Proceedings of the IEEE Workshop on Statistical and Computational Theories of Vision, Fort Collins, CO, June, 1999. (Published on Web).

Visual Search: Fundamental Bounds, Order Parameters, and Phase Transitions

A. L. Yuille and James M. Coughlan [1]

Abstract

This paper formulates the problem of visual search as Bayesian inference and defines a Bayesian ensemble of problem instances. In particular, we address the problem of the detection of visual contours in noise/clutter by optimizing a global criterion which combines local intensity and geometry information. We specialize to the specific task of tracking a contour – such as a road in an aerial image. We determine order parameters, which depend on statistical properties of the target and domain, that characterize the difficulty of the task independently of the algorithm employed to detect the target. For the road tracking problem, we show that there is a phase transition at a critical value of the order parameter – above this phase transition it is impossible to detect the target by any algorithm. We consider the case where there is a low-level *generic* model and a more specific *high-level* model of the target. We demonstrate that in certain regimes (of the order parameters) both models are effective at detecting the target. However, at a critical value of the order parameters there is a phase transition and it becomes effectively impossible to detect the target unless high-level target specific knowledge is used. At another phase transition, the target will become undetectable by any model. These phase transitions determine different regimes within which different search strategies will be effective. These results have implications for bottom-up and top-down theories of vision. In related work [16],[3], we derive closely related order parameters which determine the complexity of search and the accuracy of the solution using the A^* search strategy.

1. Smith-Kettlewell Eye Research Institute, 2318 Fillmore Street, San Francisco, CA 94115, USA. Tel. (415) 345-2144. Fax. (415) 345-8455. Email yuille@attila.ski.org, coughlan@ski.org.

1 Introduction

Which vision tasks can be solved by artificial vision systems? If they can be solved, how fast can vision algorithms solve them? And how accurately? What properties of the visual task and environment determine the ease of the problem and the speed with which it can be solved?

These issues become of growing importance with the increased sophistication of vision tasks and algorithms. Of course, the performance of biological vision systems give us some idea of what problems can be solved by artificial vision. But biological systems have evolved to perform specific visual tasks in particular environments, which often differ from the tasks/domains for which artificial vision systems are required.

In this paper, we start developing a theory for addressing these issues. For concreteness, we are specifically concerned with the problem of detecting a road from aerial images though our approach, though not necessarily our specific results, should be valid for a far broader range of vision problems.

The complexity of visual search algorithms has been previously addressed in a number of papers, see for example [8] and papers cited therein. Such approaches have assumed that elementary features, like edges, have been directly extracted from the image as input to the search algorithm. By contrast, our work starts with real images as input *and* gives fundamental limits on whether the problem can be solved *independent* of the algorithm used. Results on convergence rates of specific algorithms can then be attained for the regimes in which the problem is solvable [16],[3].

Our theoretical results depend on techniques, such as Sanov's theorem, from the theory of types in information theory [5]. This theory may not be very familiar to a computer vision audience so we introduce it with examples which also illustrate the main concepts of our work, such as order parameters and phase transitions. These examples are motivated by psychophysical experiments on texture discrimination.

Underlying our approach is the assumption that vision problems should be modelled as probabilistic inference using bayesian probability theory [10]. This assumes, for example, that although we cannot say for certain whether there is an intensity edge at a given image position we can nevertheless give a *probability* that an edge is present there (using statistics of edge detector responses in the domain, see section (3)). Moreover, we cannot specify in advance the precise shape of the target curve but we can put a probability distribution on possible target shapes and this distribution can be learnt from training data (see, for example, [22]).

The probabilistic approach seems forced on us by the complicated nature of real world vision problems. It does, however, mean that theories of computational complexity which deal with worst case estimates (such as NP-completeness results) are not directly relevant because the "worst cases" may correspond to extremely unlikely situations. Instead it seems preferable to prove results for *typical* situations: those which occur with reasonable probability [5]. Pearl [12] describes proofs of convergence of this type for a specific class of problem (see Chp. 5 [12]).

This "typicality" approach requires a probability distribution for the class of problems that will arise. Fortunately these probabilities have already been specified in our bayesian formulation. We merely have to trust them! This does mean, however, that *the probability*

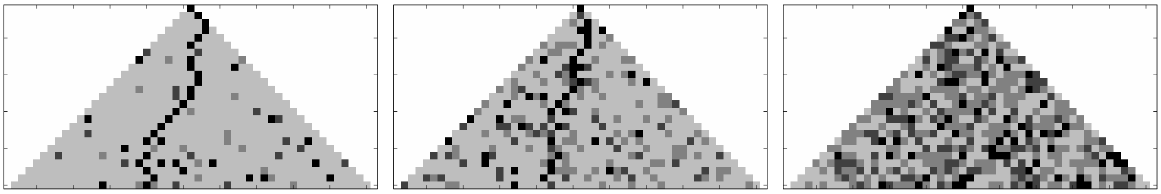


Figure 1: The difficulty of detecting the target path in clutter depends, by our theory, on the order parameter K. Left, an easy detection task with $K = 0.8647$. Middle, a harder detection task with $K = 0.2105$. Right, an impossible task with $K = -0.7272$.

distributions used in the bayesian models should be truly representative of the domain and should be learnt by statistical analysis of the domain. (See, Zhu, Wu, and Mumford's Minimax Entropy learning theory [18],[19],[21]).)

To make these issues more concrete we consider two specific examples [7],[6], which involve the detection and tracking of contours. Both examples are formulated in Bayesian terms and, for both, the speed of convergence is important (the authors report good empirical convergence rates). But for what class of problem will their algorithms succeed? What characteristics of the problem determine this convergence? Would the convergence rates, and the accuracy of the results, be the same if, for example, the shape of the contours was changed? Or if the contour was partially hidden by clutter? If not, how would they vary? This paper, and our related work in [16],[3], helps provide answers to these questions. Essentially the detectability depends on an order parameter K which, as we will show, depends on the properties of the target and background. (See figure (1)).

Our proofs rely on three basic elements: (i) Sanov's theorem which shows that the probability of rare events decreases *exponentially* with the length of the subpath, (ii) an *onion peeling* strategy which allows us to recursively analyze the search tree, and (iii) the use of standard techniques for summing, and bounding, exponential series generated by (i). See figure (7) for an illustration of this search task.

Road and contour tracking are just two of many possible examples where it is important to understand convergence rates of algorithms and determine what properties of the domain make the problem easy or hard. They are important domains and hence are a good place to start examining the complexity of visual search. We will therefore concentrate on the problem of detecting a target curve where there is local (ambiguous) evidence about the presence of edges and global geometric properties of the target. We expect, however, that the approach we develop will be applicable to other computer vision problems.

In section (2), we introduce the theory of types and, in particular, Sanov's theorem. We illustrate it by deriving order parameters for texture discrimination tasks and include one example of a phase transition. Section (3) describes the models of road tracking and snakes. In section (4) we explore the fundamental limits of road detection by using the theory of types to derive order parameters and phase transitions for this problem. Section (5) discusses the use of a hierarchy of generic and high-level models of the target. In section (6), we discuss ways to extend our results. Finally, section (7) summarizes the paper.

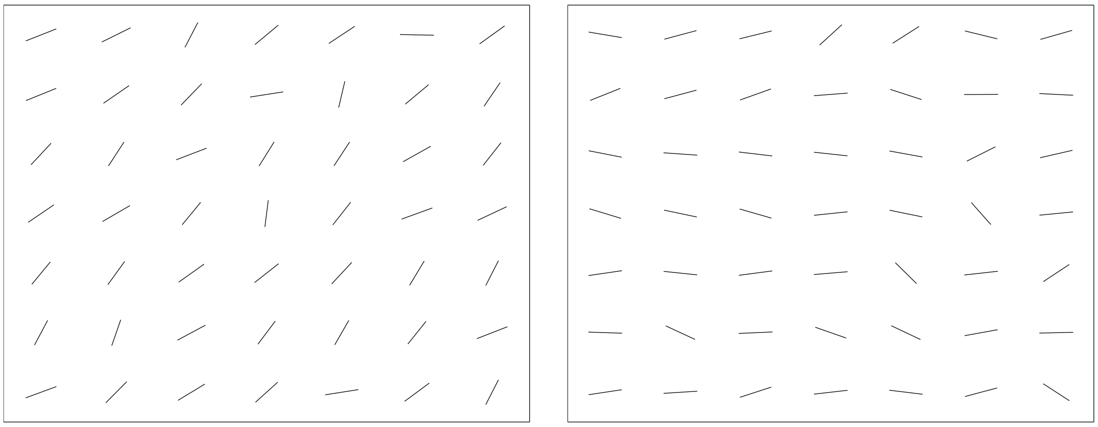


Figure 2: The first texture task: The input is a texture sample. The task is to determine if it came from A, like the texture sample on the left, or from B, like the texture sample on the right.

2 The Theory of Types

This section introduces the basic concepts and mathematical machinery that we will need to prove our results. This material is not very familiar to computer vision workers so we will introduce it by means of examples which bring out the key features of our paper. These examples are motivated by psychophysical experiments for discriminating between textures.

More specifically, we consider three related visual tasks which require distinguishing between two textures A and B. Both textures consist of N edgelets of the same length which are spaced evenly on a lattice. For each texture, the angles of the edgelets are independently identically distributed by $P_A(\theta)$ and $P_B(\theta)$ respectively. The set of possible angles θ is quantized to take M possible values (e.g. we could set $M = 24$ corresponding to a quantization of angles at 15 degrees). These quantized values $a_1, ..., a_M$ are called the *alphabet* of the problem. We wish to quantify how the difficulties of visual tasks depend on N and the distributions $P_A(.)$ and $P_B(.)$.

Our three visual tasks have different inputs. The input to the first is a texture sample and the task is to determine whether the texture sample is from A or B, see figure (2). The input to the second task is two texture samples, one each from A and B, and the task is to correctly label the samples (this is called "two-alternative forced choice"). The third task consists of many texture samples from B and a single texture sample from A – the goal is to detect the *target* A among the many *distractors* from B.

Each texture sample can be characterized by the set $\vec{\theta} = (\theta_1, \theta_2, ..., \theta_N)$ of the angles of its edgelets. The optimal tests for our three tasks will depend on the *log-likelihood ratio*[1] (see the Neyman-Pearson lemma [5]):

[1]This can be thought of as the maximum likelihood test between two hypotheses which are equally likely a priori.

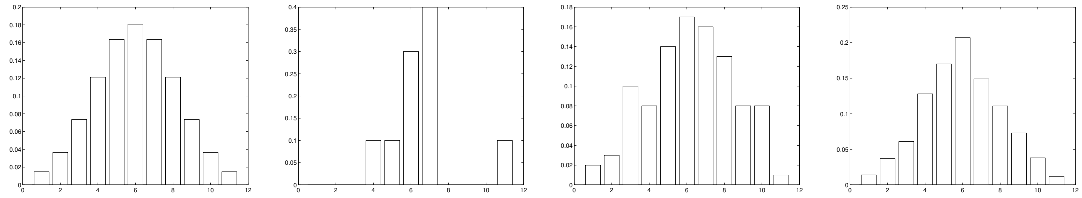


Figure 3: Samples from an underlying distributions. Left to Right, the original distribution, followed by histograms from 10, 100, and 1000 samples from the original.

$$\log\{\frac{P_A(\theta_1, ...\theta_N)}{P_B(\theta_1, ...\theta_N)}\} = \log\{\prod_{i=1}^{N} \frac{P_A(\theta_i)}{P_B(\theta_i)}\} = \sum_{i=1}^{N} \log\{\frac{P_A(\theta_i)}{P_B(\theta_i)}\}. \quad (1)$$

The larger the log-likelihood ratio then the more probable that the texture sample $\vec{\theta} = (\theta_1, \theta_2, ..., \theta_N)$ came from A rather than B (if the log-likelihood ratio is zero then both A and B are equally probable). We can obtain measures of the difficulty of the problem by evaluating the *expected value* of the log-likelihood ratio, see equation (1) when the texture samples are generated by $P_A(\theta_1, ...\theta_N)$ or $P_B(\theta_1, ...\theta_N)$. This gives:

$$\frac{1}{N} < \log\{\frac{P_A(\theta_1, ...\theta_N)}{P_B(\theta_1, ...\theta_N)}\} >_{P_A} = \sum_{\theta} P_A(\theta) \log\{\frac{P_A(\theta)}{P_B(\theta)}\} = D(P_A||P_B),$$
$$< \frac{1}{N} \log\{\frac{P_A(\theta_1, ...\theta_N)}{P_B(\theta_1, ...\theta_N)}\} >_{P_B} = \sum_{\theta} P_B(\theta) \log\{\frac{P_A(\theta)}{P_B(\theta)}\} = -D(P_B||P_A), \quad (2)$$

where the *Kullback-Leibler* divergence $D(P_A||P_B)$ is defined to be $\sum_\theta P_A(\theta)\log(P_A(\theta)/P_B(\theta))$. Observe that this definition is not symmetric – in general $D(P_A||P_B) \neq D(P_B||P_A)$ – and so the Kullback-Leibler divergence is *not* a distance metric between probability distributions. However it does have many properties of a distance metric and, in fact, approximates the squared distance between two distributions provided the distributions are very similar. In particular, it is non-negative definite so that $D(P_A||P_B) \geq 0$ with equality only if $P_A(\theta) = P_B(\theta), \ \forall\ \theta$.

Equation (2) shows that the expected value of the log-likelihood ratio differs by $N\{D(P_A||P_B) + D(P_B||P_A)\}$ depending on whether the texture sample came from A or B. The symmetric Kullback-Leibler divergence,$\{D(P_A||P_B) + D(P_B||P_A)\}$, therefore appears as a crude measure for the difficulty of distinguishing texture samples of A and B. *But this analysis completely ignores the fluctuations of the texture samples.* We need to consider the probabilities that a random texture sample from B has higher log-likelihood ratio than a texture sample from A. This requires us to put probabilistic bounds on the probabilities of unlikely events. This can be done by adapting the theory of types, see [5].

Any texture sample $\vec{\theta} = (\theta_1, \theta_2, ..., \theta_N)$ determines an empirical histogram, or *type*, $\vec{\phi}(\vec{\theta})$ which is an J-dimensional vector whose components $\phi_1, ..., \phi_J$ are the proportions of responses ϕ_i which take values $a_1, ..., a_J$. (i.e. $\phi_\mu = (1/N)\sum_{i=1}^{N} \delta_{\theta_i, a_\mu}$). The key point is that *all the relevant properties of the texture will depend only on its type* (in view of the i.i.d. assumption). This includes the result of the log-likelihood test, see equation (1), which we can re-express as:

$$\log\{\frac{P_A(\theta_1,....\theta_N)}{P_B(\theta_1,....\theta_N)}\} = \sum_{\mu=1}^{J}(N\phi_\mu)\log\{P_A(a_\mu)/P_B(a_\mu)\}. \tag{3}$$

It is important to observe that this is simply the dot-product, $N\vec{\phi}\cdot\vec{\alpha}$, of the type $\vec{\phi}$ with a weight vector $\vec{\alpha}$ (for the equation above, $\vec{\alpha}$ has components $\alpha_\mu = \log\{P_A(a_\mu)/P_B(a_\mu)\}$). Most of the quantities that we are concerned with, such as the fundamental bounds, will depend on dot products of this form. The theory of types proceeds by putting probabilistic bounds on types which can then be used to put probability bounds on the dot products. For the results which follow it is convenient to divide out by the size factor N. We therefore consider the average of the log-likelihood with respect to the texture samples – i.e. $(1/N)\sum_{i=1}^{N}\log P_A(\theta_i)/P_B(\theta_i)$.

There are five key lemmas that we will use about types [5]:

Lemma 1. The total number of types $\leq (N+1)^J$. (This is a very generous upper bound which occurs because each component of the type vector $\vec{\phi}$ can take at most $N+1$ possible values).

Lemma 2. The probability $Q^N(\vec{\theta})$ for any texture $\vec{\theta}$ drawn i.i.d. from $Q(\theta)$ depends only on the *entropy* $H(\vec{\phi}(\vec{\theta})) = -\sum_\mu \phi_\mu \log\phi_\mu$ of the type of the sequence and the Kullback-Leibler distance $D(\vec{\phi}(\vec{\theta})||Q)$ between the type and the distribution Q, and is given by:

$$Q^N(\vec{\theta}) = F(\vec{\phi}(\vec{\theta})) = 2^{-N\{H(\vec{\phi}(\vec{\theta}))+D(\vec{\phi}(\vec{\theta})||Q)\}}. \tag{4}$$

(The probability of the sequence can be expressed as $\prod_{\mu=1}^{M} Q_\mu^{N\phi_\mu} = 2^{N\sum_{\mu=1}^{M}\phi_\mu\log Q_\mu}$ and we use $H(\vec{\phi}) + D(\vec{\phi}||Q) = -\sum_{\mu=1}^{M}\phi_\mu\log Q_\mu$ to obtain the result.)

Lemma 3. The probability $P(\vec{\phi})$ that a sequence has type $\vec{\phi}$ is given by:

$$P(\vec{\phi}) = F(\vec{\phi})\left|T(\vec{\phi})\right|, \tag{5}$$

where $\left|T(\vec{\phi})\right| = \sum_{\vec{\theta}:\vec{\phi}(\vec{\theta})=\vec{\phi}} 1$ is the number of distinct sequences with type $\vec{\phi}$. (This follows from $P(\vec{\phi}) = \sum_{\vec{\theta}}\delta_{\vec{\phi},\vec{\phi}(\vec{\theta})}Q^N(\vec{\theta})$ and substituting equation (4)).

Lemma 4. We can bound the size of each type class by [5]:

$$\frac{2^{NH(\vec{\phi})}}{(N+1)^M} \leq \left|T(\vec{\phi})\right| \leq 2^{NH(\vec{\phi})}. \tag{6}$$

(Not surprisingly, the larger the entropy $H(\vec{\phi})$ the bigger the type class.)

Lemma 5. We can put a bound on $P(\vec{\phi})$ by combining Lemmas 2, 3, and 4. This gives:

$$\frac{2^{-ND(\vec{\phi}||Q)}}{(N+1)^Q} \leq P(\vec{\phi}) \leq 2^{-ND(\vec{\phi}||Q)}. \tag{7}$$

From these basic lemmas we can derive the main result we need. We are particularly interested in putting bounds on the probability that a type $\vec{\phi}$ lies within a certain set of

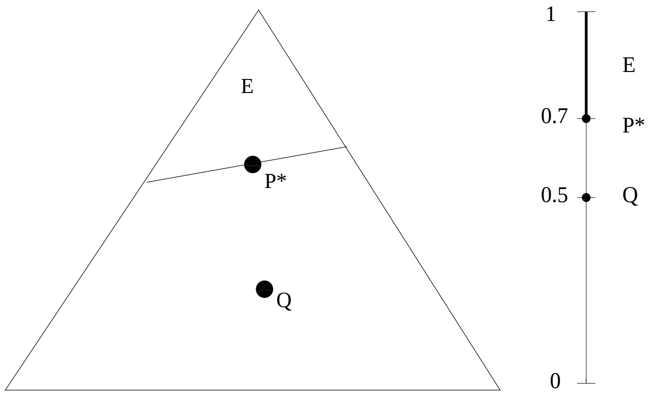


Figure 4: Left, Sanov's theorem. The triangle represents the set of probability distributions. Q is the distribution which generates the samples. Sanov's theorem states that the probability that a type, or normalized empirical histogram – see figure (3) – lies within the subset E is chiefly determined by the distribution P^* in E which is closest to Q. Right, Sanov's theorem for the coin tossing experiment. The set of probabilities is one-dimensional and is labelled by the probability $p(head)$ of tossing a head. The unbiased distribution Q is at the centre, with $Q(head) = 1/2$, and the closest element of the set E is P^* such that $P^*(head) = 0.7$.

types E. For example, for our texture tasks we define the *reward* of a type $\vec{\phi}$ to be $\vec{\phi} \cdot \vec{\alpha}$. It will then be important to bound the probability that texture samples from B have rewards above a specific threshold T. To do this, we define $E_T = \{\vec{\phi} : \vec{\phi} \cdot \vec{\alpha} \geq T\}$ and ask for the probability, $Pr(\vec{\phi} \epsilon E_T)$, that the type of a texture sample from B will lie within E_T.

The main result is called Sanov's theorem:

Sanov's Theorem. *Let $\theta_1, \theta_2, ..., \theta_N$ be i.i.d. samples from a distribution $Q(\theta)$ with alphabet size J and E be any closed set of probability distributions. Let $Pr(\vec{\phi} \in E)$ be the probability that the type of a sample sequence lies in the set E. Then:*

$$\frac{2^{-ND(\vec{\phi}^*||Q)}}{(N+1)^J} \leq Pr(\vec{\phi} \in E) \leq (N+1)^J 2^{-ND(\vec{\phi}^*||Q)}, \tag{8}$$

where $\vec{\phi}^* = \arg\min_{\vec{\phi} \in E} D(\vec{\phi}||Q)$ is the distribution in E that is closest to Q in terms of Kullback-Leibler divergence.

Proof. *It is straightforward to see that $\max_{\vec{\phi} \epsilon E} P(\vec{\phi}) \leq Pr(\vec{\phi} \epsilon E) \leq |E| \max_{\vec{\phi} \epsilon E} P(\vec{\phi})$. From Lemma 5, we can put upper and lower bounds on $\max_{\vec{\phi} \epsilon E} P(\vec{\phi})$ in terms of $\vec{\phi}^* = \arg\min_{\vec{\phi} \epsilon E} D(\vec{\phi}||Q)$. This gives the result using Lemma 1 to put $1 \leq |E| \leq (N+1)^J$.*

Sanov's theorem can be illustrated by a simple coin tossing example, see figure (4). Suppose we have a fair coin and want to estimate the probability of observing more than 700 heads in 1000 tosses. Then set E is the set of probability distributions for which $P(head) \geq 0.7$ ($P(head) + P(tails) = 1$). The distribution generating the samples is $Q(head) = Q(tails) = 1/2$ because the coin is fair. The distribution in E closest to Q is $P^*(head) = 0.7, P^*(tails) = 0.3$. We calculate $D(P^*||Q) = 0.119$. Substituting into

Sanov's theorem, setting the alphabet size $J = 2$, we calculate that the probability of more than 700 heads in 1000 tosses is less than $2^{-119} \times (1001)^2 \leq 2^{-99}$.

In this paper, we will only be concerned with sets E which involve the rewards of types. These sets will therefore be defined by linear constraints on the types (such as $\vec{\phi} \cdot \vec{\alpha} \geq T$) and will therefore allow us to derive results which will not be true for arbitrary sets E. We will often, however, be concerned with the probabilities that the rewards of samples from one distribution are greater than those from a second. It is straightforward to generalize Sanov's theorem to deal with such cases.

We now illustrate the power of these results by considering our three texture tasks. The input to the first task is a single texture sample and we must decide whether it comes from A or B (both are equally likely a priori). The Neyman-Pearson lemma says that the optimal test is to compare the loglikelihood ratio to a threshold T (choices of T will be discussed later). The texture is classified to be A provided the log-likelihood is greater than T and is set to B otherwise.

The reward for a texture sample generated by A is given by $\vec{\phi}^A \cdot \vec{\alpha}$, where $\alpha_\mu = \log P_A(a_\mu)/P_B(a_\mu)$.

Theorem 1. *The probabilities that the loglikelihoods of texture samples with N elements from B or A are above, or below, the threshold T are bounded above and below as follows:*

$$(N+1)^{-J} 2^{-ND(\phi_T||P_A)} \leq Pr\{\vec{\phi}^A \cdot \vec{\alpha} < T\} \leq (N+1)^J 2^{-ND(\vec{\phi}_T||P_A)}, \tag{9}$$

$$(N+1)^{-J} 2^{-ND(\phi_T||P_B)} \leq Pr\{\vec{\phi}^B \cdot \vec{\alpha} > T\} \leq (N+1)^J 2^{-ND(\vec{\phi}_T||P_B)}, \tag{10}$$

where $\phi_T(\theta) = P_A(\theta)^{1-\lambda(T)} P_B(\theta)^{\lambda(T)}/Z(T)$, and $\lambda(T) \in [0,1]$ is a scalar which depends on the threshold T, and $Z(T)$ is a normalization factor. The value of $\lambda(T)$ is determined by the constraint $\vec{\phi}_T \cdot \vec{\alpha} = T$.

Proof. *We apply Sanov's theorem setting $E_A = \{\vec{\phi}^A : \vec{\phi}^A \cdot \vec{\alpha} \leq T\}$ and $E_B = \{\vec{\phi}^B : \vec{\phi}^B \cdot \vec{\alpha} \geq T\}$. Determining the closest distribution $\phi_T \in E_A$ to P_A reduces to constrained minimization using Lagrange multipliers (ν and μ):*

$$\sum_\theta \phi_T(\theta) \log \frac{\phi_T(\theta)}{P_A(\theta)} + \nu\{\sum_\theta \theta_T(\theta) - 1\} + \mu\{\vec{\phi}_T \cdot \vec{\alpha} - T\}. \tag{11}$$

This can be solved, recalling that $\alpha(\theta) = \log\{P_A(\theta)/P_B(\theta)\}$, to give $\phi_T(\theta) = P_A^{1-\lambda(T)}(\theta) P_B^{\lambda(T)}(\theta)/Z(T)$ with $\lambda(T)$ being determined by the constraint $\vec{\phi}_T \cdot \vec{\alpha} = T$. A similar argument applies to P_B and same constraint, $\vec{\phi}_T \cdot \vec{\alpha} = T$, applies to both cases. Hence results.

The Neyman-Pearson lemma does not specify the threshold T. There are two important natural choices. The first is based on minimizing the *asymptotic error rate* of the classification – the rate of *falsely classifying texture samples from A as coming from B and vice versa* (i.e. we give equal weight to the false positives and false negatives),

Corollary 1. *The asymptotic error rate is minimized by setting $T = 0$. The error rate in this case is determined by the Chernoff information $C(P_A, P_B)$, where the Chernoff information is defined by the Kullback-Leibler divergence to the distribution ϕ_T^c halfway between P_A and P_B. More precisely, $C(P_A, P_B) = D(\phi_T^c||P_A) = D(\phi_T^c||P_B)$ for the unique distribution ϕ_T^c, of form $\phi_T(\theta) = P_A(\theta)^{1-\lambda(T)} P_B(\theta)^{\lambda(T)}/Z(T)$, which satisfies this constraint.*

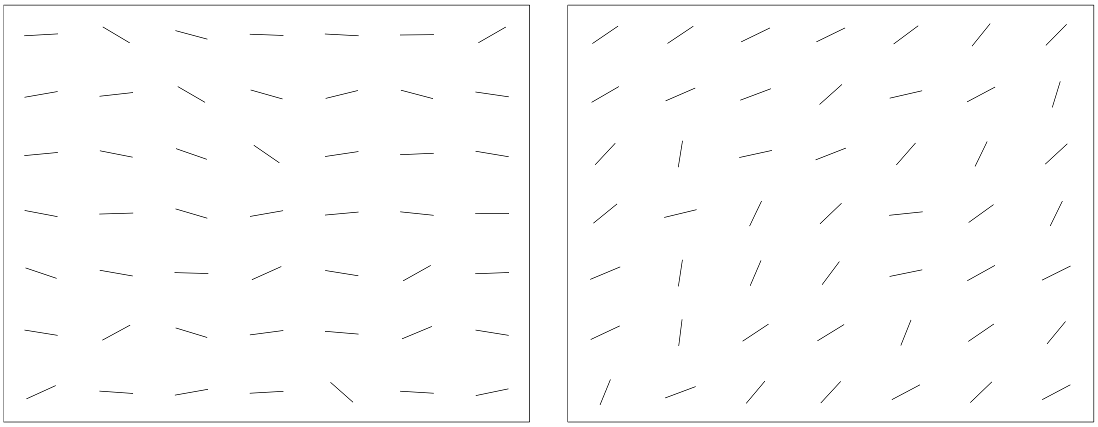


Figure 5: The second texture task: two alternative forced choice. Which texture sample is from A and which one from B?

Proof. *The error rates fall off as* $2^{-ND(\phi_T||P_A)}$ *and* $2^{-ND(\phi_T||P_B)}$. ϕ_T *is of form* $P_A^{1-\lambda(T)}(\theta)P_B^{\lambda(T)}/Z(T)$ *and has only one degree of freedom. As* $\lambda(T)$ *increases* $D(\phi_T||P_A)$ *decreases and* $D(\phi_T||P_B)$ *increases. Therefore there is a unique minimum error rate for* T^c *such that* $D(\phi_T^c||P_A) = D(\phi_T^c||P_B)$, *which defines the Chernoff information. Observe that* $\sum_\theta \phi_T^c(\theta) \log P_A(\theta)/P_B(\theta) = 0$.

The second natural choice of T corresponds to estimating the probability that the rewards of texture samples from A are less than the expected rewards for texture samples from B (or vice versa). This gives:

Corollary 2. *The probability that texture samples from* A *have lower rewards than the average reward for* B *texture samples is less than* $(N+1)^J 2^{-ND(P_B||P_A)}$ *and greater than* $(N+1)^{-J} 2^{-ND(P_B||P_A)}$.

Proof. *We set the threshold* T *to be the average reward,* $-D(P_B||P_A)$, *of texture samples generated by* B. *The result of Theorem 1 shows that we must set* $\phi_T = P_B$ *to satisfy the optimization constraint.*

We now apply Theorem 1 and Corollary 1,2 to determine order parameters which solve the first texture case. If we use a decision rule based on the minimum error rate criterion then the order parameter is the Chernoff information $C(P_A, P_B)$. The difficulty of performing this task depends only on this single number. As this number decreases the task becomes increasingly harder. But there is no critical point at which the task becomes impossible (because Chernoff information is always non-negative). So phase transitions do not occur for this task (as we will see, phase transitions will occur when we consider target detection tasks). Similar results occur if we use alternative choices of T. We will obtain different order parameters, such as $D(P_B||P_A)$ given by Corollary 2, but there will be no critical values and no phase transition.

The second texture case has two texture samples as input (one each from A and B) and the task is to classify them correctly. The best decision rule to to classify the texture sample with higher log-likelihood ratio to be A and the other to be B. This does not involve a choice of threshold. Therefore for this task we only care about the chances that a texture sample from A will have lower reward than a texture sample from B. Our main result is:

Theorem 2. *The probability that a texture sample from A has lower reward than a texture sample from B is bounded below by $(N+1)^{-J^2} 2^{-2NB(P_A,P_B)}$ and above by $(N+1)^{J^2} 2^{-2NB(P_A,P_B)}$, where $B(P_A,P_B) = -\log\{\sum_\mu P_B^{1/2}(a_\mu) P_A^{1/2}(a_\mu)\}$. ($N$ is the number of elements in each texture sample.)*

Proof. *This is a generalization of Sanov's theorem to the case where we have two probability distributions and two types (so the alphabet size becomes J^2). We define $E = \{(\vec{\phi}^A, \vec{\phi}^B) : \vec{\phi}^B \cdot \vec{\alpha} \geq \vec{\phi}^A \cdot \vec{\alpha}\}$. We then apply the same strategy as for the Sanov proof but applied to the product space of the two distributions P_A, P_B. This requires us to minimize:*

$$\begin{aligned} f(\vec{\phi}^B, \vec{\phi}^A) &= D(\vec{\phi}^B || P_B) + D(\vec{\phi}^A || P_A) \\ +\tau_1\{\sum \vec{\phi}^B - 1\} &+ \tau_2\{\sum \vec{\phi}^A - 1\} + \gamma\{\vec{\phi}^A \cdot \vec{\alpha} - \vec{\phi}^B \cdot \vec{\alpha}\}, \end{aligned} \tag{12}$$

where the τ's and γ are Lagrange multipliers. The function $f(.,.)$ is convex in the $\vec{\phi}$ and the Lagrange constraints are linear. Therefore there is a unique minimum which occurs at:

$$\vec{\phi}^{B*} = \frac{P_A^\gamma P_B^{1-\gamma}}{Z[1-\gamma]}, \quad \vec{\phi}^{A*} = \frac{P_A^{1-\gamma} P_B^\gamma}{Z[\gamma]}, \tag{13}$$

subject to the constraint $\vec{\phi}^A \cdot \vec{\alpha} = \vec{\phi}^B \cdot \vec{\alpha}$. The unique solution occurs when $\gamma = 1/2$ (because this implies $\vec{\phi}^{B*} = \vec{\phi}^{A*}$ and so the constraints are satisfied.) We define $\vec{\phi}_{Bh} = P_A^{1/2} P_B^{1/2} / Z[1/2]$ (Bh is short for Bhattacharyya). We therefore obtain:

$$\begin{aligned} (N+1)^{-J^2} 2^{-N\{D(\vec{\phi}_{Bh}||P_B) + D(\vec{\phi}_{Bh}||P_A)\}} &\leq Pr\{(\vec{\phi}^B, \vec{\phi}^A) \in E\} \\ &\leq (N+1)^{J^2} 2^{-N\{D(\vec{\phi}_{Bh}||P_B) + D(\vec{\phi}_{Bh}||P_A)\}}. \end{aligned} \tag{14}$$

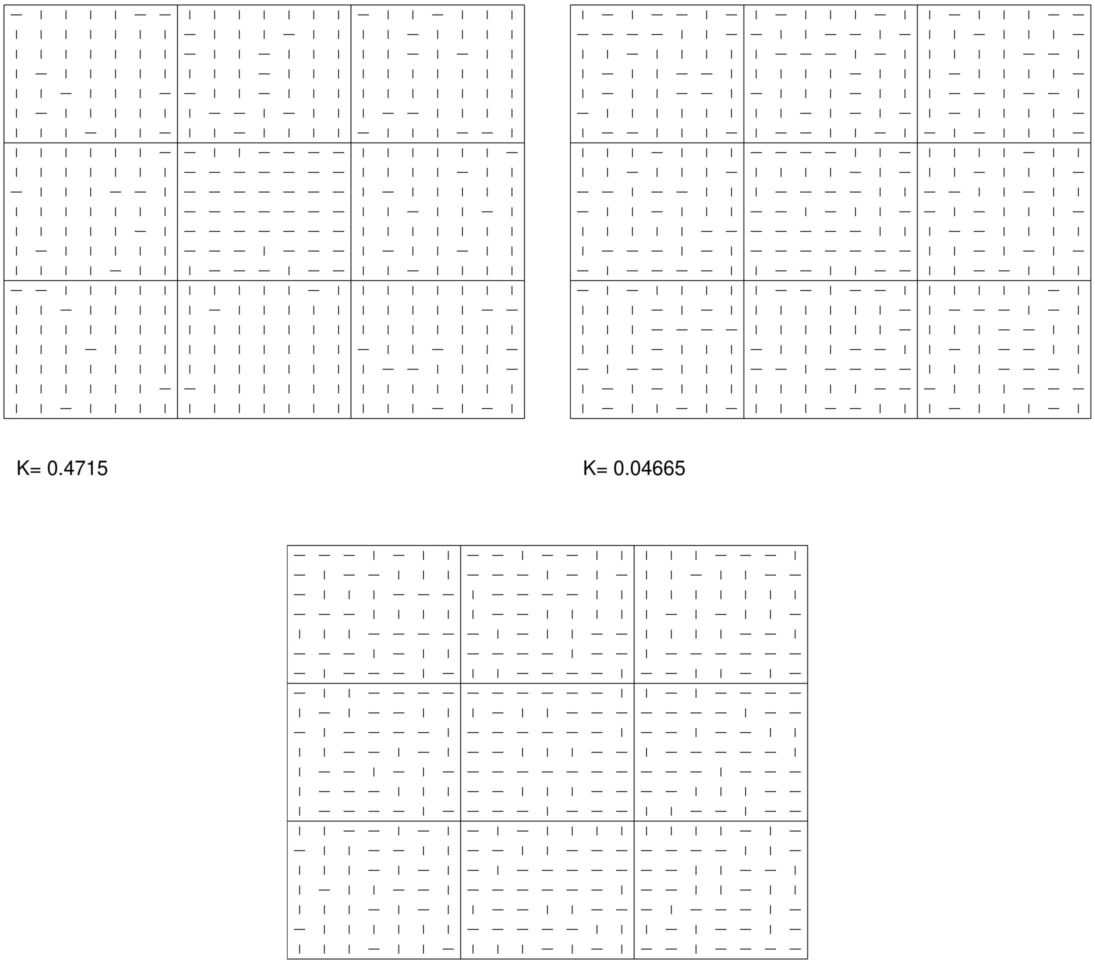
We define $B(P_A,P_B) = (1/2)\{D(\vec{\phi}_{Bh}||P_B) + D(\vec{\phi}_{Bh}||P_A)\}$. Substituting in for $\vec{\phi}_{Bh}$ from above yields $B(P_A,P_B) = -\log\{\sum_\mu P_B^{1/2}(a_\mu) P_A^{1/2}(a_\mu)\}$. Hence result.

This result tells us that the order parameter for the second texture task is just $2B(P_A,P_B)$. This is just another measure of the distance between P_A and P_B. Once again the problem becomes increasingly hard as the distributions become more similar but there is no critical point and no phase transition.

We now consider our third, and final, task of determining whether we can find a target A among a large number of texture samples B. We let the number of texture samples from B be Q^N. The interest is how the phase space of the number of texture samples affects the difficulty of the task. As we will show this leads to a phase transition.

Theorem 3. *The expected number of B texture samples which have greater reward than the A texture sample is determined by an order parameter $K = 2B(P_A,P_B) - \log Q$. If $K > 0$ then, as $N \mapsto \infty$, the expected number of such B texture samples tends to zero. If $K < 0$ then it tends to ∞. (N is the number of elements in each texture sample and the number of B texture samples is Q^N.)*

Proof. *The expected number, $< F_B >$, of B texture samples with rewards higher than the A texture sample is given by $Q^N Pr(\vec{\phi}_B \cdot \vec{\alpha} \geq \vec{\phi}_A \cdot \vec{\alpha})$. By Theorem 2, we can bound this by:*



K= 0.4715

K= 0.04665

K= −0.03228

Figure 6: Popout: P_A sample in middle, surrounded by P_B samples. Here we use a binary alphabet ($J = 2$) and vary P_A, P_B to change the order parameter K. Left, $P_A = (0.8, 0.2)$,$P_B = (0.167, 0.833)$. Right, $P_A = (0.667, 0.333)$, $P_B = (0.375, 0.625)$. Bottom, $P_A = (0.6, 0.4)$, $P_B = (0.5, 0.5)$. A non-integer value of Q ($Q > 1$) is used to save space.

$$\frac{1}{(N+1)^{J^2}} 2^{-N\{2B(P_A,P_B)-\log Q\}} \leq < F_B > \leq (N+1)^{J^2} 2^{-N\{2B(P_A,P_B)-\log Q\}}. \tag{15}$$

For large N, the bounds are determined by $K = 2B(P_A, P_B) - \log Q$. If $K > 0$ the expected number of B texture samples tends to zero as $N \mapsto \infty$. For $K < 0$, it tends to ∞.

The third task is governed by the order parameter $K = 2B(P_A, P_B) - \log Q$. There is a phase transition at $K = 0$ and the task becomes impossible to solve for $K < 0$. More intuitively, the task is only possible provided the difference between the distributions, measured by $2B(P_A, P_B)$, is bigger than the number of distractors, as measured by $\log Q$.

Corollary 3. *The probability that the A texture sample reward is lower than all the B texture samples rewards is less than $(N+1)^{J^2} 2^{-N\{2B(P_A,P_B)-\log Q\}}$.*

Proof. *This follows from the proof of Theorem 3 and the use of Boole's inequality: $Pr(A_1 \text{ or } ... \text{ or } A_n) \leq \sum_{i=1}^{n} Pr(A_i)$.*

Finally, we observe that these theorems involved several different measures of distance between probability distributions. These measures will reappear throughout the rest of the paper. For clarity, we summarize them and present ordering relations between them. Specifically, the measures are: (i) the Chernoff information $C(P_A, P_B)$ defined in Corollary 1, (ii) the Bhattacharyya distance [2] $B(P_A, P_B) = (1/2)\{D(\vec{\phi}_{Bh}||P_B) + D(\vec{\phi}_{Bh}||P_A)\}$ defined in Theorem 2, and (iii) the Kullback-Leibler divergences defined in equation (2), $D(P_A||P_B) = \sum_\theta P_A(\theta) \log(P_A(\theta)/P_B(\theta))$ and $D(P_B||P_A)$ (as stated before, these divergences are technically not measures).

The following relationship can be readily verified, see [5]:

$$0 \leq B(P_A, P_B) \leq C(P_A, P_B) \leq \min\{D(P_A||P_B), D(P_B||P_A)\}. \tag{16}$$

3 Mathematical Formulation of Road Tracking and Snakes

We now proceed to study the more realistic problem of curved tracking in real images. We consider two important examples. The first is for road tracking from aerial images by Geman (D.) and Jedynak [7] which used a novel active search algorithm to track a road in an aerial photograph with empirical convergence rates of $O(N)$ for roads of length N. Their algorithm is highly effective for this application and is arguably the best currently available. Our second example is the use of the Dijkstra algorithm to search for snakes between two feature points by Geiger and Liu [6]. They used a feature detector to find salient features, like corners, and then grew a snake between two feature points using Dijkstra's algorithm which was then used for high level grouping to detect human silhouettes. They report that Dijkstra's algorithm is 4-10 times faster than Dynamic Programming for this problem.

We wish to determine order parameters for characterizing the difficulty of these problems, to determine whether they are solvable, and how their difficulty depends on the statistical properties of the domain. In this section we give a mathematical formulation

[2] This Bhattacharyya distance arises in the Bhattacharyya bound for error rates [13].

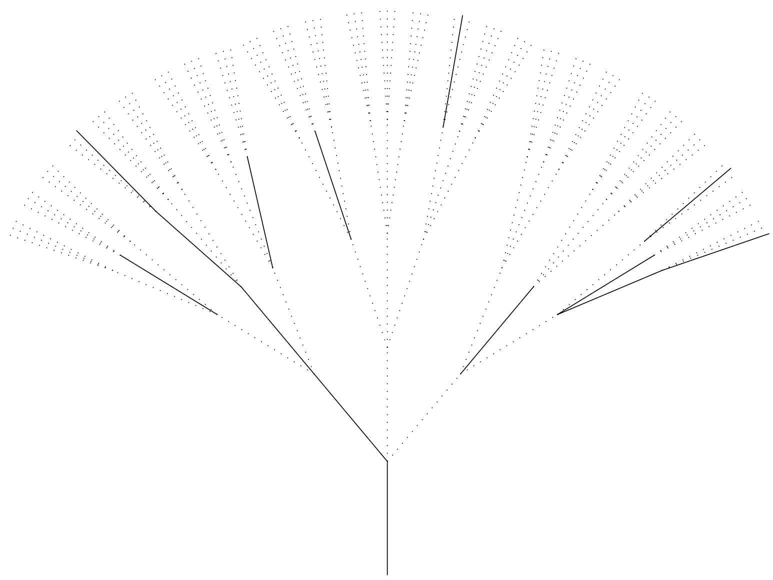


Figure 7: A simulated road tracking problem where dark lines indicate strong edge responses and dashed lines specify weak responses. The branching factor is three. The data was generated by stochastic sampling using a simplified version of the models analyzed in this paper. In this sample there is only one strong candidate for the best path (the continuous dark line) but chance fluctuations have created subpaths in the noise with strong edge responses. The A* algorithm must search this tree starting at the bottom node.

for road tracking and snakes. We follow the derivation of Geman and Jedynak [7] because their formulation is probabilistic from the start and better suited to our purposes. (By contrast, the snake formulation adopted by Geiger and Liu first specifies an energy function and then interprets it as the negative logarithm of a probability.) There are two main elements to each model: first, the optimization criterion (determined from the Bayesian formulation), and then the algorithm chosen to optimize the criterion for a given image. In this paper, we only describe the models and their optimization criteria. The algorithms, and their convergence rates, are described in [16], [3].

We first specify Geman and Jedynak's road geometry. A road hypothesis X is a set of connected straight-line segments called *arcs*, $x_1, \ldots, x_N$. The initial position of the road, arc x_0, is specified. The road is constrained to be smooth with the smoothness specified by a shift-invariant conditional probability distribution $P_G(x_{i+1}|x_i) = P_{\Delta G}(x_{i+1} - x_i)$, where $x_{i+1} - x_i$ denotes the difference in orientation between x_{i+1} and x_i. For example: the simplest case studied by Geman and Jedynak allows each road segment to join three subsequent possible road segments – straight, left (5 degrees), or right (5 degrees) – with equal probability of 1/3. The *prior probability* of any road is specified by $P(X) = P(x_0, x_1, \ldots, x_N) = \prod_{i=0}^{N-1} P_{\Delta G}(x_{i+1} - x_i)$. For the case above we have 3^N possible roads each with probability $1/3^N$.

Geman and Jedynak derive their likelihood function by first designing an oriented nonlinear filter to detect arcs of road. The intuition, see figure (7), is that the filter response Y is large for arcs where the gradient along the arc is small and the gradient across the arc is high. The response is small otherwise. They run the filter on examples of on-road and off-road arcs, gather statistics and compute empirical probability distributions $P_{on}(Y_a = y_a)$ and $P_{off}(Y_a = y_a)$. The *likelihood function* is given by $P(Y|X) = \prod_{x_a \in X} P_{on}(Y_a = y_a) \times \prod_{x_a \notin X} P_{off}(Y_a = y_a)$.

To obtain Geman and Jedynak's posterior distribution we apply Bayes Theorem $P(X|Y) = P(Y|X)P(X)/P(Y)$. We then use the prior and likelihood function above, take logarithms, and drop the constant terms. This gives:

$$\log P(X|Y) = \sum_{1=1}^{N} \log\{P_{on}(y_i)/P_{off}(y_i)\} + \sum_{i=0}^{N-1} \log P_{\Delta G}(x_{i+1} - x_i) + const. \tag{17}$$

We now solve for the most probable road $X^* = \arg\max_X \log P(X|Y)$. This gives the optimal criterion for road detection.

We now consider the alternative formulation of Geiger and Liu based on snakes [9]. As we will demonstrate, their formulation can be expressed in a similar form to Geman and Jedynak. Snakes are usually formulated in terms of energy function minimization of the position of a target curve $\{\vec{x}(t) : 0 \leq t \leq 1\}$: $E[x(t)] = \lambda \int_{t=0}^{1} dt\, |ds/dt| + \mu \int_{t=0}^{1} dt \kappa^2(t)$ $-\nu \int_{t=0}^{1} \left|\vec{\nabla} I(x(t)\right|$ [9]. This can be transformed into Bayesian form by setting $P([x(t)]|I)$ $= (1/Z)e^{-E[x(t)]}$ where the first two terms correspond to the geometric prior and the last term to the likelihood function. Why bother to make this transformation? The basic advantage is that it enables learning which will eliminate the free parameters in the model (which contrasts with the frequently expressed criticism that energy function models contain many parameters which have to be specified by hand.)

Indeed statistical analysis of real data typically gives quite different likelihood functions from those derived from a Bayesian reformulation of the standard snake model [9]. To see this compare $-E[x(t)]$ for the snake with equation (17). The log likelihood ratios $\log\{P_{on}(y_i)/P_{off}(y_i)\}$ correspond to $\nu \int_{t=0}^{1} \left|\vec{\nabla} I(x(t)\right|$. This would imply that the evidence (i.e. the log-likelihood ratio) for an edge increases *linearly* with the magnitude of the gradient. But this is counter-intuitive because it is unreasonable that a point x where $\left|\vec{\nabla} I(x(t))\right| = 100$ should have ten times more evidence for being an edge than a point where $\left|\vec{\nabla} I(x(t)\right| = 10$ (in most real images both points would definitely be edges). Instead we would expect the evidence for an edge to reach an asymptote after the gradient magnitude reaches a certain threshold. This can in fact be shown by statistical analysis of the $\left|\vec{\nabla} I(x(t)\right|$ edge detector using the same learning techniques employed by Geman and Jedynak [7]. We performed statistical analysis on a range of images (having first located the edges by hand) and obtained empirical results shown in figure (8). The general shapes of the P_{on}, P_{off} and their log likelihood ratio are very similar from image to image[3]. The log-likelihood terms clearly show the thresholding effect argued for above. We should add that Geiger and Liu [6] used a modification of snakes which makes their likelihood terms much more similar to ours than to those used in the original snake model [9].

[3]These plots of P_{on}, P_{off} are also somewhat similar to those observed by Balboa and Grzywacz [1] who obtained edge statistics in a variety of domains in order to model the retinal receptive fields of animals. A

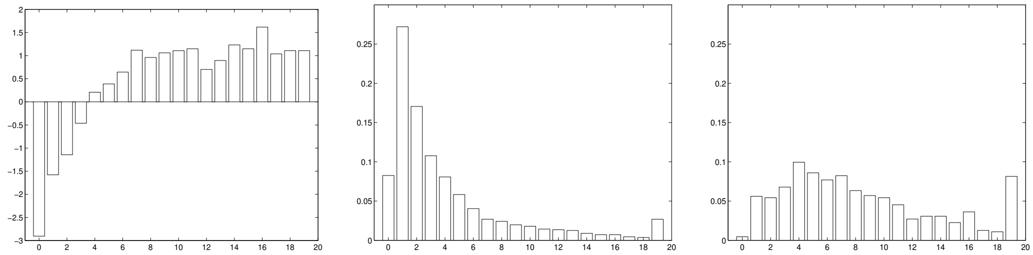


Figure 8: The log likelihood ratios (far left) of the off-edge probabilities $p_{off}(y)$ (center) and the on-edge probabilities $p_{on}(y)$(right), where $y = \left|\vec{\nabla} I\right|$. These distributions, and ratios, were very consistent for a range of images. The filter responses y, on the horizontal line, were quantized to take 20 values.

The first smoothness term for snakes, $\lambda \int_{t=0}^{1} dt\, |ds/dt|$, can be discretized and is equivalent to a shift-invariant conditional probability distribution $P(x_{i+1}|x_i) = P_{\Delta G}(x_{i+1} - x_i)$ – a first order Markov chain on position variables $\vec{x}$. The second smoothness term, $\mu \int_{t=0}^{1} dt \kappa^2(\vec{x}(t))$, can be discretized to a second order Markov chain in $\vec{x}$. Observe, however, that the order of these chains depends on the variables used. We could, for example, change variables to $\vec{q}$ which represents the position and local orientation. The smoothness term $\mu \int_{t=0}^{1} dt \kappa^2(\vec{x}(t))$ will correspond to a first order Markov chain in these variables. Zhu [22] investigates the effective of different order Markov chains for learning shape distributions from real image curves (and also describes the technical subtleties of discretizing models such as snakes).

Finally, we choose to rewrite the log posteriors by adding a constant term. This term increases the symmetry of the cost function by expressing the prior as a log-likelihood ratio and will make it easier to prove our results. We define $U(x_{i+1} - x_i)$ to be the *uniform distribution*, which of course is independent of $x_{i+1} - x_i$, and define a *reward* function:

$$R(X) = \sum_{1=1}^{N} \log\{\frac{P_{on}(y_i)}{P_{off}(y_i)}\} + \sum_{i=0}^{N-1} \log\{\frac{P_{\Delta G}(x_{i+1} - x_i)}{U(x_{i+1} - x_i)}\}. \quad (18)$$

To clarify our notation, the path is determined by a connected sequence of arcs $x_1, ..., x_N$. The $\{y_i\}$ represent measurements based on the image intensity on, or in a local neighbourhood of, these arcs. More precisely, we define $y_i = y(\{I(x) : x \in Nbh(x_i)\})$, where the function $y(.)$ specifies our choice of arc detector operator and $Nbh(x_i)$ specifies the neighbourhood of the arc x_i (i.e. the support of $y(.)$).

Both Geman and Jedynak and Geiger and Liu can be expressed in the form of equation (18). The variables X can represent either position or position plus orientation, depending on the application.

Such reward functions are ideally suited to A* graph/tree search algorithms [12],[14], which we describe and analyze in [16],[3]. A* searches the nodes – possible branches of the road/snake – which are most promising. The "goodness" $f(n)$ of a node n is $g(n) + h(n)$ where $g(n)$ is the reward to get to the node and $h(n)$ is a heuristic reward to get to the

detailed discussion of these issues is given in [11].

finish from n. Both Geman and Jedynak's and Geiger and Liu's algorithms can be shown [15] to be closely related to the A* algorithms. (Geiger and Liu's algorithm is a special case of A* and Geman and Jedynak's active searching is a close approximation).

4 Fundamental Limits: Can the problem be solved?

In this section we address the basic question of whether the target curve tracking problem can be solved at all. I.e. if we are finding a target curve in a cluttered background can we be sure that the optimal path, which maximizes a criterion like equation (18), corresponds with high probability to the target rather than to some random alignment of background clutter? Moreover, what are the statistical properties of the domain which determine the difficulty of the problem? We are therefore asking about the *fundamental limits* of the problem independent of any specific algorithm.

We will demonstrate the existence of order parameters, depending on statistical properties of the domain, and critical values of these parameters which cause phase transitions in the difficulty of detecting the target. We will also consider how good the best path will be (in terms of how far, by how many arcs, it diverges from the true path).

Our results will be obtained by the techniques described in section (2). It transpires that only simple modifications of those theorems will be sufficient to obtain our results. Our proofs rely on three basic elements: (i) Sanov's theorem which shows that the probability of rare events decreases *exponentially* with the length of the subpath, (ii) An *onion peeling* strategy which allows us to recursively analyze the search tree, and (iii) the use of standard techniques for summing, and bounding, exponential series generated by (i). See figure (7) for an illustration of this search task.

We define the problem on a Q-nary tree with the prior probabilities specified by $P_{\Delta G}$. A possible road can be represented as a sequence $x_1, x_2, ..., x_N$ of arcs of this tree. We can apply an edge detector which has quantized response values of $y \epsilon \{1, .., J\}$ (where $J << N$). By analysis of our domain we determine probabilities $P_{on}(y)$ and $P_{off}(y)$ for the probabilities of response value y depending on whether the arc we are testing is on or off the road. (We assume that the edge responses are statistically independent. This assumption may be questioned but it is assumed by [7],[6] and almost all the edge detector literature in computer vision).

There are two basic questions we can ask: (i) what is the probability that the true path has reward higher than any of the *completely false paths* (i.e. paths which are completely off the road), and (ii) by how much do we expect the path with highest reward to differ from the true path? Answering the first question is necessary to ensure that it is worth attempting to answer the second question.

To obtain our results we have to put bounds on the probable values of $R(X)$ in equation (18). We therefore have two log-likelihood ratios to consider: (i) the data term $\log P_{on}/P_{off}$, and (ii) the prior term $\log P_{\Delta G}/U$. For the true path the data will be generated by P_{on} and the geometry by $P_{\Delta G}$. Conversely, for completely false paths the data is generated by P_{off} and the geometry by U. We could obtain bounds for the data and the prior term directly by simply using the theorems, and corollaries, from section (2). All we need do is set $(P_A, P_B) = (P_{on}, P_{off})$ or $(P_A, P_B) = (P_{\Delta G}, U)$ respectively.

We are more interested, however, in dealing with the combined case of the full reward function. This can be handled by a straightforward extension of our previous theorems. First, we define $\alpha_\mu = \log P_{on}(\mu)/P_{off}(\mu)$, $\mu = 1, ..., J$ and $\beta_\nu = \log P_{\Delta G}(\nu)/U(\nu)$, $\nu = 1, ..., Q$ where the alphabet for the data and the prior are $\{\mu : \mu = 1, ..., J\}$ and $\{\nu : \nu = 1, ..., Q\}$ respectively. We let $\vec{\phi}^{off}, \vec{\psi}^{off}$ represent data and prior types for the false paths. Similarly, $\vec{\phi}^{on}, \vec{\psi}^{on}$ represent data and prior types for the true paths.

Our main result, Theorem 4, comes from extending Sanov's theorem to the product space of four distributions. The proof is a slight modification of our proof of Theorem 2, which dealt with product spaces of two dimensions, and the phase transition proof of Theorem 3.

Theorem 4. *The expected number $< F_T >$ of completely false paths which have greater reward than the true path is determined by an order parameter $K = 2B(P_{on}, P_{off}) + 2B(U, P_{\Delta G}) - \log Q$, where $B(P_{on}, P_{off}) = -\log\{\sum_\mu P_{off}^{1/2}(a_\mu) P_{on}^{1/2}(a_\mu)\}$. As $N \mapsto \infty$ there is a phase transition at $K = 0$ so that $< F_T >= 0$ for $K > 0$ and $< F_T >\mapsto \infty$ for $K < 0$. If $K < 0$ it is impossible to detect the true road.*

Proof. *We start by modifying our proof of Theorem 2. More specifically, we define the set:*

$$E_T = \{(\vec{\phi}^{off}, \vec{\psi}^{off}, \vec{\phi}^{on}, \vec{\psi}^{on}) : \ \vec{\phi}^{on} \cdot \vec{\alpha} + \vec{\psi}^{on} \cdot \vec{\beta} \leq \vec{\phi}^{off} \cdot \vec{\alpha} + \vec{\psi}^{off} \cdot \vec{\beta}\}, \tag{19}$$

and we replace equation (12) by:

$$\begin{aligned} f(\vec{\phi}^{off}, \vec{\psi}^{off}, \vec{\phi}^{on}, \vec{\psi}^{on}) = D(\vec{\phi}^{off}||P_{off}) + D(\vec{\psi}^{off}||U) + D(\vec{\phi}^{on}||P_{on}) + D(\vec{\psi}^{on}||P_{\Delta G}) \\ +\tau_1\{\sum_\mu \phi_\mu^{off} - 1\} + \tau_2\{\sum_\nu \psi_\nu^{off} - 1\} + \tau_3\{\sum_\mu \phi_\mu^{on} - 1\} + \tau_4\{\sum_\nu \psi_\nu^{on} - 1\} \\ +\gamma\{(\vec{\phi}^{on} \cdot \vec{\alpha} + \vec{\psi}^{on} \cdot \vec{\beta}) - (\vec{\phi}^{off} \cdot \vec{\alpha} + \vec{\psi}^{off} \cdot \vec{\beta})\}, \end{aligned} \tag{20}$$

where the τ's and γ are Lagrange multipliers as before. Once again the function $f(.,.,.,.)$ is convex in the $\vec{\phi}, \vec{\psi}$ and the Lagrange constraints are linear. Therefore there is a unique minimum given by:

$$\vec{\phi}^{off*} = \frac{P_{on}^\gamma P_{off}^{1-\gamma}}{Z[1-\gamma]}, \quad \vec{\phi}^{on*} = \frac{P_{on}^{1-\gamma} P_{off}^\gamma}{Z[\gamma]}, \quad \vec{\psi}^{off*} = \frac{P_{\Delta G}^\gamma U^{1-\gamma}}{Z_2[1-\gamma]}, \quad \vec{\psi}^{on*} = \frac{P_{\Delta G}^{1-\gamma} U^\gamma}{Z_2[\gamma]}, \tag{21}$$

subject to the constraint $(\vec{\phi}^{on} \cdot \vec{\alpha} + \vec{\psi}^{on} \cdot \vec{\beta}) = (\vec{\phi}^{off} \cdot \vec{\alpha} + \vec{\psi}^{off} \cdot \vec{\beta})$.

The unique solution occurs when $\gamma = 1/2$ (because this implies $\vec{\phi}^{off*} = \vec{\phi}^{on*}$ and $\vec{\psi}^{off*} = \vec{\psi}^{on*}$. Hence $\vec{\phi}^{off*} \cdot \vec{\beta} = \vec{\phi}^{on*} \cdot \vec{\beta}$ and $\vec{\psi}^{off*} \cdot \vec{\beta} = \vec{\psi}^{on*} \cdot \vec{\beta}$, so the constraints are satisfied.) We define $\vec{\phi}_{Bh} = \vec{\phi}^{off*} = \vec{\phi}^{on*}$ and $\vec{\psi}_{\mu^{-1}(1/2)} = \vec{\psi}^{off*} = \vec{\psi}^{on*}$. We define $B(P_{on}, P_{off}) = (1/2)\{D(\vec{\phi}_{Bh}||P_{off}) + D(\vec{\phi}_{Bh}||P_{on})\} = -\log\{\sum_\mu P_{off}^{1/2}(a_\mu) P_{on}^{1/2}(a_\mu)\}$ (this last equality can be verified by substituting for $\vec{\phi}_{Bh}$) and $B(U, P_{\Delta G})$ analogously). This yields:

$$\begin{aligned} (N+1)^{-J^2Q^2} 2^{-N\{2B(P_{on},P_{off})+2B(U,P_{\Delta G})\}} \leq Pr\{(\vec{\phi}^{off}, \vec{\psi}^{off}, \vec{\phi}^{on}, \vec{\psi}^{on}) \in E_T\} \\ \leq (N+1)^{J^2Q^2} 2^{-N\{2B(P_{on},P_{off})+2B(U,P_{\Delta G})\}}. \end{aligned} \tag{22}$$

We now adapt the proof of Theorem 3. The expected number of completely false paths with types in E_T is given by $< F_T >= Q^N(1 - Q^{-1})Pr(\vec{\phi} \in E_T)$. Using equation (8) we can bound this by:

$$\frac{2^{-N\{2B(P_{on},P_{off})+2B(U,P_{\Delta G})-\log Q\}}}{(N+1)^{J^2Q^2}} \leq \frac{< F_T >}{1 - Q^{-1}} \leq (N+1)^{J^2Q^2} 2^{-N\{2B(P_{on},P_{off})+2B(U,P_{\Delta G}-\log Q\}}. \tag{23}$$

The exponential factor in equation (23) is then given by $K = 2B(P_{on}, P_{off}) + 2B(U, P_{\Delta G}) - \log Q$ and we have:

$$\frac{2^{-NK}}{(N+1)^{(J^2Q^2)}} \leq \frac{< F_T >}{1 - Q^{-1}} \leq (N+1)^{(J^2Q^2)} 2^{-NK}. \tag{24}$$

It follows directly from equation (24) that $< F_T >$ undergoes a phase transition at $K = 0$ and $N \mapsto \infty$. If $K > 0$ then the expected number of completely false paths above threshold is 0. But if $K < 0$ then the expected number of paths above threshold becomes infinite.

The results of this theorem are not surprising. The order parameter $K = 2B(P_{on}, P_{off}) + 2B(U, P_{\Delta G}) - \log Q$ balances the effectiveness of the edge detector, measured by $2B(P_{on}, P_{off})$, against a geometric factor $2B(U, P_{\Delta G}) - \log Q$ which is determined by the number of possible paths. The more reliable the edge detector (i.e. the bigger $2B(P_{on}, P_{off})$) then the easier the problem. Similarly, the smaller the number of possible false paths (i.e. the larger $2B(U, P_{\Delta G}) - \log Q$) the easier the problem becomes.

Observe that, following Geman and Jedynak [7], our tree representation for paths is a simplifying assumption of the Bayesian model. It assumes that once a path diverges from the true path it can never recover (though we stress that the *algorithm* is able to recover from false starts). How bad is this approximation? In Coughlan and Yuille [3] we argue that the main effect is simply to shift the order parameters upwards. Intuitively, instead of a single target path there will be a cloud of good paths fluctuating on and off the target path. This will effectively increase the order parameter by making the target easier to detect. This order parameter shift is related to the number of additional paths close to the target and with high reward. This is illustrated in figure (1) where we give examples of target detection in clutter for different values of the order parameter K given in Theorem 4. The results are broadly consistent with Theorem 4 but observe that the path in the middle panel is visible even though $K < 0$, which can be explained by the order parameter shift (i.e. the true order parameter is greater than K).

4.1 Mixture Paths: When a Good Path goes Bad

So far, we have only compared the true path to the completely false paths. But there are a large class of paths which lie partially on the true road and partially off it. These are paths which are good and then go bad. How many of these do we expect to have higher rewards than the true path? More precisely, what is the *expected error*, where we define the error to be the number of arcs which are off the road for the path with biggest total reward?

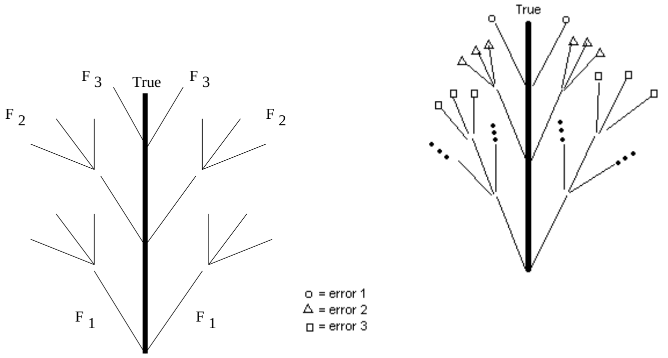


Figure 9: Left: We can divide the set of paths up into N subsets $F_1, ..., F_N$ as shown here. Paths in F_1 are completely off-road. Paths in F_2 have one on-road segment and so on. Intuitively, we can think of this as an onion where we peel off paths stage by stage. Right: When paths leave the true path they make errors which we characterize by the number of false arcs. For example, a path in F_1 has error N, a path in F_i has error $N+1-i$.

A key concept here is the onion-like structure of the tree representation, see figure (9). This structure allows us to classify all paths in terms of sets $F_1, F_2, F_3, ...$ which depend on where they branch off from the true path. Paths which are always bad (i.e. completely false) correspond to F_1. Paths which are good for one segment, and then go bad, form F_2 and so on. Our previous results have compared the properties of paths in F_1 to those of the true path. To understand the probabilities of paths in F_2 relative to the true path, we simply have to peel off the first layer of the onion (i.e. remove the first arc of the true path) and the comparison of the rest of the true path to F_2 reduces to our previous result for F_1. Thus our results for F_1 can be readily adapted to $F_2, F_3, ...$ Observe that paths in F_i share the first $(i-1)$ arcs with the true path, by definition, and hence have the same partial rewards for these arcs. Therefore we often only need to compare the rewards for the remaining arcs.

Theorem 4 also applies to the sections of the path which are off the true road. We can consider paths in F_{N+1-M}, which start on the true road and then are off it for their last M segments, see figure (9). Our theorems give us probabilistic bounds on the chances that the reward for these off-road arcs is greater than the reward for the remainder of the true path or, if we prefer, than other rewards such as the average reward of the true path. The theorems contain alphabet size dependent factors, which are unimportant for large M, and decays exponentially with M with fall-off factors given by the appropriate order parameters K. Provided the phase factor is a long way above its critical value (i.e. we are not close to the phase transition) then the chances of having a higher reward path with a significant number of arcs being off-road therefore decreases very quickly (of course, close to the phase transition we will expect many mixed paths to have rewards close to the that of the true path). We now quantify this claim.

We will bound the expected error by making a series of approximations. If the path with biggest total reward lies in F_{N-M+1} then the error will be M (in the event of a tie we pick the worst case). The probability of this occurring is less than, or equal to, the probability $Pr_F(M)$ that there is at least one path in F_{N-M+1} with reward greater than the true reward (this is an upper bound because it ignores the possibility that the *highest* reward path is in any of the other $F_j : j \neq N-M+1$.) Observe that $Pr_F(M)$ is an upper bound on the distribution of possible errors and *not* a distribution on M (i.e. $\sum_M Pr_F(M) \neq 1$). We can then get an upper bound on the expected error:

$$< Error > \leq \sum_{M=1}^{\infty} M Pr_F(M). \tag{25}$$

Observe that we sum to ∞ rather than to N. This makes the bound looser, because the extra terms are all positive, but we do not need a tighter bound.

To put an upper bound on $Pr(M)$ we observe that paths in F_{N+1-M} have their first $N-M$ arcs in common with the true path. So to determine if they have higher rewards we only need to compare their remaining M arcs. From Theorem 4 and Boole's inequality[4] we get $Pr_F(M) \leq Q^M (M+1)^{J^2Q^2} 2^{-M\{2B(P_{on},P_{off})+2B(U,P_{\Delta G})\}}$. This is of form $Pr_F(M) \leq (M+1)^{J^2Q^2} 2^{-MK}$, where $K = 2B(P_{on}, P_{off}) + 2B(U, P_{\Delta G}) - \log Q$. We now place an upper bound on the expected error by substituting into equation (25) and summing the series.

We split the sum into two parts, see Appendix for details. The first ignores the alphabet factor and uses $Pr_F(M) \leq 2^{-M(K+\epsilon)}$ which will be an upper bound for $Pr_F(M)$ for $M > M_0$, where M_0 is a cutoff factor which depends on ϵ and the alphabet factors. The second part, $\hat{\Xi}(\epsilon, J^2Q^2, K)$, is an additional term used to deal with the alphabet factors in the regime where $M < M_0$.

This gives:

$$< Error > \leq \frac{2^{-(K-\epsilon)}}{(1-2^{-(K-\epsilon)})^2} + \hat{\Xi}(\epsilon, J^2Q^2, K). \tag{26}$$

This error is small except as $K \mapsto 0$ where is becomes unboundedly large. This is intuitive because the easier the problem (i.e. the larger K) then the smaller the expected number of errors.

This proves our main result:

Theorem 5. *The path with highest reward is expected to diverge from the best path by less than* $\frac{2^{-(K-\epsilon)}}{(1-2^{-(K-\epsilon)})^2} + \hat{\Xi}(\epsilon, J^2Q^2, K)$ *arcs. The upper bound for the divergence decreases exponentially with the order parameter* K. *As* $K \mapsto 0$ *the upper bound for the expected divergence becomes infinite.*

5 Hierarchical Models and the use of information

So far we have assumed that we know the correct probability model for the target. But how much harder do we make target detection by using a weaker, or incorrect, model (i.e. a weaker prior probability distribution)? Or, more positively, can we quantify how much easier we make the task by using more information about the target?

---

[4]Recall that Boole's inequality states that $Pr(A_1 \; or \; A_2 \; or \; ... \; or \; A_n) \leq \sum_{i=1}^{n} Pr(A_i)$.

There are at least three reasons why a weaker prior might be used: (I) There may not be enough information about the target to have an accurate prior for it. (II) We may want to search for several different targets and it might be more economical to use one generic prior model which would account for all of these targets (at the cost of modelling each of them relatively poorly) rather than having different models for each target. (III) Algorithmic considerations may favour using a weaker prior rather than a prior which is more accurate but harder to compute with.

Our order parameter theory offers a way to tackle this problem which we illustrate on road tracking. Suppose we have a single high-level model for a road with a *high level* geometric prior $P_H(\Delta x)$. Let us assume a weaker *generic* prior $P_G(\Delta x)$. We can define two different rewards R_G and R_H:

$$
\begin{aligned}
R_G(\{x_i\}) &= \sum_i \log \frac{P_{on}(y_i)}{P_{off}(y_i)} + \sum_i \log \frac{P_G(x_i)}{U(x_i)}, \\
R_H(\{x_i\}) &= \sum_i \log \frac{P_{on}(y_i)}{P_{off}(y_i)} + \sum_i \log \frac{P_H(x_i)}{U(x_i)}.
\end{aligned} \tag{27}
$$

The optimal Bayesian strategy to search for the road would be to use the high level model and evaluate paths based on their rewards R_H. But we wish to evaluate how much we lose by using the generic reward R_G.

We will focus here on the case when the generic prior P_G is a projection of the high-level prior P_H onto a simpler class of probability distributions. In particular, we will be concerned with the *Amari* projection (see Yuille and Coughlan 1999 for alternatives). This projection requires that the projected distributions $P_H(y)$ are related to the generic distributions $P_G(x)$ by $\sum_x P_H(x) \log P_G(x) = \sum_x P_G(x) \log P_G(x)$. It arises, in order parameter theory, when we evaluate the probability the a false path has higher reward than the *expected reward* of the true path.

The Amari projection is particularly important because the Minimax Entropy learning theory [20] naturally gives rise to a set of probability distributions which are related by Amari projection. This learning theory involves estimating probability distributions from the statistics of a class of filters and includes a *filter pursuit* strategy to determine which filters (and corresponding statistics) should be used (ideally one would use as few filters as possible so as to keep the probability models simple and prevent overgeneralization from the data [20]). Therefore, similar to Fourier theory, Minimax Entropy learning prescribes a family of increasingly accurate probability models – depending on the number of filters included. It can be shown (Coughlan and Yuille 1999) [4] that *simplifying a Minimax Entropy probability distribution by removing filters (and their statistics) is equivalent to an Amari projection of the corresponding probabilities.*

Once again, we apply techniques from information theory to determine order parameters K_H^A, K_G^A for the high-level and generic rewards respectively (in both cases we assume that the true path is generated by the high-level model P_H):

$$
\begin{aligned}
K_H^A &= D(P_{on}||P_{off}) + D(P_H||U) - \log Q, \\
K_G^A &= D(P_{on}||P_{off}) + D(P_G||U) - \log Q.
\end{aligned} \tag{28}
$$

It follows from the definition of Amari projection that $K_H^A - K_G^A = D(P_H||U) - D(P_G||U) = D(P_H||P_G)$ (where $D(P||Q) = \sum_y P(y) \log P(y)/Q(y)$ is the *Kullback-Leibler*

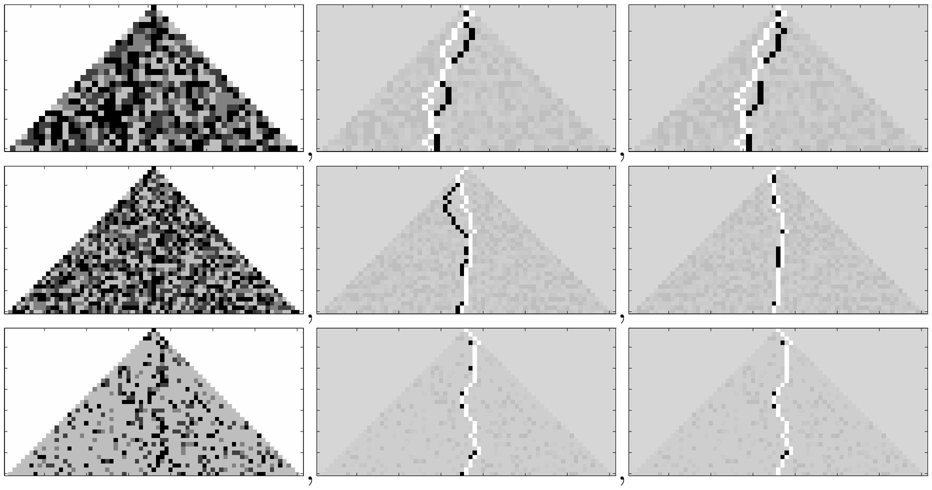


Figure 10: Top row:, The Ultra Regime $K^G_H < K^A_G < 0$. Left, the input image. Centre, the true path is shown in white and the best path found using the Generic model is shown in black (except when it overlaps with the true path). Right, similar, for the High-Level model. Observe that although the best paths found are close to the true path there is comparatively little overlap. Middle row: The Challenging Regime $K^A_G < 0 < K^A_H$. Observe that the Generic models fails (centre) but the High-Level model succeeds (right). Bottom row: The Easy Regime $0 < K^A_G < K^A_H$. In this regime both the Generic and High-Level models succeed (centre and right). The true path is always generated by P_H.

divergence between distributions $P(y)$ and $Q(y)$). Therefore the high-level prior P_H has a bigger order parameter by an amount which depends on the distance between it and P_G as measured by the Kullback-Leibler divergence $D(P_H||P_G)$. (*This is precisely the quantity used in Minimax Entropy to determine the benefit of using an additional filter*). Recall that the target detection problem becomes insolvable (by any algorithm) when the order parameter is less than zero. Hence there are three regimes (see figure (10): (I) The *Ultra Regime* is when $K^A_G < K^A_H < 0$ (i.e. $D(P_H||U) + D(P_{on}||P_{off}) < \log Q$) and the problem cannot be solved (on average) by any model (or algorithm). (II) The *Challenging Regime* is where $K^A_G < 0 < K^A_H$ (i.e. $\log Q < D(P_H||U) + D(P_{on}||P_{off}) < \log Q + D(P_H||P_G)$) within which the problem can be solved by the high-level model but not by the generic model. (III) The *Easy Regime* is where $K^A_H > K^A_G > 0$ and the problem can be solved by either the generic or the high-level model.

In our simulations, see figure (10), we *generate* the target true paths by *stochastic sampling from the high level model.* To *detect* the best path we apply a dynamic programming algorithm to optimize the high-level or generic reward functions applied to the generated data. Dynamic programming is guaranteed to find the solution with highest reward.

These ideas allow us to formulate a hierarchy in which the priors for several high-level objects would all project onto the identical low-level prior, see figure (11). For example, we might have a set of priors $\{P_{H_i} : i = 1, ..., M\}$ for different members of the cat family. There might then be a generic prior P_G onto which all the $\{P_{H_i}\}$ project and which is considered the embodiment of "cattiness".

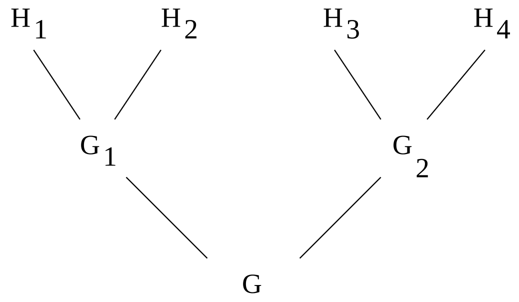


Figure 11: The Hierarchy. Two high-level models P_{H_1}, P_{H_2} "project" onto a low-level generic model P_{G_1}. In situations with limited clutter it will be possible to detect either H_1 or H_2 using the single generic model P_{G_1}. This idea can be extended to have hierarchies of projections. This is analogous to the superordinate, basic level, and subordinate levels of classification used in cognitive psychology.

This was illustrated [17] by choosing two high-level second-order Markov road models, which we called *English* and *Roman*. These two models generated very different types of roads – the English roads had many small-scale wiggles while the Roman roads tended to have long straight sections – but both models had *identical* projections to a generic (first-order) Markov road model. We determined, theoretically and experimentally, that there were, as above, three regimes. In the Ultra and Easy regimes, all models either failed (Ultra) or succeeded (Easy). In the Challenging regime the English, or Roman, models could find English, or Roman, roads but the generic model *could not find either*.

6 Discussion

In this section we discuss how certain assumptions made earlier in the paper can be relaxed.

Our analysis assumes that the results of the edge tests are independent. This assumption will be valid if the edge tests are applied a reasonable distance apart but will clearly break down if they are applied too closely together. However, the analysis given in this paper could be extended to the case when the edge process was also Markov with a local neighbourhood. Preliminary analysis on real data [11] suggests that the correlation between local edge profiles drops off rapidly with distance. This would imply that the assumptions in this paper are reasonable if the edge tests are applied close (but not too close) together and that a Markov model for edges could be used if finer resolution was needed. But further experimentation is needed to conclusively settle this issue.

We also assumed that the geometry model was stationary. However, this assumption can be relaxed to include the class of deformable templates where the deformations are local Markov. An example of this is the deformable hand template, see [2]. Such models can be written in form:

$$P_i(\vec{x}_i|\vec{x}_{i-1}) = F(f_i(\vec{x}_i) - \vec{x}_{i-1}), \tag{29}$$

where $F(.)$ and $f_i(.)$ are functions. Provided the *noise distribution* $F(.)$ is independent of i we can apply the theory of types and generalize our results. This representation can encode a prototype hand structure [2]. It should be emphasized that this approach requires that the approximate size of the object is known.

Our analysis also assumed that the starting point for the problem was given. *It should be emphasized* that our results can be directly adapted to the situation where the starting point is unknown. The only modification is that the number of false paths will increase and so we will have to modify the factor Q^N, which appeared in the proofs, to allow for these extra paths. But this modification will merely alter the phase factors by a constant which depends on the size of the image. The essence of our results remains unchanged.

7 Conclusion

This paper examined the fundamental limits of performing certain visual tasks. It was shown that the behaviour typically depended on an order parameter which could be calculated from the statistics of the problem domain. These results are algorithm independent and in some cases showed the existence of phase transitions where tasks became impossible at a critical value of the order parameter.

Our analysis of algorithms [16],[3] has shown that it is possible to detect certain types of image contours in linear expected number of node expansions (with given starting points) with expected constant time overhead per sort. We have shown how the convergence rates, and the choice of A* heuristics, depend on factors which can be used to characterize the problem domain. In particular, the entropy of the geometric prior and the Bhattacharyya bound [13] between P_{on} and P_{off} allow us to quantify intuitions about the power of geometrical assumptions and edge detectors to solve these tasks.

As shown in [15] many of the search algorithms proposed to solve these vision search problems [7],[6] are special cases of A* (or close approximations). It is hoped that the results of this paper will throw light on the success of the algorithms and may suggest practical improvements and speed ups.

One way of thinking about our results is in terms of the amount of prior information about the target needed to detect it. As we showed in section (5), in certain regimes simple models are effective at finding the target and more realistic high-level models are not needed. But in other regimes the simple models fail. So our theory can help specify "when dumb algorithms work"!

We also hope that our analysis can be extended to other visual search tasks. In such tasks there will often be multiple visual cues which can be used to detect specific visual targets. In such situations the search strategy becomes more complex and will involve several tests tuned to the different cues.

Appendix

We need to bound sums such as:

$$\sum_{m=0}^{\infty} m 2^{-Bm} (m+1)^{A}. \tag{30}$$

We pick a number ϵ and $M_0(\epsilon, A)$ such that $(m+1)^A < e^{m\epsilon}$, $\forall\ m > M_0(\epsilon, A)$. We can divide the sum into two parts:

$$\sum_{m=0}^{\infty} m 2^{-(B-\epsilon)m} + \hat{\Xi}(\epsilon, A, B), \tag{31}$$

where $\hat{\Xi}(\epsilon, A, B)$ is a correction factor used to correct for the alphabet factors for small $m < M_O(\epsilon, A)$.

Let $f(x) = \sum_{m=0}^{\infty} 2^{xm} = 1/(1-2^x)$. Then it is straightforward to differentiate both sides with respect to x to obtain $\sum_{m=0}^{\infty} m 2^{xm} = \frac{2^x}{(1-2^x)^2}$. We can therefore express:

$$\sum_{m=0}^{\infty} m 2^{-Bm}(m+1)^A = \frac{2^{-B}}{(1-2^{-B})^2} + \hat{\Xi}(\epsilon, A, B). \tag{32}$$

Acknowledgements

We want to acknowledge funding from NSF with award number IRI-9700446, from the Center for Imaging Sciences funded by ARO DAAH049510494, from the Smith-Kettlewell core grant, and the AFOSR grant F49620-98-1-0197 to A.L.Y. Lei Xu drew our attention to Pearl's book on heuristics and lent us his copy (unfortunately the book is out of print). His, and Irwin King's, hospitality at the Chinese University of Hong Kong was greatly appreciated by ALY. We would also like to thank Dan Snow and Scott Konishi for helpful discussions as the work was progressing and Davi Geiger for providing useful stimulation. Also influential was Bob Westervelt's joking request that he hoped James Coughlan's PhD thesis would be technical enough to satisfy the Harvard Physics Department. David Forsyth, Jitendra Malik, Preeti Verghese, Dan Kersten, Song Chun Zhu and Ying Nian Wu gave very useful feedback and encouragement.

References

[1] R. Balboa. PhD Thesis. Department of Computer Science. University of Alicante. Spain. 1997.

[2] J.M. Coughlan, D. Snow, C. English, and A.L. Yuille. "Efficient Optimization of a Deformable Template Using Dynamic Programming". In *Proceedings Computer Vision and Pattern Recognition. CVPR'98*. Santa Barbara. California. 1998.

[3] J.M. Coughlan and A.L. Yuille. "A Probabilistic Formulation of A*: O(N) Expected Convergence rates for Visual Search." Submitted to *Artificial Intelligence*. 1998.

[4] J.M. Coughlan and A.L. Yuille. "The Phase Space of Minimax Entropy Learning". Submitted to Neural Computation. 1999.

[5] T.M. Cover and J.A. Thomas. **Elements of Information Theory**. Wiley Interscience Press. New York. 1991.

[6] D. Geiger and T-L Liu. "Top-Down Recognition and Bottom-Up Integration for Recognizing Articulated Objects". In *Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition.* Ed. M. Pellilo and E. Hancock. Venice, Italy. Springer-Verlag. May. 1997.

[7] D. Geman. and B. Jedynak. "An active testing model for tracking roads in satellite images". *IEEE Trans. Patt. Anal. and Machine Intel.* Vol. 18. No. 1, pp 1-14. January. 1996.

[8] D.W. Jacobs. "Robust and Efficient Detection of Salient Convex Groups". *IEEE Trans. Patt. Anal. and Machine Intel.* Vol. 18. No. 1, pp 23-37. January. 1996.

[9] M. Kass, A. Witkin, and D. Terzopoulos. "Snakes: Active Contour models". In *Proc. 1st Int. Conf. on Computer Vision.* 259-268. 1987.

[10] D.C. Knill and W. Richards. (Eds). **Perception as Bayesian Inference**. Cambridge University Press. 1996.

[11] S. Konishi, A.L. Yuille, James M. Coughlan and S.C. Zhu. "Fundamental Bounds on Edge Detection: An Information Theoretic Evaluation of Different Edge Cues". In *Proceedings Computer Vision and Pattern Recognition, CVPR'99.* 1999.

[12] J. Pearl. **Heuristics**. Addison-Wesley. 1984.

[13] B.D. Ripley. **Pattern Recognition and Neural Networks**. Cambridge University Press. 1995.

[14] S. Russell and P. Norvig. "Artificial Intelligence: A Modern Approach. Prentice-Hall. 1995.

[15] A.L. Yuille and J. Coughlan. "Twenty Questions, Focus of Attention, and A*: A theoretical comparison of optimization strategies." In *Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition.* Ed. M. Pellilo and E. Hancock. Venice, Italy. Springer-Verlag. May. 1997.

[16] A.L. Yuille and J.M. Coughlan. "Convergence Rates of Algorithms for Visual Search: Detecting Visual Contours". In *Proceedings NIPS'98.* 1998.

[17] A.L. Yuille and J.M. Coughlan. "High-Level and Generic Priors for Visual Search: When does high level knowledge help?" Submitted to *Computer Vision and Pattern Recognition* CVPR'99. 1999.

[18] S.C. Zhu, Y. Wu, and D. Mumford. "Minimax Entropy Principle and Its Application to Texture Modeling". Neural Computation. Vol. 9. no. 8. Nov. 1997.

[19] S.C. Zhu and D. Mumford. "Prior Learning and Gibbs Reaction-Diffusion". IEEE Trans. on PAMI vol. 19, no. 11. Nov. 1997.

[20] S.C. Zhu and D. Mumford. "GRADE: A framework for pattern synthesis, denoising, image enhancement, and clutter removal." In Proceedings of International Conference on Computer Vision. Bombay. India. 1998.

[21] S-C Zhu, Y-N Wu and D. Mumford. FRAME: Filters, Random field and Maximum Entropy: — Towards a Unified Theory for Texture Modeling. Int'l Journal of Computer Vision 27(2) 1-20, March/April. 1998.

[22] S.C. Zhu. "Embedding Gestalt Laws in Markov Random Fields". Submitted to *IEEE Computer Society Workshop on Perceptual Organization in Computer Vision.*