and [21]). Additional properties of the proposed transformation are currently under investigation.

ACKNOWLEDGMENT

We wish to thank J. Rubinstein for his helpful suggestions regarding the proof of Theorem 4. We are grateful to E. Goldberg for improving the presentation of this correspondence.

REFERENCES

[1] C. T. Zahn and R. Z. Roskies, "Fourier description for plane close curves," *IEEE Trans. Comput.*, vol. C-21, pp. 269–281, Mar. 1972.
[2] S. A. Dudani, K. J. Breeding, and R. B. McGhee, "Aircraft identification by moment invariants," *IEEE Trans. Comput.*, vol. C-26, pp. 39–46, Jan. 1977.
[3] A. Kalvin, E. Schonberg, J. T. Schwartz, and M. Sharir, "Two-dimensional, model-based, boundary matching using footprints," *Int. J. of Robotics Res.*, vol. 5, no. 4, pp. 38–55, 1986.
[4] J. L. Turney, T. N. Mudge, and R. A. Volz, "Recognizing partially occluded parts," *IEEE Trans. Pattern Anal, Machine Intell,*, vol. PAMI-7, pp. 410–421, July 1985.
[5] J. Hong and H. Wolfson, "An improved model-based matching method using footprints," in *Proc. 9th Int. Conf. Pattern Recognit.*, Rome, Italy, 1988, pp. 72–78.
[6] J. W. Gorman, O. R. Mitchell, and F. P. Kuhl, "Partial shape recognition using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-10, pp. 257–266, Mar. 1988.
[7] D. P. Huttenlocher and S. Ullman, "Object recognition using alignment," in *Proc. 1st Int. Conf. on Comput. Vision*, London, 1987, pp. 102–111.
[8] Y. Lamdan, J. T. Schwarts, and H. Wolfson, "On recognition of 3-D objects from 2-D images," in *Proc. IEEE Int. Conf. on Robotics and Automat.*, Philadelphia, PA, 1988, pp. 1407–1413.
[9] A. Rosenfeld and J. S. Weszka, "An improved method of angle detection on digital curves," *IEEE Trans. Comput.*, vol. C-24, pp. 940–941, Sept. 1975.
[10] H. Freeman and L. S. Davis, "A corner-finding algorithm for chain-coded curves," *IEEE Trans. Comput.*, vol. C-26, pp. 297–303, Mar. 1977.
[11] M. A. Fischler and R. C. Bolles, "Perceptual organization and curve partitioning," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 100–105, Jan. 1986.
[12] E. E. Milios, "Shape matching using curvature processes," *Comput. Vision, Graphics, and Image Processing 47*, pp. 203–226, 1989.
[13] F. Mokhatarian and A. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 34–43, Jan. 1986.
[14] H. Asada and M. Brady, "The curvature primal sketch," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 2–14, Jan. 1986.
[15] D. D. Hoffman and W. A. Richards, "Representing smooth plane curves for recognition: Implications for figure-ground reversal," in *AAAI-82 Nat. Conf. on Artificial Intell.*, pp. 5–8, 1982.
[16] J. J. Stocker, *Differential Geometry.* New York: Wiley-Interscience, 1969.
[17] M. P. Do Carmo, *Differential geometry of curves and surfaces.* Englewood Cliffs, NJ: Prentice-Hall, 1976.
[18] I. D. Faux and M. J. Pratt, *Computational Geometry for Design and Manufacture.* Chichester, Sussex, England: Ellis Horwood Limited, 1979.
[19] L. O'Gorman, "Curvilinear feature detection from curvature estimation," in *Proc. Int. Conf. on Pattern Recognition*, Rome, Italy, 1988, pp. 1116–1119.
[20] K. Deguchi, "Multi-scale curvatures for contour feature extraction," in *Proc. Int. Conf. on Pattern Recognit.*, Rome, Italy, 1988, pp. 1113–1115.
[21] N. Katzir, M. Lindenbaum, and M. Porat, "Curve segmentation under partial occlusion," EE PUB No. 730, Technion, Haifa 1989.
[22] B. Chazelle and H. Edelsbrunner, "An optimal algorithm for intersecting line segments in the plane," in *29th Annu. Symp. Foundations of Comput. Sci.*, Oct. 1988, pp. 590–599.
[23] J. L. Bentley and T. A. Ottmann, "Algorithms for reporting and counting geometric intersections," *IEEE Trans. Comput.*, vol. C-28, pp. 643–647, Sept. 1979.
[24] N. Katzir and B. Sidlesky, "A fast intersection-finding algorithm," Internal Rep., Computer Vision Lab., Technion, Haifa, 1992.

# Finding Line Segments by Stick Growing

Randal C. Nelson

***Abstract*—A method is described for extracting lineal features from an image using extended local information to provide robustness and sensitivity. The method utilizes both gradient magnitude and direction information, and incorporates explicit lineal and end-stop terms. These terms are combined nonlinearly to produce an energy landscape in which local minima correspond to lineal features called sticks that can be represented as line segments. A hill climbing (stick-growing) process is used to find these minima. The method is compared to two others, and found to have improved gap-crossing characteristics.**

***Index Terms*—Line detection, feature extraction, edge detection, energy minimization, gradient descent.**

## I. INTRODUCTION

Finding lineal features in an image is an important step in many object recognition and scene analysis procedures, if only because they constitute a mathematically simple intermediate level primitive that can be used to describe many man-made objects. There have been several approaches to extracting lineal primitives, all of which have their particular advantages and disadvantages. We present here a method that seems to have different properties than the commonly used procedures.

The most widely used method involves edgel linking and segmentation. The basic idea is to find local edge pixels using some low-level process, link them into contours on the basis of proximity and orientation, and then segment the contours into relatively straight pieces, again using any of several processes. The classic example of this approach is the Nevatia–Babu line detector [10]. Other examples include work by Zhou *et al.* [12], Nalwa and Pauchon [9], and Etemadi [4]. The last, which is used here for comparison, finds chains of edgels using the Marr–Hildreth edge finder, segments the chains into pieces that are symmetric about their centroid, and then attempts to hook these into longer segments. The main problem with edgel linking approaches is that they tend to be very sensitive to the output of the edgel finder. They tend to be unstable in the presence of clutter, and have trouble bridging gaps. The segmentation process can also be unstable, particularly if there is any bumpiness to the structure. Some of these problems can be ameliorated using multiresolution representations, e.g., [5], and grouping techniques [7].

A second method of line detection is based on the Hough transform [3]. Here local edges vote for all possible lines they are consistent with, and the votes are tallied up later to determine what lines are

Manuscript received December 19, 1991; revised November 1, 1993. Recommended for acceptance by Associate Editor J.-O. Eklundh.

The author is with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA; e-mail: nelson@cs.rochester.edu.

IEEE Log Number 9400027.

0162–8828/94$04.00 © 1994 IEEE

Authorized licensed use limited to: Fachhochschule FH Darmstadt. Downloaded on May 23, 2009 at 07:56 from IEEE Xplore. Restrictions apply.

actually present. A subsequent verification phase may also take place. The main problems with this approach are complexity, coarse resolution, and lack of locality. The method is expensive to implement, because every edgel must vote for all the lines it is consistent with. This can be a large number, depending on the desired resolution and on how the segment space is parameterized. The method is nonlocal and bridges gaps well, but may combine unrelated data. Hence, extensive postprocessing is often needed. Princen *et al.* [11] address some of these problems using a hierarchical grouping process in conjunction with a local Hough transform.

A third method of lineal feature detection due to Burns *et al.* [1] utilizes the gradient direction to partition the image into a set of support regions, each of which will presumably be associated with a single feature. A least-squares fitting procedure is then used to fit a line segment to each region. This method can detect low-contrast features, but the segmentation can be unstable. Also, features can rather easily be broken up by local perturbations, i.e., the gap problem again.

Finally, there are statistical approaches. Mansouri *et al.* [8] propose a hypothesize-and-test algorithm to find line segments of a given length by hypothesizing their existence based on local information, and attempting to verify that hypothesis statistically on the basis of a digital model of an ideal segment edge. This method has some similarities to the one described here, but does not provide an efficient method to fit maximal segments.

## II. Description of Method

The essence of the method described here is to define a metric that assigns a score to any possible line segment, based on the underlying image data, and repeatedly extract the best segment from the image. The practical problems are, first, to design an appropriate matching measure, and, second, to make the method efficient since it is clearly impractical to look through all possible segments multiple times. The efficiency problem can potentially be dealt with using any of several approximate maximization techniques. In this case, the problem is well behaved enough that a hill climbing method is effective. This will be described later.

The issue of designing a matching criterion for mapping lineal features to line segments is a bit subtle. The main difficulty is that, while a line segment is well defined mathematically, the notion of a lineal feature is a subjective one. One approach is to pick a mathematical definition of a lineal feature (e.g., it is an extended step edge), and compute the corresponding matching measure. This is the approach taken by Canny in his edge detector [2]. The step edge assumption, however, does not exactly hold in real images. Another approach is simply to state a matching measure, argue that it corresponds to the subjective notion of a lineal feature, and then demonstrate that it is useful. This approach is frequently criticized as ad hoc, but at least it rests on empirical evidence. It is the approach taken here.

Intuitively, a lineal feature consists of a straight part and two ends. Hence, we define a matching criterion that includes explicit representations for the straight section of the feature and the end stops. We call the combined representation a *stick*. The matching criterion is applied to the image as follows. The gradient magnitude and direction are determined by applying a set of local convolution operators to the image. Edge points are determined by applying nonmaximum suppression in the direction of the gradient. The edge points are then assigned a value proportional to the gradient magnitude and the nonedge points are set to zero.

To compute the match-score for a particular stick, the above edge image is convolved with three templates at the appropriate positions and orientations. One of these represents a straight segment, the other two end-stop patterns. The straight segment is a Gaussian whose central profile has been linearly extended, and the end-stop patterns are differences of Gaussians with centers separated by two standard deviations. The straight segment correlation is computed using only points whose gradient direction is consistent with the direction of the segment. The match score is computed by adding the straight correlation value to any positive response from the end-stop measures. Negative values from the end-stop templates are set to zero. This nonlinearity prevents a sudden brightening in the line from inhibiting the growth of a stick.

Sticks are fitted to lineal features by first finding a high-gradient starting point. Starting with a short initial stick aligned perpendicular to the local gradient and centered at the starting point, a hill climbing procedure is performed, varying the centerpoint, length, and orientation of the stick incrementally to maximize the match score.

A few additional practical details are involved. Since it is inefficient to compute the entire straight correlation at each step for longer sticks, the full match value is computed only for sticks up to a certain length (about 14 pixels in the current implementation). During this phase, the hill climbing is performed by cycling through the four dimensions and determining whether a step in any one of them will increase the match value. When no step along a single dimension increases the score, or when the maximum length is reached, the process terminates. This constitutes a seed segment. Beyond that point, the effect of extension at both ends is explored by probing out from a base point (initially the center point of the seed) with extension templates with a restriction that the orientation can change only slightly from that of the seed. Should extension be indicated, then a new basepoint is selected from among the three adjacent pixels that would increase the length of the stick by finding the best match among the nine permitted basepoint/angle combinations. When a local maximum is finally reached, if the final stick has a length exceeding a selected threshold, the points contributing to the final score are marked and eliminated from contributing to other segment scores. As with many hill climbing processes, the space here is not convex, which means that the global maximum is not necessarily found. Nevertheless, the algorithm gives good extraction of segments.

In order to find multiple segments, the image is broken up into neighborhoods, and sticks are grown starting at the top $N$ locations in each neighborhood. A new starting location is not selected until completion of the growth phase of the previous stick, since some previously attractive start locations may be subsumed by the new feature. A stick can grow out of its original neighborhood, which can have the effect of eliminating some start-point candidates in others it passes through. The following pseudocode summarizes the algorithm.

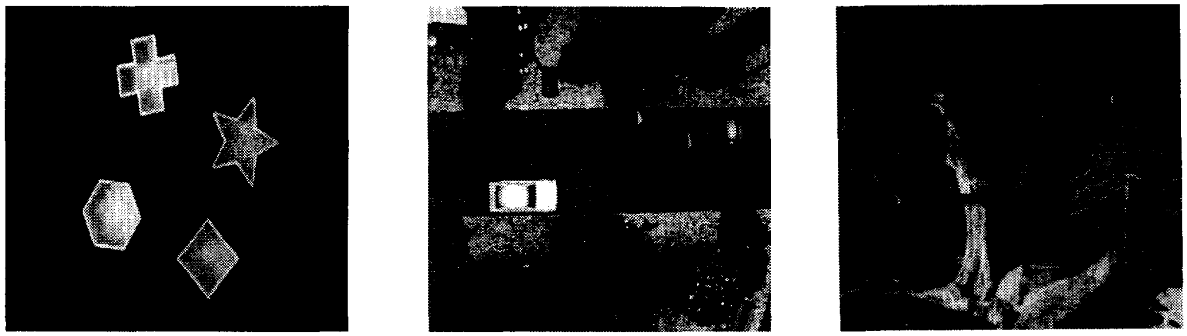
*Line Finding Algorithm:*

1) Preprocess image and extract local edge points and gradients.
2) Break image into disjoint regions ($32 \times 32$ blocks).
3) For each region, do the following.
    a) Find point with highest gradient not yet used in a segment.
    b) Starting at that point, grow segment using hill climbing process.
    c) If segment length exceeds minimal threshold (here 10 pixels), record it, and mark pixels within one pixel of segment as used.
    d) If more than $N$ (here 4) segments have been started in region, go to next region. Otherwise, go to step a.

*Segment Growing Algorithm:*

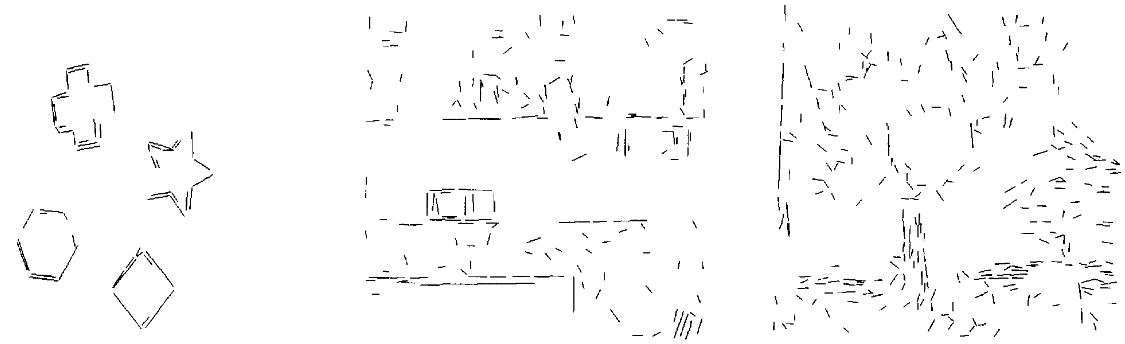
1) Initialize process with short stick centered at start point and oriented perpendicular to the local gradient.

Authorized licensed use limited to: Fachhochschule FH Darmstadt. Downloaded on May 23, 2009 at 07:56 from IEEE Xplore. Restrictions apply.



(a) (b) (c)

Fig. 1. Original images.



(a) (b) (c)

Fig. 2. Results of edgel linking method.

2) Repeat until no change in stick occurs or length grows beyond local limit (here 14 pixels). This is the hill climbing process.
   a) Try to improve match score by shifting segment center one pixel left or right.
   b) Try to improve match score by shifting segment center one pixel up or down.
   c) Try to improve match score by increasing segment length 1 unit (about 1/2 pixel).
   d) Try to improve match score by shifting orientation one unit clockwise or counterclockwise (1/32 of a revolution).
3) If loop exits on length condition, extend ends of segment.

*End Extension (Tip Growing) Algorithm:*

1) Initialize tip as half of seed segment.
2) Decide if tip can be extended by checking extension match in current and neighboring orientations.
3) If extension is indicated, do a local search in location and orientation for the best new tip, and go to step 2.

The method described above has some features in common with the "snakes" of Kass *et al.* [6] in that it is an optimization process for fitting a geometric structure to data on the basis of an energy minimization procedure. The tradeoff between strain energy and match to the image is not made explicit in this case, and the nonlinear nature of the end-stop response adversely affects some solution methods.

## III. Experiments and Discussion

The stick-growing method described above was implemented and tested on a number of images. The images were first smoothed by a Gaussian with a standard deviation of 2 pixels. Edgels were extracted by convolving the smoothed image with the four $3 \times 3$ Kirsch operators and finding the maximum. The edgel map was thinned by performing nonmaximum suppression in the direction of the gradient. The image was broken into $32 \times 32$ pixel windows, and up to 4 segments could be initiated in each window. Increasing this value had little effect, even for the more complex scenes. The minimal segment length was set to 10 pixels for all methods.

Seed sticks were grown using a set of $17 \times 17$ templates representing sticks in 32 orientations and 16 lengths between 3.5 and 14 pixels. The standard deviation of the component Gaussians was 1.7 pixels for both the linear and the end-stop templates regardless of length. The extension templates are essentially formed by cutting the two longest sets of linear templates in half perpendicular to the segment, and using just one end stop. Comparing the results of the longest to the next size down determines whether a stick can be extended. When determining scores, the linear segment score is weighted four times as heavily as the (positive) end-stop score. This arrangement will terminate segment growth when, e.g., an edgel chain drops to 1/5 or less of its previous value for more than about 3 pixels.

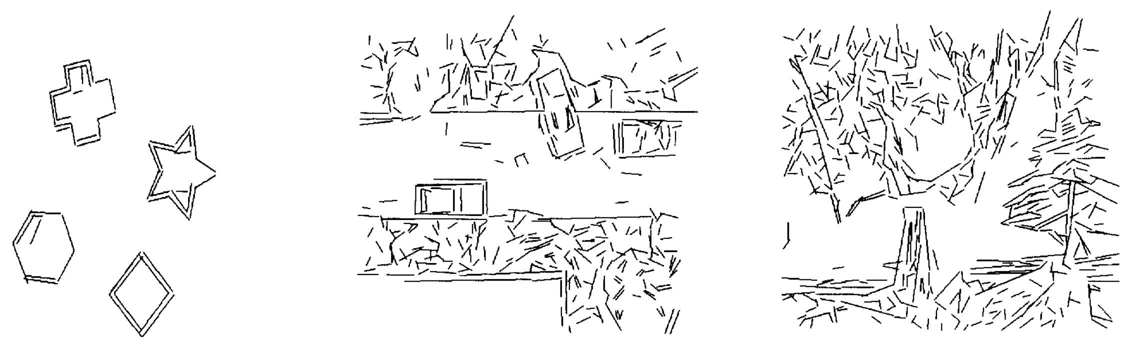
As can be seen from the discussion above, there are several parameters associated with the method. In rough order of relative effect on the algorithm these are: amount of initial smoothing; relative weight of linear and end-stop templates; gradient intensity threshold for starting seed growth; size of subimages and number of starts allowed in each (mainly run-time efficiency effect); and step size in the search. All of these parameters were set *a priori,* and remained constant for all tests. Basically, once a good set was found, they worked well for all images.

Figs. 1–4 and Tables I–III show the results of applying our method and two others to three images. Fig. 1 contains the original images;

Authorized licensed use limited to: Fachhochschule FH Darmstadt. Downloaded on May 23, 2009 at 07:56 from IEEE Xplore. Restrictions apply.

(a) (b) (c)

Fig. 3. Results of support region method.



(a) (b) (c)

Fig. 4. Results of stick-growing method.

TABLE I
SEGMENT STATISTICS FOR POLYHEDRAL IMAGE

| Length Range | Stick Growing | | | Support Region | | | Edgel Linking | | |
|---|---|---|---|---|---|---|---|---|---|
| | Count | Cum Count | Cum Length | Count | Cum Count | Cum Length | Count | Cum Count | Cum Length |
| 60–69 | 5 | 5 | 315 | 4 | 4 | 260 | 1 | 1 | 62 |
| 50–59 | 4 | 9 | 541 | 0 | 4 | 260 | 2 | 3 | 177 |
| 40–49 | 7 | 16 | 850 | 7 | 11 | 571 | 3 | 6 | 315 |
| 30–39 | 17 | 33 | 1422 | 17 | 28 | 1159 | 12 | 18 | 714 |
| 20–29 | 22 | 55 | 1957 | 14 | 42 | 1516 | 14 | 32 | 1056 |
| 10–19 | 6 | 61 | 2053 | 8 | 50 | 1642 | 26 | 58 | 1406 |

Fig. 2 the output of an edgel linking/segmentation procedure due to Etemadi [4]; Fig. 3 the output of the region of support method of Burns *et al.* [1]; and Fig. 4 the output of our method. The tables give histogram statistics for the segments found. Only segments longer than 10 pixels were kept. The amount of significant structure in the images that is well approximated by line segments varies considerably. The polygonal objects in the first image have a few strong, straight boundaries. The segments produced by all three algorithms are similar, although the one described here exhibits slightly less tendency to break up edges, and seems to localize the corners better. It also found about 50% more boundary, as can be seen from Table I.

The street scene contains considerably less linear structure. Building boundaries and the outlines of the automobiles are still pulled out by all the algorithms; ours fragments these boundaries somewhat less. The bush outlines are considerably harder. Our algorithm finds more extended structures around the boundaries of the bushes. Table II indicates that our method found approximately three times as much boundary as the other two algorithms. The segments found do correspond to underlying physical structure, but the decomposition is not particularly stable since a line segment set is not a particularly good representation for a bush. The tree scene is the least well structured in terms of linear features. The strongest are associated with large limbs and the trunks of the trees. These are generally well located by our method. The same structures are considerably more fragmented by the other algorithms. As in the previous natural scene, our method finds three to five times as much boundary as the other two algorithms, and many more long structures.

Comparing the time complexity of the different methods is a little tricky. Roughly, all used comparable total amounts of time for the different images, i.e., within a factor of two or three, but the factors that affect the time are different for the different methods.

Authorized licensed use limited to: Fachhochschule FH Darmstadt. Downloaded on May 23, 2009 at 07:56 from IEEE Xplore. Restrictions apply.

TABLE II
SEGMENT STATISTICS FOR STREET SCENE

| Length Range | Stick Growing | | | Support Region | | | Edgel Linking | | |
|---|---|---|---|---|---|---|---|---|---|
| | Count | Cum Count | Cum Length | Count | Cum Count | Cum Length | Count | Cum Count | Cum Length |
| 150–200 | 4 | 4 | 681 | 0 | 0 | 0 | 1 | 1 | 167 |
| 100–149 | 2 | 6 | 968 | 0 | 0 | 0 | 0 | 1 | 167 |
| 80–99 | 6 | 12 | 1495 | 3 | 3 | 274 | 0 | 1 | 167 |
| 60–79 | 3 | 15 | 1691 | 2 | 5 | 417 | 0 | 1 | 167 |
| 40–59 | 21 | 36 | 2610 | 11 | 16 | 937 | 9 | 10 | 581 |
| 30–39 | 20 | 56 | 3275 | 13 | 29 | 1382 | 6 | 16 | 797 |
| 20–29 | 85 | 141 | 5305 | 16 | 45 | 1753 | 26 | 42 | 1427 |
| 10–19 | 241 | 382 | 8701 | 79 | 124 | 2741 | 124 | 166 | 2994 |

TABLE III
SEGMENT STATISTICS FOR TREE SCENE

| Length Range | Stick Growing | | | Support Region | | | Edgel Linking | | |
|---|---|---|---|---|---|---|---|---|---|
| | Count | Cum Count | Cum Length | Count | Cum Count | Cum Length | Count | Cum Count | Cum Length |
| 80-102 | 6 | 6 | 532 | 0 | 0 | 0 | 0 | 0 | 0 |
| 60-79 | 9 | 15 | 1123 | 1 | 1 | 74 | 0 | 0 | 0 |
| 50-59 | 14 | 29 | 1866 | 0 | 1 | 74 | 1 | 1 | 52 |
| 40-49 | 31 | 60 | 3249 | 0 | 1 | 74 | 2 | 3 | 150 |
| 30-39 | 35 | 95 | 4410 | 7 | 8 | 308 | 8 | 11 | 411 |
| 20-29 | 112 | 207 | 7049 | 26 | 34 | 913 | 20 | 31 | 866 |
| 10-19 | 301 | 508 | 11354 | 86 | 120 | 2053 | 239 | 270 | 3904 |

All have a preprocessing phase that is proportional to the number of pixels in the image. After this, the methods diverge. The region of support method must perform a connected components analysis on the gradient direction, and hence is adversely affected by highly textured images. The edgel linker tries to hook together all the edgels found, and hence is primarily dependent on the number of edgels in the preprocessed image. The short ones are later discarded. The stick-growing method runs in time proportional to the total length of boundary found, but does not necessarily look at all the edgels.

## IV. CONCLUSION

We have described a method for extracting intermediate-level linear features using energy minimization in a landscape derived from spatially extended operators. The method differs from previous approaches primarily in the use of extended measures coupled with end-stop detectors. This allows it to cross gaps, use sparse data, and resist disruption by local anomalies, while still accurately locating the end of a line. The method also avoids the expensive and involved postprocessing steps involved in using the Hough transform to locate lineal features. The use of a hill climbing method ensures that the solutions found will at least be local minima of the energy function. As illustrated by the above examples, the method works fairly well on images containing strong lineal features, and finds relevant features in cluttered images that some other methods do not. It is not particularly well adapted to images containing many curved contours, as the segmentation induced is not particularly stable. However, the technique is easily adaptable to curve growing, and can be made to terminate at curvature maxima or inflection points, which would tend to yield a more robust segmentation. This has been done with good results.

## ACKNOWLEDGMENT

Many thanks are due to L. Wixson for producing the results for the Burns *et al.* support region line finder, and to P. VonKaenel for producing those for the Etemadi linking algorithm.

## REFERENCES

[1] J. B. Burns, A. R. Hanson, and E. M. Riseman, "Extracting straight lines," in *Proc. DARPA IU Workshop,* New Orleans, LA, 1984, pp. 165–168.
[2] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-8, pp. 679–698, 1986.
[3] R. O. Duda and P. E. Hart, "Use of the Hough transform to detect lines and curves in pictures," *Commun. ACM,* vol. 15, pp. 11–15, 1972.
[4] A. Etemadi, "Robust segmentation of edge data," Tech. Rep., Univ. Surrey, 1990.
[5] T. H. Hong, M. O. Shneier, R. L. Hartley, and A. Rosenfeld, "Using pyramids to detect good continuation," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-13, pp. 631–635, 1983.
[6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.,* vol. 1, pp. 321–331, 1988.
[7] D. G. Lowe, *Perceptual Organization and Visual Recognition.* Norwell, MA: Kluwer Academic, 1985.
[8] A. Mansouri, S. Malowany, and M. D. Levine, "Line detection in digital pictures: A hypothesis prediction/verification paradigm," *Comput. Vis., Graph., Image Processing,* vol. 40, pp. 95–114, 1987.
[9] V. S. Nalwa and E. Pauchon, "Edgel aggregation and edge description," *Comput. Vis., Graph., Image Processing,* vol. 40, pp. 79–94, 1987.
[10] R. Nevatia and K. R. Babu, "Linear feature extraction and description," *Comput. Vis., Graph., Image Processing,* vol. 33, pp. 257–269, 1980.
[11] J. Princen, J. Illingworth, and J. Kittler, "A hierarchical approach to line extraction based on the Hough transform," *Comput. Vis., Graph., Image Processing,* vol. 52, pp. 57–77, 1990.
[12] Y. T. Zhou, V. Venkateswar, and R. Chellappa, "Edge detection and linear feature extraction using a 2-D random field model," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 11, pp. 84–95, 1989.

Authorized licensed use limited to: Fachhochschule FH Darmstadt. Downloaded on May 23, 2009 at 07:56 from IEEE Xplore. Restrictions apply.