A Curvature-Dependent Energy Function for Detecting Lines in Satellite Images*

N. Merlet

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

J. Zerubia

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.
Fischler et al in particular 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, and to synthesize the needed information (contrast, grey-level, direction) in one unique cost function defined on the digital original image.*
In the present work, we show how to include curvature information in the cost by using neighborhoods of size larger than one, and we apply this energy to the recognition of roads and valleys in 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],[7],[4]). The basic approach consists in defining a cost or potential depending on the local characteristics (contrast, grey-level) of the image and to perform an iterative summation-minimization process in a graph or in the image. Although curvature information is useful and sometimes necessary for characterizing features, its global definition creates difficulties for introducing it in the local cost ([7],[4],[3],[6]).

*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).

We present herein a new solution to this problem.
In previous works [8], we proposed a mathematical formalization of the F* algorithm of Fischler et al [4], which allowed us to extend the definition of the cost to cliques of more than two points, as well as an optimization of the number of iterations, and to synthesize the needed information (contrast, grey-level, privileged direction) in one *unique* cost defined on the digital original image. We will first recall the main results of this formalization and show how to introduce curvature information with neighborhoods of size larger than one. We will then give the new algorithms obtained, and present the corresponding results with three satellite images (SPOT). The comparison with previous methods will lead us finally to the conclusion.

2 Theoretical presentation

2.1 Reminder of some basic notions

Let us first recall some basic notions which will be useful in the following [2].
We denote by S the lattice of the original image.

Definition 1 : A *neighborhood system* on S is a family $\mathcal{N} = \{\mathcal{N}_x \subset S | x \in S\}$ such that :

- $x \notin \mathcal{N}_x$,
- $\forall x_1 \in S, \forall x_2 \in S, x_1 \in \mathcal{N}_{x_2} \Longleftrightarrow x_2 \in \mathcal{N}_{x_1}$.

The elements of $\mathcal{N}_x$ are the *neighbors* of x.

Definition 2 : $c \subset S$ is a *clique* relatively to $\mathcal{N}$ iff :

- $|c| = 1$,
- or $|c| \geq 2$ and any two different elements of c are neighbors.

$|c|$ (number of elements of c) is called the *order* of the clique.

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,
- $\mathcal{F}(S, \mathcal{R}^+)$ the set of functions from S into $\mathcal{R}^+$,
- f an element of $\mathcal{F}(S, \mathcal{R}^+)$, typically a grey-level image, and f_c its restriction to c.

2.2 The distance-based energy function

We have defined in previous works [8] 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. We naturally think to define the cost as the sum of the potentials along the path.

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

We notice that in the case of cliques of order two, the minimization is not necessary since a unique clique is defined by two successive points of the path, and the cost is simply the sum of the potentials along the path : $\mu(h) = \sum_{i=0}^{p-1} \phi(c_i, f_{c_i})$, where c_i is the clique composed of x_i and x_{i+1}.
However, for higher orders, two points of the path do not necessarily define a unique clique : for cliques of order three for example, two points define two cliques (or four) on both sides of the path, and the minimization takes into account only the one with the lower potential.
The definition of the energy 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 :
$U(x, y) = \min_{h \in P(x,y)} \mu(h)$.

Under a very simple constraint on the potential, this function is also a distance.

Theorem 1 :
We suppose that ϕ satisfies the constraint :

$\forall c \in \mathcal{C}_n, \forall f \in \mathcal{F}(S, \mathcal{R}^+)$,
$\phi(c, f_c) = 0 \Longleftrightarrow |c| = 1$.

Then $\Delta(x, y) = U(x, y)$ is a distance function on S, called *directional numerical distance.*

The separability condition follows from the constraint, and the Triangular Inequality may be shown by combining two paths (see [8],[9]).

2.3 New definition of the energy

We want to keep the general structure of the method and to simply extend it such that it takes into account very local curvature. We can do it in the following way :

- We consider neighborhoods of size *two* or more (and not one as until now).
- We consider *three* or more successive points of the path (and not two).
- We consider not only one clique, but two cliques, one containing good candidates for road points, the second one good candidates for background points. This reminds us the hit-and-miss transform of Mathematical Morphology (Golay transform, [5]) which looks for configurations where one part of the structuring element belongs to the object, and the other part to the background. Contrary to Mathematical Morphology, we do not consider binary logic where the relevant part of the configuration is/is not inside the object, but we associate to the corresponding points of the image a local cost which characterizes the probability for them to fit well the configuration.
- The potential depends on the angle between the points of the clique characterizing the road. This last point is what allows us to take local curvature into account.

This leads us to substitute the following definition to *Definition 1* :

Definition 3 :
Given a potential $\phi : \mathcal{C}_n \mathrm{x} \mathcal{C}_m \mathrm{x} {\mathcal{R}^+}^{n+m} \rightarrow \mathcal{R}^+$, and given a path h, so that :
$x_0 = x, \ldots, x_i, \ldots, x_p = y$ between two points x and y of S, the attraction cost along h is defined as the function μ :
$\mu(h) = \min \sum_{i=0}^{p-n+1} \phi(c_i, c_j, f_{c_i}, f_{c_j})$
$c_i \in \mathcal{C}_n, c_j \in \mathcal{C}_m | c_i \cup c_j \in \mathcal{C}_{n+m}$
$x_i \in c_i, \ldots, x_{i+n-1} \in c_i$

For example, we may consider nine points forming a square : three adjacent points crossing the nine correspond to the road, the six others to the background. The three points define an angle, and the potential of the clique may depend on a partial cost of this angle ; a penalty may be added to the old potential depending on the angle : if the points are aligned, the penalty is null, if the angle to alignment is small, the penalty is small, and the penalty may get infinite for important deviations.
We illustrate in Fig. 1 below this choice. c_i, of order three, corresponds to the road/valley while c_j, of order six, characterizes the background. α is the angle of deviation to the straight line, and the potential should increase with it.

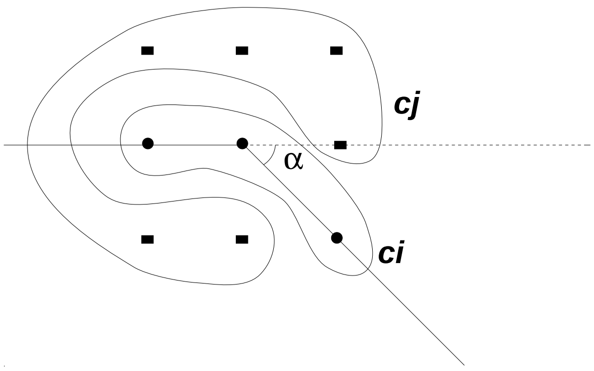

Figure 1: Couple of cliques for a neighborhood of size two

It can be noticed that with this new definition U is not a distance since the Triangular Inequality is not satisfied any longer.

3 Algorithms

Both algorithms ([8],[9]) are deeply modified due to this new constraint, since we must take into account neighborhoods of size two. However, we keep the central idea of dynamic programming to perform iteratively local minimization.

In all our applications, we considered only three successive pixels, 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.

3.1 Computation of the distance function

Now, in order to take into account the dependency between successive cliques on the path, we must remember for each point the minimum energy which was assigned to it by each of its (8) neighbors. We must therefore 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 (assumed known), 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. For practical reasons, we associate to each point M_0 the clique of order three formed of M_0, of one of its 8 neighbors N_0, and of one of the three neighbors of N_0 prolongating the two points.

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 corresponding 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.

In fact, we also perform here scannings in the four possible directions ([8],[9]), and for each scanning we consider only the neighbors which have been updated in the same scanning, in order to optimize the propagation. We obtain the result in Im1 in a coded form. The convergence of the algorithm is assured since Im1 decreases. We will see now how to exploit in a very easy way the information in $Im1$ for reconstructing the paths.

3.2 Algorithm of pattern recognition

The new algorithm of back-propagation is entirely different from the previous one for neighborhoods of size one, since it 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 (assumed to be known), 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.

The algorithm converges since we deal only with points which are not labeled and since we label them in an irreversible way. We may implement the algorithm by performing scannings in the four directions, or by using a tree structure. In both cases, we avoid bottlenecks with the diagonals and optimize the propagation. Rightness of these both algorithms is proved in [9].

4 Applications

In the following three applications, we will denote by S_0 the set of starting points (used in Algorithm 1) and by S_1 the set of goal points (used in Algorithm 2).

4.1 Geologic image

The original image is presented in Fig. 2 and the result obtained without using curvature information in Fig. 3. The valleys are the transition lines from clear regions (up left) to dark ones (down right) - the light of the sun comes from the right bottom corner (south east).

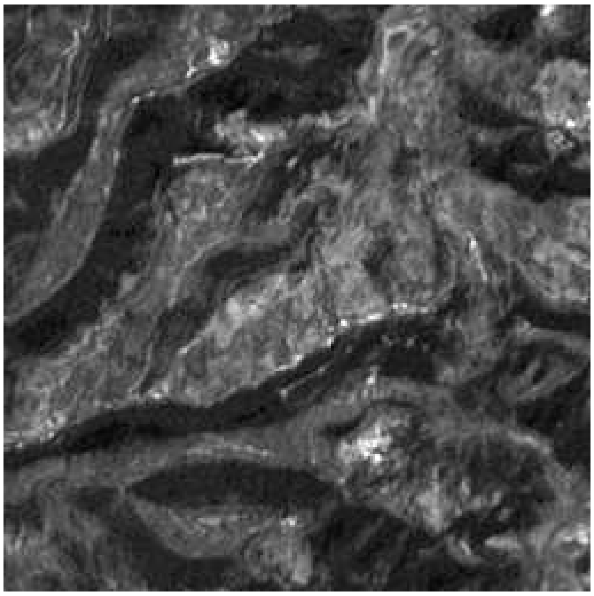

Figure 2: Original image (g)

For taking into account the three notions of grey-level, contrast and curvature, we define the potential as the sum of three terms h, g and k, which depend on two cliques c_i and c_j of order three and six (see Fig. 1). We call M_1, M_2 and M_3 the three points of c_i, ordered according to the increasing values of f : $f(M_1) \leq f(M_2) \leq f(M_3)$. ϕ is defined as :

$$\begin{aligned}\phi(c_i, c_j, f_{c_i}, f_{c_j}) = {} & h(f(M_1)) \\ & +g(f(M_1) - median(c_j)) \\ & +k(c_i).\end{aligned}$$

where $h(f(M_1))$ characterizes the worst of the three candidates for the valley, $f(M_1) - median(c_j)$ is the contrast between the valley and the background (more precisely the median value of the background), and $k(c_i)$ corresponds to the angle between the points of c_i. h and g are piecewise linear and decrease, while k increases with the angle (practically, k may take three values for reducing the computation). S_0 is defined as the left column, and S_1 is composed of the three right extreme points of the features. These points are defined manually, and current research involves automatizing their definition.

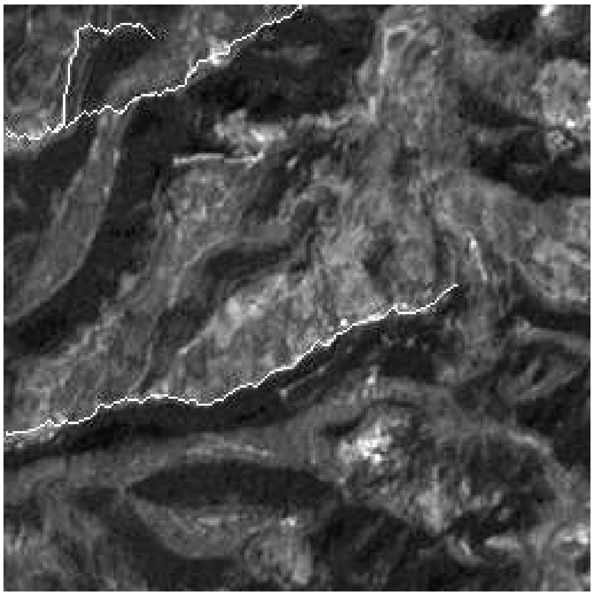

Figure 3: Segmentation of the valleys in (g) with neighbors of size one

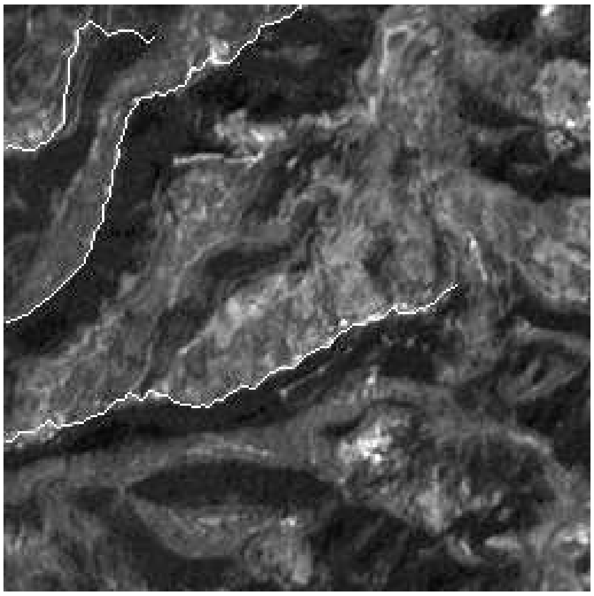

Figure 4: Segmentation of the valleys in (g) with neighbors of size two

The final result is presented in Fig. 4. The notion of curvature does not intervene here at the specific point where the propagation turns towards another feature, but on all the small curves of the deviation between both features. We notice this time the good fitting between segmentation and features.

4.2 Road image r1

When applying the same type of potential to a road image (Fig. 5), we obtain Fig. 6 as a final result. S_0 is the right bottom corner, S_1 is composed of the extreme points of the three features, and they are also defined manually. The result fits perfectly the features, although the human eye had difficulties to precisely localize the roads. The corresponding energy function U is presented in Fig. 7.
We have made other tests by varying the curvature coefficient k. In Fig. 8, it was taken four times smaller than in Fig. 6, and the path got lost in zigzags in highly textured regions. When using on the contrary a four times bigger coefficient, in Fig. 9, the real feature had too many zigzags, and the propagation jumped directly over the fields in straight lines.

Figure 5: Original image (r1)

4.3 Road image r2

We applied the same type of potential to a second road image (Fig. 10). This time, S_0 is defined as the center of the agglomeration and S_1 as the extremities

Figure 6: Segmentation of the roads in (r1) with neighbors of size two

Figure 8: (r1) with a weaker constraint

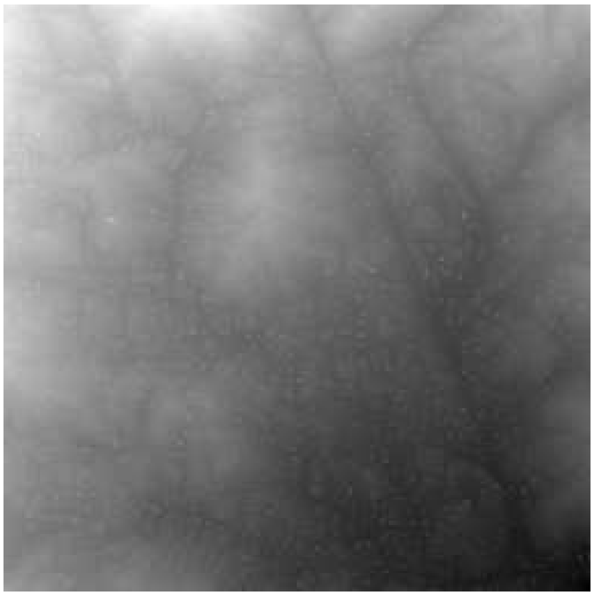

Figure 7: Energy function U for (r1) with neighbors of size two

Figure 9: (r1) with a stronger constraint

of the features around the image. Obviously, choosing other points for S_0 and S_1 would change the results.

Figure 10: Original image (r2)

Figure 11: Segmentation of the roads and channels in r2

We may see in Fig. 11 that the result fits well the features, except for the deviation induced in the right upper part by a highly-contrasted channel.

5 Discussion

Let us first stress that our method keeps the well-known advantages of dynamic programming : rapidity, robustness to noise, precision ([1],[7],[4]). The topic of this paper is not to present a general comparison of dynamic programming with other methods such as Hough Transforms, Snakes, or Canny Filter, and we refer the reader to relevant literature on this subject (same references).
Several ways of introducing the curvature in the local cost have been developed in [7], [3], [4] and [6]. We want now to compare them with our method, to detail the complexity, and then to analyze the main drawbacks of the proposed technique.

5.1 Method 1 : [7],[3]

The method is presented by Martelli in [7], while Daoud et al [3] apply it in road detection with several constraints for always getting a result, and the right one.
Martelli [7] defines the cost on a series of points such that it depends also on their curvature, and then performs an A* search in a graph for finding the shortest path, with a heuristic for speeding up the search. We avoid the search in a graph, the use of a heuristic, and the corresponding computational problems (dead ends, local maxima, exponential complexity, artifacts, see [3]). Relatively to [3], we avoid the use of redundant computations and of many control parameters, and consider cliques of higher order taking into account also contrast information.

5.2 Method 2 : [4]

Fischler et al [4] add a constant to the local cost such that the geometric length of the path is reduced accordingly, and some smoothness therefore achieved. However, this is not bound directly to the notion of curvature. Here too, our potential defined on general cliques offers more possibilities, in a mathematical formalization. Besides, we optimize the scheduling of the scannings ([8],[9]).

5.3 Method 3 : [6]

Jedynak et al [6] use a process where curvature intervenes at different steps dealing with different resolutions. First, at a low resolution, configurations where segments are perpendicular are prohibited. Then, at a high resolution, constraints are imposed on the angle between two successive segments. We should stress that their method works in fact on an image which has been binarized, and that the curvature considered is between two binary segments. In our method, the approach is different since curvature constraints are imposed at the unique high resolution and on *grey-level* features.

We should stress finally that our method combines in some sense the methods of Martelli [7] and Fischler et al [4] : the algorithm is close to [4] while the curvature cost is defined as in [7]. We combine the advantages of both at the same time (rapidity, reliability and simplicity, versus use of curvature information).

5.4 Complexity

The algorithms are performed within thirty or forty scannings, and the complexity is proportional to the number of pixels in the image. This rapidity is due to the fact that these algorithms privilege the zone of the feature (where the energy values change) and that they are both recursive : the change of one pixel value is taken into account *during the same scanning*. Thus, the number of scannings is linear with respect to the number of sinuousities of the feature.

5.5 Drawbacks

However, this curvature information remains very local in our implementation :

- the only information which we may dispose of is the direction of the last pixel to the current one.
- only local curvature is considered. Thus, a path oscillating around a fixed direction, with smooth and regular zigzags, would produce a higher cost than a path having one sudden change of direction between two straight segments.

One may envision to generalize these algorithms to n successive pixels, but the number of cliques to consider becomes huge and the programming becomes difficult.

6 Conclusion

We have presented herein an extension of the F* algorithm of Fischler et al to neighborhoods of size larger than one, which allowed us to take into account curvature information in the local cost function.
The algorithms presented may be useful for characterizing features in the image or for imposing smoothness constraints on the result, in particular in road detection. The method is efficient, simple, and easy to adapt to other applications ([8],[9]).
Future research deals with an alternative to this solution and aims at considering the curvature information in a more global way during the back-propagation.

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] J. Besag, Spatial Interaction and the Statistical Analysis of Lattice Systems, *J.R. Statist. Soc.*, vol. B-36, pp. 192-236, 1974.

[3] M. Daoud, C. Roux, A. Hillion, Une application de la theorie des graphes a l'extraction automatique des reseaux de communication dans les images du satellite SPOT. *12th Colloquium GRETSI*, Juan-Les-Pins, 12th-16th June, 1989.

[4] 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.

[5] M. Golay, Hexagonal Parallel Pattern Transformation. *IEEE Trans. On Computers*, vol. 18, n 8, pp. 733-740, August 1969.

[6] B. Jedynak, D. Geman, A. Gagalowicz, Detection de reseaux routiers a partir des images du satellite SPOT. *8th RFIA Conference*, vol. 2, Lyon-Villeurbanne, 25th-29th Nov, 1991.

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

[8] 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.

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