Volume 3, No. 1, Jan-Feb 2012

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Intelligent Object Shape Detector (IOSD) using Canny Edge detection and Shape Estimation Factor (SEF)

Mr. I. Bala Venkateswarlu*
Associate Prof., Department of Information Technology
St. Ann's College of Engineering & Technology
Chirala, India
isunuri@yahoo.co.in

Mr. S. Amarnath Babu
HOD, Department of Information Technology
St. Ann's College of Engineering & Technology
Chirala, India
amardots@yahoo.co.in

Abstract: Many edge detection algorithms are available for detecting edges of the objects in images. Canny edge detection algorithm is one of them, which is used to detect wide range of edges in the images. These algorithms can able to identify the shape of the object. This paper proposes a system, which can detect the shape of the object in images using Canny edge detection and Shape Estimation Factor. This system will work in two modes. One is Learning mode, which extracts features of the images to get intelligence in detection of shapes. Second is Detection Mode, which applies its intelligence for detecting shape of the object using Shape Estimation Factor (SEF).

Keywords: Object, Edge, Edge Detection, Machine Learning, Feature Extraction, Shape Detection, SEF, IOSD

I. INTRODUCTION

In general, any edge detection algorithm will mainly concentrate on detecting the edges of objects in the still images. Simple edge detection is not sufficient for identifying the Shape of the object. It can give region of the object in the image, but not the coordinates of the region.

Now consider the method for giving properties of the object to the system before detection. If we give the properties of object manually as training, it can identify only the shape whose dimensions are same as the training images. Because training images are mapped with current image.

Consider a sample shape of Rectangle; properties of rectangle are length of opposite sites is equal. This property can be given to the system for training; it will help the system for identifying rectangle objects in the image. But when we consider the Circle, its property is all points on circle are located at equal distance known as radius from center.

If we observe the properties of various geometric shapes, they are not unique. They do not contain any common measures for identification. So for the shape detection, it requires some factor that gives correct results even though the object shape is slightly different.

This Intelligent Object Shape Detector will cover all of the above discussed issues in order to make an efficient Shape Detection. Canny edge detection is used for extracting edges of the object. IOSD will compute Shape Estimation Factor after edge detection. This factor is used as primary measure for detecting the Shape of the object.

II. LITERATURE OF CANNY EDGE DETECTION

Canny edge detection is a multi step edge detection procedure by Canny. Steps are as follows:

A. Smooth the image with a Gaussian filter to reduce noise and unwanted details and textures:

$$g(m,n) = G_\sigma(m,n) * f(m,n)$$

Where $$G_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{m^2+n^2}{2\sigma^2}\right]$$

B. Compute gradient of g(m,n) using any of gradient opertors:

$$M(m,n) = \sqrt{g_m^2(m,n) + g_n^2(m,n)}$$

And $$\theta(m,n) = \tan^{-1}\left[g_n(m,n) / g_m(m,n) \right]$$

C. Threshold calculation:

$$M_T(m,n) = \begin{cases} M(m,n) & if \quad M(m,n) > T \\ 0 & otherwise \end{cases}$$

III. MODES OF PROPOSED SYSTEM

Proposed system will work in two modes as follows:

A. Learning Mode:

Learning mode is a training mode in which the system will be trained for detection of shapes. In order to train the system, we need to give some geometric shapes like Square, Rectangle, Triangle, Circle, Ellipse in the form of image. This mode extracts the features of the shapes as coordinates and will be store them for future use. Learning of each shape will be done in following steps:

a. When ever user gives shape image to the system with common back ground color, it will detect the edges of shape with Canny edge detection

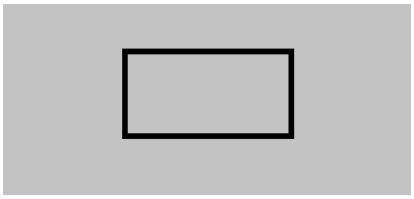

Figure 1. Original Image of a Rectangle for learning

b. After edge detection, it will find the central coordinate of the shape and located on the shape.

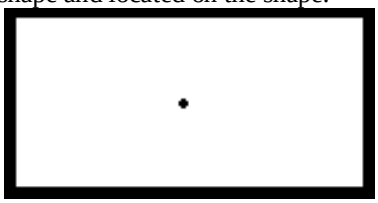

Figure 2. After Edge detection

c. If the center is not correct user need to enter the coordinates of center to locate. Otherwise, system will extract features by finding coordinates on the edges of shape by slicing the shape at some angle by drawing lines across the shape.

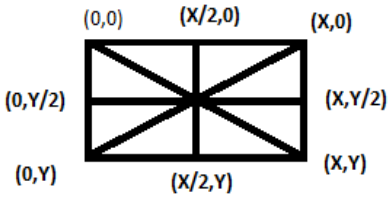

Figure 3. Feature / Coordinates extraction

d. Later it will request the user for the Name of the shape. Finally, it will store the extracted features along with Shape name attached with Shape id, if shape feature are not available in the system. This database is known as Shape database.

Table I. Sample Shape data base

Shape ID	***Shape Name***	***Center Coordinate***	***Coordinates List***
1	Rectangle	(X/2,Y/2)	(0,0), (X/2,0), (X,0), (0,Y/2), (X,Y/2), (0,Y),(X/2,Y), (X,Y)

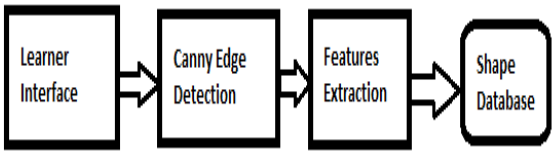

Figure 4. Learning mode block diagram

B. Detection Mode:

After completion of training or learning for the system, then only system will work in detection mode. In detection mode it will identify the shape with following steps:

a. Edge Detection: In this step Canny detection algorithm is applied on the image, to get the edges of the object in the image with all steps of algorithm.

b. Feature Extraction: After getting edges now the system will extract the features of the current object in the image. This step will be carried similarly as discussed in the learning mode.

c. ***Generating Shape Estimation Factor:*** Now the coordinates identified in the above step will be used to map with the coordinated stored in the data base to generate Shape Estimation Factor.

d. ***Shape Detection:*** Finally shape will be detected with the measure of the Shape Estimation Factor.

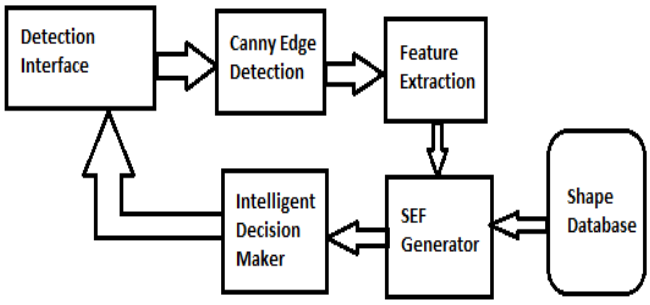

Figure 5. Detection mode block diagram

IV. FINDING SHAPE ESTIMATION FACTOR

Shape Estimation Factor includes two measures as follows:

A. Exact Matching coordinates: It will identify the number of coordinates which are exactly matching with the coordinates stored in the Shpe database. Exact Matching Percentage (EMP) is calculated.

$$EMP = \frac{No.\ of\ exactly\ matching\ coordinates}{Total\ No.\ of\ coordinates}$$

B. Approximate Matching coordinates: It will identify number of coordinates which are approximately matching by considering some threshold distaance. This matching is performed for the coordinated which not extacly matched in the previous step. Difference of shape database coordinate and detected coordinate with in the threshold value then it is counted as Approximate Matching coordinate. Approximate Matching Percentage (AMP) is calculated.

$$AMP = \frac{No.\ of\ approximately\ matching\ coordinates}{Total\ No.\ of\ No\ Matching\ coordinates}$$

C. After finding EMP and AMP values of each and every shape in the shape database. EMP with 100% will be the required shape name. if EMP is not equal to 100%, then the shape id with high EMP and AMP is identified and that will be the resulting shape name.

Case 1:

Consider an object whose coordinates are detected as (0,0), (15,0), (30,0), (0,10), (30,10), (0,20), (15,20), (30,20). If these coordinates are mapped with the coordinates of the shape database. For this sample, X=30, Y =20.

Table II. Mapping the coordinate for Exact match

S. No.	*Detected Coordinates*	*Shape Database Coordinate*	*Result*
1	(0,0)	(0,0)	OK
2	(15,0)	(X/2,0)	OK
3	(30,0)	(X,0)	OK
4	(0,10)	(0,Y/2)	OK
5	(30,10)	(X,Y/2)	OK
6	(0,20)	(0,Y)	OK
7	(15,20)	(X/2,Y)	OK
8	(30,20)	(X,Y)	OK

$$EMP = \frac{8}{8} * 100 = 100\%$$

$$AMP = \frac{0}{8} * 100 = 0\%$$

In this case EMP is 100% and AMP is 0%. Since EMP is more this shape is detected as Rectangle.

Case 2:

Consider an object which contains some destortion on the shape with coordinates as (0,0), (17,0), (30,0), (0,11), (30,21), (0,20), (15,20), (30,20). If these coordinates are mapped with the coordinates of the shape database.

Table III. Mapping the coordinate for Exact Match

S. No.	*Detected Coordinates*	*Shape Database Coordinate*	*Result*
1	(0,0)	(0,0)	OK
2	(17,0)	(X/2,0)	-
3	(30,0)	(X,0)	OK
4	(0,11)	(0,Y/2)	-
5	(30,11)	(X,Y/2)	-
6	(0,20)	(0,Y)	OK
7	(15,20)	(X/2,Y)	OK
8	(30,20)	(X,Y)	OK

Table IV. Mapping the coordinate for Approximate Match with threshold 5

S. No.	*Detected Coordinates*	*Shape Database Coordinate*	*Result*
1	(17,0)	(X/2,0)	OK, because difference is (2,0)
2	(0,11)	(0,Y/2)	OK, because difference is (0,1)
3	(30,21)	(X,Y/2)	Not OK, because difference is (0,11)

$$EMP = \frac{5}{8} * 100 = 62\%$$

$$AMP = \frac{2}{3} * 100 = 66\%$$

In this case EMP is 62% and AMP is 66%. If the EMP is not equal to 100%, then it will identify the shape with high EMP and AMP.

V. REFERENCES

[1] Canny edge detection enhancement by scale multiplication, Vol. 27 Issue 9, Jul. 2005, pp. 1485 - 1490. (Article in IEEE transactions)

[2] Practical Image and Video Processing Using MATLAB, Oge Marques, Wiley-IEEE Press (Book)

[3] Digital Image Processing, Rafael C. Gonzalez, Richard E. Woods, Pearson Education (Book)

[4] Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain, Vol. 1 No. 5, IACSIT Press, 2009, Acc. No.: IJCTE2009G205. (Article in Engineering & Technology Digital Library)

[5] Canny edge detection algorithm based on wavelet transform and RAMF, Dec. 2010, pp. 344 - 346. (Article in IEEE Conference Proceddings)

[6] A hierarchical approach for fast and robust ellipse extraction, Vol. 5, Nov. 2007, pp. 345 - 348. (Article in IEEE Conference Proceddings)

[7] A computational approach to edge detection, Vol. PAMI-8 Issue 6, Nov. 2008, pp. 679 - 698. (Article in IEEE transactions)

[8] Assemble New Object Detector With Few Examples , vol. 20 Issue 12, Dec. 2011, pp. 3341 - 3349. (Article in IEEE transactions)

[9] Line pattern retrieval using relational histograms, vol. 21 Issue 12, Dec. 1999, pp. 1363 - 1370. (Article in IEEE transactions)

[10] http://en.wikipedia.org/wiki/Canny_edge_detector (Article in Website)

[11] http://fourier.eng.hmc.edu/e161/lectures/canny/node1.html (Article in Website)

[12] http://suraj.lums.edu.pk/~cs436a02/CannyImplementation.htm (Article in Website)