Edge Detection Using Morphological Method and Corner Detection Using Chain Code Algorithm

Mr. Anjan Bikash Maity[1], Mr. Sandip Mandal[2] & Mr. Ranjan Podder[3]

1. West Bengal University of Technology. Calcutta Institute of Technology, Information Technology

Uluberia, Howrah, West Bengal, India

2. West Bengal University of Technology. Calcutta Institute of Technology, Information Technology

Uluberia, Howrah, West Bengal, India

3. West Bengal University of Technology. Calcutta Institute of Technology, Information Technology

Uluberia, Howrah, West Bengal, India

Abstract

In this paper we show a very good approach to detect edge and corner of any image. Edges and corners are very important part of an image .In our present days edge and corner detections is very essential for object identification. In this paper we show edge detection using morphological method and corner detection using chain code algorithm. This two method can work on any type of image.

Keywords: Morphology, Chain Code, Erosion, Boundary Extraction, Thinned Binary Image, Hit-and-miss, Pruning.

1. INTRODUCTION

An image may be defined as a t wo-dimensional function, f(x, y), where x and y are the spatial coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or grey level of the image at the point. When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a digital image. The terms morphology commonly denotes a branch of biology that deals with the form and structure of animals and plants. Morphology as a tool for extracting image components that is useful in the representation and description of region shape boundaries and corners.

In this project we use Morphological Edge Detection and Corner Detection Algorithm using chain codes. To find corners of an image we go to through two necessary steps

1st step: Morphological Method.

2nd step: Chain codes algorithm.

2. MORPHOLOGICAL EDGE DETECTION

First we need to read an image. That image may be RGB image. In RGB image each pixel of that image will be in the range between 0 to 6.

Then we convert that image into greyscale image. In greyscale image each pixel of that image will be in the range between 0 to 255

Then we convert that image into binary image. In binary image each pixel of that image will be either 0 or 1.

Then we follow the following four steps.

(i) Boundary extraction,

(ii) Hit-or-miss transform,

(iii) Thinning operation, and
(iv) Pruning operation.

2.1 BOUNDARY EXTRACTION

Our first step for morphological edge detection is boundary extraction. At first we take a structure element and make erosion on t he image by this structure element. Then we erode the binary image . The erosion may be defined as:

A Θ B= Minimum (A(x-i, y-j) × B (i, j) .....(2.1)

This expression shows that we make the erosion of image A by the structure element B. The structure element is entirely contained within A. In fact, erosion reduces the number of pixels from the object boundary. The number of pixels removed depends on the size of structure element.

Let us consider

A = {(1,0),(1,1),(1,2),(0,3),(1,3),(2,3),(3,3),(1,4)}

B = {(0, 0), (1, 0)}

A Θ B = {(0, 3), (1, 3), (2, 3)}

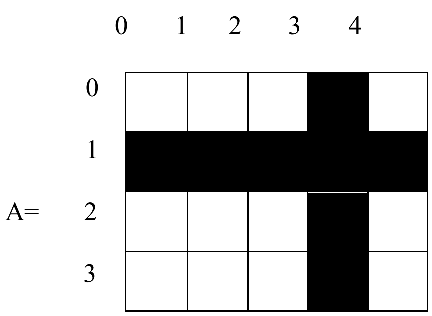

Fig.2.1: Original image

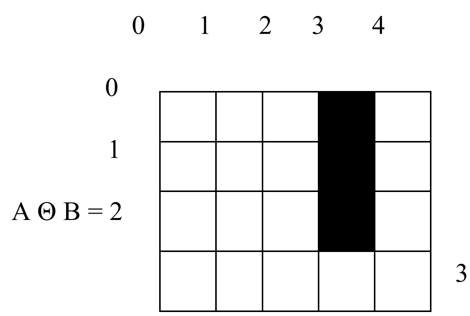

Fig.2.1: Eroted image

After eroding we subtract this eroded image from the binary image. Then we got a b oundary extracted image, which is two or more pixel thick noiseless binary image. The boundary extraction may be defined as:

C = A – (A Θ B) .... (2.2)

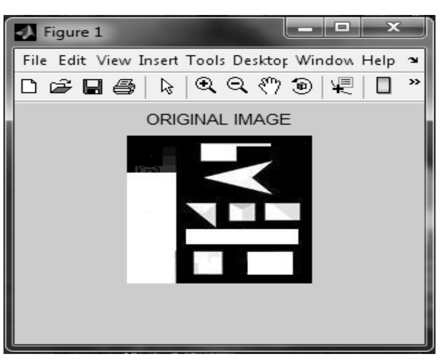

Fig.2.3: Snapshot of original image

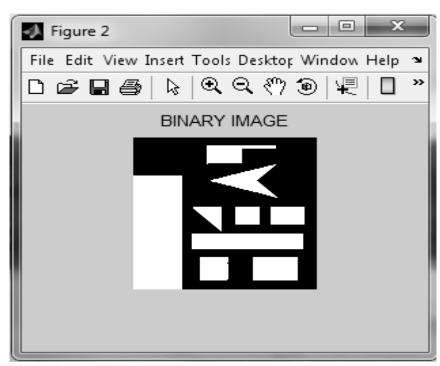

Fig.2.4: Snapshot of binary image

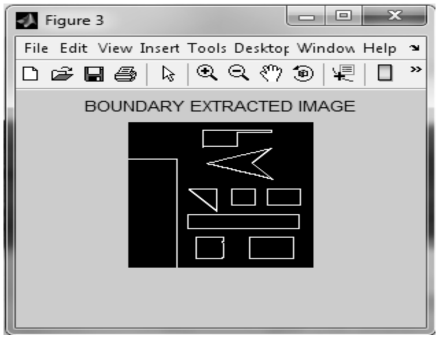

Fig.2.5: Snapshot of boundary extract image

2.2 HIT-OR-MISS TRANSFORM

The hit-or-miss transformation may be defined as morphological operator, which is used for making one pixel thick image from two or more pixel thick image. A small odd size mask typically, 3×3, can be scanned over a binary image. The hit-and-miss transformation operates as a binary matching between the image and the structuring element. If the foreground and background pixels in the structuring element exactly match the foreground and background pixels in the image, then the pixel underneath the origin of the structuring element is set to the foreground color. If it does not, that pixel is set to background color. The Hit-or-Miss transform may also be expressed in terms of erosion as:

$$A \otimes B = (A \ominus B_1) \cap (A^c \ominus B_2) \quad \ldots (2.3)$$

In this process, at first we make the Erosion Operation on the image A with the structure element B1. Then calculate the complement of image A. Then, again we make the Erosion Operation on the image which is the complement of A with the structure element B2. At last we make the Intersection Operation between the two eroted image and we find the result of the Hit-or-Miss Transformation.

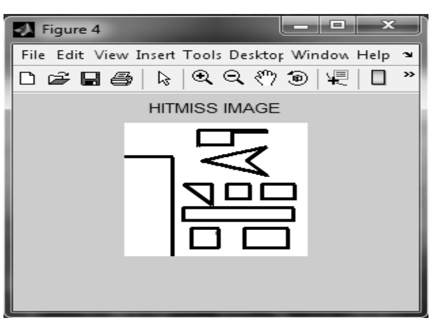

Fig.2.6: Snapshot of hit-or-miss image

2.3 THINNING OPERATION

Thinning is a morphological operation that is used to remove selected foreground pixels from binary images, somewhat like erosion or. It can be used for several applications, but is particularly useful for skeletonization. In this mode it is commonly used to tidy up the output of edge detectors by reducing all lines to single pixel thickness. Thinning is normally only applied to binary images, and produces another binary image as output. The thinning operation is related to the hit-or-miss transform.

Thinning Operation may be defined by the following expression:

$$A \oslash B = A - (A \otimes B) \quad \ldots\ldots. (2.4)$$

In this operation we must be subtract the Hit-or-Miss Transformed image from the Boundary extracted image.

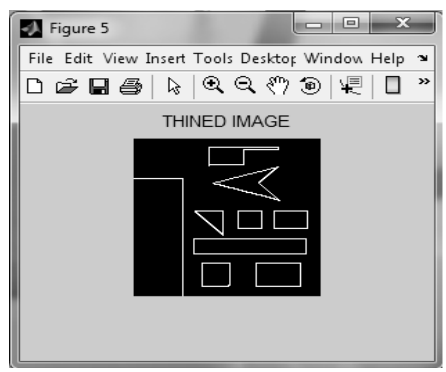

Fig.2.7: Snapshot of thinning image

2.4 PRUNING OPERATION

Thinning operation may be reduced the thickness of an object in an image to a one pixel wide skeleton representation. The problem with this operation is that they leave behind extra tail pixels.

The tail pixels required to remove from this image. The process of removing these tail pixels is called as Pruning. In morphological pruning operation we use the hit-and miss operation with this image by a composite structuring element. After the pruning operation we got the original edge of an object in an image. The morphological operation may be defined as:

$$C = A \otimes B \qquad \ldots\ldots\ (2.5)$$

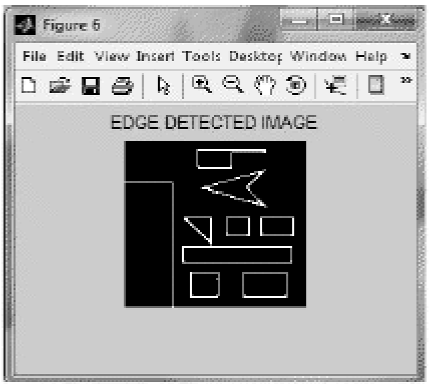

Fig.2.8: Snapshot of edge detected image

3. CORNER DETECTION

Corner detection is an important aspect in image processing and researchers find many practical applications in it. There are many algorithm for detect corner like, Corner detector based on global and local curvature properties. But, we use the Freeman chain codes algorithm for detect true corner of an object in an image.

3.1 CHAIN CODES ALGORITHM

Chain codes are used to represent a boundary by a co nnected sequence of straight-line segments of specified length and direction. Typically, this representation is based on 4- or 8-connectivity of the segments. The direction of each segment is coded by using a numbering scheme such as ones.

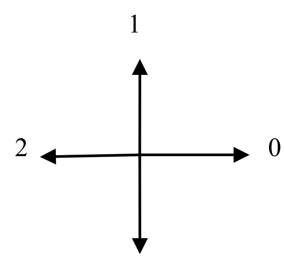

Fig. 3.1: 4-directional chain code

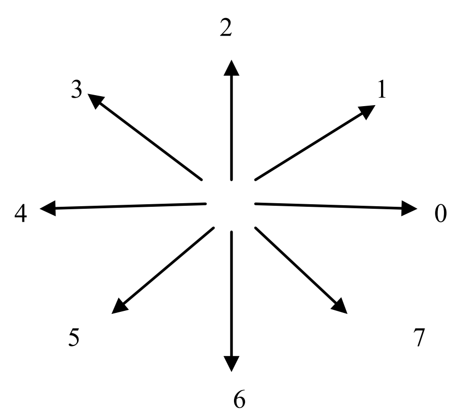

Fig. 3.2: 8-directional chain code

The chain code of a boundary depends on the starting point. However, the code can be normalized with respect to the starting point by treating it as a circular sequence of direction numbers and redefining the starting point so that the resulting sequence of numbers forms an integer minimum magnitude. We can normalize for rotation by using the difference of the chain code instead of the code itself. The difference is obtained by counting the number of direction changes that separate two adjacent element of the code.

Another advantage of chain code is that it is translation invariant but not as well for rotation and scaling. Rotation can be resolved by taking difference chain code while scaling by addressed by changing the size of the sampling grid which the shape overlays on. Each segment of the chain code has direction. If we want to move from one segment to another, the angle magnitude will change.

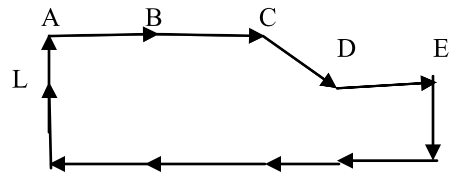

Fig.3.3: Change direction in 90 and 45 degree.

For example, from the above figure if we start from point A, then the chain code for this figure is 00706444422 in 8-directional. Here, From point A to point B the chain code is 0 and point B to C again 0. Here the code will not change so here corner will not be detected. Point C to point D the chain code is 7, here the chain code is change so in point C a corner will be detected. Point D to point E the chain code is 0, here the chain code again change so in point D again a corner will be detected.

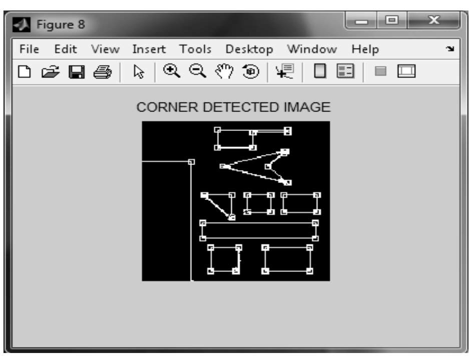

Fig.3.4: Snapshot of corner detected image

4. CONCLUSION

The mapping algorithm has been tested and validated in visualizing and transcribing thinned binary images into chain code by using three thinned binary image objects that are cube, stair, and rectangle. The results show that the visualizing algorithm capable to visualize thinned binary image into rectangular chain code cells. Reciprocally the transcribing algorithm also capable to transcribe the rectangular chain code cells into Vertex Chain Code. Both of these algorithms is called The Mapping Algorithm of Rectangular Vertex Chain Code. In this paper we proposed a novel simple and efficient method for detection of edge or boundary and corner. The method works in two stages: in the first stage the edges of an image can be derived using morphological operators. In the second stage chain-encoding is performed to derive corners of an image. Then number of insignificant corner is reduced in order to optimize the corner detection. This method works on all types of images. This project is implemented using MATLAB 7.5.

REFERENCES

[1] Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins, " Digital Image Processing Using MATLAB", 2nd edition, pp 440-488.

[2] Nor Amizam Jusoh and Jasni Mohamad Zain. "Application of Freeman Chain Codes: An Alternative Recognition Technique for Malaysian Car Plates". V OL.9 No.11, November 2009.

[3] S. H. Subri, H. Haron and R. Sallehuddin, "Neural Network Corner Detection of Vertex Chain Code", AIML Journal, Vol.6, January, 2006, pp 37-39.

[4] Neeta Nain, Viajay Laxmi, Ankur Kumar Jain and Rakesh Agarwal. "Morphological Edge Detection and Corner Detection Algorithm Using Chain-Encoding", IPCV'2006, pp 1-5.

[5] Peter Kovesi, "Phase Congruency Detect Corners and Edges", pk@csse.uwa.edu.au,2003.

[6] P. Kumar, D. Bhatnagar, and P. S. Umapathi Rao, "Pseudo One Pass Thinning Algorithm", vol 12, September 1991, pp 543-555.

[7] F. Moktharian and A. K. Mackworth, "A Theory of Multiscale, Curvature-Based Shape Representation for Planer Curves", vol 45, June 2007, pp 695-708.

Anjan Bikash Maity is a f aculty member in Calcutta Institute Of Technology, Uluberia,Howrah,West Bengal,India.He has done his B.Tech from Bankura Unnayani Institute Of Engineering and M.Tech from Haldia Institute of Technology. He has more than 2 years of teaching experience at B.Tech levels.His interests include Image processing and Cryptography.

Sanip Mandal. He is pursuing B.Tech(Final year) in Information Technology department in Calcutta Institute Of Technology, Uluberia, Howrah, West Bengal, India . Under West Bengal University of Technology.

Ranjan podder. He is pursuing B.Tech(Final year) in Information Technology department in Calcutta Institute Of Technology, Uluberia, Howrah, West Bengal, India . Under West Bengal University of Technology