International Journal of Computer Vision 41(1/2), 9–33, 2001

Order Parameters for Detecting Target Curves in Images: When Does High Level Knowledge Help?

A.L. YUILLE AND JAMES M. COUGHLAN
Smith-Kettlewell Eye Research Institute, 2318 Fillmore Street, San Francisco, CA 94115, USA
yuille@ski.org
coughlan@ski.org

YINGNIAN WU
Department of Statistics, University of California at Los Angeles, Los Angeles, CA 90095, USA
ywu@math.ucla.edu

SONG CHUN ZHU
Department of Computer and Information Sciences, The Ohio State University, Columbus, OH 43210, USA
szhu@cis.ohio-state.edu

Received November 12, 1999; Revised August 31, 2000; Accepted December 26, 2000

Abstract. Many problems in vision can be formulated as Bayesian inference. It is important to determine the accuracy of these inferences and how they depend on the problem domain. In this paper, we provide a theoretical framework based on Bayesian decision theory which involves evaluating performance based on an ensemble of problem instances. We pay special attention to the task of detecting a target in the presence of background clutter. This framework is then used to analyze the detectability of curves in images. We restrict ourselves to the case where the probability models are ergodic (both for the geometry of the curve and for the imaging). These restrictions enable us to use techniques from large deviation theory to simplify the analysis. We show that the detectability of curves depend on a parameter K which is a function of the probability distributions characterizing the problem. At critical values of K the target becomes impossible to detect on average. Our framework also enables us to determine whether a simpler approximate model is sufficient to detect the target curve and hence clarify how much information is required to perform specific tasks. These results generalize our previous work (Yuille, A.L. and Coughlan, J.M. 2000. *Pattern Analysis and Machine Intelligence* PAMI, 22(2):160–173) by placing it in a Bayesian decision theory framework, by extending the class of probability models which can be analyzed, and by analysing the case where approximate models are used for inference.

Keywords: Bayesian inference, curve detection, order parameters, minimax entropy

1. Introduction

This paper is concerned with determining the fundamental limits of visual inference and quantifying what aspects of a visual task make it easy or hard. An important related question is how much prior knowledge do we need about the task in order to solve it. Intuitively, if a visual task is easy then we will only need to use a simple model to solve it. But a more difficult task may require a sophisticated model which uses a lot of knowledge about the specific task.

Figure 1. Left to right, three detection tasks of increasing degrees of difficulty. The stop sign (left) is easy to find. The gila monster (centre) is harder. The dalmation dog (right) is almost impossible.

For example, consider the tasks of detecting the three target objects—stop sign, gila monster, and dalmation dog—from the images in Fig. 1. Intuitively, detecting the stop sign in the left panel is far easier than detecting the dalmation dog in the right panel. But can we quantify the relative difficulties of these tasks? And can we determine what precise aspects of the image and the targets makes the task easy or hard? For example, it seems likely that the difficulty of detecting the gila monster (centre panel) is because the texture of the target is very similar to the texture of the background. Finally, how much knowledge do we need about the targets and the background in order to solve the tasks? Intuitively, a simple edge detector followed by spatial grouping (e.g. a Hough transform) might be sufficient to detect the stop sign (left panel) but, by contrast, it seems impossible to detect the dalmation dog (right panel) without knowing something about the shape and texture of dalmations.

At a more practical level, at least two researchers (private communications) have been impressed with the theory of road tracking developed by Geman and Jedynak (1996) but have been unable to get this algorithm to work on the domains they are interested in. An important consequence of the analysis we perform is that *we are able to specify precisely when this algorithm will work and when it will not based on the statistical properties of the domain*. Moreover, our theory will also help determine how to modify the Geman and Jedynak algorithm, to ensure that it does work, by building into it additional knowledge of the domain.

To address these issues, we formulate visual tasks as Bayesian inference, see Knill and Richards (1996), using Bayesian decision theory (DeGroot, 1970). This gives us the necessary concepts for quantifying the difficulty of visual tasks and for determining fundamental limits by means of the *Bayes risk*. Indeed, as reviewed by the authors (Yuille, Coughlan, Zhu, 2000), most work on performance analysis of visual algorithms is either explicitly, or implicitly, formulated in these terms, see Fig. 2. This includes Cramer-Rao bounds (Young and Chellappa, 1992; Barron et al., 1994;

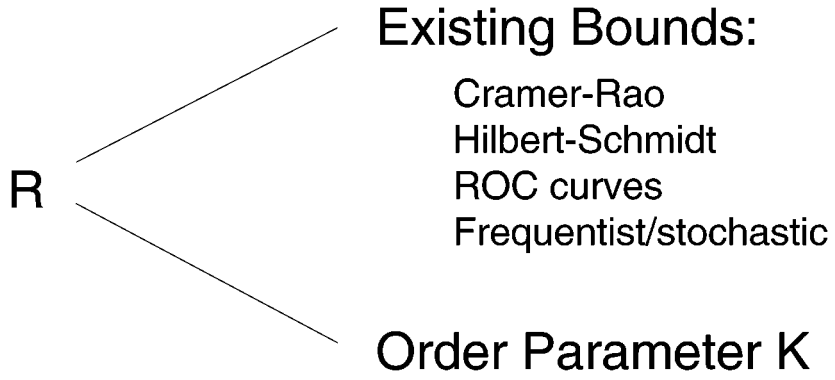

Figure 2. Decision theory gives performance bounds in terms of the Bayes risk R. Many existing performance bounds can be expressed in these terms. In this paper, we analyze the Bayes risk for detecting a target curve in clutter and show it depends on an order parameter K.

Szeliski and Kang, 1997; Rajagopalan and Chaudhuri, 1998) Hilbert-Schmidt bounds (Grenander et al., 1998), frequentist empirical analysis (Hoover et al., 1996; Heath et al., 1997; Bowyer and Phillips, 1998; Konishi et al., 1999), and order parameters (Yuille and Coughlan, 1999, 2000). In addition, related techniques from signal detection theory (Green and Swets, 1988), such as the receiver operator characteristic (ROC) curves have also been used for image analysis and ATR (Ratches et al., 1997; Bowyer and Phillips, 1998).

In this paper, we use Bayesian decision theory to analyse the performance of models for curve (or road) detection. We assume that the probability models are ergodic (Cover and Thomas, 1991) so that techniques from large deviation theory (Dembo and Zeitouni, 1998) can be used to simplify the analysis. These techniques can also be applied to analyze related problems such as texture discrimination (Zhu et al., 1997; Wu et al., 2000).

We derive a parameter K whose value characterizes the difficulty of the problem. (K is computed from the probability distributions which describe the problem). At critical values of this parameter it becomes almost impossible to detect the target because it will be confused by all the curves in the background image clutter. It becomes like looking for a needle in a haystack. The point is that the chances of confusing a *specific* background curve with the target curve are very small. But there are so many background curves that it is possible that one of them may be confused with the target curve. The precise theoretical results are stated in Sections 3 and 4. They apply in the limit as the size N of the target curve tends to infinity and they ignore curves which are partially on and partially off the target curve. In some conditions, we can prove mathematically that the Bayes risk has a jump from 0 (perfect detectability) to 1 (perfect indetectability) as the parameter K passes through a critical value. In other cases, we prove a weaker result that the *expected number* of background clutter curves which can be confused with the target curve becomes infinite at this critical value of K. We then use computer simulations to show that the Bayes risk does jump from 0 to 1 at this critical value. We refer to this informally as a *phase transition* by analogy to statistical physics (A phase transition is "a qualitative change in the dynamical properties of a system of many degrees of freedom due to a change of externally controlled parameters" Amit, 1989).

In addition, we analyze what happens if we attempt to perform Bayes inference using a simpler approximate model. An approximate model may be used because: (i) we may not know the correct models, or (ii) it may be more computationally efficient (i.e. quicker) to use an approximate model. In this case, *we are concerned with how much prior knowledge about the target is required in order to detect it*. Some detection tasks, see Fig. 1, are far more difficult than others depending on the different statistical properties of the target and the background. For some of these tasks low-level general purpose algorithms will be sufficient to segment the target from the background but other tasks, such as the dalmation dog, appear to require high-level knowledge about dogs. Our theoretical analysis shows that the parameters K change as we use simpler models. We concentrate our study on a specific form of approximation, motivated by Minimax Entropy learning theory (Zhu et al., 1997), and compute explicitly the change of K. This helps us determine how much prior knowledge about the target is required in order to detect it. (Our preliminary results on this topic were presented in a conference proceedings (Yuille and Coughlan, 1999) and applied only to factorizable distributions.)

In a previous paper (Yuille and Coughlan, 2000) we address different aspects of the same problem for the special case of the Geman and Jedynak model (Geman and Jedynak, 1996) for detecting roads, see Section 3. We exploited the factorizability of the Geman and Jedynak model to put tight bounds on the probability of successful detection of road targets of *finite size* N. In addition, we were able to provide analysis which included paths that were partially on and partially off the target road (although this analysis included assuming a tree representation which has some limitations, see Section 3). In particular, we showed that many properties of the tasks such as detectability (Yuille and Coughlan, 2000) and expected complexity (Coughlan and Yuille, 1999) fell off exponentially as 2^{-NK} where N is the length of the target road and K is a parameter. See Section 3 for a more detailed description of how this previous work overlaps with this paper.

In the next Section 2, we briefly review Bayesian decision theory and describe how it can be applied to problems such as target detection. Section 3 describes the Geman and Jedynak model for road tracking (Geman and Jedynak, 1996), briefly summarizes our previous results (Yuille and Coughlan, 2000) and then extends the analysis to deal with situations where we use an approximate model to detect the road and to determine how much information is required to solve the task. In Section 4 we extend the analysis to deal with a more

general class of probability models, including those learnt by Minimax Entropy learning (Zhu et al., 1997), and obtain similar results for how much information is required to solve the curve detection task.

2. Bayesian Decision Theory

Image analysis, like all inference problems, can be expressed in terms of Bayesian decision theory. In subsection 2.1 we briefly review decision theory (DeGroot, 1970) and in subsection 2.2 we apply it to target detection.

2.1. Decision Theory

There is a set $d \in D$ decisions, a set of observations $\mathbf{z} \in Z$ and a set of states $s \in S$ of the system observed. We have a prior distribution $P(s)$, a likelihood function $P(\mathbf{z}|s)$, and a loss function $l(d, s)$ (without loss of generality we assume that the loss functions never take negative values). For any observation $\mathbf{z}$, the risk (i.e. expected loss) is:

$$R(d; \mathbf{z}) = \int ds\, l(d, s) \frac{P(\mathbf{z} \mid s) P(s)}{P(\mathbf{z})}. \qquad (1)$$

For a set of observations drawn from $P(\mathbf{z})$, we define a decision rule $d = c(\mathbf{z})$. The risk of the decision rule involves averaging over the observations $\mathbf{z}$ with respect to $P(\mathbf{z})$. The expected risk is the loss averaged over all states s and observations $\mathbf{z}$:

$$R(c) = \int d\mathbf{z}\, ds\, l(c(\mathbf{z}), s) P(s, \mathbf{z}) P(s). \qquad (2)$$

Note that this average is taken with respect to the joint distribution $P(s, \mathbf{z})$, which we term the *Bayes Ensemble*, or distribution over all problem instances.[1] The strength of Decision Theory is that it allows us to determine the *typical* performance of inference procedures by averaging over all problem instances of the Bayes Ensemble, rather than focusing on worst-case performance measures, which may seldom be relevant in practice.

The Bayes estimator c^* is chosen to minimize the risk $R(c)$. The Bayes risk is $R(c^*)$ and is a natural performance measure for visual tasks. Note that, provided weak technical conditions are satisfied, the *Bayes risk* is obtained by minimizing equation (1) separately for all d and $\mathbf{z}$. Intuitively, if the Bayes risk is low then the visual task is easy. In this paper we concentrate on classification tasks, where each observation $\mathbf{z}$ contains a single target and multiple distractors, and the loss function takes value 1 if the target is misclassified and is 0 if the target is correctly classified. In some situations, however, it may be impractical to use the Bayes estimator (e.g. it may be impossible to compute) so we may instead compute the expected loss of a different (usually more easily computable) estimator.

Most performance measures used to evaluate visual algorithms can be interpreted as being the Bayes risk (once the problem has been framed in these terms). In other words, the problem is formulated as Bayesian inference with state variables, probability distributions and loss functions. The best estimator c^* is found and the Bayes risk evaluated. For classification problems the Bayes risk will correspond to the misclassification rate (e.g. false positives and false negatives sometimes with certain errors weighted more highly than others).

Calculating the Bayes risk is often impossible to do analytically. In the cases we study in this paper, self-averaging (or ergodic) properties of the probability distributions makes it possible to estimate the Bayes risk for large systems.

We are also interested in how performance (i.e. the risk) is degraded by using the wrong probability distributions for inference (e.g. because the true probability models are unknown). This means that we will not use the optimal decision rule (because we will pick the decision rule appropriate to the wrong distributions) and hence our performance will be worse than the Bayes risk. Intuitively, small errors in the distributions will only change the decision rule slightly and hence will cause performance to degrade by a small amount. A standard result, the concavity of the Bayes risk, formalizes this intuition (DeGroot, 1970). However, a more important situation arises when a simplified probability distribution (which may be significantly different from the true distribution) is deliberately used for computational purposes, see Sections 3.4 and 4.5.

2.2. Discriminating A from B

The first task is to determine whether a sample $\mathbf{z}$ has been generated by one of two distributions $P_A(\mathbf{z})$ or $P_B(\mathbf{z})$. We assume that the sample is equally likely to be generated by model A or model B. The penalty for misclassification is symmetric so that we pay penalty 1 if a sample from A is misclassified as B and vice versa. We pay no penalty if the sample is correctly

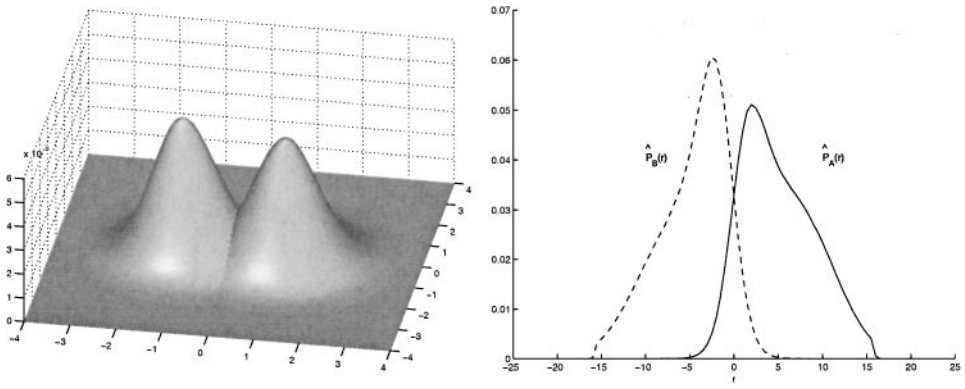

Figure 3. Discriminating between P_A and P_B. Left panel, the distributions $P_A(\vec{z})$ and $P_B(\vec{z})$ where $\vec{z}$ is a two-dimensional vector. Right panel, plots of the induced distributions $\hat{P}_A(r)$ (solid line) and $\hat{P}_B(r)$ (dashed line) as functions of the log-likelihood ratio (or reward) $r(\vec{z}) = \log\{P_A(\vec{z})/P_B(\vec{z})\}$. The induced distributions provide a complete description of the problem of discriminating P_A and P_B. Note that this description is in terms of the log-likelihood ratio, which means that the discrimination problem has been reduced to *one dimension* regardless of the dimensionality of $\vec{z}$. The greater the overlap between $\hat{P}_A(r)$ and $\hat{P}_B(r)$, the greater the misclassification rate.

classified. The optimal decision, given these assumptions, is to use the likelihood ratio test and classify the sample as A if $\log\{P_A(\mathbf{z})/P_B(\mathbf{z})\} > 0$ and as B if $\log\{P_A(\mathbf{z})/P_B(\mathbf{z})\} < 0$. The Bayes risk R^* is then given by summing the probabilities that a sample $\mathbf{z}$ is generated by one distribution $P_A(\cdot)$ or $P_B(\cdot)$ but is misclassified as being generated by the other. More precisely:

$$R^* = \frac{1}{2}\int_{\{\mathbf{z}:\log\{P_A(\mathbf{z})/P_B(\mathbf{z})\}>0\}} d\mathbf{z}\; P_B(\mathbf{z}) + \frac{1}{2}\int_{\{\mathbf{z}:\log\{P_A(\mathbf{z})/P_B(\mathbf{z})\}<0\}} d\mathbf{z}\; P_A(\mathbf{z}). \quad (3)$$

We can re-express the Bayes risk completely in terms of the log-likelihood ratio $r(\mathbf{z}) = \log\{P_A(\mathbf{z})/P_B(\mathbf{z})\}$, which we also refer to as the *reward function*. The distributions $P_A(\mathbf{z})$ and $P_B(\mathbf{z})$ *induce* distributions on r given by the formulas:

$$\hat{P}_A(r) = \int d\mathbf{z}\; P_A(\mathbf{z})\delta\left(r - \log\frac{P_A(\mathbf{z})}{P_B(\mathbf{z})}\right),$$
$$\hat{P}_B(r) = \int d\mathbf{z}\; P_B(\mathbf{z})\delta\left(r - \log\frac{P_A(\mathbf{z})}{P_B(\mathbf{z})}\right). \quad (4)$$

The induced distributions $\hat{P}_A(r)$ and $\hat{P}_B(r)$ provide a complete description of the problem of *discriminating* $P_A(\mathbf{z})$ from $P_B(\mathbf{z})$ (although there are many possible choices of $P_A(\mathbf{z})$ and $P_B(\mathbf{z})$ which give rise to the same induced distributions $\hat{P}_A(r)$ and $\hat{P}_B(r)$). For instance, $\hat{P}_A(r)$ and $\hat{P}_B(r)$ uniquely determine the ROC curve for discriminating $P_A(\mathbf{z})$ from $P_B(\mathbf{z})$ (Yuille, Coughlan, Zhu, 2000), see Fig. 3.

It is straightforward to show that $\log(\hat{P}_A(r)/\hat{P}_B(r)) = r$ for all r. The Bayes risk may then be re-expressed in terms of log-likelihood space:

$$R^* = \frac{1}{2}\int_0^\infty dr\; \hat{P}_B(r) + \frac{1}{2}\int_{-\infty}^0 dr\; \hat{P}_A(r). \quad (5)$$

2.3. *Target in Clutter*

To detect a target in clutter, the task is to determine which of $M+1$ samples $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_M$ is the target. Without loss of generality we assume that $\mathbf{z}_0$ is the target so it is generated by $P_A(\mathbf{z}_0)$ and the background $\mathbf{z}_1, \ldots, \mathbf{z}_M$ is generated by the background distribution $P_B(\mathbf{z}_1, \ldots, \mathbf{z}_M)$. Observe that we are assuming that the background samples $\mathbf{z}_1, \ldots, \mathbf{z}_M$ are *not* necessarily independent. We do, however, assume that all the distractors have the same marginal distribution $P_B(\mathbf{z})$. (i.e. $\sum_{\{\mathbf{z}_i:i\neq j\}} P_B(\mathbf{z}_1, \ldots, \mathbf{z}_M) = P_B(\mathbf{z}_j)$ for all $j = 1, \ldots, M$.)

Once again, the expected loss R^* will be determined by the misclassification rate. We define the loss to be 0 when sample A is correctly identified and to be 1 otherwise. The optimal decision rule (assuming a uniform prior on which of the $M+1$ samples comes from A) is to estimate that A generates the sample i^* given by:

$$i^* = \arg\max_{i=0,\ldots,M} \log\frac{P_A(\mathbf{z}_i)}{P_B(\mathbf{z}_i)} = \arg\max_{i=0,\ldots,M} r(\mathbf{z}_i). \quad (6)$$

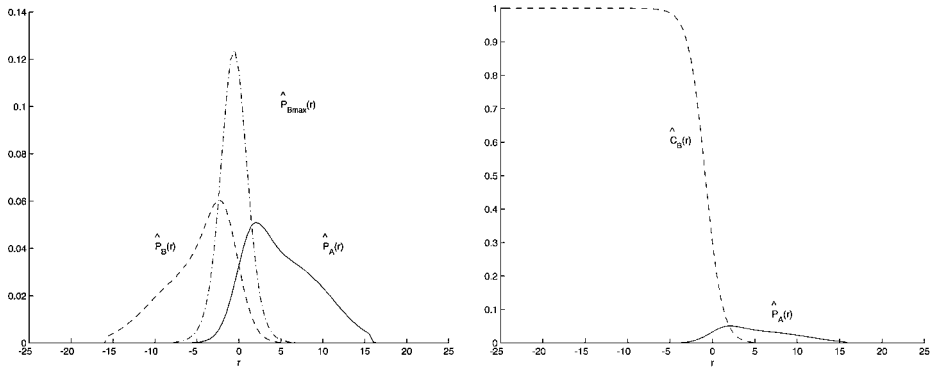

Figure 4. Target in clutter: discriminating one sample of P_A from many samples of P_B. Left panel, the induced distributions $\hat{P}_A(r)$ and $\hat{P}_B(r)$, drawn as before, and the distribution $\hat{P}_{B\max}(r_{B\max})$ (dash-dot line) of the maximum reward of all the P_B samples. This maximum reward will cause a misclassification error if it is higher than the reward of the P_A sample. Note that $\hat{P}_{B\max}(r_{B\max})$ overlaps more with $\hat{P}_A(r)$ than $\hat{P}_B(r)$ does. Right panel, the probability of misclassification can be expressed as the overlap between $\hat{C}_{B\max}(r)$ (dashed line), the anti-cumulative of $\hat{P}_{B\max}(r)$, and $\hat{P}_A(r)$ (solid line).

To determine the misclassification rate we define two random variables:

$$r_A = r(\mathbf{z}_0), \qquad r_{B\max} = \max_{j=1,\ldots,M} r(\mathbf{z}_j), \tag{7}$$

where we have assumed, without loss of generality, that the 0th sample is from A and the remaining samples $\mathbf{z}_1, \ldots, \mathbf{z}_M$ are from $P_B(\cdot)$. Misclassification will occur whenever $r_{B\max}$ is larger than r_A. In this sense, $r_{B\max}$ is the reward of the most misleading sample from $P_B(\cdot)$, see Fig. 4.

We induce a probability distribution $\hat{P}_A(r_A)$ on r_A as before by requiring that $\mathbf{z}$ is generated by $P_A(\mathbf{z})$. A distribution $\hat{P}_{B\max}(r_{B\max})$ is also induced on $r_{B\max}$ by requiring that $\mathbf{z}_1, \ldots, \mathbf{z}_M$ are generated by $P_B(\mathbf{z}_1, \ldots, \mathbf{z}_M)$. This is calculated directly (in the next two sections we will discuss how to compute it for the problems of interest) from the formula:

$$\hat{P}_{B\max}(r_{B\max}) = \int d\mathbf{z}_1 \cdots d\mathbf{z}_M \; P_B(\mathbf{z}_1 \cdots \mathbf{z}_M) \times \delta(r_{B\max} - \max(r(\mathbf{z}_1), \ldots, r(\mathbf{z}_M))). \tag{8}$$

Let $\hat{C}_{B\max}(r_{B\max})$ be the "anti"-cumulative distribution of $\hat{P}_{B\max}(r_{B\max})$, i.e. $\hat{C}_{B\max}(r_{B\max}) = \int_{r_{B\max}}^{\infty} \hat{P}_{B\max}(r)\, dr$. (The term "anti"-cumulative is chosen since the limits of integration are non-standard.) Then *the probability of misclassification* is given by:

$$R^* = Pr\,(r_A < r_{B\max}) = \int dr \; \hat{P}_A(r)\, Pr\,(r < r_{B\max}) = \int dr \; \hat{P}_A(r) \hat{C}_{B\max}(r), \tag{9}$$

and so the danger of misclassification depends on the overlap between $\hat{P}_A(r)$ and $\hat{C}_{B\max}(r)$.

For the problems we are interested in, see the next two sections, there is an additional parameter N which determines the size of the target (N is a positive integer). We normalize the reward function by N and consider the distributions $\hat{P}_A(r/N)$ and $\hat{C}_{B\max}(r/N)$. The structure of the problem (e.g. the ergodicity of the distributions) means that $\hat{P}_A(r/N)$ will be sharply peaked and $\hat{C}_{B\max}(r/N)$ will tend to a step function. The Bayes risk will therefore tend to be zero or one depending on whether the step of $\hat{C}_{B\max}(r/N)$ is to the right or left of the peak of $\hat{P}_A(r/N)$, see Fig. 5.

Clearly the results depend on the peakedness of the distributions. For the models that we study this can be determined using results from the theory of large deviations (Demba and Zeitouni, 1998). This will be described in the following sections.

A very important issue, in the context of this paper, is how the analysis is modified if the *inference* is performed using incorrect probability models. We define new variables

$$s(\mathbf{z}) = \log \frac{Q_A(\mathbf{z})}{Q_B(\mathbf{z})},$$
$$s_{B\max}(\{\mathbf{z}_j \;:\; j = 1, \ldots, M\}) = \max_{j=1,\ldots,M} \log \frac{Q_A(\mathbf{z}_j)}{Q_B(\mathbf{z}_j)}, \tag{10}$$

where Q_A, Q_B are "wrong models" used for inference instead of the correct models P_A,P_B (i.e. the $\mathbf{z}_j$ are generated by P_A,P_B as before). Then we induce distributions $\hat{Q}_A(s) = \sum_{\mathbf{z}} P_A(\mathbf{z})\delta(s - \log Q_A(\mathbf{z})/Q_B(\mathbf{z}))$

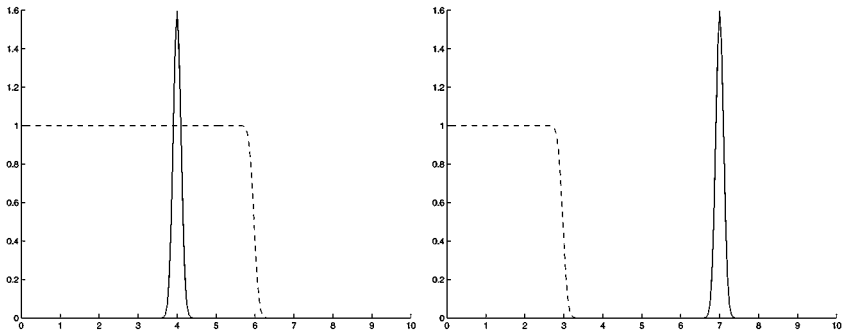

Figure 5. The Phase Transition. In each panel the solid line denotes $\hat{P}_A(r/N)$ (sharply peaked about $\bar{r}_A$) and the dashed lines denotes $\hat{C}_{B\max}(r/N)$. Left panel shows a large chance of misclassification because $\hat{C}_{B\max}(r/N)$ takes a large value near $\bar{r}_A$. The right panel shows a very small chance of misclassification because $\hat{C}_{B\max}(r/N)$ takes small values near $\bar{r}_A$.

and similarly compute the new "anti"-cumulative distribution $\hat{D}_{B\max}(s_{B\max})$ on the best distractor reward. The expected loss is then:

$$R_Q^* = \int ds\ \hat{Q}_A(s)\hat{D}_{B\max}(s). \tag{11}$$

Once again, for the problems in this paper, $\hat{Q}_A(s/N)$ and $\hat{D}_{B\max}(s/N)$ will tend to a delta function and a step function respectively for large N. The expected risk will be zero, or one, depending on whether the step is to the left or the right of the delta function spike.

This corresponds to analysing the problem using the wrong models. (Recall that the wrong model may be used because of either computational ease of inference or because the true model is not accurately known). As we will show in the next sections, there will be situations where the task can be solved (i.e. the loss is asymptotically zero) even when the wrong models are used. In other situations, the task can only be solved using the correct models.

3. Road Tracking

In this section we use concepts from Decision theory to analyse variants of the Geman and Jedynak model (Geman and Jedynak, 1996). This model was successfully applied to detecting roads from aerial images of the south of France.

We restrict ourselves to the question of whether the task can be solved (i.e. is the expected loss sufficiently small?). Our analysis will also show how the difficulty of the problem increases when we use approximate models for inference. Our analysis is done in the large N limit where the law of large numbers (or the "self-averaging" in physicists' terminology), makes estimating the expected loss straightforward (see Yuille and Coughlan (2000), for bounds on how fast the error rates change as a function of the length N of the road). We will only deal with the difficulty of distinguishing between the true road path and a set of distractor paths which have no overlap with the road. (Some analysis of the case when the distractor paths overlap with the road is presented in Yuille and Coughlan (2000)).

This paper is not concerned with specific algorithms for solving the problem. In their application domain, Geman and Jedynak (1996) demonstrated experimentally that their algorithm converged close to the optimal solution in linear expected time (i.e. $O(N)$). In related work (Coughlan and Yuille, 1999), we described an A* algorithm. We proved that, provided the task is solvable, the A* algorithm converges to a close approximation to the MAP estimate with expected complexity $O(N)$. (Properties of this algorithm, such as the size of the approximation error and the constants in the complexity results, were given in terms of quantities similar to the order parameters which we will derive in this section).

3.1. The Geman and Jedynak Model

Geman and Jedynak formulate road detection as tree search, see Fig. 6, through a Q-nary tree. The starting point and initial direction is specified and there are Q^N possible distinct paths down the tree. A road hypothesis consists of a set of connected straight-line segments called *segments*. We can represent a path by a sequence of moves $\{t_i\}$ on the tree. Each move t_i

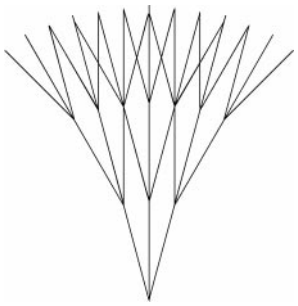

Figure 6. Geman and Jedynak's tree structure with a branching factor of $Q = 3$. The prior probabilities may express a reference for certain paths, such as those which are straight.

belongs to an *alphabet* $\{a_\mu\}$ of size Q. For example, the simplest case studied by Geman and Jedynak sets $Q = 3$ with an alphabet a_1, a_2, a_3 corresponding to the decisions: (i) a_1—go straight (0 degrees), (ii) a_2—go left (-5 degrees), or (iii) a_3—go right ($+5$ degrees). This determines a path $\mathbf{x}_1, \ldots, \mathbf{x}_N$ in the image lattice where $\mathbf{x}_i, \mathbf{x}_{i+1}$ indicate the start and end points of the ith segment. The relationship between the two representations is given by $\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{w}(\mathbf{x}_i - \mathbf{x}_{i-1}, t_i)$, where $\mathbf{w}(\mathbf{x}_i - \mathbf{x}_{i-1}, t_i)$ is a vector of approximately fixed magnitude (7 pixels plus small corrections to ensure that the segment ends on a pixel) and whose direction depends on the angle of the move t_i relative to the direction of the previous segment $\mathbf{x}_i - \mathbf{x}_{i-1}$.

There are some difficulties in mapping this tree representation onto an image lattice. These will be described in subsection 3.3 where we describe our computer simulations.

Geman and Jedynak place a prior probability on the set of paths down the tree. This can be expressed by a probability distribution $P(\{t_i\}) = \prod_{i=1}^{N} P(t_i)$. For our $Q = 3$ example, we may choose to go straight, left or right with equal probability (i.e. $P(a_1) = P(a_2) = P(a_3) = 1/3$). In a later section, we will consider first order Markov chain models where $P(\{t_i\}) = P(t_1) \prod_{i=1}^{N-1} P(t_{i+1}|t_i)$.

Geman and Jedynak derive their likelihood function by applying an oriented non-linear filter which is designed to detect straight road segments (by estimating a quantity related to the image gradient). The filter is quantized so that its response y can take one of J values $\{b_\mu\}$. The filter is trained on examples of on-road and off-road segments. For example, for all road segments $(\mathbf{x}_i, \mathbf{x}_{i+1})$ (for any i) we align the filter to the segment and compute its response y_i as a function of the image intensities on the segment (for precise details of the filter see Geman and Jedynak, 1996). This gives an empirical probability distribution $P_{\text{on}}(y_i = b_\mu)$. Similarly, they compute the empirical probability distribution $P_{\text{off}}(y_i = b_\mu)$ for the filter response evaluated off the road segments (i.e. the background). (See (7) for examples of loglikelihoods and see Konishi et al., 1999, for a detailed survey.) For any path $\{t_i\}$ through the tree we have a corresponding set of observations $\{y_i\}$. If the ith segment does lie on the true road then y_i is distributed by $P_{\text{on}}(\cdot)$ (otherwise by $P_{\text{off}}(\cdot)$). The filter responses are assumed to be independent for different segments, see Fig. 7.

As described in Geman and Jedynak (1996), MAP estimation corresponds to finding the path $\{t_i\}$ with filter measurements $\{y_i\}$ which maximizes the (scaled) loglikelihood ratio:

$$r(\{t_i\}, \{y_i\}) = \frac{1}{N}\left\{ \log P(Y \mid X) + \log P(X) - \sum_{i=1}^{N} \log U(t_i) \right\} \tag{12}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \log\{P_{\text{on}}(y_i)/P_{\text{off}}(y_i)\} + \frac{1}{N} \sum_{i=1}^{N} \log\{P_{\Delta G}(t_i)/U(t_i)\}, \tag{13}$$

where $U(\cdot)$ is the uniform distribution (i.e. $U(t) = 1/Q \ \forall t$) and so $\sum_{i=1}^{N} \log U(t_i) = -N \log Q$ which is a constant. The introduction of $U(\cdot)$ helps simplify the analysis in the following subsections.

3.2. *Analysis of the Geman and Jedynak Model Using Sanov's Theorem*

In this subsection, we analyze the performance of the Geman and Jedynak model from the perspective of decision theory. This analysis is a simplification of the more extended results (e.g. including partially overlapping paths and bounds for finite N) which are reported elsewhere Yuille and Coughlan (2000). Here we concentrate only on the qualitative aspects of performance (i.e. can we detect the road or not).

We assume that we have one sample $\{t_i\}$, $\{y_i\}$ of measurements generated by the road model (i.e. $P_{\text{on}}(\cdot)$, $P_{\Delta G}$) and we have to distinguish it from a background of distractor samples generated by $P_{\text{off}}(\cdot), U(\cdot)$. In addition, we assume that the distractor paths are based on a

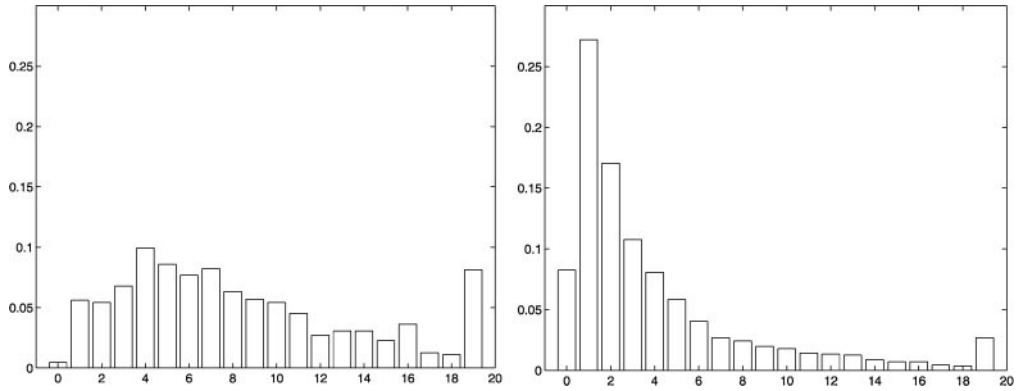

Figure 7. The quantized distributions P_{on} (Left) and P_{off} (Right) for the $|\vec{\nabla} I(\mathbf{x})|$ learnt from image data. Observe that, not surprisingly, $|\vec{\nabla} I(\mathbf{x})|$ is likely to take larger values *on* an edge rather than *off* an edge.

Q-nary tree so that these paths can overlap and are not independent. This is an approximation to the Geman and Jedynak model where one assumes that one path on the tree is the road and the other $Q^N - 1$ paths are distractors (see analysis in Yuille and Coughlan (2000). See subsection 3.3 for a discussion of the approximations needed to map the tree structure onto the image lattice.

To analyze the road detection task from our decision theory perspective, see Eq. (9), requires us to compute $\hat{P}_A(r)$ and $\hat{C}_{B\max}(r)$ where A indicates the road path and B the distractor paths.

For the road detection problem, the variable $\mathbf{z} = (\mathbf{t}, \mathbf{y})$ where $\mathbf{t} = (t_1, \ldots, t_N)$ describes the spatial geometry of the path and $\mathbf{y} = (y_1, \ldots, y_N)$ are the measurements of the edge detection filters along the path. The probability distribution $P_A(\mathbf{z})$ and $P_B(\mathbf{z})$ are replaced by:

$$P_A(\mathbf{z}) = \prod_{i=1}^{N} P_{\text{on}}(y_i) P_{\Delta G}(t_i),$$
$$P_B(\mathbf{z}) = \prod_{i=1}^{N} P_{\text{off}}(y_i) U(t_i). \tag{14}$$

The (scaled) log-likelihood ratio $r = (1/N) \log P_A(\mathbf{z})/P_B(\mathbf{z})$ is identical to the criterion, Eq. (13), that Geman and Jedynak seek to maximize. Observe we have *scaled* the log-likelihood ratio by $1/N$ so that it will tend to a finite limit as $N \mapsto \infty$.

We now introduce an alternative representation for the problem which is crucial for our analysis. Recall that the moves t_i and the observations y_i take values within the *finite* alphabets $\{a_\mu\}$ and $\{b_\nu\}$. For any path $\{t_i\}$, $\{y_i\}$ we can define *two histograms* $\vec{\psi}$, $\vec{\phi}$ with components $\psi_\mu = \frac{1}{N} \sum_{i=1}^{N} \delta_{t_i, a_\mu}$ $(\mu = 1, \ldots, Q)$ and $\phi_\nu = \frac{1}{N} \sum_{i=1}^{N} \delta_{y_i, b_\nu}$ $(\nu = 1, \ldots, J)$. These histograms are *sufficient statistics* (De Groot, 1970) for the distributions $P_A(\mathbf{z})$, $P_B(\mathbf{z})$ (i.e. the distributions $P_A(\mathbf{z})$ and $P_B(\mathbf{z})$ can be expressed as $f_A(\vec{\psi}(\mathbf{z}), \vec{\phi}(\mathbf{z}))$ and $f_B(\vec{\psi}(\mathbf{z}), \vec{\phi}(\mathbf{z}))$ for functions $f_A(\cdot)$ and $f_B(\cdot)$). In particular, the (scaled) log-likelihood ratio can be expressed as:

$$r(\{t_i\}, \{y_i\}) = \vec{\alpha} \cdot \vec{\psi} + \vec{\beta} \cdot \vec{\phi}, \tag{15}$$

where we define the two vectors $\vec{\alpha}$ and $\vec{\beta}$ to have components $\alpha_\mu = \log \frac{P_{\Delta G}(a_\mu)}{U(a_\mu)}$ for $\mu = 1, \ldots, Q$ and $\beta_\nu = \log \frac{P_{\text{on}}(b_\nu)}{P_{\text{off}}(b_\nu)}$ for $\nu = 1, \ldots, J$.

We first determine the behaviour of $\hat{P}_A(r)$ for large N. The result is that, for large N, $\hat{P}_A(r)$ becomes *sharply peaked about its mean value* $\bar{r}_A = \langle r \rangle_{\hat{P}_A} = (1/N) D(\hat{P}_A \| \hat{P}_B)$ where $D(\hat{P}_A \| \hat{P}_B) = \sum_r \hat{P}_A(r) \log \frac{\hat{P}_A(r)}{\hat{P}_B(r)}$ is the *Kullback-Leibler* divergence between $\hat{P}_A(r)$ and $\hat{P}_B(r)$. This result follows from the law of large numbers which implies that the normalized sum of a set of N independent identically distributed (i.i.d.) variables tends to the *mean* of the distribution as $N \mapsto \infty$. Moreover, the distribution $\hat{P}_A(r)$ falls off from its peak value, at $r = \bar{r}_A$, exponentially with N. The proof of this second result follows from Sanov's theorem, see Appendix A, which is a result in the large deviation theory literature. Large deviation theory (Dembo and Zeitouni, 1998) is an area of statistics which attempts to put bounds on the probabilities of rare events. This result is important because it means that we *only need to determine the value of* $\hat{C}_{B\max}(r_A)$

(*because* $\hat{P}_A(r)$ *is peaked about* $\bar{r}_A$ *for large* N). For organizational purposes we state this result as a theorem.

First we introduce the notation $\doteq$ used in Cover and Thomas (1991) to simplify the results derived from Sanov's theorem and to concentrate on the important aspects. Let x be a variable that takes J distinct values. Then we say that $f(x; N) \doteq e^{Ng(x)}$ to mean that there exist polynomial functions of N, $poly_1(N)$ and $poly_2(N)$, whose order depends only on J, such that $\frac{1}{poly_1(N)} e^{Ng(x)} \leq f(x; n) \leq poly_2(N)\, e^{Ng(x)}$ for all x, N.

Theorem 1. *The mean reward of the road is given by*

$$\begin{aligned}\bar{r}_A &= (1/N) D(\hat{P}_A(r) \parallel \hat{P}_B(r)) \\ &= D(P_{\text{on}} \parallel P_{\text{off}}) + D(P_{\Delta G} \parallel U).\end{aligned}$$

Moreover,

$$\int_{r:|r-\bar{r}_A|\geq\epsilon} \hat{P}_A(r)\, dr \doteq e^{-N\{D(\phi_\epsilon \parallel P_{\text{on}}) + D(\psi_\epsilon \parallel P_{\Delta G})\}},$$

where ϕ_ϵ, ψ_ϵ *are chosen so as to minimize* $D(\phi_\epsilon \| P_{\text{on}}) + D(\psi_\epsilon \parallel P_{\Delta G})$ *subject to the constraint that* $|r - r_A| \geq \epsilon$.

Proof: $\bar{r}_A$ can be computed directly. The remaining results follow from Sanov's theorem (Cover and Thomas, 1991) which we state in Appendix A. The details of this derivation are available as a technical report (Yuille et al., 2000). □

We now consider what is the probability distribution for the best reward $r_{B\max}$ for the distractor paths. This is done in two stages. The first stage computes $E[Z(\gamma, N)]$, the *expected number of distractor paths of length N which have rewards greater than* γ. This also makes use of Sanov's theorem. The result, see Theorem 2, shows that there is a critical value γ^*. The precise value of γ^* is given by a set of simultaneous non-linear equations, see Appendix A.

Theorem 2. *Let* $Z(\gamma, N)$ *be the number of distractor paths of length N with rewards greater than* γ *and let* $E[\cdot]$ *be the expectation with respect to* $P_B(\mathbf{z})$. *Then there exists a critical value* γ^* *such that* $\lim_{N\mapsto\infty} E[Z(\gamma, N)] \mapsto 0$ *for* $\gamma > \gamma^*$ *and* $\lim_{N\mapsto\infty} E[Z(\gamma, N)] \mapsto \infty$ *for* $\gamma \leq \gamma^*$.

Proof: The probability that any one distractor path has reward greater than γ can be tightly bounded using Sanov's theorem and shown to be of form $\doteq e^{-Ng(\gamma)}$ for a positive monotonically increasing function $g(\cdot)$, see Appendix A. Multiply by Q^N to obtain the expected number $\doteq e^{-N\{g(\gamma)-\log Q\}}$ of distractor paths with rewards greater than γ. The critical value γ^* is the solution of the equation $g(\gamma) = \log Q$. See the technical report for more details. □

A further theorem is required to prove that the maximum reward of all distractor paths is γ^* for large N. We emphasize that a key part of this proof requires that the distractor paths form a tree structure.

Theorem 3. *If the distractor paths are defined on a Q-nary tree then* $\lim_{N\mapsto\infty} \hat{C}_{B\max}(r) = 0$ *for* $r > \gamma^*$ *and* $\lim_{N\mapsto\infty} \hat{C}_{B\max}(r) = 1$ *for* $r < \gamma^*$.

Proof: $E[Z(r_{B\max}, N)] \geq \hat{P}_{B\max}(r_{B\max})$ and so $\lim_{N\mapsto\infty} \hat{P}_{B\max}(r_{B\max}) = 0$ for $r_{B\max} > \gamma^*$. To complete the proof requires showing that, with high probability, there exist distractor paths with rewards r arbitarily close to γ^*. This result follows from generalizing a theorem by Karp and Pearl (1984) and exploits the fact that the distractor paths form a tree. See the technical report (Yuille et al., 2000) for more details. □

To complete our analysis of the expected loss, see Eq. (9), (as $N \mapsto \infty$) we must determine whether γ^* is greater than, or less than, $\bar{r}_A$. Our final theorem of this section gives a simple condition to determine this.

Theorem 4. *Let* $K = D(P_{\text{on}} \| P_{\text{off}}) + D(P_{\Delta G} \| U) - \log Q$. *Then* $\lim_{N\mapsto\infty} R^* = 0$ *if, and only if,* $K > 0$.

Proof: This result follows straightforwardly by analyzing the relative size of $\bar{r}_A = D(P_{\text{on}} \| P_{\text{off}}) + D(P_{\Delta G} \| U)$ and γ^*. See the technical report (Yuille et al., 2000) for details. As a first step, it follows directly from Sanov's theorem that the expected number of distractor paths with rewards greater than $D(P_{\text{on}} \| P_{\text{off}}) + D(P_{\Delta G} \| U)$ (the expected reward of the road) is of form $\doteq e^{-NK}$. □

The bottom line is that whether the road is detectable or not depends only on the size of the *order parameter* K. (We use the words "order parameter" by analogy to parameters in statistical physics.) The order parameter increases the more reliable the local cues for detecting the road are (as measured by $D(P_{\text{on}}(\cdot) \| P_{\text{off}}(\cdot))$) and the more specific the prior knowledge about the road shape is (as measured by $D(P_{\Delta G}(\cdot) \parallel U(\cdot))$). The order parameter decreases as the number of distractors, as measured by Q^N, increases. For $K < 0$ it will be

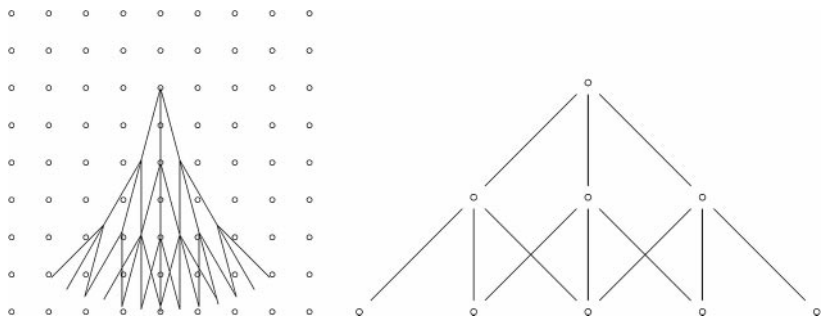

Figure 8. Left panel: the tree structure superimposed on the lattice. Right panel: the pyramid structure used in the simulations.

impossible, on average, to detect the road (because the probability becomes high that at least one distractor path has higher reward than the road). For $K > 0$ it will be possible to detect the road. (Other aspects of the problem, such as algorithmic complexity (Coughlan and Yuille, 1999) and error rates for partially overlapping paths, Yuille and Coughlan, 2000, will depend on the precise value of K.)

3.3. Computer Simulations: From Tree to Pyramid

The tree representation used by Geman and Jedynak must be modified when we map onto an image lattice, see Fig. 8. The easiest way to do this involves defining a *pyramid* where paths start at the apex and the only allowable "moves" are: (i) one step down, (ii) one step down and one step left, and (iii) one step down and one step right. This can be represented by $\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{w}(t_i)$ where $t_i \in \{-1, 0, 1\}$ and $\mathbf{w}(-1) = -\vec{i} - \vec{j}$, $\mathbf{w}(0) = -\vec{j}$, $\mathbf{w}(1) = +\vec{i} - \vec{j}$ (where $\vec{i}$, $\vec{j}$ are the x,y directions on the lattice).

To obtain computer simulations of roads in background clutter we proceed in two stages. In the first stage, we stochastically sample from the distribution $P_{\Delta G}(t)$ to produce a road path in the pyramid (starting at the apex and moving downwards). In the second stage, we must sample from the likelihood function to generate the image. We make this simple by choosing our filter responses y to be the intensity variables. So if a pixel $\mathbf{x}$ is *on* or *off* the path (which we generated in the first stage) then we sample the intensity $I(\mathbf{x})$ from the distribution $P_{\text{on}}(I)$ or $P_{\text{off}}(I)$ respectively. Dynamic programming is used in each sample image to obtain the path with best reward which is the MAP estimate of the target path.

There is one critical differences between the lattice and the tree representations: the distractor paths on the lattice can *separate and then rejoin each other*. Although the are 3^N possible paths in the pyramid (starting at the apex) there are only $(N+1)^2$ total samples from the likelihood function (as against 3^N samples for the tree model). This does not affect the proofs of Theorems 1 and 2—so the expected reward of the road is as stated and the *expected number of distractor paths with rewards greater than* γ has a phase transition at $\gamma = \gamma^*$. But the proof of Theorem 3 depends on the tree structure so we *can no longer be sure that the maximum reward of all distractor paths* tend to γ^*. By the first line of the proof, however, we do know that the maximum reward of the distractor paths cannot exceed γ^* but it may be lower.

We use computer simulations to estimate the maximum reward for distractor paths on the pyramid for the special case where the prior geometry is given by the uniform distribution. Our computer simulations, see Table 1, show that the maximum reward of the distractor paths is typically slightly smaller than γ^* for a range of different choices of $P_{\text{on}}(\cdot)$, $P_{\text{off}}(\cdot)$. This implies that the order parameter obtained by the calculation on the trees do need to be increased slightly for the lattice. We observe two trends in our simulations, see Table 1. Firstly, the shorter the length of the path then the larger the difference between γ^* and the empirical mean maximum reward. Secondly, the more similar the distributions, $P(\cdot|\text{on})$ and $P(\cdot|\text{off})$, then the smaller the difference.

It should be stressed that the calculation for the *expected* number of distractor paths with rewards higher than the mean true path reward is exact for both the lattice and the tree representations. So if the task is formulated in this way then performance in both cases is measured by the same order parameter. It seems more reasonable, however, to evaluate the task difficulty in terms of the Bayes risk. In which case the order parameters for the lattice are slightly bigger than those for the tree.

Table 1. Comparison of the Maximum reward of all distractor paths with γ^* for the pyramid case.

$P(\cdot \mid \text{on})$	$P(\cdot \mid \text{off})$	γ^*	N	Emp. mean max. reward	Standard deviation
(0.4, 0.6)	(0.6, 0.4)	0.3638	20	0.307	0.044
(0.4, 0.6)	(0.6, 0.4)	0.3638	100	0.35	0.01
(0.4, 0.6)	(0.6, 0.4)	0.3638	200	0.353	0.01
(0.4, 0.6)	(0.6, 0.4)	0.3638	400	0.362	0.0032
(0.3, 0.7)	(0.7, 0.3)	0.6182	20	0.46	0.1
(0.3, 0.7)	(0.7, 0.3)	0.6182	100	0.55	0.04
(0.3, 0.7)	(0.7, 0.3)	0.6182	200	0.57	0.02
(0.1, 0.9)	(0.9, 0.1)	0.34	20	−0.31	0.3
(0.1, 0.9)	(0.9, 0.1)	0.34	100	−0.1	0.1
(0.1, 0.9)	(0.9, 0.1)	0.34	400	−0.02	0.05

The first two columns give the Bernouilli distributions $P(\cdot \mid \text{on})$, $P(\cdot \mid \text{off})$ respectively. The third column gives the theoretical calculation of γ^* which is the value of the reward at which there is a phase transition in the expected number of distractor paths. The fourth column gives the length N of the path. Columns five and six give the empirical mean maximum reward of the distractor paths (we ran several simulations and computed the mean of the maximum reward distractor path) and the standard deviation (with respect to our simulation runs). Observe that the empirical mean maximum rewards approach γ^* quickly for the first two cases, as a function of the length of the path, but convergence is much slower for the third case where the distributions $P(\cdot \mid \text{on})$ and $P(\cdot \mid \text{off})$ are very different.

This small shift in the order parameter values makes little change in the ability to detect the true road path. In our experiments, see Fig. 9, the order parameter K computed on the tree accounts well for whether the true road can be detected. The only exceptions occur when K is negative but with small modulus. In this case, the shift in the order parameters (from tree to lattice) is needed.

3.4. High-Low for Geman and Jedynak

The analysis in the previous two subsections assumed that we used the correct reward function to perform inference. In this subsection, an early version of which appeared in a conference proceedings (Yuille and Coughlan, 1999), we analyze the value of information *lost* by using a weaker prior model. (A similar analysis can be used to investigate the effects of using approximate models for the likelihood terms $P_{\text{on}}(\cdot)$, $P_{\text{off}}(\cdot)$.)

More precisely, in place of the correct *high-level* geometric model $P_{\Delta G,H}(t)$ we replace it by a weaker *generic* model $P_{\Delta G,G}(t)$. This defines two different rewards R_G and R_H:

$$R_G(\{t_i\}) = \sum_i \log \frac{P_{\text{on}}(y_i)}{P_{\text{off}}(y_i)} + \sum_i \log \frac{P_{\Delta G,G}(t_i)}{U(t_i)},$$
$$R_H(\{t_i\}) = \sum_i \log \frac{P_{\text{on}}(y_i)}{P_{\text{off}}(y_i)} + \sum_i \log \frac{P_{\Delta G,H}(t_i)}{U(t_i)}. \tag{16}$$

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 this strategy ignores the extra computation time which may be involved in using the prior $P_{\Delta G,H}$. For example, $P_{\Delta G,H}$ might be a first or higher order Markov model (see next section) while $P_{\Delta G,G}$ might be a zeroth order Markov model which would be easier to search over. (But applying Sanov's theorem to a first-order model does require further technical conditions to hold, see Yuille et al., 2000). Also, we might not know the exact form of $P_{\Delta G,H}$. Perhaps the most important situation, to be considered in a later section, is when we can use a single generic model to search for a target which may be one of several different models. Using a single generic model (provided it is powerful enough) to detect the road can be significantly faster than testing each possible road model in turn.

In this paper we will be concerned with the *Amari* condition, which was motivated by results in Amari's theory of information geometry (Amari, 1982). This condition relates the high-level geometric distributions,

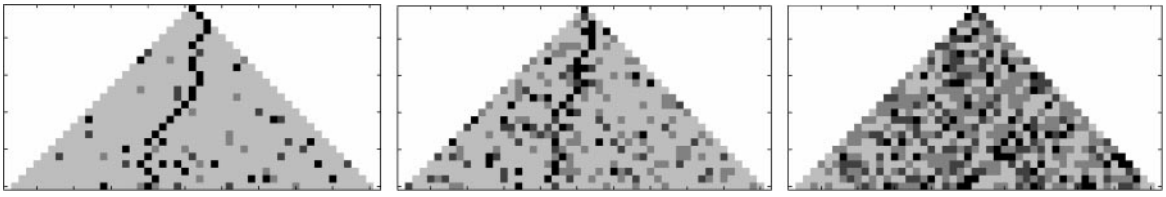

Figure 9. The difficulty of detecting the target path in clutter depends, by our theory, on the order parameter K. The larger K the less computation required. Left, an easy detection task with $K = 0.8647$. Middle, a harder detection task with $K = 0.2105$. Right, an impossible task with $K = -0.7272$.

$P_{\Delta G,H}(t)$, to the generic distributions $P_{\Delta G,G}(t)$ by:

$$\sum_t P_{\Delta G,H}(t)\ \log P_{\Delta G,G}(t) = \sum_t P_{\Delta G,G}(t)\ \log P_{\Delta G,G}(t). \quad (17)$$

This condition is special in that it allows us to obtain analytic expressions for the order parameters *and* an important connection to the Minimax Entropy Learning scheme (Zhu et al., 1997) (as we will describe in the next section). But it should be emphasized that *order parameters can be derived for other conditions* but they may not have the simple analytic expressions which arise from the Amari condition.

The analysis of the inference using $P_{\Delta G,H}$ was done in the previous two subsections. The critical concern was whether the expected high-level reward for the best path $D(P_{\text{on}} \parallel P_{\text{off}}) + D(P_{\Delta G,H} \parallel U)$ was greater than, or equal to, $\log Q$.

To deal with the generic model, we find that the expected reward for the true path using the generic model is:

$$\sum_y P_{\text{on}}(y)\ \log \frac{P_{\text{on}}(y)}{P_{\text{off}}(y)} + \sum_t P_{\Delta G,H}(t)\ \log \frac{P_{\Delta G,G(t)}}{U(t)} = D(P_{\text{on}} \parallel P_{\text{off}}) + D(P_{\Delta G,G} \parallel U), \quad (18)$$

where we have used the Amari condition to obtain the second term on the right hand side. Thus the effect of changing the model is merely to shift the spike of the distribution of the true path from $D(P_{\text{on}} \parallel P_{\text{off}}) + D(P_{\Delta G,H} \parallel U)$ down to $D(P_{\text{on}} \parallel P_{\text{off}}) + D(P_{\Delta G,G} \parallel U)$.

The analysis of the best distractor path and its comparison to the expected reward of the road proceeds as before, see the technical report (Yuille et al., 2000) for details, to yield an order parameter K_G for the generic geometry model which can be contrasted with the order parameter K_H when the high-level model is used. This gives:

$$K_H = D(P_{\text{on}} \parallel P_{\text{off}}) + D(P_{\Delta G,H} \parallel U) - \log Q,$$
$$K_G = D(P_{\text{on}} \parallel P_{\text{off}}) + D(P_{\Delta G,G} \parallel U) - \log Q. \quad (19)$$

It follows from the definition of the Amari condition that $K_H - K_G = D(P_{\Delta G,H} \parallel U) - D(P_{\Delta G,G} \parallel U) = D(P_{\Delta G,H} \parallel P_{\Delta G,G})$ (where $D(p \parallel q) = \sum_y p(y) \log p(y)/q(y)$ is the *Kullback-Leibler* divergence between distributions $p(y)$ and $q(y)$). Therefore the high-level prior $P_{\Delta G,H}$ has an order parameter larger by an amount which depends on the distance between it and $P_{\Delta G,G}$ as measured by the Kullback-Leibler divergence $D(P_{\Delta G,H} \parallel P_{\Delta G,G})$. Recall Yuille and Coughlan (1999) that the target detection problem becomes insolvable (by any algorithm) when the order parameter is less than zero. Hence there are three regimes: (I) The *Ultra Regime*, see Fig. 10, is when $K_G < K_H < 0$ (i.e. $D(P_{\Delta G,H} \parallel U) + D(P_{\text{on}} \parallel P_{\text{off}}) < \log Q$) and the problem cannot be solved (on average) by any model (or algorithm). (II) The *Challenging Regime*, see Fig. 11, where $K_G < 0 < K_H$ (i.e. $\log Q < D(P_{\Delta G,H}$

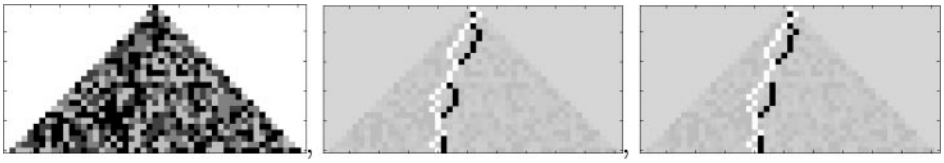

Figure 10. The Ultra Regime $K_H < K_G < 0$. Left, the input image. Centre, the true path is shown in white and the *errors* of the best path found using the Generic model are shown in black. Right, similar, for the High-Level model. Observe that although the best paths found are close to the true path there is comparatively little overlap. A dynamic programming algorithm was used to determine the best solution for either choice of reward.

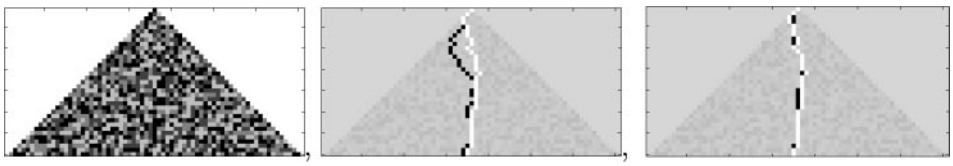

Figure 11. The Challenging Regime $K_G < 0 < K_H$. Same conventions as previous figure. Observe that the Generic models fails (centre) but the High-Level model succeeds (right).

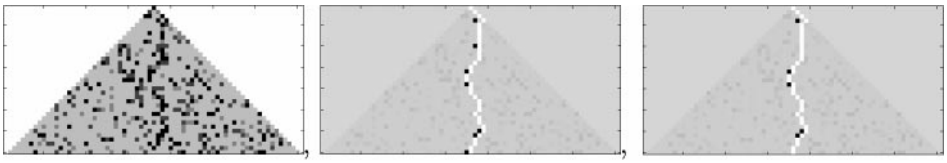

Figure 12. The Easy Regime $0 < K_G < K_H$. Same conventions as previous figure. In this regime both the Generic and High-Level models succeed.

$\|U) + D(P_{\text{on}} \| P_{\text{off}}) < \log Q + D(P_{\Delta G,H} \| P_{\Delta G,G})$) within which the problem can be solved by the high-level model but not by the generic model. (III) The *Easy Regime*, see Fig. 12, where $K_H > K_G > 0$ and the problem can be solved by either the generic or the high-level model.

We illustrate these results by computer simulations which use the same setup described in subsection 3.3. We show examples of the ultra, the challenging, and the easy regimes in Figs. 10–12. As before, 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.

3.5. *Multiple Hypotheses and Higher-Order Markov Models*

We extend the theory to deal with multiple (two or more) high-level models, see Fig. 13. In particular, we formulate the idea of a hierarchy in which the priors for several high-level objects can all be approximated by the same low-level prior, see Fig. 13. For example, we might have a set of priors $\{P_{H_i} : i = 1, \ldots, M\}$ for different members of the cat family. There might then be a generic prior P_G which approximate all these models $\{P_{H_i}\}$ and which is considered the embodiment of "cattiness." (In Section 4.4 we show that approximation can be nicely formulated in terms of projection in probability space).

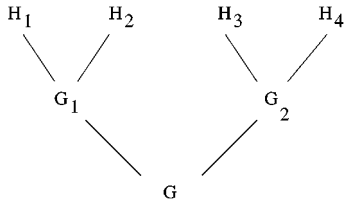

Figure 13. The Hierarchy. Two high-level models $P_{\Delta G,H_1}$, $P_{\Delta G,H_2}$ "project" onto a low-level generic model $P_{\Delta G,G_1}$. In situations with limited clutter it will be possible to detect either $P_{\Delta G,H_1}$ or $P_{\Delta G,H_2}$ using the single generic model $P_{\Delta G,G_1}$. 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.

In addition, we consider high-level models defined by second-order Markov chains. For second order Markov models the geometry is no longer i.i.d. but we can still apply Sanov's theorem for certain classes of model. See the technical report (Yuille et al., 2000) for the details.

The prototypical case for two, or more, high-level models is illustrated in Fig. 14. High-level model $P_{\Delta G,H_1}$ prefers roads which move to the right (see the white paths in the left hand panels of Fig. 14) while high-level model $P_{\Delta G,H_2}$ likes roads moving to the left (see white paths in the right panels). Both models $P_{\Delta G,H_1}$ and $P_{\Delta G,H_2}$ *project to the same generic model* $P_{\Delta G,G}$, by Amari projection, and thus form part of a hierarchy, see Fig. 13. Our theory again enables us to calculate order parameters and identify three regimes: (I) The Ultra Regime where none of the models ($P_{\Delta G,H_1}$,$P_{\Delta G,H_2}$ or $P_{\Delta G,G}$) can find the target. (II) The Challenging Regime where the high-level models $P_{\Delta G,H_1}$,$P_{\Delta G,H_2}$ can find targets generated by $P_{\Delta G,H_1}$ and $P_{\Delta G,H_2}$ *respectively* but the generic model $P_{\Delta G,G}$ cannot find either. (III) The Easy Regime where the high-level models find their appropriate targets and the generic models find both types of target. Once again, the best paths for the different rewards was found using dynamic programming (which is guaranteed to find the global solution).

In the Easy Regime, little is gained by using the two high-level models. It may indeed be more computationally efficient to use the generic model to detect the target. The target could then be classified as being $P_{\Delta G,H_1}$ or $P_{\Delta G,H_2}$ in a subsequent classification stage. We will discuss computational tradeoffs of these two approaches in the next section.

We now repeat this example using high-level models $P_{\Delta G,H_3}$,$P_{\Delta G,H_4}$ defined by second order Markov chains,

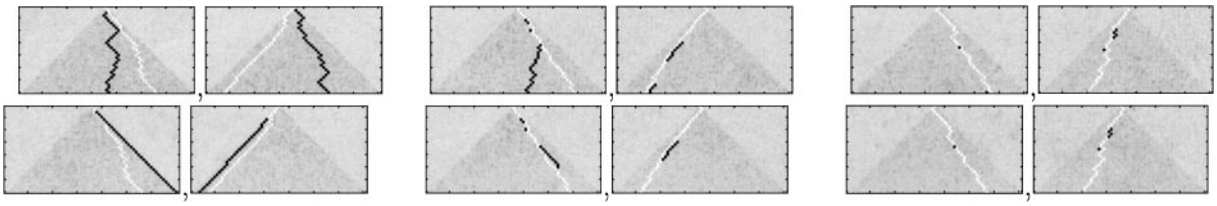

Figure 14. Two High-Level models $P_{\Delta G,H_1}$, $P_{\Delta G,H_2}$. Three sets of four panels for Ultra, Challenging, and Easy regimes (left to right). For each of the three sets, the data in the left and right columns is generated by $P_{\Delta G,H_1}$ and $P_{\Delta G,H_2}$ respectively. The lower rows gives the solutions found by the High-Level model ($P_{\Delta G,H_1}$ or $P_{\Delta G,H_2}$ as appropriate) and the upper rows give the solutions found by the Generic model with the true paths (white) and the errors of the best paths (black). Observe that all models give poor results in the Ultra regime (left panel). In the Challenging regime (centre panel) we get good results for the High-Level models and significantly poorer results for the Generic. The rightmost panel (same conventions) demonstrate the effectiveness of all models in the Easy regime.

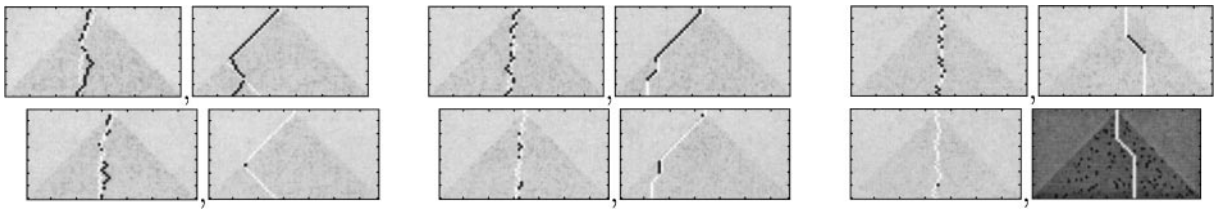

Figure 15. Two High-Level models second-order Markov models $P_{\Delta G,H_3}$,$P_{\Delta G,H_4}$. Three sets of four panels for Ultra, Challenging, and Easy regimes (left to right). For each of the three sets, the data in the left and right columns is generated by $P_{\Delta G,H_3}$ and $P_{\Delta G,H_4}$ respectively. The lower rows gives the solutions found by the High-Level model ($P_{\Delta G,H_3}$ or $P_{\Delta G,H_4}$ as appropriate) and the higher rows give the solutions found by the Generic model with the true paths (white) and the errors of the best paths (black). Observe that all models give poor results in the Ultra regime (left panel). In the Challenging regime (centre panel) we get good results for the High-Level models and significantly poorer results for the Generic. The rightmost panel (same conventions) demonstrate the effectiveness of all models in the Easy Regime.

see Fig. 15. This second order property allows us to obtain more interesting models. For example, model $P_{\Delta G,H_3}$ generates very wiggly roads ("English" roads) (see left panel of Fig. 15) while model $P_{\Delta G,H_4}$ generates roads that have long straight sections with occasional sharp changes in direction ("Roman" roads, see right hand panels). It is straightforward to compute order parameters for these models (the second-order Markov property requires slight modifications to the earlier calculations) and, as before, we get order parameters which specify the three standard Ultra, Challenging, and Easy regimes—see Fig. 15. In this figure, we point out a fluke where the high-level model $P_{\Delta G,H_4}$ correctly found the target even in the Ultra Regime. By our theory, this is possible though highly unlikely. Another unlikely outcome is shown in the bottom right panel where the $P_{\Delta G,H_4}$ model has detected the target to *one hundred percent accuracy*. This is reflected in the overall darkness of the panel because, with no black pixels to indicate errors, our graphics package has altered the brightness of the panel (compared to the other panels which do contain black errors). Dynamic programming is used to find the best solutions by global optimization.

4. Order Parameters for Non-Factorizable Models

So far, our results have assumed that the data is generated by factorizable models which enables us to use Sanov's theorem for our analysis. In this section we use more general techniques from large deviation theory to analyze more general distributions.

We are particularly interested in analyzing the behaviour of a more general class of probability distributions which includes those resulting from Minimax Entropy learning (Zhu et al., 1997; Zhu, 1999). This is a class of Gibbs distributions which are shift-invariant and obey certain scaling results (to be described later). Each distribution is of form:

$$P(\mathbf{z} \mid \vec{\beta}) = \frac{e^{-N\vec{\beta}\cdot\vec{h}(\mathbf{z})}}{Z(\vec{\beta})}, \tag{20}$$

where $\mathbf{z} = (z_1, \ldots, z_N)$ has N components, $\vec{\beta}$ is a parameter (independent of N), $\vec{h}(\cdot)$ are statistics defined on $\mathbf{z}$, and $Z(\vec{\beta})$ is the partition function (a normalization constant). We could, for example, let $\mathbf{z}$ be an intensity image of size N with the $\{z_i\}$ being the pixel intensities.

In this case, the statistics could be the (normalized) histograms of filter outputs over the entire image. Alternatively $\mathbf{z}$ might represent the geometry and image filter values on an image curve.

Our previous results can be obtained as a special case when the distribution $P(\mathbf{z})$ is factorizable. More specifically, let β_μ,h_μ be the components of the vectors $\vec{\beta}$, $\vec{h}$. Then let $h_\mu(\mathbf{z}) = (1/N)\sum_{i=1}^{N}\delta_{(\mu,z_i)}$ (i.e. the standard histogram). It is then straightforward to calculate $P(\mathbf{z})$ from Eq. (20) and show that it is factorizable and equals $\prod_{i=1}^{N} P(z_i)$ where $P(z_i = \mu) = e^{\beta_\mu}$.

The distribution $P(\mathbf{z})$ given by Eq. (20) determines an induced distribution on the *feature space* of all possible values of the statistics:

$$\hat{P}(\vec{h} \mid \vec{\beta}) = |\Omega_N(\vec{h})|\frac{e^{-N\vec{\beta}\cdot\vec{h}}}{Z(\vec{\beta})}, \tag{21}$$

where $\Omega_N(\vec{h}) = \{I : \vec{h}(I) = \vec{h}\}$ and $|\Omega_N(\vec{h})|$ is the size of this set. Let Q be the number of grayscale levels so that the total number of all possible images is Q^N. Then $|\Omega_N(\vec{h})|/Q^N$ can be considered to be a normalized probability distribution on $\vec{h}$ induced by the uniform distribution on all images (i.e. $\sum_{\vec{h}} |\Omega_N(\vec{h})|/Q^N = 1$).

As before, we want to analyze the chances of misclassification of data generated by models of this form and, in particular, for curve detection. To do this requires determining the probability of rare events such as when random alignments of background clutter appear to look like the target curve.

4.1. *Bounds for Log-Likelihood Discrimination Tasks*

In this second, we give results on detection for the new class of probability models. Our results are weaker than those obtained for the i.i.d case (see previous section) in two respects. Firstly, the results are *asymptotic* (i.e. they apply only in the limit as $N \mapsto \infty$) and not bounds for finite N. Secondly, because the analysis is based on a grid (rather than a search tree) we are unable to compute the probability distribution of the best bad path. We are, however, able to obtain results for the *expected* number of distractor paths with rewards greater than γ. This gives an upper bound for the reward of the best distractor path and our computer simulations suggest that this upper bound is exact.

To obtain our results, we make use of theorems from the large deviation theory literature. These are described in Appendix B. They can be thought of as extension of Sanov's theorem to non-factorizable distributions.

Our main result is Theorem 5 which deals with the expected number of distractor paths with rewards greater than the expected reward of the true. The following subsection gives three theorems which are generalizations to the non-iid case of results obtained by Yuille and Coughlan (2000) for the i.i.d. case. They are included here for completeness.

We state the result in this section without proof. The proofs are given in our technical report (Yuille et al., 2000) and build on the large deviation theory results of the previous section.

Theorem 5. *Suppose we have $e^{N\log Q}$ samples from distribution $P_B(\cdot)$ and one sample from $P_A(\cdot)$. Then the expected number that have reward* $\log P_A(\cdot)/P_B(\cdot)$ *higher than $\langle R\rangle_{P_A}$ is given by $e^{-N\{d(P_A \parallel P_B)-\log Q\}}$, where $d(P_A \parallel P_B) = \lim_{N\mapsto\infty}(1/N)D(P_A \parallel P_B)$. This defines an order parameter $K = d(P_A \parallel P_B) - \log Q$.*

This result is used to determine whether the true road, the sample from P_A, can be distinguished from the $e^{N\log Q}$ distractor paths sampled from P_B. There is clearly a phase transition at $Q = d(P_A \parallel P_B)$. If $d(P_A \parallel P_B) > \log Q$ then we expect there to be no distractor paths (in the large N limit) with rewards as high as those from the distractor paths. It should therefore be possible to detect the true road. On the other hand, if $d(P_A \parallel P_B) < \log Q$ then we expect it to be impossible to detect the true path.

This result is similar to that we obtained from studying the Geman and Jedynak model, see Section 3. It is slightly weaker because, like the result for Geman and Jedynak on the lattice (see subsection 3.3) it can only determine the *expected number* of distractor paths with rewards greater than the expected true reward. It does *not* determine whether the *best distractor path has a reward higher than the average true reward* (Recall that the proof of Theorem 3 requires a tree structure for the distractors).

We performed computer simulations to investigate the effect of having an unknown starting point and a more realistic (i.e. non-pyramidal) image lattice. The simulations were performed using an i.i.d. model. They showed, as for Geman and Jedynak on a lattice, that the difference between γ^* and the maximum distractor reward is usually small, see Table 2. As with the pyramid case we observe that the bigger the difference between the two distributions the bigger the difference between γ^* and the empirical results. We also observed that,

Table 2. Comparison of the maximum reward of all distractor paths with γ^* for the lattice case with *unknown* starting point.

$P(\cdot \mid \text{on})$	$P(\cdot \mid \text{off})$	γ^*	N	Emp. mean max. reward	Standard deviation
(0.4, 0.6)	(0.6, 0.4)	0.3638	20	0.399	0.0015
(0.4, 0.6)	(0.6, 0.4)	0.3638	100	0.384	0.0065
(0.4, 0.6)	(0.6, 0.4)	0.3638	400	0.3726	0.003
(0.3, 0.7)	(0.7, 0.3)	0.6182	20	0.74	0.05
(0.3, 0.7)	(0.7, 0.3)	0.6182	100	0.67	0.02
(0.3, 0.7)	(0.7, 0.3)	0.6182	400	0.63	0.01
(0.1, 0.9)	(0.9, 0.1)	0.34	20	0.48	0.24
(0.1, 0.9)	(0.9, 0.1)	0.34	100	0.25	0.7
(0.1, 0.9)	(0.9, 0.1)	0.34	400	0.12	0.03

Conventions as in Table 1. Observe, once again that the empirical mean maximum rewards approach γ^* quickly for the first two cases, as a function of the length of the path, but convergence is much slower for the third case where the distributions $P(\cdot \mid \text{on})$ and $P(\cdot \mid \text{off})$ are very different.

in contrast to the case for the pyramid, the empirical results were *bigger* than the theoretical prediction. We believe that this is because the starting point of the path is unknown for the lattice (it is for the pyramid) and this produces an extra factor which is negligible in the asymptotic regime but which is significant when the size N of the path is too small for the asymptotic results to hold.

4.2. Three Related Vision Tasks

We now consider three additional visual tasks. These tasks were used in Yuille and Coughlan (2000) applied to distinguish between two different i.i.d. textures. The generalization here (see also Wu et al., (2000)) allow the results to apply to the realistic textures generated by Minimax Entropy learning (Zhu et al., 1997), see Fig. 16.

In this subsection we are concerned only with images and so we replace $\mathbf{z}$ by I throughout. Now suppose we have probability distributions, $P_A(I \mid \vec{\beta}_A)$ and $P_B(I \mid \vec{\beta}_B)$, with corresponding potentials $\vec{\beta}_A$, $\vec{\beta}_B$, see Eq. (20) (with same function $h(\cdot)$). For concreteness, the data I can be thought of as being a texture image *but the results are general*.

The results involve two measures of distance between probability distributions: the Chernoff information and the Bhattacharyya bound. To define Chernoff and Bhattacharyya, we must introduce the *e-geodesic*

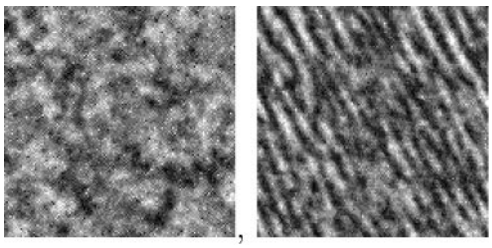

Figure 16. Texture examples, two textures generated by Minimax Entropy learning distributions.

between $P_A(I)$ and $P_B(I)$. This e-geodesic consists of all distributions of form $P_\lambda(I) = P_A^\lambda(I) P_B^{1-\lambda}(I)/Z[\lambda]$ where $0 \leq \lambda \leq 1$ and $Z[\lambda]$ is a normalization constant. The *Chernoff information* is defined by $C(P_A, P_B) = D(P_{\lambda^*} \parallel P_B)$ where λ^* obeys $D(P_{\lambda^*} \parallel P_A) = D(P_{\lambda^*} \parallel P_B)$. The *Bhattacharyya bound* is defined to be $B(P_A, P_B) = (1/2)(D(P_{1/2} \parallel P_A) + D(P_{1/2} \parallel P_B))$ and results if $\lambda = 1/2$. Our results will be summarized in the next section with detailed proofs given in Yuille et al. (2000).

We now consider three texture tasks which involve ways of distinguishing between the two textures. Each task will involve the log-likelihood ratio test $R = \log P_A(I)/P_B(I)$.

Theorem 6. *The negative log probability per pixel that a sample from $P_B(I)$ generates a reward R greater than, or equal to, the average reward $\langle R \rangle_{P_A}$ of a sample from P_A tends to $d(P_A \parallel P_B) = \lim_{N \mapsto \infty}(1/N) D(P_A \parallel P_B)$ as $N \mapsto \infty$. More informally $Pr(R(I) \geq \langle R \rangle_{P_A} \mid I$ drawn from $P_B(\cdot)) \sim e^{-N d(P_A \parallel P_B)}$.*

The second texture task involves determining whether a sample I is generated by P_A or P_B.

Theorem 7. *The negative log probability per pixel that a sample from $P_A(I)$ is misclassified as being from P_B (and vice versa) tends to $c(P_A, P_B) = \lim_{N \mapsto \infty} (1/N) C(P_A, P_B)$ as $N \mapsto \infty$, where $C(P_A, P_B)$ is the Chernoff information. $Pr(R(I) < 0 \mid I$ drawn from $P_A(\cdot)) \sim e^{-N c(P_A, P_B)}$.*

The third texture task involves two texture samples, one each from P_A and P_B, and requires determining which is which.

Theorem 8. *The negative log probability per pixel that the two samples from $P_A(I)$ and $P_B(I)$ (one*

from each) *are misclassified tends to* $b(P_A, P_B) = \lim_{N\mapsto\infty}(1/N)B(P_A, P_B)$ *as* $N \mapsto \infty$, *where* $B(P_A, P_B)$ *is the Bhattarcharyya information. Pr*(*misclassification*) $\sim e^{-Nb(P_A,P_B)}$.

4.3. Detecting Curves in Images

We now return to the task of detecting curves in images. The model we use defined directly on the image lattice (i.e. there is no tree structure). It is chosen to satisfy the conditions for large deviation theory results to apply, see Eq. (20).

The starting point is now unknown (by contrast to the pyramid case in Section 3). This does not affect the theoretical analysis in the asymptotic limit because the number of starting points is only polynomial in the image size (which we take to be a multiple of the target size N).

The target curve position is defined to be $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. The prior model for the road by $P(X) = p(\mathbf{x}_1)\prod_{i=2}^{N} p(\mathbf{x}_i|\mathbf{x}_{i-1})$ (the prior is chosen to prevent the curve from ever intersecting itself). In some cases we extend this to a second order Markov chain prior determined by distributions such as $p(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{x}_{i-2})$.

To define the likelihood function we first choose three filters:

$$\begin{aligned} F^1(I(\mathbf{x})) &= \vec{\nabla} I(\mathbf{x}) \cdot \vec{\hat{t}}(\mathbf{x}) \quad \text{if } \mathbf{x} \in X, = \vec{\nabla} I \cdot \vec{i} \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \\ F^2(I(\mathbf{x})) &= \vec{\nabla} I(\mathbf{x}) \cdot \vec{\hat{n}}(\mathbf{x}) \quad \text{if } \mathbf{x} \in X, = \vec{\nabla} I \cdot \vec{j} \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \\ F^3(I(\mathbf{x})) &= I(\vec{x}) \end{aligned} \tag{22}$$

where $\vec{\hat{t}}(\mathbf{x}), \vec{\hat{n}}(\mathbf{x})$ are the tangent and normal to the curve at $\mathbf{x}$, and $\vec{i}$, $\vec{j}$ are the horizontal and vertical unit vectors of the image plane. The curve X has N pixels and there are a total of M pixels in the entire image. In our simulations we typically allow F_3 to have eight components (i.e. the images have eight grey-level values) and F_1, F_2 are quantized to have six components.

We define $\{\vec{h}^{\alpha}_{\text{on}}(I), \vec{h}^{\alpha}_{\text{off}}(I) : \alpha = 1, 2, 3\}$ to be the empirical histograms of the filters $\{F^\alpha : \alpha = 1, 2, 3\}$ evaluated *on-curve* and *off-curve* for an image I (where α labels the filters F^1, F^2, ...). More precisely, $h^{\alpha}_{\text{on},z}(I) = \frac{1}{N}\sum_{\mathbf{x}\in X}\delta_{z,F^\alpha(I(\mathbf{x}))}$ are the components—indexed by z—of the vector $\vec{h}^{\alpha}_{\text{on}}$, and similarly for $\vec{h}^{\alpha}_{\text{off}}$, $h^{\alpha}_{\text{off},z} = \frac{1}{M-N}\sum_{\vec{x}\notin X}\delta_{z,F^\alpha(I(\mathbf{x}))}$ are the components—indexed by z—of the vector $\vec{h}^{\alpha}_{\text{on}}$. The likelihood function is then given by:

$$P(I \mid X) = \frac{1}{Z} e^{\Sigma_{\alpha=1}^{3}\{N\vec{\beta}^{\alpha}_{\text{on}}(I)\cdot\vec{h}^{\alpha}_{\text{on}} + (M-N)\vec{\beta}^{\alpha}_{\text{off}}\cdot\vec{h}^{\alpha}_{\text{off}}(I)\}}, \tag{23}$$

which we can express in terms of the curve position $\mathbf{x}_1, \ldots, \mathbf{x}_N$ as:

$$P(I|X) \propto e^{\Sigma_\alpha \Sigma_i \{\beta^{\alpha}_{\text{on}}(F^\alpha(I(\mathbf{x}_i))) - \beta^{\alpha}_{\text{off}}(F^\alpha(I(\mathbf{x}_i)))\}}, \tag{24}$$

where $\beta^{\alpha}_{\text{on},z}$, $\beta^{\alpha}_{\text{off},z}$ are the components of $\vec{\beta}^{\alpha}_{\text{on}}$, $\vec{\beta}^{\alpha}_{\text{off}}$.

This gives an overall reward function:

$$\begin{aligned} R(X \mid I) = \sum_i \log p(\mathbf{x}_i \mid \mathbf{x}_{i-1}) + \sum_\alpha \sum_i \\ \times \left\{\beta^{\alpha}_{\text{on}}(F^\alpha(I(\mathbf{x}))) - \left(\beta^{\alpha}_{\text{off}}(F^\alpha(I(\mathbf{x}))\right)\right\}. \end{aligned} \tag{25}$$

To specify the model uniquely we can *either choose the potentials directly* or use Minimax Entropy learning (Zhu et al., 1997; Zhu, 1999) to *learn the potentials from a set of empirical histogram responses*. We tried both approaches and noticed no significant differences in results. Note that because our problem can be approximated as being one-dimensional, we used a recursive algorithm to estimate the potentials, as required by Minimax Entropy learning, instead of the MCMC methods used by Zhu et al. (1997).

We obtain order parameters for these models using Theorem 5. But calculating the order parameters required estimating the Kullback-Leibler distances. We again exploited the one-dimensional structure to compute these order parameters recursively. These order parameters have contributions both from the geometry and the pixel intensity information, see Yuille et al. (2000) for details. Figure 17 shows the results of simulating from the curve model for different distributions.

4.4. The Wrong Reward Function

We now return to the question of what happens if we have a weak approximate model of the probability distributions, see subsection 3.4. We are now able to generalize our previous results and show how order parameters change when a weaker model is used.

In particular, we demonstrate an important connection to Amari's theory of information geometry (Amari, 1982) and to Minimax Entropy learning (Zhu et al., 1997). The approximations can be viewed as projections in probability space (Amari, 1982).

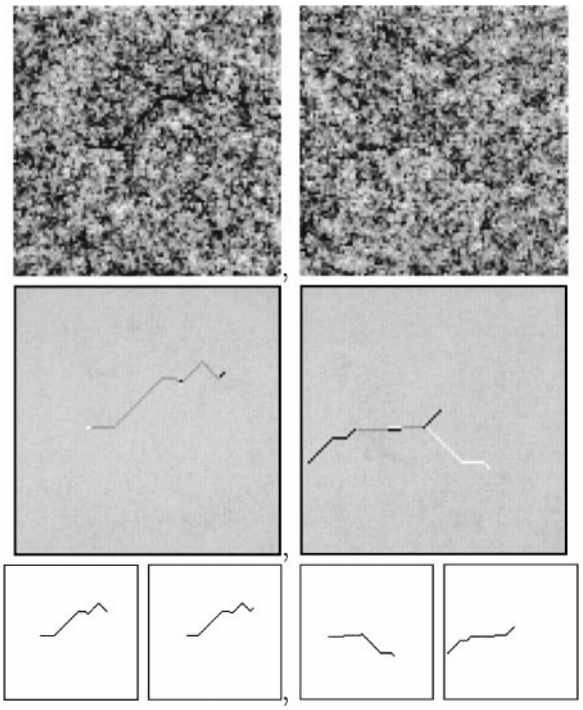

Figure 17. (Top) Samples from the Minimax Entropy curve model, $K = 1.00$ on left and $K = -0.43$ on right. (Middle) The true curve positions for the corresponding samples are shown in white. The solution path, found by dynamic programming, is in black. Places where the solution overlaps with the true path are shown in grey. (Bottom) The true path and the solution for $K = 1.0$ (far left, and left). The true path and the solution for $K = -0.43$ (right, and far right). Observe that for positive K, on the left, the solution is very close to the true path. But if K is negative, on the right, then the solution is very different from the true path—i.e. the task becomes impossible. The order parameters calculated for the models are consistent with the results. The best paths are determined by optimizing the reward functions using a dynamic programming algorithm that does not require known starting point.

Minimax Entropy learning (Zhu et al., 1997) naturally gives rise to a sequence of increasingly accurate Gibbs distributions by pursuing additional features and statistics. The sequence $P_0 = U, P_1, P_2, \ldots, P_k \rightarrow P_{\text{true}}$ (where k is the number of features and statistics included in the model P_k) starts with p_0 being a uniform distribution U and approaches the true distribution P_{true} in the limit as $k \mapsto \infty$ (Zhu et al., 1997). The more high-level (i.e. target specific) the model then the more target specific the statistics. Conversely, low-level (i.e. general purpose) models will only use those statistics which are common to many targets. More precisely, each Gibbs distribution P_i is an *Amari projection* (Amari, 1982) of the "true"

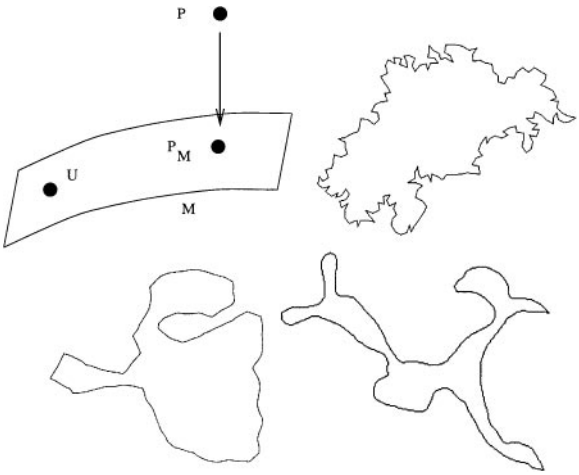

Figure 18. The *Amari projection* and a sequence of prior models for animate object shapes by minimax entropy using an increasing number of feature statistics. See text for interpretation.

distribution P_{true} onto the sub-manifold M_i, with P_i being the closest element to P_{true} in M_i, in terms of Kullback-Leibler divergence $D(P_{\text{true}} \parallel P_i)$, see Fig. 18. Distributions related by *Amari projection* will also satisfy the *Amari condition* described in Section 3.4—i.e. $\sum_t P_{\text{true}}(t) \log P_i(t) = \sum_t P_i(t) \log P_i(t)$. (But the converse is not true). As shown in Fig. 18, the first row, from left to right are typical shapes sampled from three minimax entropy models (Zhu, 1999): a uniform model, a model matching contour based statistics, and a model matching both contour and region based statistics.

For simplicity, recall that in Theorem 6 of subsection 4.2 we gave the probability that a sample I from $P_B(\cdot)$ has higher reward than the expected reward of a smaple from $P_A(\cdot)$. Now approximate the distribution $P_A(\cdot)$ by $P_{\hat{A}}(\cdot)$. We compute the expected reward $\langle \hat{R} \rangle_{P_A} = D(P_{\hat{A}} \parallel P_B)$ if the data is generated by $P_A(\cdot)$ and estimate the probability that data generated by P_B will have higher reward. We assume the Amari condition $\sum_I P_A(I) \log P_{\hat{A}}(I) = \sum_I P_{\hat{A}}(I) \log P_{\hat{A}}(I)$ and the additional condition $\sum_I \log P_B(I)\{P_{\hat{A}}(I) - P_A(I)\} = 0$ (for example, this is satisfied if P_B is the uniform distribution). More general conditions are described in Yuille et al. (2000).

Now we ask, what is the probability that we get a sample I from $P_B(\cdot)$ with reward $\hat{R}(I) > \langle \hat{R} \rangle_{P_A}$? The problem can be formulated as in Theorem 6 of the previous section. *The only difference is that, because* $\langle \hat{R} \rangle_{P_A} = D(P_{\hat{A}} \parallel P_B)$, *we can replace* P_A *by* $P_{\hat{A}}$ *everywhere in the calculation.*

We therefore obtain that the probability of error goes like $\sim e^{-D(P_{\hat{A}} \parallel P_B)}$. This means that the order parameter is higher by an amount $D(P_A \parallel P_B) - D(P_{\hat{A}} \parallel P_B)$ when we use the "correct" reward function. This can be expressed as:

$$\begin{aligned} & D(P_A \parallel P_B) - D(P_{\hat{A}} \parallel P_B) \\ & = \sum P_A \log \frac{P_A}{P_B} - \sum P_{\hat{A}} \log \frac{P_{\hat{A}}}{P_B}, \\ & = D(P_A \parallel P_{\hat{A}}) + \sum \log P_B \{P_{\hat{A}} - P_A\}, \end{aligned}$$

where we have used the Amari condition $\sum P_A \log P_{\hat{A}} = \sum P_{\hat{A}} \log P_{\hat{A}}$.

Using the condition $\sum \log P_B \{P_{\hat{A}} - P_A\} = 0$ we see that the order parameter increases by $D(P_A \parallel P_{\hat{A}})$ when we use the correct reward function. *This is precisely the entropy criterion used in Minimax Entropy learning in determining the benefit of using an additional statistic because* $H(P_{\hat{A}}) - H(P_A) = D(P_A \parallel P_{\hat{A}})$! This demonstrates that accurate prior models increase the order parameters.

4.5. Experimental Curves with Amari Projection

In this section we consider the effects of using the wrong prior. More specifically, we will consider two possible geometry priors P_H and P_G related by an Amari projection, $\sum_X P_H(X) \log P_G(X) = \sum_X P_G(X) \log P_G(X)$. We call $P_H(X)$ the *high-level* model and it is used to generate the data (i.e. it is the "true prior"). By contrast, $P_G(X)$ is called the *generic prior* (i.e. it is the "wrong prior").

We will perform inference on the data in two ways. Firstly, we use the high-level prior in the reward function (i.e. standard Bayesian inference). Secondly, we will use the generic prior in the reward function. As in Section 3.4, the theory predicts there will be three regimes, *ultra*, *challenging*, and *easy*, see caption of Fig. 19.

In Fig. 20, we consider two high-level models, second order Markov chains, which we call roman road and english road. They are both approximated by the same generic, first order Markov, road model. We illustrate the three different regimes.

5. Summary and Conclusions

This paper formulated target detection in terms of Bayesian inference so that the performance rates can be evaluated by the expected loss. We then investigated how much prior knowledge is needed to detect a target road or curve in the presence of clutter. We used order parameters to determine whether a target could be detected using a general purpose "generic" model or whether a more specific high level model was needed. At critical values of the order parameters the problem becomes unsolvable without the addition of extra prior knowledge. This theory was initially described in CVPR'99 (Yuille and Coughlan, 1999) for

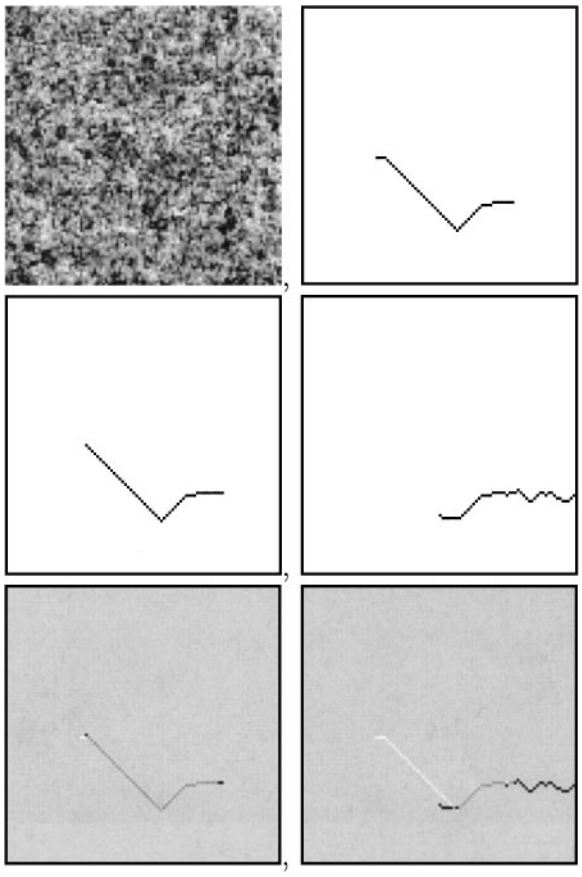

Figure 19. The Challenging regime figure. In the *ultra regime*, detection of the curve is impossible even if the high-level model is used. In the *challenging regime* we will be able to detect the curve if we use the high-level model but *not* if we use the generic model. In the *easy regime*, both models are adequate to detect the curve. The data is shown in the top left square and the true path is shown in the top right square. The results of estimation using the high-level and generic models are shown in the left and right middle squares respectively. Their overlaps with the true path are shown in the bottom two squares (similar conventions to the previous figures). Observe that the high-level model correctly finds the true path (with a few pixels of error) but the generic model fails (apart from finding one small subsection).

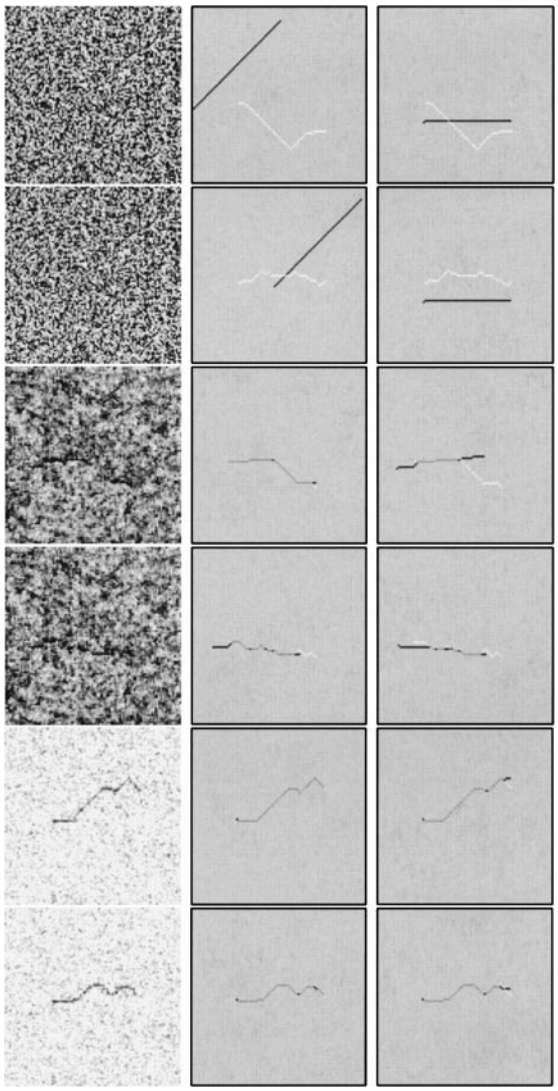

Figure 20. Three panels, of two rows each, top to bottom giving examples of ultra, challenging, and easy regimes. For each panel, the top row gives a sample generated by an *roman road model* (left), the best path found using the *roman road model* (center), and the best path found using the *generic road model* (right). Similarly, for each panel, the bottom row gives a sample generated by an *english road model* (left), the best path found using the *english road model* (center), and the best path found using the *generic road model* (right). In the ultra regime, top panel, no method works. In the challenging regime (centre panel), the high-level models (roman and english) find their targets but the generic models make errors. In the easy regime, everything works.

the restricted class of factorized probability distributions.

Our results hold for a class of probablity distributions which includes those learnt by Minimax Entropy learning theory (Zhu et al., 1997; Zhu, 1999). This generalizes our previous results (Yuille and Coughlan, 2000) on factorizable distributions (which also did not address the issue of how much prior information is needed).

The results of this paper were obtained by analysis of the Bayesian ensemble of problem instances. We anticipate that our approach will generalize to other vision problems and can be used to determine performance measures for models in terms of order parameters.

We observe that our results are in a similar spirit to the theoretical analysis by Tsotsos on the complexity of visual search (Tsotsos, 1990). Tsotsos uses techniques from computer science to analyze the complexity of detecting targets in background. This is very different from our Bayesian approach and relationship between these two approaches is a topic for further study.

Hopefully, analysis of the type performed in this paper can help quantify when high-level knowledge is needed for visual tasks. This may throw light into the development of efficient algorithms for segmentation and recognition.

Appendix A: Sanov's Theorem

Sanov's theorem is the main theoretical tool used to obtain our results in Section 3. This appendix describes the theorem and gives examples of how to apply it. We also give an expression for the function $g(\gamma)$ which occurs in Theorem 2 and which determines the critical value γ^*. We refer to Yuille et al. (2000) for a detailed description of how we apply Sanov to prove the results stated in Section 3.

To describe Sanov's theorem we need some notation. The variables z are quantized so that they can take one of a set of J values $a_1, \ldots, a_J$. We refer to $\{a_1, \ldots, a_J\}$ as the *alphabet* and J as the *alphabet size*. A sample $\mathbf{z}$ of N elements $z_1, \ldots, z_N$ can be represented by the histogram $n_1, \ldots, n_J$ of the frequency that each member of the alphabet occurs (i.e. $\sum_{j=1}^{J} n_j = 1$ and a_j occurs $n_j N$ times in the sample $\mathbf{z}$). There are a finite number of histograms which can occur and each possible histogram is called a *type*. Because the data $\mathbf{z}$ is i.i.d. then the probability of it occuring depends only on the probability of its type (i.e. the ordering of the data is irrelevant).

Sanov's Theorem. *Let $z_1, z_2, \ldots, z_N$ be i.i.d. from a distribution $P_s(\mathbf{z})$ 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*

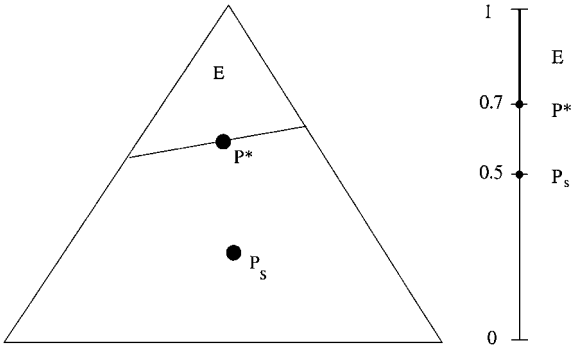

Figure 21. Left, Sanov's theorem. The triangle represents the set of probability distributions. P_s is the distribution which generates the samples. Sanov's theorem states that the probability that a type, or empirical distribution, lies within the subset E is chiefly determined by the distribution P^* 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 $P^*(head) = 0.7$.

lies in the set E. Then:

$$\frac{2^{-ND(\vec{\phi}^* \parallel P_s)}}{(N+1)^J} \leq Pr(\vec{\phi} \in E) \leq (N+1)^J 2^{-ND(\vec{\phi}^* \parallel P_s)}, \quad (26)$$

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

Sanov's theorem can be illustrated by a simple coin tossing example, see Fig. 21. 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 $D(P^* \parallel P_s) = 0.119$. 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 $2^{-119} \times (1001)^2 \leq 2^{-99}$.

To obtain the results of Section 3 requires specifying sets E which corresponds to specific values of the reward function (e.g. let E be the set of types such that the reward of a distractor path is higher than the expected reward of a true path). We then solve the equation $\vec{\phi}^* = \arg\min_{\vec{\phi} \in E} D(\vec{\phi} \parallel P_s)$ to obtain the fall-off rate.

For example, we can apply Sanov's theorem to determine the probability that a sample $\mathbf{z}$ from $P_A(\mathbf{z})$ will have log-likelihood reward $\log P_A(\mathbf{z})/P_B(\mathbf{z})$ which differs from the mean reward $D(P_A \parallel P_B)$ by more than ϵ. In this case, the set E is defined by:

$$E = \{\vec{\phi} : |\vec{\phi} \cdot \vec{\alpha} - D(P_A \parallel P_B)| > \epsilon\}, \quad (27)$$

where the vector $\vec{\alpha}$ has J components $\log P_A(a_i)/P_B(a_i)$ for $i = 1, \ldots, J$.

To apply Sanov's theorem, we have to extremize $D(\phi \parallel P_A)$ subject to the constraint $\vec{\phi} \in E$. By optimization, using lagrange multipliers, we obtain:

$$\phi^\epsilon(a) = \frac{P_A^{\mu(\epsilon)}(a) P_B^{1-\mu(\epsilon)}(a)}{Z(\mu(\epsilon))}, \quad (28)$$

where $Z(\mu(\epsilon))$ is a normalization constant and $\mu(\epsilon)$ is choosen by solving the equation:

$$\vec{\phi}^\epsilon \cdot \vec{\alpha} - D(P_A \parallel P_B) = \sum_{j=1}^{J} \phi^\epsilon(a_j) \log P_A(a_j)/P_B(a_j) - D(P_A \parallel P_B) = \pm\epsilon \quad (29)$$

This equation will have two solutions depending on the sign. We choose the solution for which $D(\phi^\epsilon \parallel P_A)$ is smallest (because this determines the slowest fall-off rate).

The probability of a deviation from the mean greater than ϵ is then less than $2(N+1)^J 2^{-ND(\phi^\epsilon \parallel P_A)}$ for large N and so falls to zero exponentially fast. Note that $\lim_{\epsilon \mapsto 0} D(\phi^\epsilon \parallel P_A) = 0$. In other words, the smaller ϵ the smaller the fall-off factor.

Finally, we give an exact expression for the function $g(\gamma)$ which is referred to in Theorem 2 and whose form determines the critical value γ^*. See Yuille et al. (2000) for the technical derivation of $g(\gamma)$.

The function $g(\gamma)$ is given by:

$$g(\gamma) = \gamma + D(\phi_\gamma \parallel p_{\text{on}}) + D(\psi_\gamma \parallel p_{\Delta G}), \quad (30)$$

which is a monotonically nondecreasing function of γ with:

$$\phi_\gamma = \frac{p_{\text{on}}^{\lambda(\gamma)} p_{\text{off}}^{1-\lambda(\gamma)}}{Z_1(\lambda(\gamma))}, \qquad \psi_\gamma = \frac{p_{\Delta G}^{\lambda(\gamma)} U^{1-\lambda(\gamma)}}{Z_2(\lambda(\gamma))},$$

where $Z_1(\lambda(\gamma))$, $Z_2(\lambda(\gamma))$ are normalization constants and $\lambda(\gamma)$ is chosen so that $\sum_y \phi_\gamma(y) \log \frac{P_{\text{on}}(y)}{P_{\text{off}}(y)} + \sum_x \psi_\gamma(x) \frac{\log P_{\Delta G}(x)}{U(x)} = \gamma$.

The expected number of distractor paths with rewards greater than γ is given by $E[Z(\gamma, N)] \doteq 2^{-N\{g(\gamma)-\log Q\}}$. The critical value γ^* occurs when $g(\gamma) = \log Q$.

Appendix B: Large Deviation Theory

The results in Section 4 require techniques from large deviation theory (Dembo and Zeitouni, 1998) which we summarize in this appendix and refer to Yuille et al. (2000) for more details.

For probability distributions of the form specified by Eqs. (20) and (21) the analysis becomes simplified as the image, and/or target size, becomes large (Lewis et al., 1995). Intuitively, this is *because the probability distribution in feature space becomes peaked as the size increases due to ergodicity*. Moreover, the theory gives results on how fast the distributions become peaked as N gets large. Recall that the equations are:

$$P(\mathbf{z} \mid \vec{\beta}) = \frac{e^{-N\vec{\beta}\cdot\vec{h}(\mathbf{z})}}{Z(\vec{\beta})},$$

$$\hat{P}(\vec{h} \mid \vec{\beta}) = |\Omega_N(\vec{h})| \frac{e^{-N\vec{\beta}\cdot\vec{h}}}{Z(\vec{\beta})},$$

We first state two limit results from the large deviation theory literature (Lewis et al., 1995; Griffiths and Ruelle, 1971).

Lemma 1. $\lim_{N\mapsto\infty} \frac{1}{N} \log \frac{|\Omega_N(\vec{h})|}{Q^N} = s(\vec{h})$, *where* $s(\vec{h}) \leq 0$ *is a concave function.*

Lemma 2. $\lim_{N\mapsto\infty} \frac{1}{N} \log \frac{Z(\vec{\beta})}{Q^N} = \rho(\vec{\beta})$ *where the "pressure"* $\rho(\vec{\beta})$ *is strictly convex.*

From these Lemmas we can determine directly the probabilities of rare events for large N. First, observe that the form of the induced distribution in feature space must obey:

Corollary 1. $\lim_{N\mapsto\infty} \frac{1}{N} \log \hat{P}(\vec{h} \mid \vec{\beta}) = s(\vec{h}) - \vec{\beta} \cdot \vec{h} - \rho(\vec{\beta})$.

This corollary implies that, for large N, $\hat{P}(\vec{h} \mid \vec{\beta}) \sim e^{N\{s(\vec{h})-\vec{\beta}\cdot\vec{h}-\rho(\vec{\beta})\}}$. This shows exponential fall-off for large N.

The concavity of $s(\vec{h})$, and hence of $s(\vec{h}) - \vec{\beta} \cdot \vec{h} - \rho(\vec{\beta})$ means that one $\vec{h}$ dominates for large N. More precisely,

Corollary 2. $\lim_{N\mapsto\infty} \frac{1}{N} \log \hat{P}(\vec{h} \in H \mid \vec{\beta}) = s(\vec{h}^*_H) - \vec{\beta} \cdot \vec{h}^*_H - \rho(\vec{\beta})$, *where* $\vec{h}^*_H = \arg\max_{\vec{h}\in H}\{s(\vec{h}) - \vec{\beta} \cdot \vec{h} - \rho(\vec{\beta})\}$.

For example, H could consist of the set of rare events that would cause misclassification (e.g. by log-likelihood ratio tests) and hence $\hat{P}(\vec{h} \in H \mid \vec{\beta}) \sim e^{Ns(\vec{h}^*_H)-\vec{\beta}\cdot\vec{h}^*_H-\rho(\vec{\beta})}$ says we only need to be concerned with the *single most likely rare event in* H, see Fig. 22.

These results can be used to give asymptotic expressions on the expected loss for visual tasks. They are therefore generalizations of Sanov's theorem which we used in the previous section for the i.i.d. case. There is, however, one important distinction. Sanov's theorem gives *tight bounds* on the expected errors as a function of the number N of samples. The results in this section are *asymptotic* only (i.e. only valid in the limit of infinite N). This limitation is not a major concern but it does reduce the power of our results for the non-iid case.

The results above are expressed in terms of the probability of observing certain statistics. There is, however,

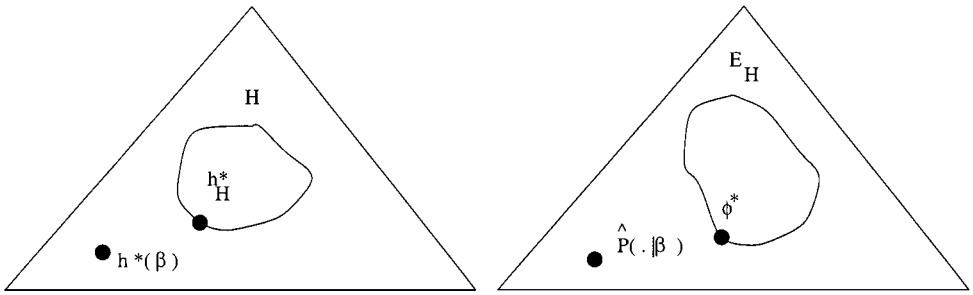

Figure 22. The left panel illustrates Corollary 2—each point is a statistic $\vec{h}$, H is a set of statistics, and $\vec{h}^*_H$ is the dominant statistic in H. The right panel uses duality to give the same result expressed in terms of distributions, see Yuille et al. (2000)—each point is a probability distribution with the set E_H of distributions corresponding to the set H of statistics, and with ϕ^* corresponding to $\vec{h}^*_H$.

a *duality* between the statistics $\vec{h}$ and the potentials $\vec{\beta}$ (which determine the probability distributions), see Langford (1973). Corollary 1 says that for any fixed $\vec{\beta}$ there will be a unique value $\vec{h}^*$ of the statistics which dominate $\hat{P}(\vec{h} \mid \vec{\beta})$ for large N. Conversely, for any value of the statistics we can determine a corresponding $\vec{\beta}$ (i.e. by finding the distribution which gets peaked at this value of the statistics). (There will be uniqueness up to simple transformations). By using this duality we can re-express these results in terms reminiscent of Sanov's theorem such as Kullback-Leibler divergences ($D(P_A \parallel P_B) = \sum_I P_A(I) \log P_A(I)/P_B(I)$) between probability distributions. See Yuille et al. (2000) for this analysis.

Acknowledgments

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 ALY.

Note

1. The term "ensemble" is sometimes used to refer to the large N limit (i.e. large systems with many degrees of freedom) but we do not restrict ourselves to this limit.

References

Amari, S. 1982. Differential geometry of curved exponential families—Curvature and information loss. *Annals of Statistics*, 10(2):357–385.

Amit, D.J. 1989. *Modelling Brain Function: The World of Attractor Neural Networks*. Cambridge University Press: Cambridge England.

Barron, J.L., Fleet, D.J., and Beauchemi, S.S. 1994. Systems and experiment performance of optical flow techniques. *Int'l Journal of Computer Vision*, 12(1):43–77.

Bowye, K.W. and Phillips, J. (Eds.), 1998. *Empirical Evaluation Techniques in Computer Vision*. IEEE Computer Society Press.

Coughla, James M. and Yuille, A.L. 1999. Bayesian A* tree search with expected O (N) convergence rates for road tracking. In *Proceedings EMMCVPR'99*, pp. 189–204. Springer-Verlag Lecture Notes in Computer Science 1654.

Cove, T.M. and Thomas, J.A. 1991. *Elements of Information Theory*. Wiley Interscience Press: New York.

DeGroot, M.H. 1970. *Optimal Statistical Decisions*. McGraw-Hill: New York.

Dembo, A. and Zeitouni, O. 1998. *Large Deviation Techniques and Applications* (2nd ed.). Springer: New York.

Geiger, D. and Liu, T.-L. 1997. Top-down recognition and bottom-up integration for recognizing articulated objects. In *EMMCVPR'97*, M. Pellilo and E. Hancock (Eds.), pp. 295–310. Springer-Verlag, CS 1223.

Geman, D. and Jedynak, B. 1996. An active testing model for tracking roads in satellite images. *IEEE Trans. Patt. Anal. and Machine Intel.*, 18(1):1–14.

Green, D.M. and Swets, J.A. 1988. *Signal Detection Theory and Psychophysics* (2nd ed.). Peninsula Publishing: Los Altos, California.

Grenander, U., Miller, M.I., and Srivastav, A. 1998. Hilbert-Schmidt lower bounds for estimators on matrix Lie groups for ATR. *IEEE Trans. Patt. Anal., and Machine Intel.*, 20(8):790–802.

Griffiths, R. and Ruelle, D. 1971. Strict convexity ("continuity") of the pressure in lattice systems. *Comm. Math. Phys.*, 23:169–175.

Heath, M., Sarkar, S., Sanocki, T., and Bowyer, K.W. 1997. A robust visual method for assessing the relative performance of edge detection algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(12):1338–1359.

Hoover, A.W., Jean-Baptiste, G., Xiaoyi, Jiang, Flynn, P.J., Bunke, H., Goldgof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A., and Fishe, R.B. 1996. An experimental comparison of range image segmentation algorithms. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 18(7):1–17.

Kass, M., Witkin, A., and Terzopoulos, D. 1987. Snakes: Active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, pp. 259–268.

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

Konishi, S., Yuille, A.L., Coughlan, J.M., and Zhu, S.C. 1999. Fundamental bounds on edge detection: An information theoretic evaluation of different edge cues. In *Proc. Int'l Conf. on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, pp. 573–579.

Lanford, O.E. 1973. Entropy and equilibrium states in classical mechanics. In *Statistical Mechanics and Mathematical Problems*, A. Lenard (Ed.), Springer, Berlin, Germany.

Lewis, J.T., Pfister, C.E., and Sullivan, W.G. 1995. Entropy, concentration of probability, and conditional limit theorems. *Markov Processes Relat. Fields*, 1:319–396, Moscow, Russia.

Murray, M.K. and Rice, J.W. 1993. *Differential Geometry and Statistics*. Chapman and Hall.

O'Sullivan, J.A., Blahut, R.E., and Snyde, D.L. 1998. Information-theoretic image formulation. *IEEE Transactions on Information Theory*, 44(6)

Pearl, J. 1984. *Heuristics*. Addison-Wesley.

Rajagopalan, A.N. and Chaudhuri, S. 1998. Performance analysis of maximum likelihood estimator for recovery of depth from defocused images and optimal selection of camera parameters. *International Journal of Computer Vision*, 30(3):175–190.

Ratches, J.A., Walters, C.P., Buse R.G., and Guenther, B.D. 1997. Aided and automatic target recognition based upon sensory inputs from image forming systems. *IEEE Trans. on PAMI*, 19(9).

Szelisk, R. and Kang, S.B. 1997. Shape ambiguities in structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):506–512.

Tsotsos, J.K. 1990. Analyzing vision at the complexity level. *Behavioural and Brain Sciences*, 13(3):423–469.

Vapnik, V.N. 1998. *Statistical Learning Theory*. John Wiley and Sons: New York.

Wu, Y., Zhu, S.C., and Liu, X.W. 2000. Equivalence of image ensembles and FRAME models. *International Journal of Computer Vision*, 38(3):245–261.

Young, G. and Chellappa, R. 1992. Statistical analysis of inherent ambiguities in recovering 3D motion from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):995–1013.

Yuille, A.L. and Coughlan, J.M. 2000. Fundamental limits of bayesian inference: Order parameters and phase transitions for road tracking. *Pattern Analysis and Machine Intelligence* PAMI, 22(2):160–173.

Yuille, A.L. and Coughlan, J. 2000. An A* perspective on deterministic optimization for deformable templates. *Pattern Recognition*, 33(4):603–616.

Yuille, A.L. and Coughlan, James M. 1999. 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, pp. 631–637.

Yuille, A.L., Coughlan, James M., and Zhu, S.C. 2000. A unified framework for performance analysis of bayesian inference. In *Proceedings SPIE*, Orlando, Florida, pp. 333–346.

Yuille, A.L., Coughlan, James M., Wu, Y.N., and Zhu, S.C. 2000. Order parameters for minimax entropy distributions. Smith-Kettlewell Technical Report (yuille@ski.org).

Zhu, S.C., Wu, Y., and Mumford, D. 1997. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660.

Zhu, S.C. 1999. Embedding gestalt laws in markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (PAMI), 21(11):1170–1187.