Fundamental Limits of Bayesian Inference: Order Parameters and Phase Transitions for Road Tracking.

A. L. Yuille* and James M. Coughlan
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

December 27, 2002

* Address correspondence to A.L. Yuille.

Abstract

There is a growing interest in formulating vision problems in terms of Bayesian inference and, in particular, the maximimum a posteriori (MAP) estimator. This approach involves putting prior probability distributions, $P(X)$, on the variables X to be inferred and a conditional distribution $P(Y|X)$ for the measurements Y. For example, X could denote the position and configuration of a road in an aerial image and Y can be the aerial image itself (or a filtered version). We observe that these distributions define a probability distribution $P(X,Y)$ on the ensemble of problem instances. In this paper we consider the special case of detecting roads from aerial images [9] and demonstrate that analysis of this ensemble enables us to determine fundamental bounds on the performance of the MAP estimate (independent of the inference algorithm employed). We demonstrate that performance measures – such as the accuracy of the estimate and whether the road can be detected at all – depend on the probabilities $P(Y|X), P(X)$ only by an order parameter K. Intuitively, K summarizes the strength of local cues (as provided by local edge filters) together with prior information (i.e. the probable shapes of roads). We demonstrate that there is a phase transition at a critical value of the order parameter K – below this phase transition it is impossible to detect the road by any algorithm. In related work [25],[5], we derive closely related order parameters which determine the time and memory complexity of search and the accuracy of the solution using the A^* search strategy. Our approach can be applied to other vision problems and we briefly summarize results when the model uses the "wrong prior" [26]. We comment on how our work relates to studies of the complexity of visual search [21] and to critical behaviour (i.e. phase transitions) in the computational cost of solving NP-complete problems [19].

Index Terms: (i) Bayesian inference, (ii) Phase transitions, (iii) Curve tracking.

In Pattern Analysis and Machine Intelligence. PAMI. Vol. 22. No. 2. February. 2000.

1 Introduction

In recent years there has been growing interest in formulating problems in terms of Bayesian inference [13]. There has been encouraging success both in techniques for learning probability distributions from real data [27],[28],[29] and for solving difficult vision problems see, for example, [9],[10].

This paper uses the Bayesian framework to address more fundamental problems: 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?

Such problems have been addressed in previous work in vision, see for example [11] and papers cited therein. In particular, Tsotsos and his collaborators [20],[21],[15] have addressed problems such as visual search – the detection of a target among distractors – and the interpretation of line drawings. It is not clear, however, how their methods and results can be applied to models formulated in Bayesian terms. Standard computer science techniques [7], as employed in the early work of Tsotsos [21], rely on worst case analysis. As we show in this paper, however, Bayesian models come with a specified probabilty distribution over problem instances. In such cases, worst case analysis may be inappropriate because, for some Bayesian models, the worst cases can be shown to occur with vanishingly small probability. Recent work by Parodi *el al* [15] does address median case complexity for labelling polyhedral scenes but analysis of this task is very diffficult and they rely on empirical experiments.

We therefore argue that analysing the performance of Bayesian models requires the development of techniques that are naturally geared to such models and directly exploit their probabilistic nature and, in particular, the probability distribution on the ensemble of problem instances. This paper is an attempt to develop such techniques.

To make these issues more concrete we consider the task of detecting roads from aerial images [9] and, more briefly, the related work of Geiger and Liu [8] on the detection 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 domain will their algorithms succeed? What characteristics of the problem domain 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 analyzing the convergence speed and memory requirements [25],[5], helps provide answers to these questions.

Our starting point is the Bayesian formulation for the Geman and Jedynak problem. This requires specifying probability distributions for the local measurements (e.g. edge filter responses) conditioned on the possible positions of the road segments together with a prior probability distribution for the relative positions of the road segments (i.e. a prior on the road geometry). (Such models can be learnt from real data – see Geman and Jedynak [9] for learning distributions for the local measurements and [30] for distributions on contour geometry). For our analysis in this paper it is necessary that these probability distributions are shift-invariant Markov. (In fact, we are sometimes using stronger assumptions but – as demonstrated by Wu and Zhu – the analysis can be generalized to such cases).

This Bayesian formulation naturally implies a probability distribution on the ensemble of problem instances. This enables us to adapt techniques from Information Theory [6] to determine the probability of rare events – such as a problem instance where, by chance, MAP estimation incorrectly selects a false road hypothesis.

Our analysis shows that most performance measures of interest, such as the detectability of the road, only depend on the underlying probability distributions via a single constant K. This is analogous to the study of the Ising model of magnetism – see statistical physics [1] – where the behaviour of the model is largely determined by its net magnetization which is called an *order parameter*. As the temperature of the system passes through a *critical point* there is a *phase transition* and properties of the system (as measured by the order parameter) change drastically (for example, the magnetization becomes zero above the critical temperature). Similarly, in our Bayesian model the detectability of the road becomes impossible at a critical value of K. We can therefore consider K to be the *order parameter* of the system and say that a *phase transition* occurs at a critical value of K. (We stress that our model *does not* involve a temperature parameter and a phase transition occurs when the local measurement cues and the prior geometrical knowledge – as specified by the probability models – provide too little information to solve the task.)

Interestingly, there are some recent studies showing that order parameters and phase transitions exist for NP-complete problems and that these problems can be easy to solve for certain values of the order parameters [3],[19]. The problems these authors consider and the techniques they employ are, however, different from ours. Often their analysis only applies at the phase transition itself and uses approximation techniques such as mean field approximations and replica symmetry methods supported by empirical numerical studies. Moreover, because they are not addressing Bayesian inference problems their probability distribution over the ensemble of problem instances is often chosen for pragmatic reasons of mathematical analysis rather than arising naturally from the problem domain. Our work, however, is more closely related to (and partially inspired by) results of Karp and Pearl [16] for a binary tree path search problem [16] which also involved a phase transition.

We stress, however, that the phase transition is only one consequence of our analysis. The most important result, in our opinion, is the derivation of the order parameter as a fundamental quantity which characterizes the problem domain and determines most of the important performance measures.

Finally, we wish to mention three additional results: (I) similar techniques can be used to analyze the performance of specific A* algorithms to solve the road tracking problem and time and memory complexity results have been obtained [25],[5] which depend on closely related order parameters, (II) the analysis of this current paper can be generalized to deal with situations in which the Bayesian models use the "wrong priors" which are only approximations to the true underlying distributions [26], and (III) the techniques used in this paper have already been extended, by adapting the theory of large deviations from statistics [22], and applied to realistic texture discrimination tasks [23].

We start with a short technical overview, section (2), of the logical structure of the analysis. In section (3), we introduce the theory of types and, in particular, Sanov's theorem. We illustrate it by deriving order parameters for texture discrimination tasks including one example of a phase transition. Section (4) described the models of road tracking and snakes. In section (5) we explore the fundamental limits of road detection by using the theory of types to derive order parameters and phase transitions for this problem. In section (6), we discuss ways to extend our results. Finally, section (7) summarizes the paper.

2 Technical Overview

This section gives a brief technical overview of the paper. *We emphasize that this is a sketch only and the rigourous proofs are given in the rest of the paper.*

The input data (i.e. a filtered image) is denoted by Y and a hypothetical road configuration by X. We define an imaging model $P(Y|X)$ and a prior probability distribution $P(X)$ on road configurations. We estimate the configuration of the road by maximum a posterior (MAP) estimation, $X^* = \arg\max_X P(X|Y)$. This can be re-expressed as maximizing a *reward function* $R(X|Y)$ (see section (4). (This all follows directly from Geman and Jedynak [9]).

We observe that this defines a *Bayesian ensemble* of problem instances $P(X, Y)$. This enables us to determine in what situations the MAP estimator gives the correct answer. More formally, we can select a true road $\bar{X}$ (sampled from the prior distribution $P(X)$), generate data $\bar{Y}$ by sampling from $P(Y|\bar{X})$ and then ask the question is $\arg\max_X P(X|\bar{Y}) = \bar{X}$? Note that this analysis assumes that we know the true models for how the data is generated and use these same models for inference. In practice, our models will only be approximations to reality. In Yuille and Coughlan [26] we relax these assumptions and obtain results when the "wrong prior" is used to make inferences.

More precisely, for the road tracking problem there are many false paths and a single true path. What are the chances of confusing a false path with the true path? This requires us to calculate the probabilities of such events as whether the reward for the true road is greater, or less, than the reward of a false road hypothesis. For the distributions $P(Y|X)$ and $P(X)$ specified by Geman and Jedynak [9], probability bounds for these events can be determined by the use of the theory of types from Information Theory [6] (for more complicated distributions, techniques from large deviation theory can be applied – see Wu and Zhu [23]). The basic result is that the probability of *a specific* false path having higher reward than a true path behaves like 2^{-ND} where N is the path length and D is a function of $P(Y|X)$ and $P(X)$. To bound the probability that *any* false path has greater reward than the true path we must multiply 2^{-ND} by the total number of false paths which can be approximated by Q^N where Q is a constant (this is a simplification – the rigourous proof is in section (5)). For $K > 0$, the probability of confusing a false path with the true path then behaves like 2^{-NK} where $K = D - \log Q$ is the *order parameter*. For $K < 0$ the probability of error becomes high and therefore a phase transition occurs at $K = 0$.

This is illustrated in figure (1) where we give examples of road detection for different values of the order parameter K. The data is generated by first by sampling a road from $P(X)$ and then sampling an image from $P(Y|X)$.

We emphasize that such results can be obtained for problems other than road tracking. Indeed, in section (3) we obtain similar results for three different texture discrimination tasks.

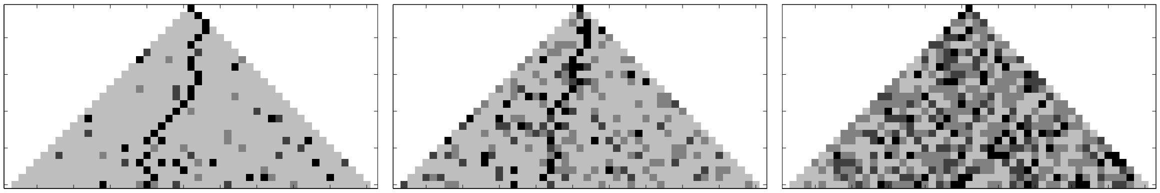


Figure 1: The difficulty of detecting the target path in clutter depends, by our theory, on the order parameter K. The larger K the less computation required. Left, an easy detection task with $K = 0.8647$. Middle, a harder detection task with $K = 0.2105$. Right, an impossible task with $K = -0.7272$.

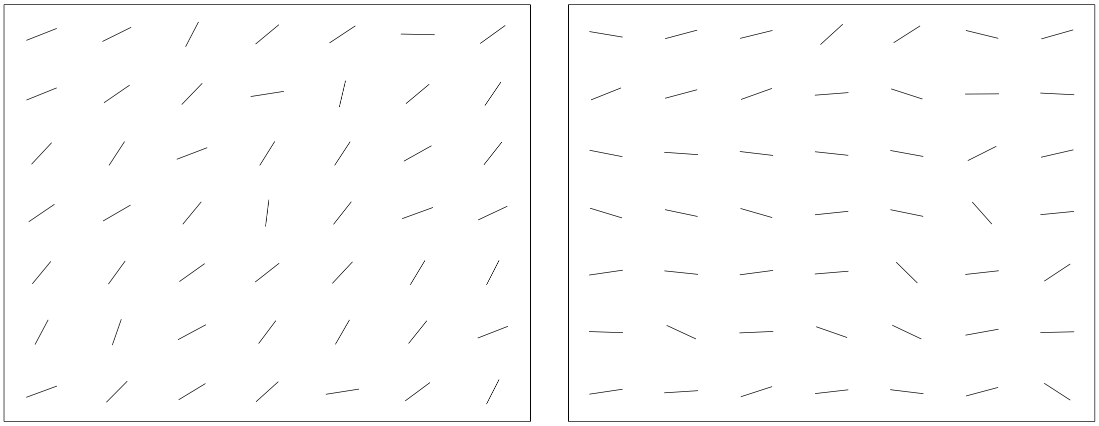


Figure 2: The first texture task: The input is a texture sample. The task is to determine if it came from A, like the texture sample on the left, or from B, like the texture sample on the right.

3 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 $J = 12$ corresponding to a quantization of angles at 15 degrees). These quantized values $a_1, ..., a_J$ 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 vector $\vec{\theta} = (\theta_1, \theta_2, ..., \theta_N)$ of the angles of its edgelets. The optimal tests for our three tasks will depend on the *log-likelihood ratio*. This can be thought of as the MAP estimate between two hypotheses which are equally likely a priori. (See the Neyman-Pearson lemma [6]):

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

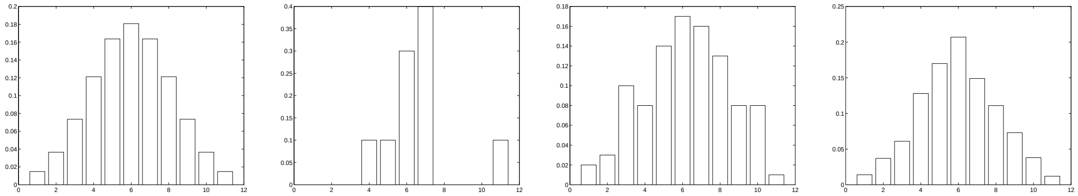


Figure 3: Samples from an underlying distribution. Left to Right, the original distribution, followed by histograms from 10, 100, and 1000 samples from the original. Observe that as the number of samples increases the histograms increasingly resemble the underlying distribution.

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:

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

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 least squared distance between two distributions provided the distributions are very similar. In particular, it is positive 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 [6].

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 θ_i which take values $a_1, ..., a_J$. (i.e. $\phi_\mu = (1/N)\sum_{i=1}^N \delta_{\theta_i, a_\mu}$). Observe, see figure (3), that as the number of samples increases we are likely to get histograms (types) which resemble the underlying distribution. 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:

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

where we have regrouped the terms taking into account the number of responses $N\phi_\mu$ of the θ's at each orientation a_μ.

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 [6]:

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 $P_s(\vec{\theta})$ for any texture $\vec{\theta}$ drawn i.i.d. from a source probability distribution $P_s(\theta)$ 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 $D(\vec{\phi}(\vec{\theta})||P_s)$ between the type and the distribution P_s, and is given by:

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

(The probability of the sequence can be expressed as $\prod_{\mu=1}^J P_s(\mu)^{N\phi_\mu} = 2^{N\sum_{\mu=1}^J \phi_\mu \log P_s(\mu)}$ and we use $H(\vec{\phi}) + D(\vec{\phi}||P_s) = -\sum_{\mu=1}^J \phi_\mu \log P_s(\mu)$ 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 $P(\vec{\phi}) = \sum_{\vec{\theta}} \delta_{\vec{\phi},\vec{\phi}(\vec{\theta})} P_s^N(\vec{\theta})$ and substituting equation (4)).

Lemma 4. We can bound the size of each type class by [6]:

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

(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:

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

From these basic lemmas we can derive the main result we need. We are particularly interested in putting bounds of the probability that a type $\vec{\phi}$ lies within a certain set of 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 (see figure (4)):

Sanov's Theorem. *Let $\theta_1, \theta_2, ..., \theta_N$ be i.i.d. from a distribution $P_s(\theta)$ 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:*

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

where $\vec{\phi}^* = \arg\min_{\vec{\phi} \in E} D(\vec{\phi}||P_s)$ is the distribution in E that is closest to P_s 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 $\vec{\phi}^* = \arg\min_{\vec{\phi} \epsilon E} D(\vec{\phi}||P_s)$. 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 $P_s(head) = P_s(tails) = 1/2$ because the coin is fair. The distribution in E closest to P_s is $P^*(head) = 0.7, P^*(tails) = 0.3$. We calculate $D(P^*||P_s) = 0.119$. 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 $2^{-119} \times (1001)^2 \leq 2^{-99}$.

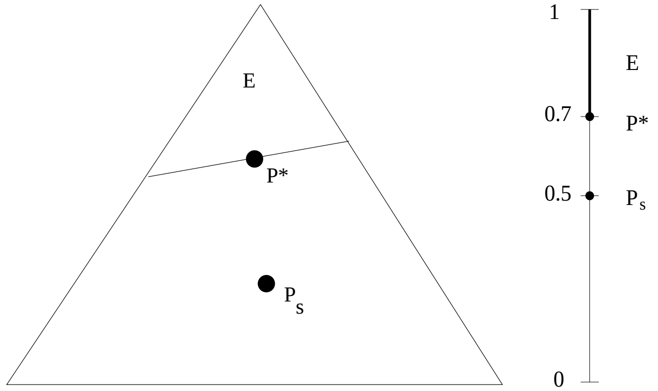


Figure 4: Left, Sanov's theorem. The triangle represents the set of probability distributions. P_s is the distribution which generates the samples. Sanov's theorem states that the probability that a type, or empirical distribution, lies within the subset E is chiefly determined by the distribution P^*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

We note that Sanov's theorem is only one of many techniques for obtaining probability bounds. Another standard technique is motivated by the Central Limit theorem but this gives less tight bounds in general [6] because it only makes use of the variance of the distribution (while Sanov's theorem exploits the whole distribution). Other bounds which do not require the i.i.d. assumption are given in [23],[22].

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, see figure (2), 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 A or B are above, or below, the threshold T are bounded above and below as follows:*

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

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

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 $\vec{\phi}_T \in E_A$ to P_A reduces to constrained minimization using Lagrange multipliers (ν and μ) (the closest distribution must satisfy $\vec{\phi}^A \cdot \vec{\alpha} = T$ since E_A is convex – similarly for B) of the following function:*

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

This can be solved to give $\phi_T(\theta) = P_A^{1-\lambda(T)}(\theta) P_B^{\lambda(T)}(\theta)/Z(T)$ (recall that $\alpha(\theta) = \log\{P_A(\theta)/P_B(\theta)\}$) 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.

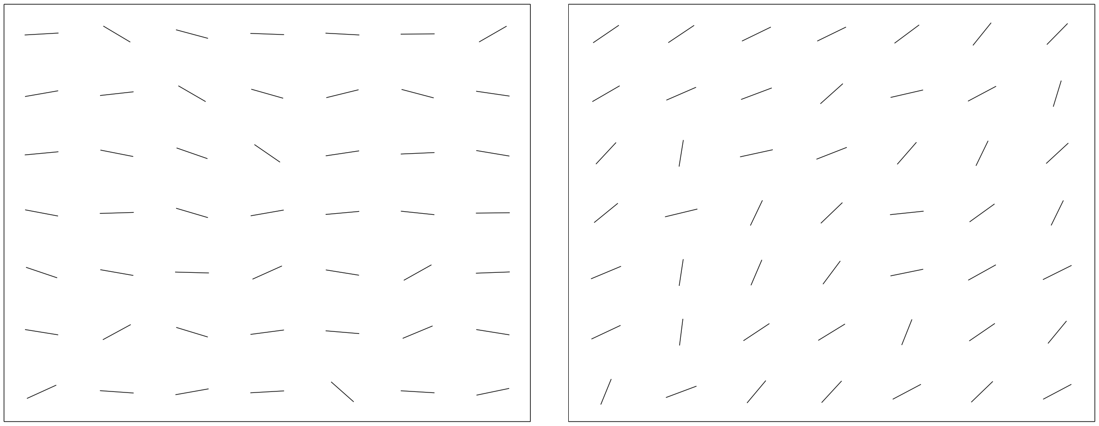


Figure 5: The second texture task: two alternative forced choice. Which texture sample is from A and which one from B?

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 $\vec{\phi}^c$ halfway between P_A and P_B. More precisely, $C(P_A, P_B) = D(\vec{\phi}^c || P_A) = D(\vec{\phi}^c || P_B)$ for the unique distribution $\vec{\phi}^c$, of form $\vec{\phi}_T(\theta) = P_A(\theta)^{1-\lambda(T)} P_B(\theta)^{\lambda(T)}/Z(T)$, which satisfies this constraint.*

Proof. *The error rates fall off as $2^{-ND(\vec{\phi}_T||P_A)}$ and $2^{-ND(\vec{\phi}_T||P_B)}$ where $\vec{\phi}_T$ 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 $D(\vec{\phi}_T||P_A)$ increases and $D(\vec{\phi}_T||P_B)$ decreases. Therefore there is a unique minimum error rate for T such that $D(\vec{\phi}_T||P_A) = D(\vec{\phi}_T||P_B)$, which defines the Chernoff information. Observe that $\sum_\theta \phi^c(\theta) \log P_A(\theta)/P_B(\theta) = 0$, hence $T = 0$ is the asymptotic error rate.*

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 $\vec{\phi}_T = P_B$ 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, see figure (5), has two texture samples as input (one each from A and B) and the task is to classify them correctly. The best decision rule is 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. 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 *(i.e.* $D((\vec{\phi}^A, \vec{\phi}^B)||(P_A, P_B)) = D(\vec{\phi}^A||P_A) + D(\vec{\phi}^B||P_B)$*). This requires us to minimize:*

$$f(\vec{\phi}^A, \vec{\phi}^B) = D(\vec{\phi}^A||P_A) + D(\vec{\phi}^B||P_B) + \\ +\tau_1\{\sum_\mu \phi^A(\mu) - 1\} + \tau_2\{\sum_\mu \phi^B(\mu) - 1\} + \gamma\{\vec{\phi}^A \cdot \vec{\alpha} - \vec{\phi}^B \cdot \vec{\alpha}\}, \quad (12)$$

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:*

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

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* $\vec{\phi}_{Bh} = P_A^{1/2} P_B^{1/2}/Z[1/2]$. *We therefore obtain:*

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

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, see figure (6). 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:*

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

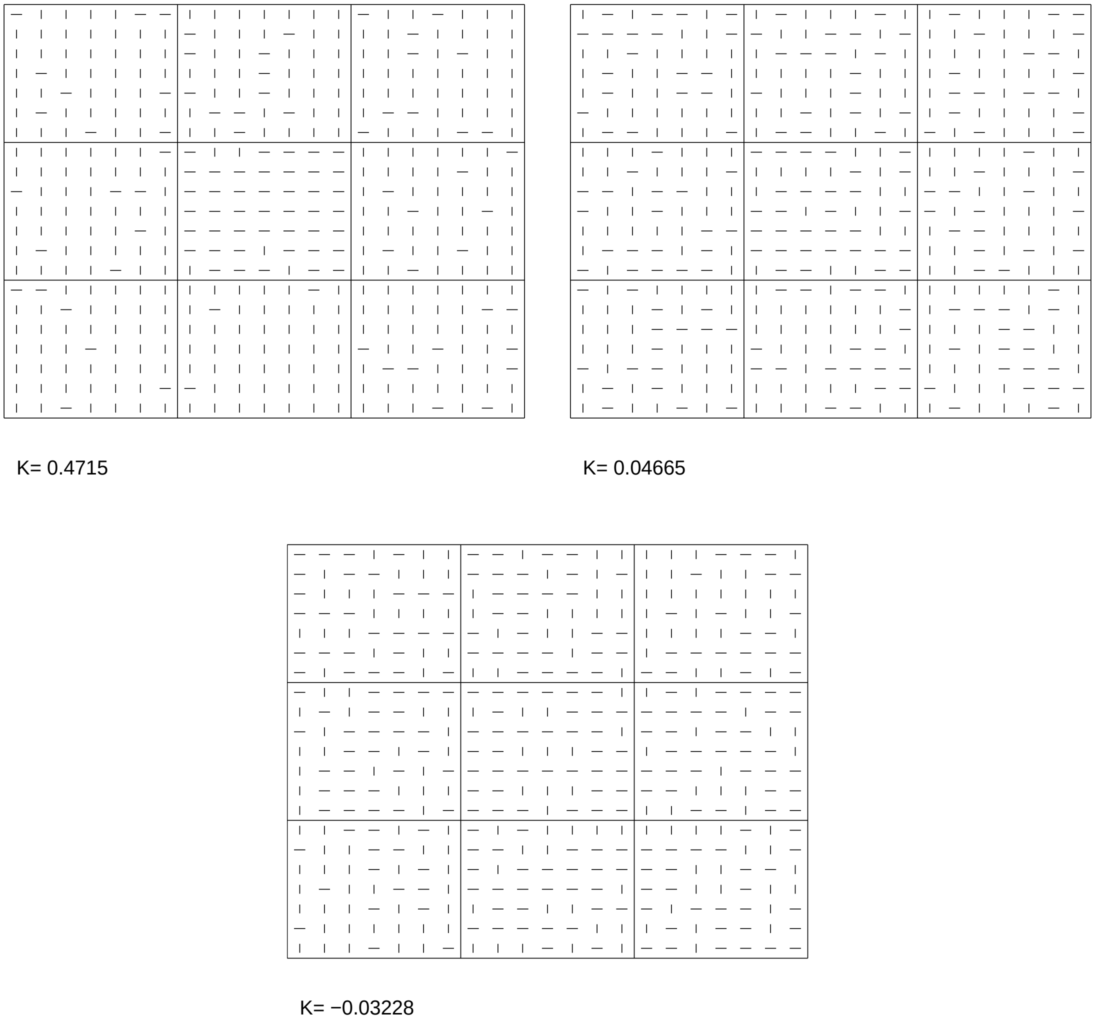
For large N, the bounds are determined by $K = 2B(P_A, P_B) - \log Q$. *If* $K > 0$ *the expected number of B texture samples tends to zero as* $N \mapsto \infty$. *For* $K < 0$, *it tends to* ∞.

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 at least one 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:* $Pr(A_1$ *or* A_2 *or ... 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



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.

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 [1] $B(P_A, P_B) = (1/2)\{D(\vec{\phi}_{Bh}||P_B) + D(\vec{\phi}_{Bh}||P_A)\}$ 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 for any P_A, P_B can be readily verified, see [6]:

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

4 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 [9] 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 [8]. 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 for road tracking and snakes. We follow the derivation of Geman and Jedynak [9] 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 [25], [5].

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 and direction 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 $P_G(x_{i+1}|x_i) = P_{\Delta G}(x_{i+1} - x_i)$. 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 non-linear filter to detect arcs of road. The intuition 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 $P_{on}(Y_a = y_a) = P(Y_a = y_a|\text{a on X})$ and $P_{off}(Y_a = y_a) = P(Y_a = y_a|\text{a off X})$. 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 $P(X|Y) = P(Y|X)P(X)/P(Y)$. Using the prior and likelihood function above, we take logarithms, and drop the constant terms, giving:

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

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

[1] This Bhattacharyya distance arises in the Bhattacharyya bound for error rates [17].

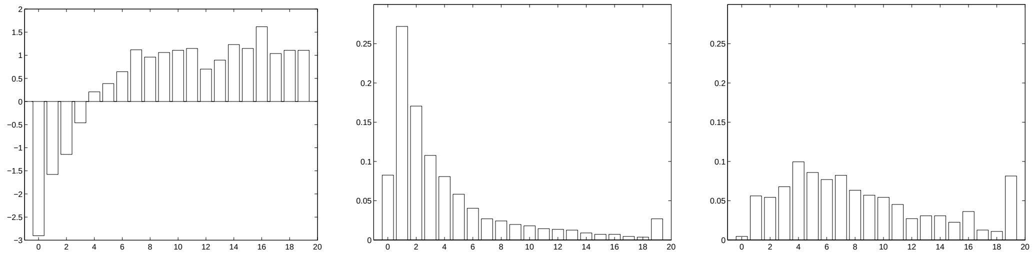


Figure 7: The log likelihood ratios (far left) of the off-edge probabilities $p_{off}(y)$ (center) and the on-edge probabilities $p_{on}(y)$(right), where $y = \left|\vec{\nabla} I\right|$. These distributions, and ratios, were very consistent for a range of images. The filter responses y, on the horizontal line, were quantized to take 20 values.

We now consider the alternative formulation of Geiger and Liu based on snakes [12]. 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 $\{\vec{x}(t) : 0 \leq t \leq 1\}$: $E[x(t)] = \lambda \int_{t=0}^{1} dt \, |ds/dt| + \mu \int_{t=0}^{1} dt \kappa^2(t) \; -\nu \int_{t=0}^{1} \left|\vec{\nabla} I(x(t)\right|$ [12]. 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 [12]. 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 [9]. We performed statistical analysis on a range of images (having first located the edges by hand) and obtained empirical results shown in figure (7). The general shapes of the P_{on}, P_{off} and their log likelihood ratio are very similar from image to image[2]. The log-likelihood terms clearly show the thresholding effect argued for above. We should add that Geiger and Liu [8] used a modification of snakes which makes their likelihood terms much more similar to ours than to those used in the original snake model [12].

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 [30] investigates the effectiveness 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

[2] These plots of P_{on}, P_{off} are also somewhat similar to those observed by Balboa and Grzywacz [2] who obtained edge statistics in a variety of domains in order to model the retinal receptive fields of animals. A detailed discussion of these issues will be given in a forthcoming paper [14].

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:

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

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 [16],[18], which we describe and analyze in [25],[5]. 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 estimate of the addition reward to get to the finish from n. Both Geman and Jedynak's and Geiger and Liu's algorithms can be shown [24] 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).

5 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 (3). It transpires that only simple modifications of those theorems will be sufficient to obtain our results.

We define the problem on a Q-nary tree with the prior conditional 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 [9],[8] 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 $E(X)$ 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 (3). 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 $B(P_{on}, P_{off}) = -\log\{\sum_{\mu} P_{off}^{1/2}(\mu) P_{on}^{1/2}(\mu)\}$. 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:

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

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 $\vec{\phi}^{off*} \cdot \vec{\alpha} = \vec{\phi}^{on*} \cdot \vec{\alpha}$ 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 $\vec{\psi}_{Bh} = \vec{\psi}^{off*} = \vec{\psi}^{on*}$. We define $B(P_{on}, P_{off}) = (1/2)\{D(\vec{\phi}_{Bh} || P_{off}) + D(\vec{\phi}_{Bh} || P_{on})\} = -\log\{\sum_{\mu} P_{off}^{1/2}(\mu) P_{on}^{1/2}(\mu)\}$ (this last equality can be verified by substituting for $\vec{\phi}_{Bh}$) and $B(U, P_{\Delta G})$ analogously). This yields:

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

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

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

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:

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

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.

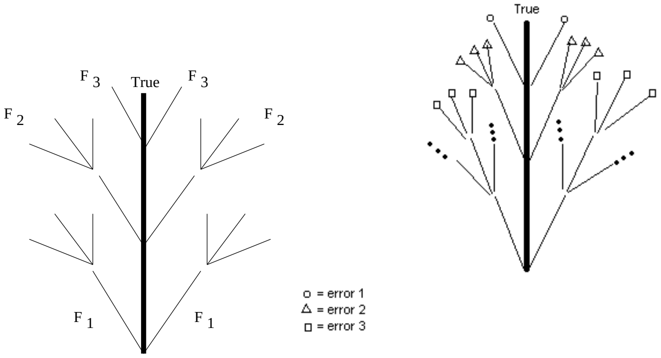


Figure 8: Left: We can divide the set of paths up into N subsets $F_1, ..., F_N$ as shown here. Paths in F_1 are completely off-road. Paths in F_2 have one on-road segment and so on. Intuitively, we can think of this as an onion where we peel off paths stage by stage. Right: When paths leave the true 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 $N+1-i$.

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 [9], 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 [5] 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 which have high reward.

5.1 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 true road for the path with biggest total reward?

A key concept here is the onion-like structure of the tree representation, see figure (8). 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. (Variants of this argument will be used throughout the rest of this section.)

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 (8). 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 with N (of course, close to the phase transition we will expect many mixed paths to have rewards close to 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[3] 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, $\hat{\Xi}(\epsilon, K, J^2Q^2)$, is an additional term used to deal with the alphabet factors in the regime where $M < M_0$.

This gives:

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

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

This proves our main result:

Theorem 5. *The path with highest reward is expected to diverge from the best path by less than* $\frac{2^{-(K-\epsilon)}}{(1-2^{-(K-\epsilon)})^2} + \hat{\Xi}(\epsilon, K, J^2Q^2)$ *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.*

6 Discussion

Similar techniques can be used to analyze the performance of A* algorithms to solve the road tracking problem and we have obtained time and memory complexity results [25],[5] which depend on closely related order parameters. In particular, we study: (i) an admissible A* algorithm which uses pruning and (ii) an inadmissible A* algorithm. In both cases we prove expected convergence rates with O(N) node expansions (where N is the problem size) and also expected constant time sorting per node expansion. The results again involve putting probability bounds on events such as the possibility that the algorithm wastes a lot of time and memory exploring a false branch of the search tree.

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

The analysis of this current paper can also be generalized to deal with situations in which the Bayesian models use the "wrong priors" which are only approximations to the true underlying distributions [26]. How much harder do we make target detection by using a weaker model for inference? (For example, it may be unrealistic to assume that we know the true distribution completely accurately). As shown in [26], we can determine order parameters when the "wrong priors" are used for inference and hence determine regimes (specified by the order parameters) in which the use of a wrong prior will not significantly affect the accuarcy of the inference. In other regimes, however, the wrong prior will not be sufficiently informative to make the correct inference (although the correct inference can be made if the true prior is known). One can think of these results, informally, as determining when one can get away with using dumb algorithms!

We now address limitations of the current models and techniques. A major limitation of the techniques used in this paper is that Sanov's theorem can only be applied to analyze i.i.d. samples, and hence restricts the class of Bayesian models that we can study. This limitation, however, can be overcome [23] by using more powerful techniques from large deviation theory [22]. As shown in [23] it then becomes possible to obtain order parameters for any shift-invariant Markov random fields such as extremely realistic texture models. Current resarch by the authors has also established order parameters for other probability distributions which are not shift-invariant. This is an active research area.

These stronger techniques will enable us to analyze more realistic Bayesian models. For this paper, we selected the Geman and Jedynak model [9] partially because of its effectiveness on real images. It does, however, contain a few limitations. One of these may be the assumption that the results of edge tests on the road are independent. In practice, however, there may be correlations between the edge tests at neighbouring locations on the roads. This is an empirical questions which can only be answered by analysis of datasets, see [14]. In any case, if local correlations exist then it is highly likely that they can be modelled by a local Markov distribution – in which case, the large deviation techniques in [22],[23] can be used to obtain order parameters. (We note that Geman and Jedynak [9] performed their edge tests on arcs of length 10-12 pixels and made a plausible argument that neighbouring arcs were only weakly correlated.) Similarly, we might prefer a more realistic distribution of the edge responses off the road which could be specified by a model such as Zhu and Mumford's [28]. Such a model could also be analyzed using large deviation theory results.

7 Conclusion

This paper examined the fundamental limits of performing certain forms of Bayesian inference on images. It was shown that the behaviour of the MAP estimator 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. In particular, the entropy of the geometric prior and the Bhattacharyya bound [17] between P_{on} and P_{off} allow us to quantify intuitions about the power of geometrical assumptions and edge detectors to solve raod and contour tracking tasks.

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.

As mentioned in the discussion, see section (6), the techniques used in this paper can also be applied to analyze the time and memory complexity of A* tree search for road tracking. They can also help quantify the cost of performing Bayesian inference using a "wrong prior" which is only an approximation to the true underlying probability distribution.

Finally, it is very encouraging that recent work [23] has shown that more powerful techniques from the theory of large deviations [22] can be applied to calculate order paramaters for a large class of Bayesian models including very realistic models for texture sysnthesis and discrimination.

Appendix

We need to bound sums such as:

$$\sum_{m=0}^{\infty} m 2^{-Bm} (m+1)^A. \tag{27}$$

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:

$$\sum_{m=0}^{\infty} m 2^{-(B-\epsilon)m} + \hat{\Xi}(\epsilon, A, B), \tag{28}$$

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:

$$\sum_{m=0}^{\infty} m 2^{-Bm} (m+1)^A = \frac{2^{-B}}{(1-2^{-B})^2} + \hat{\Xi}(\epsilon, A, B). \tag{29}$$

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 and Song Chun Zhu gave very useful feedback and encouragement.

References

[1] D.J. Amit. **Modeling Brain Function**. Cambridge University Press. 1989.

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

[3] P. Cheeseman, B. Kanefsky, and W. Taylor. "Where the Really Hard Problems Are". In *Proc. 12th International Joint Conference on A.I.*. Vol. 1., pp 331-337. Morgan-Kaufmann. 1991.

[4] 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.

[5] J.M. Coughlan and A.L. Yuille. "Bayesian A* Tree Search with Expected O(N) Convergence Rates for Road Tracking". In *Proceedings EMMCVPR'99.* Springer-Verlag. York, England. 1999.

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

[7] M.R. Garey and D.S. Johnson. **Computers and Intractability: A Guide to he Theory of NP-Completeness**. W.H. Freeman and Co. New York. 1979.

[8] 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.

[9] 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.

[10] M. Isard and A. Blake. "Contour tracking by stochastic propagation of conditional density". *Proc. Europ. Conf. Comput. Vision*, pp. 343-356, Cambridge, UK. 1996.

[11] 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.

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

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

[14] S. M. Konishi, A.L. Yuille, J.M. Coughlan and Song Chun Zhu. "Fundamental Bounds on Edge Detection: An Information Theoretic Evaluation of Different Edge Cues." In *Proceedings Computer Vision and Pattern Recognition CVPR'99.* Fort Collins, Colorado. 1999.

[15] P. Parodi, R. Lancewicki, A. Vijh, and J.K. Tsotsos. "Empirically-derived estimates of the complexity of labeling line drawings of polyhedral scenes". *Artificial Intelligence* (105) 1-2. pp 47-75. 1998.

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

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

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

[19] B. Selman and S. Kirkpatrick. "Critical Behaviour in the Computational Cost of satisfiability Testing". Artificial Intelligence. 81(1-2); 273-295. 1996.

[20] J.K. Tsotsos. "Analyzing vision at the complexity level". *Behavioural and Brain Sciences.* Vol. 13, No. 3. September. 1990.

[21] J.K. Tsotsos. "On the Relative Complexity of Active versus Passive Visual Search". *International Journal of Computer Vision.* Vol. 7., No. 2. January 1992.

[22] J.T. Lewis, C.E. Pfister, and W.G. Sullivan. "Entropy, concentration of probability, and conditional limit theorems". *Markov Processes Relat. Fields.* Vol. 1. pp 319-396. 1995.

[23] Y.N. Wu and S.C. Zhu. "Equivalence of Ensembles and Fundamental Bounds". To appear in *Proceedings of the International Conference on Computer Vision.* Corfu, Greece. 1999.

[24] 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.

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

[26] A.L. Yuille and J.M. Coughlan. 'High-Level and Generic Models for Visual Search: When does high level knowledge help?". In *Proceedings Computer Vision and Pattern Recognition CVPR'99.* Fort Collins, Colorado. 1999.

[27] 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.

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

[29] 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.

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

Biography

Biography for Dr. A.L. Yuille.

Alan Yuille received his BA in Mathematics at the University of Cambridge in 1976. He completed his PhD in Theoretical Physics at Cambridge in 1980 and worked as a postdoc in Physics at the University of Texas at Austin and the Institute for Theoretical Physics at Santa Barbara. From 1982-86 he worked at the Artificial Intelligence Laboratory at MIT before joining the Division of Applied Sciences at Harvard from 1986-1995 rising to the rank of Associate Professor. In 1995 he joined the Smith-Kettlewell Eye Research Institute in San Francisco. His research interests are in mathematical modelling of artificial and biological vision. He has over one hundred peer-reviewed publications in vision, neural networks, and physics. He has co-authored two books – ”Data Fusion for Sensory Information Processing Systems” J.J. Clark and A.L. Yuille, and ”Two- and Three- Dimensional Patterns of the Face” P.W. Hallinan, G.G. Gordon, A.L. Yuille, P.J. Giblin and D.B. Mumford – and edited a book ”Active Vision” with A. Blake.

Biography for Dr. James M. Coughlan.

James Coughlan received his B.A. in physics at Harvard University in 1990 and completed his Ph.D. in physics there in 1998. He is currently working as a post-doctoral fellow with Alan Yuille at the Smith-Kettlewell Eye Research Institute in San Francisco. His research interests are in computer vision and the applications of Bayesian probability theory to artificial intelligence. He has published papers on theoretical and experimental issues in deformable templates and the detection of targets in clutter, interpreting the layout of three-dimensional scenes, estimating optical flow, and learning theory.