ICIC Express Letters
Part B: Applications
Volume **7**, Number **3**, March **2016**

ICIC International ©2016 ISSN 2185-2766
pp. **519–524**

EFFICIENT EDGE DETECTION USING MEAN SHIFT SMOOTHING AND GRADIENT-BASED NOISE REMOVAL

Haitao Sang[1,2,*] and Zhen Zhou[1]

[1]College of Measurement and Communication Engineering
Harbin University of Science and Technology
No. 52, Xuefu Road, Harbin 150001, P. R. China
*Corresponding author: sanghaitao79@163.com

[2]College of Computer and Information Engineering
Heilongjiang University of Science and Technology
No. 2468, Tangchang Street, Songbei District, Harbin 150027, P. R. China

Received September 2015; accepted December 2015

Abstract. *For the low-contrast edge and slow calculation speed of conventional algorithms for edge detection, this paper proposes an efficient method of edge detection using mean shift smoothing and gradient-based noise removal. The mean shift smoothing could efficiently protect the edge details and amplify the contrast along image edges to give more clear contours in edge detection. Then an improved method based on the eight neighborhood gradient magnitude is designed to overcome the disadvantages of being sensitive to noise in the calculation of the traditional operators gradient. Finally, a comparative study of different edge detection techniques is tested and analyzed, which proves the effectiveness of our method such as continuity, smoothness, noise removal and execution time.*

Keywords: Edge detection, Mean shift smoothing, Noise removal, Eight neighborhood

1. **Introduction.** The edge is the basic characteristic of the image and there is a lot of typical information of the image in the edge. Edge detection is to extract the characteristics of discrete parts by the difference in the image characteristics of the object, and then to determine the image area according to the closed edge, which may help for image segmentation, image compression, image match, image understanding and so on. This applies not only in computer vision and image processing applications but also in human image understanding. For this reason, edge detection plays a very important role in the fundamentals of images. So, there is a need for design and implementation of an efficient edge detector.

There are many methods to be developed to make edge detection. Initially the Canny edge detector [1] is widely used in computer vision and medical imaging to locate sharp intensity changes and to find object boundaries in an image. To improve accuracy of detection, Zhao et al. [2] propose an algorithm for detection based on mathematical morphology, which proves more efficient than the traditional Canny operator. Similarly, Zhe et al. [3] present an improved Canny algorithm for edge detection instead of the double-threshold used in classic Canny algorithm. Song et al. [4] remove noise by smoothing image; however, they only employ mean shift clustering method and do not consider the information of gradient magnitude. Later, some advanced techniques based on soft computing like fuzzy logic [5,6] give better results than the conventional techniques but take longer execution time. For the slow calculation speed, Kang and Xu [7] give an adaptive edge-detection method based on Prewitt algorithm, and speed up the edge-detection calculation and get more smooth edge. Ray et al. [8] propose the simplified novel method to reduce the execution time; however, their results have more blunt edges. Then Patel et al. [9] give a novel algorithm in which traditional approaches of Sobel and Lapalacian of Gaussian (LoG) [10] are used with fuzzy system and fuzzy rules. Joshi and Choubey [11] present soft computing approach to represent a good mathematical framework to deal with uncertainty of information. They overcome the limitation of wavelet and mathematical morphology based edge detection. More recently, Sadiq et al. [12] present an improved edge detection algorithm for facial images using vector order statistics. The developed algorithm processes colored images directly without been converted to grey scale. Using adaptive thresholding, Tirath and Yogendra [13] design an improved simplified novel method for edge detection in grayscale images, but a drawback of the proposed technique is that it fails to detect edges in images with lots of details and noise.

Our paper differs in that we deduce mean shift smoothing and design its implementation algorithm to efficiently protect the edge details and amplify the contrast along image edges to give more clear contours. Also we evaluate the gradient magnitude in the eight neighborhood to locate the edge precisely and remove noise efficiently. Finally we give a comparative study of different edge detection techniques like Prewitt and Canny with the proposed method.

The rest of this paper is organized as follows. The derivation of mean shift smoothing is given in Section 2. Then noise removal method is designed in Section 3. Finally, experimental results are discussed in Section 4, and conclusions are drawn in Section 5.

2. **Derivation of Mean Shift Smoothing.** Given n data points $x_i \in R^d$, the multivariate kernel density estimate using a radially symmetric kernel $K(x)$, is given by

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{1}$$

where n is number of data points. d is number of dimensions. h (termed the bandwidth parameter) defines the radius of kernel, which is known as the *window radius*. This parameter has a direct influence on the resolution of the cluster detection by the algorithm, which will be shown in the following part.

For our discussion, we limit our kernel function to the Epanechnikov kernel as follows:

$$K_E(x) = \begin{cases} (d+2)\left(1 - x^T x\right)/(2c_d); & \text{if } x^T x < 1 \\ 0; & \text{otherwise} \end{cases} \tag{2}$$

where c_d is the volume of a unit d-dimensional sphere. A qualitative presentation of the density estimate function $f(x)$ is given in Figure 1, for a random 2-dimensional data set.

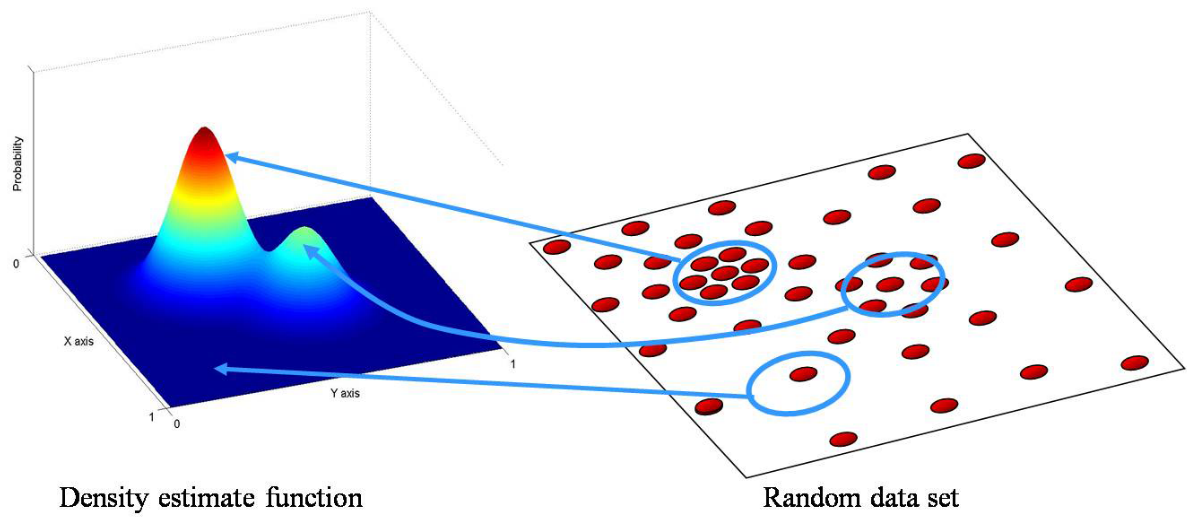


FIGURE 1. Density estimate function for a random data set

Since the Epanechnikov kernel $K_E(x)$ is differentiable, we can derive the mean shift vector based on the gradient of $f(x)$:

$$M_h(x) = \frac{h^2}{d+2}\frac{\nabla f(x)}{f(x)} = \frac{1}{n_x}\sum_{x_i \in S_h(x)} (x_i - x) \tag{3}$$

where the region $S_h(x)$ is a d-dimensional sphere of radius h, having the volume $h^d c_d$, centered at x, and containing n_x data points. The mean shift vector has the direction of the density estimate's gradient at x, when obtained with the Epanechnikov kernel. Since it always points towards the maximum increase of density, it can define a path leading to a local density maximum.

The mean shift procedure can be obtained by successive computation of the mean shift vector $M_h(x)$ and translation of the window $S_h(x)$ by $M_h(x)$. A series of successive iterations of the mean shift procedure are guaranteed to converge which can be expressed as

$$y_{k+1} = \frac{1}{n_k}\sum_{x_i \in S_h(y_k)} x_i \tag{4}$$

In the $(k+1)'$th iteration, we shift the current location by the mean position of all data points contained within the sphere of radius h, centered at y_k.

Given an image map $I(i,j)$, a set of data points can be constructed by simply assigning each pixel's location in the map as the first two coordinates, and setting the third coordinate to be the normalized value of the pixel's intensity: for the $(i,j)'$th pixel of the image, the corresponding data point will be $(i,j,I(i,j)*c)$. So the smoothing algorithm based on the mean shift procedure is as follows.

Algorithm 1 Smoothing algorithm based on mean shift procedure

1: Evaluate the normalization $c = \frac{height+width}{2} \times \frac{1}{255}$;
2: Initialize data set $I(i,j) \to (i,j,I(i,j)*c)$;
3: For each $j = 1,\dots,n$
4: Initialize $k = 1$ and $y_k = x_j$;
5: Repeat: compute y_{k+1} using the **mean shift procedure**;
6: $k \leftarrow k+1$;
7: Until convergence;
8: Assign $I_{smoothed}(x_j(1), x_j(2)) = y_k(3)$.

3. **Noise Removal Based on Eight-Neighborhood Gradient Magnitude.** For the noise is high-frequency information, we can represent noise efficiently by using gradient analysis on the image. Based on the plurality of orientations, we can select the most suitable orientation for estimating noise. Traditional detection operators calculate the gradient magnitude in the neighborhood of low pixel and the gradient directions are horizontal, vertical, left diagonal and right diagonal zones. These methods are more sensitive to noise. We present a gradient magnitude in the 8 neighborhood, which considers the brightness value of eight neighboring pixels around the checked pixel (i,j), as shown in Figure 2. Using this method, we can further reduce noise and precisely locate the edge.

Based on 8 neighborhood partition, firstly, we calculate the first order partial derivatives of x direction and y direction as follows:

$$G_x(i,j) = \frac{I(i,j+1) - I(i,j-1) + I(i-1,j+1) - I(i-1,j-1) + I(i+1,j+1) - I(i+1,j-1)}{2} \tag{5}$$

$$G_y(i,j) = \frac{I(i+1,j) - I(i-1,j) + I(i+1,j-1) - I(i-1,j-1) + I(i+1,j+1) - I(i-1,j+1)}{2} \tag{6}$$

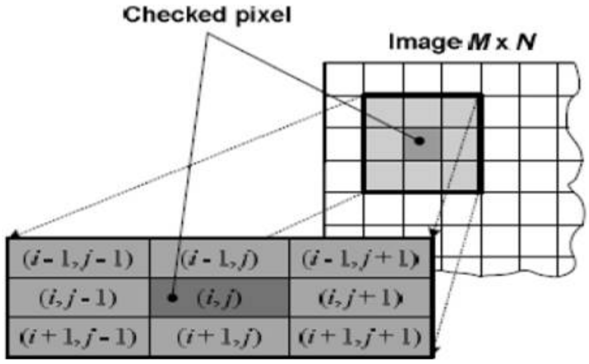


FIGURE 2. Mask 3×3 pixels in our edge detection methodology

Then the important quantity is the direction of the gradient vector, which can be evaluated as follows:

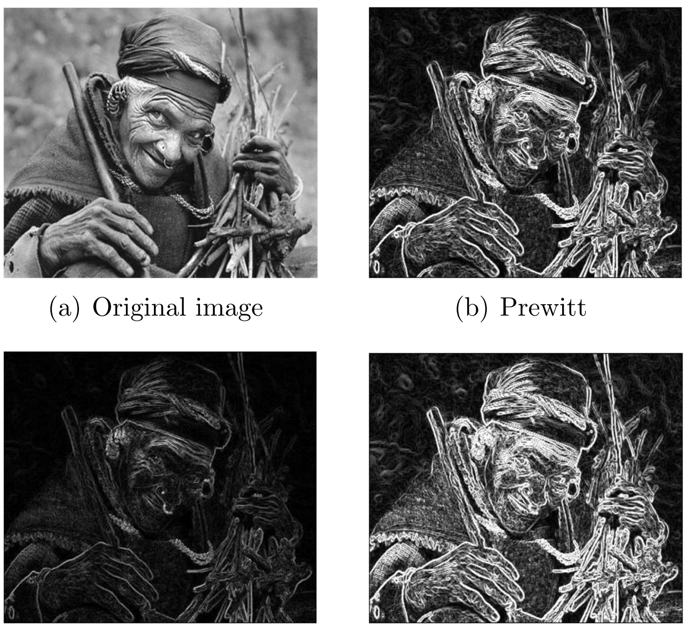
$$\theta(i,j) = \arctan \frac{G_y}{G_x} \tag{7}$$

The gradient magnitude gives the maximum rate of increase of $I(i,j)$ per unit distance in the direction of θ, which can be acquired as follows:

$$|\nabla| M(i,j) = \sqrt{G_x^2 + G_y^2} \tag{8}$$

4. **Result.** Experimentation is carried on MATLAB 2014b for different images and 0.2 Gaussian noise is added into image files. Edge detection via the proposed algorithm is performed here and yields different detection results which are compared to edge detection methods such as Prewitt algorithm [7] and Canny algorithm [3]. Then performance analysis of these detection algorithms is given.

4.1. **Results of edge detection via different algorithms.** For illustration purpose, firstly we show some detection results including one old lady before and after the edge detection using traditional methods and our method. This can give an intuitively visual effect between different detection algorithms.



(a) Original image (b) Prewitt

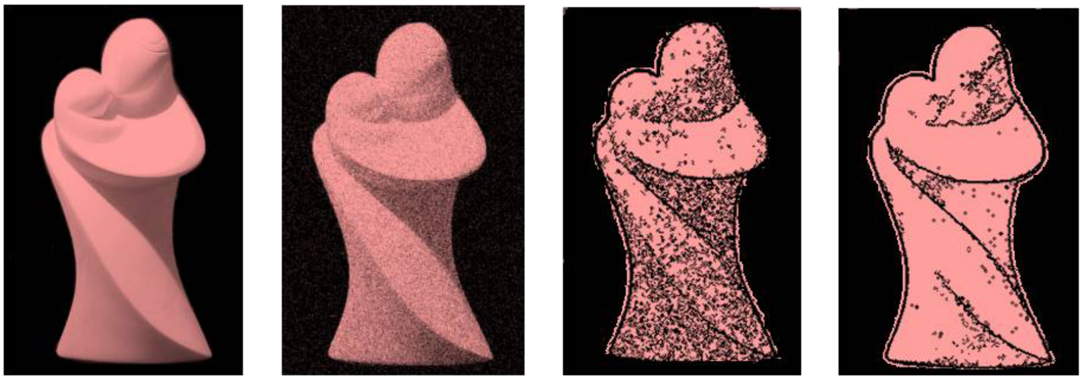
(c) Canny (d) Our method

FIGURE 3. Edge detection of image including one old lady via different algorithms

Figure 3 shows the results of the edge detection of one old lady using different algorithms. Figure 3(a) is the original image including one old lady. Figure 3(b) is the detection result via Prewitt algorithm [7]. Figure 3(c) is the detection result via Canny algorithm [3]. Figure 3(d) is the detection result via our method.

From Figure 3, due to the mean shift smoothing, results of edge detection show that our method amplifies the contrast along edges, and works better than the existing algorithms in terms of smoothness, details, sharpness and continuity.

In order to highlight the effect of noise removal, we add Gaussian noise into the sculpture image, and compare the edge detection between Canny method and our method in Figure 4.



(a) Original sculpture (b) Noisy sculpture (c) Canny (d) Our method

Figure 4. Effect of noise removal for edge detection via different algorithms

From Figure 4, we could provide a robust solution that is adaptable to the varying noise levels of image to help distinguish valid edges from visual artifacts introduced by noise. So our algorithm performs better effect of noise removal than classic operator.

4.2. **Performance analysis of edge detection algorithms.** To give an objective evaluation, we use PSNR (Peak Signal to Noise Ratio) and Entropy as performance parameters which are evaluated and shown in Figure 5. High value of PSNR is shown that the edge detected image consists of less noise as compared to the traditional operators. Entropy is defined in terms of the probabilistic behavior of a source of information. From Figure 5, we can acquire better values of PSNR and Entropy, which proves effectiveness of our method.

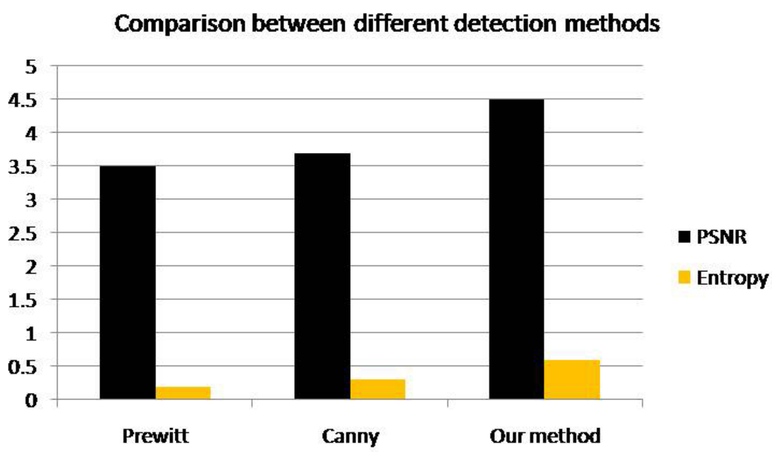


Figure 5. Performance evaluation via PSNR and Entropy

Prewitt detector has a major drawback of being very sensitive to noise. The size of the kernel filter and coefficients are fixed and cannot be adapted to a given image.

The performance of the Canny algorithm depends heavily on the adjustable parameters, σ, which is the standard deviation for the Gaussian filter, and the threshold values, T_1 and T_2. σ also controls the size of the Gaussian filter. The bigger the value for σ, the larger the size of the Gaussian filter becomes.

Also, in the execution time, our method can acquire about 30ms which respectively compared to 56ms and 43ms when using Prewitt and Canny in Figure 3, which shows that we can acquire quick calculation speed.

5. **Conclusions.** In this paper, an efficient edge detection has been proposed and implemented to improve detection effectiveness. Our method not only can detect edge in images precisely but also remove noise as well. Experimental results show that we can acquire a better visual result of detection in terms of noise removal, smoothness, details and continuous edge boundary. The values of PSNR and Entropy by our method are higher compared to the other classic methods such as Canny and Prewitt. Also some drawbacks of edge detection via Canny and Prewitt are analyzed finally.

The proposed approach involves a step that looks for further image evidence and connects short edge contours into longer ones. In future the exact relationship between mean pixel value on edges and sigma may be found out.

Acknowledgment. This work was supported by the Science and Technology Foundation of Heilongjiang Province Education Department of China (No. 12541710).

REFERENCES

[1] J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.8, no.6, pp.184-203, 1986.

[2] Y. Zhao, W. Gui, Z. Chen, J. Tang and L. Li, Medical Images edge detection based on mathematical morphology, *Proc. of the 27th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society*, pp.6492-6495, 2006.

[3] L. Zhe, F. Wang and Y. Chang, An improved Canny algorithm for edge detection, *Journal of Northeastern University*, vol.28, no.12, pp.1681-1684, 2007.

[4] X. Song, J. Luo, L. Wang and Z. Shen, Edge detection method based on mean shift clustering method, *Journal of Projectiles Rockets Missiles & Guidance*, vol.27, no.1, pp.366-368, 2007.

[5] S. Mathur and A. Ahlawat, Application of fuzzy logic on image edge detection, *Ithea*, 2008.

[6] A. A. Alshennawy and A. A. Aly, Edge detection in digital images using fuzzy logic technique, *Proc. of World Academy of Science Engineering and Technology*, pp.252-262, 2009.

[7] M. Kang and Q. Xu, Adaptive edge-detection method based on Prewitt algorithm, *Application Research of Computers*, vol.26, no.6, pp.2383-2386, 2009.

[8] M. K. Ray, D. Mitra and S. Saha, Simplified novel method for edge detection in digital images, *Proc. of International Conference on Signal Processing, Communication, Computing and Networking Technologies*, pp.197-202, 2011.

[9] J. Patel, J. Patwardhan, K. Sankhe and R. Kumbhare, Fuzzy inference based edge detection system using Sobel and Laplacian of Gaussian operators, *Proc. of the International Conference and Workshop on Emerging Trends in Technology*, pp.694-697, 2011.

[10] G. T. Shrivakshan and C. Chandrasekar, A comparison of various edge detection techniques used in image processing, *International Journal of Computer Science Issues*, vol.9, no.5, pp.269-276, 2012.

[11] N. S. Joshi and N. S. Choubey, Comparison of traditional approach for edge detection with soft computing approach, *International Journal of Computer Applications*, vol.96, no.11, pp.17-23, 2014.

[12] B. O. Sadiq, S. M. Sani and S. Garba, An approach to improving edge detection for facial and remotely sensed images using vector order statistics, *International Journal of Multimedia and Its Applications*, vol.7, no.1, pp.17-25, 2015.

[13] P. S. Tirath and K. J. Yogendra, Improved simplified novel method for edge detection in grayscale images using adaptive thresholding, *Journal of Advances in Computer Networks*, vol.3, no.2, pp.157-161, 2015.