Approximating Images Using Minimum Bounding Rectangles

Azzam Sleit, Imad Salah, Rahmeh Jabay
Department of Computer Science, King Abdullah II School for Information Technology
University of Jordan, Amman, Jordan
azzam.sleit@ju.edu.jo, isalah@ju.edu.jo, rahmeh_jabay@yahoo.com

Abstract

In surveillance systems, video cameras record specific scenes for long times. However, by the end of the recording period, such tapes may hold many useless scenes which need to be eliminated. In order to reduce the time in reviewing these worthless scenes while seeking for a specific object, an approximation technique must be considered. In this article, we propose a new technique for finding the minimum bounding rectangle of objects which appear in a specific bitmap image. The minimum bounding rectangle of an image object is the Rectangle containing the object such that the sides of the rectangle touch the object boundaries. The dimensions and locations of the minimum bounding rectangle of an object can be utilized as features to identify the corresponding object.

Keywords: Surveillance System, minimum bounding rectangle, object tracking

1. Introduction

Geometric shapes such as polygons and circles can be used to approximate the dimensions of a pictorial object. Along with other features, the dimensions and location of the bounding geometric shape can be utilized to identify the object. The minimum bounding rectangle (i.e. MBR) of an object is one of the most commonly acceptable bounding container shapes in the image and spatial databases research community [1, 2, 3]. This is due to the to the simplicity of rectangles when utilized in the recognition process in addition to the fact that most spatial data structures such as R-tree and SB^+-tree are based on rectangles [4, 5, 6, 7]. The classical method to obtain the bounding rectangle of an object is to raster scan the object in the image and subtracts all the pixels of the object from the image in order to find the object's pixels with the minimum X coordinate, minimum Y coordinate, maximum X coordinate and maximum Y coordinate. Also, recent work for finding bounding boxes in textual document images is based on skew estimation [3]. Most existing algorithms require multiple passes for the image.

This paper introduces a faster algorithm for finding the MBRs of objects which appear in an image. Section 2 introduces a new algorithm for finding the minimum bounding rectangles of objects which appear in a particular image. Section 3 gives some examples demonstrating the merits of the proposed algorithm. Section 4 discusses some concluding remarks.

2. Minimum Bounding Rectangle by Region Growing Segmentation

The goal of segmentation is to cluster pixels into salient image regions corresponding to individual objects. The proposed method is based segment the image taking into consideration the spatial similarity amongst pixels. The algorithm accepts a binary image as input and a set of points to start the region growing process. It begins at each point in the input set and scans the neighboring 8 pixels in order to grow the regions. This process repeats itself for new pixels to determine the membership pixels that constitute the region. When a pixel's value implies that it is not a member of the object, the coordinates are saved in a no match list. For each constructed region, we build a corresponding list. .The lists of objects are parsed to find the minimum and maximum X and Y coordinates of the objects. The following algorithm formally explains the process.

Algorithm: Find Minimum Bounding Rectangle

Input: Binary Image
Output: Image With rectangles corresponds to objects in the image

1. Select a seed point in the image with value of 1.

2. Test the 8-neighbors of the point.
If all neighbors of this point have the same value as the seed point **then**

Label seed point
Update seed point
Repeat step 2

Else
Initialize anew Xlist
Initialize anew Ylist

Save coordinates of pixels with values not equal to seed in Xlist, Ylist

3. Repeat step 2 until there is no seed point with value=1

4. For each pair of identified list that correspond to object determine the minimum and maximum for both Xlist and Ylist

5. For each pair of modified list draw the rectangle.

The previous algorithm obviously has run time requirement which is proportional to the size of the image.

3. Experimental Results

In order to utilize the previous algorithm, three steps must be followed. Firstly, the input image must be converted to a binary image using some threshold value. Secondly, the image is segmented by region growing method with each identified object is labeled by red. Finally, rectangles are drawn in the original image containing the objects identified. While a pixel is labeled, its distance from the seed must be given special attention. The value of distance N must be chosen in terms of the expected object and image sizes. For images of relatively large objects, N should be chosen with expected object's size in mind. If N is small with respect to object size, this will lead to additional overhead of the recursive calls to region growing and finally results with inaccurate coordinates of pixels that must touch the bounding rectangle sides. Figure 1 and 2 display sample runs of the algorithm.

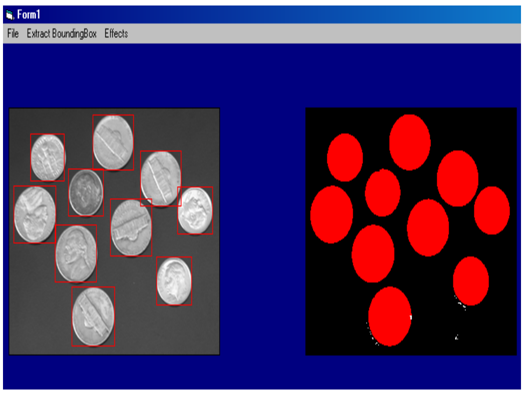

Figure 1: Output with N=1

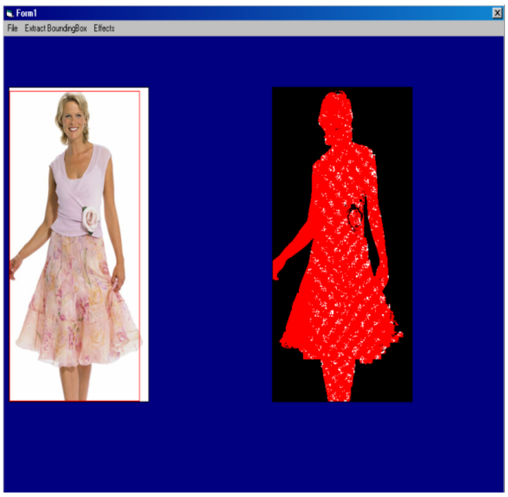

Figure 2: Output with N=11

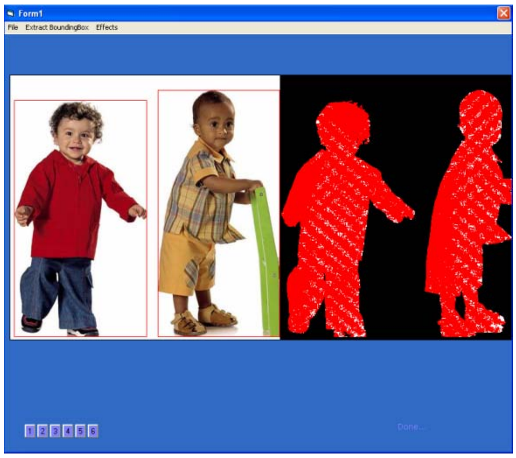

Figure 3: Output with N=12

If an image is perfectly segmented into well-recognized regions, the resulting bounding rectangles will correctly fit the objects. When objects of interest share the same boundary, they will be fit in one bounding rectangle as shown in Figure 3

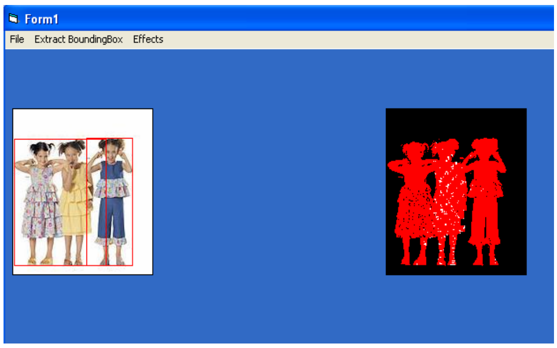

Figure 4: Output with N=7

In figure 4, the two girls on the left share the same bounding rectangle while the girl on the right is fit in one separate rectangle.

4. Conclusion

This article presents an algorithm for finding the minimum binding rectangles of objects which appear in an image. The algorithm requires converting the image into a binary image using some threshold value. Using initial seed values, the algorithm grows regions incrementally. The algorithm has the limitation that when two or more objects overlap in their X or Y coordinates, they will be bound by one rectangle.

5. References

[1] A. Guttman, "R-trees: a Dynamic Index Structure for Spatial Searching", In Proceedings of the ACM SIGACT-SIGMOD Conference on Principles of Database Systems, 1984, pp. 569-592.

[2] T. Sellis, N. Roussopoulos, and C. Faloutsos, "The R+-tree: a Dynamic Index for Multi-Dimensional Objects", In Proceedings of the Thirteenth International Conference on Very Large Databases (VLDB), 1987, pp. 507-518.

[3] H. Samet, "Hierarchical Representations of Collections of Small Rectangles", ACM Computing Surveys, 20, 4, 1988, 361-385.

[4] A. Ibrahim and F. Fotouhi, "Indexing and Retrieving Point and Region Objects", SPIE, 2670, 1996, pp.321-336.

[5] A. Ibrahim, F. Fotouhi, and S. Hasan, "The SB^+-tree: An Efficient Index Structure for Join Spatial Relations", International Journal of Geographical Information Science, 11, 2, 1997, pp. 163-182.

[6] A. Ibrahim, F. Fotouhi, and A. AL-Badarneh, "Efficient Processing of Spatial Selection and Join Operations using SB^+-tree", International Database Engineering & Applications Symposium, 1997, pp. 279-288.

[7] B. Ooi and K. Tan, "B-trees: Bearing Fruits of all Kinds", In Proceedings of 13th Australasian Database Conference, IEEE CS Press, 2002.