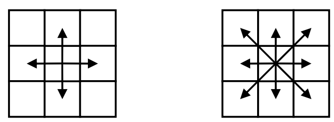
3.3 Edge Tracing Algorithm

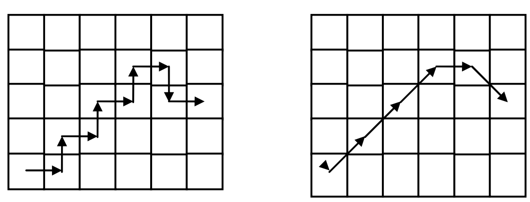
The output of the edge detection algorithm is a binary image. Binary images are coded effectively by differential chain code followed by Huffman coding. Chain code represents an arbitrary curve by assigning a codeword to each direction starting from the one end point of the curve. The purpose of edge tracing algorithm is to obtain the curves to code and to select with respect to their priority and eliminate some of which are less significant.



(a) (b)

Figure 3.3 (a) 4-connected directions,
(b) 8-connected directions

An 8-connected directions are given in Figure 3.3-(b). Our studies have shown that if 4-connected directions given Figure 3.3-(a) are converted to 8-connected directions, then the extraction of model parameters becomes less sensitive to the noise and some false edges may be corrected. An example of 4-to-8 direction conversion is given in Figure 3.4 where the corners of a square is smoothed.



(a) (b)

Figure 3.4 (a) Original edge map,
(b) Edge map after conversion

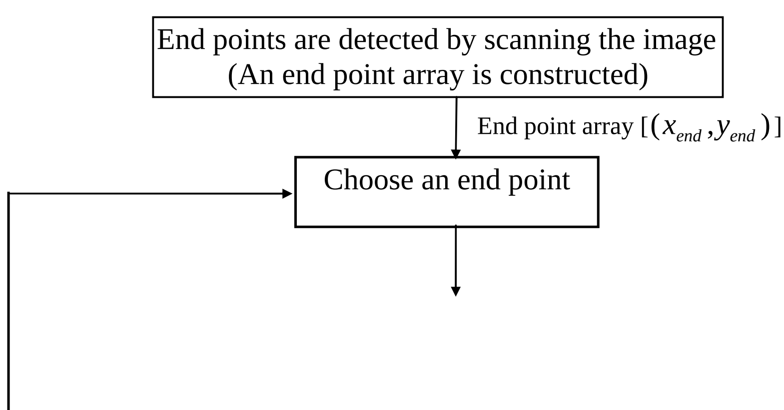
A block diagram of the tracing algorithm is shown in Figure-3.5. End points are obtained by scanning the binary image from left to right and top to bottom in the conventional manner. At the beginning of tracing, all edges are unmarked. When an edge is visited, then the visited edge point is marked as nonedge. If visited edge is on a cross- road point, edge is not marked. Since cross-road points correspond to such points where two or more edge segments intersect, marking a visited cross-road point results in breaking a continuous edge segment into two or more segments which increases the end points. At junction points, edge tracing algorithm is forced to follow the previous direction as much as possible so as to have high compression gain with the differential chain coding. If the visited point is not on a junction, then the next edge is the only edge that is in the neighborhood of the point. Otherwise the next edge is selected as a point which has the closest direction to the previous one.

3.4 Contour Filtering

Edge segments, $S = \{ s_i \mid i = 1, \ldots, N \}$, are obtained by the edge tracing algorithm explained above. These contours have to be filtered because of

- Edge detector gives some false contours,
- Since edge segments have different amount of contribution on reconstruction performance, some contextually unimportant features can be eliminated.

We have observed that edge segments can be classified and ordered with respect to their length, mean contrast calculated through normal direction of the segment and mean curvature directly derived from differential chain code. Because contextually unimportant information may consume much of the channel capacity, we offer a method controlling the selection of edge segments in a top-down manner. Through the ordering of edge segments, one can state that if leading edge segments in the order are used and others are omitted, since remaining edge segments are still enough



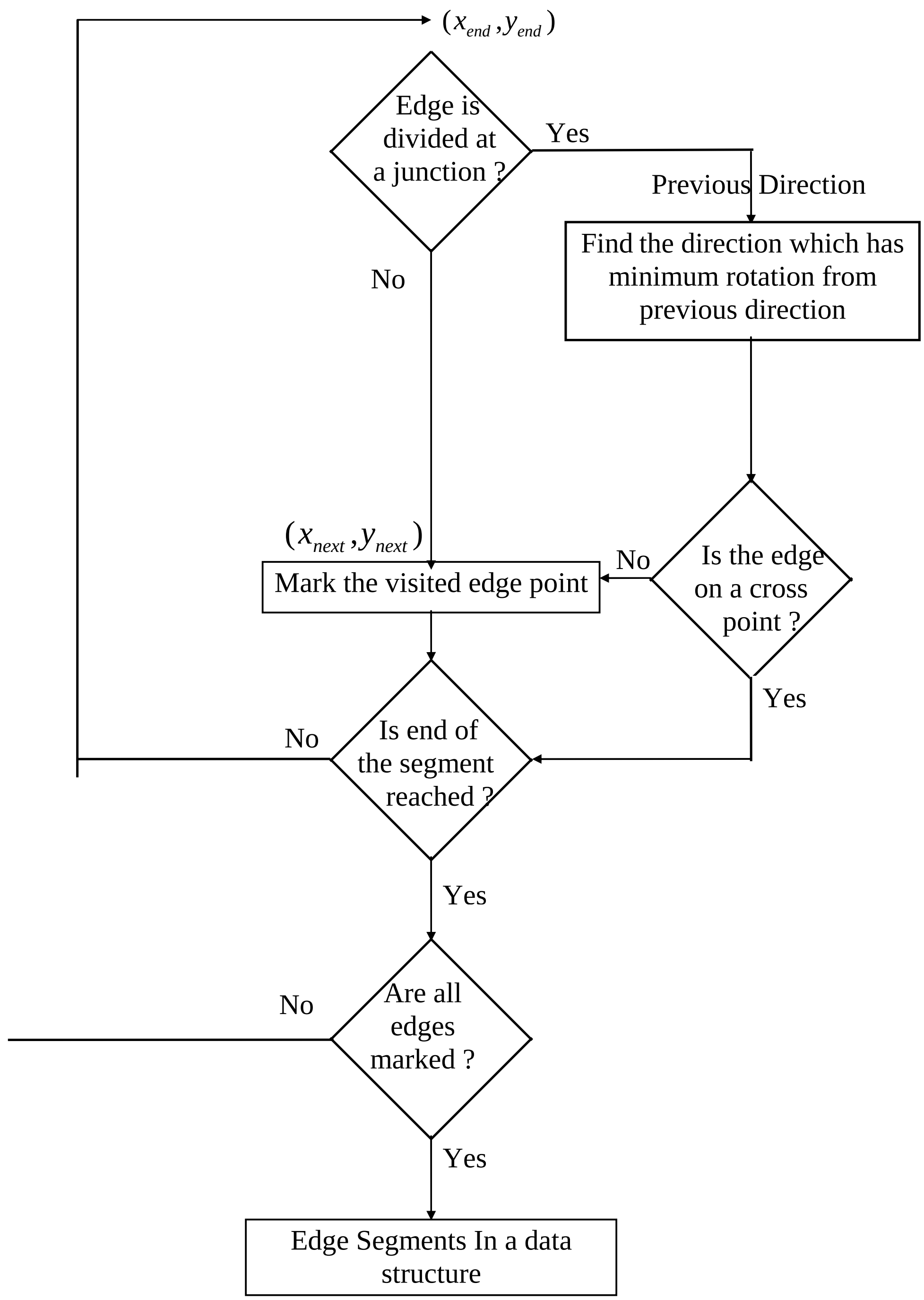


Figure 3.5 Edge Tracing Algorithm

to represent the image perceptually, the reconstructed image is still a good approximation of the original. In order to prove the validity of the selection algorithm, we have assigned a priority to each segments and eliminated some of them which has less priority for a given threshold. All segment features (length, contrast and curvature) are normalized.

$$\text{Priority}(s_i) = w_{\text{length}} \times \mathit{Length}(s_i) + w_{\text{contrast}} \times \mathit{Contrast}(s_i) + w_{\text{curvature}} \times \mathit{Curvature}(s_i) \quad (3.3)$$

The selection method is tested on three type of images having completely different characteristics. One type is a medical image (BRAIN.HIPS). In medical images, actually all features are perceptually important in some degree. Another type of image is a highly detailed one containing face (LENNA.HIPS). Other type of image contains broad regions and naturally very sparse and smooth edge data (HOUSE.HIPS). In Table 3.2-5, how SNR, NMSE, PSNR vary with the selection of edges is given. As contours are eliminated, NMSE increases, SNR and PSNR decreases expectedly, but reconstructed images (Figures 3.6-3.8, respectively) still contain most perceptual features.

Table 3.2 Selection and reconstruction results for BRAIN.HIPS
($w_{\text{length}} = -1.0,\ w_{\text{contrast}} = -1.0$)

Contour	Edge	%Threshold	NMSE	SNR(dB)	PSNR
339	4472	100	30.83	23.52	67.77
236	3998	75	31.36	23.19	67.43
164	3506	50	31.72	22.96	67.20
84	2855	25	33.01	22.16	66.40
36	1982	10	34.56	21.24	65.48

Table 3.3 Selection and reconstruction results for LENNA.HIPS
($w_{length} = -1.0,\ \ w_{contrast} = -1.0, w_{curvature} = 0.5$)

Contour	Edge	Threshold%	NMSE	SNR(dB)	PSNR
232	2754	100	20.73	31.46	63.47
165	2494	75	27.40	25.88	57.89
110	2199	50	28.98	24.76	56.88
56	1872	20	35.40	35.40	52.77

Table 3.4 Selection and reconstruction results for HOUSE.HIPS
($w_{length} = -1.0,\ \ w_{contrast} = -1.0$)

Contour	Edge	Threshold%	NMSE	SNR(dB)	PSNR
635	6626	100	12.46	41.60	72.08
475	6094	75	12.64	41.35	71.83
317	5336	50	13.37	40.23	70.70
160	4150	25	14.21	39.02	69.49
97	3466	20	15.39	37.41	67.89
64	3102	10	16.10	36.52	66.99
32	2663	5	18.66	33.56	64.04