EXTRACTING ROAD NETWORKS FROM LOW RESOLUTION AERIAL IMAGERY

David Izraelevitz
Mark Carlotto

The Analytic Sciences Corporation
55 Walkers Brook Drive
Reading, Massachusetts 01867

ABSTRACT

We present a new algorithm for the enhancement and detection of linear features such as roads in satellite imagery. A cost inversely related to the response of a local linear feature operator is associated with each pixel, and an analysis based on a dynamic programming procedure is performed to determine if each pixel under study is part of a "low cost" path of predominantly linear pixels. The technique has been applied to several SPOT and Landsat images and shown to reduce the false alarm rate (percentage of pixels incorrectly classified as roads) by 50% over techniques which rely on local information alone.

1.INTRODUCTION

Local linear feature operators assign to an individual pixel a figure of merit for the hypothesis that the pixel is part of a linear feature. In high contrast environments, road segments can be extracted by thresholding the output of a linear feature operator and linking pixels that are above a given threshold[1]. Such an approach is insufficient in situations where roads are not adequately imaged, e.g., as in attempting to extract roads from Landsat and SPOT imagery. Changes in road surface, obscuration of the road by other objects such as trees, and low signal to noise ratio, tend to result in linear segments that are highly disconnected.

One alternative is to use local operators which use a larger region of support than the typical 3 by 3 or 5 by 5 pixels used by traditional operators. Unfortunately, as the region of support increases, the hypothesis that the feature under study is strictly linear becomes suspect. Another strategy is to use relaxation procedures to iteratively reinforce and link linear segments in close proximity to one another while de-emphasizing isolated segments. While such an approach may be appropriate for small images, the computational cost involved in repeatedly sweeping large image arrays can be prohibitive especially if the rate of convergence is low.

Where the above methods start with local information and work "bottom-up", other methods which take a more global view of the problem have been developed. One method, developed by Fischler[2] is based on finding the minimum cost path between two points. Defining a cost for a path as a function of its member pixels has been used by several authors[2,3,4,5]. If the costs are low along a given path with respect to adjacent pixels, then a path which has minimum cost is likely to retain this optimality even if some pixels are replaced by high cost pixels, e.g., in the case of noise or a road obstruction.

We became very interested in the approach of looking at cumulative costs along paths because it is robust in the presence of small gaps in the road and thus is a promising avenue for road extraction in noisy environments. However a key assumption in the above method - that a road connects the two points, while appropriate for road tracking where an operator can manually specify the beginning and end points of a road, can lead to the introduction of spurious road segments or false alarms when it is applied in an unsupervised fashion to imagery.

The method described in this paper combines elements of the above methods. A local linear feature detector that has a low miss rate but a high false alarm rate is used to extract candidate pixels. Then a search for paths inside windows centered on such pixels is performed to identify connected set of pixels which have a high likelihood of being on a road. The cost associated with a given path ending at a pixel under study is defined as the sum of the costs of the pixels which are members of this path. Pixels from which emanate low cost paths are then classified as road pixels. Thus, a set of pixels which lies on a road but has not been detected by the local linear operator is retained if it lies on a path of otherwise strongly linear pixels.

Preprint from:
Visual Communications and Image Processing '88
9-11 November, 1988, Cambridge, MA

2. A MODEL FOR ROADS

The model we assume for a road is an 8-connected path where the pixels along the path have intensities which are significantly different than those of pixels not on this path. In actual imagery this model is rarely satisfied since obscuration, variable road width, limited resolution and signal to noise ratio may lead some road pixels to have lower contrast than their neighbors. A low cost is assigned to a pixel if it is "road-like" from a local analysis of its immediate neighbors. Specifically, the cost is equal to an offset minus the absolute value of the difference between the local 3 by 3 average and the pixel intensity. Other road operators such as the Duda operator described in Fischler[2] have also been considered and found to yield comparable results.

Individual pixels with low cost may not be in the vicinity of other low cost pixels. We would like to de-emphasize such pixels since they are not likely to part of roads. On the other hand, a high cost pixel which is on a path composed primarily of low cost pixels should be retained. The above difficulty can be overcome by assigning a high value to a pixel if it is part of a path consisting of "road-like" pixels. Critical to this approach is the rapid study of all "smooth" paths (of a form described in the next section) leading to a given pixel. Specifically, we are interested in paths which cross the boundary of a window around a central pixel and end at this central pixel as shown in Fig. 1. The figure of merit used for deciding whether a pixel is part of a road is equal to the sum of the costs of each path ending at the pixel under study. If the minimum cost path has a low enough value, then the pixel can be labeled as a road pixel, even if it has high cost.

Although in this paper we discuss the extraction of bright roads on a dark background, the algorithm can also identify dark roads on bright backgrounds; unfortunately, it cannot do so at the same time. In order to be able to identify both kinds of roads, two separate applications of the algorithm are needed.

3. ALGORITHMS FOR FINDING MINIMUM COST PATHS

As pointed out by several authors[1,2], it is not necessary to explicitly calculate the cost of all paths if one is just interested in paths with minimum cost. An iterative technique, called the F* algorithm, successively finds lower cost paths between two points or between a set of points and a central point, eventually finding the minimum cost path or paths. The original application was for road tracking where an human operator specifies the road endpoints and the system then finds the path linking the two endpoints which is most "road-like."

Let the cost of each pixel in an image by given by c_{ij} where the pixel under study is located at $i = j = 0$. We consider all paths which begin on the boundary of an 2N+1 by 2N+1 window about the pixel under study. The objective is to find the minimum cost path with one endpoint on the window boundary and the other endpoint on the central pixel. To disallow paths which wander outside this window, let $c_{ij} = \infty$ for i or j outside this window. A second 2N+1 by 2N+1 array is defined, t_{ij}, which is initialized as $t_{00} = c_{00}$ and $t_{ij} = \infty$ elsewhere. After each iteration, t_{ij} has the value associated with the least cost path found so far which joins (i,j) and (0,0). After the algorithm ends, t_{ij} will contain the cost of the minimum cost path joining the pixel (i,j) and (0,0). Looking at (i,j) along the boundary of the window of interest and minimizing t_{ij} for this set gives us the value of the minimum cost path joining the window boundary and the central pixel.

The algorithm consists of two passes per iteration, one which processes image lines within a window from top to bottom, and a second pass which analogously processes image lines from bottom to top. We describe the first pass below, applied to a single window, i.e., a single pixel of interest in the image.

F algorithm:*

For each image line i in window:

For each pixel j in window:

$v = min(t_{i-1\ j-1}, t_{i-1\ j}, t_{i-1\ j+1}, t_{i\ j-1});\ t_{ij} = min(c_{ij}, v + c_{ij})$

For each pixel j:

$t_{ij} = min(t_{ij}, t_{i\ j+1} + c_{ij})$

The algorithm is repeated until the values of t_{ij} along the boundary do not change between iterations. In general it is difficult to specify the number of iterations required for convergence, since it depends on the complexity of the final minimum cost paths and how often they change curvature. At the very least a single full iteration is required, and for most situations at least two iterations are used. We note that unlike relaxation procedures, this iterative procedure is strictly local and applies only to the window under study; thus it does not require many passes over the image.

The F* algorithm is completely general since it considers all 8-connected paths and can be applied to arbitrary neighborhood shapes. We have applied this algorithm to the evaluation of minimum cost paths with success. However, we found that the generality provided by the F* algorithm was unnecessary in our application. By making some appropriate assumptions about the paths likely to be encountered, a new algorithm for calculating the t_{ij} values has been developed that is computationally more efficient than the F* algorithm.

The development of a new algorithm was motivated by two observations. Since the calculation of a minimum cost path is now performed at each pixel rather than over a region, the development of a computationally efficient algorithm for calculating these paths becomes a more critical concern. A second observation is that the minimum cost path calculation in our work needs to be performed over a much smaller window than that required by the F* algorithm. Within the window sizes considered, the road direction is not expected to change very much. In other words, the undulating paths tracked by the F* algorithm (see Fig. 2a) are not likely to occur in our application. Moreover, certain paths which change directions several times as the pixel under study is approached (Fig. 2b) may require many iterations for the F* algorithm to converge. However, such relatively straight paths may represent a road segment quite frequently and it is important to be able to efficiently identify them.

In our work we have embedded the knowledge of relatively straight roads as the assumption that on a local basis, roads are "monotonically approaching". Consider the path depicted in Fig. 3 which ends at **P**, and contains two points, **A** and **B**. We define the path as monotonically approaching if the distance between **A** and **P** is less than the distance between **B** and **P**, where the distance between two points is given by the minimum number of links in a path joining two points. By this definition, the path in Fig. 2a is not allowable since it does not satisfy our restriction, however, the path in Fig. 2b is allowable.

It is easy to show that for 8-connected paths, monotonically approaching paths are also shortest paths. Suppose that there is a monotonically approaching path S which is not a minimum distance path from each point on the path to the endpoint **P**. Let **A** be the last point on S for which S is a minimum distance path connecting **A** and **P** and let their distance be d. Such a point **A** always exists since it is at least the member of S which neighbors **P**. Let **B** be the next point on S past **A**. Since **B** is connected to **A**, the distance to **P** from **B** can be at most $d+1$. However, since S is monotonically approaching, the distance to **P** from **B** must be at least $d+1$. Thus the distance from **B** to **P** is $d+1$. A path of this length is given by **B** to **A** to **P**; thus S is a minimum distance path from **B** to **P** also.

From the argument above, we see that the paths which we will consider as the minimum cost paths joining the window boundary and the central pixel must also be minimum distance. For a 2N+1 by 2N+1 window, this implies that the length of all paths considered is N. The algorithm proceeds in the typical manner of dynamic programming, finding partial solutions on the way to the full solution. The first step of the algorithm considers the problem of finding all minimum cost paths of length one. Using this information, the minimum cost paths of length two are then found in an incremental manner. After N steps, the minimum cost paths of length N are found. The present algorithm is described formally below:

For $n=1$ to N
    For all pixels (i,j) at a distance n from the central pixel
        Find minimum value, v, of all adjacent minimum cost paths of length $n-1$.
        Set cost for path joining (i,j) to central pixel to be $c_{ij} + v$.

While a single pass of the F* algorithm requires 4 adds per pixel, the above algorithm can be performed with a single add per pixel. Since the F* algorithm usually requires several passes for convergence, we can expect the our algorithm to require at least 4 to 8 times fewer additions than the F* algorithm.

The algorithm is best illustrated by an example. Fig. 4 which shows a 7 by 7 window whose values denote the cost function calculated at each pixel in this window by a local operator. We first consider the 3 by 3 neighborhood surrounding the central pixel (Fig. 5a). The minimum cost paths joining the boundary of this neighborhood and the central pixel are obviously the paths given in Fig. 5b. The costs of these paths are shown in Fig. 5c. Thus Fig. 5c contains the costs of all paths of length 1 ending at the central pixel. The costs for the paths of length 2 can now be determined using the previously calculated values for paths of length 1. As an example, consider Fig. 6a. Here we want to calculate the value of the minimum cost path joining **A** and **P**. The best path is found by considering the costs of all paths of length 1 adjacent to A (Fig. 6b), and picking the one with the minimum cost. The total cost for the path beginning at A is given by the value of the minimum cost path of length 1 adjacent to A added to the cost of pixel **A**, yielding a total cost of 16 (Fig. 6c). Similarly, all minimum cost paths of length 2 can be calculated. Using this procedure, the minimum cost paths of length 3 are finally found (Fig. 6d). In each step of the algorithm, we calculate the minimum cost paths of length N using the cost of each endpoint and the previously calculated costs for the paths of length N-1. The final step is to find the minimum cost path of length 3, which is highlighted in Fig. 6d. There are several properties to note of the optimal paths found. As expected, they are all monotonically approaching. Also, any subpath of an optimal path ending at the central pixel is also optimal. Diagonals are all minimum cost paths joining pixels on the diagonal and the central pixel. This is expected since any other path joining this pixel and the central pixel would have extra links. This implies that the paths do not cross diagonals and suggests that all paths between diagonals (Fig. 7) can be calculated separately, allowing for parallel computation.

4. APPLICATION OF NEW ALGORITHM TO ROAD DETECTION

The existence of an efficient way of calculating the cost associated with all smooths paths ending at the pixel under study allows us to pursue the strategy described in Section 2. For each pixel in the image, the minimum cost path calculation is performed and the path of a specified length with minimum cost is found. If the total cost is low enough, the pixel under study is re-labeled as a road pixel.

Two imagery examples are presented to illustrate the performance of the road extraction algorithm. Fig. 8a shows a Landsat Thematic Mapper (TM) image (band 1) over Leavenworth, Kansas. In Fig. 8b, we show the result of applying the initial cost function to this image and thresholding at the 90 percentile level, i.e., the top 10 percent pixels are displayed. There are many isolated pixels classified as roads, while several roads are highly disconnected. We applied our algorithm to calculate the minimum cost path in a 9 by 9 window ending at each pixel in the image. The output, thresholded at the same percentile as before, is displayed in Fig. 8c. There are far fewer isolated pixels while several road segments have been filled in.

Our second example, shown in Fig. 9a, is a panchromatic SPOT scene of a rural area in upstate New York. Since the expected number of roads is much smaller than in our previous example, we used a more stringent threshold. Fig. 9b shows the result of thresholding the original cost function at the 98 percentile level. Applying our algorithm with the same window size as in the previous example and thresholding at the 98 percentile level yields the result shown in Fig. 9c. For this scene, an operator traced the true road locations and we calculated the number of pixels in Figs. 9b and 9c which were more than a single pixel away from a true road, i.e., the number of false road pixels. Our algorithm reduced the number of false road pixels by about 50 percent.

5. SUMMARY

We have approached the problem of finding roads in low resolution aerial imagery by attempting to improve the performance of local linear operators. A cost is initially associated with each pixel which penalizes pixels which have low contrast compared to its neighborhood. We have used in this study a very simple measure, the discrete Laplacian. After this step, we calculate the minimum cumulative cost path ending at each pixel using a new efficient algorithm. The cumulative distance becomes a better discriminator of road vs. not road pixels than the original cost function. The same approach can be used to improve the output of other feature operators such as edge detectors. We have applied this algorithm to several Landsat and SPOT images with very encouraging results. Finally, we are in the process of taking advantage of the parallel processing aspects of our new algorithm by implementing it on a coarse-grain parallel processor (Alliant FX-8).

1001 129

6. REFERENCES

1. M. A. Fischler, J. M. Tenenbaum, and H. C. Wolf, "Detection of Roads and Linear Structures in Low-Resolution Aerial Imagery Using a Multisource Knowledge Integration Technique", *Computer Graphics and Image Processing* , vol. 15, pp. 201-223, 1981.

2. A. Rosenfeld and J. L. Pfaltz, "Sequential Operations in Digital Picture Processing",*Journal of the ACM*, vol. 13, pp.471-494, 1966.

3. U. Montanari, "On the Optimal Detection of Curves in Noisy Pictures", *Communications of the ACM*, vol. 14, pp. 335-345, 1971.

4. A. Martelli, "An Application of Heuristic Search Methods to Edge and Contour Detection", *Communications of the ACM*, vol. 19, pp.73-83, 1976.

5. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd ed., Academic Press, Orlando, 1982, pp.135-138.

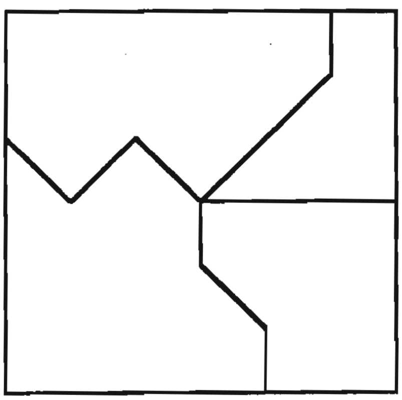


Fig. 1 Typical paths leading to a pixel under study

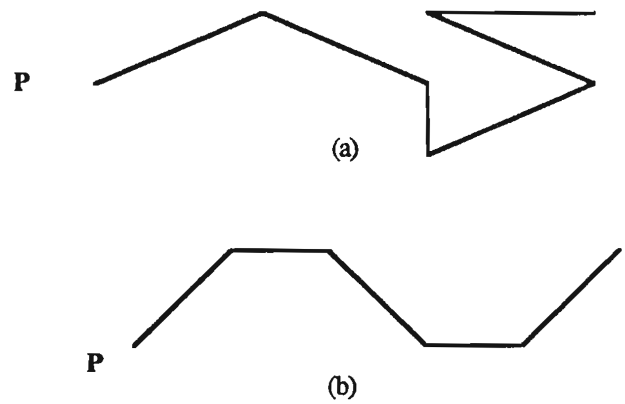


Fig. 2 (a) Severely undulating path and (b) smooth path

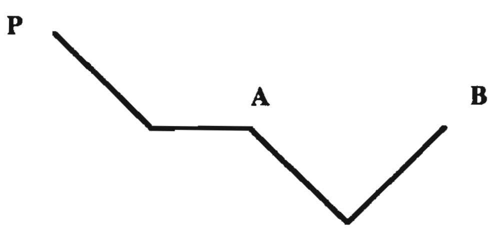


Fig. 3 Points along a minimum distance path

7	6	7	6	7	7	6
5	7	6	7	7	1	7
7	7	7	5	7	6	7
6	4	6	P↘ 6	7	3	7
7	6	7	6	5	7	7
7	5	7	5	1	5	7
7	7	6	7	3	7	7

Fig. 4 Cost function for example in text

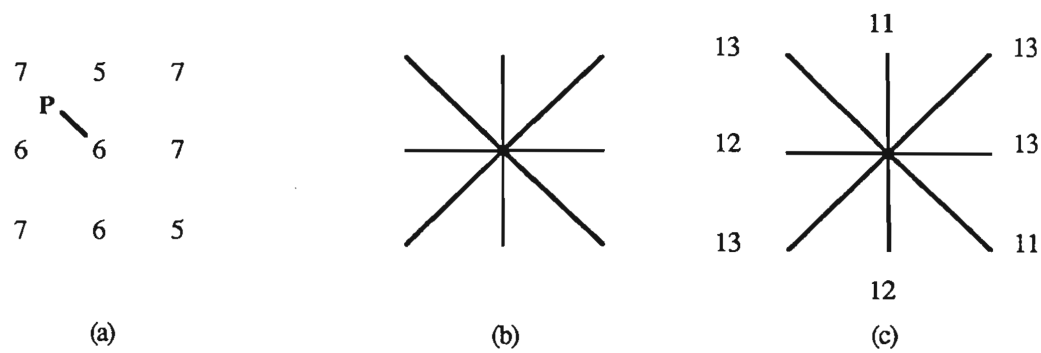
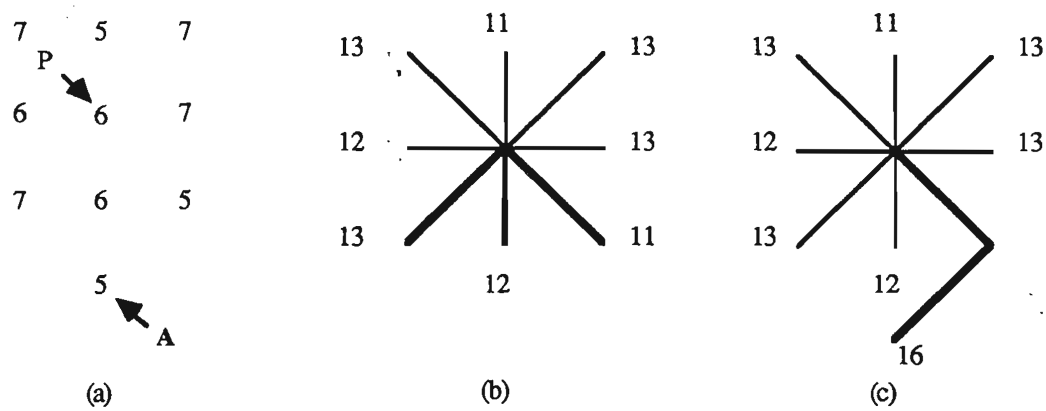
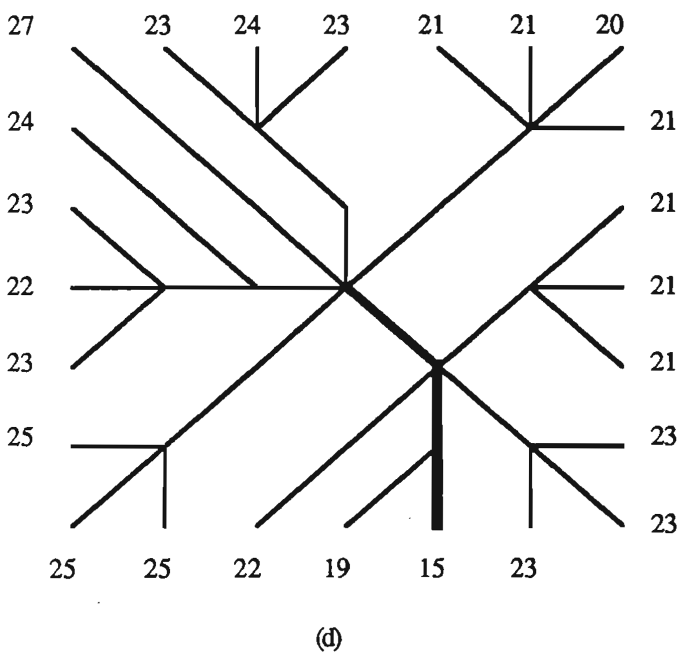


Fig. 5 Calculation of minimum cost paths of length one





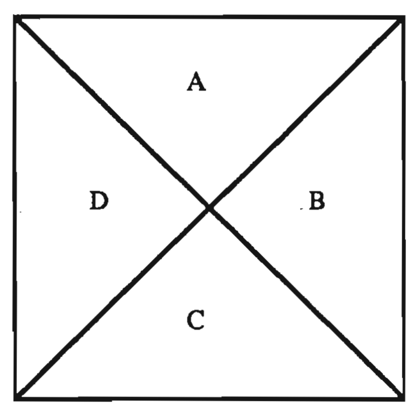


Fig. 7 Four independent calculation regions

Fig. 6 Calculation of minimum cost paths inside window

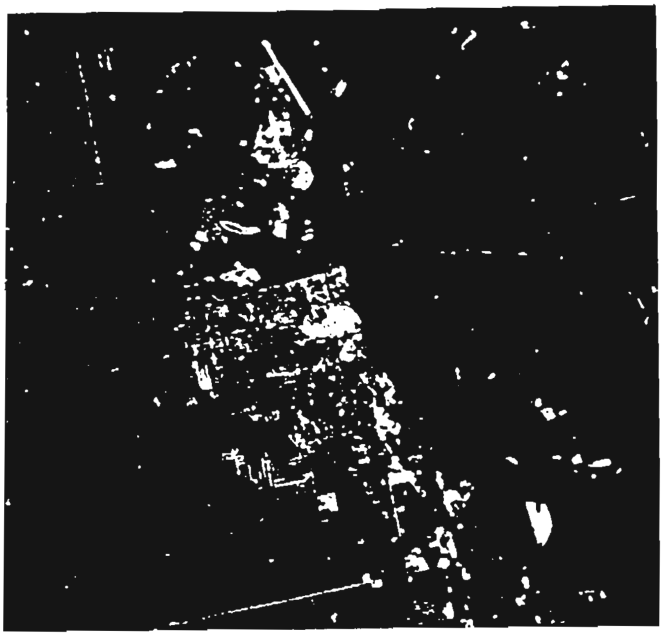


Fig. 8 (a) Landsat Thematic Mapper band 1 image of Leavenworth, Kansas

Fig. 8 (b) Initial road operator thresholded at 90%

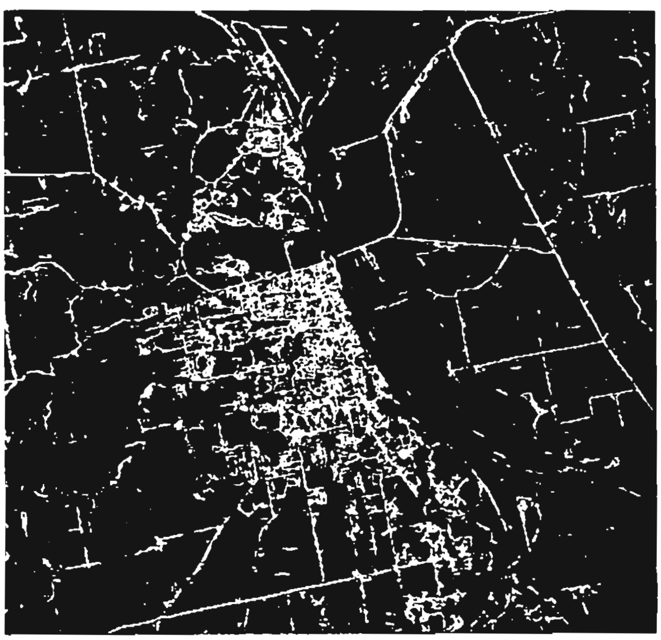


Fig. 8 (c) Output of new algorithm thresholded at 90%

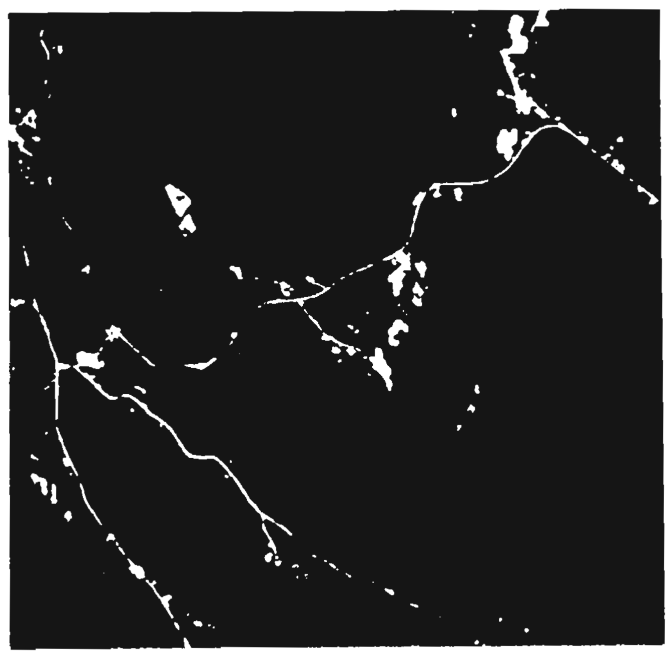


Fig. 9 (a) SPOT panchromatic scene of upstate NY

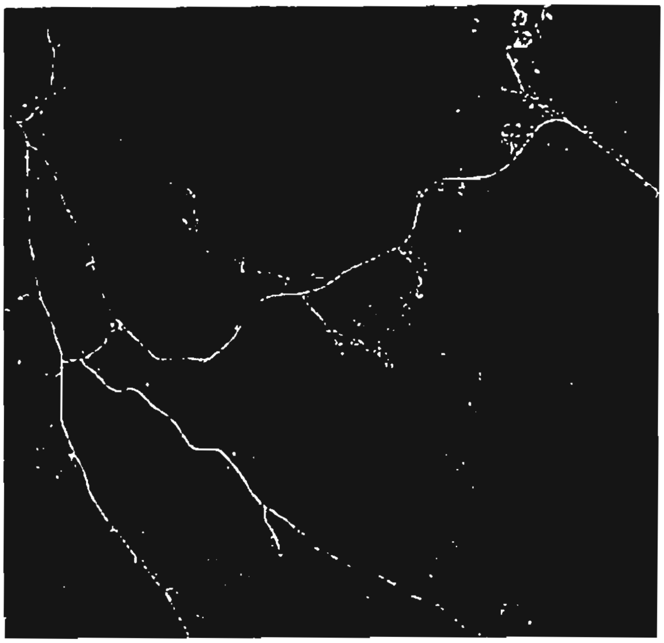


Fig. 9 (b) Initial road operator thresholded at 98%

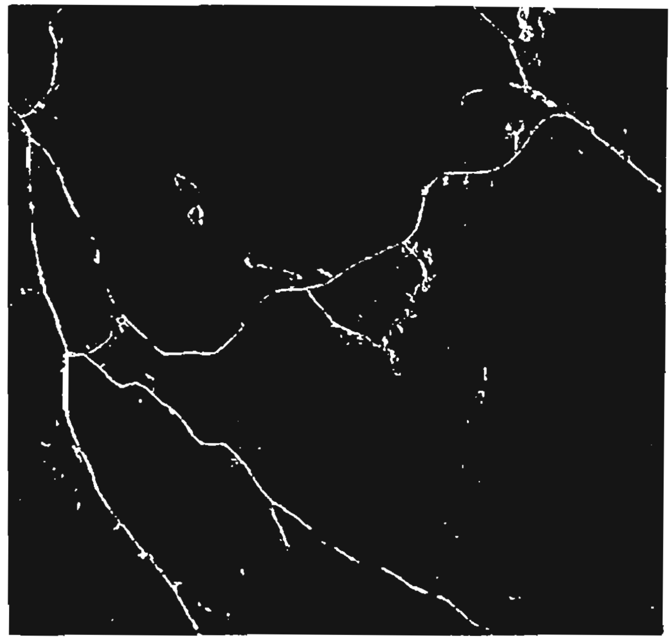


Fig. 9 (c) Output of new algorithm thresholded at 98%