Robotic Arm Grasping and Placing Using Edge Visual Detection System

Guo-Shing Huang, Hsiung-Cheng Lin, and Po-Cheng Chen

Department of Electronic Engineering, National Chin-Yi University of Technology
Taichung 41101, Taiwan, ROC
hgs@ncut.edu.tw hclin@ncut.edu.tw then_if@hotmail.com

Abstract—**In recent years, the research of autonomous robotic arms has received a great attention in both academics and industry. Therefore, this paper aims to develop the object visional detection system that can be applied to the robotic arm grasping and placing. The proposed scheme can accurately measure the relative distance between the object and robot arm using the edge detection algorithm with a camera device. Its process involves the color differentiating of input visual from the camera to locate the candidate block of fitting color requirements and thereafter define the sector. Subsequently, this visual system undergoes morphological binarization processing and convex hull calculation to find out the target's outline, and then it can apply the shape matching module to locate the target in vision. Taking visual proportion into consideration, the relative distance between target and robot arm is calculated for the arm to grip the target so that the arm is able to place object into a paper holder, where the vision of paper holder is obtained to detect its shape of hole. The experimental results have confirmed that the self-controlled robotic arm can put the object into the paper holder successfully.**

Keywords- Visual Recognition, Edge Detection Algorithm, Colour Differentiation, Convex Hull Calculation, Morphological Processing, Shape Matching, Robotic Arm

I. INTRODUCTION

Most robotic arm using vision techniques have been used in industrial areas such as automated assembly [1]. In recent years, the number of competitions about robotic arm design or performance is increasing significantly. However, the applications such as the competitions did not yet get into further discussions. One of the big challenges in industry is to reduce the cost that is not necessarily considered by academics. In this study, we attempt to design both low-cost and practical self-control system for the robotic arm, not only from the view of academics but also industry. Based on the "grasping color ball" competition of robotic arm, the study has developed the Edge Visual Detection System for the manipulation of robotic arm in order to grasp and place the object accurately.

This paper is divided five sections. Section II describes the system structure which includes a robotic arm structure and control method. Section III demonstrates the image processing technique. Section IV describes the relationship between shape matching and robotic arms control. The conclusions are given in section V.

II. SYSTEM STRUCTURE

The proposed vision system for the robotic arm control is divided into two parts: grasping and placing. In Fig. 1, firstly input the image for the image processing and then shape matching check if there is a ball. If yes, the robotic arm can move the camera lens pointing to the center of the ball and then grasp it. Finally, the ball can be placed the assigned box according to the ball color. The same procedure will be repeated once all steps are complete.

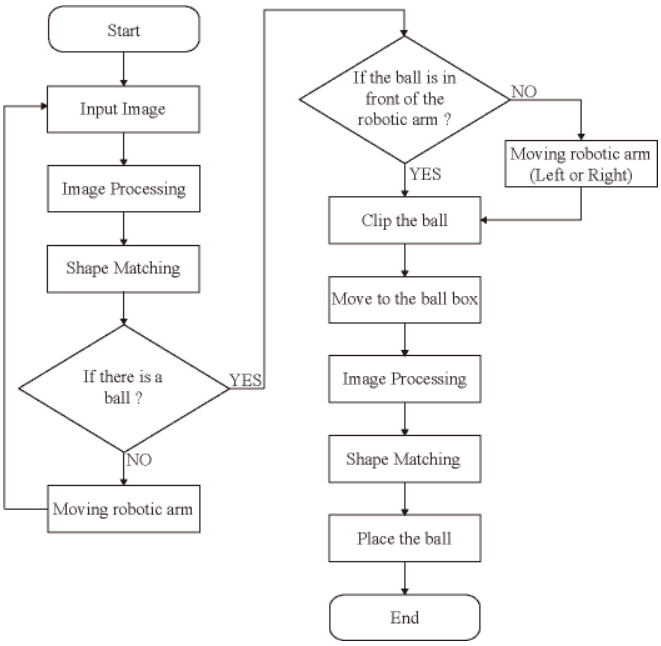


Figure 1. Flow chart of the proposed system

The robotic arm in the study uses seven RX-64 servomotors and line moving rail, as shown in Fig. 2. The webcam with the resolution of 320 × 240(pixel) is installed at the upper location of arm clip.

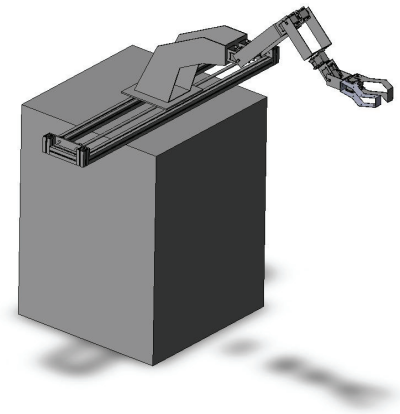


Figure 2. Structure of robotic arm

III. IMAGE PROCESSING

In the image processing, the noise, light and other interferences should be reduced in order to improve the shape matching error after capturing images. For details, it is shown in following procedures.

A. Light compensation

The image process requires the light compensation due to the dramatic impact by the light status. Many related publication about this issue may be found in the literatures [1-15]. For instance, Hsu et al. proposed a RGB color normalizing compensation method. We use "reference whit" method [2][3][4] for the light compensation and the flow chart in Fig. 3.

.

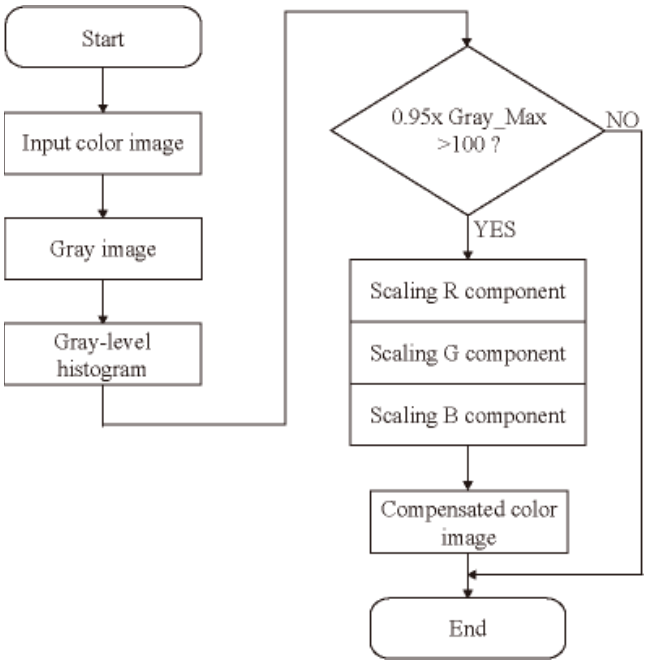


Figure 3. Light compensation module

The Gray_Max is the gray maximum in the image. If 0.95×Gray_Max of reference white is more than 100 in image, then the light compensation is necessary [4]. The light compensation is shown as follows (1) and (2):

$$AveR = \frac{1}{N}\sum_{i,j\in RW} I_R(i,j)$$
$$AveG = \frac{1}{N}\sum_{i,j\in RW} I_G(i,j) \quad (1)$$
$$AveB = \frac{1}{N}\sum_{i,j\in RW} I_B(i,j)$$

$$R_O(i,j) = I_R(i,j)\times\frac{255}{AveR}$$
$$G_O(i,j) = I_G(i,j)\times\frac{255}{AveG} \quad (2)$$
$$B_O(i,j) = I_B(i,j)\times\frac{255}{AveB}$$

B. Color segmentation

In some applications, the brightness of the impact of changes is critical. Therefore, in this study we convert the RGB color model to HSV color model. *H* is not susceptible to the brightness change. The transformation formula is shown in (3).

$$H1 = \cos^{-1}\left\{\frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right\}$$

$$H = H1 \quad \text{if } B \leqq G$$

$$H = 360° - H1 \quad \text{if } B > G \quad (3)$$

$$S = \frac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)}$$

$$V = \max(R,G,B)$$

where *S* (Saturation) values range from 0 to 1, *V* (Value) values ranged from 0 to 255.

In the HSV model, *H* value in 10°、50°、130° and 192°, can be split out of red, yellow, green, blue four colors.

C. Noise filter

Before doing binarization, we found that the image has a lot of noise outside in the non-target areas. This will affect the

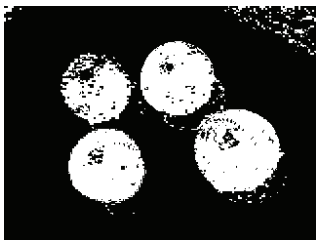
edge detection results. Consequently, we divided the image a new image$_{(t)}$ and the old image$_{(t-1)}$. The formula is shown in (4).

$$Image(x,y) = Image_{(t)}(x,y) \wedge Image_{(t-1)}(x,y) \quad (4)$$

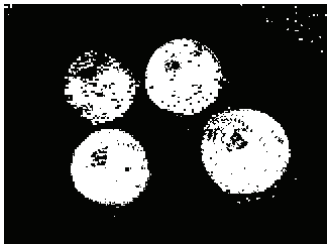
where image$_{(t-1)}$ is a former second image.

The object will remain in the image and the noise will appear as flashes of instability. The image noise can be filtered out efficiently, as shown in Fig. 4. Fig. 4(a) indicates the color segmentation image, and Fig. 4(b) shows the image before filtering. The filtered image is presented in Fig. 4(c). We see that the noise in the upper right and the ball around are successfully filtered out.

(a) The color segmentation image



(b) The image before filtering



(c) Filtered image

Figure 4. Noise filter results

D. Morphology

In order to delete some remaining little points after a preliminary image filtering, we use the dilation and erosion operations to filter it out, as follows [5][6].

Step 1: Erosion Operation

The erosion mask is showed in Fig. 5. Firstly justify if the mask element e is at 1 status in the binary image. If yes, continue to check if at least one of the surrounding elements (e1~e8) has a value in its respective location of A matrix. If the condition is satisfied, "e" is set as 1. Otherwise "d" is set as 0. The formula is shown in (5), and the transformed result is shown in Fig. 6. [7].

$$e = e1 \cap e2 \cap e3 \cap e4 \cap e5 \cap e6 \cap e7 \cap e8 \quad (5)$$

e1	e2	e3
e8	e	e4
e7	e6	e5

Figure 5. 3×3 erosion mask E

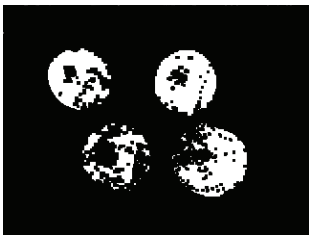


Figure 6. Eroded image

Step2: Dilation Operation

The erosion mask showed in Fig. 7. Firstly justify if the mask element d is at 1 status in the binary image. If yes, continue to check if at least one of the surrounding elements (d1~d8) has a value in its respective location of A matrix. If the condition is satisfied, "d" is set as 1. Otherwise d is set as 0. The formula is shown in (6), and the transformed result is shown in Fig. 8 [7].

$$d = d1 \cup d2 \cup d3 \cup d4 \cup d5 \cup d6 \cup d7 \cup d8 \quad (6)$$

d1	d2	d3
d8	d	d4
d7	d6	d5

Figure 7. 3×3 dilation mask *D*

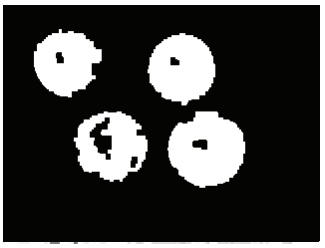


Figure 8. Dilated image

E. Image filling out

The ball has itself lines so that we filled it out within the first ball and placed in the process label. The label can be used as the basis of phase-out. The ball appears only one ball in the lens, and the result is shown in Fig. 9.

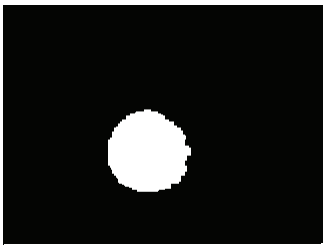


Figure 9. Yellow ball left to be chosen

F. Edge Detection

By edge detection of objects, we can find out the approximate shape and reduce the computational procedure. The Gradient method was applied in the image processing [8]. This paper applied Sobel Operator for x and y differential calculation with mask method, shown in Fig. 10. Then, ∇f can be obtained. The detection result is shown in Fig. 11.

$$\nabla f = \sqrt{\left(\nabla_x f\right)^2 + \left(\nabla_y f\right)^2} \tag{7}$$

-1	-2	-1
0	0	0
1	2	1

(a) Grey change of y direction

-1	0	1
-2	0	2
-1	0	1

(b) Grey change of *x* direction

Figure 10. Sobel mask.

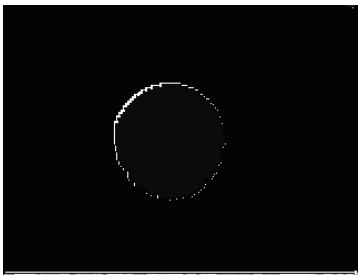


Figure 11. Edge detection

G. *Convex hull algorithm*

Here we got the outline of the ball by Convex hull algorithm. Search for a boundary point and the clockwise or counterclockwise around the circle outward. When the search point is covered, all points on the plane are the most peripheral points covered. The formula is shown in (8) [9-15], and the result is shown in Fig. 12.

$$S := \bigcap_{\substack{X \subseteq K \subseteq V \\ K \ \ is \ \ convex}} K \tag{8}$$

To show that the convex hull of a set X in a real vector space V exists, notice that X is contained in at least one convex set, and any intersection of convex sets containing X is also a convex set containing X. It is clear that the convex hull is the intersection of all convex sets containing X. This can be used as an alternative definition of the convex hull. The formula is shown in (9)

$$S := \left\{ \sum_{j=1}^{n} t_j x_j \middle| x_j \in X, \sum_{j=1}^{n} t_j = 1, t_j \in [0,1] \right\} \tag{9}$$

Where n is an arbitrary natural number, the numbers t_j are non-negative and sum to 1, and the points x_j are in X.

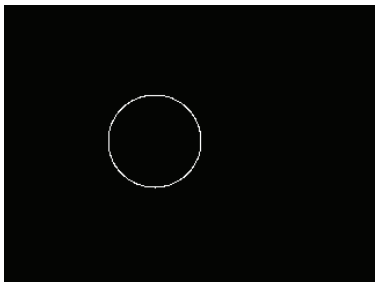


Figure 12. Convex hull image

IV. SHAPE MATCHING

Shape matching confirmed whether there was a complete ball in front of the camera. After computing the convex hull of the image, a clear boundary contour can be obtained, and then the ball shape based on the image data base can be matched successfully, as shown in Fig. 13.

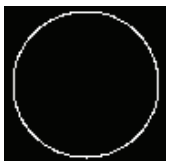


Figure 13. 89×85 ball shape

When the shape matching is successful, it means the ball appears in the lens as shown in Fig. 14. As a result, the robotic arm can move approaching to the ball and thus clip it to the assigned box. If the shape matching is not successful, the robotic arm will continue to search the ball, moving to the left or right, until the ball is found.

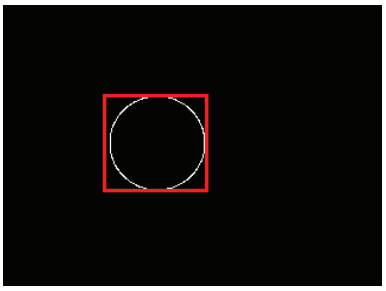


Figure 14. Successfully shape matching

By success matching, the robotic arm can be judged automatically to a correct gripping position. In this study, grasping and placing of robotic arm were based on the pre-training process. The ball can be placed to the desired box through the same image processing steps. Therefore, the robotic arm can be moved to the front of the box and then the hole on the box can be matched. Finally, the ball can be placed to the correct box hole accurately.

V. EXPERIMENTAL RESULTS

In this section, we present the results of some experiments as shown in Table I. We carried out the grabbing experiments using four kinds of color (red, yellow, green, and blue) ball for each one, each two, and each three, respectively, indoor and the hall under the different environment.

When the number of balls increases, light reflection will affect the recognition rate of color ball between the ball and ball. The blue color closes to the green one in this experiment. Although it can be obtained higher success rate under plenty of lighting indoor than less bright hall, in general speech, the less bright hall has also a good success rate. In this experiment, we do not specifically ask what kind of environment, but a basic success rate which we hope is required.

TABLE I .Success rate of correct grabbing ball under different number of color balls and environment

Number of balls		4	8	12
Accuracy (%)	Indoor	95%	88%	83%
	hall	87%	82%	78%

VI. CONCLUSIONS

Borland C++ Builder 6 programming was used for the image processing in the control of robotic arm for grasping and placing. By self-control, we have accomplished three tasks, i.e., "find the ball", "clip the ball" and "put the ball". More importantly, this system is a really streamlined and inexpensive system. In the future work, it is recommended to focus on finding, grasping and placement of objects in different shapes and colors. In fact, this topic may change the profile in the future robotic arm competition globally.

ACKNOWLEDGMENT

This work was supported by the National Science Council of Taiwan R.O.C. under grant NSC 97-2221-E-167-018- MY3.

REFERENCES

[1] Lledó Museros, Maria Teresa Escrig, "Automating Assembly of Ceramic Mosaics using Qualitative Shape Matching," Intelligent Robots and Systems , pp 4096 - 4101, 2007.

[2] J. J. Yang, "Automatic Facial Feature Extraction using Webcam," Chung Yuan Christian University, Taoyuan, Taiwan, June, 2007.

[3] Y. A. Pan, "Automatic Facial Expression Recognition System in Low Resolution Image Sequence," National Cheng Kung University, Tainan, Taiwan, July, 2004.I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, pp. 271–350, 1963.

[4] J. N. Cai, "Face Detection in Color Images Using Wavelet Neural Networks," Chaoyang University of Technology, Taichung, Taiwan, June, 2004.

[5] T. X. Huang, "A Smart Digital Surveillance System with Face Tracking and Recognition Capability," Chung Yuan Christian University, Taoyuan, Taiwan, June, 2004.

[6] Y. Z. Zeng, "DSP Based Real-Time Human Face Recognition System," National Sun Yat-sen University, Kaohsiung, Taiwan, June, 2005.

[7] S. D. Ren, "A Mobile Robot with Stereo vision Range Estimation," National Taiwan University of Science and Technology, Taipei, Taiwan, July, 2007.

[8] Y. Z. Lin, "Binocular Vision Based Human Head Tracking," National Taiwan University of Science and Technology, Taipei, Taiwan, June, 2008.

[9] R. A. Jarvis, "On the identification of the convex hull of a finite set of points in the plane," Information Processing Letters, vol. 2, no. 1, pp. 18–21, 1973.

[10] W. F. Eddy, "A new convex hull algorithm for planar sets," ACM Transactions on Mathematical Software, vol. 3, no. 4, pp. 398–403, 1977.

[11] A. Bykat, "Convex hull of a finite set of points in two dimensions," Information Processing Letters, vol. 7, no. 6, pp. 296–298, 1978.

[12] P. J. Green and B. W. Silverman, "Constructing the convex hull of a set of points in the plane," The Computer Journal, vol. 22, no. 3, pp. 262–266, 1979.

[13] F. P. Preparata and M. I. Shamos, Computational Geometry: An Introduction. Springer, 1985.

[14] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," ACM Transactions on Mathematical Software, vol. 22, no. 4, pp. 469–483, 1996.

[15] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," Information Processing Letters, vol. 1, no. 4, pp. 132–133, 1972.