A Novel Edge Detection Approach Based on Backtracking Search Optimization Algorithm (BSA) Clustering

Mehmet Akif Günen, Ümit Haluk Atasever, Erkan Beşdok
Department of Geomatics Engineering Erciyes University, Kayseri, TURKEY
{akif, uhatasever, ebesdok} @ erciyes.edu.tr

Abstract—Image edge information is very important in application areas such as machine learning, image processing, stereo vision, object tracking and pattern recognition. Intensity discontinuities or sudden intensity changes in a region are indicative of the edge region in that region. Although there are many approaches to detecting edge, generally intensity discontinuities or sudden intensity changes in a region are described as edge. In this study, we proposed a Backtracking Search (BSA) clustering based edge detection approach for noisy images. Proposed approach has two stages. In first stage, the edge map is calculated using the max-min filter defined in a window. In second stage, edge map is calculated via BSA based clustering with using a cost function.

Keywords—Edge detection, BSA Clustering, Edge Map

I. INTRODUCTION

Edge detection is one of the most important application areas in computer vision. The effectiveness of edge detection is a key factor affecting the correctly working of many autonomous systems. Correct positioning of the edges, good detection of edge points, and low false edge detection rate are key features expected from edge detection algorithms. Among these algorithms, the most commonly used methods are clustering / threshold values and gradient based methods[1, 2]. Although conventional methods such as Sobel, Canny, Prewitt and Roberts succeed in edge detection, they cannot produce the desired quality in noise-containing images and cannot distinguish edge-texture. This disadvantage causes users to get false results in achieving true edge detection of conventional methods. The advantages and disadvantages of classical methods can be examined in related articles [3, 4]. Contrary to classical methods, artificial intelligence tools are more robust and flexible, requiring more time and more calculation load. Recently, several edge detection methods have been proposed using various methods such as self-organizing map (SOM), neural network, k-means and Fuzzy C-Means, which operate on a clustering basis, have been used in applications such as edge detection, edge extraction and edge separation which outperformed classical edge detection filters at various aspects[5] [5]. In this paper, we proposed Backtracking Search (BSA) clustering based edge detection approach for edge detection. In this context, the edge map is created and the edge is detected by using BSA.

II. EDGE MAP

The intensities of the pixels within a window have values that are consistent and close to each other. However, the maximum-minimum values can be change in a large scale at edge points and noisy areas. In the proposed method, there are two main stages. First stage is edge map creation for edge detection. In this stage, both minimum and maximum filter are used because of effectiveness in noise reduction and success in determining gradient-level variation relative to the center pixel neighborhood. The maximum filter selects the largest value $I_{max}$ in the selected window, while the minimum filter selects the smallest value $I_{min}$. After that edge map is calculated according to (1).

$$K = I_{max} - I_{min}$$  \hspace{1cm} (1)

The center pixel is updated in sliding window to retrieve the new values and thus an edge map is obtained. The edge map used maintains high spatial frequency values in addition to producing fast, simple and highly effective results. Since the smallest symmetric window around the center pixel is 3x3, 3x3 window is preferred in practice and local intensity values are examined. These windows contain more information in determining the direction of the edge to add to the quality of data structure among the neighbors in the opposite directions of the center pixel. The edge map images obtained from three random scenes from Berkeley Segmentation Dataset are shown in Fig. 1.
III. CLASSICAL METHODS

Many edge detection algorithms work with local derivative calculations. A first-order derivative of an image, in other words, has a local maximum value at edge regions. In the same way second-order derivative of image produce zero value in edge regions. By applying the first and second derivative to an image at local regions, the edges of the image are determined by generating zero values at the local maximum and transition points. The gradient of an image is expressed as the partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ of each pixel of the image. The partial derivative is shown in (2-3).

$$g_x = \frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x,y)$$  \hspace{1cm} (2)

$$g_y = \frac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y)$$  \hspace{1cm} (3)

The magnitude of the change in one direction of the gradient vector is given in Eq. (4) [6].

$$\text{Mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$  \hspace{1cm} (4)

In this article, Sobel, Prewitt and Roberts edge estimators, which are non-directional edge estimators, as well as Canny operator based on 2nd derivative, are compared with the proposed method.

A. ROBERTS OPERATOR

One of the commonly used operators to detect diagonal edges is Roberts. Roberts operator is based on the method of obtaining diagonal differences. The Roberts operator with the 2x2 kernel size is an operator that is very fast due to the small size but very quickly affected by the noise. If the image does not contain very extreme sharp edges, it cannot produce quality results. It has two different kernels horizontally and vertically. These two kernels are convolved on the image in order to determine the edge. Fig. 2 shows the mask used by the Roberts Operator.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
</tbody>
</table>

Fig. 2. a) $g_x$ orientation b) $g_y$ orientation

B. SOBEL OPERATOR

The Sobel operator, which consists of a 3x3 convolution kernel, is designed to detect the vertical and horizontal edges of the pixel grid. In Sobel operator, near-center pixels are weighted with bigger numerical values despite of other pixels. Fig. 3 shows the mask used by the Sobel Operator.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>-2</td>
<td>0</td>
<td>+2</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

Fig. 3. a) $g_x$ orientation b) $g_y$ orientation

C. PREWITT OPERATOR

The results produced by the Prewitt operator are not isotropic, although they have close working procedure as the Sobel operator. The result that occurs when this operator applies each pixel in the image is either a gradient vector or norm of this vector. The Prewitt operator is applied both horizontally and vertically as the other operators. The gradient approach at high frequency variations produces rough results [7]. Fig. 4 shows the mask used by the Sobel Operator.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

Fig. 4. a) $g_x$ orientation b) $g_y$ orientation
D. CANNY OPERATOR

The above-mentioned operators produce results by convolving a kernel onto the image, but the Canny edge detection operator has different structure. The application of the Canny operator to achieve edge detection by looking at the local maxima of the gradient image is made up of the following stages.

1. By means of Gaussian filter with (5), it is removed from the noise,

\[ G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \]  

(5)

2. The image intensity gradient is obtained by (6-7).

\[ G_x = \frac{-x}{\sigma^2} G(x, y) \]  

(6)

\[ G_y = \frac{-y}{\sigma^2} G(x, y) \]  

(7)

3. The image gradient magnitude is calculated by (8-12).

\[ I_x = I \ast G_x \]  

(8)

\[ I_y = I \ast G_y \]  

(9)

\[ G_x = \frac{-x}{\sigma^2} G(x, y) \]  

(10)

\[ G_y = \frac{-y}{\sigma^2} G(x, y) \]  

(11)

\[ ||\nabla|| = \sqrt{I_x^2 + I_y^2} \]  

(12)

If the gradient value of the pixel is greater than the threshold value, this pixel is determined as the strong edge pixel. To find the local maxima value of the gradient magnitude value, a non-maxima suppression algorithm with two threshold values is applied. This allows the edge to detect edges that are not affected by noise and that are weak on the image for true edge detection [8].

IV. PROPOSED METHOD

A. BSA

Artificial intelligence optimization algorithms are the basic and effective techniques used to solve the problems of applied mathematics and engineering problems [9, 10]. Until recently, algorithms such as Genetic Algorithm (GE), Differential Evolution (DE) and derivatives, Artificial Bee Colony Algorithm (ABC) have been used frequently, but the literature has begun to be more effective and robust algorithms. [9]. Backtracking Search Algorithm (BSA) developed by Çivicioglu is one of these algorithms. In addition to having a fairly simple and easy adoptable structure, BSA is remarkable for its robustness and fast convergence speed [11]. Moreover, both the number of control parameters is less than that of similar techniques and the fact that it produces more successful results than popular algorithms is also an element that increases the appeal of the algorithm. BSA consists of 4 steps, Selection-1, Mutation, Crossover and Selection-2 [9]. The pseudo code of BSA is given in the Fig.5.

B. CLUSTERING WITH BSA

Clustering can be defined as the automatic detection of natural groups in a data collection without the need for any learning process [12-14]. Whatever the purpose of application or data set, the preferred clustering technique is one of the most important steps that affect the accuracy of the process [15]. In this phase, the attribute set obtained in the previous steps is clustered into two classes as edge and non-edge region. Although K-means and derivatives, Fuzzy-C means and density-based techniques for unsupervised classification are used frequently, artificial intelligence optimization algorithms are also attracting attention due to their recent success.[14, 16-23]. For this reason, the clustering phase of the proposed approach was performed with BSA. In order to clustering to be done in the correct way, the design of the optimization cost function must be appropriate. For this reason, a minimization cost function is used for clustering, which is given in (13-14) and has been used in many previous studies [11, 15].

\[ p_j = \{x_i; \min ||x_i - \text{cent}_j||\} \]  

(13)

\[ \arg \min_{\text{cent}_j} \sum ||p_j - \text{cent}_j|| \]  

(14)

In Eq. (13, 14) \( ||\cdot|| \) is Euclidean distance, \( x_i \) is i-th pixel, \( \text{cent}_j \) is the new center of the j-th class and \( p_j \) represents all of the pixels in the j-th class. As seen in Eq. (13, 14) BSA...
updates and optimizes the cluster centers according to total distance between the cluster centers and the pixels of each cluster in each iteration [11].

The label values for the edge and non-edge classes may change from the simulation to another simulation because BSA randomly selects the cluster centers and uses random number generators in the heuristic steps. To solve this problem, (15), a label correction operation is performed.

$$\text{EdM} = \begin{cases} EdM & m_1 = 2 \text{ and } m_2 = 1 \\ (EdM \times (-1) + 2) & m_1 = 1 \text{ and } m_2 = 2 \end{cases}$$ (15)

Where EdM, $m_1$ and $m_2$ edge map, label values of edge and non-edge are, respectively.

V. EXPERIMENT RESULTS

Several gray images were used in the dataset to show the efficiency and usability of the proposed method compared to Sobel, Canny, Prewitt and Roberts operators. The algorithms used in the application are developed with MATLAB software. In the proposed method, BSA parameters popsize=15, dim=2, 1000 iterations as epoch value were selected. These parameters can be changed according to pixel discontinuity, image size and various noise types and parameters to get better results. In classical methods, parameters are selected by default in MATLAB.

A. Dataset

The 8 bit grayscale images used for the application were randomly selected from the Berkeley Segmentation dataset. The Berkeley Segmentation dataset is quite useful and helpful because it contains thousands of images such as nature, people, building and animals, and Ground Truth (GT) images belonging to these images. Lena, Saturn, Coins, Cameraman and Peppers image sets are used for visual comparison as well as the Berkeley Segmentation dataset used for statistical comparison. 10% and 20% Salt & Peppers noise was added to images to show effectiveness of proposed approach. Fig. 6 shows randomly selected images from the Berkeley Segmentation dataset.

B. Statistical Analysis

There are various statistical comparison methods such as Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Figure of Merit (FoM), L2RAT, Mean Structural Similarity Index Metrics (MSSIM) and Running Time [26]. PSNR, FoM and L2RAT statistical comparison methods were used in this study. It is assumed that the MSE value is obtained when calculating the PSNR value and produces parallel results with the PSNR and has not been given.
1. **Pratt’s Figure of Merit (FoM)**

The FoM value, one of the most commonly used methods for edge detection performance comparison, is given in (16).

\[ \text{FoM} = \frac{1}{\max(I_i, I_d)} \sum_{i} \frac{1}{1 + \tau d_i^2} \]  \hspace{1cm} (16)

\( I_i \) and \( I_d \) are the number of ideal pixels and detected edge pixels, respectively. \( d_i \) is the Euclidean distance between the detected pixel and the nearest ideal (GT) image. The \( \tau \) value is a fine constant and is chosen to be 0.1 in practice. The FoM value produces a value in the range [0,1] [27]. In practice, FoM scaled to [0,100] to make the results more detailed. After extracting the edge map of randomly selected image from Berkeley Segmentation dataset, Salt & Peppers noise was added at different values. Then the FoM value was calculated using the GT image of the same image.

2. **Peak Signal to Noise Ratio (PSNR)**

Another statistical method used to compare edge detection performance is the PSNR value. Generating a PSNR value higher than 30 dB is usually an undesirable result. However, a high PSNR value and a low MSE value are desirable results. MSE values are shown in (17).

\[ \text{MSE} = \frac{1}{MN} \sum_{i} \sum_{j} (x_{(i,j)} - x'_{(i,j)})^2 \]  \hspace{1cm} (17)

\( x_{(i,j)} \) and \( x'_{(i,j)} \) are the image coordinates of edge map and GT image, respectively [28, 29]. After the MSE is calculated, the PSNR value is obtained by (18).

\[ \text{PSNR} = 20 \log_{10} \left( \frac{2^{n-1}}{\sqrt{\text{MSE}}} \right) \]  \hspace{1cm} (18)

3. **L2RAT**

Another method used for edge detection performance comparison is L2RAT. L2RAT is squared energy of image approximation to the input image. Image approximation and input image must be same size [30]. In (19) the mathematical expression of the method is shown.

\[ \text{L2RAT} = \frac{\| x_{\text{app}} \|}{\| x_{i} \|} \]  \hspace{1cm} (19)

The PSNR, FoM, and L2RAT values of the randomly selected images from the Berkeley Segmentation dataset are given in Table 1 for success comparison. Image numbers are given in the left column. According to Table 1, proposed approach gives more better results than classical methods; Canny, Sobel, Prewitt and Roberts. In all tests except for the application with image no: 253036, best results were obtained. It is seen that as the amount of noise increases, the value of PSNR and FoM increases and the value of L2RAT decreases.

### Table 1. Statistical Analysis of Berkeley Segmentation Dataset

<table>
<thead>
<tr>
<th>Image No</th>
<th>Proposed</th>
<th>Canny</th>
<th>Roberts</th>
<th>Prewitt</th>
<th>Sobel</th>
</tr>
</thead>
<tbody>
<tr>
<td>253036</td>
<td>21.4</td>
<td>18.6</td>
<td>18.2</td>
<td>18.8</td>
<td>18.4</td>
</tr>
<tr>
<td>253037</td>
<td>21.7</td>
<td>18.9</td>
<td>18.5</td>
<td>18.9</td>
<td>18.5</td>
</tr>
<tr>
<td>253038</td>
<td>21.9</td>
<td>19.1</td>
<td>18.7</td>
<td>19.1</td>
<td>18.7</td>
</tr>
<tr>
<td>253039</td>
<td>22.1</td>
<td>19.3</td>
<td>19.0</td>
<td>19.3</td>
<td>19.0</td>
</tr>
</tbody>
</table>

### C. Visual comparison

Visual comparison of the performance results of the Saturn, Coins, Peppers, Cameraman and Lena test images with Proposed, Canny, Prewitt, Roberts and Sobel operators in Fig. 8. Morphological thinning was applied to the image obtained by the proposed method. The proposed method produces more effective results in terms of detecting many edges that other methods cannot detect and being much less affected by noise. No noise was added in terms of visually comparing the images success shown in Fig. 8.
VI. CONCLUSION

Edge detection is one of the most important application areas in computer vision. In this paper, a new edge detection algorithm based on BSA clustering using local intensity change is presented. To evaluate effectiveness of proposed approach, images obtained from Berkeley Segmentation dataset has been used. The dataset includes ground truth images and test images. The design of the edge map has an important influence on the edge quality. In order to obtain edge map, min-max filter has been applied. The statistical (PSNR, FoM and L2RAT) and visual results obtained by the proposed method show more successful results compared with the classical methods. BSA based edge detection yields better results even though these statistical results vary according to the modified GT images.

ACKNOWLEDGMENT

This work was supported by The Scientific and Technological Council of Turkey (TUBITAK) under Grant [115Y235].

REFERENCES


