ARPN Journal of Engineering and Applied Sciences

REAL TIME STATISTICALLY BOUNDARY DETECTION

Panca Mudjirahardjo, Raden Arief Setyawan and Hadi Suyono
Department of Electrical Engineering, Engineering Faculty, University Brawijaya, Jl. MT. Haryono, Malang, Indonesia
E-Mail: panca@ub.ac.id

ABSTRACT

The first task for object detection and recognition is image segmentation. This process is initialized with boundary detection. In this paper we propose a boundary detection method based on standard deviation value of intensity within the window. We assume that the low value is similar colour where the computation process occur on the same object. While the high value, the computation process will occur on the object's boundary or edge. In this research we compare to other methods to evaluate the performance of proposed method. The experimental result, by using window size of 3×3 pixels, our method's computation time is 66.54 ms and using window size of 5×5 pixels, the computation time is 67.39 ms. Video rate is 15 fps. Processing with the proposed method yields less noise compared to other method, both for colour and grayscale processing.

Keywords: boundary detection, image segmentation, real time application, standard deviation.

INTRODUCTION

Boundary detection is very important task in pattern recognition specially for image segmentation. This task initiates process in pattern recognition. It will detect the boundary of object or part of object, known as edge [1]. It usually labels the edge of an object by detecting the gradient over it, then localize the interested object to be further processed. Unfortunately, some edge's label are not necessary for some application. Some application need the global edge, not edge of object's part. Because it need shape of the object. Some edge detection have been introduced and applied in image processing, such as Canny, Sobel and Laplace operator.

Some boundary detection methods have been researched by researcher. In [2], Kokkinos combined the Canny (Gradient-Magnitude) and the PB edge detectors. They classified each candidate edge based on the distribution of gradients in its vicinity, captured by SIFT descriptors. They simplified the boundary classification task by extracting SIFT descriptors aligned with the orientation of the initially detected edges, thereby discarding variation due to orientation. Martin *et al.* [3] detected and localized boundaries in natural scenes using local image measurements. They formulated features that respond to characteristic changes in brightness, colour, and texture associated with natural boundaries. In order to combine the information from these features in an optimal way, they trained a classifier using human labeled images as ground truth. Ma *et al* [4] utilized a predictive coding model to identify the direction of change in colour and texture at each image location at a given scale, and constructs an edge flow vector. By propagating the edge flow vectors, the boundaries can be detected at image locations which encounter two opposite directions of flow in the stable state. A user defined image scale was the only significant control parameter that is needed by the algorithm. The scheme facilitated integration of colour and texture into a single framework for boundary detection. Mudjirahardjo *et al.* [5] utilized a histogram of s-RGB to detect the image background. They assumed the dominant colour detected by this histogram to be background. In [6], they utilized a histogram of transition as a feature to detect a human's head-shoulder. They calculated the transition from background to foreground in binary image. This method relied on the performance to extract foreground. Isola *et al.* [7] proposed a novel method for detecting the boundaries based on a simple underlying principle: pixels belongin to the same object exhibit higher statistical dependencies than pixels belonging to different objects. They utilized pointwise mutual information with spectral clustering to find object boundaries in the image. Kattire *et al* [8] performed the boundary detection in two steps. The first step, vector information of image is calculated. In the second step texture information is calculated. In this method magnitude and directions, both were considered while calculating vector information. The vector details of image gave idea about edge. Texture details were used to derive edge map.

In this paper we propose and evaluate a boundary detection based on standard deviation value for real time application. We modify and simplify the available method in [7].

METHOD

We use statistical analysis to define a boundary. First we define a window surround a pixel in coordinate (x,y). Then, we calculate the average of every a colour intensity within a window of pixels centered at coordinate (x,y), $\bar{I}_{a(x,y)}$,

$$\bar{I}_{a(x,y)} = \frac{\sum_n I_a}{n} \quad (1)$$

Where a is red, green or blue colour. n is the number of pixel within the window.

Calculate the standard deviation of colour $a$in a window centered at coordinate (x,y), $SD_{a(x,y)}$,

$$SD_{a(x,y)} = \sqrt{\frac{\sum_{n}\left(I_a - \bar{I}_{a(x,y)}\right)^2}{n-1}} \quad (2)$$

Calculate the maximum standard deviation of every $SD_{a(x,y)}$ in a whole image,

$SD_{a(\max)} = \max\left(SD_{a(c,r)}\right)$ c = 1, … , image column; r = 1, … , image row. (3)

Calculate the total maximum Standard Deviation, in a whole image,

$$SD_{\max} = \max\left(SD_{R(\max)}, SD_{G(\max)}, SD_{B(\max)}\right) \quad (4)$$

Based on standard deviation characteristic, data with low value of standard deviation mean the value to be almost similar. The intensity value is almost similar within the window. It is often occurred when the window is in the one object. However, if the standard deviation value is high, the window should be in the edge or boundary of the object. So the definition of the pixel in coordinate (x,y), p(x,y), as a boundary or background is in (5),

$$p(x,y) = \begin{cases} boundary & if\ SD_{a(x,y)} \geq SD_{\max} - th \\ background & otherwise \end{cases} \quad (5)$$

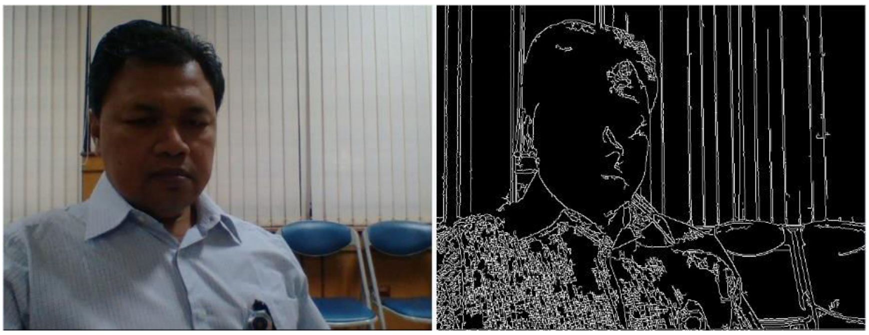
In the grayscale image, we modify equation (1) until (4) for grayscale processing and (5) we modify as (6),

$$p(x,y) = \begin{cases} boundary & if\ SD_{(x,y)} \geq SD_{\max} - th \\ background & otherwise \end{cases} \quad (6)$$

EXPERIMENT RESULT

`To conduct the experiment, we set-up the experimental environment as follows: Operating system is Windows 8.1 Pro; the processor is Intel® coreTM i5-4210 U CPU @ 1.70 GHz 2.40 GHz; 4GB RAM; the built in web camera; and the used software is Microsoft Visual Studio 2010. We evaluate the image size of 640×480 pixels. Video rate is 15 fps.

In this experiment, to evaluate this method's performance, we use window size 3×3, 5×5 pixels in colour image and window size 3×3 in grayscale image. To compare with other method, we use Canny, Sobel and Laplace operator [1].When we use window size 3×3, n is 9. When we use window size 5×5, n is 25.



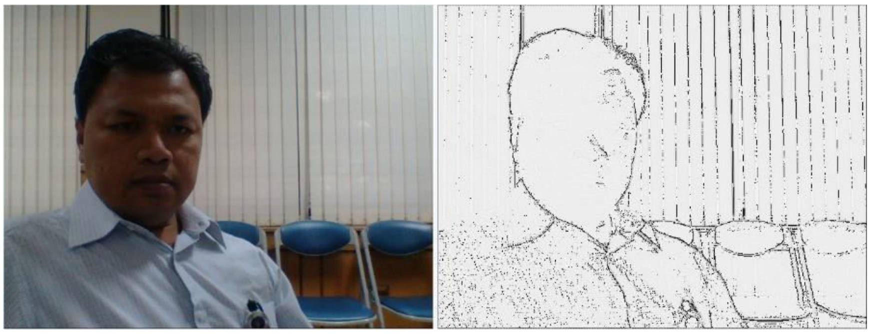
(a) (b)

Figure-1. Boundary detection based on Canny operator (a) input image (b) output image.

(a) (b)

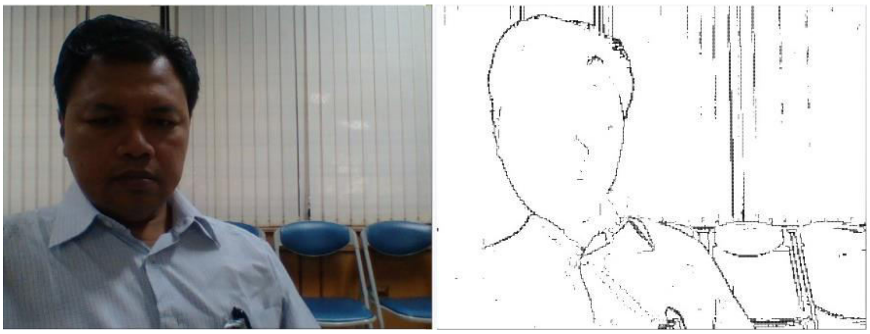
Figure-2. Boundary detection based on Sobel operator (a) input image (b) output image.

ARPN Journal of Engineering and Applied Sciences



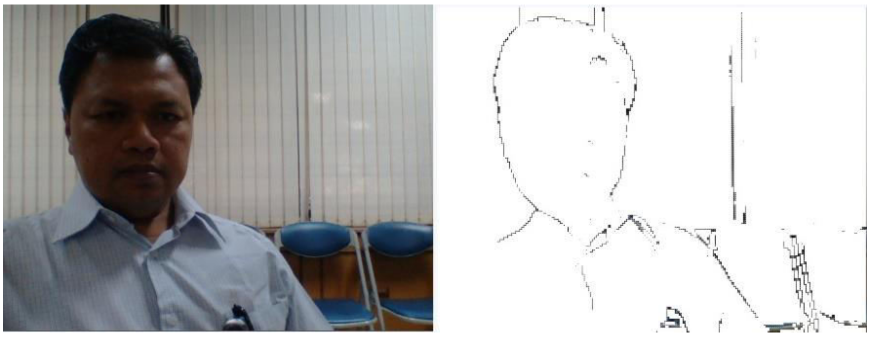
(a) (b)

Figure-3. Boundary detection based on Laplace operator (a) input image (b) output image.



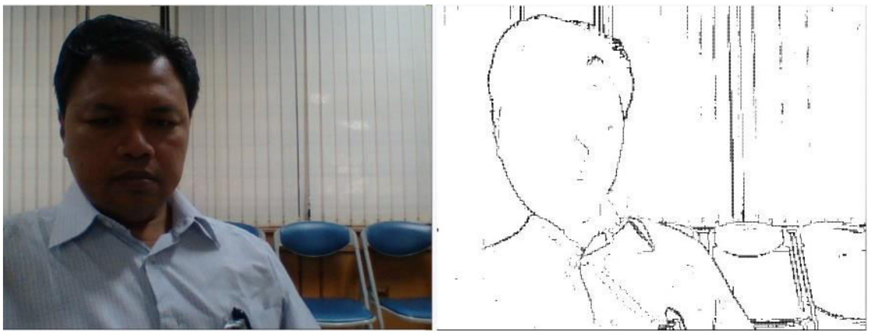
(a) (b)

Figure-4. Boundary detection based on SD colour image, window size 3×3 pixels (a) input image (b) output image.



(a) (b)

Figure-5. Boundary detection based on SD colour image, window size 5×5 pixels (a) input image (b) output image.



(a) (b)

Figure-6. Boundary detection based on SD grayscale image, window size 3×3 pixels (a) input image (b) output image.

ARPN Journal of Engineering and Applied Sciences

The experiment result is shown in Figures1-6 and Table-1.

Table-1. The computation time of the process.

Process	Computation time (ms)
Canny operator (Figure-1)	67.49
Sobel operator (Figure-2)	64.23
Laplace operator (Figure-3)	66.97
SD RGB with window size 3×3 pixels (Figure-4)	66.54
SD RGB with window size 5×5 pixels (Figure-5)	67.39
SD grayscale with window size 3×3 pixels (Figure-6)	67.03

In Figure-1, it shows too sensitive to detect a boundary. It is edge detection. Canny operator is suitable for edge detection. Figure-2, Sobel operator can detect the object's boundary and segments the object in the image. However, there are much noise there. Figure-3 with Laplace operator, there are much noise there. Figures 4-6 are with SD method, process in window 3×3, 5×5, and grayscale respectively. With SD method, there are less noise there compare to other method. The computation time is relative short.

CONCLUSIONS

We have demonstrated and evaluated the proposed method of boundary detection. This proposed method is based on standard deviation value within the processed window. Processing with the proposed method yields less noise compared to other method. It gives a better result both colour and grayscale processing. The computation time is 66.54 - 67.39 ms, with image size of 640×480 pixels. This is suitable for real time application.

To improve the detection result, it is necessary to perform a further process to complete the boundary line and reduce the noise.

STATEMENT

In this research, we involved author's face as the object of this study.

ACKNOWLEDGMENTS

This work was supported by DRPM Ditjen Penguatan Risetdan Pengembangan Kemristekdikti Indonesia, Grant No. 063/SP2H/LT/DRPM/IV/2017.

REFERENCES

[1] Gonzales R. C., Woods R. E. 202. Digital Image Processing, second edition. Prentice Hall. Upper Saddle River, New Jersey, USA.

[2] Kokkinos I. 2010.Highly Accurate Boundary Detection and Grouping. Proc. on IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010).

[3] Martin D.R., Fowlkes C.C., Malik J.2004.Learning to Detect Natural Image Boundaries Using Local Brightness, Color and Texture Cues. IEEE Transactions on Pattern Analysis and Machine Intelligence. 26(1): 1-20.

[4] Ma W.Y., Manjunath B.S. 2000.EdgeFlow: A Technique for Boundary Detection and Image Segmentation. IEEE Transactions on Image Processing. 9(8): 1375-1388.

[5] Mudjirahardjo P., Nurussa'adah Siwindarto, P. 2016. Soccer Field Detection Based on Histogram of s-RGB. ARPN Journal of Engineering and Applied Sciences. 11(21): 12405-12408.

[6] Mudjirahardjo P., Purnomo M.F.E., Hasanah R.N., Suyono H. 2016.Histogram of Transition for Human Head Recognition. Jurnal Teknologi. 78(5-9): 53-58.

[7] Isola P., Zoran D., Krishnan D., Adelson E.H. 2014.Crisp Boundary Detection Using Pointwise Mutual Information. Proc. on European Conference on Computer Vision (ECCV 2014).

[8] Kattire S.S., Shah A.V. 2014.Boundary Detection Algorithm Implementation for Medical Images. International Journal of Engineering Research & Technology (IJERT). 3(12): 285-287.