NEW PROSPECTS IN LINE DETECTION FOR REMOTE SENSING IMAGES

N. Merlet[1] *J. Zerubia*[2]

[1]Institute of Computer Science,The Hebrew University of Jerusalem, 91904 Jerusalem, Israel, E-mail : nicolas@cs.huji.ac.il*
[2]INRIA, BP 93, 2004 Route des Lucioles,06902 Sophia Antipolis Cedex, France, E-mail : zerubia@sophia.inria.fr

ABSTRACT

Dynamic Programming is one of the main methods of line detection. It defines a cost, which depends on local information, and performs a summation-minimization process in a graph or in the image.
In particular, Fischler et al presented an algorithm, called F*, which achieves important results (convergence, robustness, rapidity). In previous works, we proposed a mathematical formalization of the F*, which allowed us to extend the cost to cliques of more than two points to take into account the contrast, and to include curvature information in the cost by using neighborhoods of size larger than one.
In the present paper, we propose a method for computing this cost automatically for a wide range of images, from the probability distribution in the neighborhood of sample segments. We apply the resulting potentials on SPOT images.

1. INTRODUCTION

Among the various algorithms of line detection, those based on dynamic programming have raised great interest for their rapidity, robustness to noise and precision ([1],[2],[3], [4]). The basic approach consists of defining a cost or potential depending on the local characteristics (contrast, grey-level) of the image and of performing an iterative summation-minimization process in a graph or in the image.
We have shown in previous works ([5], [6], [7]) how one of the main algorithms of dynamic programming, the F* of Fischler et al [2], could be generalized to cliques of more than two points and to neighborhoods of size larger than one, in order to take into account grey level, contrast, and curvature in one *unique* cost function.
One of the main drawbacks of this method is the difficulty to define the cost without using "generate and test" strategies. In the present paper, we propose a systematic method for computing the cost for images having different configurations (grey level, contrast) with no a priori knowledge on these configurations. We will first recall the theoretical definitions and the basic algorithms of segmentation. Then, we will develop the different steps for computing the potential. Finally, we will apply this potential to different images.

*The authors would like to thank M. BERTHOD, G. GIRAUDON, S. HOUZELLE and S. MATHIEU for many interesting discussions. This work was partially supported by an AFIRST grant (French-Israeli Scientific Cooperation).

2. THEORETICAL PRESENTATION

Let us first recall that a clique is a set of points which are all neighbors to each other, in a given neighborhood system.
In the following, we will denote by $\mathcal{C}_n$ the set of the cliques of order n and of the cliques of order one, $P(x,y)$ the set of paths from x to y, f a grey-level image, and f_c its restriction to c.
We have defined in previous papers ([5],[6]) a distance such that a line is the shortest path between its extremities for this distance. We proceed in two steps : we first define a cost associated to a path between two points, and then the distance as the minimum cost for all the paths joining them.
In order to take into account curvature ([6],[7]), we have considered the following configuration :

- We consider *three* or more successive points of the path, and neighborhoods of size *two* or more.
- We consider not only one clique, but two cliques, one characterizing the lines, the other characterizing the background.
- The potential depends on the angle between the points of the clique characterizing the line.

The cost is naturally defined as the sum of the potentials along the path :

Definition 1 :
Given a potential $\phi : \mathcal{C}_n \mathrm{x} \mathcal{C}_m \mathrm{x} {\mathcal{R}^+}^{n+m} \to \mathcal{R}^+$, and given a path h, so that : $x_0 = x,...,x_i,..., x_p = y$ between two points x and y of S, the attraction cost along h is defined as the function μ : $\mu(h) = \min \sum_{i=0}^{p-n+1} \phi(c_i, c_j, f_{c_i}, f_{c_j})$ $c_i \in \mathcal{C}_n, c_j \in \mathcal{C}_m \| c_i \cup c_j \in \mathcal{C}_{n+m}$ $x_i \in c_i, ..., x_{i+n-1} \in c_i$

For example, we may consider nine points forming a square : three adjacent points among the nine points correspond to the road, the six others to the background. The three points define an angle, from which the potential depends. The definition of the energy (not a distance anymore) is straightforward :

Definition 2 :
The *attraction energy* U between two points x and y of S is defined as the minimum value of the attraction cost along any path between x and y : $U(x,y) = \min_{h \in P(x,y)} \mu(h)$.

3. ALGORITHMS

We call S_0 one set of feature extremities, and S_1 the other set, where the features are the shortest paths (wrt U) between S_0 and S_1. We assume that S_0 and S_1 are known. Two algorithms are used : the first one computes U relatively to S_0, the second one builds the paths, propagating from S_1 towards S_0.

In all our applications, we considered only three successive pixels, neighborhoods of size two, and maximal deviations of 45 degrees. Thus there were only 24 cliques to be considered : each point has eight direct neighbors, and only three second order neighbors could come afterwards without forming together an angle of less than +/- 135 degrees [6].

3.1. Computation of the distance function

We define two types of images :

- Im1 is a series of 8 output images corresponding to the 8 neighbors of a point. At the beginning, they are equal to zero for starting points (S_0), to $+\infty$ for other points. They store for each point the minimum energy which was assigned to it by the corresponding neighbor. (For each neighbor, three cliques may intervene corresponding to the three second order neighbors).
- Im2 is a series of 24 images, representing the potentials associated to the 24 types of cliques.

We then perform the following operations, for each point of the image M_0 and for each of its 8 direct neighbors N_0 :

- We consider each of the three cliques c_0, c_1, and c_2 which contain M_0 and N_0. Each clique corresponds to a direct neighbor P_0, P_1 and P_2 of N_0 (second order neighbor of M_0).
- We consider the three values of Im1 in N_0 for the three images corresponding to P_0, P_1 and P_2. We call them $Im1_{P_i}(N_0)$.
- We compute the three values : $U_{P_i}(M_0) = Im1_{P_i}(N_0) + Im2_{c_i}(M_0)$.
- We compute their minimum : $U_1(M_0) = \min_{i=1,2,3} U_{P_i}(M_0)$.
- If $U_1(M_0)$ is smaller than $Im1_{N_0}(M_0)$, then we update $Im1_{N_0}(M_0)$ with this new smaller value.
- We stop the algorithm when nothing is updated in a complete cycle.

Details about optimization and proof of convergence of the both algorithms may be found in [6] and [7].

3.2. Algorithm of pattern recognition

The algorithm of back-propagation is based on the directional information of Im1 (See 3.1) : we follow the direction shown by the successive cliques. We use two types of images :

- Im1 is the series of 8 output images (Im1) obtained in the previous algorithm, where the value at a point is the minimum value of the energy from paths coming from the corresponding 8 neighbors.
- Im2 is the binary image of the goal points (S_1), eventually labeled.

We then perform the following operations : for each point M_0 such that $Im2(M_0)$ is not null (M_0 is labeled),

- We look for the minimum value of the energy $Im1_{N_i}(M_0)$ in the eight images $Im1$.
- We mark in $Im2$ the point N_i corresponding to this minimum with the same label as M_0, if it is not already labeled.
- We stop the process when no labeling is performed in a complete cycle. The result is in Im2.

Let us now deal with the most difficult point of this method : the choice of the general potential.

4. CHOICE OF THE POTENTIAL

We assume that the only information available on the images is the set S of starting points (S_0) and of goal points (S_1). We want to compute a potential on this basis, which allows to get the features. We propose to exploit the information around these points, and to perform in three steps :

1. *From the windows to the partial segments* : we want to find segments of lines around the points of S_0 and S_1, where the information is the most relevant, since a line is there for sure. We use for this a general and approximative potential and apply the F* in the small windows (see 4.1.2).
2. *From the partial segments to the histograms* : we then compute the histograms of grey-level values and contrasts on the partial segments, and smooth the histograms for keeping relevant information only.
3. *From the histograms to the potential* : we then define the potential as a function of the histograms. We assume that the probability distribution in the small windows is a good approximation of the distribution in the whole image.

Let us look in more details at these three steps. In all this section, we consider indifferently points of S_0 or of S_1.

4.1. From the windows to the partial segments

4.1.1. Defining windows

We define the windows as the rectangles of the points which are at a distance not greater than 10% of the size of the image from S (in our images : 25 pixels) :

- there is a line in each of these windows, since they contain a point of S, and we get enough information
- the chance for getting a second line in the same window is low if we assume that the points of S were not chosen close to each other
- the computing time for finding the partial segments is neglectable (less than 1 %).

Moreover, we reduce the scale of grey-level values to 20% of the original scale, in order to get more data in each level of the histograms and more robustness.

4.1.2. *Defining partial potentials*

We then define partial potentials P_{pg} and P_{pc} depending respectively upon the grey-level and the contrast.

We define g_{max}, g_{min} and g_{moy} as the maximum, minimum, and average grey values over all the windows, we call g_m the grey level of any point M, g_s the grey level of any point of S, and consider three configurations :

- if the grey levels g_s of the points of S are always higher than g_{moy} (clear features) : $P_{pg}(M) = g_{max} - g_m$.
- if the grey-levels g_s are smaller than g_{moy} (dark features) : $P_{pg}(M) = g_m - g_{min}$.
- if the values g_s may be higher or smaller than g_{moy} : $P_{pg}(M) = abs(g_m - g_s)$ (g_s changes in each window).

We define P_{pc} as the opposite of the morphological gradient (which is the difference between maximum and minimum in a neighborhood of size one).

Finally, the potential used for finding the segments is $P_p = P_{pg} * P_{pc}^2$. We give more importance to the contrast than to the grey level information since the contrast does not depend upon the choice of S and is more robust.

4.1.3. *Applying the F**

We then compute the distance function with F* for the potential P_p. We look for the other extremity of the segment in each window and define it as the point on the opposite side of the window where the distance function is minimum. We then propagate from these extremities and get the complete partial segments. The results were exact on the 27 small windows that we checked.

4.2. From the partial segments to the histograms

The conditional probability $P(M \in segment | g_m = g)$ characterizes the probability for a pixel to belong to the segment given its grey level, and should be high along the path. Thus, the potential should decrease with it.

By definition :
$P(M \in segment | g_m = g) = \frac{P(M \in segment \wedge g_m = g)}{P(g_m = g)}$.

Thus, we may compute easily an approximation of this probability :
$P(M \in segment | g_m = g) \sim \frac{|M \in segment \wedge g_m = g|}{|g_m = g|}$.

The conditional probability is obtained by computing the histogram of the segment H_l and the histogram of the image H_i, and by dividing H_l by H_i. We obtain an histogram which is not smooth, due to the small number of points considered. We smooth it by

- filling the holes of the histogram
- joining the discontinuity points by linear approximation in order to get a continuous histogram.

We apply the same steps for getting the histogram H_c of the contrast, replacing the grey level values by the values of the morphological gradient.

Fig. 2 presents the histograms of grey-level, and Fig. 5 the histograms of contrast. In Fig. 2, h1 is the conditional probability for the image g2 (shown in Fig. 4), h2 is the result after filling, and h3 the smoothed result. h0 is the smoothed result for g1 (Fig. 1). In Fig. 5, h4 is the conditional probability for g1, h5 the result after filling, and h6 the smoothed result. h7 is the smoothed result for g2.

4.3. From the histograms to the potential

Two cases have to be considered (both for the grey level and for the contrast) :

1. values for which the probability is not null. We have found that defining the potential as the inverse of the probability gives better results than defining it as the opposite, since smaller probabilities must be penalized as less relevant.
2. values for which the probability is null. We define the potential for these values as a linear interpolation of the values of the first case : the potential increases faster if the histogram is narrow (robust information).

We apply this process both to the histograms of grey level and contrast, and obtain a potential of grey-level P_g and a potential of contrast P_c. The smoothness of the segmented features is a subjective criterion which depends on the considered application, and it should be chosen interactively.

Finally, we define the general potential as the sum of the potentials of grey level, contrast, and curvature, and apply the F*algorithm. Fig. 3 and Fig. 6 show the results of the segmentation for g1 and g2. We have applied this method with success on three images with geological features.

5. CONCLUSION

We have presented in this paper a general method for defining an acceptable potential to be used in the improved F*. It proved to be efficient on several images having different grey-level and contrast characteristics. Current research deals with testing the method on a wider range of images (with urban areas).

REFERENCES

[1] R. Bellman, R. Kalaba, Shortest Paths Through Networks, in *Dynamic Programming and Modern Control Theory*, pp. 50-54, Academic Press, New York and London, 1965.

[2] 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. *CVGIP* vol. 15, pp. 201-223, 1981.

[3] D. Geman, B. Jedynak, Detection of roads in satellite images, *IGARSS Conference*, Espoo, 3d-6th June, 1991.

[4] A. Martelli, An application of heuristic search methods to edge and contour detection. *Commun. Ass. Comput. Mach.*, vol. 19, pp. 73-83, Feb. 1976.

[5] N. Merlet, From music staves to brain fissure through a distance based energy function, *4th International Conference of Computer Analysis of Images and Patterns (CAIP)*, pp. 129-138, Dresden, September 17-19th, 1991.

[6] N. Merlet, J. Zerubia, Classical Mechanics and Road Detection in SPOT Images. *Joint Technical report*, INRIA/Hebrew University, April 1993.

[7] N. Merlet, J. Zerubia, A curvature-dependent energy function for detecting lines in satellite images, *8th SCIA Conference*, Tromso, 25th-28th May, 1993.

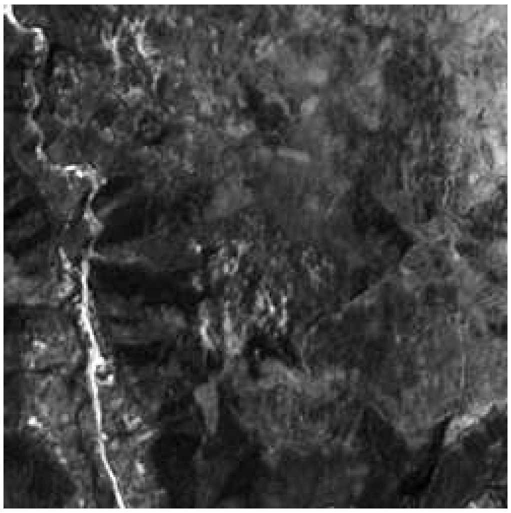

Figure 1. Original image g1

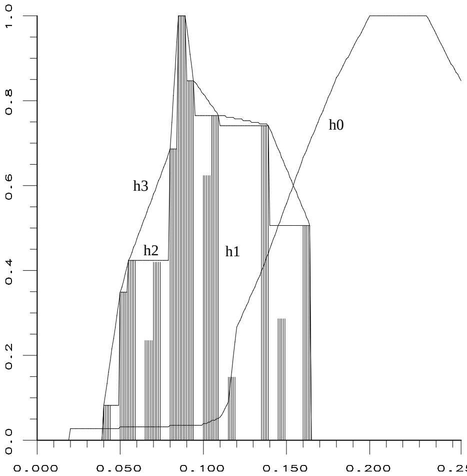

Figure 2. Histograms of grey level

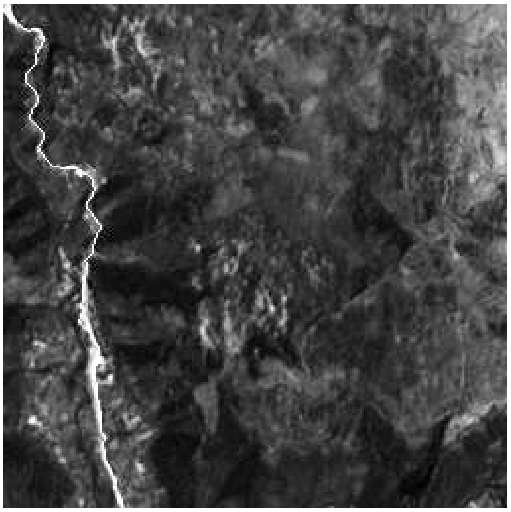

Figure 3. Result for g1

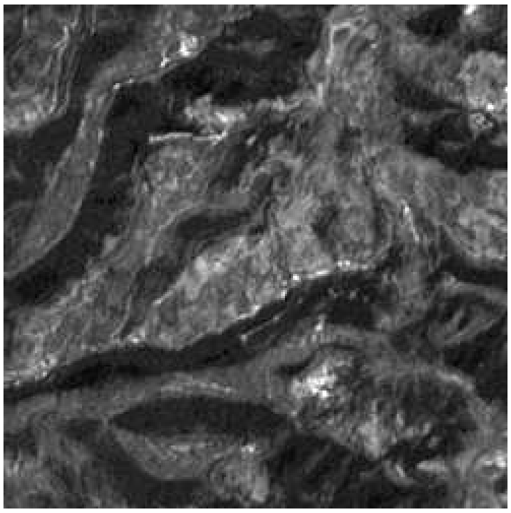

Figure 4. Original image g2

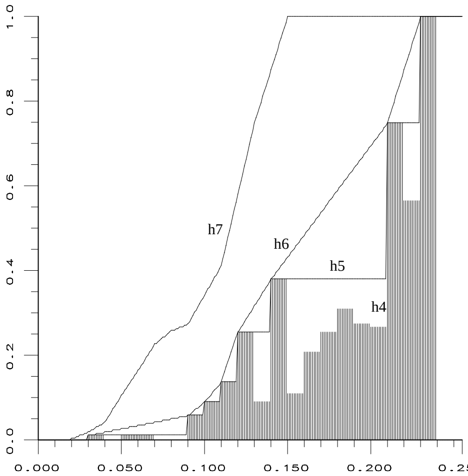

Figure 5. Histograms of contrast

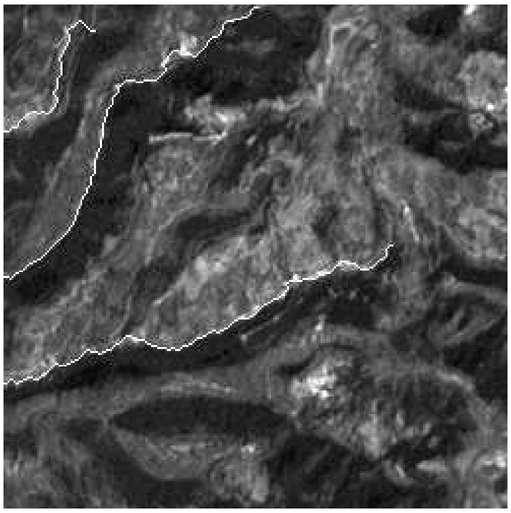

Figure 6. Result for g2

NEW PROSPECTS IN LINE DETECTION FOR REMOTE SENSING IMAGES

N. Merlet[1] *and J. Zerubia*[2]

[1]Institute of Computer Science,The Hebrew University of Jerusalem, 91904 Jerusalem, Israel, E-mail : nicolas@cs.huji.ac.il[1]

[2]INRIA, BP 93, 2004 Route des Lucioles,06902 Sophia Antipolis Cedex, France, E-mail : zerubia@sophia.inria.fr

Dynamic Programming is one of the main methods of line detection. It defines a cost, which depends on local information, and performs a summation-minimization process in a graph or in the image.

In particular, Fischler et al presented an algorithm, called F*, which achieves important results (convergence, robustness, rapidity). In previous works, we proposed a mathematical formalization of the F*, which allowed us to extend the cost to cliques of more than two points to take into account the contrast, and to include curvature information in the cost by using neighborhoods of size larger than one.

In the present paper, we propose a method for computing this cost automatically for a wide range of images, from the probability distribution in the neighborhood of sample segments. We apply the resulting potentials on SPOT images.