Bayesian A* Tree Search with Expected O(N) Node Expansions: Applications to Road Tracking

James M. Coughlan and A. L. Yuille

Smith-Kettlewell Eye Research Institute,

2318 Fillmore Street,

San Francisco, CA 94115, USA.

Tel. (415) 345-2144. Fax. (415) 345-8455.

Email yuille@ski.org, coughlan@ski.org

December 27, 2002

Abstract *Many perception, reasoning, and learning problems can be expressed as Bayesian inference. We point out that formulating a problem as Bayesian inference implies specifying a probability distribution on the ensemble of problem instances. This ensemble can be used for analyzing the expected complexity of algorithms and also the algorithm-independent limits of inference. We illustrate this problem by analyzing the complexity of tree search. In particular, we study*

the problem of road detection, as formulated by Geman and Jedynak (1986). We prove that the expected convergence is linear in the size of the road (the depth of the tree) even though the worst case performance is exponential. We also put a bound on the constant of the convergence and place a bound on the error rates.

Keywords: (I) Heuristic Search, (II) A*, (III) Order Parameters and Phase Transitions, (IV) NP-complexity versus Typical Complexity, (V) Bayesian Computer Vision.

Draft Submitted to Neural Computation.

1 Introduction

Many problems in vision, speech, reasoning, and other sensory and control modalities can be formulated as Bayesian inference (Knill and Richards 1996). It is important to understand the complexities of algorithms which can perform these inferences.

We point out that formulating a problem as Bayesian inference implies specifying a probability distribution on the *ensemble of problem instances*. More formally, in Bayesian inference the goal is to estimate a quantity x from data y by using the *posterior* distribution $P(x|y)$. Constructing this posterior requires specifying a *likelihood function* $P(y|x)$ and a *prior* distribution $P(x)$. From these distributions we can construct a distribution $P(x, y)$ on the ensemble of problem instances (x, y). See figure (4) for samples from a particular ensemble for road tracking.

There are two main advantages to analyzing the performance of an algorithm over the ensemble of problem instances. Firstly, it allows us to determine the behaviour of the algorithm for typical problem instances (i.e. those which occur with non-negligible probability) and means that we may not have to deal with worst case situations (because they may have arbitrarily small probabilities). Secondly, having a distribution over the ensemble of problem instances also enables us to quantify the accuracy of the estimates found by the algorithm.

Many problems in Artificial Intelligence can be formulated as tree search (Winston 1984, Pearl 1984, Russell and Norvig 1995). We now study a specific example of tree search: a computer vision algorithm proposed by Geman and Jedynak

(1996). (Our analysis, however, can be extended to other tree searching problems). Geman and Jedynak addressed the problem of detecting roads in aerial images. They formulated the problem as Bayesian maximum a posteriori (MAP) estimation (which enables us to determine a distribution on the ensemble of problem instances). In this paper we analyze the complexity of a variant of the Geman and Jedynak algorithm (this variant was proposed by the authors in (Yuille and Coughlan 2000b)). The complexity, and performance, of the algorithm depends on the probability distributions which characterize the ensemble of problem instances (i.e. the likelihood function and the prior). For a large class of ensembles, we are able to prove that the expected complexity is linear in the length N of the road (by contrast, the worst case complexity is exponential in N). We are also able to put bounds on the expected errors made by the algorithm.

We emphasize that we are concerned with the ability of the algorithm to detect the road path *which may not necessarily correspond to the MAP estimate*. For any problem instance there are three important paths: (i) the true road path, (ii) the MAP estimate of the true road path, and (iii) the path found by the algorithm. In this paper we are concerned with the difference between (iii) and (i) only.

These results complement our previous work (Yuille and Coughlan 2000a, Yuille, Coughlan, Wu and Zhu 2001) which used this ensemble concept to analyze the *algorithm-independent* performance of MAP estimation on problems of this type (i.e. we evaluated errors between the MAP estimate (ii) and the true road path (i)). In particular, we derived an *order parameter* K_B which is a function of the ensemble. We proved that if $K_B < 0$ then it is impossible to detect the true

road by *any algorithm*. (Our results in this paper apply to cases where $K_B > 0$ only. We can express $K_B = \{\psi_1 - \log Q\} + \psi_2$ where ψ_1, ψ_2 are positive quantities (defined in Theorem 3). We show linear complexity provided $\psi_1 > \log Q$. If $\psi_1 < \log Q$ and $\psi_1 + \psi_2 > \log Q$ then the target path can be found but we can say nothing about the algorithm complexity).

Another advantage of the concept of an ensemble of problem instances is illustrated by our choice of a *heuristic A** algorithm (Yuille and Coughlan 1999). A* algorithms (Pearl 1984, Winston 1984, Russell 1995) search trees and/or graphs using a *heuristic* to estimate future rewards. If we are using a Bayesian ensemble then the probability distributions can be used to generate heuristics.

Technically, our proofs make use of *large deviation theory* (Grimmett and Stirzaker 1992). In particular we use Sanov's theorem, see Appendix A, to put bounds on the probability of rare events. We note that many results on statistical learning theory (Vapnik 1998) are derived using similar techniques from large deviation theory.

We now mention two related classes of theoretical results which are highly relevant.

Firstly, Karp and Pearl (1983) (see also Pearl 1984) provided a theoretical analysis of convergence rates of A* search by considering an ensemble of problem instances. They studied a binary tree where the rewards for each arc were 0 or 1 and were specified by a probability p. They then obtained the complexity of algorithms for finding the minimum cost path. This work has some similarities to ours but their formulation is not Bayesian, their heuristics for A* algorithms are

different, and large deviation theory is not used. Their work was an inspiration for us and we provided an analysis of a block pruning algorithm motivated by them in (Yuille and Coughlan 1999).

Secondly, we would like to mention related work on optimization which uses the concept of ensembles. There are some recent studies showing that order parameters exist for NP-complete problems and that these problems can be easy to solve for certain values of the order parameters (Cheeseman, Kanefsky and Taylor 1991, Selman and Kirkpatrick 1996). This work involves analyzing ensembles of problem instances. But the distribution of instances in the ensemble is typically assumed to be uniform and is not derived from Bayesian methods.

The structure of this paper is as follows. In Section (2) we formulate the road tracking problem as tree search and introduce A*. Section (3) gives complexity results for a special choices of A* heuristic (making use of Sanov's theorem which is given in Appendix A). In Section (4), we extend the complexity results to other heuristics (using additional results proven in Appendix B). Section (5) proves that sorting the queue for A* takes constant expected time per sort operation. (An early version of this work was presented in Coughlan and Yuille 1999).

2 Problem Formulation

2.1 Tree Search: the Geman and Jedynak Model

Many problems in Artificial Intelligence can be formulated as tree search (Winston 1984, Pearl 1984, Russell and Norvig 1995). We now study a specific example of this class of problem.

Geman and Jedynak (1996) formulate road detection as tree search through a Q-nary tree, see figure (1). The starting point and initial direction are specified and there are Q^N possible distinct paths down the tree. The goal is to find the road path (whose statistical properties differ from those of the non-road paths). The worst case complexity for this problem is exponential but, as we will prove, in many circumstances it is possible to detect a good approximation to the road path in linear expected time. Our analysis involves considering an ensemble of problem instances.

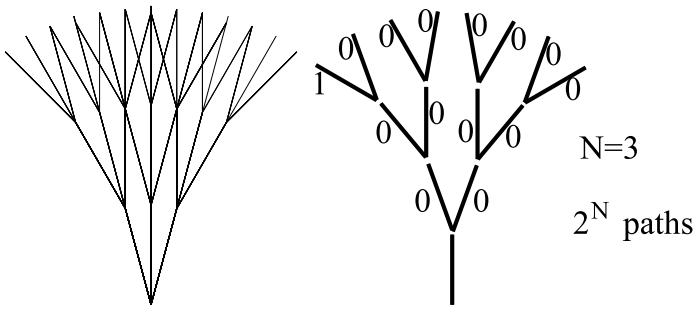

Figure 1: Left Panel: Geman and Jedynak's tree structure with a branching factor of $Q = 3$. Right Panel: The worst case complexity is exponential because, for some problem instances, the best path is determined only by the reward of the final arc segment. In this panel $Q = 2$ and all the 2^N paths have to be examined.

More formally, a road hypothesis, or path, consists of a set of connected straight-line *segments*. We can represent a path by a sequence of moves $\{t_i\}$ on the tree. Each move t_i belongs to an *alphabet* $\{b_\nu\}$ of size Q. For example, the simplest case studied by Geman and Jedynak sets $Q = 3$ with an alphabet b_1, b_2, b_3 corresponding to the decisions: (i) b_1 – go straight (0 degrees), (ii) b_2 – go left (-5 degrees), or (iii) b_3 – go right (+ 5 degrees).

Each tree will contain a *target path* which corresponds to the road to be detected. This path is sampled from a prior probability distribution $P(\{t_i\}) = \prod_{i=1}^{N} P_{\Delta G}(t_i)$, where $P_{\Delta G}(.)$ is the geometric transition probability. For our $Q = 3$ example, we may choose to go straight, left or right with equal probability (i.e. $P_{\Delta G}(b_1) = P_{\Delta G}(b_2) = P_{\Delta G}(b_3) = 1/3$), see figure (2).

A sequence of moves $\{t_i : i = 1, ..., N\}$ determines a path of segments $X = (x_1, ..., x_N)$ (ie. these segments form a connected path from the top of the tree to the bottom). Conversely, a consistent path X determines a sequence of moves. (This also applies to subpaths). Let χ denote all the Q^N segments of the tree. So a path X is a subset of χ. The set of segments not on the path is the complement $\chi \setminus X$.

There is an *observation* y_x for each segment $x \in \chi$ of the tree. The set of all observations is $Y = \{y_x : x \in \chi\}$. The values of the observations belong to an alphabet $\{a_\mu\}$ of size J. An observation y_x is drawn from a distribution $P_{on}(.)$ if the segment is on the target path (ie. if $x \in X$). If not, it is drawn from $P_{off}(.)$. For any path $\{t_i\}$ through the tree, with segments $\{x_i\}$, we have a corresponding set of observations $\{y_{x_i}\}$

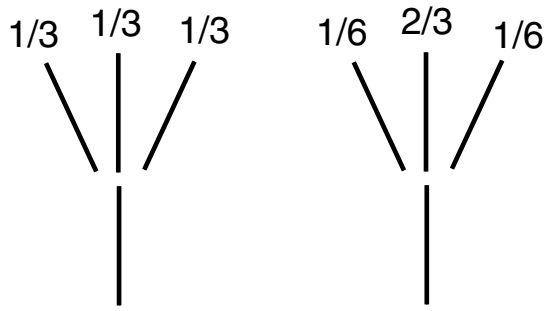

Figure 2: Different priors for the geometry. (Left Panel) the probabilities of turning left, right, or straight are 1/3. (Right Panel) the probability of going straight is 2/3 and the probabilities of turning right or left are 1/6 each, biasing towards straighter paths.

This determines the likelihood function $P(Y|X)$:

$$P(Y|X) = \prod_{x \in X} P_{on}(y_x) \prod_{x \in \chi \setminus X} P_{off}(y_x), \tag{1}$$

which we can re-express as:

$$P(Y|X) = \prod_{i=1,\ldots,N} \frac{P_{on}(y_{x_i})}{P_{off}(y_{x_i})} \prod_{x \in \chi} P_{off}(y_x) = \prod_{i=1,\ldots,N} \frac{P_{on}(y_{x_i})}{P_{off}(y_{x_i})} F(Y). \tag{2}$$

where $F(Y) = \prod_{x \in \chi} P_{off}(y_x)$ is independent of the target path X.

We formulate the problem as MAP estimation to find the mode of the posterior distribution $P(X|Y) = P(Y|X)P(X)/P(Y)$ where $P(X) = \prod_{i=1}^{N} P_{\Delta G}(t_i)$ (where $\{t_i\}$ is the sequence of moves that generates the path X).

Then MAP estimation for X is equivalent to maximizing

$$P(Y|X)P(X) = \prod_{i=1}^{N} P_{\Delta G}(t_i) \prod_{i=1,\ldots,N} \frac{P_{on}(y_{x_i})}{P_{off}(y_{x_i})} F(Y). \tag{3}$$

This is equivalent to finding the path $\{t_i\}$ with observations $\{y_i\}$ which maximizes the following *reward function*:

$$r(\{t_i\},\{y_i\}) = \sum_{i=1}^{N} \log\{\frac{P_{on}(y_i)}{P_{off}(y_i)}\} + \sum_{i=1}^{N} \log\{\frac{P_{\Delta G}(t_i)}{U(t_i)}\}, \tag{4}$$

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

Observe that the reward of a particular path depends only on the variables $\{t_i\},\{y_i\}$ which define the path (the moves and the observations). This is because the factor $F(Y)$ in $P(Y|X)$ is independent of X and can be ignored (ie. it does not affect which path is most probable).

For any path (or subpath) in the tree of length n we will call $r(\{t_i\},\{y_i\})$ the *reward function*. It can be re-expressed as:

$$r(\{t_i\},\{y_i\}) = n\vec{\phi}\cdot\vec{\alpha} + n\vec{\psi}\cdot\vec{\beta}, \tag{5}$$

where $\vec{\alpha}$ and $\vec{\beta}$ have components:

$$\alpha_\mu = \log\frac{P_{on}(a_\mu)}{P_{off}(a_\mu)},\ \mu = 1,...,J\ ,\ \beta_\nu = \log\frac{P_{\Delta G}(b_\nu)}{U(b_\nu)},\ \nu = 1,...,Q. \tag{6}$$

and $\vec{\phi}$ and $\vec{\psi}$ are normalized histograms, or *types* (see Appendix A), with components (with $\delta_{i,j}$ denoting the Kronecker delta function):

$$\phi_\mu = \frac{1}{n}\sum_{i=1}^{n}\delta_{y_i,a_\mu},\ \mu = 1,...,J\ ,\ \psi_\nu = \frac{1}{n}\sum_{i=1}^{n}\delta_{t_i,b_\nu}\ \nu = 1,...,Q. \tag{7}$$

The Geman and Jedynak model is an idealization of tracking a road on a lattice. The tree structure represents the set of all possible paths from the starting

point. A sequence of moves $\{t_i\}$ on the tree determines a path of segments in the image. Each segment is approximately 7 pixels long and its direction depends on the angle of the move t_i relative to the direction of the previous segment. The observations y are the responses to 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 trained on examples of on-road and off-road segments to determine empirical distributions $P_{on}(y)$ and $P_{off}(y)$ for the filter responses, for more details see (Geman and Jedynak 1996). See figure (3) for examples of distributions P_{on}, P_{off} (taken from Konishi, Yuille, Coughlan and Zhu 1999).

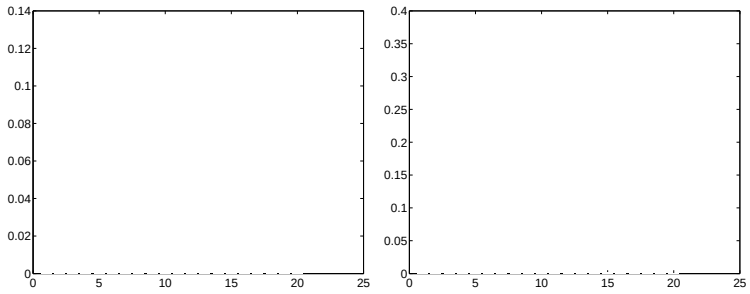

Figure 3: The quantized distributions $P_{on}(y)$ (Left) and $P_{off}(y)$ (Right), where $y = \left|\vec{\nabla} I(\mathbf{x})\right|$, learnt from image data. Observe that, not surprisingly, $\left|\vec{\nabla} I(\mathbf{x})\right|$ is likely to take larger values *on* an edge rather than *off* an edge.

We now illustrate the Bayesian ensemble by figure (4) which consists of three samples from the ensemble. In these cases the target path is easily detectable but noise fluctuations mean that some subpaths may distract the algorithm from the target. In other ensembles, the target path may be far harder to detect.

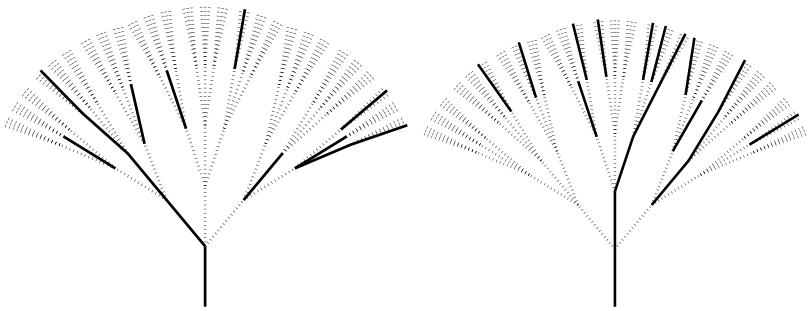

Figure 4: Samples from the Bayesian ensemble. Simulated road tracking problem where dark lines indicate strong edge responses and dashed lines specify weak responses. The data was generated by stochastic sampling using a simplified version of the models analyzed in this paper. In both examples there is only one strong candidate for the best path (the continuous dark line) but chance fluctuations have created subpaths in the noise with strong edge responses.

2.2 Can the task be solved? Distractor Paths

The goal is to detect the target path by selecting the path with highest reward. But the MAP estimate may not necessarily correspond to the target path. In this subsection, we specify conditions which ensure that the MAP estimate is expected to be significantly *similar* to the target path. Unless these conditions are satisfied it will be *impossible to find the target path by any algorithm.*

The tree contains one target path and $Q^N - 1$ distractor paths. We categorize the distractor paths by the stage at which they diverge from the target path, see figure (5). For example, at the first branch point the target path lies on only one of the Q branches and there are $Q - 1$ false branches which generate the first set of false paths F_1. Now consider all the $Q - 1$ false branches at the second target

branch, these generate set F_2. As we follow along the true path we generate sets F_i of size $(Q-1)Q^{N-i}$. The set of all paths is therefore the target path plus the union of the F_i $(i = 1, \ldots, N)$.

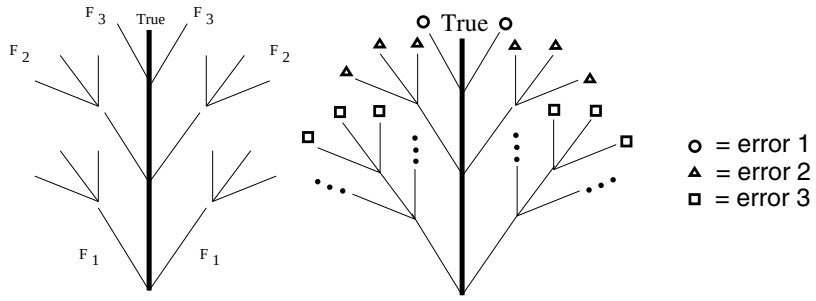

Figure 5: Left: Given a specified target path (the straight line shown in bold in this case) we can divide the set of paths up into N subsets $F_1, ..., F_N$ as shown here. Paths in F_1 have no overlap with the target path. Paths in F_2 overlap with one segment, and so on. Intuitively, we can think of this as an onion where we peel off paths stage by stage. Right: When paths leave the target path they make errors which we characterize by the number of false segments. For example, a path in F_1 has error N, a path in F_i has error $N + 1 - i$.

To determine whether the target path can be found by MAP estimation we must consider the probability that one of the distractor paths has higher reward than the target path. For example, consider the probability distribution $\hat{P}_{1,max}(r_{max}/N)$ of the *maximum* reward (normalized by N) of all the paths in F_1. We can compare this to the probability distribution of the (normalized) reward $\hat{P}_T(r_T/N)$ of the target path. In related work (Yuille, Coughlan, Wu and Zhu 2001), we use techniques similar to Sanov's theorem (see Appendix A) to estimate these quantities and to show that there is a phase transition depending on

a parameter K given by:

$$K = D(P_{on}||P_{off}) + D(P_{\Delta G}||U) - \log Q, \tag{8}$$

where $D(P_{on}|P_{off}) = \sum_y P_{on}(y) \log \frac{P_{on}(y)}{P_{off}(y)}$ is the Kullback-Leibler divergence between P_{on} and P_{off}.

If $K > 0$ then the probability distribution for the target path reward $\hat{P}_T(r_T/N)$ lies to the right of the distribution $\hat{P}_{1,max}(r_{max}/N)$ of the maximum reward of paths in F_1, see figure (6) left panel, and it is straightforward to detect the target path. At $K \approx 0$ the two distributions overlap and it becomes hard to detect the target path, see figure (6) centre panel. But if $K < 0$, then $\hat{P}_{1,max}(r_{max}/N)$ is to the right of $\hat{P}_T(r_T/N)$, see figure (6), and it is impossible to detect the target path.

To get intuition for K consider its three terms. The first term $D(P_{on}||P_{off})$ is a measure of how effective the local filter cues are for detecting the target. If $P_{on} = P_{off}$ then $D(P_{on}||P_{off}) = 0$ and the local cues are useless. The second term $D(P_{\Delta G}||U)$ is a measure of how much prior knowledge we have about the probable shape of the target (setting $P_{\Delta G} = U$ means we have no prior information). Finally $\log Q$ is a measure of how many distractor paths there are. Therefore K becomes large (and so target detection becomes easy) the better our filter detectors, the more prior knowledge we have, and the fewer the number of distractor paths.

In related work (Yuille and Coughlan 2000a) we used Sanov's theorem (see Appendix A) to show that the expected number of paths in F_1 with rewards

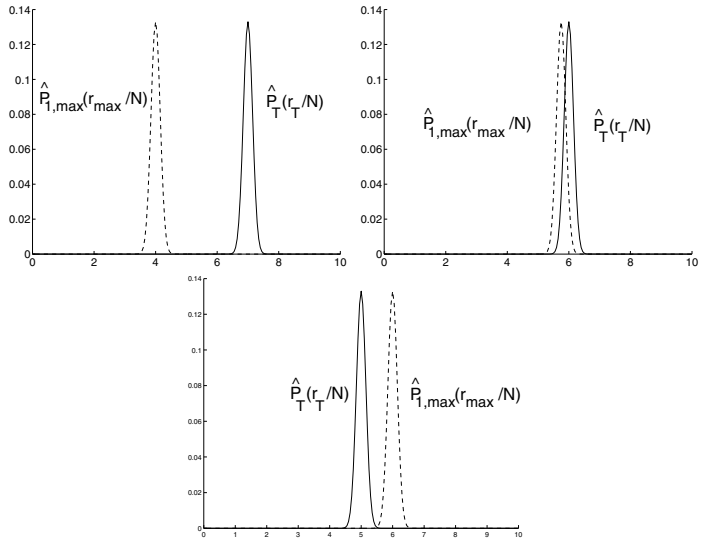

Figure 6: A schematic illustration of the Phase Transition. (Top Left Panel) the reward of the target path is higher than the largest reward of the distractor paths (so detection is straightforward). (Top Right Panel) the tasks becomes difficult because the target reward and the best distractor reward are very similar. (Bottom Panel) detecting the target path becomes impossible because its reward is lower than the best distractor path. The horizontal axis labels the normalized reward.

greater than the target path behaves as 2^{-NK_B}, where the *order parameter* K_B is defined by:

$$K_B = 2B(P_{on}, P_{off}) + 2B(P_{\Delta G}, U) - \log Q, \tag{9}$$

where $B(P,Q) = -\log \sum_{i=1}^{m} (p_i)^{1/2}(q_i)^{1/2}$ is the Bhattacharyya bound between the distributions $P = \{p_i\}$ and $Q = \{q_i\}$. Once again, there is a change in behaviour as K_B changes sign. For $K_B > 0$, we expect there to be no paths in F_1 with rewards greater than the target path.

Our analysis of the A* algorithm will proceed in the regime where $K > 0$ and

$K_B > 0$. There is little purpose in estimating how fast one can compute the MAP estimator unless one is sure that the estimator is detecting a good approximation to the correct target. Our results, see section (3), will require an additional condition to hold, see Theorems 6 and 7, which will ensure that $K_B > 0$. There will, however, be situations where the target is detectable (ie. $K_B > 0$) but where we cannot prove expected linear convergence. More specifically, we can express $K_B = \{\psi_1 - \log Q\} + \psi_2$ where ψ_1, ψ_2 are positive quantities which are defined in Theorem 3. Our complexity proofs apply provided $\psi_1 > \log Q$. If $\psi_1 < \log Q$ but $\psi_1 + \psi_2 > \log Q$ then the target path is detectable but we can say nothing about the complexity of the algorithms.

2.3 A*

The A* graph search algorithm (see Pearl 1984, Winston 1984, Russell and Norvig 1995) is used to find a path of maximum reward between a start node A and a goal node B in a graph, see figure (7). The reward of a particular path is the sum of the rewards of each edge traversed. The A* procedure maintains a tree of partial paths already explored, and computes a measure f of the "promise" of each partial path (i.e. leaf in the search tree). New paths are considered by extending the most promising node one step. The measure f for any node C is defined as $f(C) = g(C) + h(C)$, where $g(C)$ is the best cumulative reward found so far from A to C and $h(C)$ is an overestimate of the remaining reward from C to B. The closer this overestimate is to the true reward then the faster the

algorithm will run. We will refer to the value of f as the *A* reward* in contrast with the reward function of equation (5).

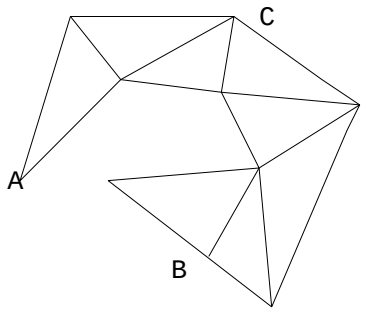

Figure 7: The A* algorithm tries to find the path from A to B with highest reward. For a partial path AC the algorithm stores $g(C)$, the best reward to go from A to C, and an overestimate $h(C)$ of the reward to go from C to B.

It is straightforward to prove that A* is guaranteed to converge to the correct result provided the heuristic $h(.)$ is an upper bound for the true reward from all nodes C to the goal node B. A heuristic satisfying these conditions is called *admissible*. Conversely, a heuristic which does not satisfy them is called *inadmissible*. The word "inadmissible" is a technical term only and *does not* imply that inadmissible heuristics are inferior to admissible ones. In fact, as we show in this paper, algorithms using inadmissible heuristics can converge rapidly to good approximations to the correct result. Conversely, as discussed below, algorithms with admissible heuristics may be slow to converge.

In this paper, we consider inadmissible heuristics. We set the heuristic reward to be $H_L + H_P$ for each unexplored segment (where $_L$ and $_P$ label the likelihood and the geometric prior respectively). Thus a subpath starting at the origin of

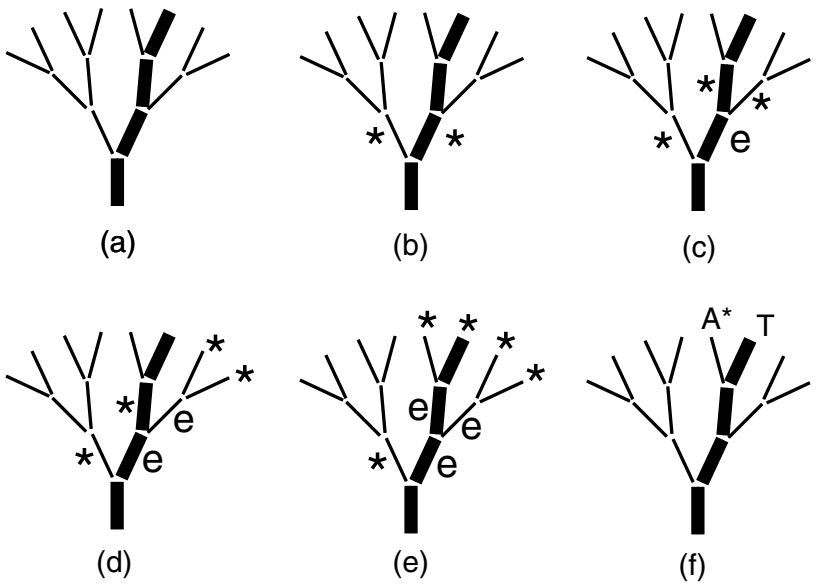

Figure 8: An A* search sequence. Panel (a) shows the target path in bold. Panel (b) shows the two segments added to the queue (marked by asterisks) at the start of the search. Panel (c) explores the right segment, removes it from the queue (label e), and adds its two children to the queue. The search continues in panels (d,e) where explored segments are eliminated from the queue (and labelled e) and their children are added to the queue (labelled with asterisks). Panel (f) A* converges to a solution (A*) that is one segment away from the target path (T).

length M will have heuristic reward of $(N - M)(H_L + H_P)$. We will drop the $N(H_L + H_P)$ term, which is the same for all paths, and simply use $-M(H_L + H_P)$ as the heuristic.

We consider the A* rewards of two partial paths, one of length m segments that overlaps completely with the target path, and the other of length n that does not overlap at all with the target path. The A* rewards of these paths are denoted

by $S_{on}(m)$ and $S_{off}(n)$ and are given by:

$$S_{on}(m) = m\{\vec{\phi}^{on} \cdot \vec{\alpha} - H_L\} + m\{\vec{\psi}^{on} \cdot \vec{\beta} - H_P\},$$
$$S_{off}(n) = n\{\vec{\phi}^{off} \cdot \vec{\alpha} - H_L\} + n\{\vec{\psi}^{off} \cdot \vec{\beta} - H_P\}, \quad (10)$$

where $\phi_\mu^{on} = \frac{1}{m}\sum_i \delta_{y_i,a_\mu}$ and $\psi_\nu^{on} = \frac{1}{m}\sum_i \delta_{t_i,b_\nu}$ for the on path segments (and similarly for the off path segments).

The effect of this heuristic is to encourage us to explore subpaths provided their reward per segment is greater than $H_L + H_P$. Note that the larger the value of $H_L + H_P$ the more the algorithm will favour a breadth-first strategy (Winston 1984) of exploring the tree (because few long subpaths will have rewards per segment exceeding $H_L + H_P$). Breadth-first search is a conservative strategy which will find the best solution but may take a long time to do so. In general, the smaller the heuristic then the faster the search time but the greater the possibility of error. In particular, if $H_L + H_P > \max_{\mu\in\{1,..,J\}} \alpha_\mu + \max_{\nu\in\{1,...,Q\}} \beta_\nu$ then the heuristic is admissible and the search is guaranteed to converge to the path with highest reward (Pearl 1984, Winston 1984, Russell and Norvig 1995). (But an algorithm which uses this heuristic is likely to be slow). We will be considering inadmissible heuristics (i.e. $H_L + H_P < \max_{\mu\in\{1,..,J\}} \alpha_\mu + \max_{\nu\in\{1,...,Q\}} \beta_\nu$) which we expect to be quicker than admissible heuristics but which will have more errors (our theorems quantify these statements).

There is a close connection between heuristics for A* search and the general issue of pruning search algorithms. In previous work (Yuille and Coughlan 1999) we analyzed the results of a search algorithm which pruned out paths whose

rewards fell below a critical value (corresponding to a heuristic). In related work, we explored the use of pruning heuristics for speeding up dynamic programming algorithms for detecting hand outlines in real images (Coughlan, Snow, English and Yuille 1998, Coughlan, Snow, English and Yuille 2000).

We will first prove convergence results for a specific choice of H_L, H_P and later generalize to a larger set of values.

3 Convergence Proof with Bhattacharyya Heuristics

This section will prove convergence results for a specific choice of heuristic which we call the *Bhattacharyya Heuristic*. The next section will generalize the results to a larger class (for which the proofs are more complicated). First we need to say something about the choice of heuristics.

We need to choose a heuristic so that it is smaller than the reward (per unit length) that we would get from the target path and larger than the reward for a distractor path. This means, see equation (10), that the A* rewards will tend to be positive for the target and negative for the distractor paths (so the algorithm will prefer to explore the target). Let us consider the reward H_L only (the analysis is similar for H_P). The expected reward for the target path is $D(P_{on}||P_{off}) = \sum_y P_{on}(y) \log \frac{P_{on}(y)}{P_{off}(y)}$ and for the distractor path it is $-D(P_{off}||P_{on}) = \sum_y P_{off}(y) \log \frac{P_{on}(y)}{P_{off}(y)}$. Therefore we want to select H_L so that

$-D(P_{off}||P_{on}) < H_L < D(P_{on}||P_{off})$.

Our complexity results will be obtained using Sanov's theorem, see Appendix A, to estimate the probability that the algorithms wastes time searching distractor paths. In order to use Sanov's theorem, it is convenient to think of the heuristic as the expected reward of data distributed according to the geometric mixture of P_{on}, P_{off} given by $P_\lambda(y) = P_{on}^{1-\lambda}(y)P_{off}^{\lambda}(y)/Z[\lambda]$ (where $Z[\lambda]$ is a normalization constant. In this section we will consider the special case where $\lambda = 1/2$. This gives the *Bhattacharyya heuristic* $H_L^* = \sum_y P_{\lambda=1/2}(y) \log \frac{P_{on}(y)}{P_{off}(y)}$. (We give it this name because the distribution $P_{\lambda=1/2}$ is associated with the Bhattacharyya bound in statistics (Ripley 1996)). By setting $\lambda = 1/2$ we are essentially choosing a heuristic midway between the target and distractors (generated by P_{on} and P_{off} respectively). In the next section we will extend the results to deal with other values of λ.

The Bhattacharyya heuristic is special in two ways. Firstly, it simplifies the analysis. Secondly, and more importantly, we can prove stronger results about convergence if the Bhattacharyya heuristic is used (although this may reflect limitations in our proofs). As we will discuss in the next section, if the algorithm converges using one of the alternative heuristics then it will also converges with the Bhattacharyya heuristic, but the reverse is not necessarily true.

The Bhattacharyya heuristics H_L^*, H_P^* are the expected rewards per segment:

$$H_L^* = \vec{\phi}_{Bh} \cdot \vec{\alpha}, \quad H_P^* = \vec{\psi}_{Bh} \cdot \vec{\beta}, \tag{11}$$

with respect to the distributions ϕ_{Bh}, ψ_{Bh}:

$$\phi_{Bh}(y) = \frac{\{P_{on}(y)\}^{1/2}\{P_{off}(y)\}^{1/2}}{Z_\phi}, \quad \psi_{Bh}(t) = \frac{\{P_{\Delta G}(t)\}^{1/2}\{U(t)\}^{1/2}}{Z_\psi}, \qquad (12)$$

where Z_ϕ, Z_ψ are normalization constants.

We first put an upper bound on the probability that any completely false segment is searched.

Let $A_{n,i}$ be the set of subpaths of length n that belong to F_i. Then we have the following result:

Theorem 1: *The probability that A* searches the last segment of a particular subpath in $A_{n,i}$ is less than or equal to $Pr\{\exists m \ : \ S_{off}(n) \geq S_{on}(m)\}$.*

Proof. *By definition of A*, a* **necessary** *condition for the segment to be searched is that its A* reward (including the heuristic) is better than the A* reward of at least one segment on the target path. This is because the A* algorithm always maintains a queue of nodes to explore and searches the node segment with highest reward. The algorithm is initialized at the start of the target path and so an element of the target path will always lie in the queue of nodes that A* considers searching. (This condition is not sufficient to ensure that the segment is searched – so we are only obtaining an upper bound).*

We now bound $Pr\{\exists m \ : \ S_{off}(n) \geq S_{on}(m)\}$ by something we can evaluate.

Theorem 2: $Pr\{\exists m : S_{off}(n) \geq S_{on}(m)\} \leq \sum_{m=0}^{\infty} Pr\{S_{off}(n) \geq S_{on}(m)\}$

Proof. *Boole's inequality.*

We now proceed to find a bound on $Pr\{S_{off}(n) \geq S_{on}(m)\}$. This is done using Sanov's theorem, see Appendix A. It will show that this probability falls-off

exponentially with n, m (provided certain parameters are positive).

Theorem 3. *$Pr\{S_{off}(n) \geq S_{on}(m)\} \leq \{(n+1)(m+1)\}^{J^2Q^2} 2^{-(n\Psi_1 + m\Psi_2)}$, where $\Psi_1 = D(\vec{\phi}_{Bh}||P_{off}) + D(\vec{\psi}_{Bh}||U)$ and $\Psi_2 = D(\vec{\phi}_{Bh}||P_{on}) + D(\vec{\psi}_{Bh}||P_{\Delta G})$.*

Proof. *The proof is an application of Sanov's theorem, see Appendix A, applied to the product space of types of $P_{on}, P_{off}, P_{\Delta G}, U$. Define:*

$$E = \{(\vec{\phi}^{off}, \vec{\psi}^{off}, \vec{\phi}^{on}, \vec{\psi}^{on}) : n\{\vec{\phi}^{off} \cdot \vec{\alpha} - H_L^* + \vec{\psi}^{off} \cdot \vec{\beta} - H_p^*\} \geq m\{\vec{\phi}^{on} \cdot \vec{\alpha} - H_L^* + \vec{\psi}^{on} \cdot \vec{\beta} - H_p^*\}\}. \tag{13}$$

(i.e. E is the set of all histograms corresponding to partial off paths with higher A^* reward than the partial on path).

Sanov's theorem gives a bound in terms of the $\phi^{\text{off}}, \psi^{off}, \phi^{on}, \psi^{on}$ that minimize:

$$\begin{aligned} f(\vec{\phi}^{off}, \vec{\psi}^{off}, \vec{\phi}^{on}, \vec{\psi}^{on}) &= nD(\vec{\phi}^{off}||P_{off}) + nD(\vec{\psi}^{off}||U) + mD(\vec{\phi}^{on}||P_{on}) + mD(\vec{\psi}^{on}||P_{\Delta G}) \\ &+\tau_1\{\sum_y \phi^{off}(y) - 1\} + \tau_2\{\sum_t \psi^{off}(t) - 1\} + \tau_3\{\sum_y \phi^{on}(y) - 1\} + \tau_4\{\sum_t \psi^{on}(t) - 1\} \\ &+\gamma\{m\{\vec{\phi}^{on} \cdot \vec{\alpha} - H_L^* + \vec{\psi}^{on} \cdot \vec{\beta} - H_p^*\} - n\{\vec{\phi}^{off} \cdot \vec{\alpha} - H_L^* + \vec{\psi}^{off} \cdot \vec{\beta} - H_p^*\}\}, \end{aligned} \tag{14}$$

where the τ's and γ are Lagrange multipliers. This function $f(.,.,.,.)$ is known to be convex so there is a unique minimum. Observe that $f(....)$ consists of four terms of form $nD(\vec{\phi}^{off}||P_{off}) + \tau_1\{\sum_y \phi^{off}(y) - 1\} - n\gamma\vec{\phi}^{off} \cdot \vec{\alpha}$ which are coupled only by shared constants. These terms can be minimized separately to give:

$$\vec{\phi}^{off*} = \frac{P_{on}^{\gamma} P_{off}^{1-\gamma}}{Z[1-\gamma]}, \quad \vec{\phi}^{on*} = \frac{P_{on}^{1-\gamma} P_{off}^{\gamma}}{Z[\gamma]}, \quad \vec{\psi}^{off*} = \frac{P_{\Delta G}^{\gamma} U^{1-\gamma}}{Z_2[1-\gamma]}, \quad \vec{\psi}^{on*} = \frac{P_{\Delta G}^{1-\gamma} U^{\gamma}}{Z_2[\gamma]}, \tag{15}$$

subject to the constraint given by equation (13).

By inspection, the unique solution occurs when $\gamma = 1/2$. *In this case:*

$$\vec{\phi}^{off*} \cdot \vec{\alpha} = H_L^* = \vec{\phi}^{on*} \cdot \vec{\alpha}, \quad \vec{\psi}^{off*} \cdot \vec{\beta} = H_P^* = \vec{\psi}^{on*} \cdot \vec{\beta}. \tag{16}$$

The solution occurs at $\vec{\phi}^{on*} = \vec{\phi}^{off*} = \vec{\phi}_{Bh}$ *and at* $\vec{\psi}^{on*} = \vec{\psi}^{off*} = \vec{\psi}_{Bh}$. *Substituting into the Sanov bound gives the result.*

From Theorem 3, it is a direct summation, and application of Theorem 2, to obtain:

Theorem 4. $Pr\{\exists m : S_{off}(n) \geq S_{on}(m)\} \leq \sum_{m=0}^{\infty} Pr\{S_{off}(n) \geq S_{on}(m)\} \leq (n+1)^{J^2Q^2} C_2(\Psi_2) 2^{-n\Psi_1}$, *where* Ψ_1, Ψ_2 *are specified in Theorem 3, and*

$$C_2(\Psi_2) = \sum_{m=0}^{\infty} (m+1)^{J^2Q^2} 2^{-m\Psi_2}. \tag{17}$$

Theorem 4 shows that the probability of exploring a particular distractor path to depth n falls off exponentially with n.

We now compute the expected number of false segments that are searched at depth n in the tree. There is a factor $Q^n - 1$ segments at this depth which we can bound above by Q^n.

Theorem 5. *Provided* $\Psi_1 > \log Q$, *the expected number of segments searched in* F_1 *is less than or equal to* $C_2(\Psi_2) C_1(\Psi_1 - \log Q)$, *where:*

$$C_1(\Psi_1 - \log Q) = \sum_{n=1}^{\infty} n(n+1)^{J^2Q^2} 2^{-n(\Psi_1 - \log Q)}. \tag{18}$$

Proof. *There are at most Q^n segments at depth n in F_1. Using Theorem 4, the expected number of segments explored is less than or equal to* $\sum_{n=0}^{\infty} Q^n(n+1)^{J^2Q^2}C_2(\Psi_2)2^{-n\Psi_1}$. *Sum the series. It will not converge unless* $\Psi_1 > \log Q$.

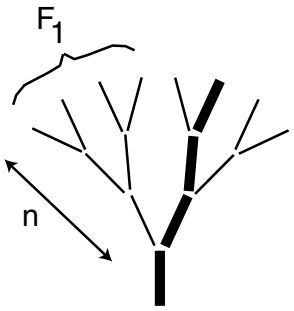

Figure 9: This figure illustrates Theorem 5. F_1 has at most Q^n segments at depth n.

Finally, by the recursive structure of the tree we see that the number of segments explored in the sets $F_2, F_3, ...$ must be less than, or equal to, the number explored in F_1. (We simply eliminate the first few segments, which are in common with the target path, and perform the same argument as above). This yields our main result:

Theorem 6. *The expected number of segments explored by an A* algorithm using the Bhattacharyya heuristic is bounded above by* $NC_2(\Psi_2)C_1(\Psi_1 - \log Q)$, *provided* $\Psi_1 > \log Q$.

The algorithm is expected to explore $O(N)$ segments and the coefficients $C_2(\Psi_2)C_1(\Psi_1 - \log Q)$ rapidly decrease as Ψ_2 and $\Psi_1 - \log Q$ increase.

In addition, we can estimate the expected error of how much our final estimate differs from the target path. Note that *this is not the same as the error with respect*

to the MAP estimate. We count the error as the expected number of incorrect segments on the path.

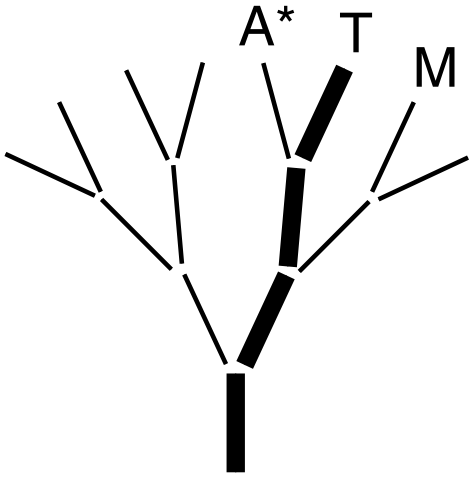

Figure 10: Illustrates Theorem 7. $A*$ marks the path found by the A* algorithm. M marks the MAP estimate of the target path position. T marks the target path. Observe that A* has an error of one segment and the MAP has an error of two.

Theorem 7. *The expected error is bounded above by* $C_2(\Psi_2)C_1(\Psi_1 - \log Q)$, *provided* $\Psi_1 > \log Q$.

Proof. *We measure the error in terms of the expected number of off-road segments. The expected error can then be bounded above by* $\sum_{n=0}^{\infty} Pr(n)n$, *where* $Pr(n)$ *is the probability that* $A*$ *will explore a path in* F_{N+1-n} *to the end. There are* Q^n *such paths. For each path, the probability that we explore it to the end is bounded above by* $\sum_{m=0}^{\infty} Pr\{S_{off}(n) \geq S_{on}(m)\} \leq (n+1)^{J^2Q^2} C_2(\Psi_2)2^{-n\Psi_1}$, *using Theorem 4. Therefore the expected error is bounded above by* $\sum_{n=0}^{\infty} nQ^n(n+1)^{J^2Q^2} C_2(\Psi_2)2^{-n\Psi_1}$, *which was summed in Theorem 5. Hence result.*

The condition $\Psi_1 > \log Q$ is related to the order parameter K_B given by

equation (9). It is straightforward algebra to check that $K_B = \Psi_1 + \Psi_2 - \log Q$. Therefore the condition $\Psi_1 > \log Q$ implies that $K_B > 0$ ($\Psi_2 > 0$ by definition) which ensures that the expected number of paths in F_1 with rewards higher than the target path will fall to zero as $N \mapsto \infty$.

4 Alternative Heuristics

We now generalize our results to other heuristics H_L, H_P which lie in the range $-D(P_{off}||P_{on}) < H_L < D(P_{on}||P_{off})$ and $-D(U||P_{\Delta G}) < H_P < D(P_{\Delta G}||U)$. The basic results and proof strategy are similar to the previous section. There is, however, one stage that is technically more complicated and requires some additional results which are proved in Appendix B.

The convergence results we obtain for these alternative heuristics are weaker than those for the Bhattacharyya heuristic. Each alternative heuristic is associated with a quantity $\hat{\Psi}_1$ and we can only prove convergence provided $\hat{\Psi}_1 > \log Q$. But for the Bhattacharyya heuristic we can prove convergence provided $\Psi_1 > \log Q$ and, as we will show, $\Psi_1 \geq \hat{\Psi}_1$. Hence there will be situations where the A* algorithm will converge if the Bhattacharyya heuristic is used but will not necessarily converge for other heuristics. (This may, of course, reflect the limitations of our proofs and not of the heuristics).

As in the previous section, in order to use Sanov's theorem, it is convenient to associate heuristics H_L, H_P to probability distributions $\phi_{H_L}(y), \psi_{H_P}(t)$ which are of form:

$$\phi_{H_L}(y) = P_{on}^{1-\lambda(H_L)}(y) P_{off}^{\lambda(H_L)}(y)/Z_1[\lambda(H_L)],\ \ \psi_{H_P}(t) = P_{\Delta G}^{1-\mu(H_P)}(t) U^{\mu(H_P)}(t)/Z_2[\mu(H_P)], \tag{19}$$

where $\lambda(H_L), \mu(H_P)$ are determined so that

$$\sum_y \phi_{H_L}(y) \log \frac{P_{on}(y)}{P_{off}(y)} = H_L,\ \sum_y \psi_{H_P}(y) \log \frac{P_{\Delta G}(t)}{U(t)} = H_P. \tag{20}$$

To remove the ambiguity in H_L, H_P (because the A* heuristic depends only on their sum $H_L + H_P$) we require that $\lambda(H_L) = \mu(H_P)$.

It is also convenient to define additional variables $(\hat{H}_L, \hat{H}_P)$ which are related to (H_L, H_P) by the conditions that $\lambda(H_L) + \lambda(\hat{H}_L) = 1$ and $\mu(H_P) + \mu(\hat{H}_P) = 1$. (For the Bhattacharyya heuristics $H_L = \hat{H}_L$ and $H_P = \hat{H}_P$.)

Next, we define two functions $\hat{\Psi}_1(H_L, H_P)$ and $\hat{\Psi}_2(H_L, H_P)$ by the equations:

$$\hat{\Psi}_1 = D(\vec{\phi}_{\hat{H}_L}||P_{off}) + D(\vec{\psi}_{\hat{H}_P}||U),\quad \hat{\Psi}_2 = D(\vec{\phi}_{H_L}||P_{on}) + D(\vec{\psi}_{H_P}||P_{\Delta G})\ \ \text{if} H_L + H_P \geq \hat{H}_L + \hat{H}_P$$

$$\hat{\Psi}_1 = D(\vec{\phi}_{H_L}||P_{off}) + D(\vec{\psi}_{H_P}||U),\quad \hat{\Psi}_2 = D(\vec{\phi}_{\hat{H}_L}||P_{on}) + D(\vec{\psi}_{\hat{H}_P}||P_{\Delta G})\ \ \text{if} H_L + H_P < \hat{H}_L + \hat{H}_P. \tag{21}$$

Our results are summarized by the following theorem which subsumes both Theorems 6 and 7 by replacing Ψ_1, Ψ_2 by $\hat{\Psi}_1(H_L, H_P), \hat{\Psi}_2(H_L, H_P)$. (It can be verified that $\hat{\Psi}_1 = \Psi_1$ and $\hat{\Psi}_2 = \Psi_2$ if we use the Bhattacharyya heuristic).

Theorem 8. *The expected number of segments explored by an A* algorithm using the heuristics H_L, H_P is bounded above by $NC_2(\hat{\Psi}_2(H_L, H_P))C_1(\hat{\Psi}_1(H_L, H_P) - \log Q)$, provided $\hat{\Psi}_1(H_L, H_P) - \log Q > 0$. In addition, the expected error is bounded above by $C_2(\hat{\Psi}_2(H_L, H_P))\ C_1(\hat{\Psi}_1(H_L, H_P) - \log Q)$.*

Proof. *The proof follows the basic strategy of Theorems 1-7. The difference is that the bounds for* $Pr\{S_{off}(n) \geq S_{on}(m)\}$*, given by Theorem 3 for the Bhattacharyya heuristic, now depend in a complicated way on* n *and* m*. It requires additional work, see Appendix B, to bound these terms by expressions which decay exponentially with* n *and* m*. This involves replacing* Ψ_1, Ψ_2 *by* $\hat{\Psi}_1(H_L, H_P), \hat{\Psi}_2(H_L, H_P)$*. After that, the remainder of the results follow directly.*

Finally, we show that $\hat{\Psi}_1 \leq \Psi_1$ for all alternative heuristics (and with equality only if we use the Bhattacharyya heuristic). This follows directly from equation (21). The condition that $H_L + H_P \geq \hat{H}_L + \hat{H}_P$ implies that $\lambda(H_L) \leq 1/2$ and $\mu(H_P) \leq 1/2$. Hence $\hat{\psi}_1 \leq \psi_1$. (A similar argument applies if $H_L + H_P \leq \hat{H}_L + \hat{H}_P$).

5 Sorting the Queue in Linear Expected Time

We have shown that the expected number of nodes searched is linear in N. But the convergence rate of the algorithm will also depend on how much time is required to sort the queue of nodes that we want to expand. In this section, we prove that the expected time to sort the queue nodes is constant.

We use a simple linked list data structure where we order the queue nodes according to their rewards (instead of a more sophisticated data structure, like a heap – see, for example, (Geiger and Liu 1997, Coughlan, Snow, English and Yuille 1998). A* proceeds by expanding the top node (the one with highest A* reward) and must then adjust the queue to accommodate its children. We now

show that the expected sort time, which is required to place the children in their correct positions in the queue, is a constant. To do this, we note that the children nodes have A^* rewards that are smaller than the top node by at most Λ, where $\Lambda = H_L + H_P - \min_y \log P_{on}(y)/P_{off}(y) - \min_t \log P_{\Delta G}(t)/U(t)$ (note that $\Lambda > 0$). We therefore only have to compare the rewards of the children with nodes whose rewards are within Λ of the top node. As we will show, the expected number of these nodes is constant. This gives the following theorem.

Theorem 9 *The expected sorting rate is constant (i.e. independent of the size N of the problem).*

Proof. *The expected sorting rate is equal to Q times the expected number of nodes in the sort queue which have rewards within Λ of the top node. The reward of the top node is guaranteed to be greater than, or equal to, the reward r_T of the longest target subpath in the queue. Let this longest target subpath have length n. To prove that the expected sorting time is constant it suffices to show that the expected number of paths in the queue with rewards greater than $r_T - \Lambda$ is constant. This requires computing the probabilities that subpaths in $F_1, ..., F_n$ have rewards higher than $r_T - \Lambda$ and bounding the expected number of such subpaths. (We do not need to consider paths in F_i, $i > n$ because, by definition of n, they involve children of nodes in the queue and so cannot be in the queue.) We can bound these probabilities using Sanov's theorem and then bound the expected number of nodes by summing exponential series. The details are given in the proof of Theorem A3 in Appendix B.*

6 Conclusion

The goal of this paper is to point out that Bayesian formulation of inference problems leads to a probability distribution on the ensemble of problem instances. Analysis of this ensemble can give complexity results and, in other work (Yuille and Coughlan 2000a, Yuille, Coughlan, Wu and Zhu 2001) algorithm-independent results.

As a specific example, we analyzed the Geman and Jedynak (Geman and Jedynak 1996) theory for road tracking. We were able to demonstrate linear (in the road size) expected convergence for a class of ensembles even though the worst case performance is exponential. This agrees with previous work (Yuille and Coughlan 1999) where we analyzed a block-pruning search strategy motivated by (Pearl 1984).

Our techniques can be applied to a range of other search problems. This can correspond either to modifications of the Geman and Jedynak algorithms (for example, allowing for multiple roads) or, more generally, to tree search problems. (Of course, the theoretical analysis may not result in closed form analytic expressions). Alternatively, one can use these techniques to determine efficient pruning algorithms. For example, the speed of dynamic programming algorithms can be improved by pruning out search hypotheses (Coughlan, Snow, English and Yuille 1998, Coughlan, Snow, English and Yuille 2000).

Acknowledgements

We want to acknowledge funding from NSF with award number IRI-9700446, from the Center for Imaging Sciences funded by ARO DAAH049510494, and from the Smith-Kettlewell core grant, and the AFOSR grant F49620-98-1-0197 to A.L.Y. Lei Xu drew our attention to Pearl's book on heuristics and we thank Abracadabra books for obtaining a second hand copy for us. We would also like to thank Dan Snow and Scott Konishi for helpful discussions as the work was progressing and Davi Geiger for providing useful stimulation. David Forsyth, Jitendra Malik, Preeti Verghese, Dan Kersten, Suzanne McKee and Song Chun Zhu gave very useful feedback and encouragement. Finally, we wish to thank Tom Ngo for drawing our attention to the work of Cheeseman and Selman.

7 Appendix A: Sanov's Theorem

This Appendix introduces results from the theory of types (Cover and Thomas 1991) which we will use to prove our results. We will be particularly concerned with Sanov's theorem. To motivate this material we will apply it to the problem of determining whether a given set of measurements are more likely to come from a road or non-road but *without* making any geometrical assumptions about the likely shape of the road. The theorem assumes that we have an underlying distribution P_s which generates a set of N independent identically distributed (i.i.d.) samples. From each sample set we can determine an empirical histogram, or *type*, see figure (11,12). The law of large numbers states that these empirical

histograms (when normalized) must become similar to the distribution P_s as $N \mapsto \infty$. Sanov's theorem puts bounds on *how fast* the empirical histograms converge (in probability) to the underlying distribution. Thereby it puts bounds on the probability of rare events.

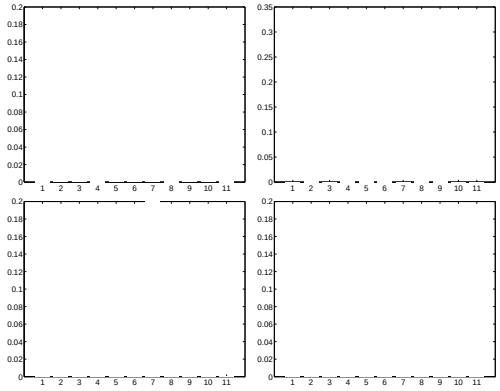

Figure 11: Samples from an underlying distribution. Left to right and top to bottom, the original distribution, followed by histograms, or types, from 10, 100, and 1000 samples from the original. Observe that for small numbers of samples the types tend to differ greatly from the true distribution. But for large N the law of large numbers says that they must converge (with high probability).

Sanov's Theorem. *Let $y_1, y_2, ..., y_N$ be i.i.d. from a distribution $P_s(y)$ with alphabet size J and E be any closed set of probability distributions. Let $Pr(\vec{\phi} \in E)$ be the probability that the type of a sample sequence lies in the set E. Then:*

$$\frac{2^{-ND(\vec{\phi}^*||P_s)}}{(N+1)^J} \leq Pr(\vec{\phi} \in E) \leq (N+1)^J 2^{-ND(\vec{\phi}^*||P_s)}, \tag{22}$$

where $\vec{\phi}^* = \arg\min_{\vec{\phi} \in E} D(\vec{\phi}||P_s)$ is the distribution in E that is closest to P_s in terms of Kullback-Leibler divergence, given by $D(\vec{\phi}||P_s) = \sum_{y=1}^{J} \phi(y) \log(\phi(y)/P_s(y))$.

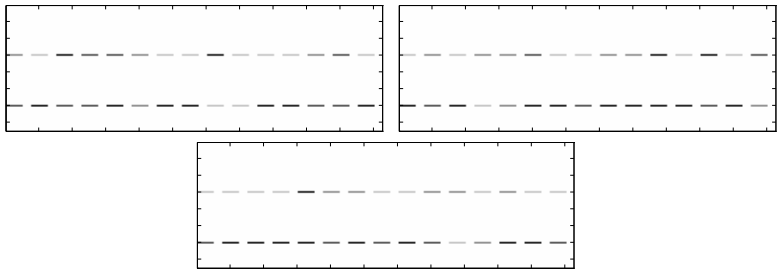

Figure 12: Sequences of edge values rendered using four gray levels ranging from light gray to black. In each pair, one sequence is drawn i.i.d. from $P_{on} = (0.1, 0.1, 0.3, 0.5)$ and the other from $P_{off} = (0.5, 0.3, 0.1, 0.1)$. Although individual edge values are unreliable, taken as a whole it is clear that the top sequences from each panel are from P_{off} and the bottom sequences from P_{on}.

This is illustrated by figure (13). Intuitively, it shows that, when considering the chance of a set of rare events happening, we essentially only have to worry about the "most likely" of the rare events (in the sense of Kullback-Leibler divergence). Most importantly, it tells us that the probability of rare events falls off *exponentially* with the Kullback-Leibler divergence between the rare event (its type) and the true distribution. This exponential fall-off is critical for proving the results in this paper. Note that Sanov' theorem involves an *alphabet factor* $(N+1)^J$. This alphabet factor becomes irrelevant at large N (compared to the exponential term). It does, however, require that the distribution P_s is defined on a finite space, or can be well approximated by a quantized distribution on a finite space.

Sanov's theorem can be illustrated by a simple coin tossing example, see figure (13). Suppose we have a fair coin and want to estimate the probability of

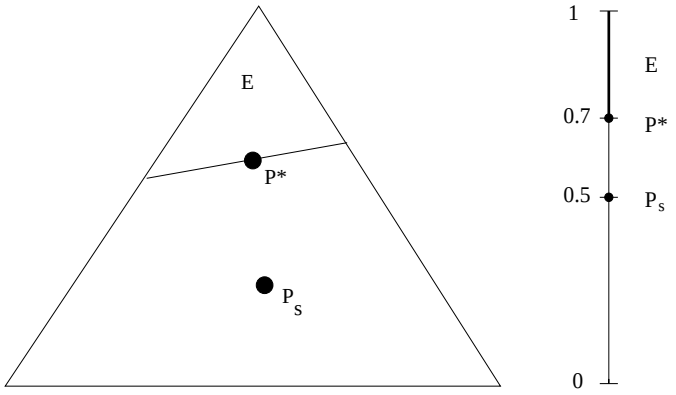

Figure 13: Left, Sanov's theorem. The triangle represents the set of probability distributions. P_s is the distribution which generates the samples. Sanov's theorem states that the probability that a type, or empirical distribution, lies within the subset E is chiefly determined by the distribution P^* 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 $P^*(head) = 0.7$.

observing more than 700 heads in 1000 tosses. Then set E is the set of probability distributions for which $P(head) \geq 0.7$ ($P(head) + P(tails) = 1$). The distribution generating the samples is $P_s(head) = P_s(tails) = 1/2$ because the coin is fair. The distribution in E closest to P_s is $P^*(head) = 0.7, P^*(tails) = 0.3$. We calculate $D(P^*||P_s) = 0.119$. Substituting into Sanov's theorem, setting the alphabet size $J = 2$, we calculate that the probability of more than 700 heads in 1000 tosses is less than $2^{-119} \times (1001)^2 \leq 2^{-99}$.

In this paper, we are only concerned with sets E which involve the rewards of types. This is because the reward function depends on the data only by the type. These sets will therefore be defined by linear constraints on the types – in particular, constraints such as $\vec{\phi} \cdot \vec{\alpha} \geq T$, where $\alpha(y) = \log(P_{on}(y)/P_{off}(y)),\ y = 1, ..., J$. (We define $\vec{\phi} \cdot \vec{\alpha} = \sum_{y=1}^{J} \phi(y)\alpha(y)$). This will enable us to derive results which will not be true for arbitrary sets E. We will often, however, be concerned with the probabilities that the rewards of samples from one distribution are greater than those from a second. It is straightforward to generalize Sanov's theorem to deal with such cases.

Appendix B

This appendix proves the analogous result to Theorem 3 in the case of general heuristics. This is more complicated because direct application of Sanov's theorem gives bounds for $Pr\{S_{off}(n) \geq S_{on}(m)\}$ which are complicated functions of n and m. To obtain bounds which fall off exponentially with n, m we must first prove a

convexity result concerning how the fall-off factors, such as $D(\vec{\phi}_T||P_{on})$, vary with the threshold T. This result, combined with Sanov's theorem, will give us the bounds we need.

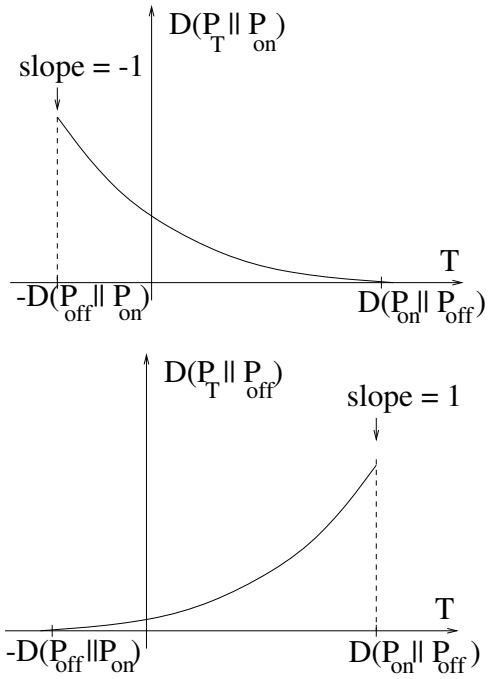

Figure 14: Top, $D(\vec{\phi}_T||P_{on})$ is a convex function of T with its minimum at $T = D(P_{on}||P_{off})$. Bottom, similarly $D(\vec{\phi}_T||P_{off})$ is convex with minimum at $T = -D(P_{off}||P_{on})$.

First we prove a convexity result, Theorem A.1., which will be used to obtain our main results. Theorem A.2. gives the details for section (4). Theorem A.3. gives the details for section (5).

Theorem A.1. *Let* $\vec{\phi}_T(y) = P_{on}^{1-\lambda(T)}(y)P_{off}^{\lambda(T)}(y)/Z(T)$ *where* $\lambda(T)$ *is determined by* $\sum_y \phi_T(y) \log \frac{P_{on}(y)}{P_{off}(y)} = T$, *then* $D(\vec{\phi}_T||P_{on})$ *and* $D(\vec{\phi}_T||P_{off})$ *are convex functions of* T *which attain minima of zero at* $T = D(P_{on}||P_{off})$ *and* $T =$

$-D(P_{off}||P_{on})$ respectively, see figure (14).

Proof. *First, observe that:*

$$D(\vec{\phi}_T||P_{on}) = D(\vec{\phi}_T||P_{off}) - T \tag{23}$$

because of the identity $\sum_y \vec{\phi}_T(y) \log\{\frac{\vec{\phi}_T(y)}{P_{on}(y)}\} = \sum_y \vec{\phi}_T(y) \log\{\frac{\vec{\phi}_T(y)}{P_{off}(y)} \frac{P_{off}(y)}{P_{on}(y)}\}$.

We calculate:

$$\frac{d}{dT} D(\vec{\phi}_T||P_{on}) = \sum_y \frac{d\vec{\phi}_T(y)}{dT} \log \frac{\vec{\phi}_T(y)}{P_{on}(y)} + \sum_y \frac{d\vec{\phi}_T(y)}{dT} = \sum_y \frac{d\vec{\phi}_T(y)}{dT} \log \frac{\vec{\phi}_T(y)}{P_{on}(y)}, \tag{24}$$

using $\sum_y \frac{d\vec{\phi}_T(y)}{dT} = \frac{d}{dT} \sum_y \vec{\phi}_T(y) = 0$. *Substituting* $\vec{\phi}_T(y) = P_{on}^{1-\lambda(T)}(y) P_{off}^{\lambda(T)}(y)/Z(T)$ *we re-express this as:*

$$\frac{d}{dT} D(\vec{\phi}_T||P_{on}) = \sum_y \frac{d\vec{\phi}_T(y)}{dT} \log \frac{P_{off}^{\lambda(T)}(y)}{P_{on}^{\lambda(T)}(y) Z(T)} = -\lambda(T) \sum_y \frac{d\vec{\phi}_T(y)}{dT} \log \frac{P_{on}(y)}{P_{off}(y)}. \tag{25}$$

We recall that $\sum_y \vec{\phi}_T(y) \log \frac{P_{on}(y)}{P_{off}(y)} = T$ *and hence* $\sum_y \frac{d\vec{\phi}_T(y)}{dT} \log \frac{P_{on}(y)}{P_{off}(y)} = 1$ *which implies:*

$$\frac{d}{dT} D(\vec{\phi}_T||P_{on}) = -\lambda(T), \quad \frac{d}{dT} D(\vec{\phi}_T||P_{off}) = 1 - \lambda(T), \tag{26}$$

The second derivatives are given by:

$$\frac{d^2}{dT^2} D(\vec{\phi}_T||P_{on}) = -\frac{d\lambda(T)}{dT}, \quad \frac{d^2}{dT^2} D(\vec{\phi}_T||P_{off}) = -\frac{d\lambda(T)}{dT}. \tag{27}$$

Now $\frac{d\lambda(T)}{dT} < 0$, *because as the threshold* T *decreases then* $\vec{\phi}_T$ *becomes closer to* P_{off} *and hence* $\lambda(T)$ *decreases. (It can be checked that* $\frac{d\lambda(T)}{dT}$ *is minus the inverse*

of the variance of $\log \frac{P_{on}}{P_{off}}$ *with respect to* $\vec{\phi}_T$*). Hence, by equation (27) we see that both* $D(\vec{\phi}_T||P_{on})$ *and* $D(\vec{\phi}_T||P_{off})$ *are convex functions of* T*.*

The minima of the functions occur at $\lambda(T) = 0$ *and* $\lambda(T) = 1$*, corresponding to* $\phi = P_{on}$ *and* $\phi = P_{off}$ *respectively. Hence the minima occur at* $T = D(P_{on}||P_{off})$ *and* $T = -D(P_{off}||P_{on})$ *respectively.*

We can use this result in combination with Sanov's theorem to put bounds on $Pr\{S_{off}(n) \geq S_{on}(m)\}$ which fall off exponentially with n, m.

Theorem A.2.

$$Pr\{S_{off}(n) \geq S_{on}(m)\} \leq \{(n+1)(m+1)\}^{J^2Q^2} 2^{-g(m;n)}, \tag{28}$$

where:

$$g(m;n) \geq n\{D(\phi_{\hat{H}_L}||P_{off}) + D(\psi_{\hat{H}_P}||U)\} + m\{D(\phi_{H_L}||P_{on}) + D(\psi_{H_P}||P_{\Delta G})\},\ if\ H_L > \hat{H}_L,$$

$$g(m;n) \geq n\{D(\phi_{H_L}||P_{off}) + D(\psi_{H_P}||U)\} + m\{D(\phi_{\hat{H}_L}||P_{on}) + D(\psi_{\hat{H}_P}||P_{\Delta G})\},\ if\ H_L < \hat{H}_L. \tag{29}$$

Proof. *We start by following the proof of Theorem 3 but with the definition of set* E *changed to allow for different heuristics* H_L, H_P*. We minimize* $f(.,.,.,.)$ *and obtain expressions for* $\phi_{T_1}, \psi_{W_1}, \phi_{T_2}, \psi_{W_2}$ *except that the minimization no longer occurs at* $\gamma = 1/2$*. The fall-off rate is determined by:*

$$g(m;n) = n\{D(\phi_{T_1}||P_{off}) + D(\psi_{W_1}||U)\} + m\{D(\phi_{T_2}||P_{on}) + D(\psi_{W_2}||P_{\Delta G})\}, \tag{30}$$

where

$$m\{T_2 + W_2 - H_L - H_P\} = n\{T_1 + W_1 - H_L - H_P\}, \tag{31}$$

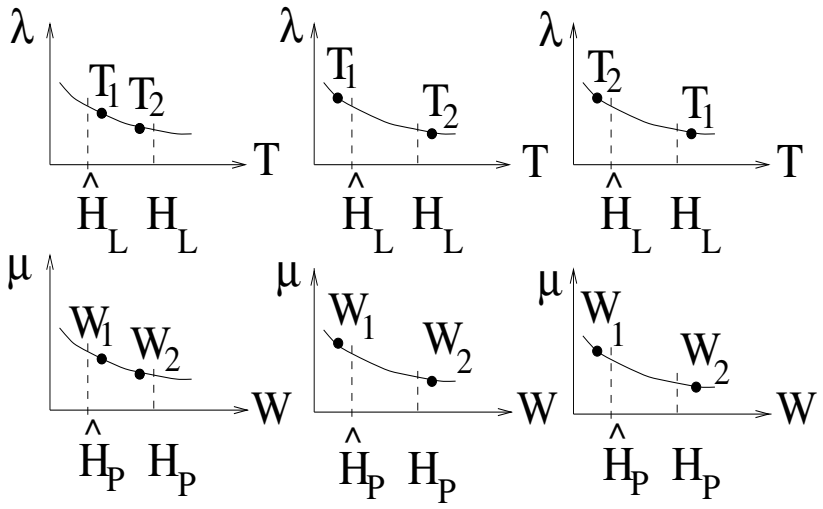

Figure 15: This figure illustrates the three cases of the argument in Theorem 7 (one in each column). The variables T_1, T_2, W_1, W_2 are functions of m, n and their values determine $g(m;n)$. If we use the Bhattacharyya heuristic then $T_1 = T_2 = H_L^*$ and $W_1 = W_2 = H_P^*$ for all values of m, n. They can therefore be thought of as "effective heuristics" and it is necessary to understand their "dynamics" as m, n vary. In Theorem A.2. we show that they are restricted to lie in the ranges illustrated by the left-most column (the configurations in the centre and rightmost column are inconsistent with the three constraints (31,32) which enables us to put bounds on $g(m;n)$).

and

$$\lambda(T_1) + \lambda(T_2) = 1 = \mu(W_1) + \mu(W_2),$$

$$\lambda(T_1) = \mu(W_1), \quad \lambda(T_2) = \mu(W_2). \tag{32}$$

Recall, that to remove the ambiguity in H_L and H_P we imposed the constraint that $\lambda(H_L) = \mu(H_P)$. We also defined $\hat{H}_L, \hat{H}_P$ by the conditions $\lambda(\hat{H}_L) + \lambda(H_L) = 1$ and $\mu(\hat{H}_P) + \mu(H_P) = 1$ which implies that $\lambda(\hat{H}_L) = \mu(\hat{H}_P)$.

There are two situations to consider: (i) $H_L \geq \hat{H}_L$, which implies $H_P \geq \hat{H}_P$ (this follows from the equations at the end of the previous paragraph plus the fact that $\lambda(.)$ and $\mu(.)$ are monotonically decreasing functions), and (ii) $H_L \leq \hat{H}_L$, which implies $H_P \leq \hat{H}_P$.

We claim, in case (i), that $T_1, T_2 \in [\hat{H}_L, H_L]$ and $W_1, W_2 \in [\hat{H}_P, H_P]$, see figure (15). Moreover,

$$g(m;n) \geq n\{D(\phi_{\hat{H}_L}||P_{off})+D(\psi_{\hat{H}_P}||U)\}+m\{D(\phi_{H_L}||P_{on})+D(\psi_{H_P}||P_{\Delta G})\}. \quad (33)$$

Moreover, in situation (ii) we claim that $T_1, T_2 \in [H_L, \hat{H}_L]$ and $W_1, W_2 \in [H_P, \hat{H}_P]$, and

$$g(m;n) \geq n\{D(\phi_{H_L}||P_{off})+D(\psi_{H_P}||U)\}+m\{D(\phi_{\hat{H}_L}||P_{on})+D(\psi_{\hat{H}_P}||P_{\Delta G})\}. \quad (34)$$

We prove the results only for situation (i) because the proofs for situation (ii) are exactly analogous. The condition $\lambda(T_1) + \lambda(T_2) = 1$ implies that there are only three possible cases: either both $T_1, T_2 \in [\hat{H}_L, H_L]$ or, using the monotonicity of $\lambda(T)$, that $T_1 > H_L$ and $T_2 < \hat{H}_L$, or $T_1 < \hat{H}_L$ and $T_2 > H_L$. The first case will ensure that $W_1, W_2 \in [\hat{H}_P, H_P]$ which solves the problem. The second requires that $W_1 > H_P$ and $W_2 < \hat{H}_P$ but this is inconsistent with the requirement that $m\{T_2 + W_2 - H_L - H_P\} = n\{T_1 + W_1 - H_L - H_P\}$ (because the left hand side is negative and the right hand side is positive). Similarly, the third case implies that $W_1 < \hat{H}_P$ and $W_2 > H_P$ which again contradicts the equality. Thus the only possible situation is the first case.

Moreover, as $n \mapsto \infty$, we have $T_1 \mapsto H_L, W_1 \mapsto H_P, T_2 \mapsto \hat{H}_L, W_2 \mapsto \hat{H}_P$.

(This is because $T_1 \leq H_L$ *and* $W_1 \leq H_P$ *so as* $n \mapsto \infty$ *we have* $T_1+W_1-H_L-H_P \mapsto 0$*).*

Theorem A.3. *Let this longest target subpath in a sort queue have length* n *and reward* r_T*. Then the expected number of paths in the queue with rewards greater than* $r_T - \Lambda$ *is constant.*

Proof. *Let the heuristics be* H_L, H_P *and consider the case when* $H_L > \hat{H}_L$ *and* $H_P > \hat{H}_P$ *(the alternative case can be solved by adapting the following argument). Suppose the longest true partial path in the queue is of length* n *and has reward* r_T*. We must consider the probabilities that paths in* $F_1, ..., F_n$ *have rewards higher than* $r_T - \Lambda$*. Applying the onion argument, for each* $m \leq n$*, we must bound the probability that any off-path of any length has reward higher than the true reward for* n *segments minus* Λ*. Following the standard application of Sanov's theorem, we define the set:*

$$E = \{(\vec{\phi}^{off}, \vec{\psi}^{off}, \vec{\phi}^{on}, \vec{\psi}^{on}) : \ m(\vec{\phi}^{off}\cdot\vec{\alpha}+\vec{\psi}^{off}\cdot\vec{\beta}-H_L-H_P)+\Lambda \geq n(\vec{\phi}^{on}\cdot\vec{\alpha}+\vec{\psi}^{on}\cdot\vec{\beta}-H_L-H_P)\}. \tag{35}$$

Following the proof of Theorem A.2. this gives thresholds: T_1, T_2, W_1, W_2 *(as before, these thresholds are functions of* n *and* m*), where:*

$$m(T_1 + W_1 - H_L - H_P) + \Lambda = n(T_2 + W_2 - H_L - H_P), \tag{36}$$

$$\lambda(T_1) + \lambda(T_2) = 1, \ \mu(W_1) + \mu(W_2) = 1, \ \lambda(T_1) = \mu(W_1), \ \lambda(T_2) = \mu(W_2). \tag{37}$$

The fall-off depends on

$$g(n:m) = m\{D(\vec{\phi}_{T_1}||P_{off})+D(\vec{\psi}_{W_1}||U)\}+n\{D(\vec{\phi}_{T_2}||P_{on})+D(\vec{\psi}_{W_2}||P_{\Delta G})\}. \tag{38}$$

Now, again following Theorem A.2., we would like to put lower bounds on $g(n:m)$. *For Theorem A.2. we were able to prove that* $T_1, T_2 \in [\hat{H}_L, H_L]$ *and* $W_1, W_2 \in [\hat{H}_P, H_P]$ *(for the situation where* $H_L > \hat{H}_L$ *and analogous results hold for the situation with* $H_L < \hat{H}_L$*). The* Λ *term prevents these results from being true. However, for large enough* n *or* m *the* Λ *term becomes negligible and we will prove that* $T_1, T_2 \in [\hat{H}_L - \epsilon, H_L + \epsilon]$ *and* $W_1, W_2 \in [\hat{H}_P - \epsilon, H_P + \epsilon]$. *These contain the most important terms and, as we will show, make only a constant contribution to the expected sorting cost. The contributions for small* m *and* n *are, of course, also constant.*

We first show, that for any fixed n, *the thresholds* T_1, W_1 *increase monotonically with* m *and tend to* H_L, H_P *as* $m \mapsto \infty$ *and, similarly,* T_2, W_2 *decrease monotonically with* m *and tend to* $\hat{H}_L, \hat{H}_P$. *From* $\lambda(T_1) = \mu(W_1)$, *see equation (37), and the monotonicity of the functions* $\lambda(.)$ *and* $\mu(.)$, *we see that the coupling between* T_1 *and* W_1 *means that they have to increase, or decrease, together. Similarly,* T_2 *and* W_2 *must either decrease, or increase, together. By equation (36), we see that at* $m = 0$ *we have* $T_2(0) + W_2(0) = H_L + H_P + \Lambda/n$ *which implies, by equation (37), that* $T_1(0) + W_1(0) < \hat{H}_L + \hat{H}_P$. *Equation (36) enforces that* $T_1 \mapsto H_L$, $W_1 \mapsto H_P$ *as* $m \mapsto \infty$ *which implies that* $T_2 \mapsto \hat{H}_L$ *and* $W_2 \mapsto \hat{H}_P$. *Therefore, we see that* $T_1 + W_1$ *increases overall from* $m = 0$ *as* $m \mapsto \infty$ *and conversely* $T_2 + W_2$ *decreases. But are these changes monotonic? From equation (36), we see that provided* $T_1 + W_1 < H_L + H_P$ *then it is inconsistent for* T_1 *and* W_1 *to decrease and* T_2 *and* W_2 *to increase. However, it is impossible for* $T_1 + W_1 > H_L + H_P$ *because, by equation (36), this would imply that* $T_2 + W_2 > H_L + H_P$ *(recall that*

$\Lambda > 0$) *which is inconsistent with equations (37). So we conclude that the only possibility is for* T_1 *and* W_1 *to increase monotonically and* T_2 *and* W_2 *to decrease monotonically.*

Now select a number N_0*, chosen so that* $N_0(\epsilon) \geq \Lambda/\epsilon$*, and let* $n \geq N_0$*. Then for* $m = 0$*, we see that* $T_2 < H_L + \epsilon$ *and* $W_2 < H_P + \epsilon$ *(this follows from equations (36,37)). Moreover,* $T_1 > \hat{H}_L - \hat{\epsilon}$ *and* $W_1 > \hat{H}_P - \hat{\epsilon}$ *(where* $\hat{\epsilon}$ *is defined by equation (37)). As* m *increases* T_1, W_1 *increase monotonically to* H_L, H_P *and* T_2, W_2 *decrease monotonically to* $\hat{H}_L, \hat{H}_P$*. Therefore we have:*

$$g(m:n) \geq m\{D(\vec{\phi}_{\hat{H}_L-\hat{\epsilon}}||P_{off})+D(\vec{\psi}_{\hat{H}_P-\hat{\epsilon}}||U)\}+n\{D(\vec{\phi}_{H_L+\epsilon}||P_{on})+D(\vec{\phi}_{H_P+\epsilon}||P_{\Delta G})\} \;\; \forall\, n > N_0(\epsilon), \tag{39}$$

which ensures that the fall-off factors are bounded below for large n*.*

We now deal with the case of small n *(i.e.* $n < N_0(\epsilon)$*) and large* m*. We claim that there is a specific value* M_0 *such that for* $m > M_0$ *we have* $T_1, T_2 \in [\hat{H}_L, H_L]$ *and* $W_1, W_2 \in [\hat{H}_P, H_P]$*, in which case we can use the same bounds for* $g(m;n)$ *as above (see equation (39)). This claim is proven by setting* $M_0 = \Lambda/(H_L + H_P - \hat{H}_L - \hat{H}_P)$ *and substituting into equation (36) to obtain* $\Lambda(T_1 + W_1 - \hat{H}_L - \hat{H}_P) = n(T_2 + W_2 - H_L - H_P)(H_L + H_P - \hat{H}_L - \hat{H}_P)$*. The consistency conditions, imposed by equation (37), mean that this equation's only solution is* $T_1 = \hat{H}_L$*,* $W_1 = \hat{H}_P$*,* $T_2 = H_L$*, and* $W_2 = H_P$ *(all other possibilities can be shown to be inconsistent using equation (37)). The monotonicity increase of* T_1, W_1*, and decrease of* T_2, W_2*, ensure that, for* $m > M_0$*, the* $T_1, T_2 \in [\hat{H}_L, H_L]$ *and* $W_1, W_2 \in [\hat{H}_P, H_P]$*.*[1]

[1]Observe that M becomes infinite if we use the Bhattacharyya heuristic (i.e. when $H_L = \hat{H}_L$

The final situation is when $n < N_0(\epsilon)$ and $m < M_0$. This is a finite case so we do not need to obtain bounds. We can simply exhaustively count the number of segments.

We now put all these results together. Let $\hat{n}$ be the length of the true partial path segment in the queue (by the nature of A there can only be one such true path segment in the queue at any time). The expected number of queue members with rewards higher than the true segment minus Λ is obtained by summing over the possible segments in $F_1, F_2, ..., F_{\hat{n}}$. We can deal with the cases $m < M_0$ and $n < N_0(\epsilon)$ by exhaustive counting which yields a finite number. For each $n \leq \hat{n}$ we can use the bounds given by equation (39) and apply the arguments from Theorem 4 to sum over m for fixed n obtaining a term which decays exponentially with n. Finally, we can apply the arguments from Theorem 5 to sum over n. The exponential decay factor means that this sum will converge for any value of $\hat{n}$ (even as $\hat{n} \mapsto \infty$). Hence we get constant expected sorting costs.*

References

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

and $H_P = \hat{H}_P$). This is because the regions $[\hat{H}_L, H_L]$ and $[\hat{H}_P, H_P]$ shrink to points H_L^* and H_P^* and the T's and W's only reach them asymptotically. This requires a modification of the proof to obtain bounds on M for which $\max\{|T_1 - H_L^*|, |W_1 - H_P^*|, |T_2 - H_L^*|, |W_2 - H_L^*|\} < \epsilon$.

- Coughlan, J.M. and Yuille, A.L. 1999. "Bayesian A* tree search with expected O(N) convergence rates for road tracking". In *Proceedings EMM-CVPR'99*. Springer-Verlag Lecture Notes in Computer Science 1654.

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

- Coughlan, J.M., Snow, D., English, C. and Yuille, A.L. 2000. "Efficient Deformable Template Detection and Localization without User Initialization". *Computer Vision and Image Understanding*. **78**, pp 303-319.

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

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

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

- Geman, D. and Jedynak, B. 1996. "An active testing model for tracking

roads in satellite images". *IEEE Trans. Patt. Anal. and Machine Intel.* Vol. 18. No. 1, pp 1-14. January.

- Grimmett, G.R. and Stirzaker, D.R. 1992. **Probability and Random Processes**. Clarendon Press. Oxford.

- Karp, R.M. and Pearl, J. 1983. "Searching for an Optimal Path in a Tree with Random Costs". *Artificial Intelligence*. 21. (1,2), pp 99-116.

- 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." *Proc. Int'l conf. on Computer Vision and Pattern Recognition.*

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

- Ripley, B.D. 1996. **Pattern Recognition and Neural Networks**. Cambridge University Press.

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

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

- Vapnik, V.N. 1998. **Statistical Learning Theory**. John Wiley and sons. New York.

- Winston, P.H. 1984. **Artificial Intelligence**. Addison-Wesley Publishing Company. Reading, Massachusetts.

- Yuille, A.L. and Coughlan, J.M. 1999. "Convergence Rates of Algorithms for Visual Search: Detecting Visual Contours". In **Advances in Neural Information Processing Systems 11**. Eds. Kearns, M.S., Solla, S.A., and Cohn, D.A. pp 641-647. 1999.

- Yuille, A.L. and Coughlan, J.M. 2000a. "Fundamental Limits of Bayesian Inference: Order Parameters and Phase Transitions for Road Tracking". *Transactions on Pattern Analysis and Machine Intelligence.* PAMI. Vol. 22, pp 1-14.

- Yuille, A.L. and Coughlan, J.M. 2000b. "An A* perspective on deterministic optimization for deformable templates". *Pattern Recognition.* Vol. 33 (4), pp 603-616. April.

- Yuille, A.L., Coughlan, J.M., Wu, Y-N. and Zhu, S.C. 2001. "Order Parameters for Minimax Entropy Distributions: When does high level knowledge help?" *International Journal of Computer Vision.* 41(1/2), pp 9-33.