USING SURFACES AND OBJECT MODELS TO RECOGNIZE PARTIALLY OBSCURED OBJECTS

Robert B. Fisher

Department of Artificial Intelligence
University of Edinburgh

ABSTRACT

This paper describes an approach to object location based on matching model regions to surface image regions. These matches, together with object models, provide hypotheses about the 3D location of the objects containing the recognized surfaces. When a hypothesis is complete, the program describes the matched data and model surface boundaries in terms of the raw model. These descriptions are used used to verify the physical consistency of the hypotheses and to explain extra or hypothesize missing features caused by obscured components.

1. Introduction

The most important recent work in high-level vision (ACRONYM [3]) has demonstrated the importance of embedding object models and an understanding of the scene-to-image transformation In an intelligent vision program. With these components, a program can predict how an object will appear given its current beliefs about the scene, rather than having to rely on image models (or, more likely, a feature space representation of these views). Concurrently, much low-level vision research has been attempting to provide a surface-based description of images.

This paper describes a program (IMAGINE) which matches surface regions to object models to recognize and locate projections of three dimensional objects in two dimensional images. The approach is data driven, with three major stages. The first stage matches image regions to model surfaces, with the goal of estimating the 3D orientation parameters for the Image region. This Information Is used to hypothesize specific object surfaces. The second stage relates the hypotheses according to the structural relationships embodied in the object models. The third stage verifies that the hypothesized objects are consistent with real world constraints, such as boundary adjacency and surface ordering. This process requires calculating descriptions of the predicted surfaces and associated data regions. Recognition is considered successful if a set of data is found that adequately accounts for all features of a model. In particular, this requires the program to reason about object back surfaces, tangent surfaces, small features, obscured features and non-rigidly attached subcomponents.

This work was supported by a post-graduate studentship from the University of Edinburgh. Thanks also go to J.A.M. Howe and R.J. Beattle.

in short, the program has four goals:

- to locate instances of specific 3D objects in 2D images,
- to locate Images features corresponding to all features of the model, or explain why the image features are not present.
- to verify that the features are consistent with the geometrical and topological predictions made by the model, and
- to extract the parameters needed to fully characterize an object's scene position, including any associated with flexibly connected objects.

Coping with obscured features of objects is an important point of this work. This has required work in two areas; first, in deciding what *are* relevant aspects of obscured objects (ie. what features disappear and what new ones appear), and second, in extracting information about those features. This second point has required work on extracting 3D positional Information from surface shapes and descriptions of surface and image regions.

This paper discusses the model and the reasoning rules used to analyze the main example image - that of a robot upper and lower arm assembly.

2. What does 'recognize* mean in this context

The meaning of "recognition" is largely dependent on the goals of the recognizer, that is. what type of output is desired for what type of input. If the only brown objects in the environment are tables, then a program could find tables by looking for brown regions. To some extent, all recognition programs are just discrimination programs, of which the table recognizer is a minimal such program. On the other hand, even the most sophisticated programs (e.g. ACRONYM [3]) are fundamentally the same, they attempt to find a set of evidence that is consistent with their model of the object and for which there is no contradictory evidence. The distinction is largely one of sophistication - richer models, greater embedded knowledge about object structure relationships and more detailed matchings reduce the possibilities of false recognitions.

In general, recognition consists of four subactions:

- location of Interesting data.
- selection of a model.
- instantiation and verification of model, and
- determination of object's location and orientation.

in particular, this program:

- Considers all data interesting. This is mainly because the input has been previously segmented into connected surface regions.
- Considers all models This is mainly because model invocation is a research problem not addressed in this work.
- Attempts to fully instantiate the model. A model has slots for a set of structurally related surfaces and subobjects The program either fills the slots or finds evidence to explain why they cannot be filled. (This is an important requirement for an intelligent vision program.) Surface slots are matched to image regions and substructure slots are matched to previously recognized objects. Fully instantiated hypotheses must pass a verification process.
- Locates and orients the objects in three dimensions. The parameters come from estimating the transformation that maps a given model surface to a given image region based on the shape of the image region, and then inverting the transformation from surface to object.

3. Surfaces

Tne project has concentrated on using surface regions as the primary data primitive. Though there are no programs yet that reliably and completely give a surface segmentation, it doesn't seem like a too distant possibility. Pentland's surface shape algorithm [61. various stereo algorithms (eg Grimson [5]). optical flow, range *finaer* systems *and* structured *light* systems all provide the information needed to construct a surace map. (Surface depth, shape and orientation are all largely equivalent representations.)

There are many reasons for using surfaces as the primitive input data element at this level of analysis. First, irrespective of what interpretation is made, what is seen is the surface of objects. Second, as surface regions are matched to model surface regions, the semantics of the relationship is simple: in unobscured situations, a six parameter geometric transformation plus a projection onto the image plane. Third, the segmentation of a surface image into regions of uniform surface properties may be simpler. To some extent, any region of homogeneous image properties is likely to correspond with some subset of a uniform surface. (The reverse is not always true, as a shadow across a flat surface would create two distinct regions.) There are many open problems on this topic, such as merging split regions, scale of surface descriptions, etc Fourth, surface regions lead to more accurate transformation parameter estimates (mainly because their size gives more accurate measurements). Point to model matching and transformation inversion algorithms (e.g. Roberts [7]), while elegant, are sensitive to image *measurement* errors. Worse, the loss of information from using just points makes It difficult to determine the image to model correspondences correctly. Lastly, *once* surfaces are chosen as the desired primitive. *other image* boundary information Is more useful:

- reflectance boundaries: surface detail
- obscuration: object limits, surface depth order
- shadow: surface location in three dimensions
- shape: surface relative orientation

The input used by the program is that of a surface image segmented into regions of either planer or simply curved orientation. Hence, the data is simplified in that we know all boundaries correspond to surface or shape discontinuities. Further, the boundaries form closed regions. The program could have used the *surface* depth or orientation information as part of its inputs, but it was decided to base the work just on the region shape.

4. Object Models

This work uses structural three dimensional models of the objects it expects to find The reasons for this are discussed by Binford [23: a capable vision system should know about objects, and how objects are seen in images, rather than what types of images an object is likely to produce. One difference between this work and that of Binford and Brooks (ACRONYM [3]) has been the choice of primitives used to construct the models They use solid object models (generalized cones) as the fundamental building block, and then deduce what image boundaries are produced. One problem with this is that what is seen is surfaces, hence it seems reasonable that the preferential representation for a vision system would make surface-based information explicit. (This argument is largely one of conceptual simplicity, in that it should be possible to automatically deduce a surface representation from a solid representation, though not necessarily easily.) Another difficulty with the generalized cone is that it seems to be a conceptually simple representation for only some object classes, though any one representation is probably not sufficient for all tasks.

Surface primitives are constructed by specifying the boundary of the region, with the assumption that the surface is either planer or has a single axis of curvature and fills the region inside the boundary.

Objects are described in a subcomponent hierarchy, with objects being composed of either connected surfaces or recursively defined sub-objects. The connections are specified using a six parameter attachment (x.y.z translations, rotation, slant and tilt). These attachments can be rigid (all parameters specified in the model) or may be flexible. Flexible attachments are handled by assigning the parameter a symbolic and which is bound value when the corresponding subcomponent is added to the object hypothesis. Symbolic parameters are only used with attachments.

Figure 2 shows some of *the* model used for the example in section 9. The full model consists of the upper and lower main arms of a Unimatlon PUMA robot. The figure *here* just shows some of the model for *the upper arm* and external linkages. Figure 1 shows the part models and the coordinate frame locations. The variable "jntN" is the symbolic parameter for the robot's Nth joint angle. In the example, the FACE primitives define the surfaces by listing key boundary points (30) and whether they are joined by a line or simple oriented curve. The ASSEMBLY blocks define named structures, in terms of either other named structures or surfaces. The "AT ((...).(..»" portion describes the XYZ translation, rotation, slant, tilt afflxment relationship between the main and substructure reference frames.

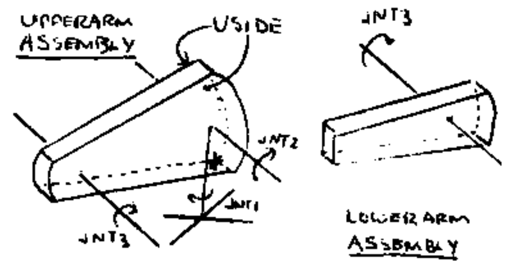


Figure 1 - robot arm model in pictorial form

```
ASSEMBLY upperarm =
     uside AT ((0,0,0),(0,0,0))
     WITH uside AT ((0,29.8,10),(0,3.142,1.57))
     WITH uendb AT ((0,0,10),(0,1.57,3.14))
     WITH uends AT ((61.8,7.4,0),(0,1.57,0))
     WITH uedge AT ((0,0,10),(0,-1.57,1.57))
     WITH uedge AT ((0,29.8,0),(0,1.57,1.57));
FACE uside = (0,0,0) LINE (19.6,0,0) LINE
     (61.8,7.4,0) CURVE[0.82,0,0] (61.8,22.4,0)
     LINE (19.6,29.8,0) LINE (0,29.8,0)
     CURVE[-0.38,0,0];
```

Figure 2 - Part of model used for the robot arm recognition example

5. Matching model surfaces to image regions

This process produces the main input used by the object recognition and construction process described in section 6. Its inputs are the boundary of the image region and a description of the surface as given by the model. It has the goal of estimating the transformation parameters that relates the image to the model

The visual motivation for this process is: *the* shape of an image region gives strong clues to the orientation of the surface. In particular, the program compares the image region shape to the model surface shape (using the boundaries) and attempts to extract measurements that lead simply to the desired parameters. The parameters *are* rotation, slant, tilt axis orientation, distance and x-y translation. The choice of rotation, slant and tilt as the orientation parameters was motivated by what information was easily found In an Image of a transformed surface. The initial parameter estimation process is based on the following observations:

- rotation and XYZ translation of a surface do not alter Its relative cross sectional dimensions, and that slant does not alter them drastically, over the range of estimated slants (less than 1.3 radians).
- the tilt axis lies approximately at the orientation of greatest slant distortion.
- the maximum distortion is an Indication of the slant relative to the line of sight.
- the relative scaling of features between the model and image indicates the distance.

The heuristics used for the parameter estimation are:

rotation - correlation of data and model cross sections. The peaks are potential rotation offsets. Figure 3 shows typical model and data shapes and their corresponding cross section versus angie functions. Note that the data function is largely just the translated model function.

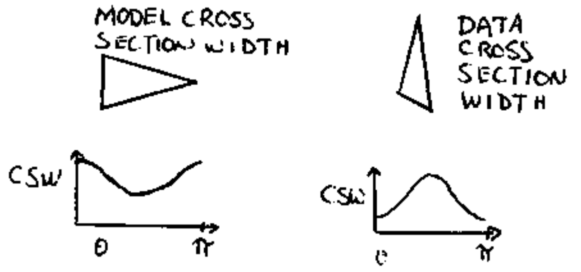


Figure 3 - model and data cross sections

slant and tilt - given an estimated rotation, create a ratio of maximum data to maximum model cross section widths as a function of cross section path angle. (Use of the maximum is a heuristic to relate approximately corresponding cross sections.) Slant, tilt and distance estimates come from the orientation of the minimum ratio, the ratio itself and the maximum ratio (figure 4).

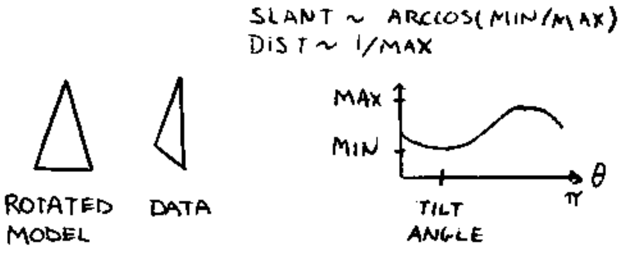


Figure 4 - data to model cross section ratio

distance - given rotation, slant and tilt, compute a ratio of data to model cross sections along the data region's major and minor axes. Two distance estimates are made from the average and most common ratios. The final value Is the average of these two and the previous estimates. The major motivation for the use of three estimators is various sensitivities to obscuration or parameter mis-estimation.

x.y translation - align data and mode centers of mass transformed by the estimated parameters

An optimization phase is then entered. The evaluation function is the weighted distance of the predicted model boundary from the observed image boundary. Because of the expected accuracy of the various parameter estimates, optimization preference is given to the worst estimate in order of tilt, distance, slant and rotation, with the x-y translation always adjusted to best fit at each stage.

Note that the transformation relating a modeled surface to an image region Is not always unique, especially when model symmetries, spatial quantization and inaccuracies of fitting numerical data are considered. Consequently, the parameter estimation process may produce more than one hypothesis.

These heuristics gave reasonable parameter estimates in 94 of 100 test cases recently. Given the tolerances in the matching process, the estimates are acceptable in the successful cases. Unfortunately calculating these parameters is largely numerical and computationally expensive - mainly because of image parameter measurement and model-to-data boundary comparisons during the optimization phase.

6. Some rules for recognizing objects

here, recognition is a construction process. At each stage, a hypothesized structure is under consideration. This hypothesis may have several empty substructure slots, in which case the program will attempt to fill the slots in various ways Or. the slots may all be full, in which case the hypothesis will be subjected to a verification process. Finally, the structure may be used to hypothesize another structure

Currently, the processing of the hypotheses is exhaustive, and is controlled by a first-in. first-out queue. The selection of which procedure to apply to a hypothesis is determined by a set of rules compiled from the set of meta-rules discussed below.

Some of the rules have binary right-hand-sides. When the matcher considers applying such a rule to a hypothesis, a prediction routine is called. Using the object models, this routine predicts where in the image the second hypothesis needed for the matching might be found (c.f. Freuder's Advice process [4]). Then, all hypotheses of the appropriate type from this location is paired with the current hypothesis node and the analysis routine is invoked for each pairing.

The general processing philosophy is for the rules to produce all hypotheses consistent with their definition. with the assumption that incorrect hypotheses will not be capable of full consistent instantiation.

About two dozen rules exist at the current time. The point of having different rules, rather than a general mechanism, is that each can carry out the specific type of reasoning needed to solve a particular problem. The rules described below are some of those used in the example given in section 9:

RULE 3: <surface hypothesls> <- <lmage reglon>

This rule selects, among all model surfaces, those that meet various pre-screening conditions relative to the particular image region (mainly minimum size relationships). Then, the parameter estimation process (section 5) is applied for each surface candidate. A surface hypothesis is generated for each suitable estimate.

RULE 1: <oblect hypothesls> <- <oblect hypothesls> <surface hypothesis>

This rule adds a new surface to an existing structure hypothesis. For the rule to apply, the global location of the new hypothesis predicted by the new surface has to be consistent with the previous hypothesis. The resulting hypothesis has all the previous information transferred. Including variable bindings.

RULE 5: <object hypothesis> <- <surface hypothesis>

This rule produces a structure hypothesis for each modeled object that has the particular face as a rigidly connected subcomponent. The location of the hypothesis is calculated by taking the location of the face and Inverting the attachment relationship to the main structure (figure 5).

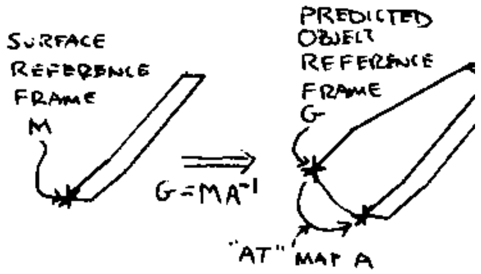


Figure 5 - surface hypothesis to object hypothesis rule

RULE 2: <object hypothesis> <-_ <object hypothesis>

This rule fills slots corresponding to surfaces invisible because they lie on the back sides of the object. This is done by using the object model to predict the surface orientation, given the hypothesis's location and orientation.

RULE 3: <object hypothesls> <- <object hypothesis>

This rule fills slots that correspond to near tangential surfaces on the front side of an object. Because of the severe slant distortion of such surfaces, parameter estimation is unreliable. Hence, the slots are filled with references to image regions that lie approximately in the right location and have approximately the correct size.

RULE 4: <verified object> <- <object hypothesis)

This rule is discussed In detail in section 7.

RULE 9: <object hypothesis> <- <object hypothesis) <bound verified subobjeci>

This rule adds a new rigidly attached subassembly to the current hypothesis. The location of the main assembly predicted by the new subassembly must be consistent with the previous hypothesis.

RULE 10: <object hypothesis) <- <object hypothesis) <bound verified subobject)

This rule adds a new flexibly attached subassembly to the current hypothesis. The location of the new subassembly, with the previous hypothesis's location and the model attachments, allows the calculation of the attachment relation that must exist between the structure and subassembly. If this is consistent with the explicitly modeled parameters, then the match is acceptable. Any remaining symbolic parameters are then bound to the corresponding calculated values, in the context of the resulting hypothesis.

7. Verification *and* Description

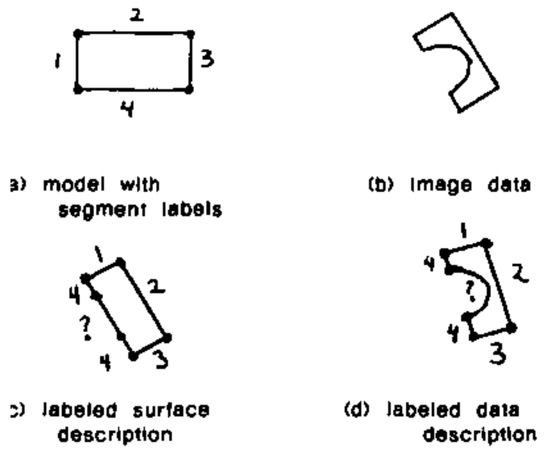
When an object hypothesis is fully Instantiated.

the verification rule is applied. This is necessary because the hypothesis construction rules are tolerant in their checking (e.g. location and orientation parameter consistency) and allow a few unlikely fully instantiated hypotheses to be formed. The major discrepancies noted were: use of one image region to support several distinct surface hypotheses and the use of widely separated image regions to support adjacent surface hypotheses.

Part of the verification phase needs a description of the surface-to-image match. Two descriptions are created, that of the surface hypothesis as supported by the image region data, and that of the image region as accounted for by the projected surface hypothesis. The descriptions are given by identifying segments of the boundary of either the image or projected model surface region in correspondence with the description of the surface database model (example below). This correspondence Is found by:

1. taking the boundary of the image region and calculating a similar boundary for the projected surface region (with segments identified by the model description labels),
2. identifying corresponding segments in the two boundaries by overlaying them and marking unlabeled segments, and
3. creating a description of the correspondence by recording segment identifiers and segment end-point locations (with some minor editing).

In the robot image example (section 9). there is a partially obscured surface in the robot lower arm assembly. In figure 6. we see how the model and image data are used to give the two descriptions.



2
1
3
4
a) model with segment labels
(b) Image data
1
2
4
?
4
3
c) labeled surface description
1
4
2
?
4
3
(d) labeled data description

Figure 6 - description of Image and matched surface boundaries

These descriptions are used partly in the general verification process, and partly for the handling of obscuration (section 8).

The total verification process can then be summarized as:

1. check for ail slots filled
2. ensure that all surfaces declared as back facing still are (necessary because of parameter adjustments)
3. ensure against duplicate use of image regions
4. re-optlmlze parameter estimates (given better averages estimates as the initial inputs)
5. describe visible surfaces (as above)
6. for all visible surfaces predicted to have shared boundaries (by the model), ensure that the portions of the surface descriptions corresponding to the shared boundary are compatible (i.e. overlap)
7. ensure that the object's final predicted boundaries coincide well with the observed boundaries. This includes the obscuration analysis discussed in section 8.

8. Coping with partially obscured objects

Any reasonable three dimensional scene analysis program has to cope with obscuration: objects always have backsides that the model predicts, but are never seen. There are two other forms of obscuration: the object may have self-obscured front surfaces (from either rigidly or flexibly attached subcomponents being closer to the viewer) and the object may be obscured by external, unrelated objects in the scene.

In the discussion below, only surfaces are considered. This is because the analysis has focussed on the use of surface descriptions, and because the objects can be described (directly, or recursively) in terms of their surfaces. Self-obscured surfaces are those obscured by other surfaces of the same object. External surfaces *are* those on unrelated objects.

Knowledge of the models, surfaces and the image formation process is the ultimate source of the Information needed to solve obscuration problems. Here, the initial use of this knowledge is during .surface description. Major deviations of the image boundary from the predicted model boundary are hypothesized to result from obscuring boundaries. The second usage is during object construction. If surfaces are only slightly obscured, then the model instantiation process proceeds normally. If they are greatly obscured, then the parameter estimation techniques will not be successful. Then, the presence of a likely image region is considered adequate at this point.

The backside surface case is easily handled by examining the predicted surface normal.

Completely obscured surfaces require evidence of closer surfaces, to support their hypothesized existence. For self-obscured surfaces, the evidence comes from the prediction of other object surfaces. Evidence for external surfaces must come from elsewhere. This case has not been Implemented yet. Possible evidence may come from: other verified object hypotheses or low level feature cues (e.g. as in Binford [1]).

Partially obscured surfaces get the greatest analysis. Because they are partially visible, they fill slots in the object hypothesis. As mentioned above, to do this, the hypothesis formation constraints are weakened. Unfortunately, weakening the constraints leads to the possibility of Incorrect hypotheses being formed. The major burden of coping with this and obscuration in general is in the verification process (rule 4). In addition to the verifications described in section 7. the routine has four obscuration-specific tasks:

- to ensure that points where the data description becomes unlabeled are points where three (or more) regions meet (figure 7). These regions are: object surface, obscuring surface and background surface. This is a variation of the TEE test for obscuring surfaces, except that it has advantages in cases of coincidental alignments, in that it is based on surface ordering properties, rather than boundary topology.

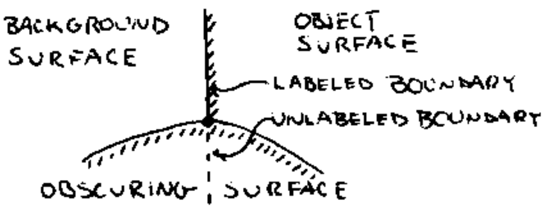


Figure 7_ 2.*three* surfaces at unlabeled segments

to ensure that unlabeled model boundaries are consistent with predictions of obscuring surfaces (figure 8). In particular, this means that the predicted missing boundary (from the model description) must lie completely outside the associated image region The hypothesis can be strengthened in the self-obscured case by showing that *the* missing segment lies within a region corresponding to a closer surface.

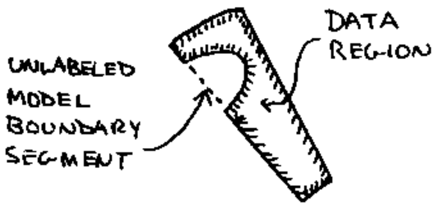


Figure 8 *2* unlabeled model boundaries

to verify that unlabeled data description boundaries are consistent with obscuring surfaces (figure 9). That is, the unlabeled data boundary must lie completely within the region predicted for the surface. Again, the hypothesis can be strengthened in the self-obscured case by showing that the unlabeled boundary lies on the boundary of a closer object surface.

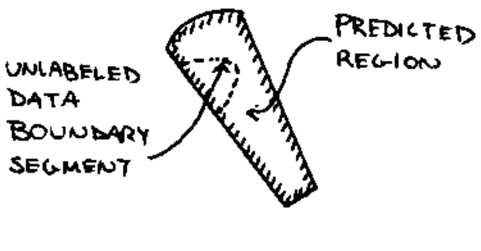


Figure 9 2 unlabeled data boundaries

- to verify that boundaries of undescribed regions coincide with predicted model boundaries (figure 10). This case has not been explored yet.

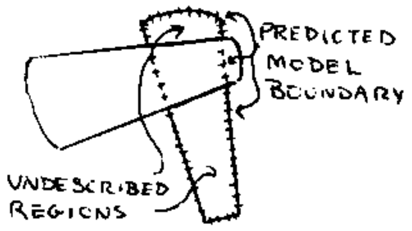


Figure 10 - segmented surface test

This type of reasoning complements that proposed by Binford [1], in that this uses the depth ordering properties of surfaces to provide clues for hypothesis verification. His approach uses similar reasoning for hypothesis formation.

9 Example

Figure 11 shows a picture of a model of the upper and lower arm assembly of a Unimation PUMA robot, with the boundaries of the hand-segmented surface regions. This image was used as the basis for the results discussed in this section. The location of the model's external coordinate frame origin and the value of the joint angle parameter are given below. The global coordinate system is centered at the focal point, with X horizontal. Y vertical and Z away from the viewer. All distances are in centimeters and angles *are* in radians

```
X:-28        Rotation:0.0     jnt1:0.85
Y:124        Slant:0.0        jnt2:0.84
Z:443        Tilt:0.0         jnt3:4.68
```

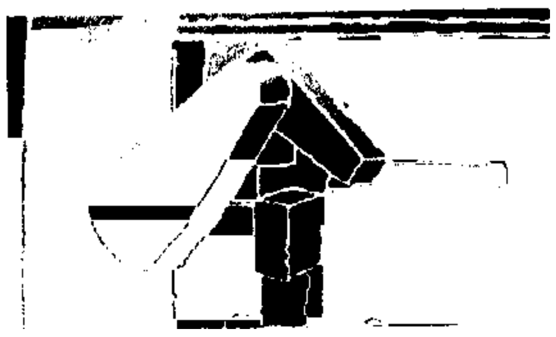


Figure 11 2 Hand segmented region boundaries on raw image

This image was hand segmented to give the major surface regions in the field of view. The segmented Image, the model shown in figure 1 and the meta-rules (e.g section 6) were then used as the external inputs Into the program. The output of the program is a database of the hypotheses at various stages of instantiation. The only fully instantiated robot hypothesis was the correct one. For that solution, the global coordinate and the corresponding joint angle parameters were:

X:-25	Rotation:0.12	jntl:0.83
Y:124	Slant:0.13	jnt2:0.88
Z:440	Tilt:5.56	jnt3:4.54

it is felt that the numerical solutions given by the program are reasonable. (The large tilt angle estimate discrepancy occurs because of the essentially zero slant estimate.) It Is not apparent, but the results at various intermediate stages are often a lot worse. The accuracy results from averaging estimates from the six visible surfaces of the robot. The predicted surface boundaries of the final robot hypothesis node superposed over the raw image are shown in figure 12 (The obscured edge is not removed due to lack of a front-surface hidden line remover.) The picture shows that the numerical estimates are approximately correct. The fitting of model surfaces to data regions aligned boundaries well, but when the parameters estimates from the various surfaces are combined, the boundary fit deteriorates, which shows In this figure.

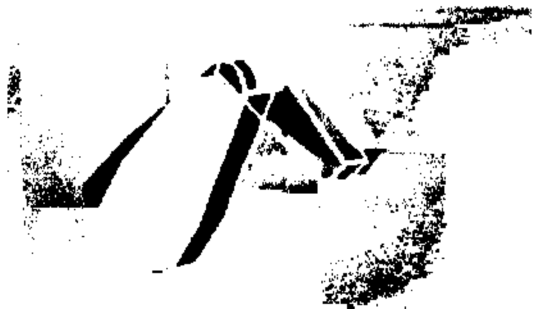


Figure 12 - Edges of located robot on raw image

10. Conclusions and criticisms

The results from section 9 show that the program successfully identified the robot assembly and made reasonable estimates of Its location, orientation and joint angle parameters. This can be attributed partly to the right choice of input data (I.e. surface regions and 3D object models), and partly to the types of reasoning done at the various stages of analysis.

The program can be criticized on several points. First. Its image surfaces must be largely planer for the current parameter estimation techniques to succeed. Second, the modeling of surfaces doesn't account for the surface shape internal to the region boundary. Third, the program doesn't take advantage of the fact that surface segmentations will probably provide two or three parameter estimates directly (slant.tilt.distance). Fourth, its models are non-generic (unlike ACRONYM), and there Is no simple mechanism (as yet) available for providing them. Fifth, its current dependence on hand-segmented images, while necessary at present, makes the program difficult to adequately evaluate. Sixth, the model Invocation process currently considers matching all surface regions to all reasonable surface models. Because any reasonable model base and Image will have many surfaces, a more powerful technique is needed. Seven. Its reasoning is much more oriented to structural, than to visual understanding. The two major uses of image data were in the parameter extraction and the structure verification processes, with the rest of reasoning devoted to matching according to the object models. Eight, it's reasoning Is largely undirected and thus leads to having a heavy computational cost. Finally, because it uses an exact parameter matching technique, with tolerance (threshold) based matching, rather than convergence of parameter ranges (c.f. ACRONYM), it is subject to complete failures when parameter estimates are inaccurate. However, even though generous tolerances were needed, the process was selective enough to only allow a few hypotheses to get close to the verification stage.

On the other hand, because It uses a more reasonable semantic primitive, surfaces, and because it uses constructive, task-specific reasoning, it can more completely recognize complex objects. In particular, the program successfully:

- located and identified a 3D object in a 2D image
- used surface regions and 3D object models to guide the process.
- extracted the 3D positional information. Including the joint angles of a flexible assembly.
- fully instantiated and explained the model of the robot assembly, and
- handled some cases of self-obscuration correctly.

it is felt that the work described in the paper describes advances int the following areas:

3D parameter estimation from surface image region to model region matching.
- successful recognition of articulated objects, and
- better understanding of handling obscured surfaces.

11. References

[1] Binford, T.O., "Inferring Surfaces From Images", Artificial Intelligence, Vol. 17, PP205-244, 1981

[2] Binford, T.O., "Survey of Model-Based Image Analysis Systems", Int J. of Robotics Research, Vol. 1, #1, pp 18-64, 1982

[3] Brooks, RA, "Symbolic Reasoning Among 30 Models and 20 Images", Stanford AIM-343, 1981

[4] Freuder, E.C., "A Computer System For Visual Recognition Using Active Knowledge", Proc. 5th Int Joint Conf. on AI, pp671-677, 1977

[5] Grimson, W.E.L., "From Surfaces to Images: A Computational Study of the Human Early Visual System", MIT Press, 1981

[6], Pentland, A.P.,, "Local Computation of Shape", Proc. European Conf. on AI, pp 199- 204, 1982

[7] Roberts, L.Q., "Machine Perception of Three-Dimensional Solids", in Optical and Electro-Optical Information Processing", pp 159-197, 1965