https://doi.org/10.2352/ISSN.2470-1173.2020.10.IPAS-133

Edge Detection Using the Bhattacharyya Distance with Adjustable Block Space

Jiho Yoon, Chulhee Lee; Department of Electrical and Electronic Engineering, Yonsei University; Seoul, Republic of Korea

Abstract

In this paper, we propose a new edge detection method for color images, based on the Bhattacharyya distance with adjustable block space. First, the Wiener filter was used to remove the noise as pre-processing. To calculate the Bhattacharyya distance, a pair of blocks were extracted for each pixel. To detect subtle edges, we adjusted the block space. The mean vector and covariance matrix were computed from each block. Using the mean vectors and covariance matrices, we computed the Bhattacharyya distance, which was used to detect edges. By adjusting the block space, we were able to detect weak edges, which other edge detections failed to detect. Experimental results show promising results compared to some existing edge detection methods.

Keywords - edge detection, Bhattacharyya distance

I. Introduction

Edge detection is one of the basic tools for feature extraction in image processing and computer vision. Various edge detection methods have been proposed, which include the Sobel filter [1] and Canny edge detection [2]. These methods show good performance in most applications. On the other hand, the methods may fail when edges are not strong. Recently, edge detection methods based on Bhattacharyya distance were proposed for color images [6] and multispectral images [3],[4]. In [7], various contours within the block were considered to improve performance.

In this paper, we propose a new method for edge detection based on Bhattacharyya distance. In particular, we adjust the distance between the two blocks to detect weak edges, which might be missed by the existing methods.

The rest of this paper is organized as follows: Section II provides the proposed method. Section III shows experimental results and Section IV presents conclusions.

II. Methodology

Figure 1 shows a block diagram of the proposed method. First, the Wiener filter was used to remove noise. Then, we selected two blocks with adjustable block spaces. The mean vector and covariance were computed from each block. Using the mean vectors and covariance matrices, we computed the Bhattacharyya distance, which was used to detect edges. By adjusting the block space, we were able to detect weak edges, which the other edge detections failed to detect.

A. Pixel-wise Wiener Filter

Noise may adversely affect edge detection. Thus, we first applied a pixel-wise adaptive Wiener filter based on statistical characteristic estimation from the local neighborhood of each pixel [3],[5]. It uses the local mean, variance and noise variance as

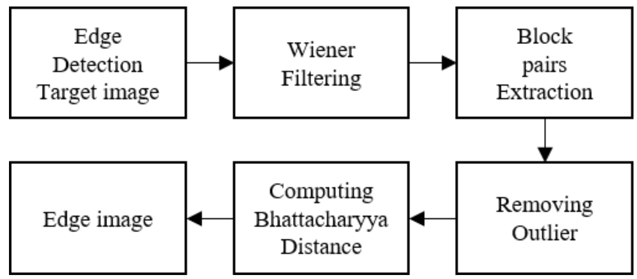

Figure 1. *Block diagram of the proposed method.*

follows:

$$\mu = \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} I(i,j) \tag{1}$$

$$\sigma^2 = \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} I^2(i,j) - \mu^2 \tag{2}$$

$$W(i,j) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2} [I(i,j) - \mu] \tag{3}$$

where $I(i,j)$ is an input image and $W(i,j)$ is an output image. ν^2 is the noise variance.

B. Point-symmetric Block Pair Extraction

In [3],[4], Youn proposed an edge detection method based on the Bhattacharyya distance, which used four pairs of neighborhood blocks: up-down (UD), left-right (LR), diagonal-left-down (DLD) and diagonal-right-down (DRD). A problem with this approach is that the method may fail to detect weak edges in some cases. In particular, when the intensity level varies slowly around edges, existing edge detection methods may fail to detect the weak edges. To detect such slowly-varying edges, we adaptively adjusted the block shape and the distance between blocks (Figure 2-4). To consider various edge directions, we rotated the two blocks. Also, to detect weak edges, we adjusted the block size and the block distance.

C. Removing Outlier Using the Mahalanobis Distance

The block may contain outliers, which can affect the mean vector and covariance matrix. These outliers may contaminate the mean and covariance estimations and produce erroneous edges. To remove such outliers, we used the Mahalanobis distance [4]. The Mahalanobis distance is defined as follows [8]:

$$MD = \sqrt{(x_i - \mu)\Sigma^{-1}(x_i - \mu)^T} \tag{4}$$

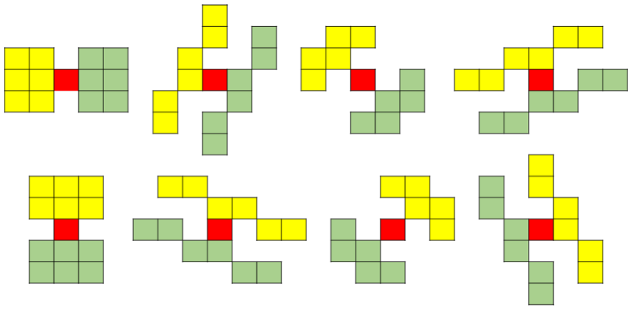

Figure 2. *Block rotation.*

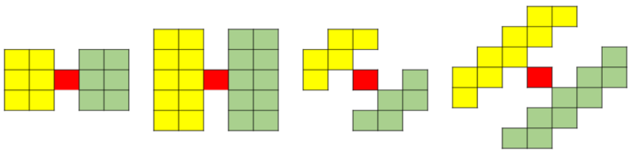

Figure 3. *Block size.*

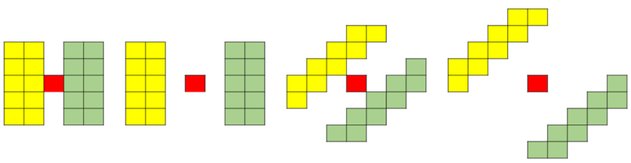

Figure 4. *Block distance.*

where MD is the Mahalanobis distance of observation $\vec{x} = (x_1, x_2, x_3, ..., x_N)^T$ and Σ is the covariance matrix. If the Mahalanobis distance of a pixel within the block is higher than a threshold, the pixel was removed.

D. Bhattacharyya Distance

The Bhattacharyya distance is used to measure the similarity between the two groups. If the statistical characteristics of the two populations show larger differences, the Bhattacharyya distance will increase. Assuming normal distributions, it can be computed by using means, covariance matrices, and determinants:

$$BD = \frac{1}{8}(\vec{\mu}_1 - \vec{\mu}_2)^T \Sigma^{-1} (\vec{\mu}_1 - \vec{\mu}_2) + \frac{1}{2} \ln\left(\frac{det\Sigma}{\sqrt{det\Sigma_1 det\Sigma_2}}\right) \quad (5)$$

where BD represents the Bhattacharyya distance, μ_1 and μ_2 represent the mean vector and the covariance matrix of class i, respectively. Σ represents the average of Σ_1 and Σ_2. We computed the Bhattacharyya distance for each pair of blocks at various block spaces. The Bhattacharyya distance of the block pair having the maximum value was chosen as the edge strength of the pixel:

$$Edgeimage = Max\left(BD_1, BD_2, BD_3, ..., BD_{N_{i,j}}\right) \quad (6)$$

where $N_{i,j}$ represents the number of block pairs.

III. Experimental Results

In this paper, we used the data set for logo recognition (MICC-Logos) [9]. This data set consists of 720 images, which include 13 logo classes. The parameters used in this experiment are as follows. In the Wiener filter, the window size was set to 3-by-3. The block distance was set to 1, 3 and 5. The proposed method was compared with the method based on the Bhattacharyya distance (four fixed block pairs) [3],[4], the Sobel filter, and the Canny edge detection. Figures 5-10 show some edge results. Figures 5-7 show a comparison of the Sobel/Canny methods and the proposed method Although the Sobel and Canny edge detection algorithms failed to detect some weak edges, the proposed algorithm successfully detected such weak edges. Figures 8-10 show a comparison of Youn's method and the proposed method. In Figure 9, most edges were successfully detected by the proposed method. In Figure 10, the edges around the dark object were detected by the proposed method. Whereas the conventional methods and the previous edge detection method based on the Bhattacharyya distance failed to detect some weak edges or slowly-varying edges, the proposed method successfully detected such difficult edges.

IV. Conclusions

In this paper, we proposed an edge detection algorithm using the Bhattacharyya distance with adjustable block space. The proposed algorithm was able to detect some weak edges missed by the conventional edge detection methods such as Sobel filtering or Canny edge detection. By adaptively adjusting the distance between the two blocks, we were able to detect slowly-varying edges, which most existing edge detection methods failed to detect.

Acknowledgments

This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2018R1D1A1B07050345) and in part by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.R2212-15-0022).

References

[1] N. Kanopoulos, N. Vasanthavada and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," IEEE Journal of solid-state circuits vol. 23(2), pp.358-367, 1988.

[2] J. Canny, "A computational approach to edge detection," Readings in computer vision, pp. 184-203, 1987.

[3] S. Youn and C. Lee, "Edge detection for hyperspectral images using the Bhattacharyya distance," International Conference on Parallel and Distributed Systems, pp. 716-719, 2013.

[4] S. Youn and C. Lee, "Multi-dimensional edge detection operators," SPIE Sensing Technology Applications, pp. 912407-7, 2014.

[5] H. Hassanpour, and P. M. Farahabadi, "Using hidden Markov models for paper currency recognition," Expert Systems with Applications, 36(6), pp. 10105–10111, 2009.

[6] Y. Cheon and C. Lee, "Color Edge Detection based on Bhattacharyya Distance," ICINCO, pp. 368-371, 2017.

[7] A. Badripour and C. Lee, "Edge Detection Based on Bhattacharyya Distance for Color Images Using Adaptive Boundary and Thresholding," Proceedings of the Korea Information Processing Society Conference, vol. 24(2), pp. 944-945, 2017. Technol., 42, 112 (1998).

[8] R. De Maesschalck, D. Jouan-Rimbaud and D.L. Massart, "The Mahalanobis distance," Chemometrics and Intelligent Laboratory Systems, vol. 50(1), pp. 1-18, 2000.

[9] H. Sahbi, L. Ballan, G. Serra, and A. Del Bimbo, "Context-Dependent Logo Matching and Recognition," IEEE Transactions on Image Processing, vol. 22(3), pp. 1018-1031, 2013.

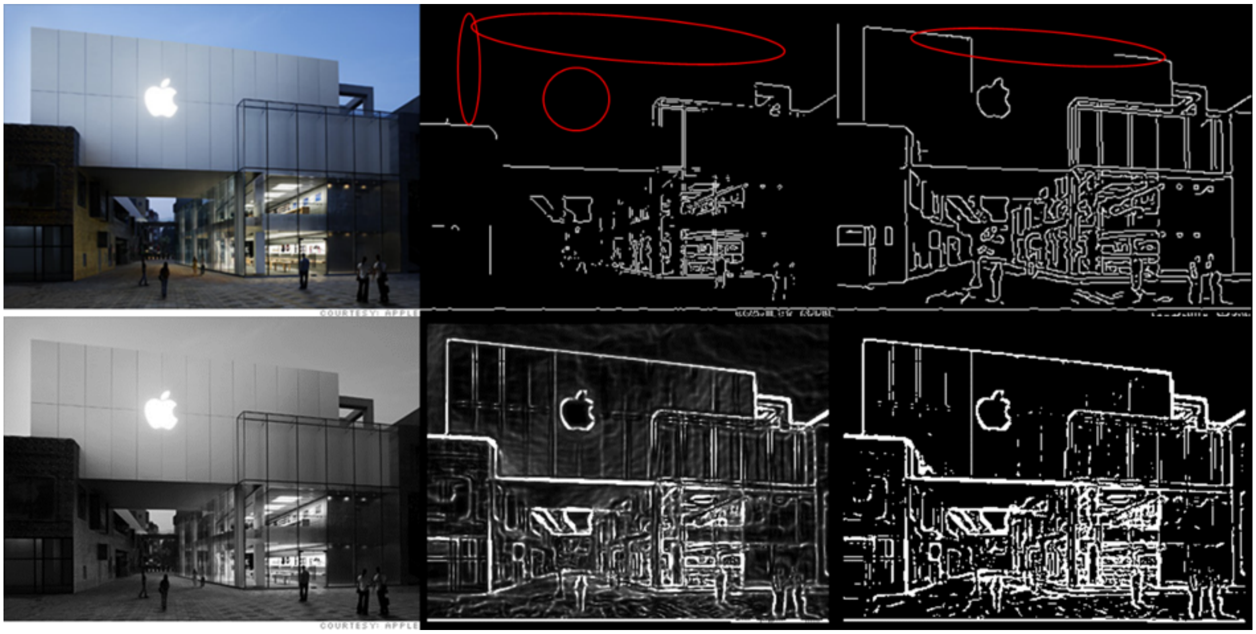

Figure 5. *(a) Original image, (b) Sobel filtering, (c) Canny edge, (d) Gray image, (e) Proposed method (gray scale), (f) Proposed method (binary)*

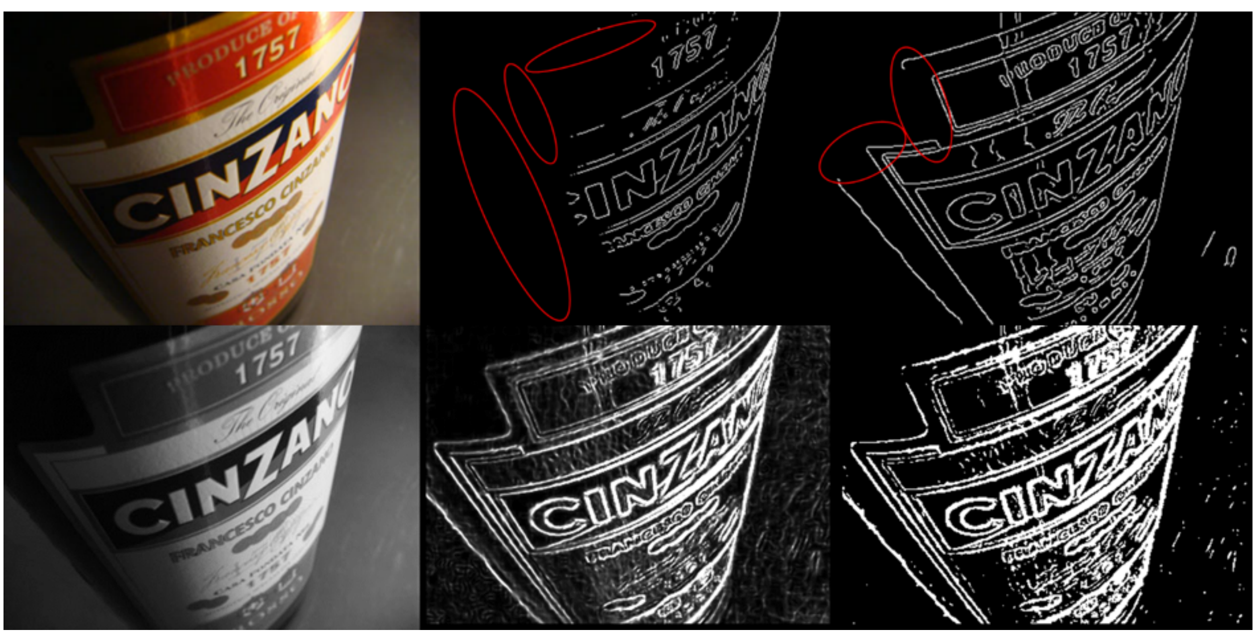

Figure 6. *(a) Original image, (b) Sobel filtering, (c) Canny edge, (d) Gray image, (e) Proposed method (gray scale), (f) Proposed method (binary)*

Figure 7. *(a) Original image, (b) Sobel filtering, (c) Canny edge, (d) Gray image, (e) Proposed method (gray scale), (f) Proposed method (binary)*

Figure 8. *(a) Original image, (b) Youn's method (gray scale), (c) Youn's method (binary), (d) Gray image, (e) Proposed method (gray scale), (f) Proposed method (binary)*

Figure 9. *(a) Original image, (b) Youn's method (gray scale), (c) Youn's method (binary), (d) Gray image, (e) Proposed method (gray scale), (f) Proposed method (binary)*

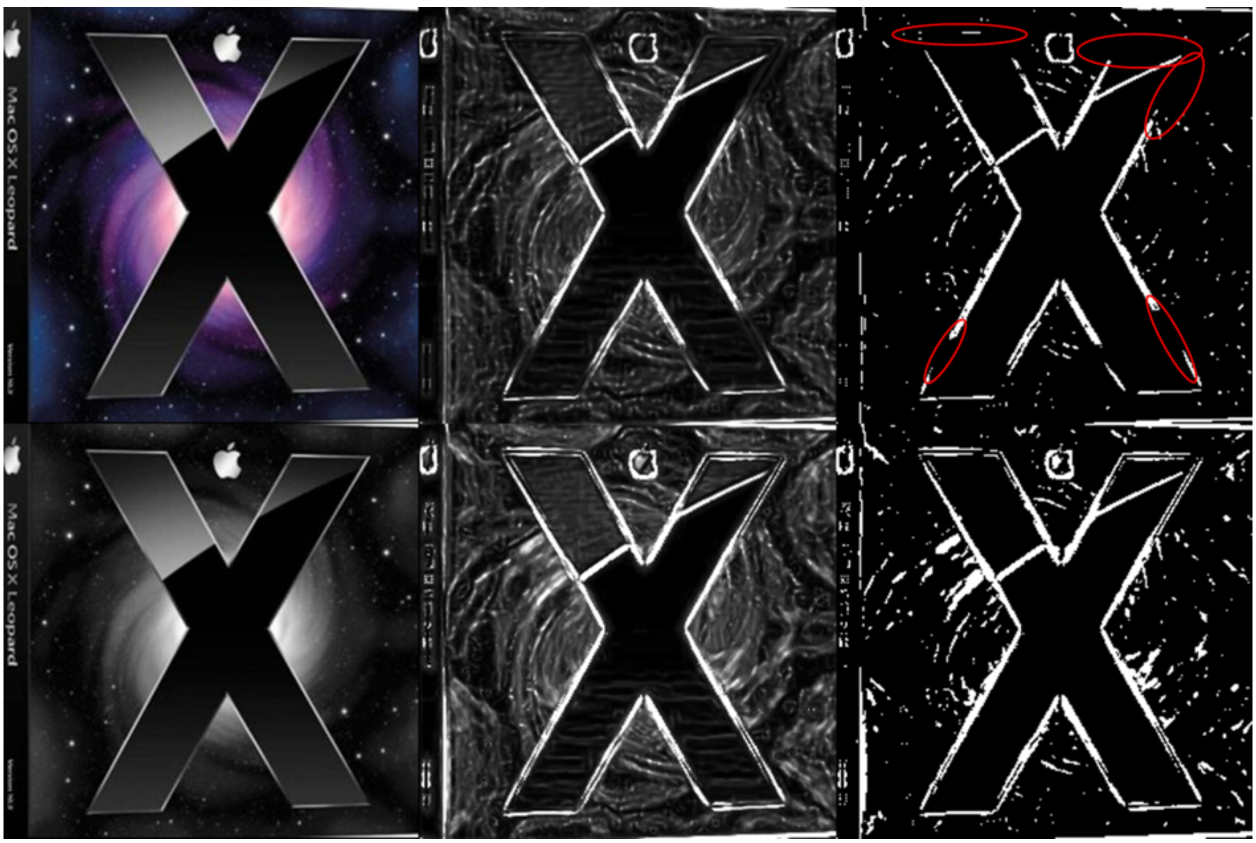

Figure 10. *(a) Original image, (b) Youn's method (gray scale), (c) Youn's method (binary), (d) Gray image, (e) Proposed method (gray scale), (f) Proposed method (binary)*