DETECTING FEATURES IN LOW RESOLUTION AERIAL IMAGES OF CITY BLOCKS

Bill Sakoda
Joseph G. Hauk

Computer Science Department
State University of NY at Stony Brook
Stony Brook NY 11794
and
Grumman Data Systems
1000 Woodbury Road
D12-237
Woodbury NY 11797

Contact: Bill Sakoda
wjs@sbcs.sunysb.edu
(516) 632-8439

ABSTRACT

This project is a case study of extraction of roads and houses from low-resolution infrared aerial photographs of city block areas. Houses and roads are about 2 pixels wide. Infrared renders houses, connecting driveways, and roads light, with significant blurring. The situation is challenging because of the similar imaging of the objects of interest and the low resolution.

We show how to combine regions from thesholding, a residual-based edgefinder, and spot (house) identification through a modified Gaussian curvature to obtain road networks and houses. A tree growing procedure for aggregating points in the plane is developed, and applied to find smooth trajectories through detected building locations, yielding rows of houses along roads. In addition to proposing a practical method for this problem domain, we hope that this and similar studies contribute to development of techniques for low-level feature extraction and methods for combining them.

INTRODUCTION

This project is a case study of extraction of roads and houses from low-resolution infrared aerial photographs of city block areas. Houses and roads are about 2 pixels wide. Infrared renders houses, connecting driveways, and

Presented at Auto Carto 10, March 23-29 1991, Baltimore MD.

roads light, with significant blurring. The situation is challenging because of the similar imaging of the objects of interest and the low resolution. We are interested in extracting maximum information from low resolution data because this reduces the number of images that need be captured, and because in surveillance applications, low resolution data may be all that are available.

We show how to combine regions from thesholding, a residual-based edgefinder, and spot (house) identification through a modified Gaussian curvature to obtain road networks and houses. A spanning-tree-based procedure for sending smooth paths through points is developed. Such a procedure has numerous applications because significant physical objects tend to have smooth boundaries, while feature detectors often produce fragments. One can deduce the boundaries by reassembling the fragments. This module is tested by applying it to house locations, yielding rows of houses along roads. In addition to proposing a practical method for this problem domain, we hope that this and similar studies contribute to development of techniques for low-level feature extraction and methods for combining them.

Figure 1 summarizes our results.

PREVIOUS WORK

There has been much work in this area, due to the variety and volume of data awaiting availability of practical systems, and suitability of numerous sub-domains as testbeds for different techniques.

Emphasis of work in the area includes low-level primitives (Nevatia and Babu 1980), detection of cultural objects by rectangular or smoothly curving contours (Fua and Hanson 1987), complete systems based on applying special knowledge of particular sub-domains (Huertas et. al. 1987), and general mechanisms for applying knowledge constraints (McKeown et. al. 1985). Most work spans the range from use of low level vision to acquire basic data, to application of domain-specific knowledge, whether it be applied as a special case or through a general mechanism. It is typical for authors to comment on special characteristics making their domain challenging to automatic analysis. The present study considers a particular sub-domain -- low resolution infrared city blocks -- and shows how mathematically well-behaved primitives can be used in conjunction with world constraints to extract features. A general system will function most efficiently with a library of such sub-domains and a kernel of mathematically precise low-level detectors. Binford (1982) argues lucidly that success in large domains using model-based analysis will require strong low-level modules.

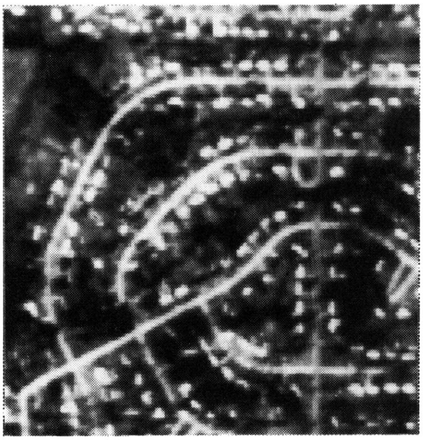


Figure 1a. Source picture CIRCLE. 128 by 128.

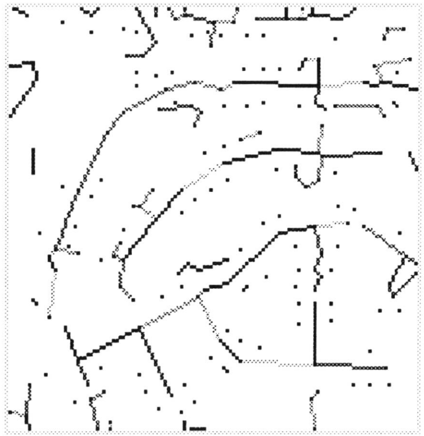


Figure 1b. Features located in CIRCLE. Straight line segments from roads are plotted different shades of gray. Isolated black dots are buildings.

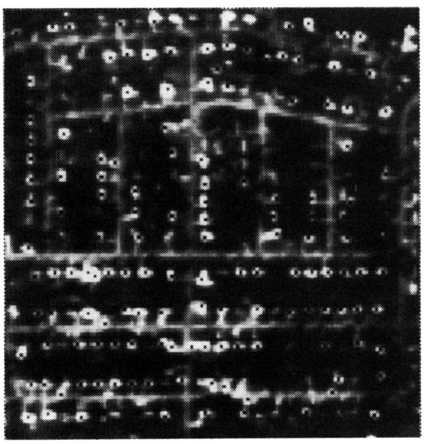


Figure 1c. Picture BLOCK. House Locations found via gaussian curvature marked black. Source image is 128 x 128.

Figure 1d. House grouping results from BLOCK after line fitting (shown in black). Results of road extraction algorithm are shown in grey.

Figure 1: Summary of road and building extraction techniques.

Cultural Feature Detection

Huertas, Cole and Nevatia (1987) demonstrated a system for detection of airport runways from very high resolution photographs. This work showed a nice balance of *simple* but well-considered low (lines) and middle (APAR) level vision, use of hand-tooled high level constraints from the problem domain, and a working demonstration. APARS are approximately parallel edges of opposite contrast (Anti-PARallel), useful for detecting bars or slowly varying ribbon shapes against a contrasting background. They point out that while runways are essentially elongated rectangles, the problem is very challenging because of runway markings, non-uniformity of runway surface (oil spots and shoulders), repair work, vehicles on the tarmac, and intersections. LINEAR (Nevatia and Babu 1980) is used to produce line segments and APARS. A variety of 5 by 5 masks are used to detect edges, which are then thresholded, thinned, linked, and approximated as piecewise linear. APARS are then identified. APAR-based approaches tend to produce many false candidates, especially when a feature has parallel sub-features (eg lines down a runway), and each line can then contributes to many APARS. This is handled by histogramming APAR widths, and selecting candidates with widths appropriate for runways, shoulders and markings. APARS are joined by analyzing continuity, collinearity, and gap texture. Finally, hypotheses of positioning of runway subfeatures are verified from FAA specifications.

Fua and Hanson (1985) used parallel and perpendicular line segments to locate cultural objects in high resolution images. Undersegmentation was resolved by using linking to connect almost-collinear lines, complete corners, and close open-ended U's and parallels. Subsequently (Fua and Hanson 1987), they proposed detecting roads by using linear edge segments to calculate road width and center; fitting a spline to the center; then using the center spline to locate splines for each side of the road. This allows the road to be continued even when one side is lost due to imaging conditions, occlusion, junctions.

Pavlidis and Liow (Pavlidis and Liow 1988) detected regions by following an oversegmented split-and-merge phase with boundary and edge modification based on contrast, boundary smoothness, and image gradient along boundaries.

The integration of top-down and bottom-up analysis has been advocated by many authors. In particular, Matsuyama (1987) presented an image understanding system that generates hypotheses to test for the existence and location objects, according to the results of low-level vision techniques.

Similarly, Nicolin and Gabler (1987) demonstrated a knowledge-based system for interpretation of aerial images of suburban scenes. Their system is divided into several functional units. One unit contains a methods base of low-level image processing techniques and a second unit contains a knowledge base for suburban scenes. The system's control module uses the knowledge base to decide which techniques from the methods base should be applied to the image.

McKeown and Denlinger (1988) constructed a system for high-resolution imagery based on cooperation between a surface correlation tracker and edge tracing. They detect edges using a 5 by 5 Sobel gradient. The correlation tracker, after a design of Quam (1978), looks for patterns such as lane markers and wear patterns. Starting position of the road, its direction, and width are assumed given. The hypothesized road trajectory is tested by pushing a cross-section of the road forward and testing for cross-correlation.

Aviad and Carnine (1988) presented a method for generating hypotheses for fragments of roads, intended to be fed to a road tracker. The Nevatia and Babu (1980) edge finder is used, followed by Road Center Hypothesis detection by antiparallel edges. RCH's are then aggregated by a greedy linker. This is followed by editing by a smoothness checker, and a final linking.

Point Grouping

In our paper, we examine how to group points in the plane (houses locations) into smoothly varying curves that will lend insight into the feature composition of an aerial image. Zahn (1971) applied graph theoretic algorithms to detection of clusters in arbitrary point patterns. By constructing a minimal spanning tree, he is able to cluster dots into groups according to their point density, measured by calculating the local average length of the spanning tree's edges. A histogram of the local point densities is then calculated and categorized. All edges having neighbors of two (or more) different point density categories are deleted. The resulting graph contains a spanning tree for each point cluster.

Stevens (1978) showed that orientation patterns in a field of random dots can be detected by the use of a local support algorithm. Local orientation is found by drawing virtual lines between neighboring points and then searching for the predominant orientation of the virtual lines. For example, Stevens' algorithm can deduce local relationships in a pattern consisting of an original set of random dots together with a duplicated translation, or with a duplicated set expanded about a center. It will also group isolated one-dimensional curves. Since it finds the *major* orientation in two-dimensional neighborhoods, it is not well suited to grouping houses, where there are nearby linear strings of different orientation. Likewise, Zucker (1985) presented an orientation-based process to infer contours from a collection of dots by locally finding the tangent fields.

Tuceryan and Ahuja (1987) performed clustering and linking according to properties of the Voronoi polygons induced by the dots, including area, eccentricity, isotropicity, and elongation. For example, dots around the boundary of a cluster can be identified because they are eccentric within their polygons.

Vistnes (1987) used a statistical model for the detection of dotted lines and curves embedded in a random dot field. His model is based on a local operator that detects regions of differing dot densities.

ROAD AND HOUSE DETECTION

The goal of this phase is efficient generation of a map of as many of the main roads and houses as possible. When in doubt we are conservative, identifying those features in which we have high confidence. We pay particular attention to *connectivity* of the road networks as this is a key semantic feature.

APARS tend to produce disconnected representations at junctions, (e.g., at a Y junction in a road) since the generating parallel edges do not continue all the way to the center of the junction. In contrast thinning naturally preserves the connectivity at the junction. Centers between APARS are much less sensitive to small glitches in data than thinning in finding skeletons for wide objects, but thinning is suitable for the present domain because features are only a few pixels wide. These data are a difficult (though possibly feasible) case for edge linking and APAR detection because very close proximity of houses causes the edge-finder to wander. Our greedy thresholding quickly and simply yields good connectivity over large segments.

Road and house detection runs in 6 steps.

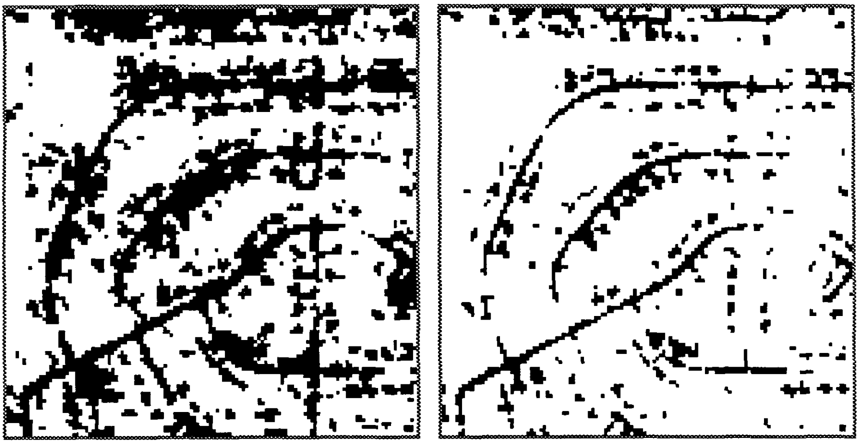
1. (Greedy thresholding): Undersegmented regions consisting of houses, roads driveways, and some adjoining areas are thresholded from the source.
2. (Residual edge cutting): The Lee/Pavlidis/Huang (1988) residual edgefinder is tuned to handle these small-scale data, and edges are used to further segment the regions.
3. (Thinning and small component removal): Resulting regions are thinned. Thinning is careful to respect connectivity, which can now be deduced by local analysis of neighbors. Small components are removed.
4. (Gaussian curvature spot detector): A modified Gaussian curvature spot detector is applied, yielding most house positions.
5. (Trimming): House locations and connectivity of the road network are used to trim the network down to major roads.
6. (Line fitting): The network is decomposed into line segments.

<u>Thresholding</u>

This data set is interesting because neither the regions from thresholding nor edgefinding by themselves yield adequate information about the road networks. A greedy threshold does maintain good connectivity, but blurs the houses into the roads. Adjacent houses blur together, mimicking the linear structure of the roads. More conservative thresholds curtail this aggregation somewhat, but even when the threshold is reduced to the point where the roads begin to disconnect, significant aggregation of distinct features remains. Figure 2 shows the regions obtained at 2 threshold values.

<u>Edgefinding</u>

Our edgefinder is based on the residual technique of Lee, Pavlidis and Huang (1988). They detect edges as zero-crossings of the difference between source image and a regularization of the image. We found that effects of small variations in the image were reduced by applying a mild smoothing to the image first. So, edges are the zero-crossings of the difference of a mildly regularized



(2 a) Image CIRCLE threshold > 80 (2 b) CIRCLE threshold > 110

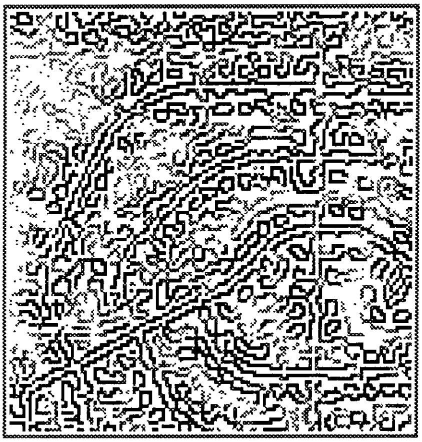
Figure 2: A greedy theshold yield excellent road connectivity but blurs houses into roads. More conservative thresholds yield neater road segments, but leave gaps, eliminate small roads, and still leave some building and driveways attached. We take the undersegmented greedy image and use edges to refine it.

image ($\beta = 5.0$), and a smoother image ($\beta = 1.0$). Zero-crossings are thresholded by slope.

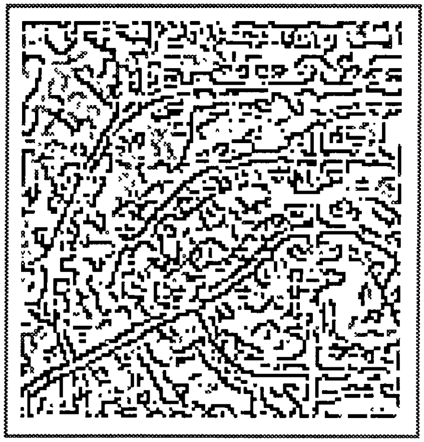
We compared Canny and residual edgefinders at different resolutions. The residual finder tended to generate more closed or almost-closed contours around small features like houses, which is especially useful in the trimming technique we are using. Figure 3 shows Canny and residual edges.

A common method for recording an edge map is to mark edge pixels on a raster the same size as the source image. Here one might choose to mark the edge on the darker side, the lighter side, or on the side nearer zero. Doing so in this case blurs nearby linear features (eg adjacent roads) together. Fortunately zero-crossings have more structure than this - they form contours. A zero-crossing falls between raster positions having positive and negative residual values. We use a zero-crossing tracker which walks this boundary, breaking zero-crossing contours into smooth segments. A raster twice the size of the source is used, with cells having even coordinates holding the source picture, and with zero-crossing information stored between.

This data structure is now used to prune the regions. Since we are trying to preserve the lighter structures, the zero crossing segment walker sets every pixel on the darker side of a zero-crossing to black. This corresponds to a cut when lighter areas are chosen by thresholding. Figure 4 shows the result of removing edge points.

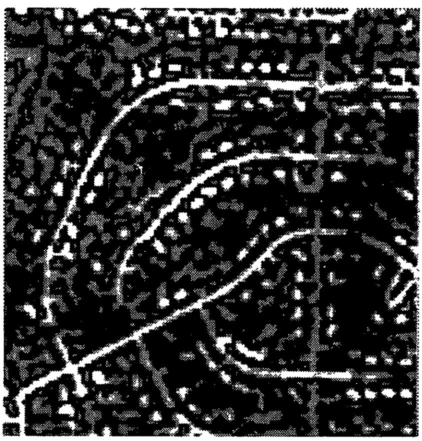


(3 a) Edges from residual edge finder for image CIRCLE

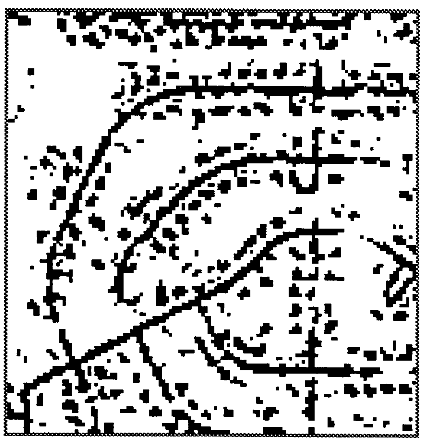


(3 b) Canny edges, $\sigma = 1.0$

Figure 3: Residual and Canny edges. Stronger edges are plotted darker. We found that the residual edges yielded more road boundaries, and tended to trace closed contours around houses.



(4 a) Image CIRCLE with edge points removed (black)



(4 b) Image CIRCLE: edge points removed from greedy threshold

Figure 4: Thresholded regions have good connectivity; Edges are sparse but have good spatial accuracy.

Residual edges in this example accurately delineate many physically significant boundaries but are sparse and would present a difficult case for a purely edge-based technique. A purely edge-based analysis *might* be possible and would be very interesting.

Thinning

The image is thinned (Pavlidis 1982), and small components are removed. The results are in Figure 5.

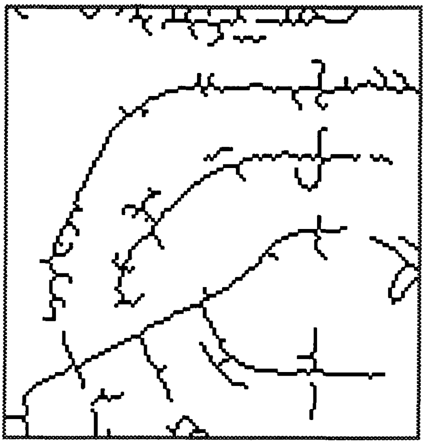


Figure 5: Image CIRCLE after thinning and removal of small components

Gaussian curvature spot finder

Gaussian curvature can be used to construct an efficient operator which responds strongly to small bright spots (eg, houses around 2-3 pixels wide, as they are in our source image), but does not respond to straight edge data (roads).

Gaussian curvature of the gray level picture g(x,y) is given by:

$$\frac{\frac{\partial g^2}{\partial x^2}\,\frac{\partial g^2}{\partial y^2} - \left(\frac{\partial^2 g}{\partial x\,\partial y}\right)^2}{\left(1 + \left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2\right)^2}$$

(See, e.g., Spivak (1970) or Horn (1986). We have been using the numerator of this expression to locate spots. It can be rewritten:

$$\left[\frac{\partial}{\partial x}\left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}\right)\right] \times \left[\frac{\partial}{\partial y}\left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}\right)\right]$$

In this rendering as cross-product of directional derivatives of gradient vectors we can see directly why there is no response to straight edges. All gradients point normal to the direction of the edge. Therefore derivatives also point in this direction, yielding a zero cross-product. This quantity is invariant of orientation of the coordinate system because the Gaussian curvature and denominator both are invariant.

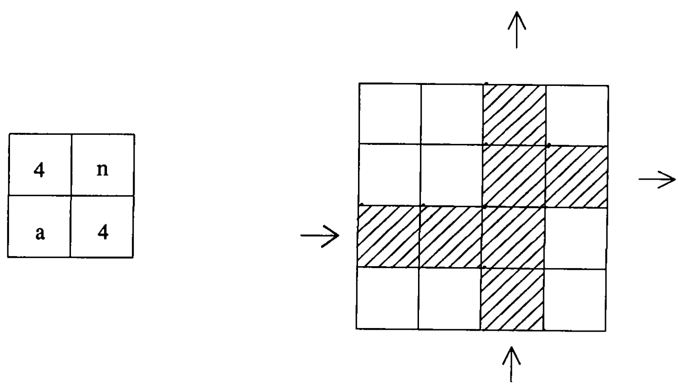
In discrete image space at point (x, y) this curvature c can be computed as:

$$dxy = \left[\frac{1}{4}\right]\left(\left[g_{x+1,y+1} - g_{x+1,y-1}\right] - \left[g_{x-1,y+1} - g_{x-1,y-1}\right]\right)$$
$$d\_sq\_x = g_{x+1,y} + g_{x-1,y} - 2\,g_{x,y}$$
$$d\_sq\_y = g_{x,y+1} + g_{x,y-1} - 2\,g_{x,y}$$
$$c = d\_sq\_x \cdot d\_sq\_y - dxy^2$$

Our Gaussian curvature module finds bright spots by first convolving the input picture with a gaussian; then finding points of high curvature. A spot is reported at locations whose curvature is greater than a specified threshold, and not less than the curvature of its 4 compass neighbors. This tends to mark a single pixel for each bright spot. Figure 1c shows the results of the Gaussian curvature operator.

Trimming via Connectivity analysis

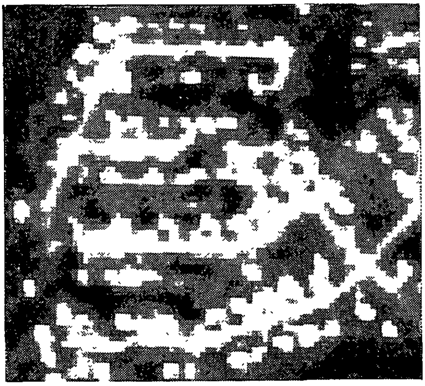
At this point the constructed network has much of the connectivity structure of the underlying roads. Junctions can be located by locally counting neighbors. One must allow for diagonal connectivity to a neighbor if there is no 4-connectivity, as in Figure 6(a). Junctions of degree greater than 3 may be spread over neighboring pixels (Figure 6b).



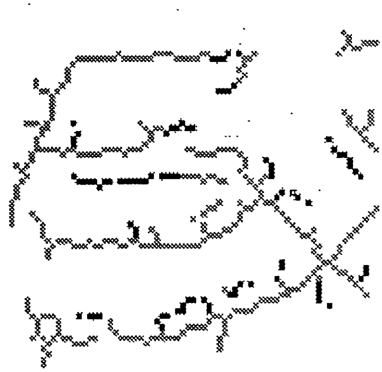
(a) (b)

Figure 6: (a): In looking for junctions, cell "a" should count "n" as a neighbor only if neither of their common 4-neighbors is occupied. (b): When ≥ 4 roads meet, the junction can be spread over nearby cells.

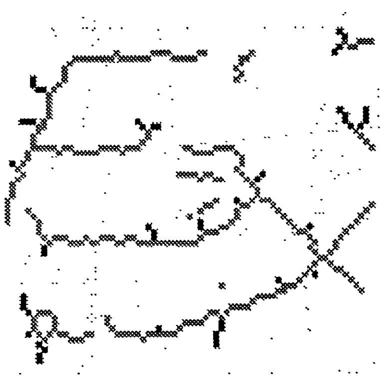
Strings of nearby houses tend to blur together, producing linear bright strips which mimic road structure. A two-stage preening is now undertaken to eliminate these. First, starting from each house point, road pixels are deleted back up to a distance 7, but not past junctions. Stopping at junctions deletes driveways up to a main road without interrupting it. Then short stubs are deleted, by starting at endpoints and looking for a junction within 5 pixels. If a junction is found, the segment from the endpoint the junction is deleted. Figure 7 shows the results.



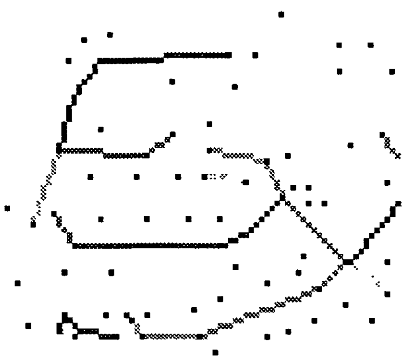
(7 a) Source picture ANGLE; 70 by 80. May be better viewed from a distance (like Harmon's Abraham Lincoln).



(7 b) Spots from Gaussian curvature detector in white. Linear segments attached to houses, to be deleted, in black.



(7 c) Remaining short stubs to be deleted.



(7 d) Final results after line fitting.

Figure 7: Deletion of driveways and aggregated houses. Starting from house locations, road pixels are traced and deleted up to distance 7, but not past junctions. (b): Result after thresholding, edge cutting, thinning, small component deletion, with road segments to be deleted in black. (c): Remaining short stubs to be deleted, in black.

Line Fitting

Straight lines are fit to the thinned segments by testing line segments between successively more distant data points, and breaking at points of sufficiently large maximum distance between the test line and data (Pavlidis 1988b).

DOT GROUPING

In this section, we develop an algorithm for grouping points in two-dimensional Euclidean space, and apply it to detecting the curvilinear structure of houses lying beside roads. The algorithm takes as input a set V of points from R^2 (the detected locations of houses in our application), and constructs a forest joining certain vertices in V. The forest is grown as a sequence of trees, and each tree is grown by adding edges. The cost metric $C(T,v,w)$ designates the cost, possibly ∞, of adding edge (v,w) to the partially constructed tree T. The algorithm is structurally similar to Prim's (1957) greedy algorithm for constructing a minimum cost spanning tree. Our model differs in allowing the metric to be a function of the partially constructed tree. A great deal of control over the grouping can be exercised by varying the metric.

Grouping algorithm

Let $T_{i,j}$ denote the $i\text{-}th$ tree after j edges have been added.

1) (Start a new tree): Let i be the number of trees constructed so far. We let $T_{i+1,1}$ consist of the edge joining a pair of vertices at minimum Euclidean distance, among all vertices not contained in any tree. If no edge is found, the construction is complete.

2) (Grow the current tree): Among all vertices w not in any current tree, and all vertices v in the current tree T_i, find a pair (v_0,w_0) minimizing $C(T_i,v_0,w_0)$. Add it to T_i and iterate this step. If no finite-cost addition can be found, go to (1) to start the next tree.

We have found it computationally and semantically advantageous to disallow edges longer than a parameter $D_{\max}$. This can be subsumed in the model by assigning any edge of length greater than $D_{\max}$ infinite cost, and leads to an implementation where vertices can be assigned to local buckets of size $2D_{\max}$ by $2D_{\max}$, and search for an appropriate neighbor of vertex v can be constrained to at most 4 buckets.

We now demonstrate how our state-dependent metric can be used to advantage. Let

$$C_{curve,\alpha}(T,v,w) = \alpha * Dist(v,w) + (1-\alpha) * angle(T,v,w) .$$

$angle(T,v,w)$ is the absolute value of the angle (in degrees) that is formed between the new branch and the neighboring branch (already included in the current tree T) having the most similar orientation. Parameter α determines the relative weighting between orientation and proximity, with $\alpha = 1$ being the simple minimum Euclidean distance metric. The value $\alpha = .9$ has produced good results in grouping houses in this dataset.

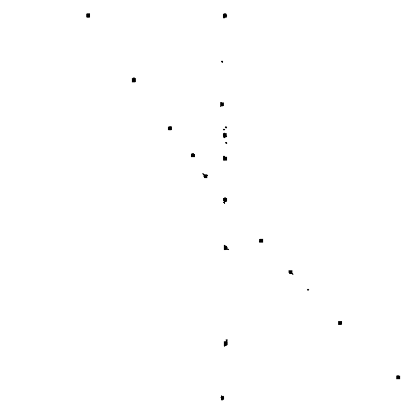


Figure 8a. Example 1: Minimum Euclidean distance spanning tree.

Figure 8b. Example 1: Tree constructed by orientation-sensitive metric $C_{curve,\alpha=.9}$

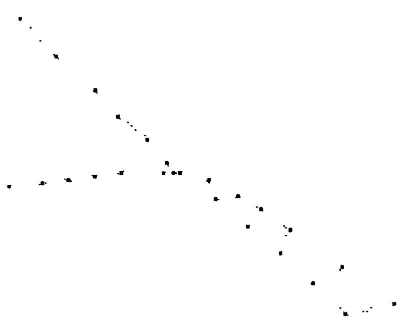


Figure 8c. Example 2: Minimum Euclidean distance spanning tree.

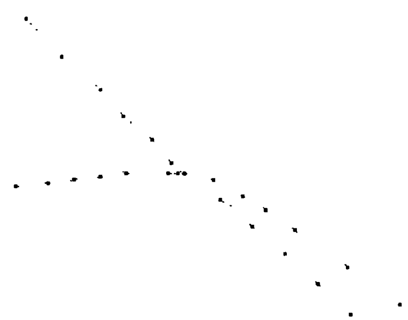


Figure 8d. Example 2: Tree constructed by orientation-sensitive metric $C_{curve,\alpha=.9}$

Figure 8: Two examples of smooth curve tracking ability of the orientation sensitive metric. Left column shows the minimum cost spanning tree under the Euclidean metric; right column shows the metric $C_{curve,\alpha=.9}$ combining distance and change in orientation.

Figure 8 shows how use of orientation information can assist in tracking smooth intersecting curves: $C_{curve,\alpha=.9}$ follows the curves (Figure 8 b, d) better than the pure proximity-based metric (Figure 8 a, c).

Figure 9a shows the result of applying $C_{curve,\alpha=.9}$ to detected building locations in our dataset. Much of the linear structure of the house groups is captured, but some of the branches selected do not lie parallel to the nearby road but instead cross perpendicular to it. This occurs when houses on opposite sides of the same street are close enough to form a link. These stubs can be avoided by looking ahead for a smooth extrapolation: if we are trying to extend from vertex v in the current tree to a new vertex w, the extension is allowed only if there is an additional vertex x with the angle between (v,w) and (w,x) close to 180 degrees. This yields a new cost metric:

$$C_{lookahead,\alpha,\varepsilon}(T,v,w) = \begin{cases} C_{curve,\alpha}(T,v,w) & \textit{if there exists a vertex } x\textit{, with the angle between } (v,w) \textit{ and } (w,x) \textit{ differing from 180 degrees by at most } \varepsilon; \\ \infty & \textit{otherwise.} \end{cases}$$

This metric prevents the growth of branches that form stubs, but keeps corners and crossings intact as shown in Figure 9.

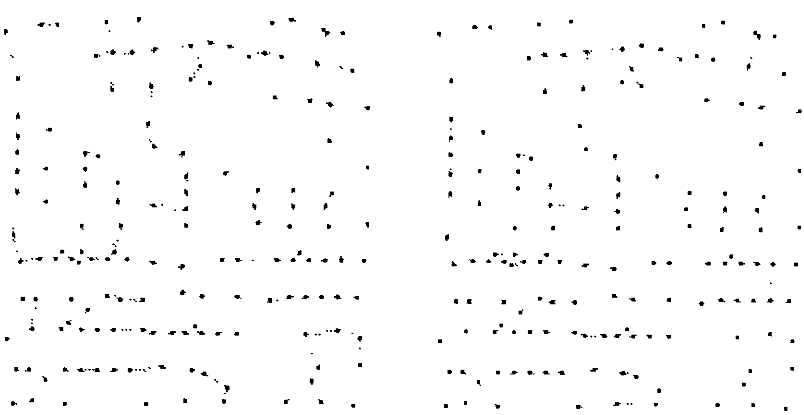


Figure 9a. $C_{curve,\alpha=.9}$ tracks curves well, but tends to generate short cross links between segments.

Figure 9b. $C_{lookahead,\alpha=.9}$ inhibits cross links by requiring that there be at least 2 adjacent edges lying nearly along a straight line.

Finally, we calculate the two most frequently occurring house link orientations and delete any links whose orientation differs significantly. This is reasonable since communities generally have roads that run in two primary directions. Then a simple line fitting program (Pavlidis 1988) is used to straighten the house clusters. The house clustering results together with the results of the road extraction algorithm are shown in Figure 1.

Future work on clustering

The next step would be to integrate the roadfinding with building grouping. Our building groups occasionally cross roads, and this could be inhibited. Where houses are relatively sparse the roadfinder alone tends to work well. With dense houses, the similarity of house and road imaging tends to complicate the roadfinding, but the houses group well, providing additional semantic clues. Grouping algorithms tend to be more tolerant to noise in the case of high density. The algorithms should synergize well, as in most places they agree, but there are places in the image where one algorithm has strongly located a road (or road segment) in which the other algorithm has difficulty or misses completely.

SUMMARY

The residual edge-finder tends to produce strong contours around small objects. For this data set, these edges can be used to significantly improve segmentation of low resolution road networks obtained from thresholding. A variant of Gaussian curvature is effective in locating buildings. We have developed a spanning tree technique sensitive to angles between branches, and shown it to be effective in detecting smoothly varying trajectories through given points.

ACKNOWLDEGEMENTS

We are grateful to Dr. Herb Tesser for suggesting this problem, and to Theo Pavlidis for many useful discussions and suggestions.

REFERENCES

Aviad, A. and Carnine, P. D. Jr (June 1988): "Road Finding for Road-Network Extraction," *Proceedings: IEEE CVPR*, pp. 814-819.

Binford, T. O. (1982): "Survey of Model-Based Image Analysis Systems," *Int. J. Robotics Res.*, vol. 1, no. 1.

Fua, P. and Hanson, A. J. (December 1985): "Locating Cultural Regions in Aerial Imagery Using Geometric Clues," *Proceedings: Image Understanding Workshop*, pp. 271-278.

Fua, P. and Hanson, A. J. (1987): "Using Generic Geometric Models for Intelligent Shape Extraction," *Proceedings: Image Understanding Workshop*, pp. 227-233.

Horn, B. K. P. (1986): *Robot Vision*, MIT Press.

Huertas, A., Cole, W., and Nevatia, R. (July 1987): "Detecting Runways in Aerial Images," *Proceedings: AAAI-87*, pp. 712-717.

Lee, David, Pavlidis, T., and Huang, K. (1988): "Edge detection through Residual Analysis," *Proceedings: IEEE CVPR*, pp. 215-222.

Matsuyama, T. (1987): "Knowledge-Based Aerial Image Understanding Systems and Expert Systems for Image Processing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 25, pp. 305-316.

McKeown, D., Harvey, W. A., and McDermott, J. (1985): "Rule-Based interpretation of Aerial Imagery," *IEEE Trans. PAMI*, vol. PAMI-7, pp. 570-585.

McKeown, David M. and Denlinger, Jerry L. (June 1988): "Cooperative Methods for Road Tracking in Aerial Imagery," *Proceedings: IEEE CVPR*, pp. 662-672.

Nevatia, R. and Babu, R. (1980): "Linear Feature Extraction and Description," *CVGIP*, vol. 13, pp. 257-269.

Nicolin, B. and Gabler, R. (1987): "A Knowledge-Based System for the Analysis of Aerial Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 25, pp. 317-329.

Pavlidis, Theo (1982): "An Asynchronous Thinning Algorithm," *CGIP*, vol. 20, pp. 133-157.

Pavlidis, Theo (1988b): Personal communication.

Pavlidis, Theo and Liow, Yuh-Tay (June 1988): "Integrating Region Growing and Edge Detection," *Proceedings: IEEE CVPR*, pp. 208-214.

Prim, R. C. (1957): "Shortest Connecting Networks and some Generalizations," *BSTJ*, vol. 36, pp. 1389-1401.

Quam, Lynn H. (May 1978): "Road Tracking and Anomaly Detection in Aerial Imagery," *Proceedings: Image Understanding Workshop*, pp. 87-100.

Spivak, M. (1970): *Differential Geometry*, vol. 2, p. 95.

Stevens, K. A. (1978): "Computation of Locally Parallel Structure.," *Biol. Cybernetics*, vol. 29, pp. 19-28.

Tuceryan, Mihran and Ahuja, Narendra (1987): "Extracting Perceptual Structure in Dot Patterns: An Integrated Approach," *University of Illonois at Urbana-Champaign Technical Report*, vol. UILU-ENG-87-2206.

Vistnes, Richard (1987): "Detecting Dotted Lines and Curves in Random-Dot Patterns," *Image Understanding*, pp. 849-861.

Zahn, Charles T. (1971): "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Trans. Computers*, vol. C-20, pp. 68-86.

Zucker, Steven W. (1985): "Early Orientation Selection: Tangent Fields and the Dimensionality of Their Support," *McGill University Technical Report*, vol. TR-85-13-R.