i,j]) \times f[i+i', j+j']$$

and its unbiased form via normalization:

$$g[i,j] = \frac{\sum_{i',j'} h(f[i+i', j+j'], f[i,j]) f[i+i', j+j']}{\sum_{i',j'} h(f[i+i', j+j'], f[i,j])}.$$ (4)

and its unbiased form via normalization:

where $h(f[i+i', j+j'], f[i,j])$ controls the contribution of the input $f[i+i', j+j']$ to the output.

Due to the input independence of the filter coefficients, a low-pass filter designed to perform effectively in the flat areas may introduce blurring artifacts in the detailed areas. However, it is desirable to preserve the details, while removing the artifacts. This effect can be achieved by imposing constraints (e.g., “if $f[i+i', j+j']$ is far from $f[i,j]$, its contribution to the output is small”). In this case, the filter coefficients $h[i,j]$ must follow the constraints:

$$\lim_{|i+i', j+j'| \to |i,j|} h(f[i+i', j+j'], f[i,j]) = 1$$

$$\lim_{|i+i', j+j'| \to \infty} h(f[i+i', j+j'], f[i,j]) = 0$$

$$h(f[i+i', j+j'], f[i,j]) \geq h(f[i+i', j+j'], f[i,j]) \text{ if } f[i+i', j+j'] - f[i,j] \leq f[i+i', j+j'] - f[i,j].$$ (8)

The function $h(f[i+i', j+j'], f[i,j])$ is called a membership function, and there are many functions that fulfill these requirements. A Gaussian membership function is given in:

$$h(f[i+i', j+j'], f[i,j]) = \exp\left(-\frac{(f[i+i', j+j'] - f[i,j])^2}{2\sigma^2}\right),$$

where $\sigma$ represents the spread parameter of the input and controls the strength of the fuzzy filter. The input $f[i,j]$ always contributes more to the output than the other samples:

$$h(f[i,j], f[i,j]) \geq h(f[i+i', j+j'], f[i,j]) \text{ for all } k.$$ (10)

For the same $|f[i+i'+j'+j'] - f[i,j]|$, the higher the value of $\sigma$, the higher the contribution of $f[i+i', j+j']$ to the output. This relationship implies that $f[i,j]$ will converge more towards $f[i+i', j+j']$. Smaller values of $\sigma$ will keep the signal $f[i,j]$ more isolated from its neighboring samples. The spread parameter should be adaptive to different areas with different activity levels, such as smooth or detailed textures. The conventional fuzzy filter uses fixed spread parameters for every surrounding sample, ignoring their relative positions. More information about fuzzy filters is available in [17-19].

### 4. Reviewing Clustering Algorithms

In this paper, we used fuzzy filtering as the kernel function to form pixons and a clustering algorithm (FCM) to segment the image. Different clustering algorithms to segment images have been described in the literature. Two of these algorithms, the k-means and fuzzy c-mean clustering algorithms, are briefly described here.

#### 4.1 K-Means Clustering Algorithm

In statistics and data mining, k-means clustering is a method of cluster analysis that aims to partition $n$ observations into $k$ clusters, in which each observation belongs to the cluster with the nearest mean [7]. The K-means algorithm is an iterative technique that is used to segment an image into $K$ clusters. The basic algorithm is:

1. Pick $K$ cluster centers, either manually, randomly, or based on some heuristic;
2. Assign each pixel in the image to the cluster that minimizes the distance between the pixel and the cluster center;
3. Recomputed the cluster centers by averaging all of the pixels in the cluster;
4. Repeat steps 2 and 3 until convergence is attained (e.g. no pixels change clusters).

In this case, distance is the squared or absolute difference between a pixel and a cluster center. The difference is typically based on pixel color, intensity, texture, location, or a weighted combination of these factors.

4.2 Fuzzy C-Means Algorithm

FCM is a method of clustering that allows one piece of data to belong to two or more clusters. This method is frequently used in pattern recognition and signal processing [8-10]. It is based on minimization of the following objective function:

$$error = \sum_{i=1}^{K} \sum_{j=1}^{N} u_{ik}^m d_{ik}^2 = \sum_{i=1}^{K} \sum_{j=1}^{N} u_{ik} \| x_i - \mu_k \|$$  \hspace{1cm} (11)

where $m$ is any real number $> 1$; $u_{ik}$ is the degree of membership of $x_i$ in the cluster $k$; $x_i$ is the $i$th element of d-dimensional measured data; $\mu_k$ is the center of the cluster with $d$ dimensions (for images $d = 2$); and $\| \cdot \|$ is any norm expressing the similarity between any measured data and the center $\mu_k$.

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership $u_{ik}$ and the cluster centers $\mu_k$ by:

$$u_{ik} = \frac{1}{\sum_{j=1}^{K} (\frac{d_{ik}}{d_{ij}})^{m-1}}$$  \hspace{1cm} (12)

$$\mu_k = \frac{\sum_{i=1}^{N} u_{ik} x_i}{\sum_{i=1}^{N} u_{ik}}$$  \hspace{1cm} (13)

This iteration will stop when $\max_{ik} \| u_{ik}^{(n+1)} - u_{ik}^{(n)} \| < \varepsilon$, where $\varepsilon$ is a termination criterion between 0 and 1 and $n$ is the number of iterations. This procedure converges to a local minimum or a saddle point of error.

The algorithm is composed of the following steps [10, 11]:
1. Initialize $U = [u_{ik}]$ matrix, $U^{(0)} U(0)$
2. At $n$-step: calculate the centers vectors $\mu^{(n)} = [\mu_k]$ with $U^{(n)}$

$$\mu_k = \frac{\sum_{i=1}^{N} u_{ik} x_i}{\sum_{j=1}^{N} u_{ik}}$$  \hspace{1cm} (14)

3. Update $U^{(n)}$, $U^{(n+1)}$

$$u_{ik} = \frac{1}{\sum_{j=1}^{K} (\frac{d_{ik}}{d_{ij}})^{m-1}}$$  \hspace{1cm} (15)

4. If $\| U^{(n+1)} - U^{(n)} \| < \varepsilon$ then STOP; otherwise, return to step 2.

5. Proposed Algorithm

In this paper, a novel pixon-based method is proposed for image segmentation. In the pixon-based algorithm proposed by Yang and Jiang [15] (hereafter referred to as “Yang’s algorithm”), after the pseudo image was obtained, the anisotropic diffusion equation was used as the kernel function to form the pixons. Hassanpour et al. [16] (hereafter referred to as “Hassanpour’s algorithm”) used wavelet thresholding as the kernel function. In the proposed algorithm, the fuzzy filter [12] was successfully used to improve the quality of the image and to prepare it to form pixons. Utilizing the fuzzy filter eliminated some unnecessary details and resulted in a smaller number of pixons, faster performance, and better robustness against unwanted noise. After forming and extracting pixons, the fuzzy clustering algorithm was used to partition and segment the image.

The proposed algorithm can be described as follow:
1. **Build a pseudo image with maximum similarity to the original image.** The pseudo image is built from the original image and has higher resolution. If the dimensionality of the original image is $D_x \times D_y$, then the dimension of the pseudo image is $ID_x \times ID_y$, where $l = 2^n$ [15]. If $n \geq 1$, then the resolution of the pseudo image is larger than that of the original image. The parameter $n$ also determines the smallest size of the pixons. When the intensities of close pixels in the image are similar (i.e., provide little information), the intensity of newly inserted pixels are similar to the intensities of the pixels in the observed image, from which the new pixels are obtained through interpolation.

2. **Convolute the pseudo image with a kernel function.** In this step, the fuzzy filter [18] is applied to the pseudo image according to (1), and the pixon image is formed.

3. **Use the k-means algorithm [15] to extract pixons, based on hierarchical clustering.** Initially, each pixel represents a cluster. The clusters are merged according to their intensities to form larger pixons. The mean value of the connected pixels comprising the pixon is defined as the pixon intensity, which forms the pixon map (See Fig. 1). To stop the algorithm, a threshold value $T$ is assigned ($T = 5$ in the present
study). The merging process iterates until the difference between intensities of two adjacent pixons is greater than the threshold value.

4. **Segment the pixon image with the FCM algorithm** [6, 8].

6. **Evaluation of Proposed Algorithm**

Simulations were performed to demonstrate the effectiveness of the proposed algorithm compared to Yang’s [15] and Hassanpour’s [16] algorithms. The proposed algorithm was applied to several standard images [20]. The visual quality, number of the pixons in image, pixon to pixel ratio, normalized variance, and computational time obtained with the different methods were compared. These measurements are described as follows:

a) **Computational time**: In most applications, the time required to perform an algorithm is an important parameter that researchers seek to minimize.

b) **Number of pixons and pixon to pixel ratio** [15]: After pixon creation, the pixons must be labeled. The computational time is minimized by decreasing the number of pixons and the pixon to pixel ratio. We should ensure that the image details are not eliminated in this process.

c) **Variance and normalized variance** [16]: Segment variance is one of the most important parameters used to evaluate the performance of image segmentation methods. Smaller variances imply more homogeneity of the region and, consequently, better segmentation results. Assume that, after the segmentation process, the images are divided into $K$ segments (“classes”) with different average values. If $N_i$ and $V(k)$ denote the number of the pixels and variance of each class, respectively, then the normalized variance of each image can be determined by:

$$V^* = \frac{V_i}{V_s},$$

where

$$V_i = \sum_{k=1}^{K} \frac{N_i V(k)}{N},$$

and

$$V_s = \frac{\sum_{x,y} (I(x,y) - M)^2}{N},$$

In the above equations, $k$ denotes the number of classes, $I(x,y)$ is the gray level intensity, and $M$ and $N$ are the average value and number of pixels in each image, respectively.

In first step, the proposed algorithm was applied to noisy images. Gaussian noise with different SNRs (ranging from 1 to 15 dB) was added to each image (same in Figure 2b). The proposed method or Yang’s or Hassanpour’s algorithm was applied to the images (Figure 2c–e). The proposed algorithm was robust to noise (Figure 2e). The number of pixons and the pixon to pixel ratio were significantly less when the proposed algorithm was used, compared to when Yang’s or Hassanpour’s algorithm was used (Table 1), due to application of the fuzzy filter technique before forming pixons. Use of the pixon concept with the fuzzy method also reduced the computational cost (Table 2).

Figure 2. Experimental results on a noisy image. Shown are the original image (a), noisy image (mean = 0, variance = 0.01) (b), and segmented results obtained with Hassanpour’s method (c), Lin’s method (d), and the proposed method (e).

Figure 3. Segmentation results of the “pepper” image. Shown are the original image (a) and segmentation results with Yang’s (b), Hassanpour’s (c), and the proposed method (d).
Table 1. Comparison of the number of pixons and ratio between the number of pixons and pixels among the three methods.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Babon (256×256)</td>
<td>65536</td>
<td>36125</td>
<td>27561</td>
<td>19564</td>
<td>64%</td>
<td>43%</td>
<td>28%</td>
</tr>
<tr>
<td>Lena (256×256)</td>
<td>65536</td>
<td>32151</td>
<td>24552</td>
<td>17530</td>
<td>49%</td>
<td>37%</td>
<td>26%</td>
</tr>
<tr>
<td>Peppers</td>
<td>65536</td>
<td>28126</td>
<td>25161</td>
<td>20541</td>
<td>42%</td>
<td>38%</td>
<td>31%</td>
</tr>
<tr>
<td>Cortex (128×128)</td>
<td>16384</td>
<td>1814</td>
<td>1651</td>
<td>1210</td>
<td>11.1%</td>
<td>10%</td>
<td>7%</td>
</tr>
</tbody>
</table>

Table 2. Comparison of computational time (in ms) between different algorithms.

<table>
<thead>
<tr>
<th>Image</th>
<th>Yang’s</th>
<th>Hassanpour’s</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Babon</td>
<td>18549</td>
<td>15316</td>
<td>12321</td>
</tr>
<tr>
<td>Lena</td>
<td>15143</td>
<td>13066</td>
<td>10526</td>
</tr>
<tr>
<td>Peppers</td>
<td>1542</td>
<td>11254</td>
<td>9512</td>
</tr>
<tr>
<td>Cortex</td>
<td>702</td>
<td>633</td>
<td>510</td>
</tr>
</tbody>
</table>

Table 3. Variance values of each class with different algorithms for the “Woman” image.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang’s</td>
<td>Average</td>
<td>175.47</td>
<td>125.24</td>
<td>72.08</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>18.33</td>
<td>14.50</td>
<td>21.36</td>
</tr>
<tr>
<td>Hassanpour’s</td>
<td>Normalized Variance</td>
<td>0.1740</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>169.76</td>
<td>124.84</td>
<td>77.80</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>12.64</td>
<td>15.31</td>
<td>18.86</td>
</tr>
<tr>
<td>Proposed</td>
<td>Normalized Variance</td>
<td>0.1658</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>164.75</td>
<td>127.56</td>
<td>89.55</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>12.31</td>
<td>10.11</td>
<td>15.19</td>
</tr>
</tbody>
</table>

Table 4. Variance values of each class with different algorithms for the “pepper” image.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang’s</td>
<td>Average</td>
<td>122.38</td>
<td>196.37</td>
<td>32.70</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>25.97</td>
<td>21.35</td>
<td>22.86</td>
</tr>
<tr>
<td>Hassanpour’s</td>
<td>Normalized Variance</td>
<td>0.0911</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>124.79</td>
<td>192.78</td>
<td>35.06</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>24.32</td>
<td>18.36</td>
<td>22.41</td>
</tr>
<tr>
<td>Proposed</td>
<td>Normalized Variance</td>
<td>0.0822</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>129.17</td>
<td>194.12</td>
<td>37.48</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>23.77</td>
<td>17.37</td>
<td>20.52</td>
</tr>
</tbody>
</table>

Table 5. Variance values of each class with different algorithms for the “cortex” image.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang’s</td>
<td>Average</td>
<td>133.84</td>
<td>180.78</td>
<td>72.51</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>15.41</td>
<td>16.88</td>
<td>24.74</td>
</tr>
<tr>
<td>Hassanpour’s</td>
<td>Normalized Variance</td>
<td>0.1652</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>135.28</td>
<td>174.93</td>
<td>77.91</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>13.59</td>
<td>13.05</td>
<td>21.97</td>
</tr>
<tr>
<td>Proposed</td>
<td>Normalized Variance</td>
<td>0.1486</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>135.38</td>
<td>168.63</td>
<td>83.44</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>11.84</td>
<td>10.56</td>
<td>15.81</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Normalized Variance</th>
<th>0.1240</th>
<th></th>
<th></th>
</tr>
</thead>
</table>
After the segmentation process, images were divided into three segments/classes and the variance and average of each class were determined (Tables 3-5). In most cases, the variance of the classes of different images were smaller with the proposed method compared to the other methods. To investigate the performance of the methods more exactly, the normalized variance of each image was calculated after applying each method. In the pixon-based approach that used fuzzy filtering, the pixels in each cluster were closer to each other and the areas of the images were more homogenous.

We also examined the results of applying Yang’s, Hassanpour’s, or the proposed method on several standard images: i.e., images of a pepper (Figure 3a), woman (Figure 4a), and cortex (Figure 5a). The homogeneity of the regions and discontinuity between adjacent regions, which are two main criteria used in image segmentation, were enhanced when the proposed fuzzy-based method was used, compared to the other methods (Figures 3-5).

### 7. Conclusion

An improved pixon-based method was proposed for use in image segmentation. The proposed algorithm uses the fuzzy filtering technique as a kernel function to form a pixonal image. Both the smoothness and uniformity of the obtained image were increased compared to the original image by using the fuzzy filter. By incorporating the pixon concept and fuzzy technique into our method, a considerable decrease in computational costs was achieved. Image segmentation criteria were greatly enhanced when the proposed method was used, compared to two other well-known methods, and good segmentation results were obtained.

### References


Amin Barari is researcher in the Department of Civil Engineering at Aalborg University, Denmark. His current research interests include offshore foundations, slope stability and landslides, soil improvement, and fuzzy sets. He is also executive editor and founder of Mechanical Sciences which is being published by Copernicus.

Ehsan Nadernejad is PhD student in Department of Photonics Engineering at Technical University of Denmark. His research interests include signal and image processing, video coding, backlight dimming algorithm and video post-processing.