Journal of Computer Applications (JCA)
ISSN: 0974-1925, Volume IV, Issue 2, 2011

An Empirical Comparison Of Three Object Recognition Methods

V. Subbaroyan [a,*] , Dr.S.Karthik [b,1]

Abstract— **In this paper, we evaluate an object recognition system building on three types of method, Gradient based method, Histogram based method and Texture based method. These methods are suitable for objects of uniform color properties such as cups, cutlery, fruits etc. The system has a significant potential both in terms of service robot and programming by demonstration tasks. This paper outlines the three object recognition system with comparison, and shows the results of experimental object recognition using the three methods.**

Index Terms — **Gradient based method, Histogram based method, Texture based method, object recognition.**

I. INTRODUCTION

Object recognition in computer vision is the task of finding a given object in an image or video sequence. Humans recognize a multitude of objects in images with little effort, despite the fact that the image of the objects may vary somewhat in different viewpoints, in many different sizes / scale or even when they are translated or rotated. Objects can even be recognized when they are partially obstructed from view. This task is still a challenge for computer vision systems in general. For any object in an image, there are many 'features' which are interesting points on the object that can be extracted to provide a "feature" description of the object. It is important that the set of features extracted from the training image is robust to changes in image scale, noise, illumination and local geometric distortion, for performing reliable recognition.

In the field of Programming by Demonstration the user teaches the robot new tasks by simply demonstrating them. The robot can initially imitate human behavior and then improve through continuous interaction with the environment. For task learning by instruction, complex systems that involve object grasping and manipulation, visual and haptic feedback may be necessary. If the kinematics of robot arm/hand system is the same as for the human, a one-to-one mapping approach may be considered. This is, however, seldom the case. The problems arising are not only related to the mapping between different kinematics chains for the arm/hand systems but also to the quality of the object pose estimation provided by the vision system. Considering specifically object manipulation tasks, the work on automatic grasp synthesis and planning is of significant relevance.

V. Subbaroyan [a,*], Research Scholar, Sathyabama University, Chennai. (E-mail: V_subbaroyan@yahoo.com)
Dr.S.Karthik [b,1] , Senior Associate, Cognizant Technology Solutions, Chennai.

One of the most important problems in grasping and manipulation is the selection of contact points to grasp an object. Grasping involves fixturing the object relative to a gripper, and forms a necessary condition for object manipulation without changing object or finger contacts. Grasp planning requires a method of evaluating the potential quality of contact points for fixturing the object. Solutions to this problem involve specifying an appropriate grasp quality measure and an algorithm that optimizes this measure to formulate a reaching plan. A successful reaching plan directs the fingers to contact points on the object that are both high quality and achievable by the particular robotic gripper [16].

II. LITERATURE REVIEW

Although generic object recognition and classification have been one of the goals of computer vision scientists since its beginnings, there are still a number of major obstacles for achieving this goal. However, in terms of the identification of known objects in different poses considering novel viewing conditions, significant progress has been made recently, [12]. The two main approaches to the problem are appearance and shape/model based methods. Appearance based approaches represent an object in terms of several object views, commonly raw brightness images. By acquiring a set of object images or reference views, an appearance based object model is constructed.

Since our previous work considered appearance based approaches, [10], [3], we will in this paper, consider the latter approach. The basic idea is that closed 2D curves can be represented by a periodic function, and hence by Fourier descriptors. One such method is for example described in [13]. In this work, closed 2D curves are parameterized, and Fourier descriptors are used to produce a set of normalized coefficients which are invariant under affine transformations. The method is demonstrated on silhouettes of aircraft. Since the shapes of airplanes are more or less planar when seen from large distances, they give rise to affine transformations when rotated in 3D. Hence, the method is ideal for this specific task.

Syntactic matching of curves has also been used, for example in [15]. Here, the curve is represented by an ordered list of shape primitives, and syntactic matching between two curves is performed by dynamic programming. In this particular paper the syntactic matching is only used to align the curves. Proximity matching is then used to measure the similarity between the shapes. The method can deal with partial occlusion, and substantial deformations. Experiments matching the occluding contours of real 3D objects have been carried out, and the method has also been used to classify a large set of 2D silhouettes into classes of similar shapes. Like in [14], this method can be applied to open curves.

III. COMPARISON OF THREE METHODS

A similar method, also using local features, and also specialized in recognizing objects with long thin parts such as bikes and rackets is the method by Mikolajczyk et al. [20]. In this work, the SIFT feature [21] has been generalized to represent the edges in a neighborhood. The algorithm we have chosen to investigate further is a method by Nelson and Selinger [2], [11].

The given input image is converted into gray scale image. The main stage in the 2D correlation based is the creation of a correlation kernel. Sobel horizontal edge emphasizing filter is chosen. The Sobel operator is used in image processing, particularly within edge detection algorithms. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. The Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. Mathematically, the gradient of a two-variable function (here the image intensity function) is at each image point a 2D vector with the components given by the derivatives in the horizontal and vertical directions. This implies that the result of the Sobel operator at an image point which is in a region of constant image intensity is a zero vector and at a point on an edge is a vector which points across the edge, from darker to brighter values. Mathematically, the operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives, one for horizontal changes and one for vertical. If we define A as the source image, and G_x and G_y are two images which at each point contain

3.1 2D Correlation Based Method

The block diagram of simple 2D correlation based object recognition method is shown in Figure (1).

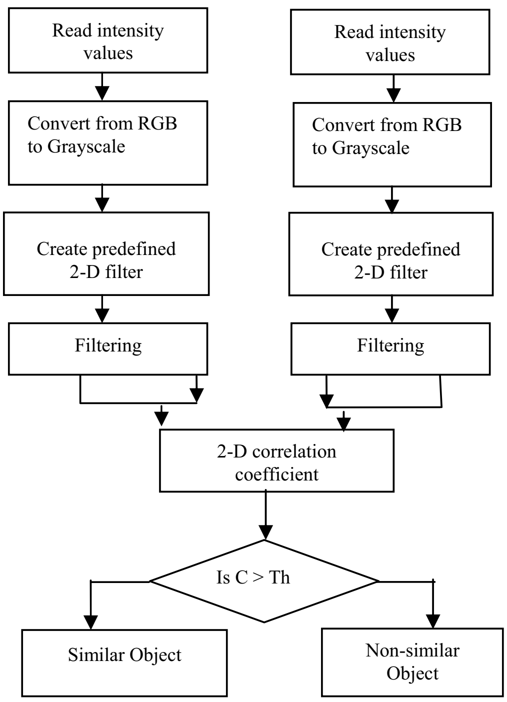


Figure 1. 2D Correlation based Object Recognition method

the horizontal and vertical derivative approximations, the computations are as follows:

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad \text{and}$$

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A \qquad (1)$$

Where * here denotes the 2-dimensional convolution operation.

The x-coordinate is here defined as increasing in the "right"-direction, and the y-coordinate is defined as increasing in the down"-direction. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using the equation (2)

$$G = \sqrt{{G_x}^2 + {G_y}^2} \qquad (2)$$

Using this information, we can also calculate the gradient's direction by using (3)

$$\theta = \arctan\left(\frac{G_x}{G_y}\right) \qquad (3)$$

Where, for example, Θ is 0 for a vertical edge which is darker on the left side. Finally 2D correlation coefficients are computed for the images in the database. If the correlation coefficients between the two images are less than a predefined threshold T, then the two images are identical or non similar object.

3.2 Histogram Based

The equation used in deriving the distance between two color histograms is the quadratic distance metric:

$$d^2(Q,I) = (H_Q - H_I)^t A (H_Q - H_I) \qquad (4)$$

The equation (4) consists of three terms. The derivation of each of these terms will be explained in the following sections. The first term consists of the difference between two color histograms; or more precisely the difference in the number of pixels in each bin. This term is obviously a vector since it consists of one row. The number of columns in this vector is the number of bins in a histogram. The third term is the transpose of that vector. The middle term is the similarity matrix. The final result **d** represents the color distance between two images. The closer the distance is to zero the closer the images are in color similarity. The further the distance from zero the less similar the images are in color similarity.

Global color histogram is used to extract the color features of images. In analyzing the histograms there were a few issues that had to be dealt with. First there was the issue of how much we would quantize the number of bins in a histogram. By default the number of bins represented in an image's color histogram using the ***imhist()*** function in MATLAB is 256. Meaning that in our calculations of similarity matrix and histogram difference, the processing would be computationally expensive. The second issue was in which color space we would present our color map. Should it be **RGB** or **HSV**? This was solved right away when we found that ***QBIC***'s similarity matrix equation was using the **HSV** color space in its calculation.

3.2.1 Similarity Matrix

As can be seen from the color histograms of two images ***Q*** and ***I***, the color patterns observed in the color bar are totally different as shown in Figure (2). This metric is referred to as a *Minkowski-Form Distance Metric*, shown in Figure (3) which only compares the "same bins between colour histograms".

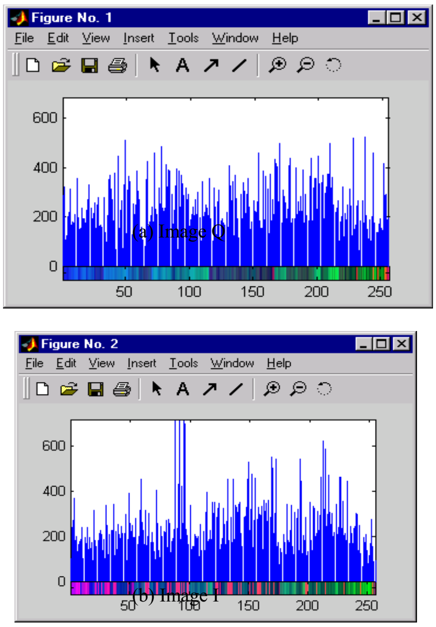
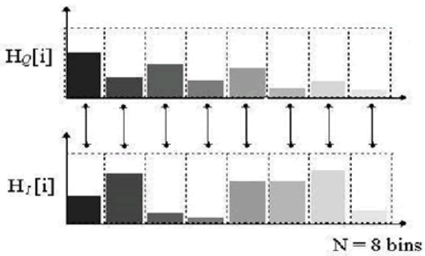


Figure 2. Colour Histograms of two images



This is the main reason for using the quadratic distance metric. More precisely it is the middle term of the equation or similarity matrix ***A*** that helps us overcome the problem of different colour maps. The similarity matrix is obtained through a complex algorithm:

$$a_{q,i} = 1 - \frac{\left[(v_q - v_i)^2 + (s_q \cos(h_q) - s_i \cos(h_i))^2 + (s_q \sin(h_q) - s_i \sin(h_i))^2\right]^{1/2}}{\sqrt{5}} \quad (5)$$

Which basically compares one colour bin of $\boldsymbol{H_Q}$ with all those of $\boldsymbol{H_I}$ to try and find out which colour bin is the most similar, as shown in Figure (4) :

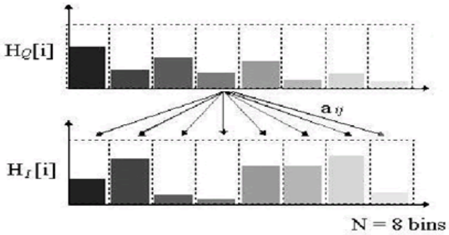


Figure 4. Quadratic Distance Approach

This is continued until we have compared all the color bins of $\boldsymbol{H_Q}$. Finally, we get an ***N* x *N*** matrix, ***N*** representing the number of bins. If the diagonal entirely consists of one's then the color patterns are identical as in Figure (5).

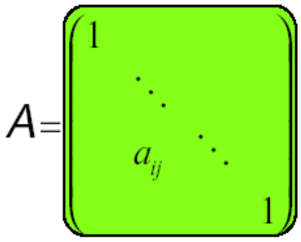


Figure 5. Similarity Matrix A, with a diagonal of one's

3.3 Texture Based Method

We used a method called the pyramid-structured wavelet transform for texture classification. Its name comes from the fact that it recursively decomposes sub signals in the low frequency channels. It is mostly significant for textures with dominant frequency channels.

Using the pyramid-structured wavelet transform, the texture image is decomposed into four sub images, in low-low, low-high, high-low and high-high sub-bands. This is first level decomposition. Using the low-low sub-band for further decomposition, we reached second level decomposition. The reason for this is the basic assumption that the energy of an image is concentrated in the low-low band.

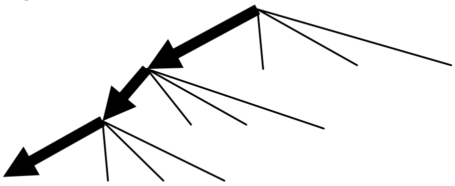


Figure 6. Pyramid-Structured Wavelet Transform.

3.3.1 Energy Level Algorithm:

The Energy Level Algorithm is given below.

1. Decompose the image into *four* sub-images
2. Calculate the energy of all decomposed images at the same scale, using:

$$E = \frac{1}{MN} \sum_{i=1}^{m} \sum_{j=1}^{n} |x(i,j)| \qquad (6)$$

Where ***M*** and ***N*** are the dimensions of the image, and ***X*** is the intensity of the pixel located at row ***i*** and column ***j*** in the image map.

3. Repeat from step 1 for the low-low sub-band image, until index ***ind*** is equal to 5. Increment ***ind***.

Using the above algorithm, the energy levels of the sub-bands were calculated and further decomposition of the low-low sub-band image. This is repeated five times, to reach fifth level decomposition. These energy level values are stored to be used in the Euclidean distance algorithm.

3.3.2 Euclidean Distance:

The calculation of Euclidean distance algorithm is given below.

1. Decompose query image.
2. Get the energies of the first dominant k channels.
3. For image i in the database obtain the k energies.
4. Calculate the Euclidean distance between the two sets of energies:

$$D_i = \sum_{k=1}^{k}\left(x_k - y_{i,k}\right)^2 \tag{7}$$

5. Increment i. Repeat from step 3

Using the above algorithm the query image is searched in the image database. The Euclidean Distance is calculated between the query image and every image in the database. This process is repeated until all the images in the database have been compared with the query image. Upon completion of the Euclidean distance algorithm, we have an array of Euclidean distances, which is then sorted. The five topmost images are then displayed as a result of the texture search.

IV. EXPERIMENTAL RESULTS

We have tested the three proposed system with many objects of different orientation. The test objects are shown in Figure (7).

Figure 7. Test Images

The experimental results are listed in Table (1).

Methods	Matches/No of images	% of recognition
2D Correlation based	28/30	93.3
Histogram Based	28/30	93.3
Texture Based Method	29/30	96.6

Table (1) Recognition Results of three methods

V. CONCLUSION

In this paper, three different methods for object recognition are evaluated. Experimental results demonstrate that the 2D correlation based is suitable for objects having less texture and less colour variation. Histogram based method is well suited for objects having less textures and is independent of the colour variation. Texture based method is well suitable for all type of objects irrespective of colour and texture. All three methods are independent of shapes and orientation.

Although our proposed system tests a small number of objects, our system is designed to continuously adapt in a real-time environment, such as robot navigation. In future, we are going to investigate the possibility of allowing objects having more complicated shapes.

REFERENCES

[1] Bicchi A. and Kumar V., “Robotic grasping and contact: A review,” in Proceedings of the IEEE International Conference on Robotics and Automation, ICRA’00, 2000, pp. 348–353.

[2] Nelson R. and Selinger A., “A cubist approach to object recognition,” in ICCV’98, 1998, pp.614–621.

[3] Bjorkman M. and Kragic D., “Combination of foveal and peripheral vision for object recognition and pose estimation,” Proceedings. IEEE International Conference on Robotics and Automation, ICRA’04, vol. 5, pp. 5135 – 5140, 2004.

[4] Kaiser M. and Dillman R., “Building elementary robot skills from human demonstration,” Proceedings of the IEEE International Conference on Robotics and Automation, v. 3, pp. 2700–2705, 1996.

[5] Chen J. and Zelinsky A., “Programming by demonstration: removing suboptimal actions in a partially known configuration space,” Proceedings of the IEEE Intl. Conf. on Robotics and Automation (ICRA ’01), vol. 4, pp. 4096–4103, 2001.

[6] Ekvall S. and Kragic D., “Interactive grasp learning based on human demonstration,” in Proc. IEEE/RSJ International Conference on Robotics and automation, ICRA’04, 2004.

[7] Petersson L., Jensfelt P., Tell D., Strandberg M., Kragic D., and Christensen H. I., “Systems integration for real-world manipulation tasks,”in IEEE International Conference on Robotics and Automation, ICRA 2002, vol. 3, 2002, pp. 2500 – 2505.

[8] Chaumette F., “Image moments: a general and useful set of features for visual servoing,” IEEE Trans. on Robotics, vol. 20 (4), 2004.

[9] Taylor G. and Kleeman L., “Grasping unknown objects with a humanoid robot,” Australiasian Conference on Robotics and Automation, 2002.

[10] Ekvall F. H. S. and Kragic D., “Object recognition and pose estimation for robotic manipulation using color co-occurrence histograms,” in Proc. IEEE/RSJ International Conference Intelligent Robots and Systems, IROS’ 03, 2003.

[11] Selinger A. and Nelson R, “A perceptual grouping hierarchy for appearance - based 3d object recognition,” CVIU, vol. 76, no. 1, pp. 83– 92, October 1999.

[12]http://www.nada.kth.se/~caputo/cognitive-nipsw03.html.

[13] Arbter K., Snyder W. E., Burkhardt H., and Hirzinger G., “Application of affine-invariant Fourier descriptors to recognition of 3-d objects.” IEEE Trans. Pattern Anal. Mach. Intell., vol.12, no. 7, pp. 640–647, 1990.

[14] Singh R. and Papanikolopoulos N., “Planar shape recognition by shape morphing.” Pattern Recognition, vol. 33, no. 10, pp. 1683–1699, 2000.

[15] Gdalyahu Y. and Weinshall D., “Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes.” IEEE Trans.

Pattern Anal. Mach. Intell., vol. 21, no. 12, pp. 1312–1328, 1999.
[16] Christopoulos Vassilios N. and Schrater Paul," Handling shape and contact location uncertainty in grasping two-dimensional planar objects" Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, Oct 29 - Nov 2, 2007.

BIOGRAPHY

Prof V. Subbaroyan Professor & Head of Department of Electronics & Communication Engineering in K.C.G College of Technology, Chennai. Finished B.Tech (Elec. HC) at IIT Madras in 1971. After working in the Electronic Industry for 30 years, completed M.S. in Digital Electronics at Anna University, Guindy. About to complete Ph.D at Sathyabama University, Chennai.

Karthik Subbaian was born in Erode, Tamilnadu, India, in 1976. He graduated in Electrical and Electronics Engineering from Kongu Engineering college, affiliated to Bharathiar University. In 1998 he obtained his B.E degree and M.E (Applied Electronics) from Coimbatore Institute of Technology. He completed his Doctorate in College of Engineering, Guindy, Anna University. He has so far published more than 20 International Journals and has attended many International and national conferences. Currently he is working in Cognizant Technology Solutions, US Corporation as a Senior Associate. His current research interests include TCP/IP Network Performance Analysis, XML technology, Healthcare Claim processing.