Fusion Through Interpretation

Mark J. L. Orr[1]

Advanced Robotics Research Ltd.
University Road
Salford M5 4PP
England
(tel.: +44 61 745 7384 e-mail: mjo@arrc.salf.ac.uk)

John Hallam and Robert B. Fisher

Department of Artificial Intelligence
Edinburgh University
Edinburgh EH1 2QL
Scotland
(tel.: +44 31 650 3097/3098 e-mail: john/rbf@aifh.ed.ac.uk)

Abstract

We investigate the use of interpretation trees to solve the correspondence problem for a mobile robot fusing data from a range image into a world model consisting of planar surface patches. Uncertainty is handled by stochastic techniques where errors are represented by normal joint probability distributions. We show that for problems of a typical size the search time is too long unless the world model can be structured into parts only one of which can be occupied by the robot at any given moment.

1 Introduction

This paper is concerned with the correspondence problem within the context of data fusion for a mobile robot. We restrict our attention to the case of a single range imaging sensor delivering planar surface patch features which are to be fused into a world model which also consists of planar patches. A separate problem, which we do not cover here, is how to update the world model once the correspondences have been established, a problem which is not easy because of partial occlusion.

A similar correspondence problem arises in the computer recognition of objects where the concern is to match image features with features from object

[1] Currently on secondment from SD-Scicon UK Ltd.

models. One of the popular techniques used, originating with Grimson and Lozano-Perez [8], involves searching the correspondence space by traversing an interpretation tree. In practice an exhaustive search is intractable so researchers have devised various constraint based methods to prune the tree [7]. The search can be terminated after a small number of feature correspondences have been accumulated, enough to constrain the transform between the model and image frames, and a test performed to see which transformed model features have correspondences in the image. Such an algorithm belongs to the *hypothesise-and-test* paradigm [6, 4] and is adopted in this paper. In the last few years, stochastic methods, which we also employ, have become standard for handling sensor uncertainty in vision and robotics [1, 5, 10].

The aim of the paper is to investigate whether the hypothesis-and-test method using interpretation trees and stochastic techniques is fast enough for fusing range images. Our own sensor can acquire and process range images in about two seconds so we want to fuse them in this time or less. Typically, an image will contain about ten surface patches while the world model will contain thousands. We do not assume any prior knowledge of the position of the sensor relative to the world model coordinate frame, rather this is something we want to find out.

Each surface patch in the image or model is described by the infinite plane in which it is embedded, an outer boundary, the boundaries of any holes, a total surface area and a central point. The outer boundary consists of a sequence of labeled straight line segments, the labels indicating boundary type (concave join, convex join and occluding or occluded edge) and any adjacency relations with other surfaces. The true 4-dimensional parameter vector of the infinite embedding plane, $\mathbf{p}$, is a concatenation of the surface's outward normal vector and ± 1 times its perpendicular distance from the coordinate origin such that

$$[\mathbf{x}^T \ 1]\mathbf{p} \ = \ 0 \tag{1}$$

for any point $\mathbf{x}$ in the plane. In practice what is specified is an estimate, $\hat{\mathbf{p}}$, of $\mathbf{p}$ and a 4-by-4 covariance matrix, P, expressing the uncertainty associated with the estimate. Similarly, the other numerical parameters (area, central point, and so on) are specified by estimates and uncertainties.

The next section describes interpretation trees. Stochastic techniques are used to provide constraint inequalities and transform estimates. The idea of switching from an exponential to a quadratic search when sufficient correspondences have been accumulated [3] is also explored. Section 3 then analyses the computation costs and predicts average running times for solving interpretation tree problems of a given size. A final section presents our conclusions and ideas for further work.

2 Interpretation Trees

The interpretation tree [8, 7] is a mechanism for exploring different combinations of image features from a set $\{d_i\}$, $1 \leq i \leq D$, with model features from a set $\{m_j\}$, $1 \leq j \leq M$. Usually, we cannot assume that each d_i has a correspondence in the set $\{m_j\}$ and an extra model feature, the often called wild card, must be added, increasing the number of models to $M+1$. In this case the number of nodes in the tree is

$$\sum_{i=0}^{D}(M+1)^i \;\approx\; (M+1)^D .$$

It is immediately obvious that searching the entire tree will take too long which is why constraints are used to find inconsistent combinations and cause the search to backtrack.

2.1 Constraints

If some node in the tree is found to contain an inconsistency, all nodes below it must contain the same inconsistency and so it is pointless to search for a solution amongst them. We examine below unary and binary constraints which must be satisfied by consistent pairings.

2.1.1 Unary Constraints

Unary constraints are used to test whether a single pairing is plausible on the basis of directly comparable attributes of the data and model features. In our case, since the features are surface patches, the most obvious attribute to use is area. If $(\hat{a}_i, A_i)$ and $(\hat{b}_j, B_j)$ are the data and model surface area estimates and variances then

$$d \;=\; \frac{(\hat{a}_i - \hat{b}_j)^2}{A_i + B_j}$$

is the *Mahalanobis distance* separating them which has a χ^2 distribution. The greater this distance, the less likely it is that the two measurements refer to the same feature. In fact it is possible to choose a threshold on the value of d (from a χ^2 table) such that if it is exceeded then the probability that random noise alone has made $\hat{a}_i$ and $\hat{b}_j$ different is less than a certain amount. For example, if $d > 3.84$ then there is only a 5% chance that the difference is due to noise and if we decide, on that basis, that the two measurements are not of the same surface then we have a 5% chance of being wrong.

The problem with using area estimates is that they are often inaccurate due to partial occlusion. The accuracy with which the area of a partially occluded

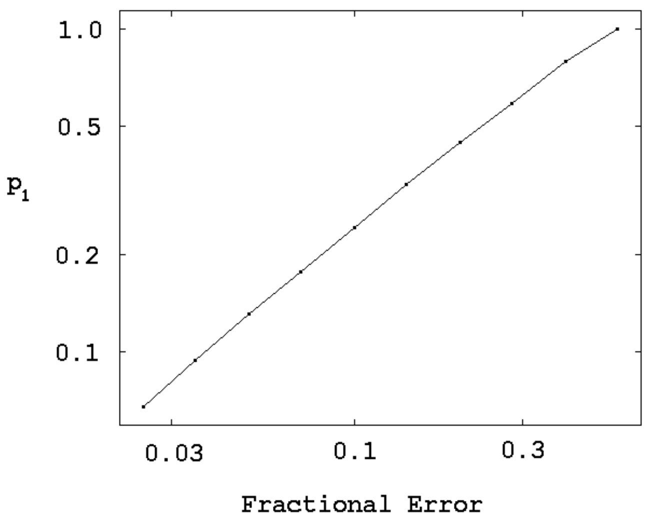

Figure 1: The probability of two unrelated surfaces passing the unary constraint on area as a function of the fractional error in area.

surface can be estimated depends on how much area is visible, so we associate a fraction κ of the estimated visible area $\hat{a}$ with its standard deviation. Figure 1 shows the empirically estimated probability, p_1, of two unrelated surfaces passing the unary constraint as a function of κ, the fractional error in area (here, and throughout the paper, we use a 5% threshold for χ^2 tests).

2.1.2 Binary Constraints

Before adding a new pairing, $(d_{i+1}, m_{j_{i+1}})$, to an existing partial interpretation at level i,

$$I_i = \{(d_1, m_{j_1}), (d_2, m_{j_2}), ...(d_i, m_{j_i})\},$$

the new pairing can be checked against each old pairing to ensure that together they satisfy any available constraints on pairs of pairings. These *binary constraints* often involve notions of invariance: something is the same in both the image and the model. They can be expressed as equations for the case in which the feature's parameters are known precisely or as inequalities when, as always in practice, there is uncertainty. It is relatively straightforward to generate constraint inequalities from constraint equations using stochastic techniques [1]. Suppose

$$\mathbf{g}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{u}_1, \mathbf{u}_2) \;=\; \mathbf{0}$$

is the constraint equation relating the true image parameter vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ to the true model parameter vectors $\mathbf{u}_1$ and $\mathbf{u}_2$. When we only know estimates $\hat{\mathbf{v}}_i$ and $\hat{\mathbf{u}}_i$ and if $\mathbf{g}$ is a linear function or the noise is small then[2]

$$G \;=\; \sum_{i=1}^{2} \left[\frac{\partial \mathbf{g}}{\partial \mathbf{v}_i} V_i \left(\frac{\partial \mathbf{g}}{\partial \mathbf{v}_i} \right)^T + \frac{\partial \mathbf{g}}{\partial \mathbf{u}_i} U_i \left(\frac{\partial \mathbf{g}}{\partial \mathbf{u}_i} \right)^T \right] \tag{2}$$

is the covariance of $\mathbf{g}$. V_i and U_i are the covariance matrices associated with the estimates $\hat{\mathbf{v}}_i$ and $\hat{\mathbf{u}}_i$. The Mahalanobis distance is

$$d \;=\; \mathbf{g}^T G^{-1} \mathbf{g} \tag{3}$$

($\mathbf{g}$ and its derivatives being evaluated at the estimates) and has, to first order, a χ^2 distribution. Thus the constraint is satified if

$$d \;<\; \epsilon \tag{4}$$

where ϵ is taken from a standard tabulation of χ^2 thresholds for various numbers of degrees of freedom (the dimension of $\mathbf{g}$) and various confidence limits.

For dealing with surface patches we use the constraints that the angle between surface normals and the distance between central points is invariant. In the first case we can identify $\mathbf{v}_i$ and $\mathbf{u}_i$ with surface normals in the image and model respectively while the constraint function and its variance (which are both scalars) evaluated at the estimates are

$$\begin{aligned} g \;&=\; \hat{\mathbf{v}}_1^T \hat{\mathbf{v}}_2 - \hat{\mathbf{u}}_1^T \hat{\mathbf{u}}_2 \\ G \;&=\; \hat{\mathbf{v}}_2^T V_1 \hat{\mathbf{v}}_2 + \hat{\mathbf{v}}_1^T V_2 \hat{\mathbf{v}}_1 + \hat{\mathbf{u}}_2^T U_1 \hat{\mathbf{u}}_2 + \hat{\mathbf{u}}_1^T U_2 \hat{\mathbf{u}}_1. \end{aligned} \tag{5}$$

In the second case, namely the invariance of the (squared) distance between central points, the parameters $\mathbf{v}_i$ and $\mathbf{u}_i$ are the central points and the constraint function and its variance (again, both scalars) are

$$\begin{aligned} g \;&=\; (\hat{\mathbf{v}}_1 - \hat{\mathbf{v}}_2)^T (\hat{\mathbf{v}}_1 - \hat{\mathbf{v}}_2) - (\hat{\mathbf{u}}_1 - \hat{\mathbf{u}}_2)^T (\hat{\mathbf{u}}_1 - \hat{\mathbf{u}}_2) \\ G \;&=\; 4[(\hat{\mathbf{v}}_1 - \hat{\mathbf{v}}_2)^T (V_1 + V_2)(\hat{\mathbf{v}}_1 - \hat{\mathbf{v}}_2) + (\hat{\mathbf{u}}_1 - \hat{\mathbf{u}}_2)^T (U_1 + U_2)(\hat{\mathbf{u}}_1 - \hat{\mathbf{u}}_2)]. \end{aligned} \tag{6}$$

The effectiveness of such constraints as these depends on the probability of unrelated features being able to conspire randomly to satisfy the constraint. If this probability is too high, then the constraint will not be very effective at pruning the tree. Figure 2 shows the experimentally determined probability of two and three pairs of randomly generated planes conspiring to satisfy the invariant angles constraint as a function of error. The error is the standard deviation on each component of the planes' normal vectors. It is apparent that

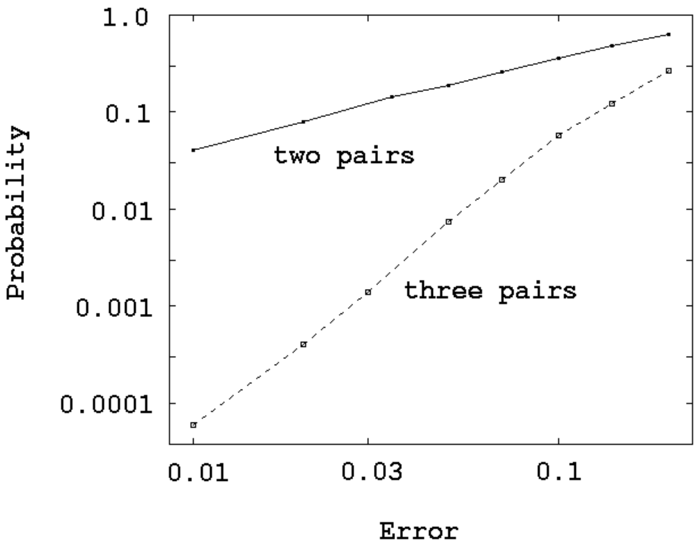

Figure 2: The probability that two and three pairs of random planes pass the invariant angles binary constraint as a function of the error associated with each component of the surface normal vectors.

as the error increases the probability of random combinations of planes being able to satisfy the constraints also increases.

The same trend appears in Figure 3 which shows the probability of two and three pairs of central points conspiring to satisfy the invariant distance constraint as a function of the fractional error in area. There are two main differences from the invariant angles case discussed previously. First, there is a dependency on the volume of space in which the image and model points are distributed since, for a given error, there is a greater chance of the constraint being satisfied if the points are sampled from a smaller volume. We have assumed that the image points come from a cubic volume of 100m^3 and the model points are distributed inside a volume of 10000m^3, roughly simulating the situation where the robot is viewing a room while maintaining a model of a building.

The second difference is in the size and shape of the error envelopes around each central point. Surfaces are often partially occluded, either by the image edge or by foreground objects, so the uncertainty in their central points can be mainly due to their full extent not being visible rather than to the limited range resolution of the sensor. However, this uncertainty occurs only in the plane of the surface. Perpendicular to the plane it is possible to locate the points with

[2]Defining the Jacobian $\partial \mathbf{y}/\partial \mathbf{x} = [\nabla_{\mathbf{x}} \mathbf{y}^T]^T$

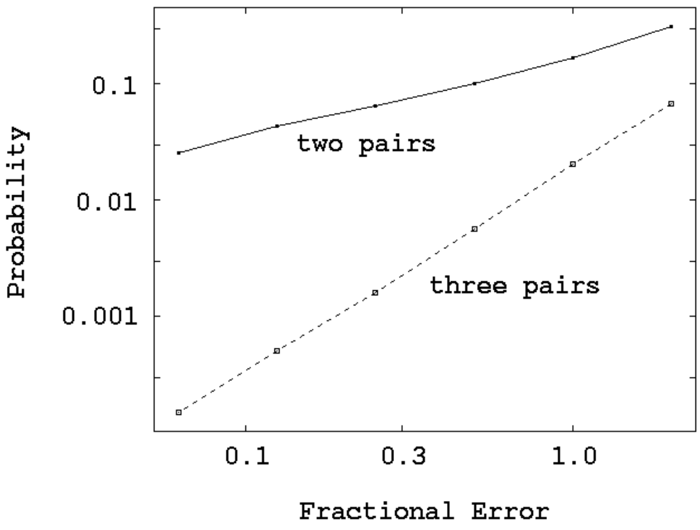

Figure 3: The probability that two and three pairs of random planes will pass the invariant distance binary constraint as a function of the fractional error in surface area.

the full accuracy of the sensor (about 0.02m in our case). To generate Figure 3 we modeled the uncertainty accordingly by ensuring that the covariance matrix of each surface's central point had an eigenvector parallel to the surface normal with a relatively small eigenvalue. The error perpendicular to the surface normal we related to the fractional error in area, κ (see Section 2.1.1), since once again the uncertainty of the estimate depends on how much area is visible.

Our own laser ranger typically gives errors of about 0.05 in the components of unit direction vectors and range errors of about 2cm. The uncertainty associated with surface central points will vary from case to case as different parts and amounts of surfaces are occluded but an average value of 0.5 for κ seems reasonable. With these parameter values, we estimate the probabilities for two and three pairs satisfying the combined binary constraints as

$$p_2 \approx 2 \times 10^{-2},$$

$$p_3 \approx 7 \times 10^{-5}.$$

2.1.3 Efficient Computation of Binary Constraints

In both invariant angle and invariant distance cases the expression for G, the covariance of $\mathbf{g}$, involves terms of the form $\mathbf{x}^T A\mathbf{x}$, where $\mathbf{x}$ is a column vector and A is a covariance matrix. Since covariance matrices are real and symmetric, we can use the Rayleigh-Ritz theorem [9] to set bounds on the terms $\mathbf{x}^T A\mathbf{x}$, then on G and, ultimately, on the Mahalanobis distance d (Equation 3). The point of doing this is that calculating bounds on G is quicker than calculating its value and the bounds may be sufficient for determining the result of the test in Equation 4. When the bounds include the threshold ϵ the exact value G must still be computed but, on average, the computation time for performing the test will be reduced.

The Rayleigh-Ritz theorem states that for a real symmetric matrix A which has minimum and maximum eigenvalues $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$

$$\lambda_{\min}(A) \;\leq\; \frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \;\leq\; \lambda_{\max}(A) \quad \forall \mathbf{x} \neq \mathbf{0}$$

When a matrix is diagonal, its minimum and maximum eigenvalues are the minimum and maximum diagonal entries and so are easy to calculate. In the general case there are no simple bounds on the largest and smallest eigenvalues. However, the eigenvalues of a covariance matrix, which is positive definite, are all positive so

$$\lambda_{\max}(A) \;\leq\; \mathrm{trace}(A)$$

because the trace of a matrix is the sum of its eigenvalues. So for covariance matrices we at least have an easy upper bound on G and a corresponding lower bound on the Mahalanobis distance d (Equation 3). This is useful because most of the time the constraint will be testing false hypotheses, d will be large and its lower bound will be greater than the threshold ϵ.

For the invariant angles constraint, noting that for unit directions $\hat{\mathbf{u}}_i^T\hat{\mathbf{u}}_i = \hat{\mathbf{v}}_i^T\hat{\mathbf{v}}_i = 1$, G (Equation 5) is bounded above by

$$G \;\leq\; \mathrm{trace}(V_1 + U_1) + \mathrm{trace}(V_2 + U_2)$$

For the invariant distance constraint G (Equation 6) is bounded above by

$$4\left[\mathrm{trace}(V_1 + V_2)(\hat{\mathbf{v}}_1 - \hat{\mathbf{v}}_2)^T(\hat{\mathbf{v}}_1 - \hat{\mathbf{v}}_2) + \mathrm{trace}(U_1 + U_2)(\hat{\mathbf{u}}_1 - \hat{\mathbf{u}}_2)^T(\hat{\mathbf{u}}_1 - \hat{\mathbf{u}}_2)\right].$$

Most of the calculations involved in performing this quick test can be reused if, when the lower bound on d is lower than ϵ, the exact value of G still has to be calculated. We found that the average computation time was reduced by a factor of about five by the incorporation of this quick test.

2.2 Estimating a Transformation

Once a partial interpretation acquires three or more pairings which do not involve the wild card (we will call them *proper* pairings) it is possible to estimate the transformation from the image to model reference frames. We use a similar formulation to Ayache and Faugeras [1] involving an iterated extended Kalman filter [2]. The Kalman filter is a tool for estimating a *state* vector and its covariance from a number of *observation* vectors and their covariances and linear *measurement equations* relating each observation to the state. The iterated extended form of the filter is an adaptation for non-linear measurement equations. In our case, each pairing of an image plane to a model plane contributes one 8-dimensional observation vector, $[\hat{\mathbf{q}}_i^T \; \hat{\mathbf{p}}_i^T]^T$, together with an 8-by-8 covariance matrix, towards the estimation of a 6-dimensional state vector. The state vector, $[\mathbf{r}^T \; \mathbf{t}^T]^T$, contains the six parameters of the transformation from image to model frames: three for rotation, $\mathbf{r}$, and three for translation, $\mathbf{t}$. The measurement equation is

$$\mathbf{p}_i \;=\; \begin{bmatrix} R & \mathbf{0} \\ \mathbf{t}^T R & 1 \end{bmatrix} \mathbf{q}_j \tag{7}$$

where R is a rotation matrix parameterised by $\mathbf{r}$. We used the parameterisation where $\mathbf{r}$ is the product of the rotation angle and the rotation axis [1, 5, 11]. Successively better estimates, $[\mathbf{r}_i^T \; \mathbf{t}_i^T]^T$, $i = 1, 2, 3$, and smaller state covariance matrices are generated by the filter as each observation is processed. Some care is needed in the choice of $[\mathbf{r}_0^T \; \mathbf{t}_0^T]^T$ to initialise the sequence and ensure a good linearisation of the measurement equation [11]. We found that an effective method was to use the three sets of estimates, $[\hat{\mathbf{q}}_i^T \; \hat{\mathbf{p}}_i^T]^T$, to generate an analytic solution as if they were uncorrupted by noise and to use this as the initial estimate.

Satisfaction of the binary constraints by three pairs of corresponding plane patches does not ensure that a transformation can be found which maps the data planes onto the model planes, because the constraints discussed in Section 2.1.2 do not exclude the possibility that the relationship between the two sets of planes involves a reflection. Consequently we must impose consistency tests on the observations used to estimate the transform and reject the three pairings if any of the tests fail. Each test consists of evaluating the function

$$\mathbf{g}(\mathbf{r}, \mathbf{t}, \mathbf{p}, \mathbf{q}) \;=\; \mathbf{p} - \begin{bmatrix} R & \mathbf{0} \\ \mathbf{t}^T R & 1 \end{bmatrix} \mathbf{q}$$

and its covariance matrix, G, at the estimates $\hat{\mathbf{r}}_{i-1}$ and $\hat{\mathbf{t}}_{i-1}$ (derived from the previous $i-1$ observations) and $\hat{\mathbf{p}}_i$ and $\hat{\mathbf{q}}_i$ from the current one. If the Mahalanobis distance, $\mathbf{g}^T G^{-1} \mathbf{g}$, is larger than the appropriate χ^2 limit then the test fails. With random data there is exactly one chance in two that there will be a reflective component in the relationship between the data and model surfaces passing the binary tests, so the probability, p_T, of obtaining a consistent and accurate transform estimate is 0.5.

2.3 Switching to a Quadratic Search

Once in possession of a good estimate of position it is possible to transform the set of image plane parameters with their covariances, $\{(\hat{\mathbf{q}}_i, Q_i)\}$, into the world model reference frame obtaining $\{(\hat{\mathbf{s}}_i, S_i)\}$. These can be compared with the model plane parameters and their covariances, $\{(\hat{\mathbf{p}}_j, P_j)\}$, using a Mahalanobis distance of

$$d_{ij} = (\hat{\mathbf{s}}_i - \hat{\mathbf{p}}_j)^T (S_i + P_j)^{-1} (\hat{\mathbf{s}}_i - \hat{\mathbf{p}}_j)$$

based on the measurement equation $\mathbf{s}_i - \mathbf{p}_j = \mathbf{0}$.

A similar technique to that used in Section 2.1.3 can be used here to speed up the average computation time for calculating and testing the value of d_{ij}. A lower bound on d_{ij} is

$$\begin{aligned} d_{ij} &\geq \lambda_{\min}((S_i + P_j)^{-1})(\hat{\mathbf{s}}_i - \hat{\mathbf{p}}_j)^T (\hat{\mathbf{s}}_i - \hat{\mathbf{p}}_j) \\ &= \frac{(\hat{\mathbf{s}}_i - \hat{\mathbf{p}}_j)^T (\hat{\mathbf{s}}_i - \hat{\mathbf{p}}_j)}{\lambda_{\max}(S_i + P_j)} \\ &\geq \frac{(\hat{\mathbf{s}}_i - \hat{\mathbf{p}}_j)^T (\hat{\mathbf{s}}_i - \hat{\mathbf{p}}_j)}{\mathrm{trace}(S_i + P_j)}. \end{aligned}$$

We found that checking if this lower bound was greater than the chosen χ^2 threshold *before* resorting to inverting the matrix $(S_i + P_j)$ resulted in a decrease in average computation time of about one order of magnitude.

The first step of transforming all the image planes can be accomplished with a Kalman filter. Equation 7 above (with $\mathbf{p}_i$ replaced by $\mathbf{s}_i$) is again used as the measurement equation but this time with the observation vector $[\mathbf{q}_i^T \; \mathbf{r}^T \; \mathbf{t}^T]^T$ and state vector $\mathbf{s}_i$. Any further pairings of image and model planes passing the test are added to the original three and the complete set of pairings, together with its estimated transform (which can be further refined using the new pairings), are stored as a candidate solution.

2.4 Interpreting the Candidate Solutions

When the tree has been completely searched the list of candidate solutions is examined. If there are no candidate solutions then we can conclude that all the surfaces in the image are being observed for the first time and must simply be inserted into the world model with covariances reflecting the fact that little is known about where they are (unless there is some information from the robot's odometry). If there is only one solution we can transform all the image surfaces into the world frame, update those surfaces in the world model which participated in the solution and insert the others as new additions observed for the first time. The latter will be relatively accurately placed because the correspondence solution has enabled a good estimate of the image to model transform.

If there is more than one candidate solution then there are two cases to consider. In the first case, all the solutions are mutually exclusive (no two contain the same pairing). One of these will probably correspond to the set of static surfaces in the scene which haven't moved between building up the world model and acquiring the image. The other solutions will correspond to groups of three or more surfaces which have changed position together (i.e. belong to moving objects). These surfaces cannot be updated in the world model because their parameters have changed. The old surfaces must be deleted and replaced by their counterparts in new positions. Odometry, if there is any, can be employed to help choose which solution corresponds to the static part of the scene. The alternative is to examine the kinds of surfaces participating in each solution. The static scene is likely to include more large surfaces (walls, floors and so on) than the moving objects.

In the second case not all candidate solutions are mutually exclusive and so there are competing solutions either in terms of different model surfaces for the same image surface or different transforms for the same pairings. Again, the robot's own position estimate might help in making the right choice, but in the last resort it may be necessary to discard a particular image as presenting an unresolvable conflict and instruct the robot to move somewhere else for the next view of the world.

3 Complexity Analysis

We now derive an expression for the average time required to search the tree as a function of the critical parameters listed in Table 1. We examine the case where both the model and image surfaces are randomly and independently generated so that there is no correspondence between them. The analysis would be somewhat different if a correspondence existed but the case we examine is relatively simple to analyse while still being a useful guide to the complexity. It will at least provide a lower bound to the time required to search a tree of a given size. As well as search time, we want to predict how many spurious solutions are likely to be produced by random conspiracies of surfaces in the image and world model passing the consistency tests.

Table 1 lists those parameters which are critical to the search time together

description	name	estimate
probability of random pass of unary constraint	p_1	0.8
probability of random conspiracy of two pairs	p_2	2×10^{-2}
probability of random conspiracy of three pairs	p_3	7×10^{-5}
probability of finding a consistent transform	p_T	0.5
time to evaluate one unary constraint	t_U	2×10^{-5}
time to evaluate one binary constraint	t_B	2×10^{-4}
time to estimate a transform	t_E	2×10^{-1}
time to transform a data plane	t_T	2×10^{-2}
time to compare two planes in the same frame	t_C	1×10^{-4}

Table 1: Parameters used in the complexity analysis. Times are in seconds.

with estimates for them. The probability p_1 corresponds to the use of the unary constraint on area for a fractional error of $\kappa = 0.5$ (see Figure 1). The probabilities p_2 and p_3, estimated in Section 2.1.2, determine how efficient the binary constraints are in pruning the tree. The probability p_T (see Section 2.2) is the probability of finding a consistent transform for three pairs of planes which pass the binary constraint tests. The bulk of the search time will be spent performing either unary or binary tests, estimations of a transform from three pairings, transforms of data surfaces into the model frame and comparisons of planes in the model frame. Table 1 gives estimates of the average time, (in seconds), required to perform each of these tasks on a SUN Sparcstation. The binary test time, t_B, and the plane comparison time, t_C, correspond to the efficient versions discussed in Sections 2.1.3 and 2.3, respectively.

The number of unary tests that have to be carried out is only MD and the time to compute one unary test is small compared to the other operations. Consequently, the contribution of unary tests to the overall running time is negligibly small. The total number of binary constraint tests performed on average is

$$p_1^2 M^2 \binom{D}{2} + p_2(1+p_2)p_1^3 M^3 \binom{D}{3} \approx \frac{p_1^2 M^2 D^2}{6}(3 + p_1 p_2 M D)$$

(for $M, D \gg 1$ and $p_2 \ll 1$). The first term arises from tests of the consistency of nodes which have exactly two proper pairings. The second term arises from testing the consistency of a third proper pairing being added to those nodes which survived the previous test. The subterm $(1 + p_2)$ is explained by the fact that it is unnecessary to perform the second binary test if the first fails when combining a third pairing with two existing ones. The total number of transform estimates made at nodes which have three proper pairings and have survived the binary constraint tests is

$$p_3 p_1^3 M^3 \left(\begin{array}{c} D \\ 3 \end{array} \right) \approx \frac{p_3 p_1^3 M^3 D^3}{6}.$$

Only p_T of these will give rise to a consistent transform estimate, the rest will involve reflection. For each solution which survives after the transform estimate, $D-3$ image surfaces have to be transformed into the world model and so there are

$$p_T p_3 p_1^3 M^3 (D-3) \left(\begin{array}{c} D \\ 3 \end{array} \right) \approx \frac{p_T p_3 p_1^3 M^3 D^4}{6}$$

such transformations. Each transformed data surface has to be compared to all M model surfaces (since we allow the same model surface to match different image surfaces) and there are in total

$$p_T p_3 p_1^3 M^4 (D-3) \left(\begin{array}{c} D \\ 3 \end{array} \right) \approx \frac{p_T p_3 p_1^3 M^4 D^4}{6}$$

of these comparisons. Thus the total time to search the tree is approximately

$$\frac{p_1^2 M^2 D^2}{6}[(3 + p_2 p_1 M D) t_B + p_3 p_1 M D (t_E + p_T D (t_T + M t_C))].$$

Note that as M and D get very large the search time is dominated by the term involving $M^4 D^4$, namely the time for the surface comparisons in the quadratic search.

The number of solutions with at least three proper pairings, all spurious since both the model and data surfaces are random, is

$$p_T p_3 p_1^3 M^3 \left(\begin{array}{c} D \\ 3 \end{array} \right) \approx \frac{p_T p_3 p_1^3 M^3 D^3}{6}.$$

Figure 4 shows the variation of search time (in seconds) with number of model surfaces (M) for three representative values for the number of image surfaces (D). The estimates in Table 1 have been used for the critical parameters. Clearly, as M and D get large the search becomes intractable.

These results show that even for moderate sized problems the search time is too large. One way to reduce the search time in a dramatic fashion is to structure the world model. Instead of treating it as a single collection of a large number of surfaces, treat it as multiple collections of small numbers of surfaces. The intuition here is that the sensor cannot be in two places at the same time and the correspondence problem becomes the problem of recognising which part of the world model is being viewed. This makes our way of handling data fusion

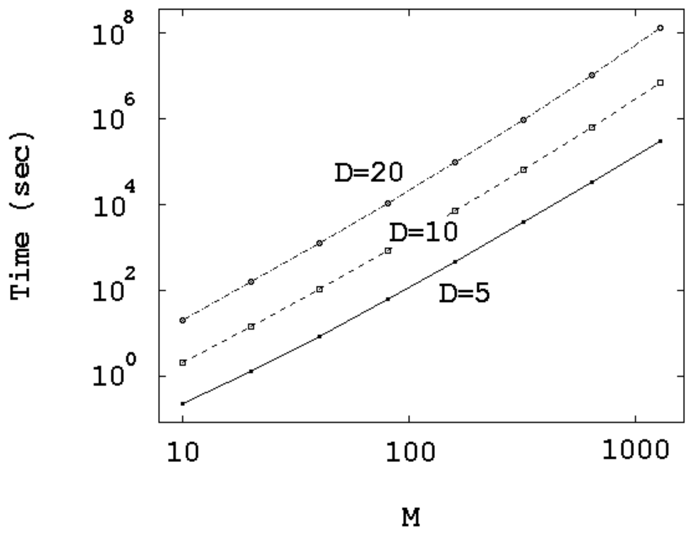

Figure 4: The variation of search time with M for three values of D.

even more like object recognition because the image features are compared to a number of different models each with a relatively small number of features.

To illustrate, suppose we have an image containing ten surface features and a world model containing 50 groups of 20 surfaces each, for a total of 1000 features. From Figure 4 we see that such a problem, with $M = 1000$ and $D = 10$, would require a search time of about thirty days. If, instead of searching one large tree, we search 50 small trees with $M = 20$ and $D = 10$ then the search time reduces to 50×14 seconds ≈ 12 minutes.

We do not yet know a good set of general principles for partitioning the world model into separate parts. However, for indoor scenes, with which we are primarily interested, an obvious choice is to make each room one part, discovering rooms by the six or more large inward facing surfaces which enclose them. As the robot could conceivably view two rooms at once, it may be necessary to base each part of the world model on the interfaces (doors) between rooms, thus allowing parts to contain the same features. For example, if room A is connected to room B and B is connected to room C then the parts of the world model based on the doors connecting A to B and B to C will both contain the features of room B. The governing principle here is that whatever the robot views it should all be contained in one part of the world model.

4 Conclusions

Grimson [7] has shown that for recognising objects in cluttered environments the basic interpretation tree approach leads to a search which is exponential in the worst case, despite the use of unary and binary constraints. We have shown that if a switch is made to a quadratic search when enough pairings have accumulated to make a motion estimate (three, in our case) then the average time complexity is a polynomial of low degree in the number of image and model features (D and M respectively). While this is better than exponential, the relatively expensive stochastic computations and the size of a typical world model mean that large search times are inevitable. The single most effective solution to this problem is, we believe, to divide the world model into parts. This makes a considerable difference in the search time and also reduces the number of spurious solutions. We are currently investigating this method in more detail.

However, even with a good partitioning of the world model our work has some way to go before we can reach the goal of fusing a range image in two seconds and we are exploring a number of other avenues for further improvement. Increased accuracy results in a decrease in the values of p_2 and p_3 and a strengthening of the pruning power of the constraints, so more accurate sensing would help. Exploiting other sorts of information such as colour, texture or shape would strengthen the power of the unary constraint (decrease the value of p_1). Another way is to order the tree in terms of most likely pairings first, least likely last. This requires single attribute comparisons of the sort used for unary constraints (to do the ordering) along with some notion of what a "good" solution is (to stop the search). Whenever a good solution is found, the model and image surfaces participating in it can be removed from consideration before continuing to search for any remaining correspondences. One final possibility is to be selective about which surfaces from the image and world model are allowed to participate in the search for a correspondence, in order to keep the values of M and D as low as possible. It would make sense to focus on large unoccluded surfaces with small uncertainties. It should be borne in mind that we are not assuming any prior knowledge about where the sensor is. In practice, although the robot may take a relatively long time to discover where it is starting from scratch, thereafter the knowledge gained about its whereabouts can be exploited to reduce the amount of search required for subsequent data, since it is not likely to move very far between successive observations.

References

[1] N. Ayache and O.D. Faugeras. Maintaining representations of the environ-

ment of a mobile robot. In *Robotics Research 4*, pages 337–350. MIT Press, USA, 1988.

[2] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association.* Academic Press, UK, 1988.

[3] C.H. Chen and A.C. Kak. A robot vision system for recognising 3-d objects in low-order polynomial time. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1535–1563, 1989.

[4] T.J. Fan, G. Medioni, and R. Nevatia. Recognzing 3-d objects using surface descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1140–1157, 1989.

[5] O.D. Faugeras. A few steps towards artificial 3-d vision. In M. Brady, editor, *Robotics Science.* MIT Press, USA, 1989.

[6] O.D. Faugeras and M. Hebert. The representation, recognition, and locating of 3d shapes from range data. *International Journal of Robotics Research*, 5(3):27–52, 1986.

[7] W.E.L. Grimson. *Object Recognition by Computer: the Role of Geometric Constraints.* MIT Press, USA, 1990.

[8] W.E.L. Grimson and T. Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3):3–34, 1984.

[9] R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, USA, 1985.

[10] M.J.L. Orr, R.B. Fisher, and J. Hallam. Uncertain reasoning: Intervals versus probabilities. In *British Machine Vision Conference*, pages 351–354. Springer-Verlag, 1991.

[11] Z. Zhang and O.D. Faugeras. Determining motion from 3d line segment matches: a comparative study. *Image and Vision Computing*, 9(1):10–19, 1991.