SEGMENTATION OF IMAGES INTO REGIONS USING EDGE INFORMATION*

Gerard G. Medioni

Intelligent Systems Group
University of Southern California
Los Angeles, California 90089-0272

ABSTRACT

A method for segmenting aerial images using edge information to create regions of similar or smoothly varying intensity is discussed. Region segmentation using edges directly as input cannot be successful because boundaries are seldom perfectly closed. In the present method, we preprocess the edge image to close gaps and create a binary image from which we extract the connected regions. We compare the results obtained with this method and a traditional region-splitting method for 2 different views of an aerial scene.

I INTRODUCTION

There have traditionally been two main approaches to the segmentation of images, edge based and region based.

- Edge based methods proceed by locating discontinuity points in the intensity image and connecting them to obtain primitives. They have the advantage of preserving most of the information present in the intensity picture but produce very low level primitives even after further processing (segments) [1]. They are very appropriate to describe elongated objects such as roads and rivers.

- Region based methods proceed either by merging regions that have similar intensity and a weak boundary separating them [2,3], or by recursively splitting regions using a threshold defined by histograms [4]. This last technique is very effective on multispectral images. These methods produce higher level primitives (regions with a set of attributes), but most of the time these regions do not correspond to physical entities unless their intensity differs everywhere from the background. If the contrast is too weak, the object will "leak" and will be merged with its background.

We present here a method trying to combine the good points of the 2 methods described above. It substantially differs from the expansion-contraction approach used by Perkins [5] to bridge gaps in the edge image, and does not require an object's interior to contrast with its surround as in Milgram's "superlice" technique [6].

II DESCRIPTION OF THE METHOD

From the grey-level image, we extract the edge points and organize them into linear segments. Using this edge information, we create a new image in which pixels belonging to an edge segment get an intensity depending upon the contrast of the edge at this point and the total length of the segment. We now bridge the gaps in the edges by replacing the intensity at each point by the sum of the intensities in a small square window centered at this point. By thresholding this new image, we obtain a binary image from which we extract the connected regions of intensity 0. These regions are smaller than the expected ones because of the smoothing process, so we expand each one individually to obtain the final result.

A. Processing the grey-level image

We first extract the edges from the image, thin them and link them using the technique developed by Nevatia and Babu [1]. The final primitives we obtain are SEGMENTS, linear pieces approximating a set of edge points. The attributes of a segment are its 2 end points, its length L and its strength S, which is the sum of the contrast of each point. Since we want to eliminate the gaps in the edges following the boundary of an object and reduce the influence of small random or textured edges, we create an image f(i,j) as follows:

```
if (i,j) belongs to a segment SEG

then if LENGTH[SEG] < MINLENGTH

     then f(i,j) = LENGTH/STRENGTH

     else f(i,j) = STRENGTH

else f(i,j) = 0.
```

*This research was supported by the Defense Advanced Research Projects Agency and was monitored by the Air Force Wright Aeronautical Laboratories under Contract F-33615-80-C-1980, ARPA Order No.3119.

This non-linear process permits us to recognize long (>MINLENGTH) segments and to give a high weight to their points.

B. Summing the Image

Given the image f(i,j), we use a simple texture/no-texture discrimination process by creating a new image g(i,j) as follows:

$$g(i,j) = \sum_{l=i-n}^{i+n} \sum_{m=j-n}^{j+n} f(l,m)$$

That is, g(i,j) is the sum of f(i,j) in a square window of size 2n+1 centered in (i,j). We then threshold this image to get a binary version of it:

$$h(i,j) = 0 \text{ if } g(i,j) < THR$$
$$= 1 \text{ otherwise.}$$

C. Extracting Regions

From the image h(i,j), we extract all connected regions of intensity 0. Each region represents a shrunk version of a region in which no edges, or very small and weak edges, are present and the gap between the edge and the border created by the edge is n pixels, n being defined above. In order to reconstruct the physical region, we use a growing procedure on each region as described in [7]: for each pixel, we consider a square window of size 2n+1 centered at that point, and set the pixel to 1 if any pixel in the window is 1. One problem with this technique is that some corners get rounded.

D. Interpretation

Each region now corresponds to a set of edges forming a nearly closed boundary enclosing this region. These regions can be further filtered by looking at their attributes, such as area, ratio of perimeter^2/area and others. They can be the input of a region matching program or can be looked at individually to see if there is an adjoining projected shadow.

III RESULTS

We tried the above procedure on 2 views of the same scene showing part of the Fort Belvoir Military Reservation in Virginia. The original images have a resolution of 600 by 600 and are shown on figures 1a and 2a. Figures 1b and 2b show the segments extracted from the intensity array. Note that the boundary of the large building in the lower left of fig. 1b is not closed, or even nearly closed. Figures 1c and 2c show the image after summation. The following parameters were used: n = 4 (that is, windows are 9 by 9). MINLENGTH = 12 (minimum length of a segment for non-linear processing). From these images we extract connected regions of intensity < 150. We now expand each region individually and filter out all regions with a value of perimeter^2/area > 35 to obtain the final result, as shown on figures 1d and 2d. As we can see, no buildings are missed and their shape is rather well conserved. Figures 1e and 2e show the set of regions obtained by a conventional region splitting [4]. In both images the large building in the lower left is totally lost and some other buildings are merged into a single region.

IV CONCLUSION

The method described above provides better segmentation than region growing or region splitting techniques without semantic information. Computing histograms, especially on monochromatic images, does not always provide a good threshold, even though edges define a clear boundary. We are currently investigating the exact effect of the parameters and a segmentation method coordinating edge information and region splitting.

V REFERENCES

[1] R. Nevatia and K. Ramesh Babu, "Linear Feature Extraction and Description," Computer Graphics and Image Processing, Vol. 13, June 1980, pp. 257-269.

[2] C. R. Brice and C. L. Fennema, "Scene Analysis Using Regions," Artificial Intelligence 1:3, 1970, pp. 205-226.

[3] Y. Yakamovsky and J. A. Feldman, "A Semantics-Based Decision Theory Region Analyser," Proc. IJCAI-73 Stanford, 1973, pp. 580-588.

[4] R. Ohlander, K. Price and R. Reddy, "Picture Segmentation Using a Recursive Splitting Method," Computer Graphics and Image Processing, Vol. 8, 1978, pp. 313-333.

[5] W. A. Perkins, "Region Segmentation of Images by Expansion and Contraction of Edge Points," Proc. IJCAI-79 Tokyo, Aug. 79, pp. 699-701.

[6] D. Milgram, "Region Extraction Using Convergent Evidence," Computer Graphics and Image Processing, Vol. 11, No. 1, 1979, pp. 1-12.

[7] A. Rosenfeld and A. C. Kak, "Digital Picture Processing," Academic Press, 1976.

Figure 1a. Original image resolution 600x600

Figure 1b. Segments extracted from the original image

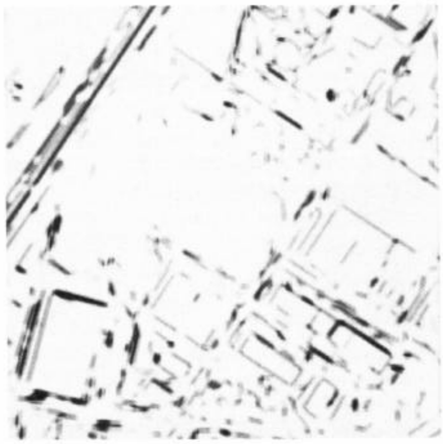

Figure 1c. Summed image

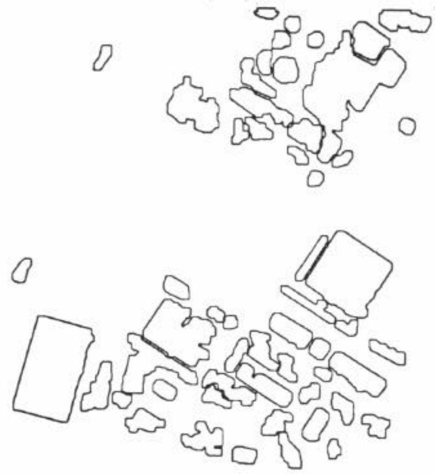

Figure 1d. Regions from the summed image

Figure 1e. Regions obtained by splitting method

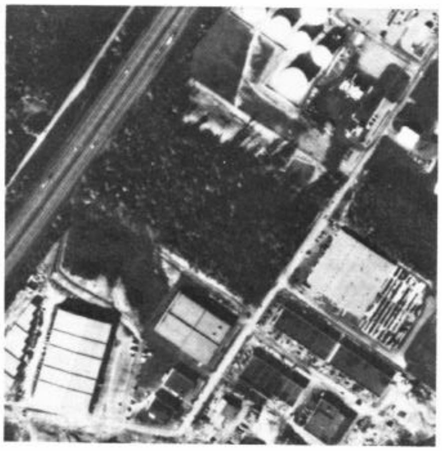

Figure 2a. Original image resolution 600x600

Figure 2b. Segments extracted from the original image

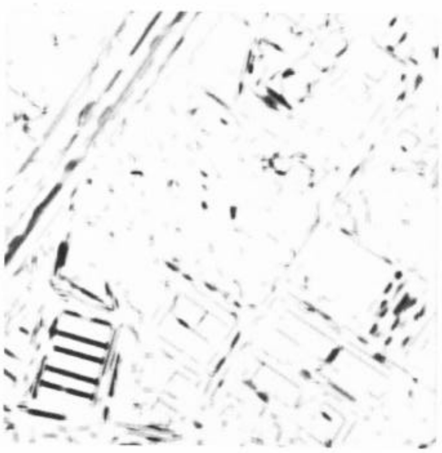

Figure 2c. Summed image

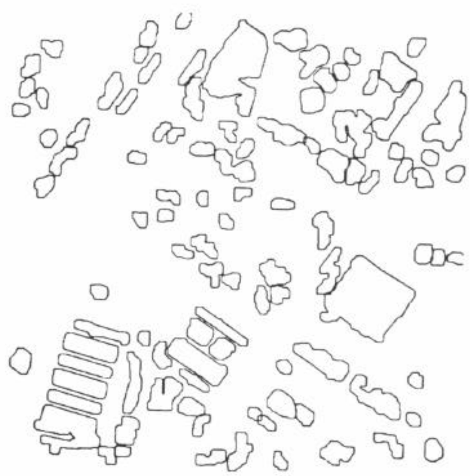

Figure 2d. Regions from the summed image

Figure 2e. Regions obtained by splitting method