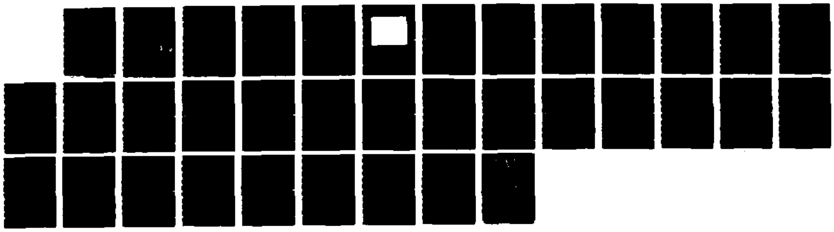
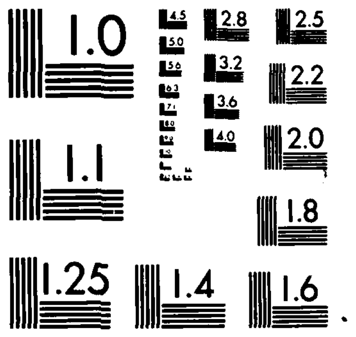
AD-A184 441   ON THE RECOGNITION OF CURVED OBJECTS(U) MASSACHUSETTS   1/1
INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB
W E GRIMSON JUL 87 AI-M-983 N00014-86-K-0685
UNCLASSIFIED   F/G 12/9   NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A184 441 DTIC FILE COPY

(12)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AI Memo 983	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) On the Recognition of Curved Objects		5. TYPE OF REPORT & PERIOD COVERED memorandum
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) W. Eric L. Grimson		8. CONTRACT OR GRANT NUMBER(s) N00014-86-K-0685 N00014-85-K-0124
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory 545 Technology Square Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE July 1987 13. NUMBER OF PAGES 31
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Unlimited

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Object recognition
Constrained search

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Determining the identity and pose of occluded objects from noisy data is a critical part of a system's intelligent interaction with an unstructured environment. Previous work has shown that local measurements of the position and surface orientation of small patches of an object's surface may be used in a constained search process to solve this problem, for the case of rigid polygonal objects using two-dimensional sensory data, or rigid polyhedral objects using three-dimensional data. This note extends the recognition system to deal with the problem of recognizing and locating curved objects. The extension is done in two

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

87 9 8 075

Block 20 cont.

dimensions, and applies to the recognition of two-dimensional objects from two-dimensional data, or to the recognition of three-dimensional objects in stable positions from two-dimensional data.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 983 July, 1987

On the Recognition of Curved Objects

W. Eric L. Grimson

Abstract. Determining the identity and pose of occluded objects from noisy data is a critical part of a system's intelligent interaction with an unstructured environment. Previous work has shown that local measurements of the position and surface orientation of small patches of an object's surface may be used in a constrained search process to solve this problem, for the case of rigid polygonal objects using two-dimensional sensory data, or rigid polyhedral objects using three-dimensional data. This note extends the recognition system to deal with the problem of recognizing and locating curved objects. The extension is done in two dimensions, and applies to the recognition of two-dimensional objects from two-dimensional data, or to the recognition of three-dimensional objects in stable positions from two-dimensional data.

Acknowledgements: 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 the System Development Foundation, in part by an Office of Naval Research University Research Initiative grant under contract N00014-86-K-0685, and in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124.

1. Introduction

The general problem considered in this note is how to locate a known object from sensory data, especially when that object may be occluded by other (possibly unknown) objects. In previous work [Grimson and Lozano-Pérez 84, 87] we described a recognition system, called **RAF** (for Recognition and Attitude Finder), that identifies and locates objects from noisy, occluded data. In that work, we concentrated on a particular subclass of rigid models. If the sensory data provided two-dimensional geometric data, for example intensity edges from a visual image, we considered the recognition of objects that consisted of sets of linear segments, or equivalently, polygonal objects in which some edges are not included. If the sensory data was three-dimensional, we considered the recognition of objects that consisted of sets of planar fragments, or equivalently, polyhedral objects in which some of the faces are not included.

In general, of course, we cannot guarantee that the recognition system will be confronted only with polyhedral objects. Since the **RAF** system is reasonably insensitive to noise, one could deal with curved objects by simply approximating them with polyhedral models that are required to deviate from the actual object by no more than some bounded amount. This has the effect of introducing some additional error into the process, which the system has been able to tolerate. While the **RAF** system has been successfully tested on a range of real data, including visual, laser, sonar and tactile, using approximations to curved objects as well as polyhedral ones [Grimson and Lozano-Pérez 87], the assumption of polyhedral models is overly restrictive.

In particular, one of the difficulties with using polyhedral approximations is that they are not stable, so that several images of the same object may lead to different approximations, due to small variations in the imaging. This may lead to difficulties in matching, either causing incorrect matches or removing large portions of an object from consideration. Moreover, systematic errors in the approximation can have serious effects on the recognition process. Consider an object with a circular hole, which is approximated in the model by a regular polygon. Now suppose that we take an image of the part in some other orientation, and extract a polygonal approximation of the visible edges of the object. The boundary of the hole will again be approximated by a regular polygon. If, however, the approximations are rotated relative to one another, this can lead to a drastic error in locating the part, since matching the two descriptions will lead to a large error in the orientation of the overall part.

In this note, we consider the problem of extending the method to deal directly with two dimensional objects that include linear and curved segments, where the curved segments can be approximated by circular arcs. To describe this extension to **RAF**, we must specify the characteristics of the object models, the requirements on the sensory data, and the search technique used to correctly identify the object model from the sensory data. Our goal is to obtain a system that can perform as indicated in Figure 1.

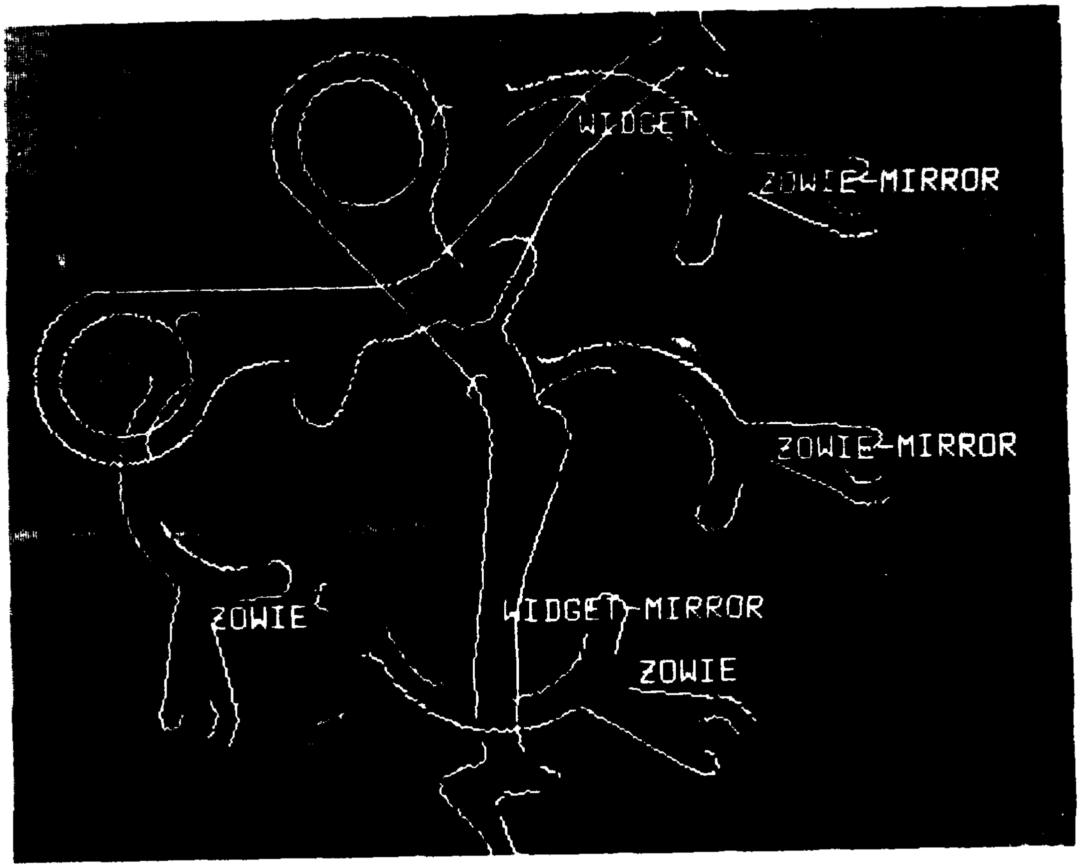


Figure 1. An example of recognizing curved objects. A grey level image of a set of overlapping parts has been processed. Each part has been identified by name, and the position and orientation (pose) of the part has been identified. This is shown in the figure by overlaying the computed pose of the object model, together with the identified name.

2. Recognition as constrained search

2.1 Definition of a solution

Our definition of the recognition problem can be simply stated. We are given a set of data fragments, obtained from the boundary of an object or objects, and measured in a coordinate system centered about the sensor. We are also given a set of object models, specified by a set of faces (whose definition we will make formal shortly) measured in a local coordinate frame specific to the model. A solution to the recognition problem consists of determining the following components: a subset of the sensory data fragments that are believed to come from a single object; an identification of which object, selected from the library of known objects; the model face associated with each data fragment in the subset; and finally the coordinate frame transformation that maps the model from its local coordinate frame into the sensor coordinate frame in such a manner that each data fragment correctly lies on its assigned model face. In more formal terms, a solution is a three-tuple

$$\langle \texttt{object}_i, \{(d_{i_1}, m_{j_1}), (d_{i_2}, m_{j_2}), \ldots (d_{i_k}, m_{j_k})\}, (R, \mathbf{v}_0, s)\rangle$$

where $\texttt{object}_i$ is the name of the i^{th} object in the library, the d, m pairings are associations of a subset of the sensory data d with model faces m from $\texttt{object}_i$ and R is a rotation matrix, $\mathbf{v}_0$ is a translation vector and s is a scale factor such that a vector $\mathbf{v}_m$ in model coordinates is transformed into a vector $\mathbf{v}_d$ in sensor coordinates by

$$\mathbf{v}_d = sR\mathbf{v}_m + \mathbf{t}.$$

As has been described elsewhere [Grimson and Lozano-Pérez 84, 87], we approach the recognition problem as one of search. Thus, we first focus on finding legitimate pairings of data and model fragments, for some subset of the sensory data. We chose to structure this search process as a constrained depth first search, using an ***interpretation tree*** (IT). Each node of the tree describes a partial interpretation of the data, and implicitly contains a set of pairings of data fragments and model faces. Nodes at the first level of the tree contain assignments for the first data fragment, nodes at the second level contain assignments for the first and second data fragments, and so on. Each node branches at the next level in up to $n + 1$ ways, where n is the number of model faces in the object. The last branch is a ***wild card*** or ***null*** branch and has the effect of excluding the data fragment corresponding to the current level of the tree from part of the interpretation.

Given s data fragments, any leaf of the tree specifies an interpretation

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_s, m_{j_s})\},$$

where some of the m_{j_k} may be the wild card character. By excluding such matches, the leaf yields a partial interpretation

$$\{(d_{i_1}, m_{j_1}), (d_{i_2}, m_{j_2}), \ldots (d_{i_k}, m_{j_k})\}$$

where $1 \leq i_1 < i_2 < \ldots < i_k$ but these indices may not include the entire set from 1 to s. This interpretation may then be used to solve for a rigid, scaled transformation that maps model faces into corresponding data fragments, if such a transformation exists. Thus, by searching for leaves of the tree and testing that the interpretation there yields a legal transformation, we can find possible instances of object models in the data. The process is shown in Figure 2.

Since this search process is inherently an exponential problem, the key to an efficient solution is to use constraints to remove large subtrees from consideration without having explicitly to explore them. We next consider the explicit form of the

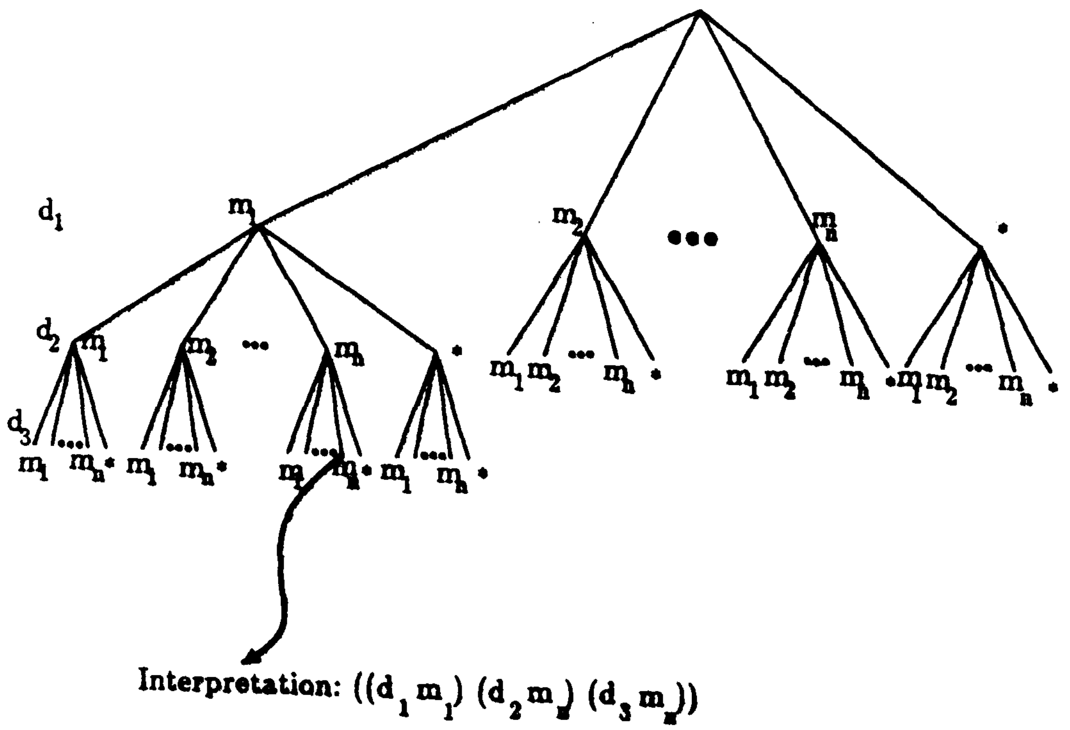


Figure 2. An Interpretation Tree. Each node of the tree defines a partial interpretation, where the level of each ancestor defines a sensory data point, and the branch leading to each such node defines the corresponding model face. An example of a partial interpretation is shown, where d_i denotes the ith data point and m_k denotes the kth model face.

sensory data, and the explicit form of the model faces, and then consider constraints on the assignment of one to the other that can be used to restrict the search process.

2.2 Object models and sensory data

In this note, we restrict ourselves to two-dimensional, or laminar, objects, although much of the work has been extended to three dimensions [Grimson and Lozano-Pérez 84, 87]. We allow our two-dimensional object models to consist of two different types of components.

The first type of component is a **linear edge fragment**, consisting of two endpoints, and a unit vector normal to the line between them, and pointing away from the interior of the object. Formally, this is given by

$$\mathbf{linear}_i = (\hat{\mathbf{n}}_i, (\mathbf{b}_i, \mathbf{e}_i)) .$$

Note that a point on the edge can be represented by

$$\hat{\mathbf{n}}_i \qquad \text{and} \qquad \mathbf{b}_i + \alpha_i \hat{\mathbf{t}}_i, \quad \alpha_i \in [0, \ell_i]$$

where $\hat{\mathbf{n}}_i$ is the unit normal vector, $\hat{\mathbf{t}}_i$ is a unit tangent vector, oriented so that it points from **b** to **e**, and α_i can vary from 0 to the length of the edge ℓ_i (see Figure 3).

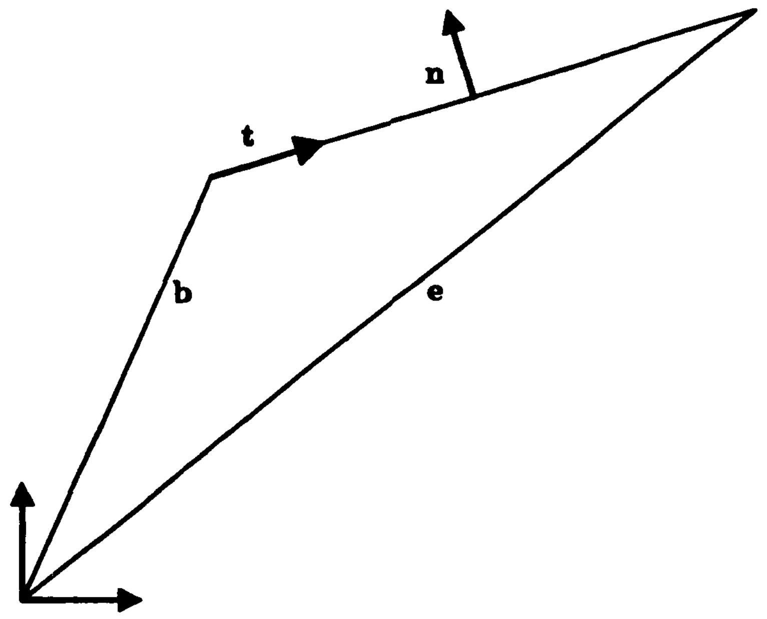


Figure 3. The representation of an edge. An edge is given by the pair

$$\hat{\mathbf{n}}_i \quad \text{and} \quad \mathbf{b}_i + \alpha_i \hat{\mathbf{t}}_i, \quad \alpha_i \in [0, \ell_i]$$

where $\hat{\mathbf{n}}_i$ is a unit normal vector, $\hat{\mathbf{t}}_i$ is a unit tangent vector, oriented so that it points from **b** to **e**, $\mathbf{b}_i$ is a vector to the base point of the edge, and α_i can vary from 0 to the length of the edge ℓ_i.

The second type of component is a **circular arc**, consisting of a center, a radius, a pointing direction, and a range of angles, measured relative to the x axis. Formally, this is given by

$$\mathtt{circular}_i = (\mathbf{c}_i, r_i, d_i, (\phi_i, \psi_i)).$$

The pointing direction, if known, is an indication of which side of the circular arc is the interior of the object. Specifically, it indicates whether the circular arc is the boundary of a hole, or if it is on the exterior of the object. An example of a circular segment is shown in Figure 4.

We assume that both the sensory data fragments and the object models are composed of such components. We will discuss shortly how to obtain such fragments from grey level images.

2.3 Constraints between object models and sensory data

Given such simple fragments, we now consider how to use them to reduce the search

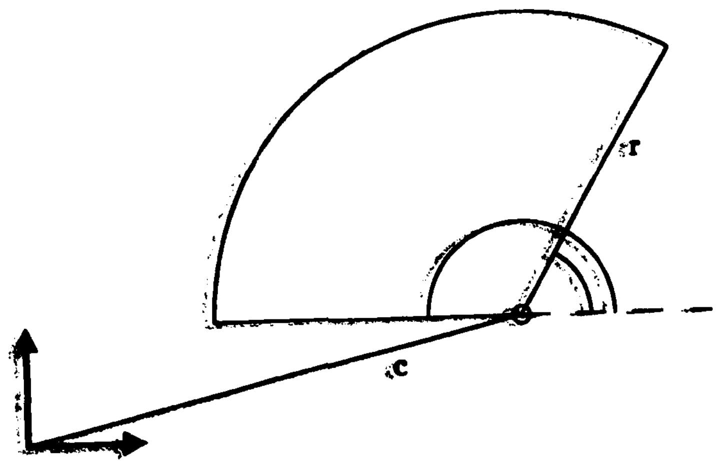


Figure 4. The representation of a circular arc. An arc is defined by a center $\mathbf{c}_i$, a radius r_i, a pointing direction d_i, and a range of angles (ϕ_i, ψ_i).

process. We consider both unary and binary constraints. Since the transformation from model to sensor coordinates is one of the things to be determined, we need constraints that can compare coordinate frame independent measurements from sensory and model fragments.

2.3.1 *Unary constraints*

Length constraint

Consider a linear data fragment,

$$\mathbf{linear}_i = (\hat{\mathbf{n}}_i, (\mathbf{b}_i, \mathbf{e}_i))$$

and a possible matching model fragment

$$\mathbf{LINEAR}_p = \left(\hat{\mathbf{N}}_p, (\mathbf{B}_p, \mathbf{E}_p)\right).$$

We let ℓ_i denote the length of the data fragment, and L_p denote the corresponding length of the model fragment, where these lengths are given by

$$\ell_i = |\mathbf{b}_i - \mathbf{e}_i|, \qquad L_p = |\mathbf{B}_p - \mathbf{E}_p|.$$

We let

$$\textbf{length-constraint}(i,p) = \text{True iff } \ell_i \leq L_p + \epsilon_L$$

capture the notion of the unary length constraint, where ϵ_L is a predefined upper bound on the amount of error inherent in measuring the length of an edge.

This constraint says that if the length of the i^{th} linear data fragment is less than the length of the p^{th} linear model fragment, subject to some bounded error, then it is possible to consistently assign this data fragment to lie on this model one.

Note that there is only an upper inequality in determining consistency, since a data fragment could be partially occluded.

Radius, swept angle and pointing constraints

Now consider a circular data fragment,

$$\texttt{circular}_i \quad (c_i, r_i, d_i, (\phi_i, \psi_i))$$

and a possible matching model fragment

$$\texttt{CIRCULAR}_p = (\mathbf{C}_p, R_p, D_p, (\Phi_p, \Psi_p)) .$$

We can define three unary constraints in this case. We let

$$\textbf{radius-constraint}(i,p) = \text{True iff } |r_i - R_p| \leq \epsilon_r$$

$$\textbf{swept-angle-constraint}(i,p) = \text{True iff } \psi_i - \phi_i \leq \Psi_p - \Phi_p + \epsilon_c$$

$$\textbf{pointing-constraint}(i,p) = \text{True iff } d_i \text{ and } D_p \text{ are known and identical.}$$

Here, ϵ_r is a predefined upper bound on the amount of error inherent in measuring the radius of a circular arc, and ϵ_c is an upper bound on the amount of error inherent in measuring the angular extent of a circular arc.

These constraints say that if the radii of the i^{th} data fragment and the p^{th} model agree to within some error, their ranges of swept angles agree to within some error, and they are both either interior or exterior arcs, then it is possible to consistently assign this data fragment to lie on this model one.

2.3.2 Binary constraints

Consider two linear data fragments,

$$\texttt{linear}_i = (\hat{\mathbf{n}}_i, (\mathbf{b}_i, \mathbf{e}_i)) \qquad \texttt{linear}_j = (\hat{\mathbf{n}}_j, (\mathbf{b}_j, \mathbf{e}_j))$$

and two possible matching model fragments

$$\texttt{LINEAR}_p = \left(\hat{\mathbf{N}}_p, (\mathbf{B}_p, \mathbf{E}_p)\right) \qquad \texttt{LINEAR}_q = \left(\hat{\mathbf{N}}_q, (\mathbf{B}_q, \mathbf{E}_q)\right) .$$

We need to derive a set of constraints that will determine the consistency of assigning the data fragments to lie on the model ones.

Angle constraint

Let θ_{ij} denote the angle between $\hat{\mathbf{n}}_i$ and $\hat{\mathbf{n}}_j$, and let Θ_{pq} denote the angle between $\hat{\mathbf{N}}_p$ and $\hat{\mathbf{N}}_q$. We let

$$\textbf{binary-angle-constraint}(i,j,p,q) = \text{True iff } \theta_{ij} \in [\Theta_{pq} - 2\epsilon_a, \Theta_{pq} + 2\epsilon_a]$$

where all arithmetic comparisons are performed modulo 2π and where ϵ_a is an upper bound on the amount of error inherent in determing the direction of a normal.

This says that if the angle between the data normals agrees with the angle between the model normals, within some error, then it is possible to consistently assign these data fragments to lie on these model ones.

Distance constraint

Given two data fragments, there is a range of distances associated with the family of vectors having tail on one edge and head on the other. We can compute the range of such distances, denoted by $[d_{\ell,ij}, d_{h,ij}]$, in a straightforward manner. If $i = j$, then the minimum distance is 0 and the maximum distance is the length of the edge. In the more general case, let $\rho(\mathbf{v}, \mathbf{u})$ denote the distance between two points. Then the maximum distance is given by

$$d_{h,ij} = \max\{\rho(\mathbf{v},\mathbf{u}) | \mathbf{v} \in \{\mathbf{b}_i, \mathbf{e}_i\}, \mathbf{u} \in \{\mathbf{b}_j, \mathbf{e}_j\}\}.$$

For the minimum distance, we must also consider the possibility that the projection from an endpoint of one edge in the direction of the normal of the second edge intersects that edge, so that

$$\begin{aligned} d_{\ell,ij} = \min\{ & \{\rho(\mathbf{v},\mathbf{u}) | \mathbf{v} \in \{\mathbf{b}_i, \mathbf{e}_i\}, \mathbf{u} \in \{\mathbf{b}_j, \mathbf{e}_j\}\} \\ & \cup \{\rho(\mathbf{v}, \mathbf{b}_i + \langle \mathbf{v} - \mathbf{b}_i, \hat{\mathbf{t}}_i \rangle \hat{\mathbf{t}}_i) | \mathbf{v} \in \{\mathbf{b}_j, \mathbf{e}_j\}, \langle \mathbf{v} - \mathbf{b}_i, \hat{\mathbf{t}}_i \rangle \in [0, \ell_i]\} \\ & \cup \{\rho(\mathbf{v}, \mathbf{b}_j + \langle \mathbf{v} - \mathbf{b}_j, \hat{\mathbf{t}}_j \rangle \hat{\mathbf{t}}_j) | \mathbf{v} \in \{\mathbf{b}_i, \mathbf{e}_i\}, \langle \mathbf{v} - \mathbf{b}_j, \hat{\mathbf{t}}_j \rangle \in [0, \ell_j]\}\} \end{aligned}$$

where we let $< .,. >$ denote the dot (or inner) product of two vectors. For the model fragments, we can compute similar ranges, which we denote by $D_{\ell,pq}, D_{h,pq}$. We let

distance-constraint(i, j, p, q) = True iff $[d_{\ell,ij}, d_{h,ij}] \subseteq [D_{\ell,pq} - 2\epsilon_p, D_{h,pq} + 2\epsilon_p]$

where we assume that the position of an edge point is known to within an error bound ϵ_p.

This says that if the range of distances between the data edges is contained within the range of distances between the model edges, subject to some error, then it is possible to consistently assign these data fragments to lie on these model ones.

Component constraint

The third constraint concerns the separation of the two edge fragments. In particular, we consider the range of components of a vector between the two edge fragments, in the direction of each of the edge normals. Algebraically, this is expressed by the dot product

$$\langle \mathbf{b}_i + \alpha_i \hat{\mathbf{t}}_i - \mathbf{b}_j - \alpha_j \hat{\mathbf{t}}_j, \hat{\mathbf{n}}_i \rangle$$

which reduces to

$$\langle \mathbf{b}_i - \mathbf{b}_j, \hat{\mathbf{n}}_i \rangle - \alpha_j \langle \hat{\mathbf{t}}_j, \hat{\mathbf{n}}_i \rangle \qquad \alpha_j \in [0, \ell_j]$$

Of course, there is an equivalent constraint for components in the direction of $\hat{\mathbf{n}}_j$. Note that this expression actually determines a range of values, with extrema when $\alpha_j = 0, \ell_j$. We denote this by

$$d_{\ell,ij} = \min\{\langle \mathbf{b}_i - \mathbf{b}_j, \hat{\mathbf{n}}_i \rangle - \alpha_j \langle \hat{\mathbf{t}}_j, \hat{\mathbf{n}}_i \rangle | \alpha_j \in \{0, \ell_j\}\}$$
$$d_{h,ij} = \min\{\langle \mathbf{b}_i - \mathbf{b}_j, \hat{\mathbf{n}}_i \rangle - \alpha_j \langle \hat{\mathbf{t}}_j, \hat{\mathbf{n}}_i \rangle | \alpha_j \in \{0, \ell_j\}\}$$

These ranges can be computed both for pairs of data edges and pairs of model edges. In the ideal case, consistency will hold only if the data range is contained within the model range (since the data edges may correspond to only parts of the model edges). As in the case of the other constraints, we also need to account for error in the measurements. We derive a simple method for doing this below.

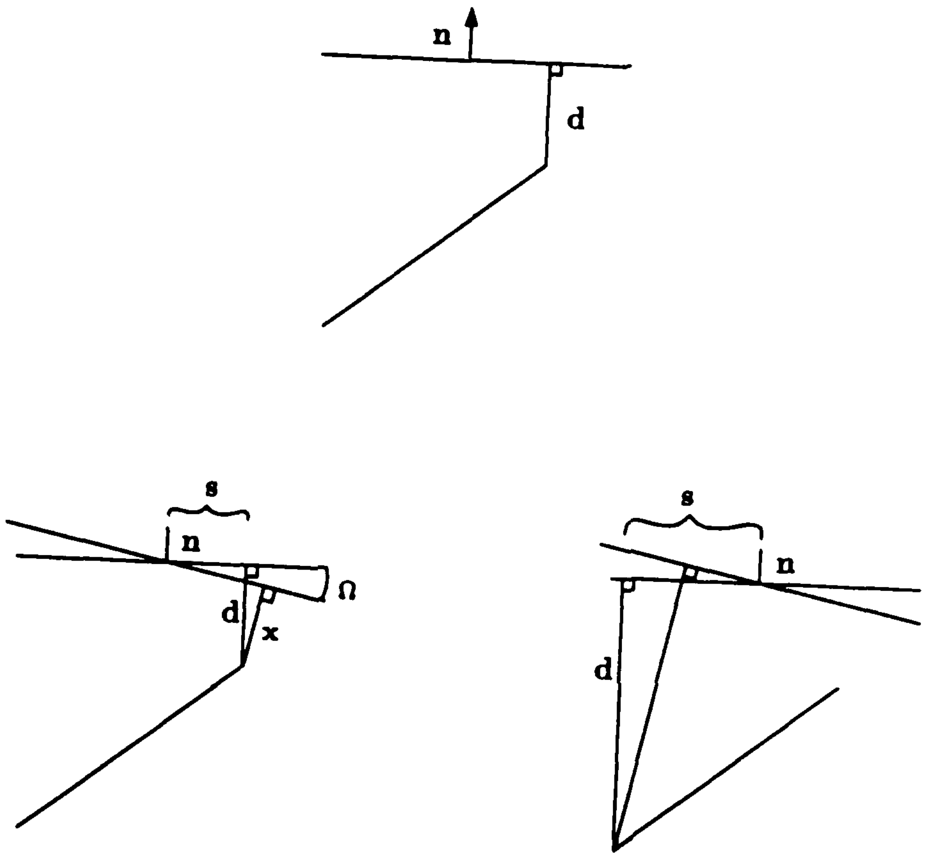


Figure 5. Errors in computing the direction constraint. (a) The component of a vector from one endpoint in the direction of the other edge's normal is given by the perpendicular distance **d** to the extended edge. (b) Since the actual normal is only accurate to within ϵ_a, one extreme case is given by rotating the extended edge about its midpoint by that amount and finding the new perpendicular distance. (c) The other extreme is obtained by considering the other endpoint.

Consider the base case, shown in Figure 5a. The perpendicular distance from the endpoint of one edge to the other edge is shown as $D^{\perp}$. In Figure 5b, the edge is rotated by ϵ_a about its midpoint, and the new perpendicular distance X is shown. We need to relate X to measurable values. We already have $D^{\perp}$. We can also measure S, the distance from the midpoint of the edge to the perpendicular dropped from the endpoint of the other edge, as shown. Straightforward trigonometry then yields the new distance

$$X = (D^{\perp} - S \sin \epsilon_a) \cos \epsilon_a.$$

Since the position of the second edge is not known exactly, we must adjust this expression, to yield one limit on the range of possible measurements:

$$D^{\perp}_{\ell,pq} = (D^{\perp} - S\sin\epsilon_a)\cos\epsilon_a - \epsilon_p.$$

The other extreme is shown in Figure 5c. Trigonometric manipulation yields the following upper bound

$$D^{\perp}_{h,pq} = (S - D^{\perp}\sin\epsilon_a)\sin\epsilon_a + D^{\perp}\sec\epsilon_a + \epsilon_p.$$

Thus, given two model edges indexed by p, q, we can compute a range of possible measurements (modulo known error bounds), by using $D^{\perp}_{\ell,pq}$ and $D^{\perp}_{h,pq}$ computed over all the endpoints of the edges. We denote this range by $[M^{\perp}_{\ell,pq}, M^{\perp}_{h,pq}]$.

We let

component-constraint(i,j,p,q) = True iff $[d^{\perp}_{\ell,ij}, d^{\perp}_{h,ij}] \subseteq [M^{\perp}_{\ell,pq}, M^{\perp}_{h,pq}]$.

This says that if the range of distance components between the data edges is contained within the corresponding range between the model edges, subject to some error, then it is possible to consistently assign these data fragments to lie on these model ones.

Circle center constraint

Now consider two circular data fragments,

$$\texttt{circular}_i = (\mathbf{c}_i, r_i, d_i, (\phi_i, \psi_i)) \qquad \texttt{circular}_j = (\mathbf{c}_j, r_j, d_j, (\phi_j, \psi_j))$$

and two possible matching model fragments

$$\texttt{CIRCULAR}_p = (\mathbf{C}_p, R_p, D_p, (\Phi_p, \Psi_p)) \qquad \texttt{CIRCULAR}_q = (\mathbf{C}_q, R_q, D_q, (\Phi_q, \Psi_q)).$$

We need to derive a set of constraints that will determine the consistency of assigning the data fragments to lie on the model ones.

The first constraint arises from considering the distance between the centers of the two circles, given by $\rho(\mathbf{c}_i, \mathbf{c}_j)$, in the case of the two data fragments. We let

center-constraint(i,j,p,q) = True iff $\rho(\mathbf{c}_i, \mathbf{c}_j) \in [\rho(\mathbf{C}_p, \mathbf{C}_q) - 2\epsilon_{cd}, \rho(\mathbf{C}_p, \mathbf{C}_q) + 2\epsilon_{cd}]$

where ϵ_{cd} is an upper bound on the amount of error inherent in measuring the position of the center of a circular arc.

This says that if the distance between the centers of two data arcs is within some bounded error of the distance between the centers of the model arcs, then it is possible to consistently assign these data fragments to lie on these model ones.

Circle swept angle constraint

The ranges of swept angles of two fragments must also be constrained. We let

binary-swept-angle-constraint(i,j,p,q) = True iff $\phi_i - \psi_j \le \Phi_p - \Psi_q + 2\epsilon_a$

and $\psi_i - \phi_j \ge \Psi_p - \Phi_q - 2\epsilon_a$

where as before ϵ_a is the amount of error inherent in measuring the angular extent of a circular arc, and the angles are measured modulo 2π.

This says that if the difference in the range of swept angles between two data arcs is contained within the corresponding range of two model arcs, subject to some error, then it is possible to consistently assign these data fragments to lie on these model ones.

2.3.3 Cross type constraints

All of the binary constraints described above deal with the relationship between pairs of data fragments of the same type, and corresponding pairs of model fragments. Note that there are other constraints possible, especially cross constraints between fragments of opposite type.

Cross distance constraint

Consider a circular data fragment

$$\texttt{circular}_i = (\mathbf{c}_i, r_i, d_i, (\phi_i, \psi_i))$$

and a linear data fragment

$$\texttt{linear}_j \quad (\hat{\mathbf{n}}_j, (\mathbf{b}_j, \mathbf{e}_j))$$

and corresponding model fragments

$$\texttt{CIRCULAR}_p = (\mathbf{C}_p, R_p, D_p, (\Phi_p, \Psi_p)) \qquad \texttt{LINEAR}_q = \left(\hat{\mathbf{N}}_q, (\mathbf{B}_q, \mathbf{E}_q)\right).$$

The range of distances between the center of a circular fragment and a linear fragment can be constrained, similar to the distance constraint between two linear fragments. We let

$$\begin{aligned} F_{\ell,pq} &= \rho(\mathbf{C}_p, \mathbf{B}_q + \left\langle \mathbf{C}_p - \mathbf{B}_q, \hat{\mathbf{T}}_q \right\rangle \hat{\mathbf{T}}_q) \qquad \text{if } \left\langle \mathbf{C}_p - \mathbf{B}_q, \hat{\mathbf{T}}_q \right\rangle \in [0, L_q] \\ &= \min\{\rho(\mathbf{C}_p, \mathbf{B}_q), \rho(\mathbf{C}_p, \mathbf{E}_q)\} \qquad \text{otherwise} \\ F_{h,pq} &= \max\{\rho(\mathbf{C}_p, \mathbf{B}_q), \rho(\mathbf{C}_p, \mathbf{E}_q)\} \end{aligned}$$

and we let

$$\begin{aligned} \textbf{cross-distance}(i, j, p, q) = \text{ True iff } & \rho(\mathbf{c}_i, \mathbf{b}_j) \in [F_{\ell,pq} - \epsilon_{cd} - \epsilon_p, F_{h,pq} + \epsilon_{cd} + \epsilon_p] \\ \text{and } & \rho(\mathbf{c}_i, \mathbf{e}_j) \in [F_{\ell,pq} - \epsilon_{cd} - \epsilon_p, F_{h,pq} + \epsilon_{cd} + \epsilon_p]. \end{aligned}$$

This says that if the range of distances from the center of the data circle to the data edge is contained within the corresponding model range, subject to error, then it is possible to consistently assign these data fragments to lie on these model ones.

Cross component constraint

Given the same data and model fragments as above, we can consider the perpendicular distance from the circle center to the extended line defined by the linear segment.

Using the same method described for the component constraint between two linear fragments, we let

$$D^{\perp}_{\ell,pq} = (D^{\perp} - S\sin\epsilon_a)\cos\epsilon_a - \epsilon_p$$
$$D^{\perp}_{h,pq} = (S - D^{\perp}\sin\epsilon_a)\sin\epsilon_a + D^{-}\sec\epsilon_a + \epsilon_p$$

where in this case

$$D^{-} = \left\langle \mathbf{B}_q - \mathbf{C}_p, \hat{\mathbf{N}}_q \right\rangle .$$

We let

cross-component(i,j,p,q) = True iff $\langle \mathbf{b}_j - \mathbf{c}_i, \hat{\mathbf{n}}_j \rangle \in [D^{\perp}_{\ell,pq}, D^{\perp}_{h,pq}]$.

This says that if the component of distance from the center of the data circle to the data edge is contained within the corresponding model range, subject to error, then it is possible to consistently assign these data fragments to lie on these model ones.

Cross angle constraint

Let γ_j, Γ_q be the angle between the unit normal vector of the linear edge and the x axis, for the data and model linear fragments respectively. Then the range of angles between the swept angles of the circular fragment and the unit normal must be consistent.

We let

cross-angle(i,j,p,q) = True iff $[\phi_i - \gamma_j, \psi_i - \gamma_j] \subset [\Phi_i - \Gamma_j - \epsilon_c, \Psi_i - \Gamma_j + \epsilon_c]$.

2.4 The constraints reduce the search

Given these unary and binary constraints, the constrained search process can be straightforwardly specified. Suppose the search process is currently at some node at level k in the interpretation tree and with a consistent partial interpretation given by

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_k, m_{j_k})\} .$$

We now consider the next data fragment d_{k+1}, and its possible assignment to model face $m_{j_{k+1}}$, where j_{k+1} varies from 1 to $n+1$.

The following rules hold.

- If $m_{j_{k+1}}$ is the wild card match, then the new interpretation

  $$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_{k+1}, m_{j_{k+1}})\}$$

  is consistent, and we continue downward in our search.
- If $m_{j_{k+1}}$ is a linear edge segment, we must verify that

  length-constraint$(k+1, j_{k+1})$ = True.

  Moreover, for all $i \in \{1, \ldots, k\}$ such that d_i is a linear edge fragment, we must verify that

  binary-angle-constraint$(i, k+1, j_i, j_{k+1})$ = True

$$\textbf{distance-constraint}(i, k+1, j_i, j_{k+1}) = \text{True}$$
$$\textbf{component-constraint}(i, k+1, j_i, j_{k+1}) = \text{True}$$
$$\textbf{component-constraint}(k+1, i, j_{k+1}, j_i) = \text{True}.$$

And, for all $i \in \{1, \ldots, k\}$ such that d_i is a circular edge fragment, we must verify that

$$\textbf{cross-distance}(i, k+1, j_i, j_{k+1}) = \text{True}$$
$$\textbf{cross-component}(i, k+1, j_i, j_{k+1}) = \text{True}$$
$$\textbf{cross-angle}(i, k+1, j_i, j_{k+1}) = \text{True}.$$

- If $m_{j_{k+1}}$ is a circular arc segment, we must verify that

$$\textbf{radius-constraint}(k+1, j_{k+1}) = \text{True}$$
$$\textbf{pointing-constraint}(k+1, j_{k+1}) = \text{True}$$
$$\textbf{swept-angle-constraint}(k+1, j_{k+1}) = \text{True}.$$

Moreover, for all $i \in \{1, \ldots, k\}$ such that d_i is a circular arc fragment, we must verify that

$$\textbf{binary-swept-angle-constraint}(i, k+1, j_i, j_{k+1}) = \text{True}$$
$$\textbf{binary-swept-angle-constraint}(k+1, i, j_{k+1}, j_i) = \text{True}$$
$$\textbf{center-constraint}(i, k+1, j_i, j_{k+1}) = \text{True}.$$

And, for all $i \in \{1, \ldots, k\}$ such that d_i is a linear edge fragment, we must verify that

$$\textbf{cross-distance}(k+1, i, j_{k+1} j_i,) = \text{True}$$
$$\textbf{cross-component}(k+1, i, j_{k+1} j_i,) = \text{True}$$
$$\textbf{cross-angle}(k+1, i, j_{k+1} j_i,) = \text{True}.$$

- If all of these constraints are true, then

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_{k+1}, m_{j_{k+1}})\}$$

is a consistent partial interpretation, and we continue our depth first search. If one of them is false, then the partial interpretation is inconsistent. In this case, we increment the model face index j_{k+1} by 1 and try again, until $j_{k+1} = n + 1$.

If the search process is currently at some node at level k in the interpretation tree, and has an inconsistent partial interpretation given by

$$\{(d_1, m_{j_1}), (d_2, m_{j_2}), \ldots (d_k, m_{j_k})\}$$

then it is in the process of backtracking. If $j_k = n + 1$ (the wild card) we backtrack up another level, otherwise we increment j_k and continue.

2.5 Model tests

Once the search process reaches a leaf of the interpretation tree, we have accounted for all of the data points. We are now ready to determine if the interpretation is in fact globally valid. To do this, we solve for a rigid transformation mapping points $\mathbf{v}_m$ in model coordinates into points $\mathbf{v}_l$ in sensor coordinates,

$$\mathbf{v}_l = sR\mathbf{v}_m + \mathbf{v}_0$$

where R is a rotation matrix, $\mathbf{v}_0$ is a translation vector, and s is a scale factor. We can solve for this transformation in a number of ways [e.g. Grimson and Lozano-Pérez 84, 87, Ayache and Faugeras 86]. The method described in [Ayache and Faugeras 86] deals with finding transformations from line segments to line segments. It is straightforward to extend the method to deal with transformations of sets of points (the centers of the circular arcs) to sets of points. In our implementation (described later) we solve for two transformations, one based on the linear fragments of the match, and one based on the circular fragments. We then require that the two transformations be roughly identical.

Given such a transformation, which is usually some type of least squares fit, we must then ensure that the interpretation actually satisfies it. We do this by considering each of the linear data fragments associated with a real model face in the interpretation, and transforming the associated linear model face by the computed transform. For each such face, we then verify that the transformed fragment differs in position and orientation from its associated data fragment by amounts that are less than some acceptable error bounds. These bounds on transform error can be obtained from the predefined bounds on the sensor error [Grimson 86b]. For each of the circular data fragments associated with a real model face in the interpretation, we also transform the model fragment into sensor coordinates. In this case, we both verify that the transformed center of the circular arc lies within some bounded error of the center of the associated data fragment, and that the set of actual points lying on the data circular arc, are within a bounded distance of some point among the set of transformed model points. Any interpretation that passes such a model test is a consistent interpretation of the data.

2.6 Additional search reductions

While the constrained search technique described above will succeed in finding all consistent interpretations of the sensory data, for a given object model, it is not particularly computationally efficient. This is mostly due to the problem of segmenting the data to determine subsets that belong to a single object. Indeed, for the case of linear fragments only, if all of the sensory data do belong to one object, the described method is known to be quite efficient, as has been verified both empirically [Grimson and Lozano-Pérez 84, 87] and theoretically [Grimson 1986a]. In order to improve the efficiency of the method, we add two additional methods to our search process, both previously discussed for the case of linear fragments in [Grimson and Lozano-Pérez 87], and extended here to circular segments.

Hough transforms

The first is to use the Hough transform [Hough 62, Merlin and Farber 75, Sklansky 78, Ballard 81] as a preprocessor to restrict our attention to small portions of the search space. In brief, the Hough transform works as follows. Consider a three dimensional configuration space, with axes denoting the x and y components of a translation vector, along with a θ axis which denotes an angle of rotation. Thus, any point in this space defines a unique rigid transformation. We tesselate the space into buckets, based on some sampling of the axes, say h_x, h_y, h_θ.

Now, we consider a linear data edge, i and a linear model fragment, indexed by q. Suppose that the data edge is shorter than the model edge, subject to the error in measuring length ϵ_L. There is a unique angle, call it θ', needed in order to rotate the model fragment so that its normal aligns with the data fragment's normal. Let $R_{\theta'}$ denote the rotation matrix associated with this rotation angle. Ignoring for the moment the effects of scaling, any point on the model edge can be transformed into sensor coordinates, as

$$\mathbf{u} = R^{-1}[\mathbf{B}_q + \beta\hat{\mathbf{T}}_q - \mathbf{v}_0].$$

Any translation $\mathbf{v}_0$ such that the endpoints of the transformed line lie within range of the associated data line are possible valid translations. Since the data edge may be partially occluded, it will in general be shorter than the model edge and hence there will be a range of translations possible, corresponding to sliding the shorter edge along the longer one.

These translations are given by the set of $\mathbf{v}_0$ that satisfy

$$\langle \mathbf{u} - \mathbf{b}_i, \hat{\mathbf{n}}_i \rangle \in [-\epsilon_p, \epsilon_p]$$
$$\langle \mathbf{u} - \mathbf{b}_i, \hat{\mathbf{t}}_i \rangle \in [-\epsilon_L, \ell_i + \epsilon_L]$$

for all $\beta \in [0, L_q]$. If we let $\mathbf{v}_0 = c_n(R\hat{\mathbf{n}}_i) + c_t(R\hat{\mathbf{t}}_i)$ then the range of possible translations is given by

$$c_n \in \Big[\langle \mathbf{B}_q, R\hat{\mathbf{n}}_i \rangle - \langle \mathbf{b}_i, \hat{\mathbf{n}}_i \rangle - \epsilon_p - \max\left\{0, L_q \left\langle \hat{\mathbf{T}}_q, R\hat{\mathbf{n}}_i \right\rangle\right\},$$
$$\langle \mathbf{B}_q, R\hat{\mathbf{n}}_i \rangle - \langle \mathbf{b}_i, \hat{\mathbf{n}}_i \rangle + \epsilon_p - \min\left\{0, L_q \left\langle \hat{\mathbf{T}}_q, R\hat{\mathbf{n}}_i \right\rangle\right\}\Big]$$
$$c_t \in \Big[\langle \mathbf{B}_q, R\hat{\mathbf{t}}_i \rangle - \langle \mathbf{b}_i, \hat{\mathbf{t}}_i \rangle - \ell_i - \epsilon_L + \max\left\{0, L_q \left\langle \hat{\mathbf{T}}_q, R\hat{\mathbf{t}}_i \right\rangle\right\},$$
$$\langle \mathbf{B}_q, R\hat{\mathbf{t}}_i \rangle - \langle \mathbf{b}_i, \hat{\mathbf{t}}_i \rangle + \epsilon_L - \min\left\{0, L_q \left\langle \hat{\mathbf{T}}_q, R\hat{\mathbf{t}}_i \right\rangle\right\}\Big].$$

Thus, for the given rotation θ', these expressions define a polygon in the translation subspace of the Hough space. Any bucket in the tesselate Hough space that intersects this polygon denotes a possible transformation consistent with the given pairing of data and model fragment. Thus, we place the pair ($\texttt{linear}_i$, $\texttt{LINEAR}_q$) into each such bucket. This computation was done assuming a rotation θ' based on aligning the data normal and the model normal. Since, in general, there may be error in the data normal, we repeat the above process for a sampling of angles, chosen from the range

$$[\theta' - \epsilon_a, \theta' + \epsilon_a].$$

If the data edge is longer than the model edge, subject to the error in measuring length, then nothing is done.

A similar computation holds for the pairing of a circular data arc, i and a circular model arc, q. Suppose that the radius of the data arc agrees with the radius of the model arc, subject to the error in measuring such radii. Then, given a rotation angle θ, the condition on the translation part of the transformation is simply given by

$$\langle R\mathbf{C}_q + \mathbf{v}_0 - \mathbf{c}_i, R\mathbf{C}_q + \mathbf{v}_0 - \mathbf{c}_i \rangle \leq \epsilon_{cd}^2.$$

In general, as we sweep through all possible rotation angles, the position of this circle of possible translation vectors will trace out a helix in Hough space. However, we need only consider the range of angles θ' such that the rotated range of swept angles for the model arc will lie within the range of swept angles for the data arc, subject to error in measuring such swept angles For each such angle, there will be a set of translations associated with it. Now, as before, for every bucket in the Hough space that interesects the circle defined by the above condition, we place the pair $(\texttt{circular}_i, \texttt{CIRCULAR}_q)$ into the bucket. If the radius of the data arc does not agree with the radius of the model arc, subject to the error in measuring such radii, then nothing is done.

We can repeat this process for all possible pairings of data elements to model fragments, adding pairs to appropriate Hough buckets. Each such pair essentially votes for the set of transformations with which it may be consistent. Having done this, we can then rank the Hough buckets. We do this by assigning to each bucket a measure, determined by the sum of the lengths of the linear data edges assigned to the bucket plus the sum of the arc lengths of the circular data edges assigned to that bucket. This allows us to sort the Hough buckets, in decreasing order.

Now each bucket defines a new interpretation tree. It contains a number (usually much less than the total number) of data fragments, and associated with each one is a set of possible matching model fragments. By adding the wild card character as before, we can apply our constrained search process to this much smaller interpretation tree, to obtain consistent interpretations. We can simply search through the Hough buckets in sorted order until we obtain a valid interpretation.

Note that this process has ignored the effect of scale in the object transformation. We can incorporate scale in at least two different ways. The first would be to add an additional dimension to our Hough space, and then to place data-model pairs in this four dimensional space based on the set of translation, rotation and scale factors consistent with such a pairing. A second method is to increase the number of buckets into which a data-model pair are placed by increasing the bounds on the distance allowed between a Hough bucket and the set of translation and rotation factors deemed consistent with a pairing. By placing bounds on the range of possible scale factors, one can determine appropriate bounds on this distance. Note that such a range of scale factors will only affect the translation components of the Hough space. In our implementation, we choose the latter approach.

Also note that we need not use a Hough space whose dimensionality matches the number of degrees of freedom of the object models, since we are not relying on the Hough transform to directly interpret the data. Rather, since we only use the

Hough transform to reduce the search space, we can use any number of dimensions in our Hough space, trading off expense of computing the Hough transform against the gain in reduction of the final search space.

Premature termination

We can add a second heuristic to our search method, which also drastically reduces the effort involved. Suppose we have reached a leaf in one of our interpretation trees, and the interpretation associated with it is consistent. Since many of the data fragments in the interpretation are likely to have been assigned the wild card character, our search method would proceed to backtrack, attempting to find another interpretation that accounted for more of the data. In many cases, this is a fruitless task [Grimson and Lozano-Pérez 87]. We can truncate this search, at the possible risk of occasionally misinterpreting the data. In particular, we can apply a measure of goodness of match to each consistent interpretation. If that measure exceeds some predefined threshold, then we can accept the interpretation, and terminate the search in that particular interpretation tree. Reasonable measures of match include the number of data fragments accounted for, and a measure of the percentage of the object model accounted for, determined by the ratio of the sum of lengths of the linear data fragments accounted for plus the sum of the arc lengths of the circular data fragments accounted for, relative to the overall perimeter of the object model. In our *implementation*, we use the perimeter method.

These two techniques can be combined to produce a very efficient recognition system. We can search through the sorted Hough buckets, applying our constrained search method to the interpretation tree defined by the bucket contents. If we find an interpretation that exceeds our predefined measure of match, we can remove the data fragments that have been accounted for, adjust our Hough buckets accordingly, and continue the process, until we have either identified all of the edges in the data, or all of the Hough buckets have been exhausted.

Note that in using a cutoff based on percentage of object accounted for, one can weight the edges based on relative importance, possibly by using a measure of saliency [Turney, Mudge and Volz, 86].

3. Getting the fragments from real data

We have assumed that both the *object models* and the sensory data consist of sets of edge fragments, both linear and circular, as characterized in Section 2.2. Given such assumptions, we have developed a constrained search technique that will find interpretations of the data relative to the model. We must show, however, that the assumption on the form of the models and sensory data is valid. To do this, we describe a method for obtaining linear and circular edge fragments from grey level images. This will be used both to build the object models automatically, and to process the input sensory data.

The first stage in our processing is the extraction of sharp intensity changes in the grey-level input image. There is a large body of literature on the problem of edge detection, and any of several different edge detectors would suffice for our purposes. For a variety of reasons, we use a Marr-Hildreth [Marr and Hildreth 80] Laplacian of Gaussian edge detector. Applying this operator to the image reduces the sensory input to an array of connected edge points, where a 1 in a pixel indicates an edge point, and all other points are 0.

Next, we extract connected contours from this array. This can be done by a simple tracing operation. Note that it is not critical if missing edge points cause the tracing operation to fragment the edge contours into a set of smaller ones.

As we extract each edge point of a contour, we record two pieces of information, an estimate of the local orientation of the edge at that point, and an estimate of the change in arclength between the previous edge point and the current one. Since these measurements tend to be noisy, we smooth both of them by recursive averaging. This yields a transformed representation of the edge contour, now mapped into an arclength-orientation (Θ-s) space [Perkins, 78, 80, McKee and Aggarwal 77].

The advantage of such a transformation is that the edge fragments are now easily extracted. Note that a straight line in the original image space maps to a horizontal line in (Θ-s) space, and a circular arc in the original image space maps to a slanted line in (Θ-s) space. Thus, to extract our edge fragments, we simply need to parse the (Θ-s) space representation. We do this by applying a simple split-and-merge [Horowitz and Pavlidis 76, Chen and Pavlidis 79] algorithm to extract linear segments from the transformed representation. To do this, we must specify a bound ϵ_{sm} on the maximum deviation between the straight line and the contour being approximated.

Any non-horizontal line identifies a circular arc. Note that to determine horizontal from non-horizontal lines, we need a bound on the angle between the line and the s-axis, say ϵ_h. The radius of the circular arc is given by the inverse slope of the linear segment in (Θ-s) space. To find the center of the circle, we use the following method. First, we transform the circular segment back into the image space, and choose a sampling of pairs of points from the transformed segment. Let 2ℓ denote the separation of the two points. Next, we construct a perpendicular bisector to this chord. The center of the circle must lie a distance $\sqrt{r^2 - \ell^2}$ along the bisector. We can determine on which side of the chord the center lies, by ensuring that the points between the two sample points lie on the opposite side of the chord. We can collect all such hypothesized centers, over some set of sample points and use the midpoint of the collection to determine the circle center.

This gives us an estimate of the center of the circle. Since the computation of the circle radius may be noisy, we can extend this method by performing the above computation for a range of possible values for the radius. For each hypothesized radius and center, we can measure the deviation of the data points from the hypothesized circular, and select the circle with minimum error.

Given the circle center and the two endpoints of the circular arc in image coordinates, we can determine the limits on the swept angle straightforwardly. Finally, if we know the sign of the contrast between the object and the background, we can use the direction of the change in edge intensity across the edge to determine the pointing direction of the arc.

To ensure that the computed fragments are optimal, we perform a second split-and-merge stage, this time in the image space. That is, given a circular fragment, computed as above, we test that all of the data points lie within a given error range of the hypothesized fragment. If they do not, we split the data points at the point of maximum deviation, and perform the same computation on each of the subparts.

We are left with the horizontal lines in (Θ-s) space. To extract the linear edge fragments, we transform all of the points along these lines back into the image space, and run the split-and-merge algorithm again in this space. This allows us to extract the endpoints of the linear fragments. The normal is orthogonal to the line between the endpoints. If we know the sign of the contrast between the object and the background, we can use the direction of the change in edge intensity across the edge to determine the sign of the normal.

In our experience, this second split-and-merge stage in the image space is important. If we simply rely on the first split-and-merge operation, we have found that the actual edges in the image space corresponding to the horizontal lines in the (Θ-s) space have significant residual curvature. This is not surprising, since in one case we are thresholding based on deviation in curvature, and in the other, we are thresholding based on deviation from linear. As a consequence, the second split-and-merge stage results in the segmentation of horizontal (Θ-s) lines into several image space lines, with much tighter fit.

4. Putting it all together

4.1 Building the library of objects

We now have the pieces needed to build our recognition engine. We begin by building a library of object models. This is accomplished by placing each part in isolation under a camera, and running the fragment extraction process described in Section 3. This produces a set of linear edge fragments and a set of circular edge fragments, defined in a local coordinate frame.

We can improve the efficiency of our recognition system by doing some preprocessing on this representation. In particular, for each object, we build a set of tables capturing the model halves of each of the constraints. For each unary constraint, we build a one dimensional table, indexed by face number, in which we store the value of the model half of the constraint. For example, for the length constraint, this would involve computing and storing the length of each edge plus the error bound,

$$L_p + \epsilon_L.$$

For each binary constraint, we build a two dimensional table, in which we store the value of the model half of the constraint. For example, for the angle constraint, this would involve computing and storing the range of angles between a pair of edge normals, adjusted for error,

$$[\Theta_{pq} - \epsilon_a, \Theta_{pq} + \epsilon_a].$$

This precomputation makes the search process significantly faster, since half the computation is reduce to a table lookup.

Having built a model for a single object, we can straightforwardly build a second model for the mirror reversal of the object. This gives us two models per object, but allows us to recognize laminar objects in either stable orientation.

4.2 Processing the sensory data

Once we have constructed the library of objects, we are ready to process arbitrary images of the objects. Using the process described in Section 3, we reduce a grey-level image of a pile of parts to a set of linear and circular edge fragments. Next, we apply a Hough transform to the data, for each model in the object library. This yields a sorted list of Hough buckets for each model. We use our bound on the goodness of match to remove any Hough buckets without sufficient contents from consideration. Then, starting with the best Hough bucket, as measured over all the objects, we apply our constrained search, using premature termination to stop when a sufficiently good interpretation is found. If such an interpretation is found for the current Hough bucket, we remove the edge fragments accounted for from consideration, adjust the contents of the Hough buckets for all objects, and resort each list of Hough buckets. We then proceed as before, continuing until no further Hough buckets remain. If no interpretation is found for a Hough bucket, we simply move on to the next best bucket and continue. An example of such processing is shown in Figure 1.

4.3 Unknown edge normals

In the preceeding discussion, we have assumed that we can identify the correct direction of the normals to linear edge fragments, and the pointing direction of the circular arcs. This operation relies on knowing the contrast between the background and the objects. If such information is available, the contrast across an intensity edge will determine these properties.

In many cases, however, it is unreasonable to assume that this information will be known. We can extend our system to deal with this case. One solution is based on the following observation, and has been reported in [Grimson and Lozano-Pérez 87]. 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. As a consequence, 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 choose the sign of one of the normals. The angle constraint can then be used to consistently select the signs for other edges in that interpretation. This does allow the method to correctly interpret data with unknown data edge normal signs, at a small increase in the search cost.

A second solution is simply to double the number of sensory linear edge fragments, one with each possible sign of the edge normal. The constraints will then ensure that at most one of each such pair of edge fragments in included in the interpretation, and the process can proceed as before. Here, the pointing direction constraint is not used.

4.4 Other extensions

Although we have presented the system as recognizing objects from their occluding boundaries, it is more broadly applicable than this. In particular, since we use an edge detector to extract our primitives for matching, other object markings, such as albedo or material changes, or surface texture, that are stable across a range of imaging conditions would also suffice.

Three dimensional objects that are known to be in stable positions can also be handled using this method. For each stable position, we can build an object model by running the front end of the system. The assumption of stable position removes the effects of perspective, and allows us to treat the problem as essentially a two-dimensional one.

5. Testing

We have implemented and tested a version of the curved object recognition system. Our implemented version differs slightly from the description given above. In particular, we have not included any of the cross constraints, relying only on the constraints between segments of the same type. Our expectation is that the non-inclusion of such constraints should at worst increase the search time spent in finding correct interpretations, without causing any incorrect interpretations to be found.

We have run the system on a sequence of images similar to that shown in Figure 1. Each image consisted of six overlapping parts, selected with repetition from two different types of parts, and placed at random, with possible mirror reversals, as shown. In each case, we asked the system to find as many interpretations as possible from the library of parts, where each part could appear an arbitrary number of times. After each interpretation was found, the accounted for edges were removed from the data, and the process was continued, until no further portions of the search space remained unaccounted for.

For each image, the system was run in three different settings, using perimeter percentage thresholds of .10, .20 and .50. For each such triple of settings, the system was run with two different tesselations of the Hough space. In the coarse case, the Hough sampling was 50 pixels in the translation components (where the entire image was 576 by 454) and 36 degrees along the rotation axis. In the fine case, the Hough sampling was 25 pixels in the translation components and 18 degrees along the rotation axis. Over 5 trials, the system had the performance indicated in the following table.

	Coarse		Hough	Fine		Hough
Perimeter %	.50	.20	.10	.50	.20	.10
Correct	2.8	5.2	4.8	2.6	5.2	5.4
Multiple		0.6	0.8		0.8	1.2
Mirror		0.2	0.6		0.2	0.2
Incorrect		0.4	0.8		0.2	0.2
Perimeter	.66	.43	.38	.67	.45	.41
Real Nodes	741	473	585	867	754	576
Real Model Tests	236	149	203	344	341	245
Final Nodes	1942	1989	3252	1879	2385	6029
Final Model Tests	627	745	1571	787	1143	290
Parsing Time	290			290		
Parsing Time	135			686		
Search Time	90	106	135	75	398	434
% Search in Final Stage	.72	.81	.85	.68	.76	.91

Each of the columns of the table indicates the results of using a different threshold on the percentage of the perimeter of an object needed for a valid interpretation. The **correct** line indicates the mean number of correct interpretations found over the set of trials. The maximum number of valid interpretations is 6 per trial. The **incorrect** line indicates the mean number of incorrect interpretations found per trial. We also indicate the mean number of multiple interpretations, that is, situations in which the system found nearly identical, correct interpretations, based on different subsets of data, and we indicate the mean number of incorrect interpretations involving the mirror reversal of an object.

Note that in the case of a perimeter percentage of .20, the system found almost all of the possible correct interpretations. Each of the incorrect interpretations involved the larger object in Figure 1, in which the circular structure was correctly matched, but at the wrong orientation. Since the circle contributes a large amount to the total object perimeter, only a small number of other edges were needed to find a feasible but incorrect match. The interpretations that were not found all involved the small object shown in Figure 1, and in all cases, the object was heavily occluded. In the case of a perimeter percentage of .50 all the found interpretations were correct. In the case of a perimeter percentage of .10, the performance degraded slightly, with more incorrect or mirror interpretations. This is not surprising, since we are only requiring 10% of the object to be matched in situations involved a reasonable amount of clutter. The `perimeter` line of the table indicates the average percentage of the object's perimeter actually included in the interpretation.

The `real nodes` line and the `real model tests` line indicate the mean number of nodes of the interpretation tree, and the mean number of model transformation tests performed for each of the interpretations found. The `final nodes` and `final model tests` lines indicate the amount of search performed after the last interpretation was found in each trial. Not surprisingly, these numbers are much higher, since considerably more effort is involved in verifying that no further interpretations can be found using the remaining scattered data fragments.

The `Time` lines in the table indicate the mean time involved in parsing the intensity edges into linear and circular fragments, in transforming these fragments into the Hough space, and in executing the actual search process. The times are reported in seconds of elapsed time for an implementation on a Symbolics Lisp Machine, without floating point hardware. The final line indicates the portion of the search time that was spent in verifying that no further interpretations remained. These timing statistics are intended only for comparative purposes. A number of optimizations of the code are possible, and would considerably reduce these numbers. For instance, in the Hough transformation, we are using a very fine sampling of rotation angles, yielding a large number of nearly overlapping polygons in Hough space, which are then intersected with the buckets of the Hough space. Considerable savings could be obtained by using a coarser sampling at the expense of possibly missing a feasible Hough bucket on occasion. Similarly, in the parsing of the input data, we are using an exhaustive search to find the best estimate of the radius and center of the circular fragments. This accounts for 80% of the time reported, and could clearly be sped up.

There are several interesting points about the described testing. First, note that most of the search is spent in verifying that no further interpretations exist. In general, the correct interpretations are found with very little search. This suggests that the system in fact behaves as a hypothesize-and-test system, in which the Hough transform serves to hypothesize possible interpretations, that are then verified by the constraint satisfaction process. We note that the Hough transform is not sufficient alone, as we have frequently observed that the biggest Hough bucket did not result in a correct interpretation. Moreover, there can be considerable diffusion in the Hough space, due to the errors in the sensory data. As a consequence, a large number of Hough buckets may have comparable sized contents. This is illustrated in Figure 6, which shows the Hough space for one of the objects of Figure 1, at two different resolutions.

Second, we note that if the system can correctly identify most of the objects in the scene, then it would seem that it should not have to spend considerable time in additional search verifying that no further interpretations exist, since in general most of the data fragments should already be accounted for. As can be seen from the table, however, considerable effort is spent in doing this. In part, this follows from the fact that the interpretations found by the system frequently do not account for all the data fragments arising from the object. This occurs for several reasons. First, the (Θ-s) space segmentation scheme does not produce canonical partitions of the input data. Hence, small deviations in the image may cause a noticably different data segmentation, and some data edges may be different enough from the model to be excluded from the interpretation. Our experience suggests that this sensitivity may be more true of the (Θ-s) space segmentation than of strictly polygonal segmentations. This sensitivity to segmentation could be handled by increasing the error bounds discussed in the next section. This is dangerous, however, since the increased bounds are also likely to cause more accidental alignments of data fragments to be incorrectly interpreted. A better solution would be to do additional verification in the image space. That is, having found a correct interpretation based on moderate error bounds, one could then project the interpretation back into the image, and using looser bounds search for additional data fragments that are in agreement with the projected object position.

Many of the incorrect interpretations involved solutions in which the large circular hole of the large object shown in Figure 1 was matched correctly, but the overall orientation of the solution was incorrect. Since there is an inherent ambiguity in the rotation of the object about the center of the hole, while at the same time, the perimeter of the hole contributes a large portion of the overall perimeter, if a small portion of the object happens to align accidentally with some data fragment, we can obtain an incorrect interpretation that accounts for a noticeable portion of the object's perimeter. We need some means of handling this problem, perhaps by using a variant of the Feature Focus method of Bolles 1982 .

Finally, note that the different samplings of the Hough space did not lead to significantly different performances in terms of the number of interpretations.

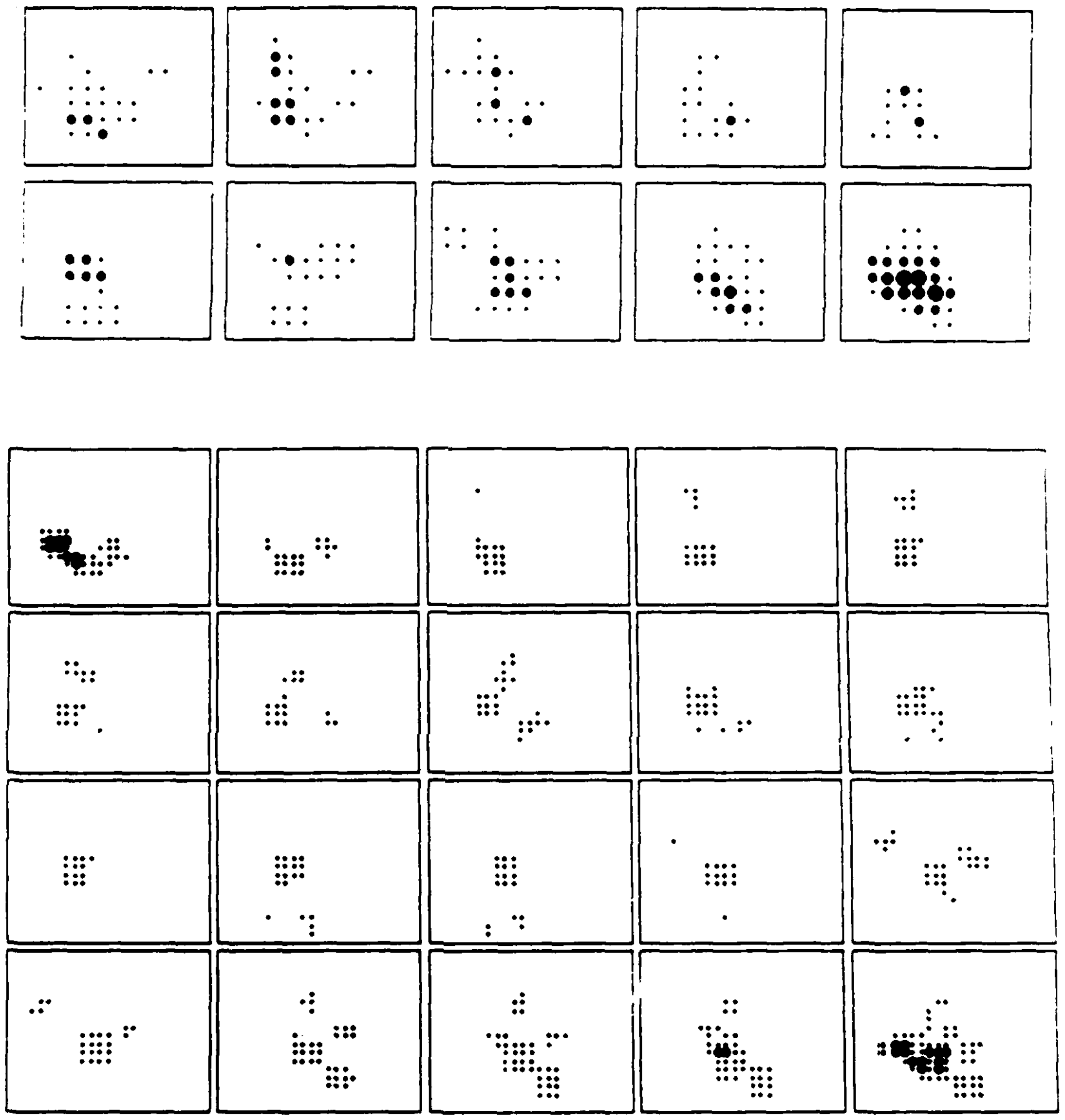


Figure 6. Samplings of the Hough space. The top part shows the Hough space for the smaller object of Figure 1, sampled at a coarse resolution. Each frame represents a different orientation, increasing from left to right. Within each frame, the contents of each of the translation buckets is shown, with the size of the dot indicating the size of the buckets contents. Note the smearing of this measure over a broad extent of the space. The bottom part shows the same Hough space at a finer resolution.

6. Free parameters and error bounds

In describing our recognition system, we have used a number of free parameters and error bounds. While at first glance there appear to be a large number of such free parameters, in fact, many of them are interrelated, and only a few need to be determined in order to run the system.

The first parameter is the bound on the accuracy of measuring the position of an edge point ϵ_p. This bound is a function of the camera system and the edge detector used. Since we are using a Marr-Hildreth operator, the accuracy of the system could be determined from formal analysis [Berzins 84], or could be measured empirically. Note that since this is simply an upper bound, we can be conservative in our estimates.

Given this bound ϵ_p, a number of the other free parameters follow directly. For example, suppose that the position of an edge point is known to within the error bound ϵ_p. If L_{min} is a lower bound on the length of the edges, it is straightforward to show that the maximum error in the measured angle between edge normals is given by

$$\epsilon_a = \tan^{-1}\frac{2\epsilon_p}{L_{min}}.$$

The bound on measuring the length of a linear fragment is also determined by ϵ_p, in particular, the worst case bound is given by

$$\epsilon_L = 2\epsilon_p.$$

The error in measuring the radius of a circular arc ϵ_r will generally be on the order of the error in measuring the position of an edge point ϵ_p. Since the radius is determined by taking the slope of a line in Θ-s space, it is likely to be less than this, but using ϵ_p is a conservative bound. The error in measuring position of the center of a circular arc ϵ_{cd} will also typically be bounded by ϵ_p. Similarly, a conservative bound on the error in measuring the swept angle range is $\epsilon_c = \epsilon_a$.

The bound on the split and merge algorithm ϵ_{sm} is something that we must set by hand. Note that so long as our models are built using the same value of the parameter as that used in processing sensory data, and so long as this value is not too large, the exact value is not critical.

Setting the parameter that distinguishes straight lines from circular arcs, ϵ_h can be done based on properties of the objects to be recognized. In particular, since ϵ_h is a bound on the angle between the horizontal axis and a line in Θ-s space, if the radius of the largest circular arc on any object is R_{max}, then we can set

$$\epsilon_h = \frac{1}{R_{max}}.$$

Thus, the various error bounds in the algorithm can be determined by measuring the accuracy of the system in determining the position of an edge point, ϵ_p, by specifying the minimum length required for an edge fragment L_{min} and by specifying the maximum radius of a circular arc R_{max}.

There is one other threshold in our system, namely the threshold used to determine an acceptably sized interpretation. We have indicated that our measure of an interpretation is the sum of the lengths of the linear data fragments in the interpretation plus the sum of the arc lengths of the circular data fragments in the interpretation. We use a threshold on this measure in two places: to remove small Hough buckets from the search process, and to prematurely terminate the IT search once an acceptable match is found. Unfortunately, there does not seem to be any principled way of setting this threshold. Clearly, we can trade of false positives and false negatives by varying it, since the smaller the threshold, the more likely an incorrect interpretation is accepted, while the larger the threshold, the more likely that correct interpretations will be missed. In our experiments, we have typically left the threshold at 20 - 25% of the total perimeter of an object.

Aldo note that a straightforward application of a threshold on perimeter ignores information about what portions of the object are matched. For example, an interpretation accounting for .25 percent of the object, but in which all .25 percent came from one end of the object, may be less reliable than an interpretation in which the .25 percent is spread out over the perimeter of the object.

7. Relation to previous work

The literature on object recognition systems is extensive, and stretches over a period of at least twenty years. Of the variety of different techniques examined, a number of authors have taken a similar view to ours that recognition can be structured as an explicit search for a match between data elements and model elements [Ayache and Faugeras 86, Baird 85, Bolles and Cain 82, Bolles, Horaud and Hannah 83, Browse 87, Drumheller 87, Faugeras and Hebert 83, [illegible] and Lozano-Pérez 84, Goad 83, Kalvin et al. 86, Knoll and Jain 85, Lowe 86, Murray 87, Pollard et al. 87, Schwartz and Sharir 87, Stockman and Esteva 84]. Of these, the work of Bolles and his colleagues, Faugeras and his colleagues, and that of Baird are closest to the approach presented here.

The interpretation tree approach is an instance of the consistent labeling problem that has been studied extensively in computer vision and artificial intelligence [Waltz 75, Montanari 74, Mackworth 77, Freuder 78, 82, Haralick and Shapiro 79, Haralick and Elliott 80, Mackworth and Freuder 85]. This paper can be viewed as suggesting a particular consistency relation (the constraints on distances, angles, and radii) and exploring its performance. An alternative approach to the solution of consistent labeling problems is the use of relaxation. A number of authors have investigated this approach to object recognition [Ayache and Faugeras 82, Bhanu and Faugeras 84, Davis 79, Rutkowski et al. 81, Rutkowski 82]. These techniques are more suitable for implementation on parallel machines.

The use of a Θ-s space or some equivalent to extract representations of curved laminar objects has been previously investigated. Perkins 78, 80 describes a system using similar representational fragments, extracted from a Θ-s space, as well as some simple constraints for determining potential matches. These are then evaluated using cross-correlation in Θ-s space. Other systems that use Θ-s space to partition input data into segments include Barrow and Popplestone 71, Clemens 86, Martin and Aggarwal 79, McKee and Aggarwal 77, Turney et al. 85. The Curvature Primal Sketch developed by Asada and Brady 86, and used in a recognition system by Ettinger 87 also uses an explicit representation of changes in the edge contours as a basis for matching objects.

References

Asada, H. and Brady, M. 1986. The Curvature Primal Sketch. *IEEE Trans. Pattern Anal. Machine Intell.* **8**(1):

Ayache, N. J. and Faugeras, O. D. 1982. Recognition of partially visible planar shapes. *Proc. 6th Intl. Conf. Pattern Recognition*, Munich.

Ayache, N. J. and Faugeras, O. D. 1986. HYPER: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-8(1):44 54.

Baird, H. 1986. *Model-Based Image Matching Using Location*. Cambridge: MIT Press.

Ballard, D. H. 1981. Generalizing the Hough transform to detect arbitrary patterns. *Pattern Recogn.* **13**(2):111 122.

Barrow, H. G., and Popplestone, R. J. 1971. Relational descriptions in picture processing. *Machine Intell.* **6**:37700396.

Berzins, V. 1984 Accuracy of Laplacian edge detectors. *Comp. Vision, Graphics and Image Proc.* **27**(2):195 210.

Bhanu, B. and Faugeras, O. D. 1984. Shape matching of two-dimensional objects. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-6(3).

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. Paper delivered at First International Symposium of Robotics Research, Bretton Woods, N.H. (Also in Robotics Research: The First International Symposium, edited by M. Brady and R. Paul, MIT Press, 1984, pp. 113 421.)

Browse, R. A., 1987. Feature-based tactile object recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, to appear.

Chen, P. C., and Pavlidis, T. 1979. Segmentation by texture using a co-occurence matrix and a split-and-merge algorithm. *Comput. Graphics Image Proc.* **10**:172 182.

Clemens, D. T. 1986. The recognition of two-dimensional modeled objects in images. M. Sc. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Davis, L. Shape matching using relaxation techniques. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-1(1):60 72.

Drumheller, M. 1987. Mobile Robot Localization Using Sonar. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-9(2): 325–332. (See also: S. B. Thesis, Dept. of Mechanical Engineering, MIT, 1984 and MIT AI Lab Memo 826, Mobile Robot Localization Using Sonar.)

Ettinger, G. J. 1987. Hierarchical object recognition using libraries of parameterized model sub-parts. M. Sc. Thesis. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

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.

Faugeras, O. D., Hebert, M., and Pauchon, E. 1983 (June, Washington DC). Segmentation of range data into planar and quadratic patches. *Proc. CVPR '83.*

Freuder, E. C. 1978. Synthesizing constraint expressions. *Comm. of the ACM*, 21(11), pp. 958 966.

Freuder, E. C. 1982. A sufficient condition for backtrack-free search. *J. ACM*, 29(1), pp. 24–32.

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.

Goad, C. 1983. Special purpose automatic programming for 3d model-based vision. in *Proceedings of DARPA Image Understanding Workshop*

Grimson, W. E. L., 1986a. The combinatorics of local constraints in model-based recognition and localization from sparse data. *J. ACM* 33(4):658 686.

Grimson, W. E. L. 1986b. Sensing strategies for disambiguating among multiple objects in known poses. *IEEE Journal of Robotics and Automation*, **2**, 196 213.

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.

Grimson, W. E. L. and Lozano-Prez, T. 1987. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. Patt. Anal. and Mach. Intel.* **9**(4) 469 482. (see also MIT AI Lab Memo 841, June 1985.)

Haralick, R. M. and Elliott, G. 1980. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, Vol. 14, pp. 263 313

Haralick, R. M. and Shapiro, L. G. 1979. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-1(4):173 184.

Horowitz, S. L., and Pavlidis, T. 1976. Picture segmentation by a tree traversal algorithm. *J. ACM* **23**:368-388.

Hough, P. V. C. 1962. Methods and means for recognizing complex patterns. *U.S. Patent 3069654.*

Kalvin, A., Schonberg, E., Schwartz, J. T., and Sharir, M. 1986. Two-dimensional, model-based, boundary matching using footprints. *Int. J. Robotics Res.* **5**(4):38 55.

Knoll, T. F. and Jain, R., 1985. Recognizing partially visible objects using feature indexed hypotheses. *University of Michigan Center for Research on Integrated Manufacturing Report* RSD-TR-10-85.

Lowe, D. G. 1986. Three-dimensional object recognition from single two-dimensional images. Courant Institute Robotics Report, No. 62.

Marr, D. and Hildreth, E. C. 1980. Theory of edge detection. *Proc. R. Soc. Lond. B* 207:187-217.

Mackworth, A. K. 1977. Consistency in networks of constraints. *Artificial Intelligence*, Vol. 8, pp. 99-118.

Mackworth, A. K. and Freuder, E. C. 1985. The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artificial Intelligence*, Vol. 25, pp. 65-74.

Martin, W. N., and Aggarwal, J. K. 1979. Computer analysis of dynamic scenes containing curvilinear figures. *Pattern Recogn.* **11**:169 178.

McKee, J. W., and Aggarwal, J. K. 1977. Computer recognition of partial views of curved objects. *IEEE Trans. Comput.* **C-26**(8):790 800.

Merlin, P. M., and Farber, D. J. 1975. A parallel mecahnism for detecting curves in picture. *IEEE Trans. Comput.* **C-24**:96 98.

Montanari, U. 1974. Networks of constraints: Fundamental properties and applications to picture processing. *Inform. Sci.*, Vol. 7, pp 95 132.

Murray, D. W. 1987. Model-based recognition using 3D structure from motion. *Image and Vision Computing* **5**(2):85 90.

Perkins, W. A. 1978. A model-based vision system for industrial parts. *IEEE Trans. Comput.* **C-27**(2):126 143.

Perkins, W. A. 1980. Simplified model-based part locator. *Proceedings 5th Intern. Conf. Pattern Recognition* pp. 260 263.

Pollard, S. B., Porrill, J., Mayhew, J. E. W., and Frisby, J. P. 1987. Matching geometrical descriptions in three-space. *Image and Vision Computing* **5**(2):73 78.

Rutkowski, W. S. 1982. Recognition of occluded shapes using relaxation. *Computer Graphics and Image Processing.* **19**:111-128

Rutkowski, W. S., Peleg, S., and Rosenfeld, A. 1981. Shape segmentation using relaxation. *IEEE Trans. Pattern Anal. Machine Intell.* **3**(4):368 375

Schwartz, J. T. and Sharir, M. 1987. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *Int. J. Robotics Res.* **6**(2):29-44.

Sklansky, J., 1978. On the Hough Technique for curve detection. *IEEE Trans. Comput.* **C-27**:923-926.

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

Turney, J. L., Mudge, T. N., and Volz, R. A. 1985. Recognizing partially occluded parts. *IEEE Trans. Pattern Anal. Machine Intell.* **7**(4):410-421.

Waltz, D. 1975. Understanding line drawings of scenes with shadows. in *The Psychology of Computer Vision*, P. Winston, Ed. New York:Mc Graw Hill, pp 19-91.

END

10-87

DTIC