RECOGNITION AND LOCALIZATION OF OVERLAPPING PARTS FROM SPARSE DATA IN TWO AND THREE DIMENSIONS

W. Eric L. Grimson
Tomás Lozano-Pérez

Massachusetts Institute of Technology
Artificial Intelligence Laboratory
545 Technology Square
Cambridge, Massachusetts

Abstract. This paper discusses how sparse local measurements of positions and surface normals may be used to identify and locate overlapping objects. The objects are modeled as polyhedra (or polygons) having up to six degrees of freedom relative to the sensors. The approach operates by examining all hypotheses about pairings between sensed data and object surfaces and efficiently discarding inconsistent ones by using local constraints on: distances between faces, angles between face normals, and angles (relative to the surface normals) of vectors between sensed points. The method described here is an extension of a method for recognition and localization of non-overlapping parts previously described in [Grimson & Lozano-Pérez 84] and [Gaston & Lozano-Pérez 84].

1. Problem Definition

The specific problem we consider in this paper is how to identify a known object and locate it, relative to the sensor, using relatively few measurements. We want a recognition technique that is applicable to a wide range of sensors, so we make few assumptions about the sensory data available. We assume only that the sensory data can be processed to obtain sparse measurements of the position and surface orientation of small planar patches of object surfaces in some coordinate frame defined relative to the sensor. The measured positions are assumed to be within a known error volume and measured surface orientations to be within a known error cone. Furthermore, the object is assumed to be overlapped by other unknown objects, so that much of the data does not arise from the object of interest.

If the objects have only three degrees of freedom relative to the sensor (two translational and one rotational), then the positions and surface normals need only be two-dimensional. If the objects have more than three degrees of freedom (up to three translational and three rotational), the position and orientation data must be three-dimensional.

Since the measured data approximate small planar patches of the object's surface, we assume that the objects can be modeled as sets of planar faces. Only the individual plane equations and dimensions of the model faces are used for recognition and localization. No face connectivity information is used or assumed; the model faces do not even have to be connected. This is important. It is easy to build polyhedral approximations of moderately curved objects, but we cannot expect these approximations to be perfectly stable under sensor variations. The connectivity among the faces is particularly vulnerable. Since our recognition method does not use face connectivity, but only local geometry, it can be readily applied to curved objects approximated by planar patches.

Our basic approach to recognition proceeds in two steps:

Generate Feasible Interpretations: A set of feasible interpretations of the sense data is constructed. Interpretations consist of pairings of each sensed patch with some object surface on one of the known objects. Interpretations inconsistent with local constraints, derived from the model, on the sense data are discarded.

Model Test: The feasible interpretations are tested for consistency with surface equations obtained from the object models. An interpretation is legal if it is possible to solve for a rotation and translation that would place each sensed patch on an object surface. The sensed patch must lie *inside* the object face, not just on the surface defined by the equation.

There are several possible methods of actually searching for consistent matches. For example, in Grimson and Lozano-Pérez [84] we chose to structure the search as the generation and exploration of an interpretation tree. That is, starting at a root node, we construct a tree in a depth first fashion, assigning measured patches to model faces. At the first level of the tree, we consider assigning the first measured patch to all possible faces, at the next level, we assign the second measured patch to all possible faces, and so on. The number of possible interpretations in this tree, given s sensed patches and n surfaces, is n^s. Therefore, it is not feasible to explore the entire search space in order to apply a model test to all possible interpretations. Moreover, since the computation of coordinate frame transformations tends to be expensive, we want to apply this part of the technique only as needed.

The goal of the recognition algorithm is thus to exploit local geometric constraints to minimize the number of interpretations that need testing, while keeping the computational cost of each constraint small. In the case of the interpretation tree, we need constraints between the data elements and the model elements that will allow us to remove entire subtrees from consideration without explicitly having to search those subtrees. In our case, we require that the distances and angles between all pairs of data elements be consistent with the distances and angles possible between their assigned model elements.

The recognition algorithm described is related to several recent approaches to recognition based on geometric constraints [Bolles and Cain 82, Bolles, Horaud and Hannah 83, Faugeras and Hebert 83, Gaston and Lozano-Pérez 84, Grimson and Lozano-Pérez 84, Stockman and Esteva 84]. See [Grimson and Lozano-Pérez 84] for a more thorough discussion of the relevant literature.

Acknowledgments. This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's Artificial Intelligence research is provided in part by a grant from the System Development Foundation, and in part by the Advanced Research Projects Agency under Office of Naval Research contracts N00014-80-C-0505 and N00014-82-K-0334.

In this paper, we deal with two different, but related, sets of geometric constraints. The first set is a simple set of constraints in which position and orientation are decoupled. The simplicity leads to very efficient implementations, but the decoupling reduces their pruning power. The second set retains the natural coupling between positions and orientations. This set is more powerful, but computationally more complex. Both sets are developed first for the case of a single, isolated object, and then for the case of several overlapping objects.

2. Decoupled Constraints

We begin by deriving a set of coordinate-frame-independent constraints, which were first presented in [Grimson and Lozano-Pérez 84]. The first question to ask is what types of coordinate-frame-independent constraints are possible, given that the sensory data are sparse planar patches, each consisting of a position measurement and a unit surface normal (see Figure 1). Clearly, a single patch provides no constraint on the model faces that could consistently be matched to it. Therefore, we consider pairs of patches. Each such pair can be characterized by the pair of unit normals, $\mathbf{n}_1$ and $\mathbf{n}_2$, and the separation vector between the patch centers (assuming small patches) $\mathbf{d}$, as shown in Figure 1.

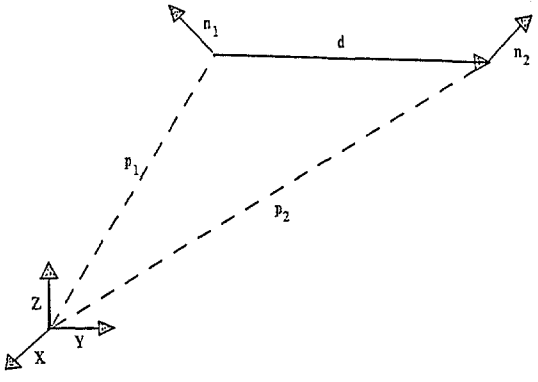

Figure 1. The constraints between pairs of measured surface patches. A given pair of sensory points $\mathbf{P}_1, \mathbf{P}_2$ is characterized by the components of the vector $\mathbf{d}$ between them, in the direction of each of the surface normals $\mathbf{n}_1, \mathbf{n}_2$ and in the direction of their cross product, $\mathbf{n}_1 \times \mathbf{n}_2$, and by the angle between the two normals, $\mathbf{n}_1 \cdot \mathbf{n}_2$.

2.1. The Constraints

First construct a local coordinate frame relative to the sensed data; we use both unit normals as basis vectors. In two dimensions, these define a local system, except in the degenerate case of the unit normals being (anti-)parallel. In three dimensions, the third component of the local coordinate frame can be taken as the unit vector in the direction of the cross product of the normal vectors. In this frame, one set of coordinate-frame-independent measurements is: the components of the vector $\mathbf{d}$ along each of the basis directions and the angle between the two measured normals. More formally,

$$\mathbf{n}_1 \cdot \mathbf{n}_2$$
$$\mathbf{d} \cdot \mathbf{n}_1$$
$$\mathbf{d} \cdot \mathbf{n}_2$$
$$\mathbf{d} \cdot \mathbf{u}$$

where $\mathbf{u}$ is a unit vector in the direction of $\mathbf{n}_1 \times \mathbf{n}_2$.

These measurements are equivalent, but not identical to the set used in [Grimson and Lozano-Pérez 84]. In the earlier paper, we used the magnitude of $\mathbf{d}$ and two of its components; this is equivalent, up to a possible sign ambiguity, to using the three components of the vector. This possible ambiguity was resolved using a triple product constraint.

To turn these measurements into constraints on the search process, we must relate them to measurements on the model elements. Since objects are modeled as sets of planar faces, the relationship is straightforward. Consider the first measurement, $\mathbf{n}_1 \cdot \mathbf{n}_2$. If this is to correspond to a measurement between two faces in the model, then the dot product of the model normals must agree with this measurement. If they do not agree, then no interpretation that assigns those patches to these model faces need be considered. In the interpretation tree, this corresponds to pruning the entire subtree below the node corresponding to that assignment. The test can be implemented efficiently by precomputing the dot product between all pairs of faces in the models.

Note that in the case of perfect data, this is a very powerful constraint, since it requires that two data points have a relative orientation that is identical to the relative orientation of the corresponding model faces. In practice, however, the surface normals can only be measured to within some cone of error, and this implies that the dot product computed from the sensory data is actually a range of values, defined by bounds on the sensed error. As a consequence, the constraint is somewhat weaker, since if the dot product of the face normals lies within this range, this is a locally consistent assignment of model faces to data points.

Similar constraints can be derived for the components of the separation vector in the directions of the unit normals. Each pair of model faces defines an infinite set of possible separation vectors, each one having its head on one face and its tail in the other. We can compute bounds on the components of this set of vectors in the direction of each of the face normals. Again, for an assignment of patches to model faces to be consistent, the measured value, plus some range of values about it due to error in the sensor, with must agree with the precomputed model values.

Grimson [84] argues from a combinatorial analysis that these constraints are very powerful, and in the case of data all obtained from a single object, will converge quickly to a small set of interpretations. The analysis also shows that the constraints should exhibit a graceful degradation with increasing sensor noise. The performance of the constraints has also been demonstrated by simulation; Grimson and Lozano-Pérez [84] report on a large set of simulations run on a series of test objects, for varying types of error conditions. The technique has also been applied to several different types of real data, including sonar, laser range data, binary images, and edges detected from grey-level images (see Section 2.4.3).

2.2. Adding A Model Test

Once the interpretation tree has been pruned, there are typically only a few non-symmetric interpretations of the data remaining.

It is important to realize, however, that these constraints are not guaranteed to reject all impossible interpretations. Let d_{ij} be the distance between two sensed patches, P_i and P_j. This measured distance may be consistent with the range of distances between faces f_u and f_v, but only if the patches are inside of small regions on the candidate surfaces. Now consider what happens when adding another patch-surface pairing, (P_k, f_w), to an interpretation that already includes (P_i, f_u) and (P_j, f_v). Our constraints permit adding this pairing only if the distances d_{ik} and d_{jk} are consistent with the range of distances between f_u, f_w and f_v, f_w, respectively. In doing this, however, it uses the ranges of distances possible between any pair of points on these faces. It does not take into account the fact that only small regions of f_u and f_v are actually eligible.

Because of this decoupling of the constraints, the fact that all pairs of patch-surfaces assignments are consistent does not imply that the global assignment is consistent. To determine global consistency, we solve for a transformation from model coordinates to sensor coordinates that maps each of the sensed patches to the interior of the appropriate face. There are many methods for

actually solving for the transformation, one is described in [Grimson and Lozano-Pérez 84]. This model test is applied to interpretations surviving pruning so as to guarantee that all the available geometric constraint is satisfied. As a side effect, the model test also provides a solution to the localization problem.

2.3. Data from Multiple Objects

The method as described so far assumes that all the data comes from a single object. Assume that all of the sensed patches, except one, originate from the same object. Let P_i be the extraneous measurement. Usually, it will be impossible to find an interpretation that includes this measurement. But, not all interpretations will fail at level i in the tree; it may require adding a few more data points to the interpretation before the inconsistency is noted. It is only when all possible single-object interpretations fail that we are certain to have at least one extraneous data point.

It may still be possible to find an interpretation of all the data, including extraneous measurements, that is consistent with the pairwise constraints. It is even possible, by a fortuitous alignment of the data, for interpretations involving extraneous data to pass the model test. There is nothing within the approach described here to exclude this possibility. Of course, the larger the number of patch-surface pairings in the interpretation, the less likely this is to happen. In many cases, it may be necessary to verify the interpretation by acquiring more data. We will not pursue this point here; we will assume, instead, that the presence of extraneous points will cause all interpretations to fail either the local constraints or the model test.

One straightforward approach to handling extraneous data points is to apply the recognition process to all subsets of the data, possibly ordered by some heuristic. But, of course, this approach wastes much work determining the feasibility of the same partial interpretations. Can we consider all subsets of the data without wasting the work of testing partial interpretations? The simple way we have done this is by adding one more branch to each node of the interpretation tree, IT. This branch represents the possibility *of discarding the sensed patch as extraneous. Call this branch the null face.* The remainder of the process operates as before except that, when applying the local constraints, the null face behaves as a "wild card"; assigning a patch to the null face will never cause the failure of an interpretation.

It is easy to see that if an interpretation is legal, the process described above will generate all subsets of this interpretation as leaves of the tree. This is true of partial interpretations as well as full interpretations since every combination of assignments of the null face to the sensed patches will still produce a valid interpretation.

The same condition that ensures the validity of this process guarantees its inefficiency. We do not want to generate all subsets of a valid interpretation. In general, we want to generate the *interpretations that are consistent with as much as possible of the* sensed data. The following simple method guarantees that we find only the most complete interpretations.

The IT is explored in a depth-first fashion, with the null face considered last when expanding a node. In addition, the model test is applied to any complete interpretations, that is, any that reach a leaf of the IT. Now, assume an external variable, call it MAX, that keeps track of the longest valid interpretation found so far (where length is taken to be the number of non-null faces paired with sensed data by the interpretation). At any node in the tree, let M denote the number of non-null faces in the partial match associated with that node. It is only worth assigning a null face to patch P_i, if $s - i + M \geq MAX$; s is the total number of sensed patches. Otherwise, the length of the interpretations at all the leaves below this node will be less than that of the longest interpretation already found. If we initialize MAX to some non-zero value, then only interpretations of this length or longer will be found. As longer interpretations are found, the value of MAX is incremented, thus ensuring that we find the most complete interpretation of the data. Note that if an interpretation of length s is found, then no null-face assignments will be considered after that point.

Looking for the longest consistent interpretation allows the matching algorithm to overcome many of the combinatorial problems of the null-face scheme, but it makes the algorithm susceptible to a potentially serious problem. One of the bases of our approach to recognition has been to avoid any global notion of "quality" of match. We have simply defined generous error bounds and found all interpretations that did not violate these bounds. Once all the valid interpretations have been found, a choice between them can be made on a comparative basis rather than on some arbitrary quality measure. The modified algorithm, however, discards valid interpretations that are shorter than longest valid interpretation. Therefore, a long interpretation on the margin of the error bounds can force us to ignore a shorter interpretation that may be the correct match.

We know of no general solutions to this problem. Quality measures such as how well the transformation maps the measured patches onto the faces [Faugeras and Hebert 83] are useful but also susceptible to error. Our choice would be to consider all the valid interpretations whose length is within one or two of the longest interpretation and which are not subsets of a longer interpretation. This is also heuristic. We have avoided this issue in the rest of the paper and simply coped with the occasional recognition error.

2.4. Testing

We have tested the extended method on simulated data as well as on actual data from three sensory modalities.

2.4.1. Simulations with Two-Dimensional Data

We have done extensive testing of the algorithm with simulated two-dimensional data of the type illustrated in Figure 2. A number of polygons, representing the outlines of parts, are overlapped at random. The position and orientation of a number of data patches are determined by computing the outermost intersection of randomly-chosen rays with the polygon boundaries. The position and normal information is then corrupted by random errors designed to simulate the effect of imperfect sensors (see Figure 2, the small circles indicate the sensed patches).

The algorithm performs quite well in this application. As long as enough patches are sensed on the desired object, the algorithm can locate it. On average only one or two legal interpretations are obtained. Furthermore, the time to do the recognition and localization is relatively low: on the order of a few seconds on a Symbolics 3600 Lisp Machine. The time grows when the measurement

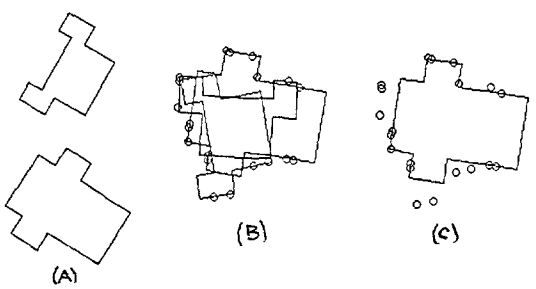

Figure 2. Simulations of overlapping two dimensional parts. (a) Set of outlines (b) Overlapped outlines, circles indicate positions of sensed patches (c) Interpretations.

error grows, in the manner illustrated by the simulations reported in [Grimson and Lozano-Pérez 84].

2.4.2. Simulations with Three-Dimensional Data

We have performed similar simulations with overlapping three dimensional objects, each with six degrees of freedom, as illustrated in Figure 3. The average number of legal interpretations is only slightly higher than for a single object. The computation time, however, is significantly larger: on the order of tens of seconds as opposed to two or three seconds for the single object case.

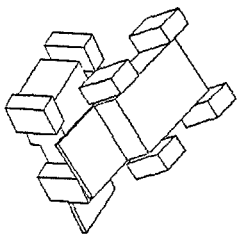

Figure 3. Simulations of overlapping three dimensional parts. Simulations similar to those shown in Figure 2 were performed in three dimensions, with overlapping parts such as those shown here.

In the two dimensional case, a model transformation was computed whenever a leaf of the tree was reached. This transform is checked to see if it transforms the patches onto their associated faces. This ensures that the cutoff using MAX will not exclude any correct interpretations, but it is slow. A faster but less reliable alternative is to generate the IT, using the MAX cutoff as before, but simply collect all pairwise consistent interpretations, ordered by the number of non-null matches. We then apply the model test find the longest consistent interpretations.

Such a technique is typically faster than applying a model test at each leaf of the tree, but it is not guaranteed to find the correct interpretation. In particular, suppose that the correct interpretation is of length m, but that the pairwise constraints allow an interpretation with $m+1$ non-null matches to pass through. This will cause MAX to be at least $m+1$, which will cutoff the correct interpretation. In our simulations of 3D overlapping parts, we have applied this faster, but less reliable technique, and recorded the percentage of cases in which no interpretation was found. The failure rate was typically less than one in twenty.

2.4.3. Edge Fragments from Gray-Level Images

A modified version of the algorithm described here has been applied to locating a simple object in cluttered scenes, using edge fragments from images obtained by a camera located (almost) directly overhead. The images are obtained under lighting from several overhead fluorescent lights. The camera is a standard vidicon located approximately five feet above the scene. The edge fragments are obtained by linking edge points marked as zero crossings in the Laplacian of Gaussian-smoothed images. Edge points are marked only when the gradient at that point exceeds a pre-defined threshold; this is done to eliminate some shallow edges due to shadows. The algorithm is applied to some pre-defined number of the longest edge fragments.

This application requires extensions to the general method. One point to notice is that we have large edge fragments rather than small patches; therefore, we can use the length of the fragments as an additional local constraint. In our implementation, we do not assign edge fragments to model edges that are shorter than the measured fragment; we do assign small edge fragments to long model edges. More importantly, we could compute whole ranges of measurements from the edge fragments (as we do from model edges) rather than the single values from point-like patches we assume elsewhere. The constraints would then require that the measured range be contained in the model range. An easy way of approximating these stronger constraints is by treating the edge as two small patches located at endpoints of the edges, but constraining both patches to be assigned to the same model edge. Both of these approaches can be generalized to three-dimensions.

The most difficult problem faced in this application is that we cannot tell which side of the edge contains the object, that is, the edge normals can be determined only up to a sign ambiguity. The algorithm can be modified to keep track of the two possible assignments of sign and to guarantee that all the pairings in an interpretation have consistent assignments of sign. This approach, however, causes a noticeable degradation in the performance of the algorithm, since it reduces the pruning power of the constraints. Fortunately, we can use another form of the constraints to reduce the effect of this ambiguity.

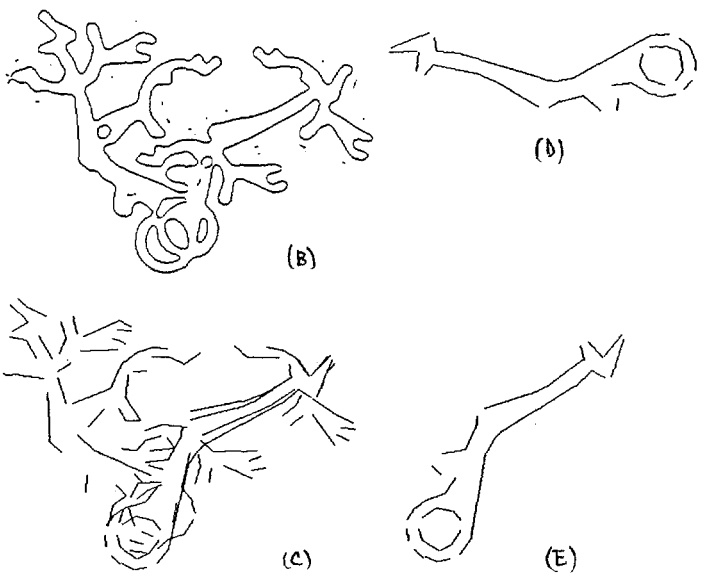

Figure 4. Two dimensional edge data. (a) Grey level images, (b) zero-crossings, (c) edge fragments, (d) model, and (e) located object.

As long as two edges do not cross or are not collinear, at least one edge must be completely within one of the half planes bounded by the other. This means that the components along one of the edge normals of all possible separation vectors will always have the same sign. Given a tentative pairing of two measured edge fragments and two model edges, we can use this property to pick the sign of one of the normals. The angle constraint between normals can then be used to consistently select the signs for other edges in that interpretation. Of course, the sign assignment is predicated on the initial pairing being correct, which it may not be, so we have lost some pruning power in any case.

The algorithm succeeds in locating the desired object in images where the edge data is very sparse (see Figure 4). The speed of performance under these circumstances is not yet acceptable for practical applications; it can take up to 20 seconds of elapsed time on a Symbolics 3600 Lisp Machine to process an image with 50 edge fragments. This is quite a bit slower than the performance of the algorithm when the sign of the normal is available. We are currently investigating extensions of the algorithm to improve this performance.

2.4.4. Range Data from an Ultrasonic Sensor

Michael Drumheller [Drumheller 84] has developed a modified version of the algorithm described above and applied it to range data obtained from an unmodified Polaroid ultrasonic range sensor.

The intended application is navigation of mobile robots. The system matches the range data obtained by circularly scanning from the robot's position towards the walls of the room. The robot has a map of the walls of the room, but much of the data obtained arises from objects on the walls, such as bookshelves, or between the robot and the walls, such as columns.

The algorithm first fits line segments to the range data and attempts to match these line segments to wall segments. After matching, the robot can solve for its position in the room. The data obtained from the sensor is far from perfect. In particular, the beam width is approximately 10 degrees, which leads to significant errors on the length of data segments as well as a wide "penumbra" around nearby obstacles. See [Drumheller 84] for more details.

3. Coupled Constraints

As we noted earlier, the decoupled constraints typically prune most of the non-symmetric interpretations of the data, but they are not guaranteed to reject all impossible interpretations. Consider Figure 5, for example. Consider matching patch P_i to face f_u, patch P_j to face f_v and patch P_k to face f_w. These assignments are pairwise consistent, and the sections of the faces that are feasible locations for the sensed patches are indicated by the sections labeled ij, etc.

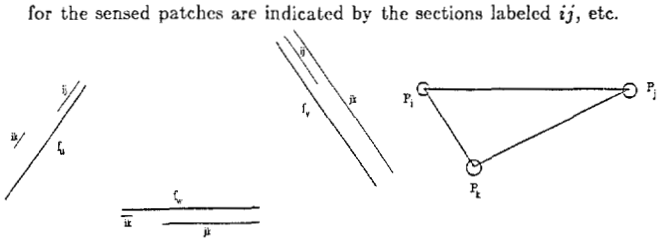

Figure 5. The constraints are decoupled.

The assignment is not globally consistent, however, as indicated by the fact that the segments for face f_u and f_w do not overlap. Thus, since the patches are pairwise consistent with the candidate faces, they are accepted as part of a feasible interpretation, even though clearly they are not. Using the decoupled constraints, it is only after the model test is applied to interpretations surviving pruning that all the available geometric constraint is exploited. For the case of a single object, this merely implies some inefficiency. For the case of multiple, overlapping objects, we may actually miss a correct interpretation. For example, a locally consistent (but globally inconsistent) interpretation of length M will cause us to ignore a globally consistent interpretation of length $m < M$. We would not discover our error until the model test is applied after pruning of the interpretation tree.

One solution is to interdigitate the model test with the tree generation stage. That is, whenever we reach a leaf of the interpretation tree, we apply the model test to ensure that the interpretation is globally consistent. If it is, then we update our global counter MAX, and continue. If it is not, then we continue our search with the current value of MAX. The problem with this method is that it may be computationally expensive. As we stated, the purpose of finding effective local constraints is to enable us to avoid applying an expensive model transformation, except when necessary.

An alternative solution is to find constraints that maintain global consistency without requiring an explicit model transformation. One such set of constraints is developed below for the two-dimensional case.

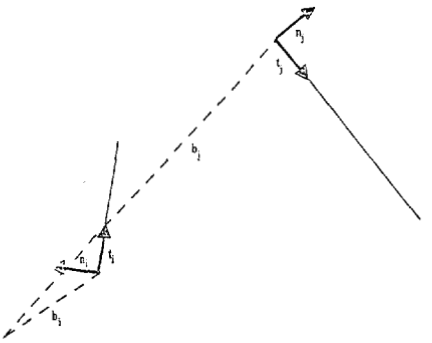

Figure 6. The constraints are recoupled.

3.1. The Coupled Constraints in Two-Dimensions

Suppose we consider two edges of an object, oriented arbitrarily in sensor coordinates, as shown in Figure 6. With each edge we will associate a base point, defined by the vector $\mathbf{b}_i$, a unit tangent vector $\mathbf{t}_i$, which points along the edge from the base point, and a unit normal vector $\mathbf{n}_i$, which points outward from the edge. Thus, the position of a point P_1 along edge f_i in this coordinate system is given by

$$\mathbf{b}_i + \alpha_1 \mathbf{t}_i \qquad \alpha_1 \in [0, \ell_i]$$

where ℓ_i is the length of the edge. Similarly, a point P_2 on face f_j can be represented by

$$\mathbf{b}_j + \alpha_2 \mathbf{t}_j \qquad \alpha_2 \in [0, \ell_j].$$

The vector between two small measured patches is given by

$$\mathbf{d}_{12} = \mathbf{b}_i + \alpha_1 \mathbf{t}_i - \mathbf{b}_j - \alpha_2 \mathbf{t}_j. \tag{1}$$

As in the earlier case, we know that we can measure $\mathbf{d}_{12}$ and $\mathbf{n}_1$, thus we know $\mathbf{d}_{12} \cdot \mathbf{n}_1 = m_{12}$ up to some error range that is a function of the bounds on the sensitivity of the sensor [Grimson and Lozano-Pérez 84]. That is, we know that the true value of $\mathbf{d}_{12} \cdot \mathbf{n}_1$ lies in the range

$$\mathbf{d}_{12} \cdot \mathbf{n}_1 \in [m_{12} - \epsilon, m_{12} + \epsilon]$$

where ϵ can be computed straightforwardly.

From (1) we have

$$\mathbf{d}_{12} \cdot \mathbf{n}_1 = (\mathbf{b}_i - \mathbf{b}_j) \cdot \mathbf{n}_i - \alpha_2 (\mathbf{t}_j \cdot \mathbf{n}_i). \tag{2}$$

The first term is a constant and is a function of the object only, independent of its orientation. Thus, equation (2) provides us with a constraint on the value of α_2. In particular, if $\mathbf{t}_j \cdot \mathbf{n}_i = 0$, then this assignment of patches to faces is consistent only if

$$r_{ij} = (\mathbf{b}_i - \mathbf{b}_j) \cdot \mathbf{n}_i \in [m_{12} - \epsilon, m_{12} + \epsilon].$$

If this is true, then α_2 can take on any value in its current range. If it is false, then the assignment of these patches P_1, P_2 to these faces f_i, f_j is inconsistent and can be discarded.

In the more common case, when $\mathbf{t}_j \cdot \mathbf{n}_i \neq 0$, we have

$$\alpha_j \in \left[\frac{r_{ij} \cdot \mathbf{n}_i - m_{ij} - \epsilon}{\mathbf{t}_j \cdot \mathbf{n}_i}, \frac{r_{ij} \cdot \mathbf{n}_i - m_{ij} + \epsilon}{\mathbf{t}_j \cdot \mathbf{n}_i} \right].$$

Thus, we have restricted the range of possible values for α_2 and hence the set of positions for patch P_2 that are consistent with this interpretation.

Similarly, by using the values for $\mathbf{d}_{12} \cdot \mathbf{n}_2$ obtained from the measurements, we can restrict the range of values for α_1 and, thereby, the position of P_1.

We can also consider the coordinate-frame-independent term

$$\mathbf{d}_{12} \cdot \mathbf{t}_1 = (\mathbf{b}_i - \mathbf{b}_j) \cdot \mathbf{t}_i + \alpha_1 - \alpha_2 (\mathbf{t}_j \cdot \mathbf{t}_i). \tag{3}$$

As before, we can place bounds on the measured value for $\mathbf{d}_{12} \cdot \mathbf{t}_1$. Then, given a legitimate range for α_1 we can restrict the range of α_2 and vice versa. A similar argument holds for $\mathbf{d}_{12} \cdot \mathbf{t}_2$.

These constraints allow us to compute intrinsic ranges for the possible assignments of patches to faces. The key to them is that we can propagate these ranges as we construct an interpretation. For example, suppose that we assign patch P_1 to face f_i. Initially, the range for α_1 is

$$\alpha_1 \in [0, \ell_i].$$

We now assign patch P_2 to face f_j, with

$$\alpha_2 \in [0, \ell_j]$$

initially. By applying the constraints derived above, we can reduce the legitimate ranges for these first two patches to some smaller set of ranges. We now consider adding patch P_3 to face f_k. When we construct the range of legal values for α_3, we find that the constraints are generally much tighter, since the legal ranges for α_1 and α_2 have already been reduced. Moreover, both α_1 and α_2 must be consistent with α_3, so the legal range for this patch is given by the intersection of the ranges provided by the constraints. Finally, the refined range of consistent values for α_3 may in turn reduce the legal ranges for α_1 and α_2 and these new ranges may then refine each other by another application of the constraints, and so on. In other words, the legal ranges for the assignment of patches to faces may be relaxed via the constraint equations, and in this manner, a globally consistent assignment is maintained. Of course, if any of the ranges for α_i becomes empty, the interpretation can be discarded as inconsistent without further exploration.

We thus have the basis for a second recognition and localization technique. As before, we generate and prune a tree of interpretations, by assigning sensed patches to faces of an object. Here there are two types of constraints. The first is that the angle between two sensed normals, modulo error in the sensor, must be consistent with the angle between the corresponding face normals, as in the previous case. The second involves the relaxation of mutual constraints on the range of positions on a face consistent with points of contact on those faces, as described above.

These constraints can be extended to three dimensions.

3.2. Simulations with Two-Dimensional Data

The crucial question to consider for the range propagation technique is to what extent the implicit recoupling of the local constraints reduces the amount of explicit exploration of the interpretation tree. We have done extensive testing of the algorithm with simulated two-dimensional data of the type illustrated in Figure 2. We compared the ratio of the number of nodes explored using the range propagation technique, to the number of nodes explored for the same data using the decoupled constraints. Overall, we found that the average number of nodes explored in the interpretation tree is not significantly reduced from the normal method.

We note that since the error ranges associated with each constraint differ between the normal constraints and the coupled constraints, it is possible for the coupled constraints to actually be less effective in removing portions of the interpretation tree. This is especially noticeable for large values of error in the surface normal. Given the additional overhead associated with computing and intersecting the ranges of feasible positions along edges, it may not be worth while to use the range propagation method. We note that these results may differ when considering objects whose faces do not all form right angles with one another.

4. Summary

We have presented a recognition technique based on a search for consistent matches between local geometric measurements and model faces. The technique offers a number of advantages: it is very simple yet efficient; it can operate on sparse data; it is applicable to a wide range of sensory modalities and choice of features; it degrades gracefully with error.

In addition to the advantages of the particular technique, the framework within which it has been developed has proven useful both to analyze expected performance of this method and to model a number of other methods. In fact, we have described a framework for a class of recognition algorithms. We considered two major variations depending on the class of constraints employed; some minor variations were employed in dealing with different sensory modalities, notably grey-scale edges. These variations, however, share many common assumptions as to the structure of the search for consistent matchings. We have assumed, for example, that we match some subset of the data elements against all the model elements at once; that we obtain all (longest) consistent interpretations; that the objects have comparable number of degrees of freedom as the measurements. Beyond these algorithmic assumptions, we have preserved some assumptions about our domain. We have assumed, for example, that the data is made up of simple local measurements such as surface patches; that the model is made up of planar faces; that the dimensions of the objects are fixed and known a priori. All of these assumptions can be relaxed while retaining the characteristic flavor of the approach presented here. We have implemented all of these extensions with relatively minor modifications to the program code.

Acknowledgments

The idea of using a null face to handle multiple objects was first suggested to one of us (TLP) by V. Melenkovic of CMU; we are very thankful for his remark. The image processing was done on a hardware/software environment developed by Keith Nishihara and Noble Larson.

References

Bolles, R. C., and Cain, R. A. 1982. Recognizing and locating partially visible objects: The Local-Feature-Focus method. *Int. J. Robotics Res.* 1(3):57–82.

Bolles, R. C., Horaud, P., and Hannah, M. J. 1983. 3DPO: A three-dimensional part orientation system. *Robotics Research: The First International Symposium*, edited by M. Brady and R. Paul, MIT Press, 1984, pp. 413–424.

Drumheller, M. 1984. Robot Localization Using Range Data. S. B. Thesis, Dept. of Mechanical Engineering, MIT.

Faugeras, O. D., and Hebert, M. 1983 (Aug. Karlsruhe, W. Germany). A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces. *Proc. Eighth Int. Joint Conf. Artificial Intell.* Los Altos: William Kaufmann, pp. 996–1002.

Gaston, P. C., and Lozano-Pérez, T. 1984. Tactile recognition and localization using object models: The case of polyhedra on a plane. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-6(3):257–265.

Grimson, W. E. L., 1984. The combinatorics of local constraints in model-based recognition and localization from sparse data. A. I. Memo 763, Massachusetts Institute of TechnologyArtificial Intelligence Lab., Cambridge, Mass.

Grimson, W. E. L., and Lozano-Pérez, T. 1984. Model-based recognition and localization from sparse range or tactile data. *Int. J. Robotics Res.* 3(3):3–35.

Stockman, G., and Esteva, J. C. 1984. Use of geometrical constraints and clustering to determine 3D object pose. TR84-002, Michigan State University Department of Computer Science, East Lansing, Mich.