VISION BASED EXTRACTION OF GRASPING REGION OF OBJECTS

[1]SAINT SAINT PYONE, [2]WUT YI WIN, [3]AUNG MYAT SAN

[1,2,3]Mechatronic Engineering Department, Mandalay Technological University The Republic of Union of Myanmar
Email: [1]saintsaint85@gmail.com, [2]wutyiwin@gmail.com, [3]agmyatsanmc@gmail.com

Abstract: Many applications of robotics include the grasping and manipulation of objects. Working in assembly robotic environments, the robot has to accurately not only locate the part but also to recognize it in readiness for grasping. This paper is mainly focused on the extraction of grasping region that is robust to changes in appearance of objects that have different shapes, sizes by using predefined rules. The classification performance of Backpropagation Neural Network (BPNN) classifier based on PCA features for grasping objects are evaluated. The experiments are carried out by using MATLAB programming language. The proposed system has been tested successfully to a dataset of 200 images for seven type and achieves good classification accuracy for 2D images. This system can examine types of the hand tools. It can also be evaluated the size of tool and the locations of the tool at which to grasp the object. This system can be easily adopted to grasp a number of common hand tools with large, medium and small size such as wrenches, screw drivers, brushes, washers and nails and hex keys, etc.

Keywords: Appearance, Backpropagation Neural Network, classification, grasping region.

I. INTRODUCTION

Grasping and assembly operations using industrial robots is currently based on the accuracy of the robot and the precise knowledge of the environment. Even though grasping has been a central topic in robotics for decades, robots still have great difficulty to pick up arbitrary objects while assembling complex machinery by using tools. Image-based object recognition system is extensively used in many industries for identification, inspection, grasping and categorization purposes [1]. Visual perception involves mapping pixel values and light information onto a model of the universe to infer surroundings. General scene understanding requires complex visual tasks such as segmenting a scene into component parts, recognizing what those parts are, and disambiguating between visually similar objects. Due to these complexities, visual perception is a large bottleneck in real robotic systems.

General purpose robots need the ability to interact with and manipulate objects in the physical world. Humans see novel objects and know immediately, almost instinctively, how they would grab them to pick them up. Robotic grasp detection lags far behind human performance. Grasps are classified taking into account size and shape characteristics of the grasped object and the required precision and stability for the task. That is, the grasp method is decided by both the shape of the object and the purpose of grasp. In this paper, we assume that the purpose of grasp is to carry an object. In other words, neither the characteristics of the object nor the usage are taken into consideration.

This paper is addressed on the problem of finding a good grasp given an image of the object. The rest of this paper is organized as follows. Section II will discuss background theories of the proposed method. The proposed technique is presented in Section III. The experimental results of proposed method are provided in Section IV. Section V concludes the paper.

II. RELATED WORKS

Recent work on grasp detection requires full knowledge of 2D or 3D models of objects. Based on this information, methods such as one based on the mechanics of grasping and the finger-object contact interactions [2] and [3] focus on designing control and planning algorithms to achieve successful and stable grasps.

Significant past work uses 3-D simulations to find good grasps [4], [5], [6] and [7]. These approaches are powerful but rely on a full 3-D model and other physical information about an object to find an appropriate grasp. Full object models are often not known a priori. General purpose robots may need to grasp objects without first building complex 3-Dmodels of the object. In real-world grasping, the full 3D shape of the object is hard to perceive. Some early work considers only objects of simple shapes to address this problem. For example, Miller et al. [8] used heuristic rules to generate and evaluate grasps for three-fingered hands by assuming that the objects are made of basic shapes such as spheres, boxes, cones and cylinders, each with pre-computed grasp primitives. Other methods focus on grasping 2D planar objects using edges and contours to determine form and force closure. Bowers and Lumia [9] also considered grasping planar objects by classifying them into a few basic shapes, and then used pre-scripted rules based on fuzzy logic to predict the grasp. Saxena et al. [10] showed that a ‘grasping

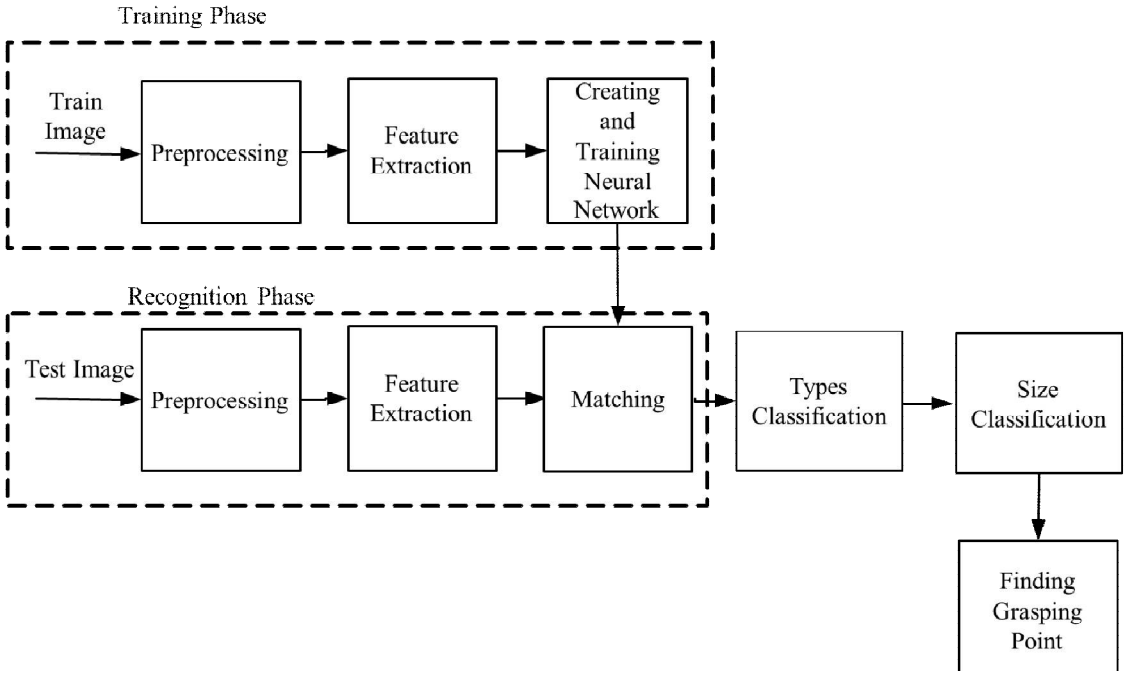

Fig.1. Block Diagram of Proposed System

point' could be estimated from the image using supervised learning algorithms, and that this method generalized to a large number of novel objects. We address the same problem as [10] but use a different real 2Dimages and processing that is capable of higher accuracy at much faster speeds

III. PROPOSED MEHTOD

In the proposed method, the Principal Component Analysis (PCA) is used for extracting features and Backpropagation (BP) Neural Network is used to classify the object types. The main concept of the proposed method is to identify a location at which to grasp the object by using its image. The block diagram of this proposed object recognition system is illustrated in Fig. 1.

The process starts from imported images into the system by camera, detected image is extracted by PCA and trained the object by backpropagation neutral network.Objects are recognized by comparison with the data that are trained in neural network.The proposed method is to recognize the objects and provide the gripper to grasp them efficiently in various appearance changes for grasping tasks of industrial robots.

IV. GRASPING REGION EXTRACTION

4.1. Data Preprocessing

First the training set is created with tool images. Here the training set consists of 75 images of tools. The images are taken under normal light conditions. Varying conditions such as pose and lighting has minimal effect on recognition process. A minimal amount of preprocessing is performed on the data before feeding it to the network. In preprocessing stage, the images are cropped to get the required region. And then the images are changed to gray scale. There are some noise in this step. So we apply the area opening process. This process remove completely region that cannot contain the structuring element and then resized to 500x200 to fit the input layer of our architecture. For test images we simply take it without cropping or resizing.

4.2. Feature Extraction

The purpose of feature extraction is to extract the significant features of images [11]. Firstly the number of features that want to take for each image is defined because the number of features that have to train must equal for all images.

In the features extraction step, PCA is used for extracting 200 features and dimensional reduction. PCA computes the basis of a space which is represented by its training vectors [12]. PCA can be computed by the following steps:

- Get 2D image data.
- Subtract the mean for individual dimensions

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}, \bar{y} = \frac{\sum_{i=1}^{n} y_i}{n} \qquad (1)$$

- Calculate the covariance matrix: Covariance is a measure of how much the two dimensions vary from the mean with respect to each other. Equation (2) is the formula for covariance matrix.

$$\mathrm{cov}(x, y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{(n-1)} \qquad (2)$$

where

x_i= Value of pixel at the i-th position,

$x\square$=the mean value of x of all the pixels

y□ = the mean values of y of all the pixels
n=the number of pixels

- Calculate the eigenvalues and eigenvectors
- Selection of components and transformation (getting new data)

The reduced features input to the Neural Network for recognition purpose. The extracted features are saved in data1 file and which is applied to the input of the Neural Networks. Extracted Feature Points in Database are described in Fig.2.

trainset <200x60 double>

	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	0.3006	0.0868	0.5135	0.1601	0.3938	0.2935	0.0766
3	0.1336	0.0507	0.3869	0.0740	0.1543	0.1481	0.0445
4	0.0439	0.0330	0.1045	0.0383	0.1270	0.0445	0.0313
5	0.0367	0.0287	0.0784	0.0295	0.0599	0.0382	0.0271
6	0.0255	0.0149	0.0637	0.0229	0.0381	0.0299	0.0199
7	0.0183	0.0126	0.0588	0.0183	0.0345	0.0216	0.0089
8	0.0164	0.0083	0.0253	0.0133	0.0236	0.0174	0.0080
9	0.0124	0.0071	0.0204	0.0112	0.0200	0.0146	0.0072
10	0.0105	0.0063	0.0204	0.0083	0.0200	0.0090	0.0051
11	0.0065	0.0055	0.0122	0.0067	0.0145	0.0083	0.0047
12	0.0065	0.0047	0.0106	0.0058	0.0109	0.0070	0.0038
13	0.0059	0.0035	0.0098	0.0050	0.0091	0.0070	0.0034
14	0.0052	0.0035	0.0090	0.0050	0.0073	0.0070	0.0030
15	0.0046	0.0031	0.0073	0.0042	0.0054	0.0056	0.0025
16	0.0039	0.0031	0.0065	0.0042	0.0054	0.0049	0.0025

Fig. 2.Extracted Feature Points in Database

4.3. Training

There are two approaches to training; supervised and unsupervised. In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights, which control the network.

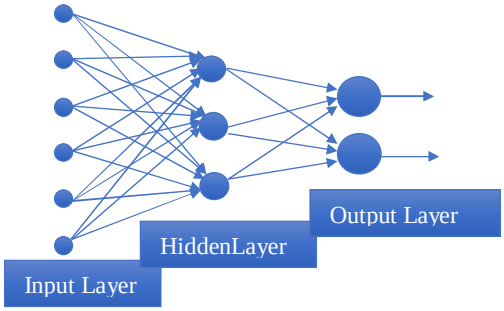

Figure.3.The Architecture of BPNN of the System

There are many laws (algorithms) used to implement the adaptive feedback required to adjust the weights during training. Among then feed forward back propagation is used for this system. The architecture of backpropagation neural network is described in Fig.3.

In this system, BP is designed with three layers; input layer, hidden layer and output layer. The number of input, hidden and output nodes are200, 100, and 7 respectively. Sigmoid functions are used for computation of hidden layer nodes activities and output activities.The input and output data set of the network is denoted as {(p1, t1), (p2, t2)…. (p_Q,t_Q)}.The objective function of the network is defined as

$$V = \frac{1}{2}\sum_{q=1}^{Q}\left[\sum_{i=1}^{S_m}(t_q(i)-o_q^M(i))^2\right] = \frac{1}{2}\sum_{q=1}^{Q} e_q^T e_q \quad (3)$$

where o_q^M is the output of the ith node corresponding to p_q, $t_q(i)$ is the target output of the ith node, and S_m is the number of the output layer node. The network reaches convergence after taking about 879 iteration steps.

Seven types of object are used for this system. The images were captured by placing each object in four different angles (values with 0□, 90□, 270□, and 360□) as shown in Fig.4. Each object has 3 sizes (large, medium and small) so, the total image is 75 for training. Therefore 75 images are collected for creating the knowledge base in the neural network.

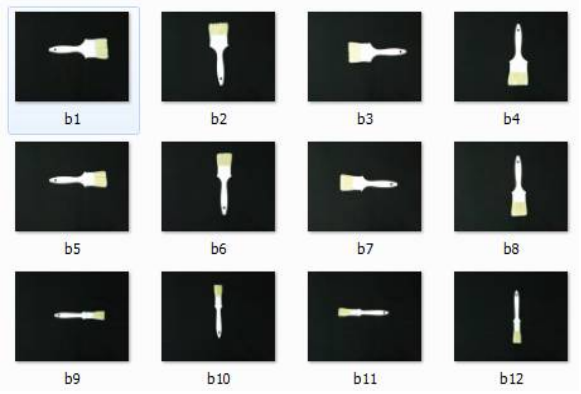

Fig.4 The example training dataset of brush, each with various orientation.

4.4. Object Types Classification

This portion of the system is used to identify the name of the device. The process starts from imported images into the system by camera, detected image is extracted by PCA and trained the object by backpropagation neutral network. The saved neural network is loaded first, then the input feature vector is extracted from the user input image file. After the inputs features of collected device images are extracted, it is put into the neural network.Objects are classified by comparison with the data that are trained in neural network. There are 75 images are used in train data set, all of them can be classified after the Neural Network has trained. The result is displayed with output box. The images in the Dataset is classified into eight distinct categories, with categories like “brush”, “hex key”, “open-end wrench”, “combination wrench”, “nail”, “washer”, and “screwdriver”. Each device image input to recognition system and classified correctly as shown in Fig. 5.



Fig.5.Testing of (top) Wrench and Screwdriver (bottom) Hex-key and Nail

4.5. Size Classification

Each device image input to recognition system and classified correctly as shown in Fig. 6. The images are captured by placing the camera 1feet from the object as the size of objects can be classified according to image pixel value.

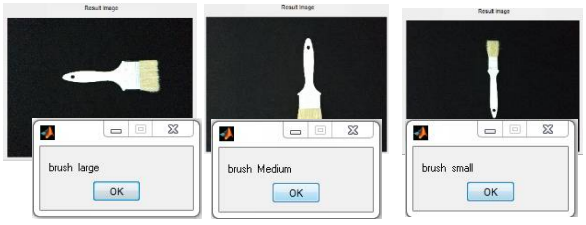

Fig.6.Testing (a) Large (b) Medium (c) Small size

4.6. Grasping Region

For most objects, there is typically a small region that a human (using a two-fingered pinch grasp) would choose to grasp it; with some abuse of terminology, we will informally refer to this region as the “grasping point”. We note that there can be more than one good grasp, and among these, some grasps may be more desirable. For example, a screwdriver may be grasped both by the handle or the shaft, however the handle is preferred due to its size, material and such. In this proposed system, examples of grasping points include the mid-point region for a brush, the centroid region for a screw driver, etc. Grasping regions are given in red rectangle as shown in Fig.7.

Proposed grasping region calculation algorithm can be described as follow.

Grasping Region Calculation Algorithm
Input image of object
Output grasping region
Step 1
Read the image of object.
Step 2
Compute the grasping point of object (x_g,y_g).
if object are Brush and Washer
find the center of object else
find the centroid of object end
Step 3
Compute area of grasping region.
If object is vertical
X=length of object

Y= y_g ±30
else
If object is horizontal
X=x_g±30
Y=length of object
end if
Step 4
Convert grasp region to rectangle region
Fill rectangle area with pixel value 1.
Step 5
Compute boundary of rectangle.
Step 6

Plot the boundary.

But the object is washer, it is necessary to find the label of object after computing the center of object as shown in Fig.8.



Fig.7.The grasping dataset, each with various size (top row) with the corresponding labels (shown in red in the bottom row) of the six object classes was tested.

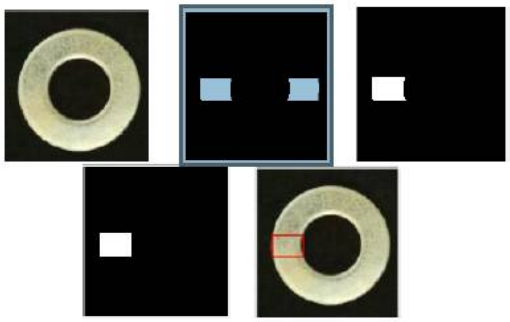

Fig.8. (a) original image, (b) centre region of image (c) result image of selecting label (d) filling label image and (e) result of grasping region shown in red.

V. EXPERIMENTAL RESULTS

For testing: Size 1, 2, 3, angle 45□,135□,225□,and 315□ patterns were also used as shown in Fig.9. The proposed method is implemented by using Matlab. In this paper, the experiment was carried out using a dataset of 25 images for each type and achieves good classification accuracy for 2D images.As the proposedmethod has two main sections, object classification and grasping region extraction,the experimentalresults can be described as two results according to this sections.

No.	Type of Tools	Images		Accuracy	
		Test (known)	Test (unknown)	Recognition	Grasping Region (error nos:)
1	Combination Wrench	12	18(16)	93.33%	2
2	Brush	12	18(15)	90.00%	3
3	L-key	12	18(12)	80.00%	5
4	Screw Driver	12	18(15)	90.00%	2
5	Nail	12	18(12)	80.00%	5
6	Open-end Wrench	12	18(15)	90.00%	3
7	Washer	3	17(16)	95.00%	1
Overall				88.00%	89.50%

Table I:The Accuracy of the System



Fig.9. The testing dataset contains a variety of object, each with various orientation.

This system can correctly predict the object category it has previously seen are classified correctly 100%. For unknown images it is able to recognize objects such as wrench, brush and screwdriver most of the time. The average recognition rate is 88%. Grasping regions are given in red rectangle.The accuracy in predicting the grasping point is reported in Table 1.

CONCLUSIONS

This paper was presented vision based extraction of grasping region of objects using Backpropagation Neural Network that predicts the grasping region of an object directly as a function of its image. The proposed algorithm automatically extracts grasping points that are trained images of a different number of objects. Simulation results of feature extraction and recognition system were illustrated. The system can conclude that average recognition rate of 88.00% that able to recognize objects such as wrench, brush and screwdriver most of the time with a grasp detection accuracy of 89.50%. However, perceiving such as L-key, screw drivers and similar objects was a difficult problem for the proposed algorithm because of reflections and similar features. In the future work, we will go on the grasping research with other feature extraction methods such as profile, shape and so on to achieve more precise and better results. It can also extend to apply in real time application.

ACKNOWLEDGMENTS

The author is thankful to all teachers in Department of Mechatronic Engineering, Mandalay Technological University for their effective guidance, helpful suggestion and supervision for this paper and all of her friends who have directly or indirectly assisted her throughout the year.

REFERENCES

[1] I. López-Juárez, R. Rios-Cabrera, M Peña-Cabrera Fast Object Recognition for Grasping Tasks using Industrial Robots Computación y Sistemas Vol. 16 No. 4, pp. 421 432 ISSN 1405-5546, 2012.

[2] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization Using radial basis function networks," IEEE Trans. on Neural Networks, vol. 4, pp. 570-578, July 1993.

[3] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility," IEEE Trans. Electron Devices, vol. ED-11, pp. 34-39, Jan. 1959.

[4] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in IEEE International Conference on Robotics & Automation (ICRA). Citeseer, 2000, pp. 348–353.

[5] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in IEEE International Conference on Robotics & Automation (ICRA), vol. 2. IEEE, 2003, pp.1824–1829.

[6] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," Robotics & Automation Magazine, IEEE, vol. 11, no. 4, pp.110–122, 2004.

[7] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An svm learning approach to robotic grasping," in IEEE International Conference on Robotics & Automation (ICRA), vol. 4. IEEE, 2004, pp. 3512–3518. C.

[8] T. Miller, S. Knoop, P. K. Allen, and H. I. Christensen, "Automatic grasp planning using shape primitives," in ICRA, 2003.

[9] D. Bowers and R. Lumia, "Manipulation of unmodeled objects using intelligent grasping schemes," IEEE Trans Fuzzy Sys, vol. 11, no. 3, 2003.

[10] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," The International Journal of Robotics Research, vol. 27, no. 2, pp. 157–173, 2008.

[11] Gonzalez, Woods, E. R. C. R. E. S. L.: Degital Image Processing Using MATLAB, Pearson Prentice Hall Pearson Education, Iric, New Jersey, USA, (2004).

[12] Mark Richardson, Principal Component Analysis, May (2009).

★★★