IDENTITY INDEPENDENT OBJECT SEGMENTATION IN 2½D SKETCH DATA

Robert B Fisher

D.A.I. RESEARCH PAPER NO. 285

Paper being presented at 1986 *European Conference on Artificial Intelligence*, Brighton, July, 1986.

IDENTITY INDEPENDENT OBJECT SEGMENTATION IN $2\frac{1}{2}$D SKETCH DATA

Robert B. Fisher
Department of Artificial Intelligence
University of Edinburgh
5 Forrest Hill, Edinburgh EH1 2QL, SCOTLAND, UK

Abstract: This paper describes how the surfaces in a segmented $2\frac{1}{2}$D sketch can be grouped into surface clusters according to rules derived from image boundary and solid object properties. The clusters designate complete, isolated visible object surfaces usable as a rough scene description for a blob-level vision system, or as input for more detailed recognition.

1.0 Introduction

A competent object recognition system clearly needs a figure/ground separation mechanism to indicate both which image features are related and the object's spatial extent. The bounding context also reduces feature relationship combinatorics, thus improving recognition performance. This paper shows how the explicit information in a surface image (e.g. $2\frac{1}{2}$D sketch) can be used to group surfaces to form a blob-level, identity-independent representation of the 3D solids in the scene. These surface clusters (SC) form the first object-level interpretation of an image and organize the surface information for later processing as part of object recognition ([FIS86]).

Initial work on object segmentation was based on recognizing the objects in a restricted domain and then isolating their features from the remainder of the scene (e.g. [ROB65]). Classical work in line labeling isolated distinct bodies using connection heuristics ([GUZ67]) or connected obscuring boundaries (e.g. [CLO71]). Waltz ([WAL75]) later extended this by adding crack and separable concave edge labels. Sugihara ([SUG79]) used light stripe data to acquire 3D surface information about a scene. Some labels (e.g. concave, convex, obscuring) could be directly assigned using a combination of stripe behaviour and valid label configuration rules. Two heuristics were proposed for separating objects in the labeled scene: (1) where two obscuring and two obscured segments meet, depending on a depth gap being detectable from either illumination or viewpoint effects and (2) along concave boundaries terminating at special types of junctions (mainly involving two obscuring junctions). These are subcases of the obscuring segment rule proposed below.

None of the methods could segment a cube lying flush in a corner. Further, when a laminar surface curves or is creased back to obscure itself, the surfaces are separated by an obscuring boundary but should still be considered connected.

2.0 Assumptions about input data and model segmentation

The input is the labeled segmented surface image, a $2\frac{1}{2}$D sketch-like structure, which records scene depth and surface orientation in a viewer based image registered like the usual intensity image. It is segmented into regions of similar shape property, at surface depth, orientation and curvature discontinuities, etc. The boundaries resulting from this are labeled according to the type of segmentation ([FIS86]), and are annotated as being front surface or back surface (occluding) or convex or concave (shape discontinuity). A surface orientation boundary that is both concave and convex in places is broken up during image segmentation.

For this paper, the test data represents a combination of hand measured and real data. An intensity image was used to guide manual segmentation into the different surface shape regions. Figure 1 shows the labeled region boundaries for the scene and the referenced surface regions.

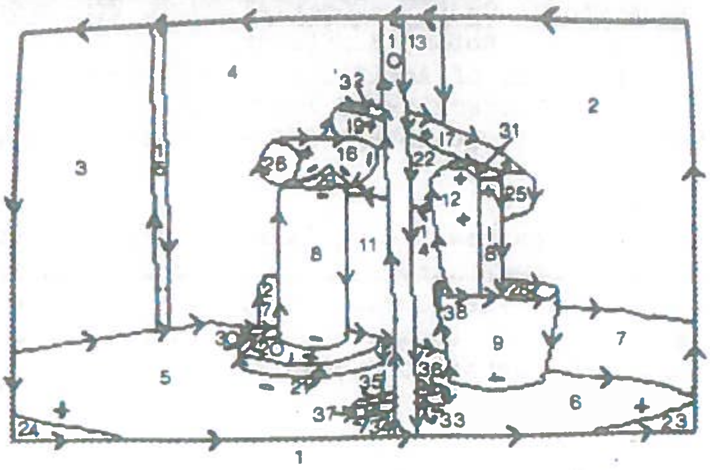
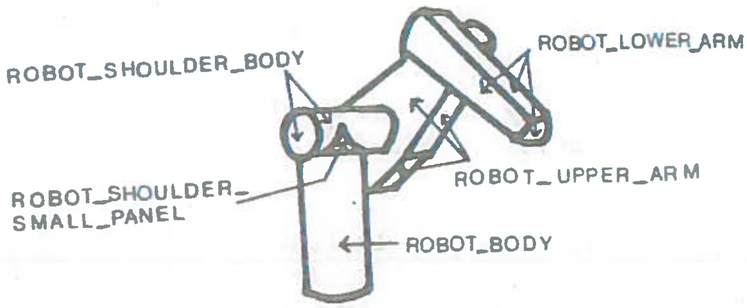


Figure 1: Labeled Boundaries for Test Scene

Arrows indicate obscuring boundaries with the front surface to the right. Plus and minus indicate convex and concave shape discontinuities respectively.



Larger Assemblies	Components
robot_upper_assembly	lower_arm + upper_arm
robot_shoulder	robot_shoulder_small_panel + robot_shoulder_body
robot_link	robot_upper_assembly + robot_shoulder
robot	robot_link + robot_body

Figure 2: Labeled Robot Model

To achieve correspondence between model and data features, similar groupings must be achieved for both models and data. So model surfaces are segmented by the same criteria as the data surfaces. Then, concave surface shape discontinuities isolate groups of model surfaces making convex volumes the primitive model features. Larger features are generated by joining smaller features hierarchically. A wine glass thus segments into a bowl, a stem and a base. Clearly, this does not always work, such as where the back of a dinner chair is flush with the legs.

Figure 2 shows the robot model used for this example, with the primitive segmented assemblies indicated and the larger aggregated assemblies listed below.

3.0 "Why Surface Clusters?"

Surface clusters (SC), are maximal sets of related data surfaces such that any one surface is adjacent to at least one other within the set via a suitable connecting boundary. They recreate the complete, visible, 3D portion of each distinct object's surface.

The surface cluster is a new representation. It is a "blob" level interpretation, in which there are unidentified solid objects with approximate spatial relationships and is useful for some tasks such as navigation or object avoidance. Such interpretations are needed for unidentifiable objects, whether because of faults or lack of models in the database. They provide an intermediate level of interpretation upon which further interpretations can be built. Surface clusters help bridge the conceptual distance between the object and the image. With this structure, the key image understanding representations now become: image - primal sketch - surface image - surface clusters - objects

Surface clusters may be incomplete, as when an object is split up by a closer obscuring object, though surface hypothesizing may bridge the occlusion. They may also be over-aggregated - from images where there is insufficient evidence to segregate two objects. The goal of the process is to produce a partitioning without a loss of information. These failures may reduce recognition performance (i.e. speed), but not its competence (i.e. success): incompletely merged SCs will be merged in a larger context and insufficiently split SCs will just cause more searching during model directed matching.

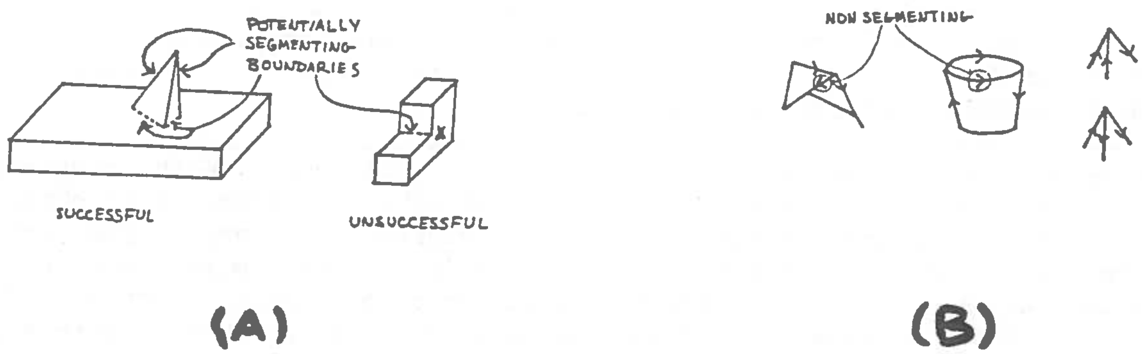


Figure 3: Surface Segmentation Boundary Cases

4.0 Creating Surface Clusters

This process aims to produce minimal clusters of image surfaces corresponding to distinct model components. The solution is conservative, because it avoids splitting the components at the expense of merging distinct adjacent objects. This produces contexts guaranteed to contain complete segmented model primitives (as defined in section 2), but may contain more than one.

4.1 Determining Segmentation Boundaries

Whenever one object sits in front of or on top of another, the intervening boundary is always either concave or obscuring. Separate objects may be on opposite sides of a concave boundary, but it is indeterminate whether the two surfaces are joined or merely contact. So, the conservative approach suggests that this boundary is provisionally segmenting. Assuming concave boundaries always segment leads to contradictions as seen in figure 3a, where there is no reasonable shape boundary at point X to continue segmentation, so only the "provisionally segmenting" label is used at this point.

Obscuring boundaries usually give no cues to the relation between opposing surfaces (other than being depth separated), so these usually separate. Connectivity holds across obscuring boundaries when laminar surfaces fold back on themselves. Figure 3b shows a leaf folded over and the two surfaces of a trashcan, where the surfaces are connected even though an obscuring boundary intervenes. What distinguishes this case is the presence of the arrow vertex shown at the right. Viewpoint analysis shows that these are the only special cases for laminar surfaces.

4.2 Finding Primitive Surface Clusters

Primitive surface clusters are formed by collecting all surfaces regions that have any direct or transitive connection, once all provisionally segmenting boundaries are identified. Surface clusters are maximally connected to provide complete contexts for object subcomponents, so if there is doubt, then connect. This contrasts with model segmentation, where if there is doubt, then segment. Then, all model features will lie completely within some data context.

4.3 Depth Aggregation of Surface Clusters

One goal of the SC process is to associate all subcomponents of an object in some SC. Aggregation is necessary because self-occlusion may segment the visible portions of an object into several depth levels. In the test image, the robot lower_arm obscures the upper_arm, so parts of the robot_upper_assembly will appear in different primitive SCs. Depth aggregated SCs provide the context for complete objects.

Merging all SCs behind a given SC is not a solution, because more than one subcomponent may be in front of the linking component. Similarly, merging all SCs in front fails if the object has several components behind. To solve this problem, a combinatorial solution was adopted.

Certain sets of SCs can be initially grouped into equivalent depth clusters. These occur when either SCs mutually obscure each other, or there is no obvious depth relationship as when across a concave surface boundary. The robot lower_arm and trash can SCs mutually obscure in the test image. When these cases occur, all such primitive SCs are merged into a single cluster. The combinatorial depth merging process applies only to these equivalent depth SCs.

The computation producing the equivalent depth clusters is:

Let:

$\{P_1, \ldots P_n\}$ be the primitive surface clusters

front(P_i, P_j) is true if P_i is in front of P_j, which is true if there is a surface in P_i with an obscuring relation to a surface in P_j

beside(P_i,P_j) is true if P_i is beside P_j, which is true if not front(P_i,P_j) and not front(P_j,P_i) and there is a surface in P_i that shares a concave boundary with a surface in P_j

$\{E_1, \ldots E_m\}$ be the equivalent depth clusters

$E_i = \{P_{i1}, \ldots P_{is}\}$,

Then:

(1) If $E_i \wedge E_j \neq \emptyset$ then $E_i = E_j$

(2) for any $P_{ia} \in E_i$ there is a $P_{ib} \in E_i$ such that:

front(P_{ia},P_{ib}) and front(P_{ib},P_{ia})
or
beside(P_{ia},P_{ib}) [and not front(P_{ia},P_{ib}) and not front(P_{ib},P_{ia})] redundant.
or
sizeof$(\ E_i\ ) = 1$

Then, the depth aggregated SCs are sets of equivalent depth SCs:

Let:

direct(E_i,E_j) be true if surface cluster E_i is directly in front of surface cluster E_j, which occurs if there are primitive surface clusters $P_{ia} \epsilon E_i$ and $P_{jb} \epsilon E_j$ such that front(P_{ia},P_{jb})

linked(E_i,E_j) if direct(E_i,E_j) or direct(E_j,E_i)

$\{D_1, \ldots D_n\}$ be the depth aggregated clusters

$D_i = \{E_{i1}, \ldots E_{it}\}$

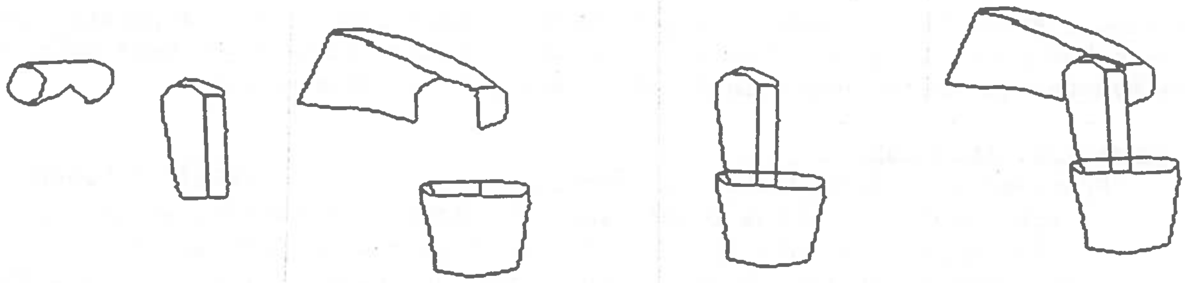
Then:

for any $E_{ia} \epsilon D_i$ there is a $E_{ib} \epsilon D_i$ such that linked(E_{ia},E_{ib})

That is, there is a chain of relationships between the equivalent depth SCs. The background (e.g. surfaces that lie behind all others) is omitted.

5.0 Evaluating the Surface Cluster Formation Process

Using the surface hypotheses for the test image, some SCs are shown in figure 4. The data required a minor intervention to produce correct behavior because the boundary between the robot body (region 8) and the robot shoulder (region 29) was a crack and was forced to act as a concave boundary. This allowed the body to be depth equivalent with the shoulder, which is appropriate.



(a) PRIMITIVE (b) DEPTH EQUIVALENT (c) DEPTH MERGED

Figure 4: Some Surface Clusters From the Test Scene

Below, there is a listing of the SC to model component correspondences for the test image. Clearly, the SC formation process groups the data surfaces into what corresponds to the structural units defined in the model.

This example shows that the SC formation process is successful in a variety of circumstances and is not limited to just planar surfaces. All the primitive and most of the larger SCs corresponded with object features. Primitive model assemblies were completely contained in single primitive surface clusters and larger assemblies were completely contained in larger surface clusters. The only irrelevant surfaces in the main SCs were those of the trashcan which was intertwined with the robot. Though several SCs contained multiple assemblies, this causes no recognition failures, only greater matching effort.

The combinatorial formation of depth merged SCs is a problem. Here, the number of depth merged SCs was not excessive as the object also has a strong depth order, so 2 of the 6 corresponded to ASSEMBLYs. If more objects had been behind, even more SCs (roughly $O(2^n)$) would have been created.

Table 1: SC to Model Correspondence

SC	TYPE	REGIONS	MODEL
1	PRIMITIVE	20,21,30	
2	PRIMITIVE	27	
3	PRIMITIVE	16,26	robot_shoulder_body
4	PRIMITIVE	8	robot_body
5	PRIMITIVE	29	robot_shoulder_small_panel
6	PRIMITIVE	33,34,35,36,37	
7	PRIMITIVE	12,18,31	robot_lower_arm
8	PRIMITIVE	9,28,38	trashcan
9	PRIMITIVE	17,19,22,25,32	robot_upper_arm
10	EQUIVALENT	20,21,27,30	
11	EQUIVALENT	8,16,26,29	robot_shoulder + robot_body
12	EQUIVALENT	9,12,18,28,31,38	trashcan + robot_lower_arm
13	DEPTH	9,12,17,18,19,22, 25,28,31,32,38	robot_upper_assembly
14	DEPTH	8,16,17,19,22,25,26,29,32	
15	DEPTH	8,9,12,16,17,18,19,22, 25,26,28,29,31,32,38	robot_link + robot
16	DEPTH	8,16,20,21,26,27,29,30	
17	DEPTH	8,16,17,19,20,21,22,25, 26,27,29,30,32	
18	DEPTH	8,9,12,16,17,18,19,20,21,22, 25,26,27,28,29,30,31,32,38	

The theory could be extended easily to make "crack" type lines separate (e.g. between the robot body and shoulder) and using Waltz labeling on separable concave boundaries could remove inappropriately labeled boundaries.

To summarize, this paper:

- Proposed an intermediate representation (Surface Clusters) between the $2\frac{1}{2}$D sketch and the model based 3D object hypotheses, which segments the image data into blob level, identity independent solids.
- Elaborated rules for producing these surface clusters, including objects with curved surfaces and some laminar surface groupings.

7.0 references

[CLO71] Clowes, M. B., "On Seeing Things", Artificial Intelligence, Vol. 2, pp79-116, 1971.

[FIS86] Fisher, R. B., "From Surfaces to Objects: Recognizing Objects Using Surface Information And Object Models", University of Edinburgh, PhD thesis, forthcoming.

[GUZ67] Guzman, A., "Decomposition of a Visual Scene Into Bodies", MIT AI memo #139, 1967.

[ROB65] Roberts, L. G., "Machine Perception Of Three-Dimensional Solids", Tippett, J. T. (ed.), Optical and Electro-Optical Information Processing, MIT Press, Ch. 9, p159-197, 1965.

[SUG79] Sugihara, K., "Automatic Construction Of Junction Dictionaries and Their Exploitation of the Analysis of Range Data", Proc. Int. Joint Conf. on Artif. Intel. 6, 1979.

[WAL75] Waltz, D, "Understanding Line Drawings of Scenes with Shadows", Winston, "The Psychology of Computer Vision", pp19-91, 1975.

Acknowledgements: This work was supported by a University of Edinburgh studentship. Thanks to B. Beattie, J. Hallam, J. Howe and M. Orr for suggestions.