Object Shape Recognition in Image for Machine Vision Application

Mohd Firdaus Zakaria, Hoo Seng Choon, and Shahrel Azmin Suandi

***Abstract*—Vision is the most advanced of our senses, so it is not surprising that images contribute important role in human perception. This is analogous to machine vision such as shape recognition application which is important field nowadays. This paper proposed shape recognition method where circle, square and triangle object in the image will be recognizable by the algorithm. This proposed method utilizes intensity value from the input image then thresholded by Otsu's method to obtain the binary image. Median filtering is applied to eliminate noise and Sobel operator is used to find the edges. Thinning method is used to remove unwanted edge pixels where these pixels may be counted in the parameter estimation algorithm, hence increase the false detection. The shapes are decided by compactness of the region. The experimental results show that this method archives 85% accuracy when implemented in selected database.**

***Index Terms*—Object area, object parameter, and shape recognition.**

I. Introduction

Machine vision is one of the applications of computer vision to industry and manufacturing, whereas computer vision is mainly focused on machine-based image processing. Machine vision usually requires additional digital input or output devices and computer networks to control other manufacturing equipment such as robotic arms. Machine vision is subfield of engineering that encompasses computer science, optics, mechanical engineering and industrial automation. One of the most common applications of machine vision is the inspection of manufactured goods such as semiconductors chips, automobiles, foods and pharmaceuticals. Just like human inspectors working on assembly lines using their vision to inspect part visually to judge the quality of workmanship, machine vision systems use input device such as camera and image processing software to perform similar inspections.

Machine vision systems are programmed to perform narrowly defined tasks such as shape recognition on a conveyor, reading serial numbers and searching for surface defects. The interaction between human and machine typically consists of programming and maintaining the machine by the human operator. As long as the machine acts out preprogrammed behavior, a direct interaction between man and machine is not necessary anyway. However, if the machine is to assist a human such as in complex assembly operation, it is necessary to have means of exchanging information about the current scenario between man and machine in real time. The problem cannot be solved if the operator needs to type in the object's coordinates or move the mouse pointer to an image of the object on a screen to enable the machine to detect the objects present in the conveyor. As a result, the machine needs to be equipped with a camera so that it will use the image captured to do further processing and identify types of shape on the conveyor.

There are several methods that have been developed by the past researchers for the shape detection such as using generalizes Hough transform [1]–[3], template matching [4], [5] etc. However, both mentioned methods are sensitive to noise and sampling artifact. To overcome this problem, M. Kass proposed active contour models [6], [7] but this method suffers from complexity and high computational time.

As being described above, this paper proposed a method for shape recognition especially for object on the conveyor with simple algorithm with low computational time. This proposed method used intensity value from the input image which is then threshold by Otsu's method to obtain the binary image. Otsu's method selects the threshold automatically from the grayscale histogram and the thresholded image contains two regions, i.e., foreground and background. Median filtering is applied to eliminate noise and Sobel operator is used to find the edges. Thinning method is used to remove unwanted edge pixels where these pixels may be counted in the parameter estimation algorithm, hence increase the false detection. The shapes are decided by compactness of the region. The experimental results show that this method archives 85% accuracy when implemented in selected database.

The rest of the paper is organized as follows: Section II presents details explanations of the proposed method. Section III will show the results and discussions and finally conclusion in section IV.

II. Proposed Method

Fig. 1 shows the block diagram of the proposed method. The input image taken by the input device is first converted to hue, saturation, and lightness (HSL) color space where only L value will be processed. The processed L component will be used as template to determine the shape to produce the final output.

A. Color Space Conversion

In the proposed method, HSL color space is chosen and only one channel, L will be processed instead of using three channels as in RGB color space. The advantages of using one

Manuscript received on October 20, 2010; revised February 5, 2011. This work is supported in part by the Universiti Sains Malaysia Postgraduate Incentive Research Grant No. 1001/PELECT/8021023.

Mohd Firdaus Zakaria and Shahrel Azmin Suandi are with Intelligent Biometric Group, School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Engineering Campus, 14300 Nibong Tebal, Seberang Prai Selatan, Penang, Malaysia (e-mail: firdauszakaria@gmail.com, shahrel@eng.usm.my).

color channel instead of three channels are the processing time and complexity can be reduced significantly. The L value contains lightness value of the input image where L is calculated as shown in Eq. (1).

$$L = \frac{max(R,G,B) + \min(R,G,B)}{2} \quad (1)$$

where, L is the lightness value, R is the red channel of the input image, G is the green channel of the input image and B is the blue channel of the input image.

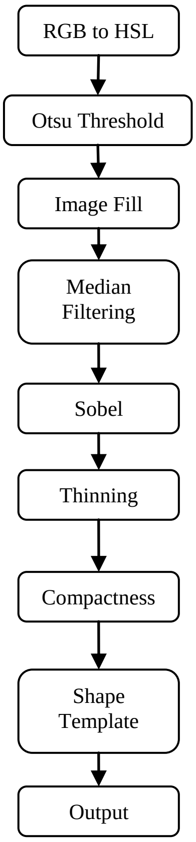


Fig. 1. Block diagram of the proposed method

Fig. 2 shows the conversion of the input image in red, green, blue (RGB) color space to L channel in HSL color space. The L image produces good color separation between the object and its background.

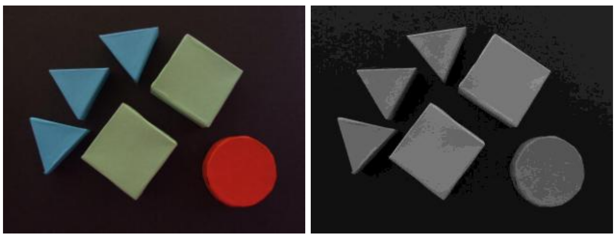


Fig. 2. *RGB* color space (left) to *L* channel (right image) of HSL color space conversion result

B. *Otsu's Threshold*

Otsu's threshold [8] is a method that selects a threshold automatically from a gray level histogram. In this method, it is important to select an adequate threshold of gray level to extract the object from their background. In an ideal case, the histogram has a deep and sharp valley between two peaks representing object and background, respectively, so that the threshold can be chosen at the bottom of this valley as proposed by Prewitt and Mendelsohn [9]. However, for most real image, it is usually difficult to detect the bottom valley precisely, especially in such cases as when the valley is flat and abroad, imbued with noises or when the two peaks are extremely unequal in height, often producing no traceable valley.

Otsu's method is nonparametric and unsupervised method of automatic threshold selection for image segmentation. An optimal threshold is selected by the discriminant criterion [10], namely, so as to maximize the separability of the resultant classes in gray level. The procedure is simple, utilizing only the zero and the first-order cumulative moments of the gray level histogram.

There are three types of discriminant criteria and the one used in this paper to obtain an optimal threshold value is shown in Eq. (2).

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2} \quad (2)$$

where λ is the measure of separability of the resultant classes in gray levels, σ_B is between-class variance and σ_W is within-classes variant.

The value of λ must be maximized to obtain a suitable threshold value. The optimal threshold value is the one that maximizes the between-classes variance, σ_B or conversely minimizes the within-classes variance, σ_W. This directly deals with the problem of evaluating the goodness of threshold.

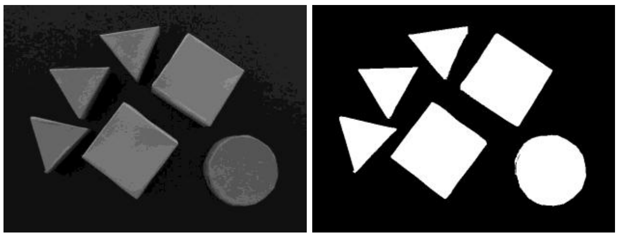


Fig. 3. Otsu's threshold

In Fig. 3, by using the Otsu's method, the binary image clearly shows the differences between the object and background. The objects are marked with one while the background is marked with zero value.

C. *Image Fills*

Image fills is a function to fill the 'holes' in the binary image of the input image. This method is suitable to eliminate the noise that exists in the image.

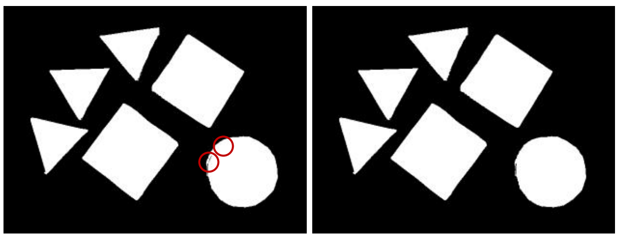


Fig. 4. Image fills function

The small circles in Fig. 4 depict the 'holes' in the input image. By implementing image fills algorithm, the 'holes' region will be converted to neighboring value hence eliminate the noise.

D. Median Filtering

Median filtering [11] is usually used to reduce 'salt and pepper' noise and preserve edges. In the proposed method, the size for median filter operator is set to 10×10 matrixes.

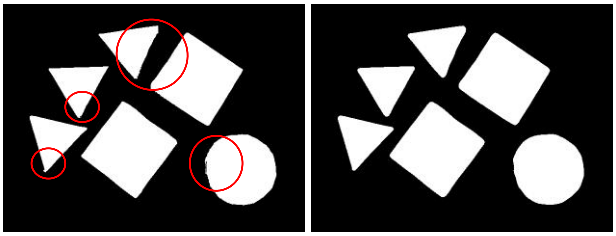


Fig. 5. 10 ×10 median filtering results

Fig. 5 illustrates the effect of median filtering. From the output image, noise has been reduced to minimum and some edges also been smoothed. This process is essential to make sure all corresponding edges for each object are connected properly so that the perimeter can be computed appropriately.

E. Sobel Operator

Sobel operator [11] is an operator used in image processing, particularly for edge detection algorithm. Actually, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. It is also two dimensional map of gradient at each point and can be processed and viewed as if it itself an image, with the area of high gradient or the likely edges visible as white lines. In the proposed method, Sobel mask is used to detect the shape's outer edges. The outer edge of each shape is needed to compute the perimeter of each shape. The perimeter is obtained by counting the total white pixels in the edge of a shape. At each image point, the gradient vector of the Sobel mask points increases in the direction of largest possible intensity.

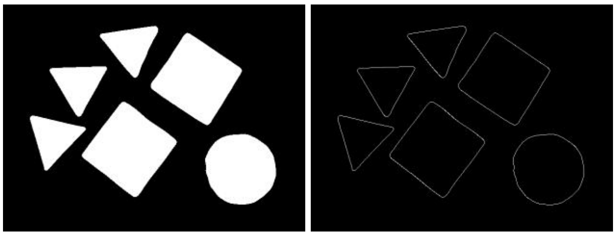


Fig. 6. Edge detection results using Sobel operator

Fig. 6 demonstrates the edge detection by Sobel operator. The convolution between the Sobel operators with input image will produce edge, i.e. pixel values equal to one, where same value region will produce zeros and otherwise will produce ones.

F. Thinning

The morphological thinning operator is the subtraction between the input image and the sub generating operator with structuring A and B. Both structuring elements will be rotated 90° for four times. This means that there will be eight structuring elements. The result will be the input image with pixels in which its center contains the pattern specified by A and B marked as zero. This operation removes pixels which satisfy the pattern given by the structuring elements A and B [12].

Fig. 7 shows the effect of thinning process. Thinning is needed here because there will be an increase of pixel count if the arrangement of the pixels is not in a straight line.

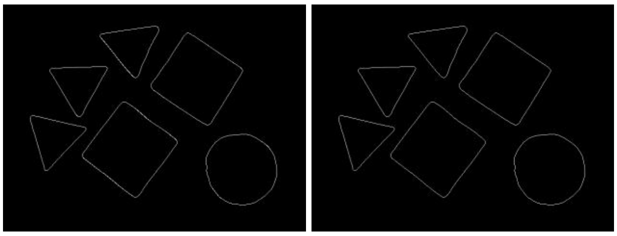


Fig. 7. Images before (left) and after (right) thinning process

G. Shape Recognition

The proposed method recognizes the shapes of an object by computing the compactness [13]. Eq. (3) shows the equation for compactness calculation.

$$c = \frac{p^2}{A} \quad (3)$$

where c is the compactness, c is the perimeter and A is the area.

Computing c like this is applicable to all geometric shapes, independent of a scale and orientation and its value is dimensionless. In the proposed method, according to compactness value, circle has compactness in the range of 1 to 14, square's compactness is from 15 to 19 and triangle's compactness is from 20 to 40.

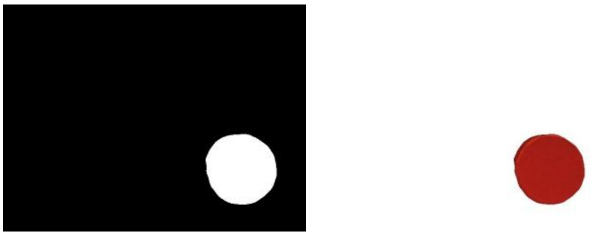


Fig. 8. Circle template and circle detection output

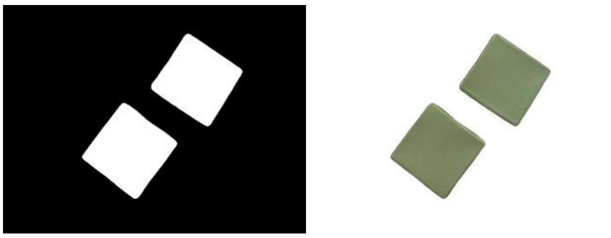


Fig. 9. Square template and square detection output

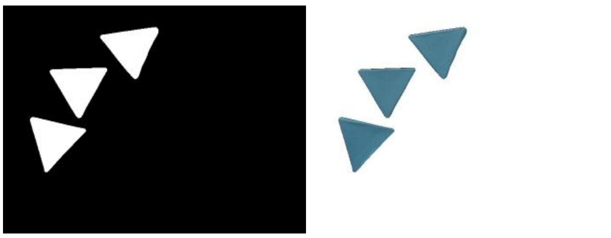


Fig. 10. Triangle template and triangle detection output

Fig. 8, Fig. 9 and Fig 10 depict the template of corresponding shape of circle, square and triangle, respectively. This template is determined by compactness value, c and applied on the input image in RGB color space to produce the output image.

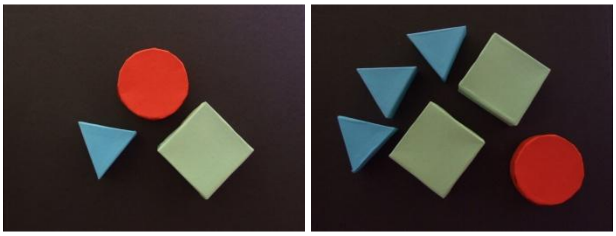
III. Results and Discussions

The proposed method is tested on a database consists of 70 images with size 640×480. This dataset can be divided into four groups which are dataset that contains only one object, three same objects, three different objects and multiple different objects. Fig. 11 illustrates the categories in the dataset and Table I shows the corresponding results.

TABLE I: Proposed Method Accuracy

Number of Objects	Objects Shape	Number of Objects	Accuracy %
One Object	Circle	3	100
	Square	7	100
	Triangle	4	100
Three Same Objects	Circle	6	50
	Square	9	67
	Triangle	7	86
Three Different Objects	Circle Square Triangle	18	100
Multiple Different Objects	Circle Square Triangle	16	75

(a) One object (b) Three same object



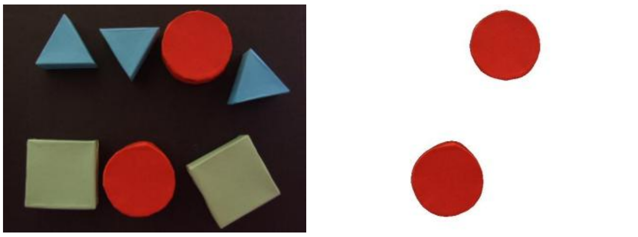
(c) Three different objects (d) Multiple different objects

Fig. 11. Four categorizes dataset

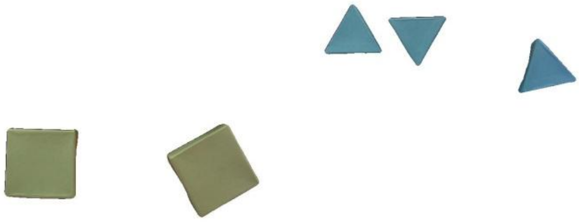
Fig. 12 shows the example of successful detection by using the proposed method. Fig. 13 demonstrates the example of incorrect detection of the proposed method. There are several reasons why the proposed method produced undesirable detection:

- Due to the input image has uneven intensity, the image is not thresholded properly and thus the shapes cannot be detected.
- Some of the objects are touching each other which contribute to inaccurate calculation in the parameter and area estimation.
- Noises not totally eliminated where these noises will be detected as objects.

Fig. 14 depicts the advantages of using HSL color space to obtain the *L* channel over using typical grayscale level from the RGB color space.

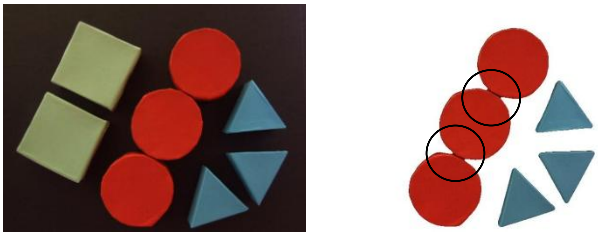


(a) Input image (b) Circles detection



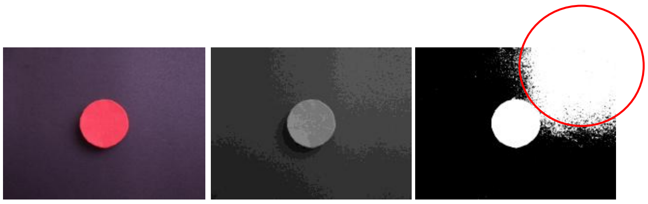
(c) Squares detection (d) Triangles detection

Fig. 12. Example of successful detection

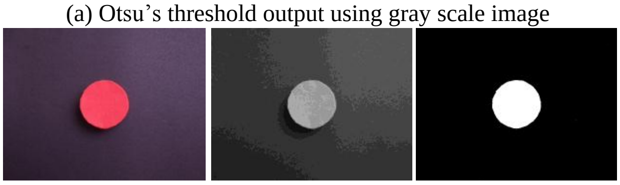


(a) Input image (b) Triangles detection

Fig. 13. Example of inaccurate detection



(a) Otsu's threshold output using gray scale image



(b) Otsu's threshold output using lightness channel

Fig. 14. Advantages of HSL color space

IV. Conclusion

Shapes detection method has been proposed in this paper. Its main objective is to differentiate basic shape such as circle, square and triangle in the given input image by merely employing computer vision techniques. This method utilize compactness as the shape indicator where the compactness for circle is fixed from 1 to 14, square's compactness is in range 15 to 19 and triangle's compactness is from 20 to 40. From the result in the Section III, the proposed method achieved 85% detection accuracy in the selected database. However, this method is sensitive to noise and lighting condition. Poor lighting condition image will bring complexity in Otsu's threshold algorithm and the outcome of the result is not desirable.

REFERENCES

[1] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Comm. ACM*, vol. 15, pp. 11–15, 1972.

[2] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[3] D. Shi, L. Zheng, and J. Liu, "Advanced hough transforms using a multilayer fractional fourier method," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1558–1566, 2010.

[4] J. P. Lewis, "Fast Template Matching," in *Proc. of Canadian Image Processing and Pattern Recognition Society*, Quebec, 1995, pp. 120-123.

[5] R. Brunelli, "Template matching techniques in computer vision: theory and practice," *Wiley*, ISBN 978-0-470-51706-2, 2009.

[6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*. vol. 1, no. 4, pp. 321–331, 1987.

[7] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[8] N. Otsu, "A threshold selection method from gray-level histogram," *IEEE Transaction on Systems, Man and Cybernatics*, vol. 9, no. 1, pp. 62–66, 1979.

[9] J. M. S. Prewitt and M. L. Mendelsohn, "The analysis of cell images," *Annals of the New York Academy of Sciences*, vol. 128, pp. 1035–1053, 1966.

[10] K. Fukunage, *Introduction to Statistical Pattern Recognition*, New York: Academic Press, pp. 225–257, 1972.

[11] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, 2nd ed., New Jersey: Prentice Hall, 2002.

[12] V. E. Duro, "Fingerprints thinning algorithms," *IEEE Aerospace and Electronic System Magazine*, vol. 18, no. 9, pp. 28–30, 2003.

[13] M. Pomplun. [2007] Compactness. [Online]. Available: http://www.cs.umb.edu/~marc/cs675/cv09-11.pdf.