Model-based recognition using 3D structure from motion

D W Murray

The effectiveness of shape information alone, without size, for recognizing stored 3D models is considered. The geometric constraint filtering method of Grimson and Lozano-Pérez is used to curb the potentially combinatorial expansion of the model search space. Results, typical of those from several models experimented with, are given for the task of recognizing a plug from uncertain surface normal data. They show that, at least for an 'interesting' view, shape data is highly effective, even when the sensed surface normals are uncertain in direction, to, say, ± 10°. The loss of size information does, however, result in a drop in search efficiency, but this appears relatively small: a factor of ≈ 5 for the example of a plug at the 10° error mark.

Keywords: shape recognition, 3D models, constraint filtering

In many situations the primary task for a robot vision system is to construct for its control unit explicit descriptions of what is there in its surroundings. In a typical bottom-up 3D machine vision system, one can envisage doing this by carrying out some model-matching process after low-level image processing and shape-from-X techniques have been used to elicit detailed information about the 3D positions of points in the scene and about the orientation of their underlying surfaces.

This paper examines to what extent structural primitives describing the shape of an object but not its size can assist in object recognition. The immediate concern is recognition using 3D information obtained from visual motion, where the well-known depth-speed scaling ambiguity means that absolute depth and size information is not obtained. The issue, moreover, is more generally pertinent to object recognition strategies in computer vision. First, even if one has information about absolute size, one might be concerned with recognizing objects in shape classes, e.g. similarly shaped spanners of different sizes. Second, unless an active rangefinder or some high-accuracy stereo ranging system is used, the precision of absolute size information obtained could be rather poor. In the model recognition problem, the efficacy of shape in combination with accurate depth information is established, but there have been no quantitative studies on the effectiveness of shape, or surface orientation, alone in constraining model-matching.

It appears that in much of the work on exploiting visual motion there remains a gap between the quality of optic flow obtainable from real imagery and the quality of flow that current realistic 3D scene reconstruction algorithms require. The problems on both sides of this gap merit, and are receiving, continued attention, but it is important to relate the problem to one of the primary goals, i.e. object recognition. It is hoped that study at the object recognition stage will focus attention on the weak spots in lower-level vision processing and guide the choice and design of reconstruction algorithms.

The present study is based on the recognition method of Grimson and Lozano-Pérez, who approach model-based recognition as a search problem, mitigated by geometrical constraint propagation. The technique has several advantages in the present context. First, the search strategy is quasi-exhaustive and the geometric constraints are simple and explicit, making it straightforward to assess the result of, say, losing the absolute distance constraint, or increasing the error in the sensed surface orientations. Second, it uses planar-faced models; techniques for determining the orientation and relative motions of planar surface facets have already been demonstrated and are under further development.

OBJECT RECOGNITION PARADIGM

Grimson and Lozano-Pérez divide the matching problem into two stages. In the first they generate feasible inter-
Figure 1. Example of an IT for a model with four faces

The interpretation of the sensed data, using geometric constraints to control the potentially combinatorial expansion of the search space. The aim of this is to produce interepresible few possible interpretations, from which, in the second step, the best is chosen by trial localizations of the model object into sensor space. The crucial step is the first, and it is chiefly this which is explored here for shape data.

There are several assumptions about data and models. First, sensing yields the scaled (not absolute) positions and surface orientations of infinitesimal planar patches of objects in the sensor coordinate frame, to within some volume and cone of error respectively. Second, the data is segmented. The search strategy of Grimson and Lozano-Pérez in its simplest form is inefficient if data from another object is included. Third, the objects can be modelled as a set of planar faces. Only the plane equation and extent of each face are used: no connectivity information is assumed.

Given k sensed data points and a model object with f faces, there are naively a potential $f^k$ ways of assigning the data to the model. Each of these interpretations is represented by a leaf of the interpretation tree, the upper part of which is sketched in Figure 1. This typically vast search space is reduced by introducing geometric constraints on pairs of data, that is, by establishing facts such as

IF | sensed datum 1 is assigned to face 2 |
THEN | sensed datum 2 (can’t can) be assigned to face 4 |

on the basis of consistency between the model and sensed data geometries. The ‘can’t’ answer, suggested in Figure 1, means that an entire sub-tree need not be expanded. The pair constraints become more powerful as the tree is explored deeper, because to establish a branch for sensed datum j requires an unbroken ancestry of $(j - 1)$ ‘can’ facts.

The pair constraints adopted comprise an angle constraint and three simple direction constraints and are derived from the four coordinate-frame independent metrics

$$\hat{u}_a \cdot \hat{u}_b, \hat{d}_a \cdot \hat{u}_a, \hat{d}_b \cdot \hat{u}_a, \hat{d}_b \cdot \hat{d}_b$$

for sensed data $P_a$ and $P_b$. The vectors $\hat{u}_a, \hat{d}_a$ are the measured surface normals at points $P_a$ and $P_b$, and $\hat{u}_b$ is $(\hat{u}_a \times \hat{u}_b)/|\hat{u}_a \times \hat{u}_b|$. Vector $\hat{d}_b$ is the unit vector in the direction of the displacement between $P_a$ and $P_b$.

Figure 2. Vectors describing: a, sensed data; b, model data

as illustrated in Figure 2. Also shown are the corresponding model vectors for faces i and k: these are $\hat{t}_i, \hat{t}_i$, and $\hat{t}_k$. Because structure-from-motion data does not supply absolute depth data, the absolute displacement vector $\hat{d}_b$ cannot be used in the direction constraints, as in Grimson and Lozano-Pérez.

For a pairing of $P_a$ and $P_b$ with model faces i and k, these data metrics must be consistent with the model metrics

$$\hat{t}_a \cdot \hat{t}_i, \hat{t}_a \cdot \hat{t}_k, \hat{t}_i \cdot \hat{t}_i, \hat{t}_k \cdot \hat{t}_k$$

where $\hat{t}_i$ is $(\hat{t}_i \times \hat{t}_i)/|\hat{t}_i \times \hat{t}_i|$. Full details of the constraint geometry are given elsewhere, but an example is given here of the use of the angle constraint. Specifically, consider the pruning instance mentioned above, with point 1 on face 2 and point 2 on face 4. On the model the angle between face 2 and face 4 is computed as $\varphi_{24} = \cos^{-1}(\hat{t}_2 \cdot \hat{t}_4)$, where $\hat{t}_i$ is the normal face i. Suppose that the sensed normal $\hat{u}_i$ at $P_i$ has two data points 1 and 2 are uncertain to cone half-angles $\alpha_1$ and $\alpha_2$. There is thus a range of possible angles between the sensed normals and for a valid (can’t) pairing this range must include the value from the model. That is,

$$\cos^{-1}(\hat{t}_i \cdot \hat{t}_i) - \alpha_1 \leq \varphi_{24} \leq \cos^{-1}(\hat{t}_i \cdot \hat{t}_i) + \alpha_2$$

where the angle range is clipped if necessary at 0 and $\pi$.

It may be noted here that the values obtained for all the model metrics (for example, the $\varphi_i$ values for the angle constraint) need be computed off-line once only and can be stored in $f \times f$ look-up tables for runtime comparison with the data. The interpretation tree-
Table 1. Pruning algorithm performance for unsorted edge data from plug (no distance constraint)

<table>
<thead>
<tr>
<th>Normal cone half-angle (rad)</th>
<th>Leaves on IT</th>
<th>Total nodes</th>
<th>Maximum width</th>
<th>Pruning effort (k calls)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>1</td>
<td>1147</td>
<td>136</td>
<td>42.3</td>
</tr>
<tr>
<td>0.05</td>
<td>1</td>
<td>1614</td>
<td>202</td>
<td>59.6</td>
</tr>
<tr>
<td>0.10</td>
<td>1</td>
<td>2932</td>
<td>451</td>
<td>109.1</td>
</tr>
<tr>
<td>0.15</td>
<td>4</td>
<td>3913</td>
<td>538</td>
<td>148.4</td>
</tr>
<tr>
<td>0.20</td>
<td>4</td>
<td>5206</td>
<td>735</td>
<td>207.1</td>
</tr>
<tr>
<td>0.25</td>
<td>4</td>
<td>12187</td>
<td>2746</td>
<td>496.0</td>
</tr>
</tbody>
</table>

Table 2. Pruning algorithm performance for sorted edge data from plug (no distance constraint)

<table>
<thead>
<tr>
<th>Normal cone half-angle (rad)</th>
<th>Leaves on IT</th>
<th>Total nodes</th>
<th>Maximum width</th>
<th>Pruning effort (k calls)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>1</td>
<td>97</td>
<td>27</td>
<td>4.3</td>
</tr>
<tr>
<td>0.05</td>
<td>1</td>
<td>108</td>
<td>27</td>
<td>4.7</td>
</tr>
<tr>
<td>0.10</td>
<td>1</td>
<td>1109</td>
<td>293</td>
<td>42.2</td>
</tr>
<tr>
<td>0.15</td>
<td>4</td>
<td>1766</td>
<td>408</td>
<td>73.3</td>
</tr>
<tr>
<td>0.20</td>
<td>4</td>
<td>3298</td>
<td>534</td>
<td>148.6</td>
</tr>
<tr>
<td>0.25</td>
<td>4</td>
<td>10055</td>
<td>1542</td>
<td>497.0</td>
</tr>
</tbody>
</table>

Figure 3. Sensed positions from near the face edges of a plug

The pruning algorithm itself has been coded as a recursive depth-first search. Before a branch is placed at level $j$, its ancestry is ascertained over $(j-1)$ generations. In making the consistency check at each generation, the constraints are applied in turn, with the most powerful first (empirically it is found that in the absence of a pairwise absolute distance constraint, this is the angle constraint). If a leaf is reached, that completed interpretation (i.e., root–leaf path) is delivered to be tested by geometrical transformation, as outlined below, while the tree search continues for further possible interpretations.

**EXPERIMENTS WITH THE SEARCH ALGORITHM**

Before experiments with the pruning algorithm are described, a set of baseline performance measures on a particular set of data is presented. Table 1 shows the results of pruning with the angle and three simple direction constraints on the data illustrated in Figure 3, orientation data from near the face edges of an electrical plug. This model has 27 faces, of which 14 are visible in the given view, and there are 56 sensed data. (The combinatorial $k^3$ is $>10^{60}$.) The pruning performance is given as a function of the error cone half-angle of the sensed normals. The second column shows the deduced number of feasible interpretations, that is, feasible within the geometric constraints used. The third column shows the total number of nodes in the IT, and the fourth gives the maximum width of the tree. The last column gives the number of calls to the compatibility tester, which is a measure of the computational effort required to obtain the tree. The immediate result to be noticed is that even without the absolute distance the pruning is highly effective. Out of more than $10^6$ possibilities in this example, the constraints allow only a handful of interpretations to survive over a range of error angles. The effort appears to increase steadily at first with increasing angle but then to rise much more steeply above cone half-angles of 0.20 rad.

It has previously been pointed out that the IT pruning efficiency improves when the sensed data is presorted, so that the most effective pruning occurs as close to the tree root as possible. There, because the absolute distance constraint was the most powerful, sorting was carried out on the basis of pairwise separation, with the most distant pair placed first. Although the distance constraint is not used here, it has been found that a similar sorting by pairwise scaled separation is effective in improving the efficiency. Table 2 shows the results after presorting the data, for comparison with those in Table 1. The number of leaves is of course unchanged, but the size of tree and effort are reduced substantially at half-angles up to 0.20 rad. In this example, sorting becomes valueless at higher angles. The profiles of the width of the IT, or number of active interpretations, at the different data depths (generations) are compared graphically in Figure 4a for an error angle of 0.10 rad. The shape is characteristic of rapid quenching of the initial combinatorial explosion by the increasing power of the constraints.

Table 3 shows the performance of the pruner after inclusion of an absolute distance constraint. (This is a separate distance constraint, as described previously, but its inclusion is equivalent to using the angle and four extended direction constraints.) These results may be directly compared with those in Table 2. There is obviously a drop in efficiency when the absolute distance is lost, but the important result here is that the effectiveness of pruning does not deteriorate significantly, even at quite large normal error angles. The profiles of the width of the IT at the different data depths from Tables 2 and 3 are compared graphically in Figure 4b for an error angle of 0.10 rad.

The effect of sensing data from various typical positions on the surface of the model is now examined. It might be guessed that sensing data from the edges of faces should be most effective, as it is then more likely that extreme values of angles and directions will be tested. This is indeed the case. In Figure 4c are shown IT profiles (for an error cone half-angle of 0.10 rad) when data is sensed near the edges, uniformly and from central clustered sites.

Although absolute distances are not available to use a pairwise distance constraint, it is possible to construct a triple scaled distance constraint. If three sensed data $P_1$, $P_2$, and $P_3$ are considered, the ratio of their scaled
displacements is equal to that for the absolute displacements, i.e.

\[ R = \frac{|e \cdot d_i|}{|e \cdot d_o|} \]

is independent of scaling \( e \). If pairing of the data a, b, c, with model faces \( i, j, k \) respectively is considered, then the required ratios from the model are the maximum and minimum of

\[ \frac{|s_i|}{|s_j|} \]

where both vectors originate from the same point on face \( i \), and terminate anywhere on faces \( k \) and \( j \).

An immediate problem with this approach is that to create the look-up table for such a triple constraint requires \( 2^7 \) entries, a considerable size for, say, the plug with 27 faces. However, there is a way of constructing a simpler though weaker triple constraint. Suppose that a pairwise distance look-up table \( D \) contains the minimum and maximum distances between faces \( i \) and \( k \) in \( D_{ik} \) and \( D_{ik} \), respectively, then

\[ \frac{D_{ik}}{D_{ij}} \leq R \leq \frac{D_{ij}}{D_{jk}} \]

Although this triple constraint is successful in reducing the size of the interpretation tree, as shown by comparing Table 4 with Table 2, there is a deterioration in overall efficiency up to error cone half-angles of \( \approx 0.15 \) rad. This occurs because to establish a branch at data level \( j \) under a triple constraint requires the order of \( j \) triple ancestry checks. At larger error angles, however, the weak triple constraint becomes increasingly valuable.

Table 5 shows the cross rejection achieved by the pruner between several different models and sensed data. For example, row = 3, column = 1 shows how many ways sensed data from a brick could be interpreted as coming from a plug model. The bracketed figure is the pruning effort required. Most off-diagonal elements are zero, but the effort required to compute table entries (brick, plug) and (brick, desk) was enormous. In other words, the plug pins and the desk supports are brick-shaped: given shape data from a brick, it is possible to interpret them as plug pins. This highlights an obvious
difficulty with recognition from shape alone. With absolute range data it would have been simple to distinguish between these items by their size.

**TESTING INTERPRETATIONS BY GEOMETRICAL TRANSFORMATION**

Recall that it is the role of the IT pruning algorithm to produce all the interpretations of the data that are consistent with the model within the applied geometric pairwise constraints. This does not guarantee that an interpretation is consistent with the model in a global sense. For this reason each feasible interpretation must be validated by trying to make a globally consistent transformation of the model into sensor space, i.e. there is a need to find if there is a consistent rotation matrix $[R]$, translation vector $t$ and scaling $S$ relating the sensed vector $\sigma_s$ to its corresponding vector in model space $\mu_s$:

$$\sigma_s = S[R]\mu_s + t$$

for all data 'a'. The rotation is obtained first by a simple averaging technique\(^a\) and then a least-squares procedure is used to find $S$ and $t$ simultaneously. Once these quantities have been determined by averaging all the sensed data, each sensed data item is required, after transformation, to lie within some small distance of its respective model face. Otherwise the interpretation is rejected as globally invalid. For those that are globally valid, the sum of the squares of the small deviations from the model faces provides a measure of 'goodness' for each interpretation.

If the data had provided absolute depth and the database contained uniquely sized objects, the scaling factor would have been fixed at unity. However, even with absolute distance, if a search were being made for a scale model in a shape class, the scaling factor should be allowed to vary. On the other hand, if the data provides only scaled depth but the database includes uniquely sized and shaped objects in the database, the computed scale fixes the absolute depth onto the scene. Thus in the structure-from-motion problem the factor $S$ can resolve the depth:speed scaling ambiguity. In this case the scene sizes and depths will have been supplied using an assumed speed: it is found that

$$\text{True size} = (\text{supplied size})/S$$

$$\text{True speed} = (\text{supplied speed})/S$$

Table 6 shows an example of the refinement stage, eliminating interpretations of data found as feasible by

<table>
<thead>
<tr>
<th>Normal cone half-angle (rad)</th>
<th>Interpretations after pruning</th>
<th>Interpretations after testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>0.05</td>
<td>23</td>
<td>6</td>
</tr>
<tr>
<td>0.10</td>
<td>72</td>
<td>6</td>
</tr>
<tr>
<td>0.15</td>
<td>137</td>
<td>6</td>
</tr>
<tr>
<td>0.20</td>
<td>141</td>
<td>6</td>
</tr>
</tbody>
</table>

**Figure 5.** a, true and b, 'rogue' desk positions

The resolution to the puzzle is shown in Figure 5. The particular desk model had a backplane and in the rogue interpretation the writing surface and backplanes have interchanged roles. The existence here of two completely different but viable interpretations stresses the importance of making a quasi-exhaustive search of the matching space. Methods which stop when a good-enough interpretation is found could have had unfortunate consequences for the well-being of the robot in this example.

**CONCLUDING REMARKS**

The experiments show that, at least for an interesting view of an object, shape alone is sufficient to constrain
the number of possible matches to a handful even when the sensed normals are uncertain in direction to \( \pm 10^\circ \).
In the electrical plug example, this handful is pruned from more than 10\(^6\) combinatorial possibilities. The performance improves if the data is pre-sorted and sensed near the face edges, again as one might expect. A triple relative distance constraint, which examined the datum-face pairings of three sets of points and set bounds on the possible ratios of the relative distances between points, reduced the total number of nodes in the tree but did not improve overall efficiency at reasonable error angles. One possibility might be to use this constraint dynamically, i.e. use it only when the tree width is larger than some threshold.

A geometrical refinement stage has been used to validate feasible solutions from the pruner. The scaling factor derived can be used to disambiguate depth and scale in structure from motion.

Finally, consideration is being given to a parallel implementation of the recognition algorithm on the ICL DAP, on which other early vision algorithms already run\(^{12,13}\). There have been three SIMD parallel versions of the tree search devised, all for the Connection Machine. The first\(^{14}\) used the routing capabilities of the machine but did not run much faster than a serial version. The next was a static algorithm\(^{15}\) which exploited its large size (250,000 processors) by placing an IT node on each processor. The most recent version\(^{16}\) is a dynamic variant of the second, though as yet no results have been presented. The relatively small size and limited connectivity of the DAP appear to rule out immediate use of any of these algorithms and probably some other way of propagating the constraints will have to be found.

REFERENCES