Object Detection Using Blob Extraction

Saurabh Dhole, Vaibhav Gadewar, Deepika Raut, Radhika Chandwadkar,
Computer Engineering Department,
Vishwakarma Institute of Information Technology, Pune-48

***Abstract* - Separation of object from image i.e. extraction of object from image serves for feature extraction of obtained object or blob and its recognition. The main focus of this paper is to develop an efficient blob extraction algorithm. Due to its capability, it can detect objects and can be used for Object Tracking. The feature extraction of object for Object Recognition, Tracking and Surveillance application can be achieved by using the proposed algorithm for Object separation from background. Experimental results prove the effectiveness of proposed algorithm.**

***Index Terms* – Blob Extraction, Segmentation, Object Recognition, Tracking System**

I. INTRODUCTION

Object Detection refers to detecting various classes of real life object in images or videos. It is the process which detects the interested objects and discards the unwanted background from the image. It has applications in many areas such as Image retrieval and Video surveillance.

Object Extraction is one of the important process in Object Detection.To extract specific features of object from digital image it is necessary to perform segmentation process on original object. As a result of segmentation operation obtained image contains well differentiated object. But the image may contain multiple objects. Also some objects may be unwanted. Objects of interest can be extracted from such an image using Connected Component or the Blob Extraction method.

An object or blob is a set of connected pixels with the same logical state. Blob extraction, also known as region detection or labelling, is a technique that clusters the pixels according to the regions or objects to which they belong [].

Blob detection is generally performed on the binary image resulting from image segmentation. The blob extraction process can give many different regions for a single image. In order for a blob to be confirmed as a candidate, the result of the blob detection process should be filtered and false positives should be discarded for that different filtering techniques can be used.

A two scan connected components labelling algorithm of binary images using raster scan format is recently used algorithm for the blob extraction. The algorithm consists of scanning and numbering any new regions that are encountered, but also integrating old regions when they are found to be joined on a lower numbered label. Therefore, the image is scanned and every pixel is individually labelled with an identifier which signifies the region to which it belongs [2].

II. THE CLASSICAL APPROACH

In connected components of two scan labelling algorithm includes following steps:
In the first pass:

1. Visit each pixel of the image, i.e, Raster Scanning.
2. If the pixel is not the background then
    a. Get the neighbouring pixels of the current pixel.
    b. If there are no neighbours, uniquely label the current pixel and continue, otherwise, find the neighbour with the least label and assign that label to the current pixel.
    c. Store the equivalence between neighbouring labels

On the second pass:

1. Visit each pixel of the image by column, then by row
2. If the pixel is not the background
    1. Relabel the pixel with the lowest equivalent label

Here, the background is a classification, specific to the pixel, used to distinguish from the foreground. If the background variable is not there, then the two-pass algorithm will treat the background as another region. As a result the algorithm becomes more complex. The above mentioned algorithm is complex than the one proposed in this paper.

III. CONNECTED COMPONENT APPROACH

This algorithm is an alternative approach to connected component labeling that is relatively less time consuming. Here we consider only simple images, no occlusion or shadows.

First the image has been pre-processed for noise removal and segmented using Canny Edge Detection algorithm to get the edges of the objects in the image.

The resulting image is a binary image in which foreground pixel is represented as '1' or white and background pixel is represented as '0' or black.

The base of the algorithm is that the pixels are search recursively by scanning the neighbors. It uses 8-connected neighbor approach for scanning. Suppose that foreground pixel is 'P(x, y)', and then 8 neighbor pixels of that pixel are as –

P(x-1, y-1)	P(x, y-1)	P(x+1, y-1)
P(x-1, y)	**P(x ,y)**	P(x+1, y)
P(x-1, y+1)	P(x, y+1)	P(x+1, y+1)

Table 1 8-Connected Pixel

The variable P is indicating the pixel position while x and y are the coordinate of the pixel by referring to the pixel position in graphics coordinate system. In this way due to this recursive scanning, each pixel is scanned exactly once throughout the whole process of blob extraction.

Algorithm:

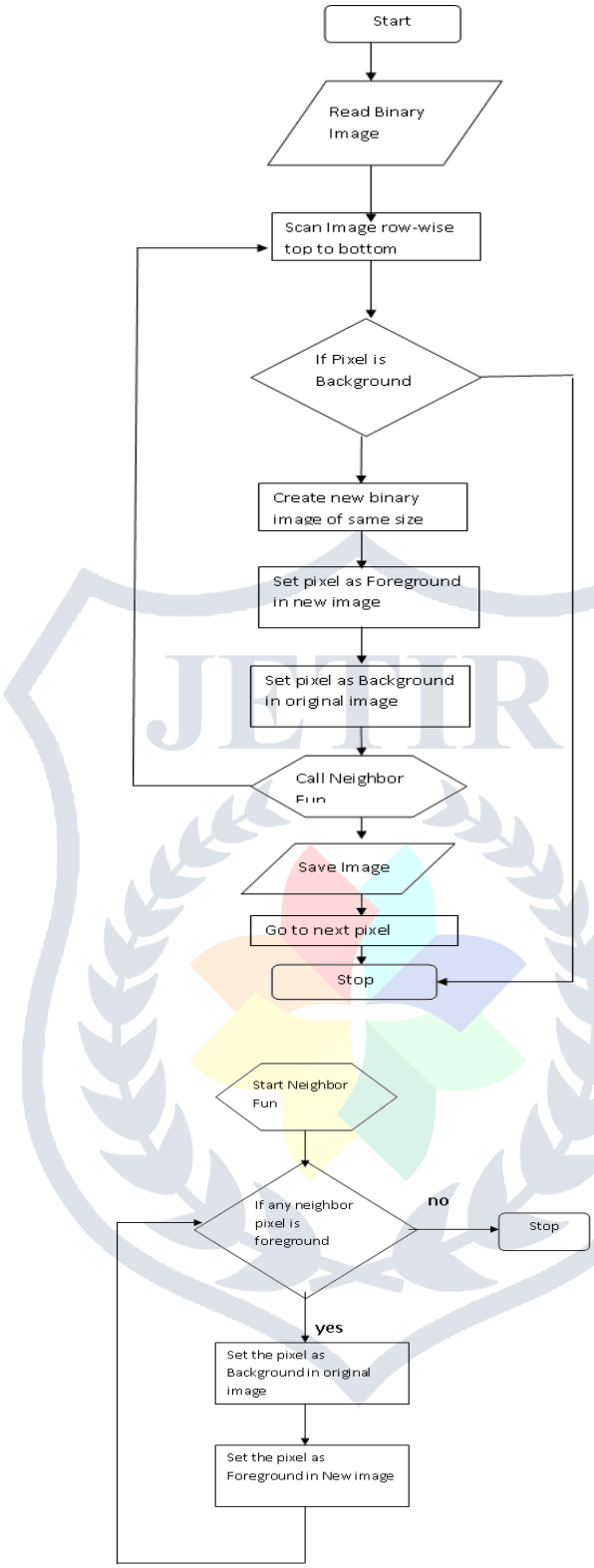
1. Read binary image in which foreground is represented as '1' and background is represented as '0'.
2. Scan the image row-wise top to bottom and working on each pixel as follows:
   a. If pixel is foreground then
      a.1) Call Init_Fun.
      a.2) Set the pixel as background i.e., '0'in the original image.
      a.3) In new image make it as foreground
      a.4) Call Neighbour_Fun.
      a.5) Save the new image.
3. Stop

Init_Fun :

1. Create a new binary image of same size as original image
2. Set the all pixel as background i.e., '0'

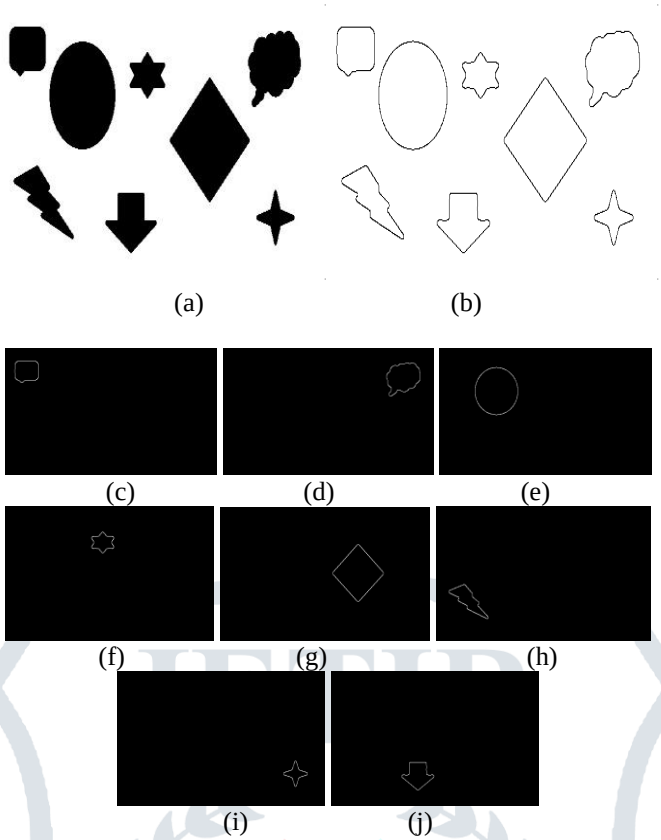
Neighbour_Fun:

1. Check all 8 neighbors of current foreground pixel.
2. If any neighbor pixel has foreground color then make that foreground pixel as background in Original image
3. Make that pixel as foreground in new image
4. Repeat all steps from (1) until no neighbors have foreground color.



IV. EXPERIMENTAL RESULT

The algorithm has been implemented using Java technology and its libraries. The figure below shows the output of the algorithm. The objects are extracted from the image and placed in new image.



The fig (a) shows the original image. This image is processed using canny edge detector to get the image in fig (b) which is a binary image. The resulting binary image is provided as input to the blob extraction algorithm which extracts all the objects from the image as shown above.

Following are the advantages and disadvantages of the proposed algorithm:

Advantage:

1. Less computational time as every pixel is scanned exactly once during the whole process.
2. Simple and Easy to implement.

Disadvantage:

1. Does not handle occlusion.
2. Stack is required to handle recursion.

V. CONCLUSION

Blob extraction is an important step in object detection and recognition. The paper describes how the proposed algorithm works to provide appropriate object extraction. This algorithm for blob extraction is different from the two-scan connected component labeling algorithm. The proposed algorithm relies on efficiency and simplicity.

As regards efficiency, it has been shown by experimental results that, unlike the two-scan connected component labeling algorithm, the proposed algorithm does not label each and every pixel and their classification which improves algorithm efficiency. The algorithm gives the approximate object region for further processing. Also it has much other applications in surveillance, retrieval and multimedia.

REFERENCES

[1] M. King, B. Zhu, and S. Tang, "Optimal path planning," *Mobile Robots*, vol. 8, no. 2, pp. 520-531, March 2001.
[2] H. Simpson, *Dumb Robots*, 3rd ed., Springfield: UOS Press, 2004, pp.6-9.
[3] M. King and B. Zhu, "Gaming strategies," in Path Planning to the West, vol. II, S. Tang and M. King, Eds. Xian: Jiaoda Press, 1998, pp. 158-176.
[4] B. Simpson, et al, "Title of paper goes here if known," unpublished.
[5] J.-G. Lu, "Title of paper with only the first word capitalized," *J. Name Stand. Abbrev.*, in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Translated J. Magn. Japan*, vol. 2, pp. 740-741, August 1987 [*Digest 9th Annual Conf. Magnetics Japan*, p. 301, 1982].
[7] M. Young, *The Technical Writer's Handbook*, Mill Valley, CA: University Science, 1989.