An Active Testing Model for Tracking Roads in Satellite Images [*]

Donald Geman
Department of Mathematics and Statistics
University of Massachusetts
Amherst, MA. 01003
Email: geman@math.umass.edu

and

Bruno Jedynak
INRIA - Rocquencourt
Domaine de Voluceau, B.P. 105
78153 Le Chesnay Cedex, France
Email: Bruno.Jedynak@inria.fr

August, 1994

[*] This work was partially supported by the National Science Foundation under grant DMS-9217655, the Office of Naval Research under contract N00014-91-J-1021, the Advanced Research Projects Agency under contract MDA972-93-1-0012, and the CNES under Marché 833/CNES/93/0972/00.

Abstract

We present a new approach for tracking roads from satellite images, and thereby illustrate a general computational strategy ("active testing") for tracking 1D structures and other recognition tasks in computer vision. Our approach is related to recent work in active vision on "where to look next" and motivated by the "divide-and-conquer" strategy of parlor games such as "Twenty Questions." We choose "tests" (matched filters for short road segments) one at a time in order to remove as much uncertainty as possible about the "true hypothesis" (road position) given the results of the previous tests. The tests are chosen *on-line* based on a statistical model for the joint distribution of tests and hypotheses. The problem of minimizing uncertainty (measured by entropy) is formulated in simple and explicit analytical terms. To execute this entropy testing rule we then alternate between data collection and optimization: at each iteration new image data are examined and a new entropy minimization problem is solved (exactly), resulting in a new image location to inspect, and so forth. We report experiments using panchromatic SPOT satellite imagery with a ground resolution of ten meters: given a starting point and starting direction, we are able to rapidly track highways in southern France over distances on the order of one hundred kilometers without manual intervention.

Index terms - Decision tree, model-based tracking, active testing, roads, SPOT images.

I Introduction

We present a new algorithm for tracking major roads from panchromatic SPOT satellite imagery with a ground resolution of ten meters. This is the immediate goal of this work, and our approach will be demonstrated on SPOT images of size 6000×8000, representing a $60km \times 80km$ square on the ground, in this case in southern France. At the same time, the tracking algorithm illustrates a very general approach ("active testing") to classification and recognition problems, which is motivated by the proposition that one can make fast and accurate decisions "simply by" asking the right questions in the right order, like in parlor games such as "Twenty Questions". We have also applied this method to recognizing other two-dimensional, deformable shapes, such as handwritten numerals appearing in photographs of zip codes. A preliminary version of the road and numeral studies appeared in [16]; a full report of the numeral work will appear elsewhere.

In the general classification problem there is finite list of possible "hypotheses" or "states of nature" and an initial (or "prior") distribution over hypotheses which reflects the initial uncertainty about which one is true. We wish to determine X, the true hypothesis, on the basis of various "questions" or "tests." (In our case, the hypotheses are the possible continuations of a road passing through a known point and the tests are based on local filters for detecting short road segments.) The tests are performed sequentially and a decision is made when the uncertainty about X becomes sufficiently small, i.e., when one hypothesis (or small cluster) dominates the updated (or "posterior") distribution on X given all the tests to date. In our case this corresponds to localizing the road to pixel accuracy.

Implicit in this approach is the concept of a decision tree: each interior node is assigned one of the tests, each departing branch represents a possible test response, and each terminal node is assigned one of the hypotheses. The assignment of tests, or "testing rule," is adaptive in the sense that the choice of the test at each node may depend on the test values observed at all the antecedent nodes along the same branch. The testing rule is regarded as a code for efficient classification. Ideally, the choice would be driven by a global measure of efficiency, such as achieving the most accurate classifier for a fixed average number of tests, or reaching the fastest decision at a fixed level of accuracy. But these optimization problems are intractable, and instead one usually chooses some type of "greedy" testing rule in which the tests are chosen one at a time. The principal example is "entropy testing": at each junction choose the next test in order to remove as much additional uncertainty as possible about X.

The standard construction of decision trees (e.g., in coding, CART, and machine learning) is off-line, nonparametric, and based on "training data." However, in our formulation of tracking, it is impossible to pre-compute and store the entire decision tree: it has too many branches from each (interior) node, it is too deep (i.e., too many tests are needed to reach a decision), and the number of possible values of X is enormous. Instead, the testing rule is computed *on-line*: each new filter is chosen during the actual tracking based on the particular filter results previously encountered; in other words, we only compute the branch of the tree that is needed for the data at hand. In fact, the amount of time necessary to perform the tests is small compared to determining the "right" test to perform. On the other hand, compared to maximum likelihood estimation, the number of tests actually performed until a decision is made is exponentially small.

Our approach is model-based rather than nonparametric. Usually, the testing rule is computed from the *empirical joint distribution* of tests given X, derived from training data (sample roads) and without reference to a model. Instead, in our case, there is a statistical model for the distribution of the tests given the road placement, X, which involves assumptions of conditional independence and space invariance. As a result, we need only specify two *univariate* distributions: the test response for "background" and for "road." These two densities are easy to learn from image data, after which the system is fully determined mathematically. As a result, we can formulate the problem of minimizing entropy in explicit and relatively simple analytical terms. To execute the testing rule we then alternate between data collection and optimization: at each iteration, new image data are examined and a new entropy minimization problem is solved (exactly) resulting in a new image location to inspect, and so forth. The model is stochastic but the algorithm is deterministic.

The locations which are chosen are usually *not* on the actual road, although they are rarely very far away. They are simply the most informative places to gather data at the time they are chosen. Moreover, the corresponding sequence of physical locations does not trace out a regular path through the image, one location "following" another. Instead, there is considerable "backtracking" and, in general, the behavior is erratic, except at a coarse scale. This staccato effect is very evident in a video illustration.

Previous work on road extraction, including detection and tracking, is reviewed in Section II. In Section III we present some background material on the strategy of "divide-and-conquer," decision trees, and related paradigms (e.g., in active vision). The mathematical formulation is given in Section IV including the active testing model, an efficient characterization of entropy testing, and a description of the resulting algorithms for tracking and localization. Experiments are reported in Section V, including comparisons with maximum likelihood tracking. Some possible extensions are mentioned in Section VI and concluding remarks are made in an Section VII. Finally, some deferred mathematical arguments appear in an Appendix.

II The Road Problem

A SPOT Images

The SPOT satellite provides high resolution images of the whole world. The amount of geographical information displayed by an image covering an area of about $4,000\ \text{km}^2$ is enormous, and the database of available images is gigantic and growing. Examples of extracted sub-images are presented in Figures 1, 12 and 13. For any specific use, such as cartography, urban planning, or resource management, the raw data must be converted into meaningful (semantic) information. This task is usually long and tedious for a human operator.

B Previous Work

For about twenty years there has been an intensive effort to produce software that could support a specialist, like a geographer or cartographer, in the analysis of structures appearing in remotely sensed images. This work is aimed at providing assistance in the automatic extraction of roads. These are probably the most easily identified man-made structures in SPOT images, at least for major highways; see again Figures 1, 12 and 13.

Despite this effort, there is no software sufficiently reliable for practical use. This might seem less surprising after looking at Figure 1, which shows us that the *local* spectral and geometric characteristics of SPOT roads are often not sufficiently distinctive to separate them from background structures. (This particular image is not special; see, for example, the discussion in [40].) The same difficulty arises with Landsat TM images, even when combining all seven spectral bands; see [5]. In other words, the problem of road extraction is more *global* than it might at first appear.

B.1 Local Methods

The first studies on road extraction from aerial or satellite images were mainly devoted to classifying individual pixels as either “road” or “background” using only the image data in

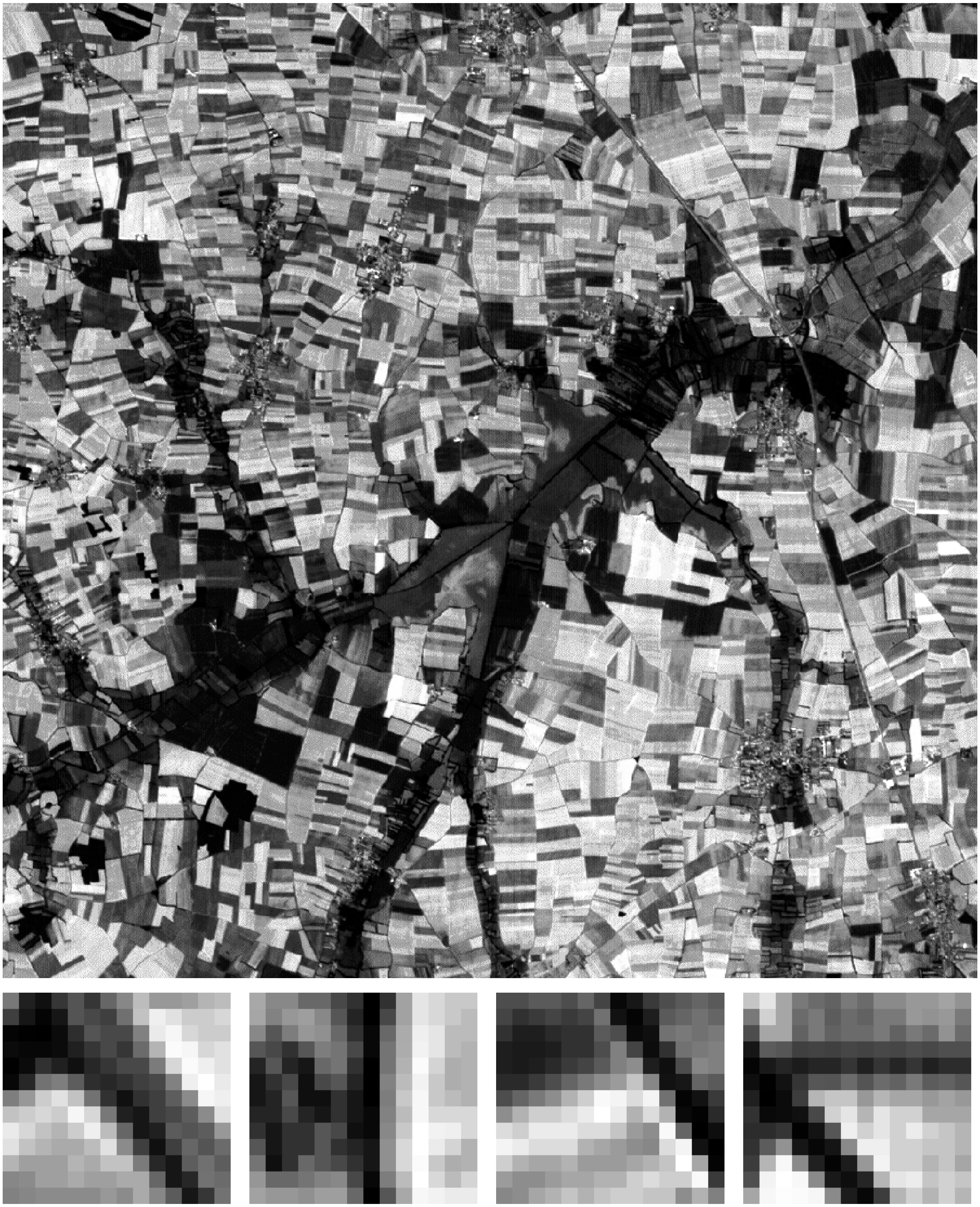


Figure 1: Top: A 1024x1024 SPOT image of La Rochelle, France. Bottom: Four subimages: the two on the left are on the main road and the two on the right are from the background.

a small neighborhood surrounding the target pixel. Classical image processing techniques were tried, such as generic edge detectors ([19], [35], [45]), "crest detectors" ([3]), and morphological operators ([8], [10], [14], [40]). Comparisons among these methods appear in [14] and [40]. Operators dedicated to the extraction of roads were also developed, such as the so-called "Duda-Road-Operator" in [9]. More recently, statistical classifiers and neural networks have been applied and compared; see [5].

The actual grey level values on the main roads are not informative in the sense that roads do not appear systematically brighter or darker than the surrounding background. This is evident upon examining the data. Although our method is global, we also employ local operators (Section IV-C); however, in contrast to most of the work cited above, we have chosen *invariant* operators, based only on *relative* brightness, returning the same result under any linear transformation of the grey levels.

B.2 Global Methods

The most often-cited paper about road extraction is Fischler et al. [11]. The authors present a *global* formulation of the road problem involving a graph over the pixel lattice. The vertices are assigned weights using local operators and there is a corresponding cost function on paths through the graph; dynamic programming is used to find the path with minimal cost between two specified nodes. This model was extended to incorporate curvature constraints in [33]; see also [19] and the promising approach for aerial imagery in [2]. In a previous study (see [15]), we worked in a similar framework and attempted to produce a fully automatic algorithm, meaning there was no need to pre-select points along the road. We abandoned this approach because we could not extend it robustly and efficiently to very large images.

Graph-based methods are also used in [40], where the focus, as here, is on *tracking.* The algorithm is iterative: at each step, a certain number of paths emanating from the seed are inspected, some of which are kept and extended and others discarded. These choices are based on a cost function motivated by heuristics about road properties, such as brightness, width, curvature and the presence of edges. Our algorithm is also an iterative tracking algorithm and will be compared in Section V-C to one that is simpler but in the same spirit as the one in [40].

Another research domain in road extraction from satellite images involves the use of additional data, such as road maps (see e.g. [38]) or Geographical Information Systems (see e.g. [44]). The methodology is then rather different.

III A Divide-and-Conquer Strategy

A Ideal Testing

There is a certain parlor game which seems to be known everywhere, for example as "Twenty Questions" or "jeu des métiers." One player chooses an instance from a general category such as famous persons, unusual occupations ("What's My Line?"), or historical events; another player, who knows the category, tries to guess the particular instance by asking questions (e.g., "is the person dead or alive?") which divide the possibilities into two groups .

This is an *ideal* game because there is no restriction on the questions and no ambiguity in the responses. Still, it is an interesting mathematical problem to determine the optimal strategy for minimizing the mean number of questions that are asked until the chosen instance or "hypothesis" is determined. The solution, and bounds on the mean decision time, are known from results in coding theory ([24], [39], [48]) and formalize the strategy of divide-and-conquer. Naturally, the results depend on the a priori likelihoods assigned to hypotheses; in particular, the *Huffman code* is optimal and gives a mean decision time roughly equal to the entropy of the starting distribution.

B Constrained Testing

This framework is not useful as it stands for practical decision-making problems. For one thing, the hypotheses are too well separated if *every* subset question is allowed; in reality it may be impractical or impossible to have exactly the right split available at each junction, not to mention entertaining a data structure whose size is exponential in the number of hypotheses.

In one variation ("Constrained Twenty Questions"; [12], [13],[30],[32]) the set of possible questions (allowable splits) is limited. Let M denote the number of possible hypotheses and let N denote the number of possible questions, each of which divides the set of hypotheses into two groups. There is also an initial distribution μ_0 over hypotheses. For "small" problems, say for $M = 100$ and $N = 20$, one can in fact find the optimal strategy (for minimizing the mean number of questions) with dynamic programming. But the general problem is NP complete [25] and considerable effort has been applied (see [13],[32] and the references therein) to finding good "suboptimal" strategies. In particular, we shall focus below on a class of "greedy" strategies for choosing and ordering tests which are sometimes called "splitting algorithms" ([12],[13],[32]) in applications to fault-testing, machine diagnostics, and related problems.

A still more general framework would incorporate *ambiguous* responses. It is not realistic to assume the test results are determined without error (nor that the answers are simply "yes" or "no"). Given the true hypothesis, say X, the tests are usually *non-deterministic* due to image noise and the enormous variability in the presentation of roads in real satellite data. The tests must then be regarded as *random variables*, say $Y_1, Y_2, ..., Y_N$. In our case, for example, the tests are local functionals of the image data (see Section IV-C) based on matched filters for detecting short road segments. The responses along actual roads are often more characteristic of background locations and vice-versa. The tests are then distinctly non-deterministic. (In contrast, in Constrained Twenty Questions, the random variables are binary and conditionally degenerate: there is no randomness in the test outcomes once X is known.)

Notice that there are now two distinct sources of randomness: uncertainty about the true hypothesis X and uncertainty in the test answers. In general, the family of tests is very large and we may assume that, with probability one (or nearly one), the entire set of tests (essentially) determines X. In other words, given all the test results, the conditional distribution on X would be extremely peaked. (Equivalently, the maximum likelihood estimator is very accurate in principle.)

The purpose of active testing is to perform only a small fraction of the tests and still accurately estimate X.

C Testing Rules

The tests are performed sequentially and adaptively, meaning that at each stage we may utilize the results of previous tests in order to choose the next one. For simplicity, assume each test Y_n assumes values in a common, finite set $\mathcal{Y}$ of size J; these are the possible answers to the questions. A *testing rule* π is then a sequence $\pi_1, ..., \pi_N$ of distinct test indices where π_k represents the index of the k'th test. The first test is π_1 which is determined by the joint distribution of X and $Y_1, ..., Y_N$; see Section III-D below. After that, each test π_{k+1}, $k \geq 1$, depends on the answers to the previous k tests. Consequently, $\pi_{k+1} = \pi_{k+1}(y_1, y_2, ..., y_k)$, where $y_1, ..., y_k$ denote the answers to the previous tests $Y_{\pi_1}, ..., Y_{\pi_k}$. A graphical representation is presented in Figure 2.

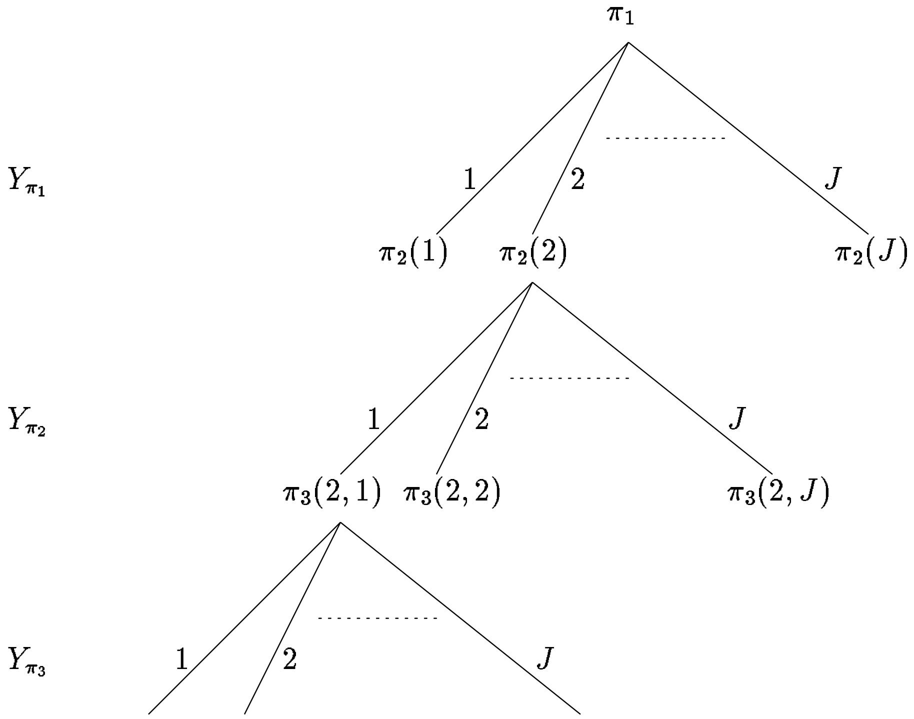


Figure 2: Graphical representation of a testing rule

Ideally, one would seek the "optimal" π relative to some criterion, for instance using as few tests as possible at a given accuracy level, or achieving the best accuracy with a given number of tests. Since these problems are intractable we simply choose each new test to minimize the residual uncertainty about the location of the true road; this will be made precise in Section IV-E.

D Statistical Model

In order to have a well-defined mathematical problem we must specify the joint distribution of hypotheses and tests. One way is to specify two distributions: the marginal (or "prior") distribution $\mu_0 = \{\mu_0(x) = P(X = x), x \in \mathcal{X}\}$ over the set $\mathcal{X}$ of hypotheses and the joint (conditional) distribution of the tests, namely $P(Y_1 = y_1, ..., Y_N = y_N | X = x)$. If the tests are conditionally independent given X then it suffices to provide the marginal (conditional) distributions, $P(Y_n = y_n | X = x)$ for each test.

It is necessary to estimate the conditional distributions of tests from data. In our case, since the tests are assumed to be conditionally independent, the testing rule may be computed once the marginal test distributions are estimated. (This is admittedly a simplifying assumption; see Section IV-B.) In particular, given any specific history for the "first" k tests $Y_{\pi_1}, ..., Y_{\pi_k}$ the conditional (or "posterior") distribution $P(X = x | Y_{\pi_1} = y_{\pi_1}, ..., Y_{\pi_k} = y_{\pi_k})$ is then determined analytically. (Notice that also conditioning on $\pi_1, ..., \pi_k$ is superfluous since π_1 is constant, Y_{π_1} determines π_2, etc.)

In contrast, in nonparametric approaches, the tests are conditionally dependent and the testing rule *itself* must be estimated from training data rather than computed analytically. The estimation problem is then far more difficult. In particular, one usually requires a very large database of sample hypotheses and corresponding test results; otherwise one lacks a sufficient supply of test histories to properly estimate the residual mean uncertainty in X for each new candidate test.

E Decision Tree

A *decision tree* is a convenient structure for representing the testing rule as well as stopping and classification rules. Each interior node of the tree is assigned one of the tests and each terminal node is assigned one of the hypotheses. The decision tree then has test Y_{π_1} at the root (or level-one) node, which branches into $J = |\mathcal{Y}|$ nodes corresponding to the possible answers $y \in \mathcal{Y}$ to the first test. Each of these J level-two nodes then branches into J level-three nodes corresponding to the outcomes of Y_{π_2}, and so forth, as in Figure 2. In addition, certain nodes are declared to be "terminal" and labeled by one of the hypotheses $x \in \mathcal{X}$. This labeling, $\hat{X}$, is the estimator of X. In general, each hypothesis x will be associated with many terminal nodes, and following such a terminal node back to the root will produce one sequence of observations for which $\hat{X} = x$. Testing is ceased when one of the hypotheses (or one coherent class of hypotheses) becomes overwhelmingly likely. One obvious stopping time is then

$$\tau = \min\{1 \leq k \leq N | \max_{x} P(X = x | Y_{\pi_1}, ..., Y_{\pi_k}) \geq 1 - \epsilon\}.$$

This is equivalent to stopping as soon as the entropy of the posterior distribution falls below a threshold. In our application to road tracking, we simply fix the total number of tests performed. However, to reduce the amount of memory and book-keeping, we occasionally "prune" (see Section IV-G) by re-setting the starting point when one portion of the full road X becomes overwhelmingly likely in the sense above. It should be emphasized that this is merely a convenience; in particular, our construction of the testing rule is an *exact* calculation and re-initialization has no effect on the choice of tests: the arcs that are fixed are too likely to be on the true road to ever be chosen for examination.

F On-Line Implementation

In nearly all previous applications involving decision trees the entire tree is constructed off-line and stored for later use. This can be a massive job. Of course during the actual execution only *one branch* of the tree is traversed from the root node to a terminal node, the branch dictated by the data. (This is usually very fast since all that needs to be done is to follow pre-computed instructions; in particular, there is no on-line optimization.)

We only compute the branch we need - the one determined by the responses to our tests. In fact, in our case it is quite impossible to compute and store the decision tree since we are going to ask hundreds or even thousands of questions. Since there are then many *on-line* iterations of a basic optimization subroutine, it is vital to reduce the computation of the testing rule to a form as efficient as possible. This is computationally feasible since our analytical model yields a very simple characterization of the entropy testing rule; see Section IV-E.

G Previous Applications of this Methodology

Connections with coding theory have already been mentioned. Papers dealing with the general methodology of decision trees and the role of entropy in pattern recognition include [7], [23], [29] and [34] and [47].

Decision trees have already been used for shape and object recognition, for example for recognizing printed and handwritten characters ([20], [46]), biological shapes ([31]) and other two-dimensional shapes ([1]), and three-dimensional objects ([18], [21],[41],[42]). Usually, the testing rule is based on entropy reduction and involves extracting local image features.

Our work differs markedly from the popular "hypothesize and test" paradigm in computer vision in which (complete) hypotheses are repeatedly elicited in a process referred to as "indexing." On the contrary, we do not isolate and test specific hypotheses about the *global* road placement; it is not computationally feasible - there are simply too many possibilities. Instead, our "indexing," such as it exists, is dynamic and stochastic: we gradually formulate specific conjectures by sequentially performing simple tests and re-evaluating the evolving distribution over hypotheses until it becomes sufficiently peaked to declare a solution.

Our approach is much closer, at least in spirit, to work in "active vision" ([43]) concerning the manner in which the data is processed and the selection of new fixation points. In particular, the active testing model is an example of a sequential decision making process and one way to formulate the "where to look next" problem. Another way to formulate the same control problem (i.e., choosing the most useful information to process) uses Bayesian networks and decision theory; see [37] and the references therein.

Decision trees also appear in statistics ([4]), machine learning ([36]), and classical pattern recognition (see, e.g., [22]). One difference is that in applications to image analysis no "feature vector" is given a priori; the raw data are intensities. In some applications in design of experiments and adaptive control ([6], [17], [28]) the tests are *repeatable* and the emphasis is on *asymptotic* results for large numbers of tests. In our formulation there is a limited number of nonrepeatable tests.

IV Tracking by Active Testing

A A Geometric Model for Roads

A major road in a SPOT image is modeled as a discretization of a smooth, planar curve whose curvature is bounded by a known value. We use a piecewise linear approximation; each knot corresponds to a pixel and the segments, which we call "arcs," are digitized lines (actually of width *two* pixels - see Section IV-C below), each of approximatively the same length. Such an object is completely determined by an ordered list of regularly spaced pixels (the knots) and a rule which constructs a digitized line between any two pixels. The curvature constraint is expressed by limiting the angle between successive arcs. Experimental results show that if the length of the segments is sufficiently small (say 12 pixels, although this is conservative), then major roads in SPOT images can be well approximated by allowing only three angles, labeled $\{0, 1, -1\}$, between successive segments. These correspond to "no turn," meaning that successive segments have the same orientation, and the two smallest turns possible at the level of discretization, which correspond to changes in orientation of approximately plus or minus five degrees. Figure 3 is an example of a road containing three arcs (shown as one pixel wide for simplicity). In practice, the number of arcs we consider is at least several hundred.

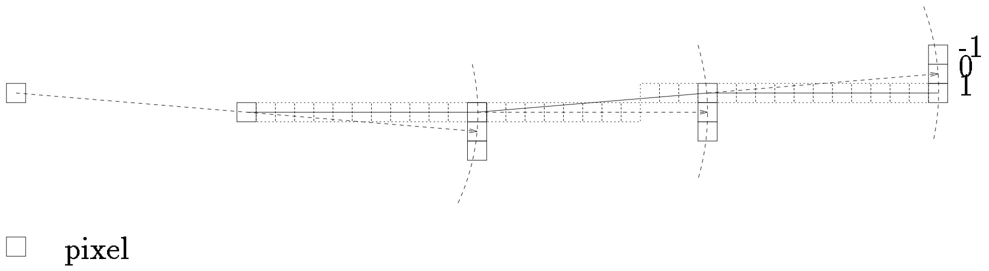


Figure 3: This road containing 3 arcs is identified with the sequence $(-1, -1, 1)$

We suppose from now on that the first arc is given. In order to simplify the exposition, we also fix the length L in arcs of the portion of road we wish to track. Actually, in implementing the entropy testing rule, the computations do not involve L until a test is made at an image location corresponding to an arc of depth L, so that any "large" value of L produces the same result - the same arcs to examine in the same order.

Each road candidate is then identified with a sequence of L digits in $\{0, 1, -1\}$. These are the hypotheses $x \in \mathcal{X}$, which can then clearly be identified with the branches of a ternary tree of depth L. (Notice that two arcs on separate branches of this tree may correspond to the same *physical* location since the two roads represented by these branches have merged.) The total number of hypotheses is then $M = 3^L$. In Figure 4 we show the tree for $L = 3$.

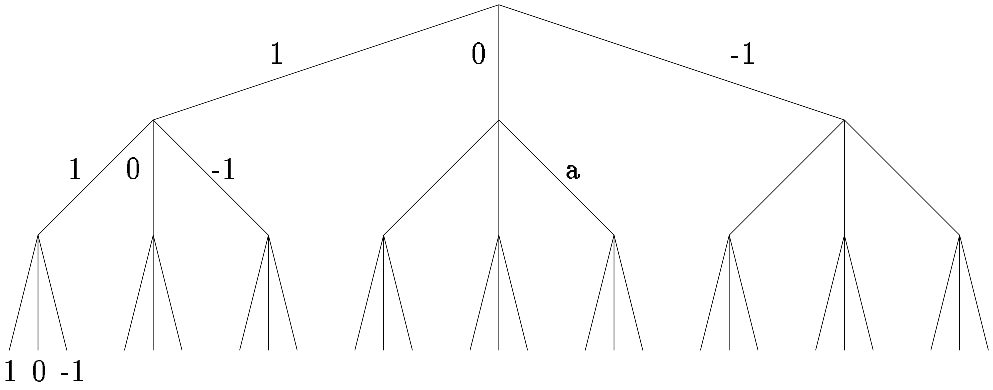
Note: This "representation" tree should not be confused with the *decision* tree, which is J-ary (not ternary) and which represents the testing rule (not the set of hypotheses). (Recall that J is the number of test answers.)

B A Statistical Model for Roads and Tests

We assume that exactly one hypothesis is true, denoted by X, and chosen according to some initial distribution $\mu_0 = \{\mu_0(x), x \in \mathcal{X}\}$, $\mu_0(x) = P(X = x)$. In fact, we shall simply take μ_0 to be *uniform* on $\mathcal{X}$: $P(X = x) = 3^{-L}$ for each $x \in \mathcal{X}$. The following development is not specific to this choice and extends immediately to any "real" prior distribution, meaning one which expresses "soft constraints" in addition to the hard constraints embodied in the piecewise linear approximation and curvature bounds. For example, the sequence of arcs along each branch might be modeled as a Markov process in order to introduce a dependence structure associated with *maintaining* curvature; see Section VI.

Let $\mathcal{A}$ denote the set of all arcs. Also, let $\mathcal{C}_a$ denote the set of roads $x \in \mathcal{X}$ that contain arc a. There is a test, or question, associated with each arc in $a \in \mathcal{A}$, namely: "Does the true road contain arc a?" or, equivalently, "Is $X \in \mathcal{C}_a$?" (See Figure 4.) Consequently, the set-up is the same as the game of "Constrained Twenty Questions," with the additional feature that the set of allowed questions can be represented graphically by the nodes of a ternary tree: subsets $\mathcal{C}_a$ and $\mathcal{C}_b$ associated with two distinct nodes are either disjoint (when a, b lie on different branches) or one subset is included in the other (when a, b lie on the same branch).

The possible answers are not simply "yes" or "no" but rather are labeled $\{1, ..., J\}$. These values represent the output of a matched filter (see below) that uses the local image data near the location corresponding to arc a in order to measure how well these data match our expectations about the relative brightness difference between real road segments and the local background.



$$\mathcal{C}_a = \{(0,-1,-1),(0,-1,0),(0,-1,1)\}$$

Figure 4: The set of roads containing three arcs. Testing for X in $\mathcal{C}_a$ consists in evaluating a matched filter at the location in the image corresponding to arc a.

We make two assumptions about the nature of the ambiguous responses, i.e., the statistical distribution of the family of tests $\{Y_a, a \in \mathcal{A}\}$:

- **Conditional Independence.** The tests are *conditionally independent* given X. In reality this is false due to overlap in the support of the image operators and the fact that occasionally two distinct arcs may correspond to the same image location; nonetheless it is a reasonable approximation and provides a tractable model. Thus,

$$P(Y_a = y_a, a \in \mathcal{A} | X = x) = \prod_{a \in \mathcal{A}} P(Y_a = y_a | X = x).$$

- **Space Invariance.** The marginal (conditional) distributions $P(Y_a = y_a | X = x)$ depend only on whether or not arc a is contained in road x. In particular, for any two arcs a and b along the road x ($x \in \mathcal{C}_a \cap \mathcal{C}_b$) the corresponding tests are identically distributed as well as independent; similarly for any two arcs in the background. Let us denote these distributions by p_1 and p_0 respectively. Then for any $y_a \in \{1, ..., J\}$:

$$P(Y_a = y_a | X = x) = \begin{cases} p_1(y_a) & \text{if } x \in \mathcal{C}_a \\ p_0(y_a) & \text{if } x \notin \mathcal{C}_a \end{cases}$$

Thus,

$$P(Y_a = y_a, a \in \mathcal{A} | X = x) = \prod_{a:x\in\mathcal{C}_a} p_1(y_a) \prod_{a:x\notin\mathcal{C}_a} p_0(y_a).$$

C Local Filtering

The response of a test is based on a local filter applied to the raw image data. The filter is designed to identify short, linear segments that are likely to lie on major roads. The distributions p_0 and p_1 quantify its performance and were estimated using several test images; see the top of Figure 5. The testing rule depends rather critically on these distributions, as does the performance of the tracker. Consequently, some care was taken to estimate them accurately. In particular, the smaller roads appearing in the images were taken into account when estimating p_0. They are then fixed throughout and all the results in Section V use these estimated distributions.

The basic assumption that is utilized is that two pixels in close proximity and both on the road should have a smaller intensity difference than that of two pixels, one of which lies on the road and the other off the road. Since the major roads appear in SPOT images as structures of width between two and three pixels, the filter is based on a pair of adjacent, digitized lines (an "arc"); due to digitization, the actual lengths of these arcs in pixels differ slightly depending on the orientation, but all are between eight and twelve pixels. The overall response of the filter is the aggregate of the individual, binary responses of elementary "cliques" along the arc. There are about twenty cliques per arc, each based on a pattern of six pixels $(t_1, ..., t_6)$ arranged as shown in Figure 6. Let $I(t)$ denote the image intensity at pixel t. Each such clique is assigned a value 0 or 1; the value is 1 if $|I(t_1) - I(t_2)| < \min\{|I(t_3) - I(t_1)|, |I(t_5) - I(t_1)|, |I(t_4) - I(t_2)|, |I(t_6) - I(t_2)|\}$ and 0 otherwise. The test result is simply the sum of clique responses quantized to J distinct values, denoted by $\{1, ..., J\}$. (The value $J = 10$ is used for all the experiments in Section V.)

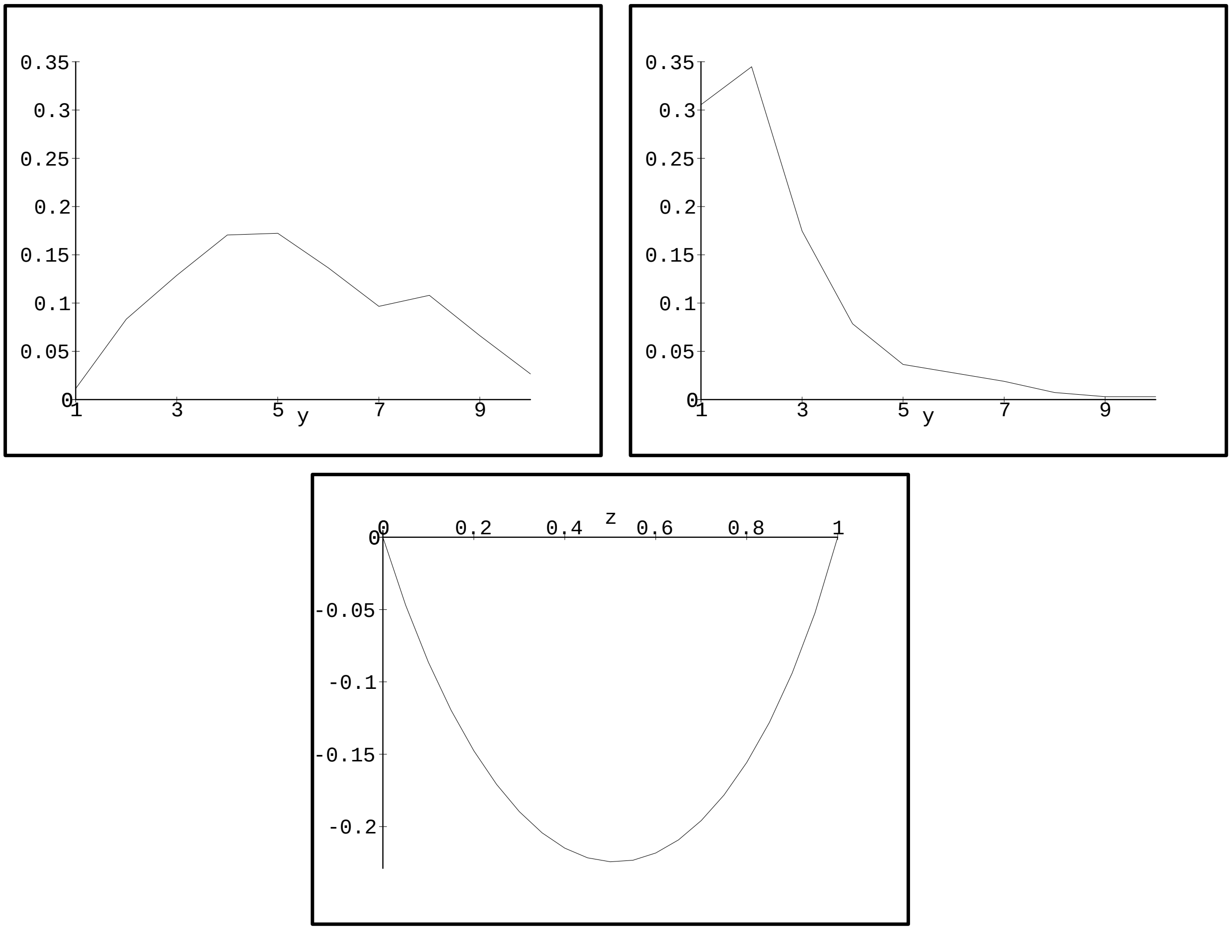


Figure 5: Top: Estimated distribution of the filter on roads (left) and in the background (right). Horizontal axis: possible filter values. Vertical axis: frequency of response. Bottom: the function $z \mapsto \phi(z)$.

Notice that this filter, in contrast to most crest detectors, is invariant with respect to *linear* (non degenerate) transformation of the image intensities. This might partially explain why the distributions p_0 and p_1 in Figure 5 can be used for a large variety of images, such as those presented in Section V.

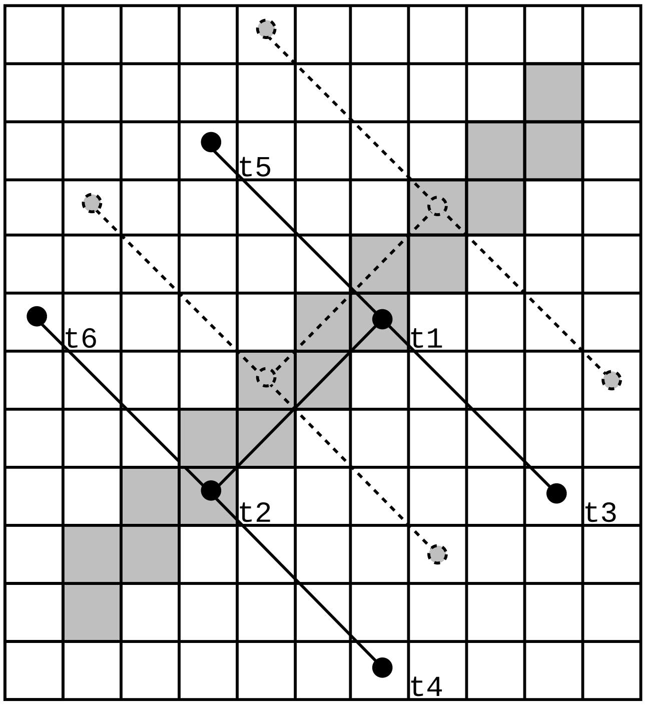


Figure 6: Local filtering.

D Maximum Likelihood Classification

In principle, we can imagine doing all the tests and computing the maximum a posteriori (MAP) estimator

$$\begin{aligned}\hat{X}^{MAP} &= \arg\max_{x \in \mathcal{X}} \; P(X = x | Y_1, ..., Y_N) \\ &= \arg\max_{x \in \mathcal{X}} \; \mu_0(x) P(Y_1, ..., Y_N | X = x).\end{aligned}$$

Since our prior is uniform, this is equivalent to the maximum likelihood estimator (MLE)

$$\hat{X}^{ML} \doteq \arg\max_{x \in \mathcal{X}} \; P(Y_1, ..., Y_N | X = x)$$

$$
\begin{aligned}
&= \arg\max_{x \in \mathcal{X}} \prod_{a: x \in \mathcal{C}_a} p_1(Y_a) \prod_{a: x \notin \mathcal{C}_a} p_0(Y_a) \\
&= \arg\max_{x \in \mathcal{X}} \prod_{a: x \in \mathcal{C}_a} \frac{p_1(Y_a)}{p_0(Y_a)} \prod_{a \in \mathcal{A}} p_0(Y_a) \\
&= \arg\max_{x \in \mathcal{X}} \prod_{a: x \in \mathcal{C}_a} \frac{p_1(Y_a)}{p_0(Y_a)}
\end{aligned}
$$

Thus we maximize the product of the likelihood ratios.

This is obviously more accurate than any estimator based on evaluating the posterior distribution on X after doing only *some* of the tests. (See [26] for a result on the consistency of the MLE as $L \to \infty$.) However, it is not feasible computationally. The piecewise linear assumption is violated (even for the largest highways) if the length of the segments exceeds about twenty pixels and we wish to track roads for many kilometers. Consequently we must choose L rather large (say order 100). Since there are order 3^L possible roads and order 3^L total tests, it is clearly impossible to compute $\hat{X}^{ML}$.

Indeed, we wish to get roughly the same performance as maximum likelihood by considering only a very small fraction of all the tests, such as order L to $100L$ tests rather than 3^L tests. *In effect, we approximate the maximum likelihood estimator.*

E Entropy Testing

In order to formulate the testing rule more precisely we need a measure of uncertainty. Recall that the (Shannon) entropy of a random variable U is defined by

$$H(U) = -\sum_u P(U = u) \log_2 P(U = u).$$

Actually, we need to measure *conditional entropy*: the anticipated uncertainty remaining in X given the new candidate (and previous test results). First, for any event B the entropy of U relative to the conditional probability measure $P(.|B)$ will be denoted by

$$H(U|B) = -\sum_u P(U = u|B) \log_2 P(U = u|B).$$

Now the conditional entropy of U given another random variable V is defined as $H(U|V) = \sum_v P(V = v) H(U|V = v)$ and $H(U|B, V)$ denotes the conditional entropy under $P(.|B)$, i.e.,

$$H(U|B,V) = \sum_v P(V=v|B)H(U|B,V=v).$$

We now examine the entropy testing rule in our case. The first arc chosen is π_1; it is simply a parameter of the original joint distribution over tests and roads. The second arc chosen is π_2, which depends on π_1 and the response to test Y_{π_1}. In general, the $k+1$'st arc chosen is π_{k+1}, which depends on the results of the previous k tests, denoted by $B_k = \{Y_{\pi_1} = y_{\pi_1}, ..., Y_{\pi_k} = y_{\pi_k}\}$. (We can assume $P(B_k) > 0$.)

The basic idea is this: at stage $k+1$ choose the test Y_a ($a \neq \pi_1, \pi_2, ..., \pi_k$), which maximizes the expected gain in information about X from observing Y_a, having already observed B_k. This is the same as minimizing the expected amount of uncertainty about X under the pending posterior distribution. Thus,

$$\pi_1 = \arg\min_{a \in \mathcal{A}} \; H(X|Y_a)$$

and, for $k \geq 1$,

$$\begin{aligned} \pi_{k+1}(y_1, ..., y_k) &= \arg \min_{a \neq \pi_1, ..., \pi_k} \; H(X|B_k, Y_a) \\ &= \arg \min_{a \neq \pi_1, ..., \pi_k} \sum_y P(Y_a = y|B_k) H(X|B_k, Y_a = y). \qquad (**) \end{aligned}$$

(The condition that a differ from the preceding choices can be dropped since doing the same test again provides no information at all.)

The solution of $(**)$ can be reformulated as follows. Define

$$\phi(z) = H(p_1)z + H(p_0)(1-z) - H(zp_1 + (1-z)p_0)$$

for $0 \leq z \leq 1$. Here $zp_0 + (1-z)p_1$ is the mixture distribution of p_0 and p_1 with weights z and $1-z$ respectively. It is easy to show that ϕ is convex.

Theorem (Characterization of Entropy Testing). *The optimization problem* $(**)$ *is equivalent to choosing the test* $a \neq \pi_1, ..., \pi_k$ *which minimizes the function* $a \rightarrow \phi(P(X \in C_a|B_k))$.

The proof is given in Appendix A. The function $z \mapsto \phi(z)$ for our estimates of p_0 and

p_1 is presented at the bottom of Figure 5.

Note: If the supports of p_0 and p_1 are disjoint, then each test result *eliminates* certain hypotheses (i.e., their posterior probabilities become zero). In this case, it is easy to see that $\phi(z) = z \log_2 z + (1-z)\log_2(1-z)$, which is symmetric about $z = \frac{1}{2}$. Consequently, the testing rule amounts to choosing the test a for which $|P(X \in \mathcal{C}_a | B_k) - \frac{1}{2}|$ is as small as possible. In other words, we choose the arc which most nearly divides the "active" roads into two groups of equal probability. This case includes the noiseless game (e.g., constrained Twenty Questions) wherein the value of Y_a is *determined* by X. Of course it is impossible to design a local test which discriminates perfectly between roads and background, so that the supports are never disjoint in practice.

F Implementation

The on-line execution of the entropy testing rule proceeds as follows. The first step is to determine the arc $a \in \mathcal{A}$ for which $\phi(P(X \in \mathcal{C}_a))$ is minimized; this is π_1. Due to simplifications resulting from the convexity of ϕ and the tree structure of $\mathcal{X}$, this only involves evaluating the function $\phi(P(X \in \mathcal{C}_a))$ for the arcs in $\mathcal{A}_1$, a small subset of $\mathcal{A}$. We will proceed with the broad outlines of the computation; the details are in the Appendix. Then we perform the test Y_{π_1}, i.e., we apply the filter to the image data at the location corresponding to π_1. This yields an integer between 1 and J which we denote by y_{π_1}.

Now suppose we have determined $\pi_1, ..., \pi_k$ and obtained the results $B_k = \{Y_{\pi_1} = y_{\pi_1}, ..., Y_{\pi_k} = y_{\pi_k}\}$. We must choose π_{k+1} in order to minimize the function $a \rightarrow \phi(z_k(a))$ over all $a \in \mathcal{A}$ where $z_k(a) = P(X \in \mathcal{C}_a | B_k)$. First we select an appropriate subset of arcs $\mathcal{A}_{k+1}$ from $\mathcal{A}$. For each $a \in \mathcal{A}_{k+1}$, we evaluate the function ϕ at $z_k(a)$, and select π_{k+1} as the arc a which minimizes $\phi(z_k(a))$. Then we perform the test $Y_{\pi_{k+1}}$ and proceed to the next iteration.

The set $\mathcal{A}_{k+1}$ is constructed by modifying $\mathcal{A}_k$. The purpose is to restrict the search for $\pi_1, \pi_2, ...$ as much as possible but still *guarantee* that the arc $a \in \mathcal{A} \setminus \{\pi_1, \ldots, \pi_k\}$ that minimizes $\phi(z_k(a))$ does indeed lie in the $\mathcal{A}_{k+1}$; again, see the Appendix.

It is here that convexity plays a key role. Here is a simple example. Let $\bar{z}$ denote the number $z \in [0,1]$ for which $\phi(z)$ is minimum. For typical densities p_0 and p_1 this value is between .4 and .6; for the estimated densities shown in Figure 5, $\bar{z} = 0.51$. Notice that $z_k(b) \leq z_k(a)$ for all arcs b such that $\mathcal{C}_b \subset \mathcal{C}_a$. Now suppose that $z_k(a) \leq \bar{z}$ for some arc $a \in \mathcal{A}_{k+1}$; then, by convexity, we must have $\phi(z_k(a)) \leq \phi(z_k(b))$ and hence none of the arcs b which lie "below" a needs to be included in $\mathcal{A}_{k+1}$.

Specifically, then, if $\{a_1, a_2, a_3\}$ are the three arcs branching from the (known) arc at the root of the tree, and if $P(X \in \mathcal{C}_{a_i}) \leq \bar{z}$ for $i = 1,2,3$, then the arc $a \in \mathcal{A}$ which minimizes $\phi(P(X \in \mathcal{C}_a))$ must be among $\{a_1, a_2, a_3\}$. This is the case for our model: since our prior distribution is uniform, $P(X \in \mathcal{C}_{a_i}) = \frac{1}{3} \leq \bar{z}$ for $i = 1,2,3$.

The fact that the arcs reside on a tree is further exploited to compute the necessary probabilities in a recursive fashion; see Appendix C. For example, for any arc $a \in \mathcal{A}_k \cap \mathcal{A}_{k+1}$, we can compute $P(X \in \mathcal{C}_a | B_k)$ in terms of $P(X \in \mathcal{C}_a | B_{k-1})$. Conditional independence and space-invariance are also utilized.

Finally, we wish to emphasize that

- The computation extends to any prior model μ_0;

- The implementation of the entropy testing rule is exact;

- The chosen arcs $\pi_1, \pi_2, ...$ are often *not* on the road.

The explanation for the last observation is that, given the tests to date, the entropy testing rule will never choose an arc which is *very* likely to be on the road; indeed, if $P(X \in \mathcal{C}_a | B_k)$ is much larger than one-half, then test Y_a is not particularly informative and we will learn more by examining an arc which appears less likely to be on the road.

G Estimation of the Road

As it turns out, we can estimate X to any given length (i.e., number of arcs) rather easily by continually re-examining the likelihood that X passes through certain arcs, as we now explain. The estimator so obtained matches the actual road (to the extent this can be visually determined) quite well, in fact to within a few pixels on either side (see Section V-A).

Localization is actually a by-product of "pruning." When tracking on large images it is useful to prune the representation tree from time to time by declaring some arc to be the new starting arc and only considering branches emanating from this arc. One reason for this is to limit the growth of the sets $\mathcal{A}_k$.

Of course we can never know *for certain* that a given arc (other than the original starting arc) is on the true road. However, with some added computational cost, we can identify arcs for which this event is extremely likely. Specifically, at each iteration k, and for each arc $a \in \mathcal{A}_k$, we can update the conditional probability that $X \in \mathcal{C}_a$ given the history of tests B_k. Eventually this probability, namely $P(X \in \mathcal{C}_a | B_k)$, will exceed a threshold $1 - \epsilon$ for some arc $a \in \mathcal{A}_k$. (Due to the construction of the sets $\mathcal{A}_k$, any arc in $\mathcal{A}$ for which this happens must lie in $\mathcal{A}_k$; see the Appendix B.) In this case, the tracking is reinitiated at this arc and the path from this arc back to the previous starting arc is our estimate for the corresponding piece of road.

A conservative value is $\epsilon = .001$. In experiments to date the new starting arc (as well as its antecedents) has always been very close to the actual road. In fact, we have been able to reinitiate the tracking without ever losing the true road. Moreover, the number of arcs in $\mathcal{A}_k$ never exceeds several hundred.

V Experiments

All the results below are obtained by assuming the roads are a priori equally likely. The estimated test distributions are those displayed in Figure 5. None of the experiments presented is based on images that were included in the training set used to produce p_0 and p_1.

A Tracking on Large Images

Figure 7 is extracted from Figure 1. We see the highway on the east side of a small town. There is a smaller road which branches from the highway into the town. We started the tracking near the bottom of the image. The result is shown on the left side of Figure 8. The arcs $\{\pi_1, \pi_2, \ldots\}$ chosen by the algorithm are superimposed on the image; the grey level assigned to arc π_k is proportional to the likelihood ratio $p_1(y_{\pi_k})/p_0(y_{\pi_k})$, with dark representing low values. Notice that high responses are obtained along portions of the smaller road, yet the tracker eventually chooses the highway after a significant detour. We see on the right side of Figure 8 the results of the tracking when initiated on a small road. The number of tests performed (500) is enough to track over considerable distances when properly initiated but the tracker is stalled and continues to "fan out" in vain.

The tracker does indeed "backtrack" rather frequently. Figure 9 is a histogram of the depth in the representation tree of the arc π_k relative to the depth of π_{k-1} based on 500 tests from tracking over Figure 1. For example, we see that less than fifty percent of the time the arc π_k is either at the same depth as the arc π_{k-1} or one arc below. In contrast, if the supports of p_0 and p_1 were disjoint, then π_k would *always* be in one of these two positions, with asymptotic frequencies .4 and .6, respectively; see [26].

In Figure 10, on the left side, we show the result of tracking starting from the bottom of Figure 1. This took about 10 seconds on a SUN-SPARC 2 computer. The arcs which are tested are shown in white. An estimate of X itself (by the method discussed in Section IV-G) is shown on the right side of the same Figure. Notice that the road cannot be estimated all the way to the border of the image; additional data would be needed.

B Tracking on Very Large Images

Figure 11 is taken from the Michelin 240 map of the Languedoc Roussillon region in southern France. The area shown is covered by a single SPOT image (ref. SPOT2 HRV1 K-J: 046-263 20/01/93) of dimension 6012×7870. The tracking was begun on the north side of the image, at the location pointed to by the arrow. The tracker was able to follow the main highway to the southern border of the region. From north to south, the two images corresponding to the superimposed rectangles are shown on the top of Figures 12 and 13. The corresponding tracking histories are shown on the bottom of these Figures. (We do not show the estimate itself, which is very close to the true road.) The field of view was occasionally pruned (see Section IV-G above) but the algorithm was never *manually* re-initiated. Tracking required about one minute on a SUN-SPARC 2 computer.

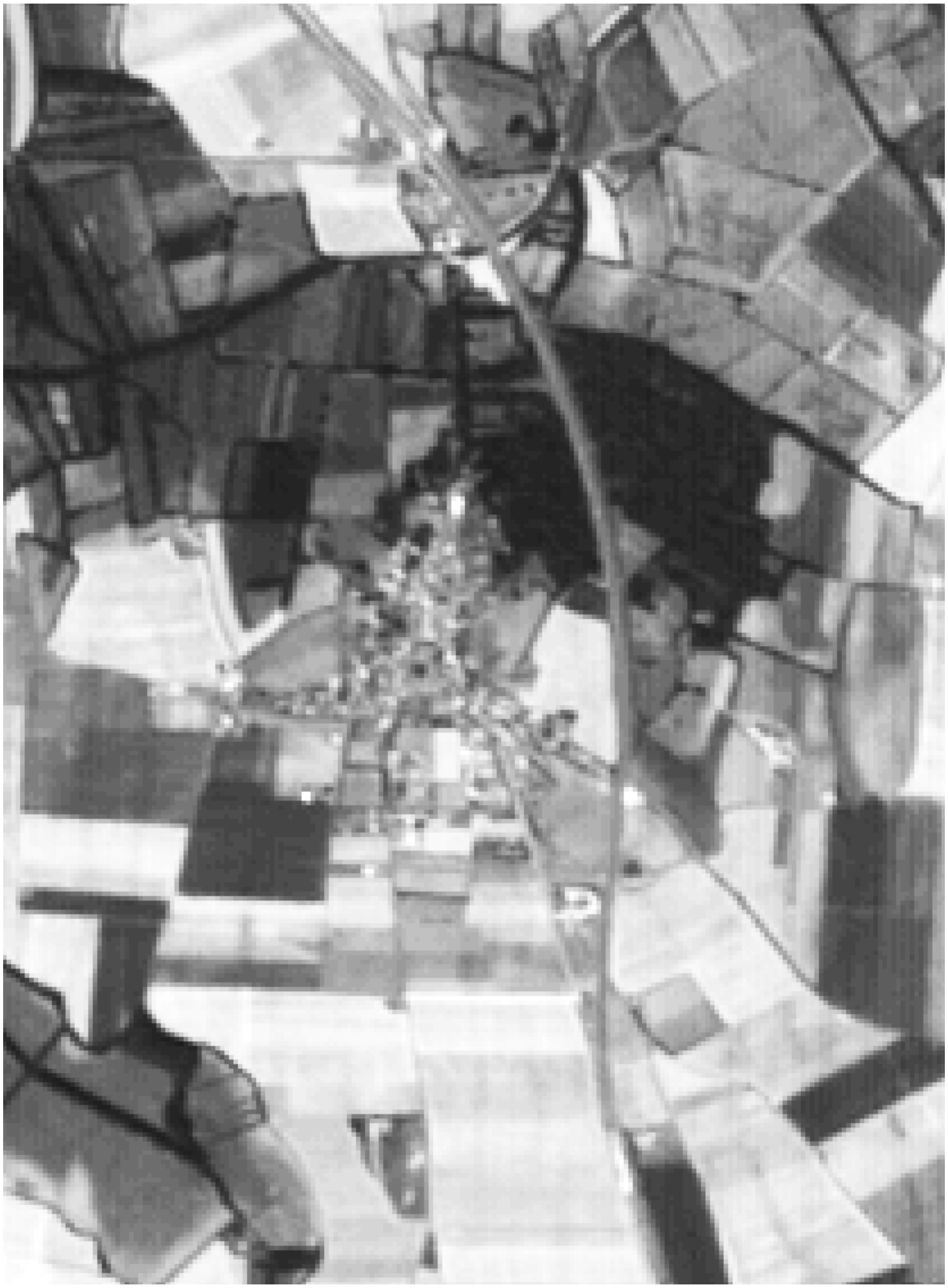


Figure 7: This image is extracted from Figure 1.

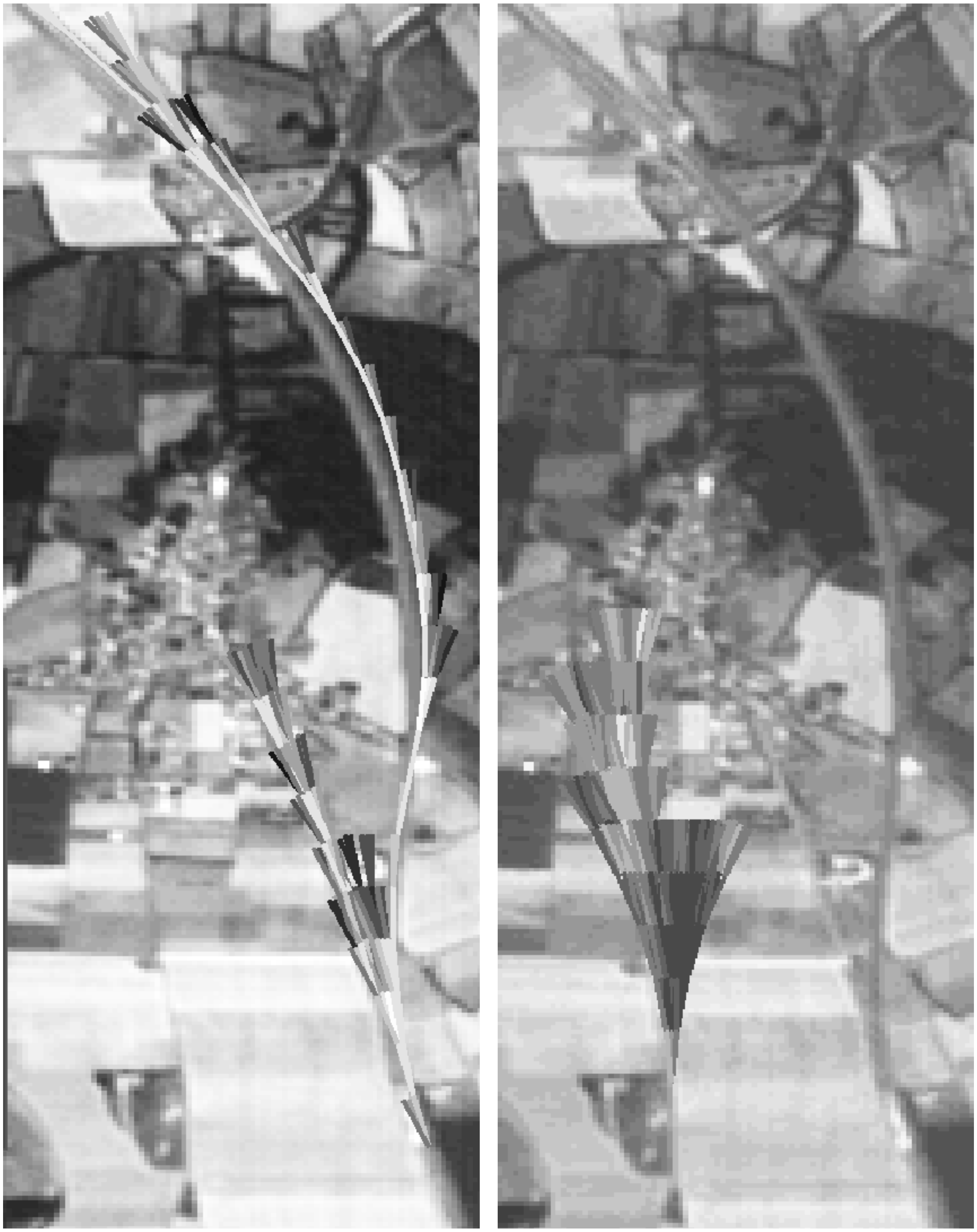


Figure 8: **Left**: Output of the tracking algorithm showing the arcs investigated by the algorithm in colors proportional to the likelihood ratios. **Right**: The tracking was initiated on a small road and is permanently stalled.

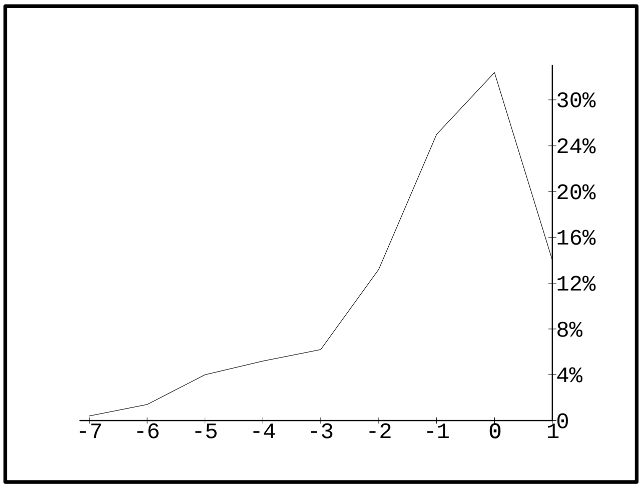


Figure 9: Histogram of the extent of "backtracking." Horizontal axis: depth in the tree of the current test relative to the previous test. Vertical axis: frequency of occurrence in 500 tests.

C Comparisons with Maximum Likelihood

We briefly consider a modified version of the MLE which is computationally feasible. From Section IV-D the MLE is given by

$$\hat{X}^{ML} = \arg\max_{x \in \mathcal{X}} \prod_{a: x \in \mathcal{C}_a} \frac{p_1(Y_a)}{p_0(Y_a)}.$$

If the common length of the roads is $L = 100$ arcs, then the maximum is taken over 3^{100} paths. Instead, we might approximate $\hat{X}^{ML}$ by successively tracking over much smaller pieces, say of length l ($l < 10$) arcs. For example, in the first step we might compute

$$X^l = \arg\max_{x \in \mathcal{X}, |x| = l} \prod_{a: x \in \mathcal{C}_a} \frac{p_1(Y_a)}{p_0(Y_a)}$$

and then continue from the last arc of X^l. However, the arcs near the bottom of X^l are not sufficiently reliable to re-initiate tracking. Instead we keep only the *first* arc of each maximizing segment and discard the rest. (This algorithm is in the same spirit as the one described in [40], where at each iteration all 3^l paths emanating from the seed are inspected, one-third of which are kept and extended.)

Experiments show that for $l = 5$ the algorithm is fast (about twice as fast as the entropy testing rule) but only satisfactory for easy images. The value $l = 7$ is more reliable, but then

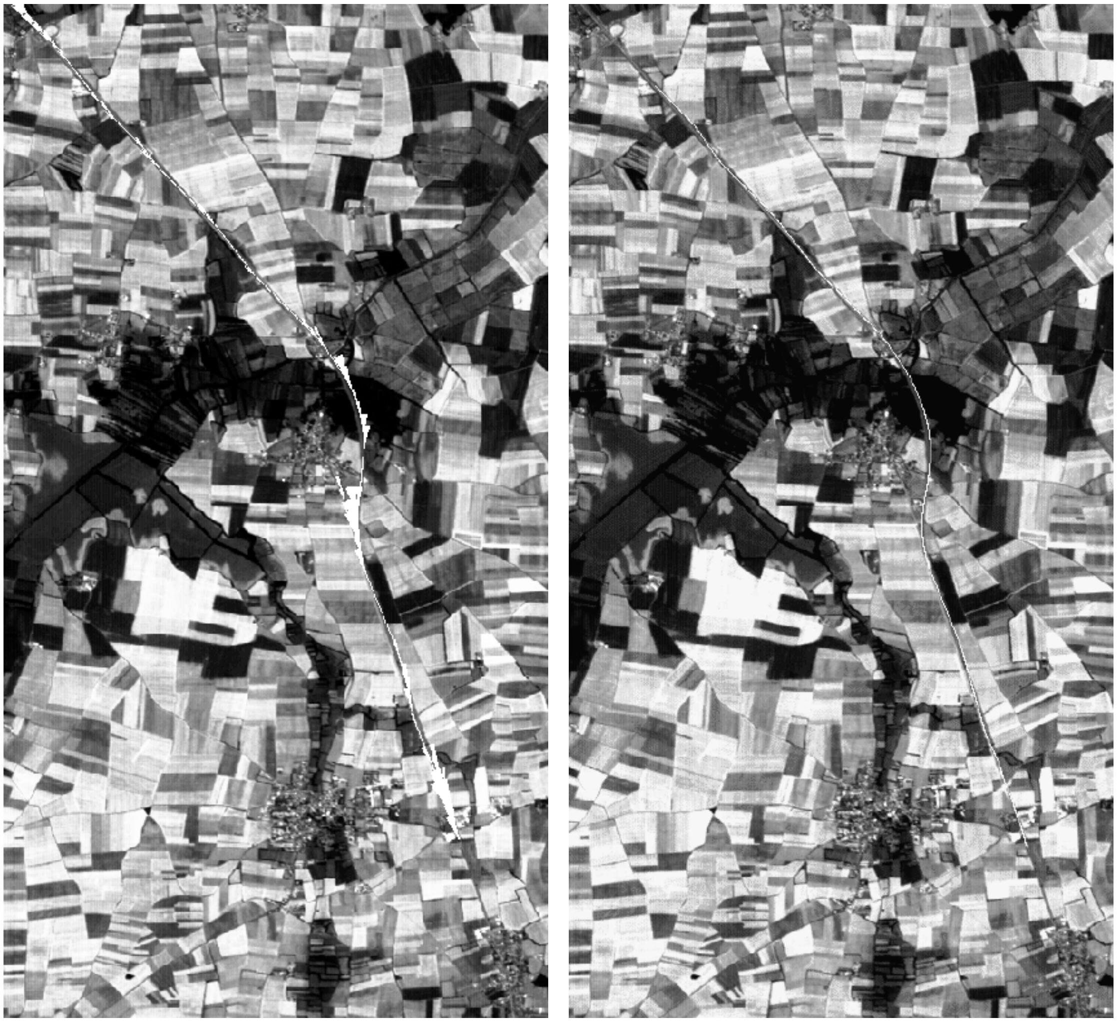


Figure 10: **Left:** Tracking results on a SPOT image from La Rochelle, France. The arcs selected by the algorithm are shown in white. **Right:** Estimate of the road

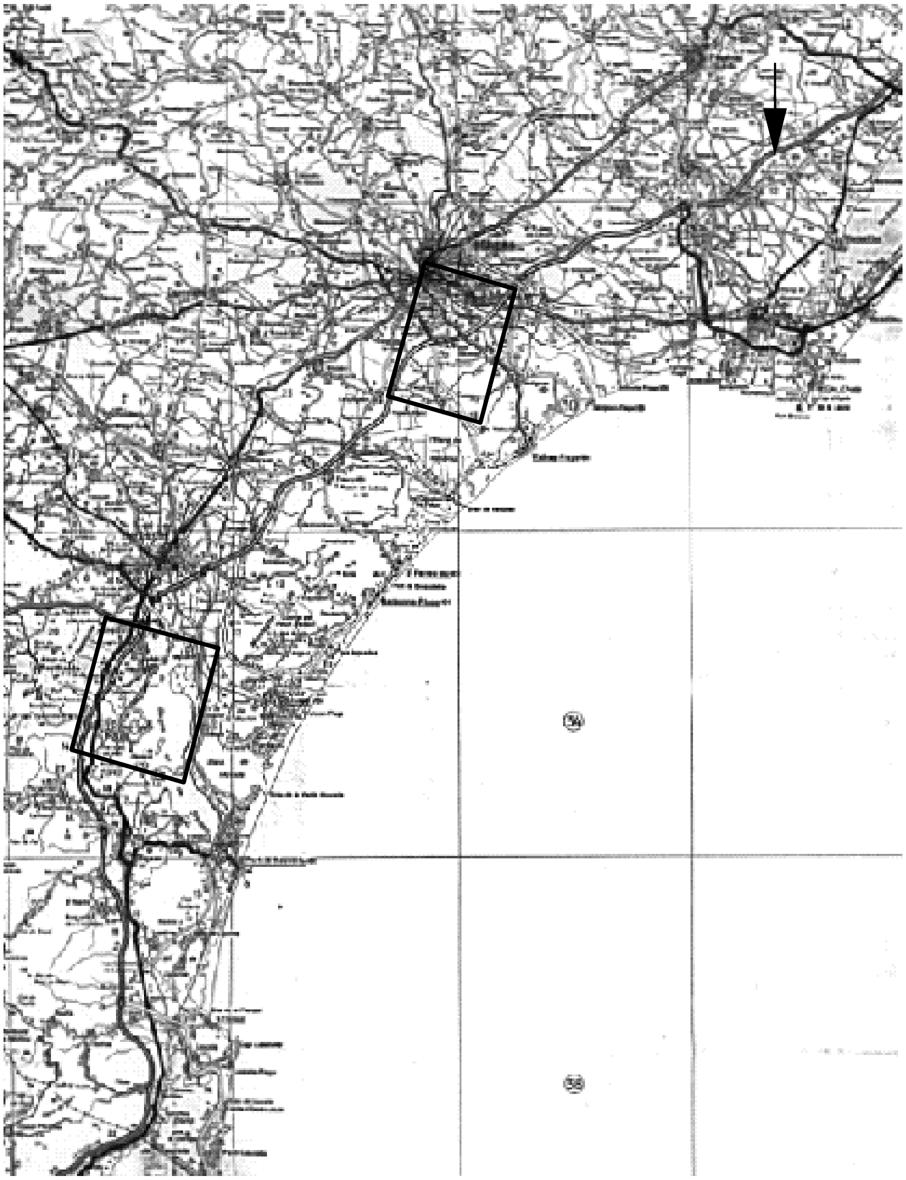


Figure 11: This map covers the Languedoc Rousillon region in southern France; SPOT images of the two rectangles superimposed are shown on the top of Figures 12 and 13.

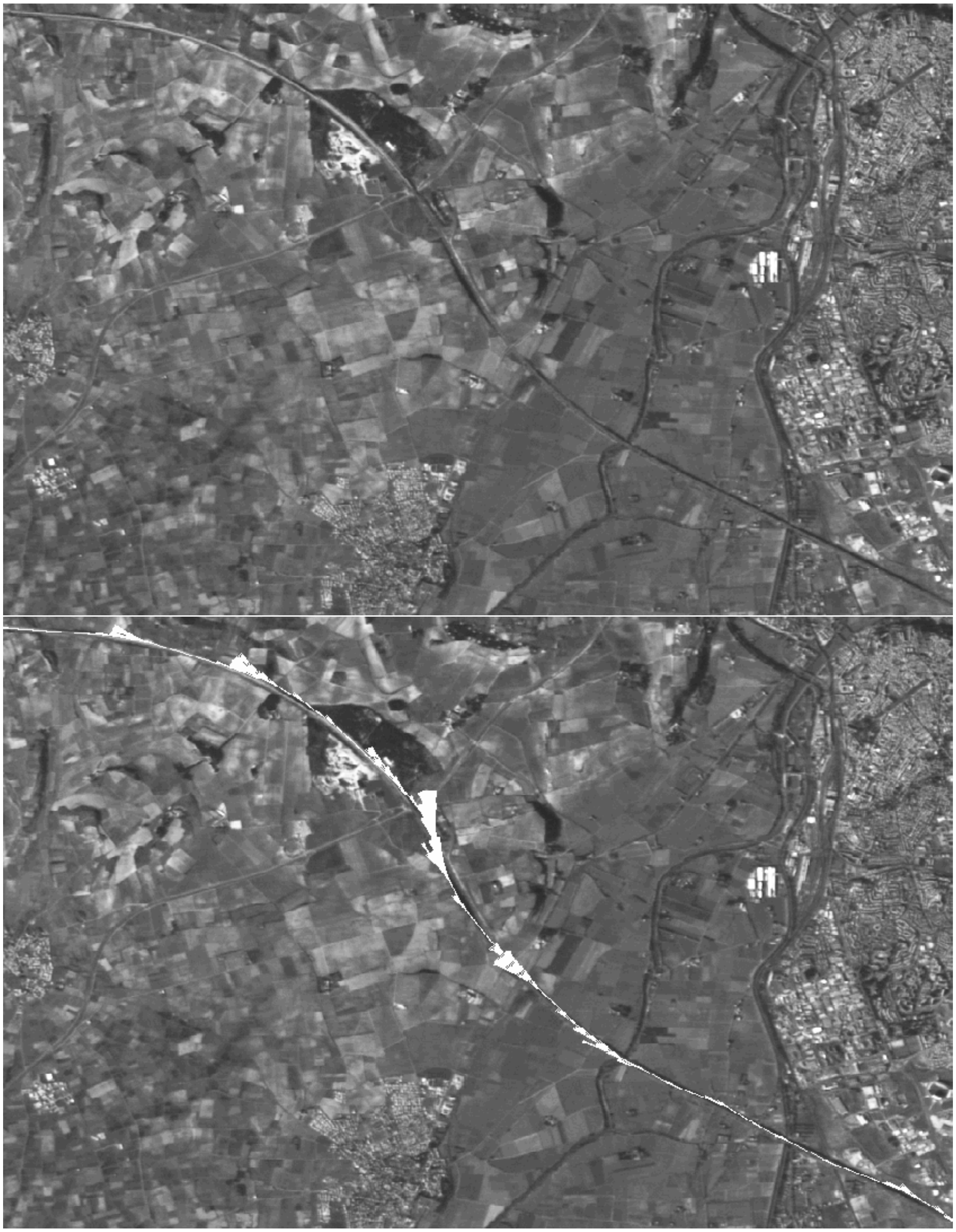


Figure 12: **Top:** Extracted from the SPOT image of the region in Figure 11. Notice that the main road is not systematically brighter or darker than the surrounding area. **Bottom:** Tracking results.

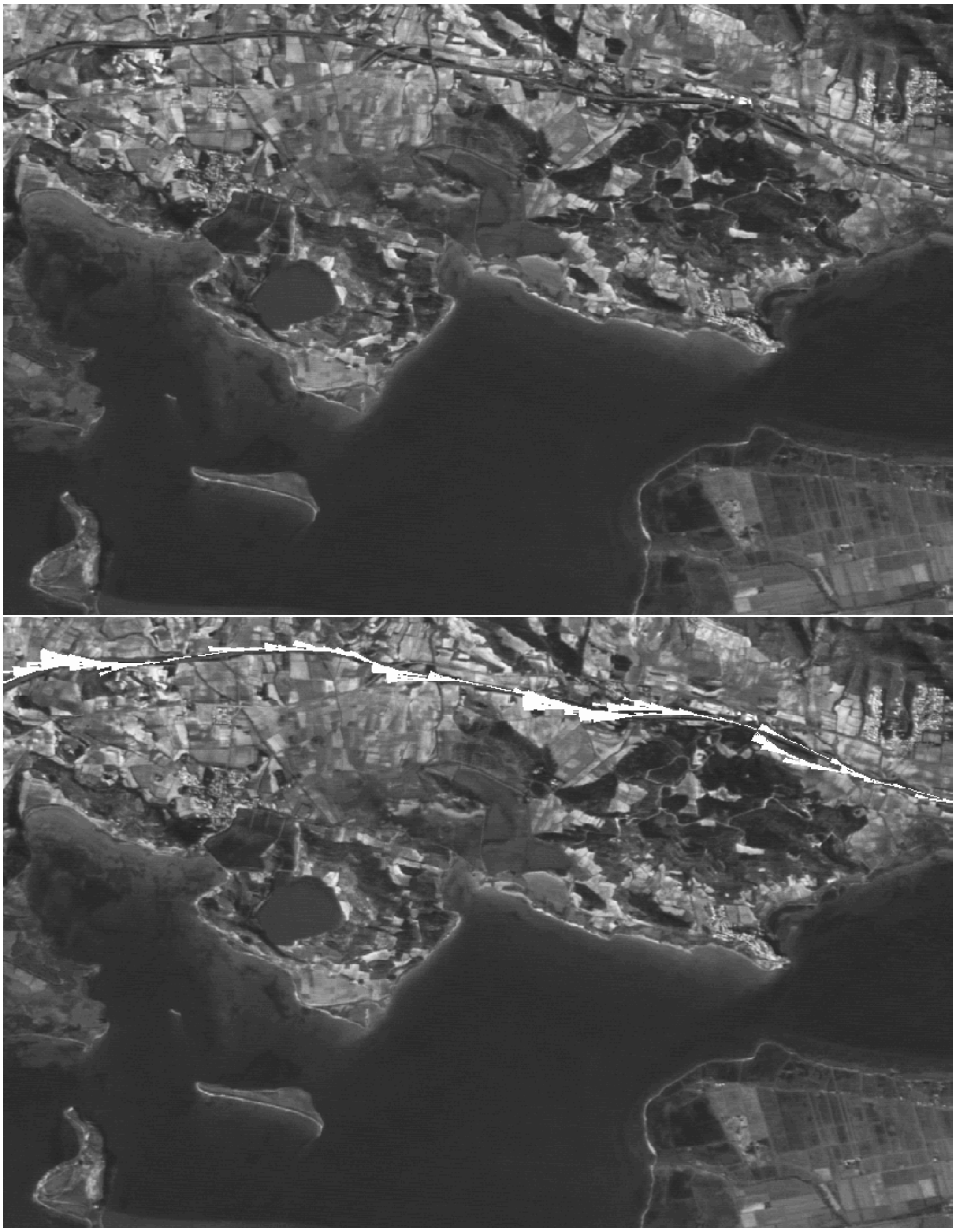


Figure 13: **Top:** Extracted from the SPOT image of the the region in Figure 11. This image is quite difficult: there are competing structures such as small roads, one of which follows the highway, as can be seen on the map. **Bottom:** Tracking results.

the algorithm is about nine times slower than for $l = 5$ and still frequently loses the road. The trade-offs with still higher values of l are then obvious. An interesting situation is the one in Figure 7. Choosing the starting arc as in Figure 8, the modified MLE tracker follows the small road into the village and is permanently lost. As in [40], this might be controlled by demanding that the likelihood of the maximizing path be sufficiently high before continuing.

VI Extensions

There are a number of directions in which this work could be extended within the realm of tracking. (Applying active testing to other visual recognition problems will be considered elsewhere.)

For one thing, we have only considered the problem of tracking based on one given point on the true road. A more realistic scenario might be that *several* points are specified; this would be the case, for example, in an interactive system in which an operator is assisted by the algorithm. (Fully automated extraction - detection and tracking - is another matter, evidently more difficult, and perhaps of *less* practical utility at the moment.) The mathematical problem is to extend the active testing model to incorporate the additional information; clearly the choice of tests will be strongly influenced by the additional restrictions. It is not clear whether or not the ternary tree is still an appropriate representation for the hypotheses; perhaps another graph structure is more suitable. Nor is it clear how to extend the key recursions mentioned in Section IV-F and developed in more detail in the Appendix.

Another issue concerns the prior distribution over roads. We have used only crude prior constraints on width and curvature; after that all piecewise linear paths (of a fixed length) meeting these constraints are considered equally likely road candidates. This is not realistic for major highways since curvature tends to be *maintained*: there are long straight segments and roads tend to continue turning at the same rate. In fact, such constraints may even appear in the construction plans. One way to accommodate these constraints would be to introduce a dependency structure between the angles of the segments for each individual road. From a statistical point of view, in the current model, the angles between successive segments are independent variables each assuming values in $\{0, 1, -1\}$. A more realistic model for the sequence of angles would be a low-order Markov chain, say first-order or second-order. Such a model is currently under investigation ([27]).

Finally, besides major highways, there are other significant linear, deformable structures that could be extracted by our method. One such class is railroads. Another is smaller roads, which are especially difficult to follow because they are so easily confused with other man-made structures, such as furrows, clearings for transmission lines, and field boundaries. There are also natural structures of interest, such as rivers, where spectral information should be useful.

VII Conclusion

We contribute a partial solution to the problem of extracting important 1D structures, especially road networks, from medium resolution satellite imagery. Given a starting point and starting direction, we are able to track highways over considerable distances, for example one hundred kilometers, without manual intervention. The tracker is sufficiently fast and stable to support the work of specialists, such as cartographers, and could be modified to help analyze other linear, deformable structures which are markedly visible in remotely sensed images.

More ambitious problems are to track roads over sizeable geographic regions assuming multiple fixed points, to track smaller roads and other linear structures, and to eliminate seeding altogether and extract complete networks in a fully automated fashion. We hope to address some of these problems within the framework developed in this paper.

That framework is in fact very general; active testing is a variant of "divide and conquer," as popularized in familiar parlor games, and related to current work in active vision. It can provide an expedient alternative or approximation to maximum likelihood (or MAP) estimation, as demonstrated here. More generally, we believe this paradigm offers a promising alternative to existing computational strategies for high level vision, especially shape and object recognition.

Appendix: Mathematical Arguments

A Proof of the Theorem

First, it is easy to see, and intuitively clear, that for any random variables U, V, and event B,

$$H(U,V|B) = H(U|B,V) + H(V|B).$$

Now take $U = X$, $V = Y_a$ and $B = B_k$, the history of the first k tests. Then

$$\begin{aligned} H(X|B_k, Y_a) &= H(X, Y_a|B_k) - H(Y_a|B_k) \\ &= H(Y_a|B_k, X) + H(X|B_k) - H(Y_a|B_k). \end{aligned}$$

Thus, the entropy testing rule chooses the next test by minimizing $H(Y_a|B_k, X) - H(Y_a|B_k)$ as a function of $a \neq \pi_1, ..., \pi_k$. This value will then be $\pi_{k+1}(y_{\pi_1}, ..., y_{\pi_k})$ and the next test will be $Y_{\pi_{k+1}}$.

$$\begin{aligned} H(Y_a|B_k, X) &= \sum_{x \in \mathcal{X}} P(X = x|B_k) H(Y_a|B_k, X = x) \\ &= \sum_{x \in \mathcal{X}} P(X = x|B_k) H(Y_a|X = x) \\ &\quad \text{(since the tests} \\ &\quad \text{are conditionally independent)} \\ &= \sum_{x \in \mathcal{C}_a} P(X = x|B_k) H(Y_a|X = x) \\ &\quad + \sum_{x \notin \mathcal{C}_a} P(X = x|B_k) H(Y_a|X = x) \\ &= H(p_1) P(X \in \mathcal{C}_a|B_k) + H(p_0) P(X \notin \mathcal{C}_a|B_k). \end{aligned}$$

Similarly, for any $y \in \mathcal{Y}$:

$$\begin{aligned} P(Y_a = y|B_k) = &= \sum_{x \in \mathcal{X}} P(Y_a = y|B_k, X = x) P(X = x|B_k) \\ &= \sum_{x \in \mathcal{X}} P(Y_a = y|X = x) P(X = x|B_k) \end{aligned}$$

$$\begin{aligned} &= \sum_{x\in\mathcal{C}_a} P(Y_a = y|X = x)P(X = x|B_k) \\ &\quad + \sum_{x\notin\mathcal{C}_a} P(Y_a = y|X = x)P(X = x|B_k) \\ &= p_1(y)P(X \in \mathcal{C}_a|B_k) + p_0(y)P(X \notin \mathcal{C}_a|B_k). \end{aligned}$$

It follows that $H(X|B_k, Y_a) = \phi(z) + H(X|B_k)$ with $z = P(X \in \mathcal{C}_a|B_k)$, which completes the proof.

B Constructing the Sets $\mathcal{A}_k$

The subsets $\mathcal{A}_k \subset \mathcal{A}, k = 1, 2, ...,$ are defined recursively. Set $\mathcal{A}_0 = \{a_1, a_2, a_3\}$, the three arcs branching from the root of the tree, i.e., the three "children" of the starting arc, π_0. By construction, each subset $\mathcal{A}_k$ will be a tree, meaning that, for any arc $a \in \mathcal{A}_k$, all the arcs along the path from a to the root π_0 are also included in $\mathcal{A}_k$. Clearly this is true for $\mathcal{A}_0$. Suppose now that $\mathcal{A}_k$ is determined for $k \geq 0$; we modify $\mathcal{A}_k$ to obtain $\mathcal{A}_{k+1}$ as follows.

Initialize $\mathcal{A}_{k+1}$ with $\mathcal{A}_k$. If the three children of π_k are not already in $\mathcal{A}_{k+1}$, then add them to $\mathcal{A}_{k+1}$. Compute $z_k(a) = P(X \in \mathcal{C}_a|B_k)$ for each $a \in \mathcal{A}_{k+1}$. Now for each *leaf* (i.e., terminal arc) $a \in \mathcal{A}_{k+1}$ with the property that $z_k(a) > \bar{z}$, add the three children of a to $\mathcal{A}_{k+1}$. Compute z_k for each of these children (which necessarily become leaves of $\mathcal{A}_{k+1}$) and add their children whenever the z_k value of the parent exceeds $\bar{z}$, and so forth. Every time a leaf is created the condition is checked and the tree is enlarged accordingly. When no further leaf satisfies the property the construction is complete; clearly the set so created is again a tree.

There is a special case in which it is possible to *remove* some arcs from $\mathcal{A}_{k+1}$. Suppose $\{b_1, b_2, b_3\}$ all belong to $\mathcal{A}_{k+1}$, are siblings, and are all leaves. Moreover, suppose $z_k(b_i) < \bar{z}, i = 1, 2, 3$. Let b denote the parent arc and suppose $b \notin \{\pi_1, \ldots, \pi_k\}$ and $z_k(b) \leq \bar{z}$. Then $\{b_1, b_2, b_3\}$ can be removed from $\mathcal{A}_{k+1}$.

Using convexity, one can then check that, for each $k \geq 1$:

$$\begin{aligned} \pi_{k+1} &\doteq \arg \min_{a\in\mathcal{A}\setminus\{\pi_1,\ldots,\pi_k\}} \phi(z_k(a)) \\ &= \arg \min_{a\in\mathcal{A}_{k+1}\setminus\{\pi_1,\ldots,\pi_k\}} \phi(z_k(a)). \end{aligned}$$

C Updating Probabilities

Recall that at step $k+1$ arc π_k has been computed based on evaluating $P(X \in \mathcal{C}_a|B_{k-1})$ for all $a \in \mathcal{A}_k$ and we wish to compute π_{k+1}. Thus we need to evaluate $P(X \in \mathcal{C}_a|B_k)$ for $a \in \mathcal{A}_{k+1}$.

First, since the tests are conditionally independent given X,

$$\begin{aligned} P(X \in \mathcal{C}_a|B_k) &= (P(B_k))^{-1} \sum_{x \in \mathcal{C}_a} P(B_{k-1}, Y_{\pi_k} = y_{\pi_k}, X = x) \\ &= (P(B_k))^{-1} \sum_{x \in \mathcal{C}_a} P(B_{k-1}, X = x) P(Y_{\pi_k} = y_{\pi_k}|X = x) \end{aligned}$$

Second, we can express $P(B_k)$ as a function of $P(B_{k-1})$ as follows:

$$\begin{aligned} P(B_k) &= \sum_x P(B_{k-1}, X = x) P(Y_{\pi_k} = y_{\pi_k}|X = x) \\ &= p_1(y_{\pi_k}) \sum_{x \in \mathcal{C}_{\pi_k}} P(B_{k-1}, X = x) + p_0(y_{\pi_k})(P(B_{k-1}) - \sum_{x \in \mathcal{C}_{\pi_k}} P(B_{k-1}, X = x)) \\ &= p_0(y_{\pi_k}) P(B_{k-1})(1 + P(X \in \mathcal{C}_{\pi_k}|B_{k-1})(v(y_{\pi_k}) - 1)) \end{aligned}$$

where

$$v(y_{\pi_k}) = \frac{p_1(y_{\pi_k})}{p_0(y_{\pi_k})}.$$

Case One: $a \in \mathcal{A}_k \cap \mathcal{A}_{k+1}$. We exploit the fact that the arcs reside on a tree to compute $P(X \in \mathcal{C}_a|B_k)$ in terms of $P(X \in \mathcal{C}_a|B_{k-1})$. For any two arcs $a, b \in \mathcal{A}$, there are three possible subcases: (i) $\mathcal{C}_a \cap \mathcal{C}_b = \emptyset$; (ii) $\mathcal{C}_a \subset \mathcal{C}_b$; (iii) $\mathcal{C}_b \subset \mathcal{C}_a$. Now take $b = \pi_k$ and let $\lambda = 1 + P(X \in \mathcal{C}_{\pi_k}|B_{k-1})(v(y_{\pi_k}) - 1)$. In (i),

$$\sum_{x \in \mathcal{C}_a} P(B_{k-1}, X = x) P(Y_{\pi_k} = y_{\pi_k}|X = x) = p_0(y_{\pi_k}) P(B_{k-1}) P(X \in \mathcal{C}_a|B_{k-1})$$

which implies that

$$P(X \in \mathcal{C}_a|B_k) = \lambda^{-1} P(X \in \mathcal{C}_a|B_{k-1}).$$

In case (ii) $P(X \in \mathcal{C}_a|B_k) = \lambda^{-1} v(y_{\pi_k}) P(X \in \mathcal{C}_a|B_{k-1})$ and, in case (iii), $P(X \in \mathcal{C}_a|B_k) = \lambda^{-1}(P(X \in \mathcal{C}_a|B_{k-1}) + P(X \in \mathcal{C}_{\pi_k}|B_{k-1})(v(y_{\pi_k}) - 1))$. Observe that all the quantities appearing on the right-hand side of these expressions are known at this step since π_k necessarily belongs to $\mathcal{A}_k$.

These formulae can be interpreted as follows. Suppose that $v(y_{\pi_k}) > 1$, meaning that the event $Y_{\pi_k} = y_{\pi_k}$ is more likely if π_k lies on X than if not. Then, for all arcs a such that $\mathcal{C}_a \cap \mathcal{C}_{\pi_k} = \emptyset$, $P(X \in \mathcal{C}_a | B_k) \leq P(X \in \mathcal{C}_a | B_{k-1})$. For all the other arcs the inequality is reversed.

Case Two: $a \in \mathcal{A}_{k+1} \setminus \mathcal{A}_k$. The quantity $P(X \in \mathcal{C}_a | B_k)$ is computed as follows. Each such arc a is the child of a leaf b of a tree; see the Appendix. First, we already have $P(X \in \mathcal{C}_b | B_k)$. Second, it's easy to check that $P(B_k | X \in \mathcal{C}_a) = P(B_k | X \in \mathcal{C}_b)$. Now by Bayes formula,

$$\begin{aligned} P(X \in \mathcal{C}_a | B_k) &= \frac{P(X \in \mathcal{C}_a, B_k)}{P(B_k)} \\ &= \frac{P(B_k | X \in \mathcal{C}_b) P(X \in \mathcal{C}_a)}{P(B_k)} \\ &= \frac{P(X \in \mathcal{C}_b | B_k) P(X \in \mathcal{C}_a)}{P(X \in \mathcal{C}_b)}. \end{aligned}$$

and

$$\frac{P(X \in \mathcal{C}_a)}{P(X \in \mathcal{C}_b)} = P(X \in \mathcal{C}_a | X \in \mathcal{C}_b) = \frac{1}{3}.$$

Acknowledgements

The road detection work was carried out at Project Syntim at INRIA-Rocquencourt; we are grateful to Andre Gagalowicz for his valuable support, to Jean Philippe Roze for teaching us how to implement the algorithm efficiently, to Hua Zhao for suggesting how to prune, and to the CNES for providing data and insisting on practical results. Finally, the remarks of the referees and the Associate Editor were most helpful and the paper has benefited accordingly.

References

[1] Arkin, E., Meijer, H., Mitchell, J., Rappaport, D., and Skiena, S., "Decision trees for geometric models," Proc. Ninth ACM Symp. on Computational Geometry, 1993.

[2] Barzohar, M. and Cooper, D.B., "Completely automatic reliable finding of main roads in aerial imagery by using Bayesian methods," Technical Report, Brown-LEMS-118, March, 1993.

[3] Blanc, H. V., "Application du groupement perceptuel à la reconnaissance de routes sur une image satellite SPOT," Technical Report, Université de Montpellier II Sciences et Techniques du Languedoc, 1993.

[4] Breiman, L., Friedman, J., Olshen, R., and Stone, C., *Classification and Regression Trees*, Wadsworth, Belmont, CA., 1984.

[5] Boggess, J. E., "Identification of roads in satellite imagery using artificial neural networks: a contextual approach," Technical Report, Mississippi St. Univ., August, 1993.

[6] Chernoff, H., *Sequential Analysis and Optimal Design*, SIAM, Philadelphia, 1972.

[7] Chou, P.A., "Optimal partitioning for classification and regression trees," IEEE Trans. PAMI, 13, 340-354, 1991.

[8] Daoud, M., Roux, C., and Hillion, A., "Une application de la théorie des graphes à l'extraction automatique des réseaux de communication dans les images SPOT," GRETSI, 699-701, June, 1989.

[9] Duda, R.O. and Hart, P.E., *Pattern Classification and Scene Analysis*, John Wiley, New York, 1973.

[10] Estival, I. and Le Men, H., "Detection of linear networks on satellite images", Proc. Conf. on Pattern Recognition, 856-858, October, 1986.

[11] Fischler, M., Tenenbaum, J. and Wolf, H., "Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique," Computer Graphics and Image Processing, 15, 201-223, 1981.

[12] Garey, M.R., "Optimal binary identification procedures," SIAM J. Appl. Math., 23, 173-186, 1972.

[13] Garey, M.R. and Graham, R.L., “Performance bounds on the splitting algorithm for binary testing,” Acta Informatica, 3, 347-355, 1974.

[14] Garnesson, P., “MESSIE : un système d’analyse de scènes,” Technical Report, Université de Nice-Sophia Antipolis, 1991.

[15] Geman, D. and Jedynak, B., “Detection of roads in SPOT satellite images,” Proc. IGRASS 91, Helsinski, 1991.

[16] Geman, D. and Jedynak, B., “Shape recognition and twenty questions,” Technical Report No. 2155, INRIA-Rocquencourt, November, 1993.

[17] Gittins, J.C., *Multi-armed Bandit Allocation Indices*, John Wiley and Sons, 1989.

[18] Goad, C., “Special purpose automatic programming for three- dimensional model-based vision,”Proc. ARPA Image Understanding Workshop, 94-104, 1983.

[19] Graffigne, C., and Herlin, I. “Modélisation de réseaux pour l’imagerie satellite SPOT,” Technical Report, INRIA, 1989.

[20] Gu, Y.X., Wang, Q.R., and Suen, C.Y., “Application of multilayer decision tree in computer recognition of Chinese characters,” IEEE Trans. PAMI, 5, 83-89, 1983.

[21] Hansen, C. and Henderson, T., ”Towards the automatic generation of recognition strategies,” Second International Conf. on Computer Vision, IEEE, 275-279, 1988.

[22] Haralick, R.M. and Shapiro, L.G., *Computer and Robot Vision*, Vol. I, Addison-Wesley, Reading, 1992.

[23] Hartmann, C., Varshney, P., Mehrotra, K. and Gerberich, C., “Application of information theory to the construction of efficient decision trees,” IEEE Trans. Info. Theory, 28, 565-577, 1982.

[24] Huffman, D.A., “A method for the construction of minimum redundancy codes,” Proc. I.R.E., 40, 1098-1101, 1952.

[25] Hyafil, L. and Rivest, R., "Constructing optimal binary decision trees is NP-complete," Information Processing Letters, 5, 15-17, 1976.

[26] Jedynak B., "Modèles stochastiques et méthodes déterministes pour extraire les routes des images de la Terre vues du ciel" Thèse de Doctorat, Modèlisation Stochastique, Université Paris Sud, 1995.

[27] Keller, I., "Recherche d'un meilleur modèle a priori pour une méthode d'extraction des routes dans une image satellite," Technical Report, Universite de Paris-Sud (Orsay), 1994.

[28] Kumar, P.R., "A survey of some results in stochastic adaptive control," SIAM J. Control and Optim., 23, 329-380, 1985.

[29] Kurzynski, M.W., "The optimal strategy of a tree classifier," Pattern Recognition, 16, 81-87, 1983.

[30] Lawler, E., *Combinatorial Optimization: Networks and Matroids*, Saunders College Publishing, 1976.

[31] Lin, Y.K. and Fu, K.S., "Automatic classification of cervical cells using a binary tree classifier," Pattern Recognition, 16, 69-80, 1983.

[32] Loveland, D.W., "Performance bounds for binary testing with arbitrary weights," Acta Informatica, 22, 101-114, 1985.

[33] Merlet, N. and Zerubia, J., "A curvature-dependent energy function for detecting lines in satellite images," 8th SCIA Conference, May, 1993.

[34] Miyakawa, M., "Criteria for selecting a variable in the construction of efficient decision trees," IEEE Trans. Computers, 38, 130-141, 1989.

[35] Neviata, R., "Locating structures in aerial images," IEEE Trans. PAMI, 476-484, September, 1982.

[36] Quinlan, J.R., "Induction of decision trees," Machine Learning, 1, 81-106, 1986.

[37] Rimey, R. and Brown, C.M., "Control of selective perception using Bayes nets and decision theory," Inter. J. of Computer Vision, 12:2/3, 173-207, 1994.

[38] Roux, M., "Recalage d'images multi-sources. Application au recalage d'une image SPOT et d'une carte," Thèse de Doctorat, Ecole Nationale Supèrieure des Télécommunications, September, 1992.

[39] Sandelius, M., "On an optimal search procedure," Am. Math. Monthly, 68, 133-134, 1961.

[40] Serendero, M. A., "Extraction d'informations symboliques en imagerie SPOT : réseaux de communication et aglomations," Thèse de Doctorat, Université de Nice, 1989.

[41] Spirkovska, L., "Three-dimensional object recognition using similar triangles and decision trees," Pattern Recognition, 26, 727-732, 1993.

[42] Swain, M., "Object recognition from a large database using a decision tree," Proc. of the DARPA Image Understanding Workshop, Morgan Kaufman, 1988.

[43] Swain, M. J. and Sricker, M. A., "Promising directions in active vision," Inter. J. of Computer Vision, 11:2, 109-126, 1993.

[44] Vanderbrug, G. J., "Line detection in satellite imagery," IEEE Trans. Geoscience Electronics, 14, 37-44, January, 1976.

[45] Wang, F. and Newkirk, R. , "A knowledge-based system for highway network extraction," IEEE Trans. Geoscience and Remote Sensing, 26, September, 1988.

[46] Wang, Q.R. and Suen, C.Y., "Analysis and design of a decision tree based on entropy reduction and its application to large character set recognition," IEEE Trans. PAMI, 6, 406-417, 1984.

[47] Watanabe, S., "Pattern recognition as a quest for minimum entropy," Pattern Recognition, 13, 381-387, 1981.

[48] Zimmerman, S., "An optimal search procedure," Am. Math. Monthly, 66, 690-693, 1959.