

LINEAR FEATURE EXTRACTION AND DESCRIPTION

Ramakant Nevatia
Computer Science Department and
Image Processing Institute
University of Southern California
Los Angeles, California 90007

K. Ramesh Babu
Computer Science Department and
Image Processing Institute
University of Southern California
Los Angeles, California 90007

Abstract

A technique of edge detection and linking for linear feature extraction and its applications to detection of roads and runway like structures is described. Experimental results are included.

1. INTRODUCTION

The importance of effective early processing of visual input for a complete image understanding system is generally accepted. The early processing may consist of a description of image discontinuities, usually in the form of edges and lines, or segmentation into uniform regions. In this paper, we describe a technique for extracting linear features in an image by a process of edge detection and line linking and also of deriving higher level descriptions from the extracted lines. These techniques are aimed to be general. Applications to road detection and airport recognition tasks are described.

Edge detection and line finding techniques, of course, have been studied since the early days of this field. The literature is too numerous to list here; partial surveys may be found in [1-3]. In spite of the large amount of previous research in this area, no algorithms suitable for complex imagery are apparent. Here, we describe an edge detection and line finding technique with superior performance on a wide variety of images. Many of the steps in our process have been suggested by other researchers previously. However, due to limitations of space, we are unable to elaborate on these in this paper. More details may be found in [4].

2. EDGE DETECTION AND LINE FINDING

Convolution with Edge Masks

Edge detection starts by convolving a given image with masks corresponding to ideal step edges in a selected number of directions. The

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense under contract NO. F-33615-76-C-1203. Dr. Keith Price has been extremely helpful in many discussions and providing image manipulation software.

magnitude of the convolved output and the direction of the mask giving the highest output at each pixel are recorded as edge data. We have found 5 x 5 masks in six directions to be suitable for most images of interest.

Thinning and Thresholding

The presence of an edge at a pixel is decided by comparing the edge data with some of the 8 neighboring pixels. An edge element is said to be present at a pixel if:

1. the output edge magnitude at the pixel is larger than the edge magnitudes of its two neighbours in a direction normal to the direction of this edge. (The normal to a 30 degree edge is approximated by the diagonals on a 3 x 3 grid);
2. the edge directions of the two neighboring pixels are within one unit (30 degrees) of that of the central pixel; and
3. the edge magnitude of the central pixel exceeds a fixed threshold.

Further, if the conditions 1 and 2 above are satisfied, the two neighboring pixels are disqualified from being candidates for edges. This algorithm produces results independent of the order in which the pixels are examined.

Linking

The first step in boundary tracing is to determine the continuing neighbors on the boundary for each edge point. Typically each point has two neighbors, known as a predecessor and a successor, except for the end-points, isolated points and where two boundary segments join or intersect. This connectivity information is stored in two files, called p and s files, of the same size as the input image. Each element in these files contains an integer corresponding to the preceding (or succeeding) edge among the 8 neighbors. Forks are indicated by an extra bit as explained later.

Our criteria for connecting two edge points is that they be neighbors, in the 8-neighbor sense, and that they have edge directions differing by not more than a certain value, set at 30 degrees for masks described previously. Due to the nature of thinning, only three locations are potential candidates for predecessor or successor elements as shown in Figs. 1(a) and (b) for edges of 0 and 30 degree directions respectively. The determination of successor (predecessor) pixels is elaborate due to the several cases that are possible at each pixel:

1. Only one element is an acceptable successor. In this case the successor (predecessor) is recorded in the s(p) file as an integer corresponding to its location.
2. Two candidates are acceptable successors. If they are not 4-neighbors, a fork is present as shown in Fig. 2(a). If they are 4-neighbors, a fork exists only if their directions differ by more than 2 units (60 degrees), as in Fig. 2(b). Otherwise no fork exists and the nearer of the two (using Euclidean distance), forms the successor (predecessor), as shown in Fig. 2(c). These rules are for smooth continuation of lines and were derived by complete enumeration of such configurations. In case of a fork, the stronger of the two candidates in edge magnitude forms the main stream. The fact that a fork exists is noted in the s(p) file by marking one bit. This information is sufficient to trace both streams of a fork by examining the p and s files simultaneously.
3. Three candidates are acceptable successors. Fig. 3 shows all possible such configurations for a vertical edge (no three successor configurations occur for 30 degree edges). In these cases, a fork exists. The main stream is formed by the nearer of the two edges having the same direction, and the other candidate with different direction forms the other branch.

The predecessor-successor representation of the linked edge elements is in contrast to explicit lists of elements forming a connected segment. For larger images, not entirely resident in core, it is more convenient to form predecessor and successor matrices first, as the processing requires only a sequential scan of the image file. Further, this representation is "iconic" and certain proximity computations can be made more easily.

Tracing Boundary Segments

Boundary segments containing ordered lists of edge elements are computed from the predecessor-successor (p-s) files. An edge point with no predecessors determines the starting point. A boundary segment is traced by

following the successor elements until an element with no successors is reached. At a point with multiple successors, the "main" fork is chosen. The secondary branches of forks are picked up on a second pass that starts tracing at such points. A third pass is required to trace circular or closed segments. This pass begins arbitrarily at an edge point that has not been thus far visited, i.e. included in a segment. (A temporary binary file is used to store this information).

Linear Approximation

Each boundary segment is approximated by a series of piecewise linear segments. The algorithm used is a variation of the well known iterative end-point fit algorithm (see [5] pp. 338-339). Results of processing an airport image are shown in Figure 4.

3. OBJECT DESCRIPTION

A major goal of this research is to compute descriptions of objects useful for their recognition. Certain classes of objects, such as roads and airport runways, are characterized as being bounded by elongated parallel lines. These lines have opposite contrasts and we call them "anti-parallel" lines, abbreviated as "apars". These apars are conveniently described by a medial line and width of the pair. These descriptions were motivated as being two-dimensional special cases of Binford's "Generalized Cone" representation [6]. Finding apars is facilitated by sorting line segments by their orientations.

Proper choice of apars that correspond to objects can be difficult and is like resolving figure-ground relationships. However for many applications, such as for roads and runway detection, a choice of the closest pairs may suffice and be aided by knowledge of the desired objects being brighter or darker than the background.

Fig. 4(c) shows the axes of the apars computed from the segments in fig. 4(b). Only the closest pairs were retained. Work is in progress to recognize specific airports from the spatial relationships of the partial axes of the runways and taxiways as shown here.

REFERENCES

- [1] A. Rosenfeld and A. Kak, Digital Picture Processing, Academic Press, 1976.
- [2] W.K. Pratt, Digital Image Processing, John Wiley & Sons, 1978.
- [3] L.S. Davis, "A Survey of Edge Detection

Techniques", Computer Graphics and Image Processing, Vol. 4, NO. 3, pp. 248-270, Sept. 1975.

[4] R.Nevatia and K.R. Babu, "Linear Feature Extraction," in USCIP Report 840, Oct. 1978.

[5] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, Wiley, 1973.

[6] T.O. Binford, "Visual Perception by a Computer," IEEE Conf. on Systems and Controls, Miami, Dec. 1971.



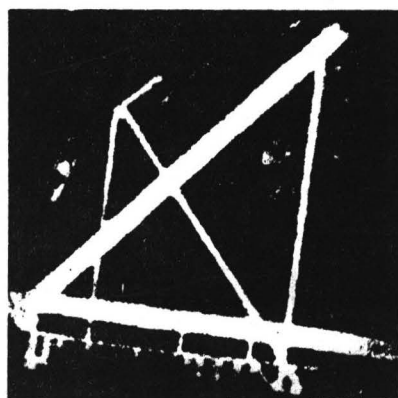
Fig. 1. Possible Successor Locations for Two Edges.



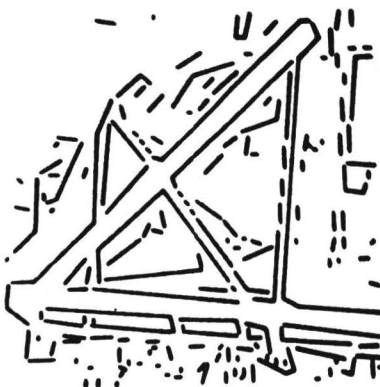
Fig. 2. Three Instances of Two Successors.



Fig. 3. All Instances of Three Successors.



a) Image



b) Linear Segments



c) Anti-Parallel Axes

Fig. 4. Processing of an Airport Image.