Hybrid Image Thresholding Method using Edge Detection

Febriliyan Samopa† and Akira Asano††,

Department of Information Engineering, Graduate School of Engineering,
Hiroshima University, Higashi-Hiroshima, Hiroshima, Japan

Summary
The main disadvantage of traditional global thresholding techniques is that they do not have an ability to exploit information of the characteristics of target images that they threshold. In this paper, we propose a hybrid thresholding method that combines the P-tile method with an edge detector to assist it in the thresholding process. This method successfully generates more accurate object shape extraction than the conventional methods.

Key words:
Image Thresholding, P-tile, Edge Detection.

1. Introduction

In many applications of image processing, pixel values belonging to the object are substantially different from those in its background. Thresholding is one of the simplest and most commonly used technique to separate the foreground from its background [1][2][3].

Thresholding techniques can be categorized into two classes: global thresholding and local (adaptive) thresholding. In the global thresholding, a single threshold value is used in the whole image. In the local thresholding, a threshold value is assigned to each pixel to determine whether it belongs to the foreground or the background pixel using local information around the pixel.

Because of the advantage of simple and easy implementation, the global thresholding has been a popular technique in many years. Several successful thresholding methods based on histogram techniques have been proposed, for example, the methods proposed by Kittler and Illingworth [2], Otsu [4], and the P-tile method [5]. Thresholding techniques based on entropy measures [1][6][7][8] and fuzzy approaches [2][9] have also been proposed.

The main disadvantage of traditional thresholding techniques is that they do not have an ability to exploit information of the characteristics of the images that they threshold. They treat all images in the same way, regardless of the specific nature of the images. For some situations, this 'one-fits-all' approach is sufficient. However, when greater accuracy and more consistent performance are required, more information should be used to assist the thresholding process.

Manuscript received April 5, 2009
Manuscript revised April 20, 2009

This paper proposes a method of utilizing shape information to assist thresholding process. We combine the P-tile global thresholding method with some edge detection methods to retrieve shape information for assistance, and demonstrate its usefulness in various situations. This is a promising approach because it generates more accurate thresholded images than conventional methods especially for applications that need to extract the object shape.

2. P-tile Thresholding Method

P-tile is a shorter form of the word "percentile". The threshold is chosen to be the intensity value where the ratio of the number of pixels whose value is higher than the threshold to the total number of pixels in the image is closest to the given percentile.

The P-tile method is one of the earliest thresholding methods based on the gray level histogram [5]. It assumes the objects in an image are brighter than the background, and occupy a fixed percentage of the picture area. This fixed percentage of picture area is also known as P%. The threshold is defined as the gray level that mostly corresponds to mapping at least P% of the gray level into the object.

Let n be the maximum gray level value, $H(i)$ be the histogram of image (i = 0 .. n), and P be the object area ratio. The algorithm of the P-tile method is as follows:

$s \leftarrow \mathrm{sum}(\ H(i)\ )$ # total image area #
$f \leftarrow s$ # initialize all area as object area #
for $k \leftarrow 1$ to n
  $f \leftarrow f - H(k-1)$ # remove k–1 from object area #
  if $(\ f / t\ ) \leqslant P$ then stop
$T \leftarrow k$ # final threshold value #

This method is simple and suitable for all sizes of objects. It yields good anti-noise capabilities, however, it is obviously not applicable if the object area ratio is unknown or varies from picture to picture.

Unfortunately, we do not usually have such definite information about the object area ratio. This information can sometimes be substituted by knowledge of another property, for example the average width of lines in drawings, shape, etc.

3. Edge Detection Methods

Edge detection is a fundamental tool used in most image processing applications to obtain information from images as a precursor step to feature extraction and object segmentation. This process detects boundaries between objects and the background in the image at which the image brightness changes sharply or more formally has discontinuities. The image containing these boundaries is known as edge map. The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world

There are many ways to perform edge detection, however most of them grouped into two categories, gradient and Laplacian. The gradient method detects the edges by looking for the local maximum and minimum in the first derivative of the image. The Laplacian method searches for zero crossings in the second derivative of the image.

Some of the early gradient operators include Roberts [10], Prewitt [11], Sobel [12], Canny [13] edge operators. They involve small kernels to convolve with an image to estimate the first-order directional derivatives of the image brightness distribution. The edge value is calculated by forming a matrix centered on each pixel. If the value is larger than a given threshold, then the pixel is classified as an edge. All the gradient-based algorithms have kernel operators that calculate the edge strength in directions which are orthogonal to each other, commonly vertically and horizontally. The contributions of the both components are combined to give the total value of the edge strength.

Table 1. Comparison of 5% and 1% Steps

No	Name	5% Step			1% Step			MSE Difference (%)	Speed-Up Ratio
		Time	Threshold (%)	MSE	Time	Threshold (%)	MSE		
1	airplane.bmp	0.5216	35	7056.46	2.9633	38	7013.14	0.62	5.7
2	apples.bmp	0.2123	80	7301.62	1.452	79	7292.14	0.13	6.8
3	bag.bmp	0.0705	35	4833.49	0.3894	37	4815.93	0.36	5.5
4	barbaragray.bmp	0.4561	50	7254.73	2.4527	47	7217.93	0.51	5.4
5	bracelet.bmp	0.1971	95	2749.52	1.0747	95	2749.52	0.00	5.5
6	brain.bmp	0.0231	15	3763.97	0.124	17	3728.90	0.94	5.4
7	brainweb.bmp	0.0571	15	4579.14	0.3007	16	4563.38	0.35	5.3
8	cameraman.bmp	0.0929	60	7214.09	0.4883	61	7208.07	0.08	5.3
9	cell.bmp	0.3614	75	12484.92	1.7345	76	12484.33	0.00	4.8
10	circuit.bmp	0.0561	30	7477.75	0.2941	29	7471.06	0.09	5.2
11	circuitry.bmp	0.1017	35	5460.24	0.539	36	5460.08	0.00	5.3
12	city.bmp	0.3244	60	6681.03	1.7745	58	6660.55	0.31	5.5
13	coast1.bmp	2.2715	35	10518.16	11.4059	35	10518.16	0.00	5.0
14	coast2.bmp	1.9625	40	10837.48	15.1488	41	10833.93	0.03	7.7
15	coast3.bmp	1.9701	25	10473.15	12.3095	27	10461.79	0.11	6.2
16	coins.bmp	0.1012	30	5079.47	0.5194	32	5006.88	1.45	5.1
17	fluorescence.bmp	2.9611	10	2456.43	15.3493	8	2380.84	3.17	5.2
18	house.bmp	0.7094	50	4767.58	3.6364	52	4745.01	0.48	5.1
19	koi.bmp	0.0312	30	10270.54	0.1651	26	10258.55	0.12	5.3
20	lenagray.bmp	0.5278	55	8159.16	2.4702	56	8151.58	0.09	4.7
21	lung.bmp	1.2517	20	2904.09	6.5399	19	2903.96	0.00	5.2
22	map.bmp	0.1239	55	5889.52	0.6337	54	5884.94	0.08	5.1
23	moon.bmp	0.2932	25	2174.10	1.6295	23	2088.37	4.11	5.6
24	pcb.bmp	0.0919	25	7902.84	0.4994	26	7896.84	0.08	5.4
25	pendant.bmp	0.2125	20	11121.04	1.1137	20	11121.04	0.00	5.2
26	petals.bmp	1.0123	30	6591.94	4.1686	31	6590.75	0.02	4.1
27	rabbit.bmp	0.3105	25	2975.88	1.8959	25	2975.88	0.00	6.1
28	rice.bmp	0.0944	55	8872.80	0.4875	55	8872.80	0.00	5.2
29	ricefield.bmp	0.2501	55	4570.45	1.3518	57	4533.68	0.81	5.4
30	shamrock.bmp	0.2189	80	6089.21	0.7108	79	6085.97	0.05	3.2
31	ship1.bmp	1.038	55	11082.04	5.4114	58	11069.06	0.12	5.2
32	ship2.bmp	1.0289	40	11244.54	5.8308	40	11244.54	0.00	5.7
33	ship3.bmp	1.172	70	11234.51	5.725	72	11230.95	0.03	4.9
34	text.bmp	0.2886	50	6495.66	1.5196	50	6495.66	0.00	5.3
35	textbook.bmp	1.3162	90	2131.53	7.2764	90	2131.53	0.00	5.5
	Average							**0.40**	**5.3**

The Canny edge detection operator was developed by John F. Canny in 1986 and uses a multi-stage algorithm to detect a wide range of edges in images. It arises from the earlier work of Marr and Hildreth [14], who were concerned with modeling the early stages of human visual perception. His work is a gradient-based edge-finding algorithm that has become one of the most widely used edge detectors. This algorithm is known the optimal edge detector. In this situation, an "optimal" edge detector means following three criteria:

- Good detection: The algorithm should mark as many real edges in the image as possible.
- Good localization: Marked edges should be as close as possible to the edge in the real scene.
- Minimal response: A given edge in the image should only be marked once, and where possible, image noises should not create false edges.

Based on these criteria, the Canny edge detection process included the following stages:

- Noise removal: The canny edge detector smoothes the image to eliminate noise.
- Differentiation: It finds the image gradient in order to highlight regions with high spatial derivatives.
- Non-maximum suppression: The algorithm tracks along these already highlight regions and suppress any pixel that is not at the maximum.

(a) (b) (c) (d)

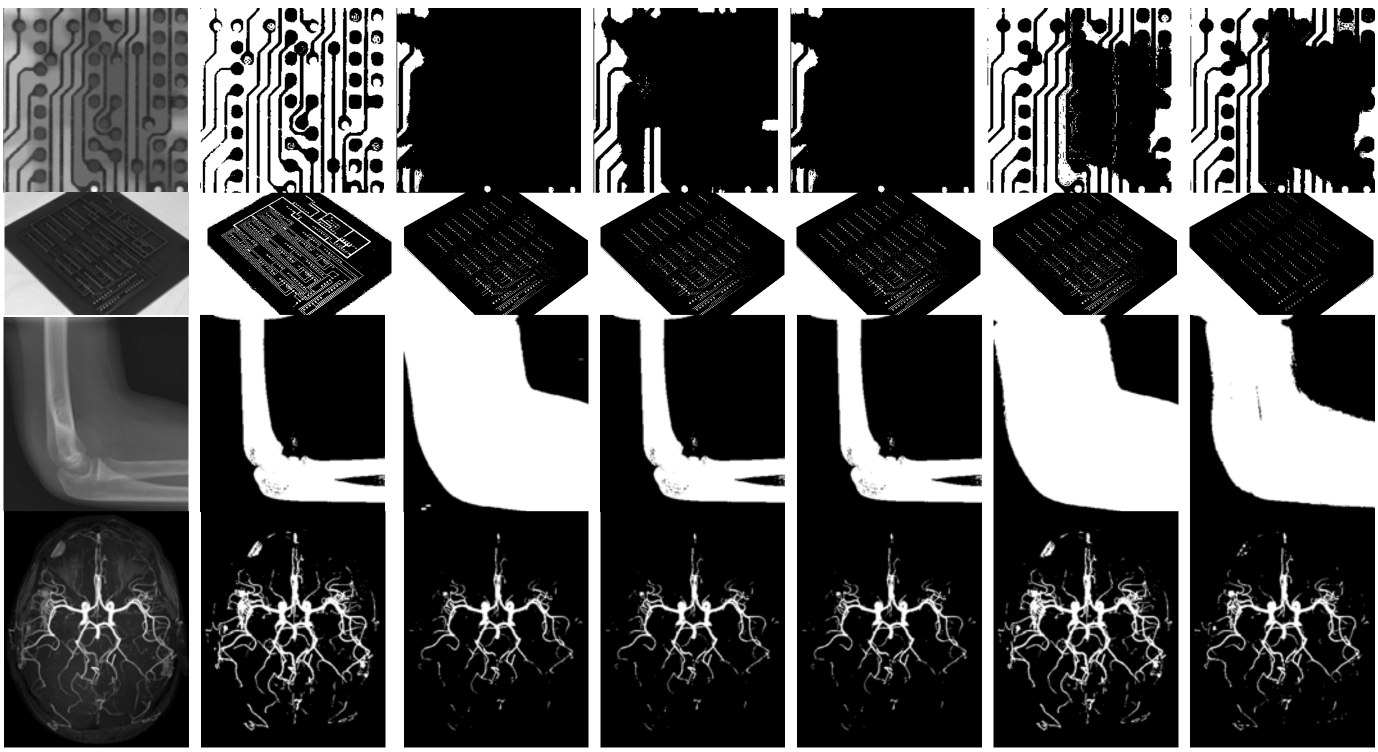
Figure 1. Some samples images from comparison of 5% and 1 % steps.
All images from top to bottom: original image, image at 1% step, and image at 5% step.
(a) petals.bmp, MSE difference 0.02%. (b) house.bmp, MSE Difference 0.48%.
(c) cameraman.bmp, MSE Difference 0.08%. (d) moon.bmp, MSE Difference 4.11%.

- Edge threshold, canny edge detector use a method called "hysteresis". The hysteresis method tracks along the remaining pixels that have not been suppressed. It uses two thresholds and if the gradient of the pixel is below the lower threshold, it is set to zero (regarded as a non-edge). If the gradient is above the higher threshold, it is set as an edge. If the gradient is between these thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above the higher threshold.

A widely used method for noise removal is the Gaussian filter, in which signals, in one and two dimensions, are smoothed out by the convolution of the image with a Gaussian kernel. The Gaussian operator is isotropic and therefore smoothes the image in all directions blurring sharp boundaries. All these approaches deal with the first derivatives of the image, thus slightly, but not totally, eliminate noises.

4. Hybrid Image Thresholding Method

The goal of Hybrid Image Thresholding method utilize image characteristics to assist the thresholding process by combining the P-tile method as a global thresholding method with an edge detector to retrieve shape information. By using an edge detector, information of object area ratio acquired is determined by the shape of objects. This information is useful especially for applications that need to preserve the shape of objects in the original image.



(a) (b) (c) (d) (e) (f) (g)

Figure 2. Some examples of edge detector selection results.

(a) Original Images. (b) Canny. (c) Prewitt. (d) Roberts. (e) Sobel. (f) LoG. (g) Otsu

Edge detectors are used to calculate the object area ratio by comparing the difference between edge map of the original image and edge map of the thresholded image. By trying all of the possible object area ratio value to threshold an image and comparing each of their respective edge maps to the edge map of the original image, the best estimate of the object area ratio value is determined as the value where the produced edge map that has the smallest difference to that of original image. We employ the MSE (Mean Squared Error) to calculate the difference between edge map of the thresholded image and edge map of the original image.

Let I be the original image and G be the threshold value being searched, the algorithm of Hybrid Image Thresholding method are as follow:

```
O ←EdgeMap(I)       # calculate Edge Map from I #
v ← initial_Value
e ← RealMax         # set e as maximum real value #
Loop until v = max_Value in Step increment.
    T ← P-tile(I,v)  #threshold I using P-tile method#
                     #and v as threshold value      #
    C ← EdgeMap(T) #calculate Edge Map from T #
    r ← MSE(O,C)     # calculate MSE value  #
                     # between O and C      #
    If  r < e        # if MSE value is smaller than e #
        e ← r        # replace e with MSE value #
        G ← v        # set v as the searched value #
```

This method is simple and suitable for all kind of edge detectors, since it only iterate in constant time (determine by *Step* value). It does not add anymore complexity to the P-tile method and the edge detector composing this hybrid approach.

5. Experiments

5.1. Finding the Practically Optimal Step

In the hybrid method algorithm above, the smaller the *Step* value, the more precise the threshold is, however, it requires more computational cost. Moreover, too precise setting of this value has little meaning since it is used by the P-tile method as the target percentage object ratio to extract the objects from the image, and it is not usually possible to extract the objects that have pixels whose number is exactly the same as the assigned percentage.

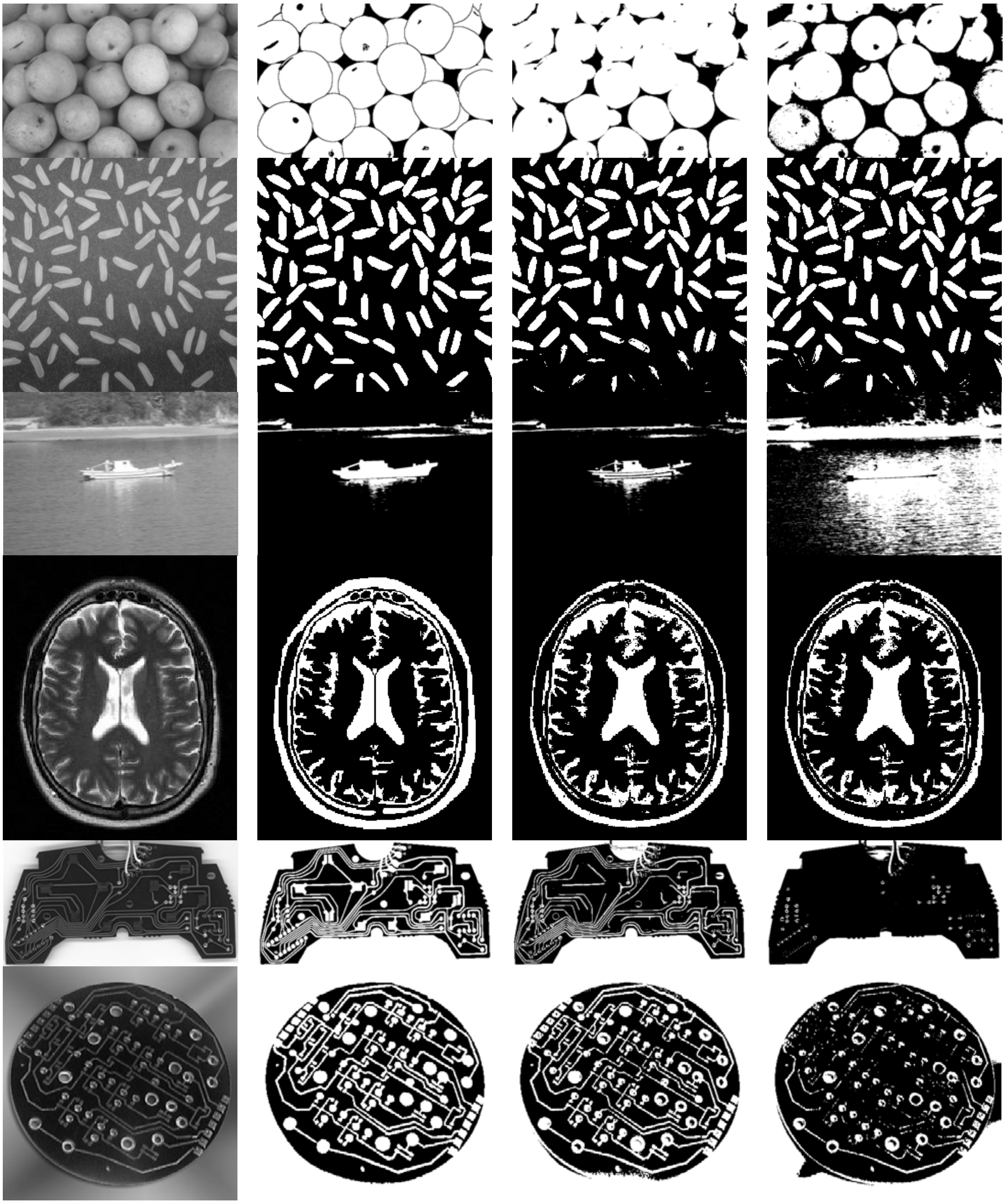
Table 2. Thresholding Performance Result

No	Name	MSE Hybrid	MSE Otsu	Difference	Ratio (Otsu/Hybrid)
1	MRI001.bmp	0.0882	0.0879	-0.34%	99.66%
2	MRI002.bmp	0.0610	0.0793	30.00%	130.00%
3	MRI003.bmp	0.0730	0.0723	-0.96%	99.04%
4	PCB001.bmp	0.0972	0.2250	131.48%	231.48%
5	PCB002.bmp	0.0225	0.1110	393.33%	493.33%
6	PCB003.bmp	0.0484	0.2790	476.45%	576.45%
7	PCB004.bmp	0.0541	0.1464	170.61%	270.61%
8	PCB005.bmp	0.0348	0.0628	80.46%	180.46%
9	PCB006.bmp	0.0579	0.2585	346.46%	446.46%
10	PCB007.bmp	0.0546	0.0841	54.03%	154.03%
11	PCB008.bmp	0.0472	0.1320	179.66%	279.66%
12	PCB009.bmp	0.0850	0.0935	10.00%	110.00%
13	PCB010.bmp	0.1308	0.3326	154.28%	254.28%
14	apples.bmp	0.0692	0.2168	213.29%	313.29%
15	bone.bmp	0.0077	0.3615	4594.81%	4694.81%
16	cell.bmp	0.0705	0.0868	23.12%	123.12%
17	moon.bmp	0.0170	0.0320	88.24%	188.24%
18	rice.bmp	0.0412	0.0303	-26.46%	73.54%
19	ship2.bmp	0.0222	0.1602	621.62%	721.62%
20	ship3.bmp	0.0229	0.4085	1683.84%	1783.84%
		Average		**461.20%**	**561.20%**

Since we need to find the practically optimal *Step* value which balances between the computational cost and the precision of thresholding, we made a preliminary experiment to find it. We tried our experiment by comparing between 1% and 5 % *Step* values which translates 99 possibilities (1%-99%) for 1% value and 19 possibilities (5%-95%) for 5 % value, respectively.

We performed the experiments on 40 images representing many kind of situations. All of them were thresholded using 5% and 1% *Step* value, converted them into gray level images using the P-tile method, and calculated the MSE between them and their respective original images. The results of these experiments are shown in Table 1.

Table 1 shows that the average MSE difference between 5% and 1% value is 0.4%. This value indicates that the quality of both images is almost similar. There are several cases where the results are exactly the same, denote by MSE difference of 0%. Some examples of these results are shown in Fig. 1. Table 1 also shows that the method in 5% case is 5.3 times faster than that in 1% case in average.



(a) (b) (c) (d)

Figure 3. Some examples of thresholding performance
(a) Original Images (b) Ground Truth (c) Hybrid (d) Otsu

5.2. Edge Detector Selection

In the hybrid method, we need to find the best edge detector to be combined with the P-tile method. We tried to combine the P-tile method with five kinds of edge detectors, Canny, Prewitt, Roberts, Sobel and Laplacian of Gaussian (LoG). The first four edge detectors are gradient based and the last one is Laplacian based.

We use three different scenarios representing different applications which are the extraction of the copper route from PCB images, the extraction of the bone from radiographs, and the extraction of object from MRI. In all of scenarios, shape information is needed to threshold images accurately. For each scenario, we employ 70, 40, and 25 images respectively. Some examples of the results are shown in Fig. 2.

According to the subjective evaluation of the results, we found that combining the P-tile method with Canny edge detectors produce the most stable result. This combination consistently produces images that have quality better than or equal to the others.

5.3. Performance of Thresholding

To measure the performance of the hybrid method by the combination of the P-tile method with Canny edge detector, we used 20 gray scale images which were selected images from those three scenarios used in edge detector selection with some additional images from experiments in Sec. 5.1 We manually converted these images into binary images and use these binary images as “ground truths”. The gray level images were thresholded using this combination and calculate the difference with the ground truth images using MSE to measure the fidelity of the images produced by the hybrid method. We also applied the same procedure using Otsu’s method, which is well-known and used as one of the standards, for comparison. The result of this experiment is shown in Table 2.

The result of this experiment shows that the hybrid method is better than the Otsu’s method in 17 out of 20 images. Two out of three images where Otsu’s method is better than the hybrid method, denoted by negative value in “Difference” column, the MSE difference is less than 1%, so it is safe to say that the performances of both methods on these two images are similar in these case.

The only case where Otsu performance is substantially better is on the image “rice.bmp”, shown in second row of Fig. 3, which contains many small grains of rice in different shapes and directions. In this case the shape information obtained by the hybrid method may be not sufficient. The resultant images are shown in Fig. 3.

6. Conclusions

We have proposed a hybrid method of image thresholding by combining the P-tile global thresholding method and Canny edge detector. The Experimental results show that in average the performance of this method is significantly better than Otsu’s. We are now working on the application of this method using dental panoramic radiographs.

References

[1] A.S. Abutaleb, “Automatic Thresholding of Gray-Level Pictures Using Two Dimensional Entropy”, *Computer Vision, Graphics, and Image Processing*, vol.47, pp.22-32, 1989.

[2] J. Kittler and J. Illingworth, “Minimum Error Thresholding”, *Pattern Recognition*, vol.19, no.1, pp.41-47, 1986.

[3] K.H. Liang and J.J.W Mao, “Image Thresholding by Minimizing the Measures of Fuzziness”, *Pattern Recognition*, vol.28, no.1, pp.41-51, 1995.

[4] N. Otsu, “A Threshold Selection Method From Gray-Level Histogram”, *IEEE Transactions on Systems, Man, and Cybernetics*, vol.9, pp.62-66, 1979.

[5] W. Doyle, “Operation useful for similarity-invariant pattern recognition”, *J. Assoc. Comput. Mach*, vol.9, pp.259-267, 1962.

[6] A.D. Brink and N.E. Pendock, “Minimum Cross-Entropy Threshold Selection”, *Pattern Recognition*, vol.29, no.1, pp.179-188, 1996.

[7] J.N. Kapur, P.K. Sahoo, and A.K.C. Wong, “A New Method for Gray-level Picture Thresholding Using the Entropy of the Histogram”, *Computer Vision, Graphics, and Image Processing*, vol.29, pp.273-285, 1985.

[8] N.R. Pal and S.K. Pal, “Entropic Thresholding”, *Signal Processing*, vol.16, pp.97-108, 1989.

[9] S.K. Pal and A. Rosenfeld, “Image Enhancement and Thresholding by Optimization of Fuzzy Compactness”, *Pattern Recognition Letters*, vol.7, pp.77-86, 1988.

[10] L. G. Roberts, "Machine perception of three-dimensional solids", in *Optical and Electro-Optical Information Processing*, J. T. Tippet et al., MIT Press, Cambridge, MA, pp. 159-197, 1965.

[11] J.M.S. Prewitt, “Object Enhancement and extraction”, in *Picture Analysis and Psycho-pictorics* B.S. Lipkin and A. Rosenfeld (Eds.), Academic Press, New York, 1970.

[12] I. Sobel, "Neighbourhood coding of Binary images for test contour following and general array binary Processing", *Computer Graphics Image Process*, pp.127-135, 1975.

[13] J. F. Canny, "A computational approach to edge detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.8, no.6, pp.679-698, 1986.

[14] D. Marr and E.C. Hildreth, "Theory of edge detection", *Proc. Roy. Soc. London*, B207, pp.187-217, 1980.

Febriliyan Samopa, received the bachelor degree in Computer Engineering from Sepuluh Nopember Institute of Technology Surabaya in 1997 and Master degree in Computer Science from University of Indonesia in 2001. During 1997-2001, he taught in Department of Informatics, Sepuluh Nopember Institute of Technology, Surabaya. During 2001-2006, he taught in Department of Information System, Sepuluh Nopember Institute of Technology, Surabaya. He is now a PhD student in Department of Information Engineering, Faculty of Engineering, Hiroshima University, Japan.

Akira Asano, graduated from the Department of Applied Physics, Osaka University in 1987. He received M. Eng. degree from Osaka University in 1989, and Ph. D. in applied physics from Osaka University in 1992. During 1992 – 1998, he worked as Research Associate, Department of Mechanical System Engineering, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Japan. During 1998 – 2005, he was Associate Professor at Faculty of Integrated Arts and Sciences / Graduate School of Engineering, Hiroshima University, Japan. He is now a Professor at Graduate School of Engineering / Faculty of Integrated Arts and Sciences, Hiroshima University, Japan.