This is to certify that the

dissertation entitled

OBJECT RECOGNITION FROM RANGE IMAGES

presented by

Richard Lee Hoffman

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Computer Science

Anil Kumar Jain
Major professor

Date September 9, 1986

OBJECT RECOGNITION FROM RANGE IMAGES

By

Richard Lee Hoffman

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science

1986

ABSTRACT

OBJECT RECOGNITION FROM RANGE IMAGES

By.

Richard Lee Hoffman

The recognition of objects in 3-dimensional space is an essential capability of the ideal computer vision system. Range images directly measure 3D surface coordinates of the visible portion of a scene and are well suited for this task. We report a procedure to identify 3D objects in range images, which makes use of four key processes. The first process segments the range image into "surface patches" by a clustering algorithm using surface points and associated surface normals. The second process classifies these patches as planar, convex, or concave based on a nonparametric statistical test for trend. The third process derives patch boundary information, and the results of this and the second process are used to merge compatible patches to produce reasonable object faces. The fourth process takes the patch and boundary information provided by the earlier stages and derives a representation of the range image. A list of salient features of the various objects in the database forms the core of an object recognition system, which looks for instances of these features in the representation. Occurrences of these salient features are interpreted as evidence for or against the hypothesis that a given object occurs in the scene. A measure of similarity between the set of observed features and the set of salient features for a given object in the database is used to determine the

identity of an object in the scene or reject the object(s) in the scene as unknown. This evidence-based object recognition system correctly identified objects in 30 out of 31 range images. Four range images showing objects not included in the object database were rejected, as desired. Recognition degraded only slightly when evidence weights were perturbed.

For mom and dad

ACKNOWLEDGEMENTS

I wish to thank my guru and thesis advisor Professor Anil K. Jain for his guidance and encouragement in the development of this thesis. I am very grateful for his persistence and patience during my apprenticeship in the art of scientific research and precise writing. His critiques have contributed greatly to the quality of this manuscript.

I also wish to thank Professor Richard C. Dubes for invaluable guidance and support towards my professional development, and Professor George C. Stockman for helpful advice and guidance in the research of this thesis. I thank Professor Jacob Plotkin, who has been associated with me in one way or the other throughout my graduate career, for his advice and for teaching me the fundamentals of mathematics and computer science in many excellent courses.

During my lengthy stay with the Pattern Recognition and Image Processing (PRIP) group at MSU I have been fortunate to be acquainted with many of its past and present members. I would especially like to thank my officemates through the years, Dr. Gautam Biswas, Dr. Xiaobo Li, Dave Ittner, Chaur-Chin Chen, Joe Miller, Jean Vincent Moreau, and Pat Flynn for many helpful discussions and critiques of my work. Acknowledgement is due also to the system managers of our PRIP lab for their vigilance in maintaining our system, Phil Nagan, Pat Cameron, Dongyul Ra, and Henry Davies, and to other PRIP members for diverse assistance: Dr. Steve Smith, Dr. Ardeshir Goshtasby, Dr. Vernon Rego, Zhisheng You, Guangshou Zeng, Lixen Mann, Gongzhu Hu, Mark Jones, Bill

Baer, Sei-Wang Chen, Ywhyng Lee, and Jill Giampa. Last but not least, I wish to acknowledge the Emmy Lou and Franco Harris 500 computers for their faithful years of service.

I also wish to acknowledge the financial support I received during my doctoral studies from NSF Grants ECS-8007106 and ECS-8300204, and the Department of Engineering Research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

f	range image function;
r,c	row, column indices;
p,q	pixels;
$p_{r,c}$	pixel at row r, column c;
$\bar{p}$	3D vector corresponding to spatial location of pixel p;
g	unit of noise level;
s	shift value (background removal);
N_p	number of object pixels
$\bar{n}=(n_x,n_y,n_z)$	unit surface normal vector;
Ω	neighborhood of a pixel;
Δ	maximum difference for jump edge detection;
(d_i)	distance sequence;
w	sensitivity constant for jump edge detection;
U	set of connected non-jump pixels (surface normals);
a_i	coefficients for polynomial fitting;
f_s	sampling frequency of pixels for segmentation;
CLAVGD	average within-cluster interpoint distance;
S, S_{ave}	isolation and compactness measure of clusters;
c_m, G_m	center and membership set of cluster m;
C_i	cluster i;
θ_i^r, θ_i^c	directional increments;
L	label image function;
E_j, E_j^*	patch-extent function in direction j;
P	surface patch;
Ψ,η	step size and number of steps for trend test;
$\lfloor x \rfloor$	greatest integer less than or equal to x;
$\bar{\gamma}_i$, $\gamma(t)$	surface slice function;
(r_i)	rank sequence;

$S_j, \bar{S}_j$	trend test statistic values;
T_j	trend test result;
e_1, e_2, e_3	eigenvalues;
E	eigenvalue analysis statistic;
ν	difference of normals values;
$s(\bar{p}, \bar{q})$	sign factor;
D	difference of normals statistic;
$T(\cdot)$	trend decisions cardinality function;
EI	eigenvalue decision;
DN	difference of normals decision;
$\hat{n}_i$, $\bar{N}_i$	for average unit normal derivation
$N(i,j)$	normal angle;
B	border pixel set;
ℓ_j	population of border pixel set;
ρ	probability used in crease edge detection;
β	crease edge detection statistic;
$\bar{\chi}$	spatial points to calculate surface area;
R_i	representations;
Ξ	evidence condition;
H_i	hypothesis that object is model object i;
$\perp$	instance vector;
$\Phi_i = \Phi_i^+ + \Phi_i^-$	observed evidence magnitude vector;
ξ_i^+	positive evidence vector;
τ_i	similarity value;
$\hat{i}$	identified object index;
$\mathbf{E}$	matrix of degree of evidence weights

CHAPTER I
INTRODUCTION

The task of endowing a computer with a visual capability to sense, recognize, and interpret its surroundings has made the field of Computer Vision a challenging research area over the years. The ease with which humans perform day-to-day visual tasks has been remarkably difficult to reproduce using computers. Only simple tasks in restricted environments have yielded any success when assigned to computers. Particularly difficult is the implementation of three-dimensional vision, yet this task is very important for a system to navigate and operate in the real world.

This chapter briefly covers the methods and results involved in the 3D object recognition system developed in this thesis. Related 3D object recognition systems in the literature are discussed. Approaches to depth perception and object representation are also discussed, and the basic structure of the thesis is outlined with corresponding methodology and contributions.

1.1 -- 3D Object Recognition

The aim of this work has been to develop a computer vision system to recognize 3D objects in a scene. Implementation of complete recognition systems is not common in the literature; some complete systems are discussed in Section 1.2. The input to our system is a range image [Jar83], which supplies arrays of 3D spatial data which correspond to points on object surfaces. Segmentation of object surfaces into surface patches is performed with the goal of detecting natural object faces. These patches are primitives for the object recognition procedure, which is based on matching salient features of objects with observed features. The salient features of known objects are stored in an evidence rule base as a set of evidence conditions with corresponding evidence weights. This evidence rule base is currently constructed by hand.

A few assumptions have been made:

* There is only one object in a given scene.
* Natural object faces are smooth. Hence information will not be lost when smoothing is performed to reduce noise.
* The system need not perform object inspection. By looking only for notable features of objects, some imperfections can easily be overlooked.

A set of 31 range images has been analyzed, each showing a view of one of 10 objects. Under initial segmentation, we found that 46% of natural object faces were correctly identified as single surface patches. After applying a merging procedure to help recover from oversegmentation, this proportion increased to 59%. In addition, 61% of surface patches are correctly classified as planar, convex, or concave. Therefore, preliminary conversion of the numerical range data into symbolic patch information was shown to be

moderately accurate for detecting and classifying natural object faces.

Based on the surface patches and corresponding classifications passed to the object recognition process, objects in 30 of the 31 range images were correctly identified. Four range images showing objects not included in the object database were rejected, as desired. Recognition degraded only slightly when evidence weights were perturbed. The evidence-based object recognition procedure which has been developed here performs well.

1.2 -- Background

Although it could be said that the technology of range image sensors is in its infancy, the general problem of range image processing has been well covered in the literature in the stages of acquisition, analysis, and recognition. Few of the results report complete vision systems that implement all three of the aforementioned tasks; however, each task is in itself so fertile that many papers concentrate on only one or two of these tasks. This section will present some of the complete range image based vision systems that occur in the literature; detailed surveys of specific topics related to range images are presented in appropriate chapters of this thesis.

Oshima and Shirai [Osh83] use plane projection structured light to obtain range values. A region-growing process identifies approximate planar regions, which are classified as curved, planar, or undefined, and merged with compatible neighbors to obtain global planar and curved regions. Each region is characterized by the best fit plane or quadric surface, relations to neighboring regions, and other features which are then compared to features of a priori models to obtain object recognition. This approach

is used to identify regular solids and various machine parts. Natural object faces are assumed to be delineated by creases in object surface; this validates the region-growing process, but objects with smooth joins [Bra85], such as many sculptured objects, cannot be handled. The recognition technique is based on characteristic views: many sample views of an object are shown to the system, and to recognize an object the observed scene must be compared to all of these views.

Horaud and Bolles [Hor84] describe a two-step recognition process designed to identify objects within a jumble of objects. The first step identifies jump edges, classifies these edges as convex or concave, examines the visible surfaces on either side of the edge to classify and parameterize them as planar or cylindrical (the only types of object primitives assumed to exist). The next step is a tree search strategy to find an object match; curved edges are preferred because they are more useful in limiting the search space. An obvious problem with this approach is the dependence on jump edges in deriving primitives for recognition, so that objects must be rich in jump edges. Furthermore, the restriction of object surfaces to planes and cylinders reduces the class of objects which can be recognized.

Boyter and Aggarwal [Boy86] work only with polyhedral objects. Planar faces and corresponding borders are detected by first fitting a piecewise linear function to each row of the range image, and then integrating these row-functions. For each model object, a number of translation-rotation matrices (TRM's) which transform a portion of the observed object into the model is computed. A TRM is saved if it transforms the model object into the observed scene with small error. A Hough technique selects that TRM which has occurred most often to provide object identification and location.

The evidence-based recognition procedure implemented in this thesis has been designed to avoid some of the drawbacks of the above object recognition systems:

(1) Fitting quadric surfaces to object faces, and
(2) Restricting the class of objects which can be recognized to those composed of simple primitives.

The use of quadric surface equations for recognition is a dubious endeavor, as is demonstrated in Section 3.2; therefore, we have restricted our classification of a surface patch to one of three fundamental classes: planar, convex, and concave. This broad classification requires no specific shape of a surface, and therefore can be applied to sculptured objects as well as to objects composed of simple primitives.

1.3 -- Depth Perception

A single reflectance image will show the projection of 3D scene coordinates onto a 2D image plane; the inverse reconstruction problem of recreating the true 3D coordinates from this image generally suffers from the existence of many conflicting solutions. For the recognition of objects situated in 3-dimensional space it is essential that depth information be derived from the images which enter the system. Once this depth information is obtained it is possible to construct a 2.5-D sketch [Mar82] of the scene. This typically consists of depth values and surface orientations (representable as surface normals). This representation provides explicit information about visible surfaces and is the last stage of precognition in which no specific assumptions or hypotheses about the objects present in the scene are needed.

There are three major techniques for deriving depth information from a scene [Jar83]: ambient systems, structured lighting systems, and time-of-flight rangefinders. Ambient systems depend only on existing lighting; the images used are reflectance images and depth is implicit in the data -- that is, nontrivial procedures must be used to deduce the depth values. Structured lighting systems impose a pattern of light on the imaged scene and require less processing to acquire depth information than ambient systems. Time-of-flight rangefinders directly provide depth values; we call the output of such sensors range images. The following three subsections discuss these techniques in more detail.

1.3.1 -- Ambient Systems

Monocular or binocular cues can be used to deduce depth from reflectance images. The human visual system certainly uses both, since depth in a photograph can often be inferred as easily as the depth in the corresponding real world view. Some monocular cues which have been used in the literature (e.g., [Ros78]) to detect depth are:

(1) Linear and size perspective: Parallel lines converge as they recede into the distance; objects look smaller at further distances.

(2) Occlusion: If object A obstructs part of the view of object B, object A is in front of object B, i.e. closer.

(3) Shading: Object surfaces facing away from the light source(s) will be darker than those facing the light. A shadow of one object falling on another object is a form of occlusion. If it is assumed that the object surfaces are Lambertian, a precise formula relating grey level to surface orientation (with respect to the angle of incidence of the light source) can be used.

(4) Texture density gradient: A texture will become

finer in resolution as it recedes from the viewer.

(5) Motion cues: Moving objects which are further away move slower across the retina.

Note that some perspective cues require that objects in the view be known a priori before depth may be determined. This forms a vicious circle if depth is required to identify objects.

An easier approach utilizes two or more views of the same scene to deduce depth by binocular cues. The general technique is to derive point-by-point correspondences between two images to establish a disparity measure at each point. Knowledge of the positions of the cameras used then allows depth to be calculated by triangulation. The challenging task confronting CV research is a reliable and quick correspondence (image registration) algorithm. Work by Bela Julesz [Jul71] has been instrumental to the study of human stereo vision, particularly his experiments with random dot stereographs -- pairs of images that singly appear to be nothing more than random dot patterns but when viewed through a stereoscope reveal structured depth patterns. Figure 1-1 shows an example of such a stereo dot pattern. If viewed with a stereoscope, a floating "H" should appear. Thus depth perception needn't depend on high-level cues such as object recognition; depth may be detected solely from the disparity between the stereo images.

A multitude of algorithms have been developed to solve the correspondence problem. Of the various approaches, area-based correlation and edge-based correlation techniques dominate the literature. Below is a discussion of each of these general approaches. Other techniques that have been applied include relaxation [Bar80] and a hypothesize-test paradigm [Wil80]. In almost all situations, the two-camera model is designed such that there is no relative rotation between the views.

Figure 1-1
Stereo Dot Pattern

In area-based correlation the general approach is to pick a point in one image and search through the other image to find the optimal match point, basing the goodness of the match on various grey value or texture statistics of the neighborhood of each point. Thus a window about the point to be matched is correlated with windows of equal size in the other image to find this optimum match. This is generally time consuming, unless it is known a priori that the images have one axis in common. An "interest" operator can be used to identify those points with high local variance of grey levels, and these "distinguished points" are correlated first to help guide the correlation of the remaining points [Nev76],[Gen77].

The foundation of edge-based image matching lies in identifying the edges in the images, perhaps by the presence of zero-crossings in the second derivative of intensity. It has been claimed that the best filter to perform this second derivative is the Laplacian of the two-dimensional Gaussian distribution, and this is typically approximated by a difference of Gaussian (DOG) filter [Mar80]. Horizontal scan line pairs are then independently processed, correlating pixels belonging to an edge in one image with corresponding edge points in the same horizontal scan line of the other image by considering local information such as intensity slope and contrast, and the sense of the zero-crossing ("+" to "-" or "-" to "+"). In [Mar79] and [Gri80] edge maps are constructed by four masks of differing sizes; the results of matching image pairs filtered by the wider masks are used to bring the images into better correspondence so that the finer masks, which detect smaller disparities at higher resolution, are effective.

The photometric stereo technique [Ike81] uses two or three views of a scene taken with the camera in a fixed position but with different positions of the light source. Differences in shading from one image to the other provide cues about depth. Another potential ambient range-finding process is range from focusing [Jar76],[Kro86]. The sharpness of focus has to be quantified over a small image window for several focus settings, and the depth is derived by maximizing certain measures of sharpness. Disadvantages of this scheme are (1) accuracy decreases with distance, and (2) homogeneous regions cannot be ranged.

1.3.2 -- Structured Lighting Systems

The use of an additional (high energy) light source to project points, lines, or grids onto the scene being imaged allows depth to be deduced from a single 2D image by correlating the known projector model with the detected light pattern in the image.

The simplest technique is the projection of a single point onto the scene [Nim82],[Ode80]. Location in the image of the beam where it hits the scene will directly provide depth through triangulation. Hence this is almost an explicit ranging technique, except that certain special cases (which may occur rather frequently) must be appropriately handled. These cases are:

(1) The point of intersection of the beam with the scene is occluded from the camera;

(2) The beam is reflected off the scene in such a way as to produce multiple spots of light in the camera image;

(3) The spot may be elongated if the surface is almost colinear with the beam.

Several industrial vision systems use a spot laser beam to determine the position of objects [Bam83].

Another way is to project a line (a plane of light) onto the scene [Osh79]. It is possible to obtain depth for each point in the resulting line image without solving a correspondence problem [Kre82]. A range image, dense except in shadows, may be derived by sweeping the light plane over the scene. This is the principle behind the commercially available White scanning system produced by Technical Arts Corporation.

Finally, multiple plane and grid projections have been used as well [Hal82],[Sto85]. However, these require that a correspondence be made between the projected and imaged planes and grids. This correspondence is sometimes solved by encoding the light planes (e.g. dashed lines, colors) [Ino84],[Pos82].

Inherent disadvantages of these techniques are:

(1) Without a coherent light source, accuracy at larger depths decreases due to beam spread;

(2) Due to the special lighting setups required, these techniques are not useful for outdoor scenes.

1.3.3 -- Time-of-Flight Rangefinders

Time-of-flight rangefinders typically use one of two types of energy sources: ultrasonic sound or laser beams. The procedure is to measure the time taken for an outgoing ray to strike and reflect off the scene, and return to trigger a detector. A major difficulty with such a method occurs if the scene surface acts like a mirror surface and reflects very little of the ray back toward the detector unless the angle is just right. This is called specular reflection.

A commercially available ultrasonic rangefinder is produced by Polaroid. However, the resolution is not very good (10x10 pixels over a 90 degree solid angle field of view at best), and the wavelength of the sound is so large that specular effects are pronounced.

Laser rangefinders can operate in three ways: time-of-flight (TOF) systems measure the elapsed time between emission of the laser beam and detection of its reflected energy; an interferometric (IM) system counts interference bands when the reflected light is added to a reference beam; and in a continuous wave (CW) system the laser is amplitude modulated, and the phase of the returned signal is compared to that of the original signal to deduce depth. The Environmental Research Institute of Michigan has a laser range finder system which uses the CW method, and was used to obtain the real range images in this thesis.

A large disadvantage of a TOF system is in the accuracy, or resolution, which is typically on the order of a meter. Although its accuracy is on the order of the wavelength of the light used, the IM system requires a strong return signal and counting the interference bands is time consuming. A CW system has the better characteristics of the TOF and IM systems: good accuracy, which can be on the order of 0.0005", and high speed.

1.4 -- Object Features and Representations

There are a number of features which can be used to recognize 3D objects. For example, global characteristics such as size and shape are useful. Edges on object surfaces may show self-occlusion or creases in the surface; a network of such edges can provide strong evidence of certain objects. Segmentation of object surfaces into natural object faces provides a number of features: there are color, texture, and

shading of each patch, as well as spatial characteristics of natural object faces such as shape, size, and relationships to other faces. Objects may also be decomposed into 3D primitives, such as spheres, boxes, and cylinders, so that recognition may follow from these primitives and their interrelationships.

Features such as edges, surfaces, and volume primitives can be used to construct representations of objects. The means of constructing and operating on these representations is commonly called modeling. Representations (or models) form the link between image processing and image understanding.

Three-dimensional representations have been a topic of intense interest over the years. A big portion of this interest comes from Computer Aided Design and Computer Aided Manufacturing (CAD/CAM) where a quick alternative to manual design and manufacture (e.g. design of an automobile body) is already making a significant impact on productivity and quality [Req80]. Some issues in 3D modeling are:

(1) Domain: What is the universe of objects that can be modeled under a representation scheme?

(2) Validity: Is the scheme safe from nonsensical representations?

(3) Uniqueness and completeness: Is the mapping from representable objects to representations one-to-one?

(4) Conciseness: How "verbose" must a representation be?

Models of 3D objects fall into three general groups:

(1) Volume representations: Objects are represented in terms of 3D primitives (cell decomposition, spatial occupancy, generalized cylinders, etc.) [Nev77] or set-theoretic combinations of primitives (Constructive Solid Geometry, or CSG) [Bad79].

(2) Boundary (surface) representations: Objects are represented by collections of "surface patches" forming the boundary of these objects (e.g. Coons patches, B-splines, planar and quadric patches, etc.) [Rad84],[Tom84].

(3) Wireframe (edge) representations: Objects are represented by the set of edges (lines of intersection of smooth surfaces) [Gu84].

A difficulty with volume representations is that to recognize an object via matching "volume-based representations", many views of the object must be used, because there is uncertainty in the extent of an object in a direction parallel to the line of view. However, the human visual system can identify objects given only a partial view; this capability is shared by matching via "surface-based representations" in which matching only observed surface patches with those of a stored model can cue recognition. Volume-representational models are based on a small set of primitives, which either naturally excludes from their domains sculptured objects [Bad79] or allows sculptured objects at the expense of approximating the object with a large number of primitives [Chi85]. Wireframe models have many notable shortcomings, such as verbose representations, ambiguity, and nonsensical representations [Bes85]. The information contained in a wireframe model may be derived from a surface-based model, since edges on an object define boundaries between adjacent surface patches.

All three representation types have been employed in CAD. In fact, recent work in the CAD community is favoring multirepresentational schemes (e.g. CSG/boundary schemes in [Boy79]) such that advantages of each individual representation may be enjoyed [Req82]. However, new issues relating to conversion procedures between representations must be addressed.

The goals of the object recognition system developed here are:

* Make use of characteristic features of objects. This may involve view dependent recognition rules.
* Recognize objects from a single view.
* Allow objects to be sculptured or articulated.

Since recognition is to be made from a single view of an object, a volumetric approach is not reasonable. Thus the approach taken by our 3D object recognition scheme is based on surface/boundary representations, with wireframe information available as intersections of surface patches. The representations obtained for recognition are derived from information which may be computed from surface patches and boundaries between patches. However, recognition does not use models of objects, but only lists of salient features of objects, which are expressed in terms of natural object faces. A salient feature of an object is some characteristic which provides supportive evidence of that object. For example, if a cylindrical surface component is detected on an object, then the hypothesis that the object is an orange would be weakened, whereas the hypothesis that the object is a cup would be strengthened. Further discovery that the cylindrical component is about 4" long and has a handle would make the hypothesis that the object is a cup very strong. Different evidence for cup is a view of a convex patch occluding a concave patch across a jump edge of a few inches.

1.5 -- Organization of the Thesis

A computer vision system is presented in this thesis which recognizes 3D objects in range images. The analysis stage consists of three major components: segmentation, classification, and merging. The first component segments the range image into "surface patches" by a squared error

criterion clustering algorithm using surface points and associated surface normals. The second component classifies these patches as planar, convex, or concave based on a nonparametric statistical test for trend, curvature values, and eigenvalue analysis. In the third component, compatible patches are merged to produce reasonable faces of the object(s). The recognition stage takes the patch information obtained from the analysis stage to create a representation of the range image based on properties of the patches and relations between the patches. By seeking salient features within a representation, evidence is accumulated which is used to make a decision about the contents of the range image. This procedure has been applied to both real and synthetic range images, with a high proportion of successes. Figure 1-2 illustrates the proposed range image analysis procedure.

Chapter 2 presents terminology and definitions used throughout the thesis pertaining to range images. Also included are enhancement algorithms for smoothing, background elimination, and jump edge detection which improve the image quality and aid further processing. Chapter 3 discusses segmentation of range images by clustering and a good cluster algorithm for this task is empirically derived.

Chapter 4 is concerned with classifying the surface patches derived by the segmentation process. One contribution of this chapter is the design of a nonparametric statistical test for trend which is used to discover curvature in a surface patch. The advantage of this test is that the technique is quick and very reliable for larger patches, and requires no arbitrary thresholds except specification of the critical level of the test. Another contribution involves the design of a tree decision procedure which integrates the nonparametric statistical test results with curvature and eigenvalue measures to decide on surface type.

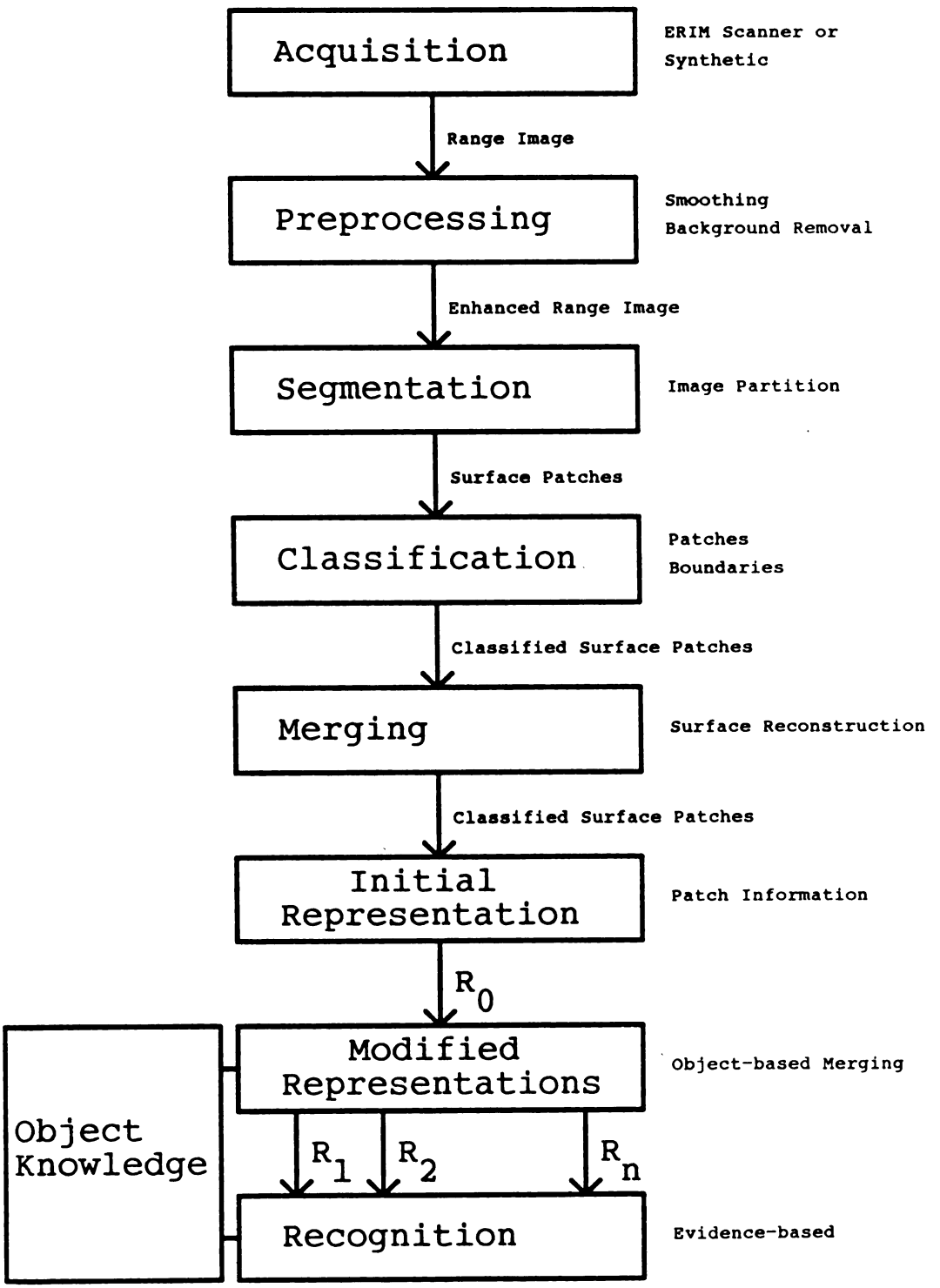


Figure 1-2
Range Image Analysis

Chapter 5 deals with classifications of boundaries (or edges) between adjacent surface patches. An important type of edge is a crease edge, formed when two surface patches intersect at a distinct "fold" or edge of the object. After deriving the type of boundary between every pair of adjacent surface patches, these patches are merged based on both the classification of the patches and by the classification of the boundary between patches. At the end of this stage a final segmentation of our range image into (hopefully) natural object faces is obtained.

The problem of identifying 3D objects from a representation of visible surface segments is treated in Chapter 6. The contribution of this chapter is an evidence-based recognition technique which identifies objects via their notable features. A representation of the range image is derived from the patch and boundary information provided by the analyses described in Chapters 3 through 5. A list of salient features of the various objects in the database forms the core of an object recognition system, which looks for instances of these features in the representation. Occurrences of these salient features are interpreted as evidence for or against the hypothesis that a given object occurs in the scene. A measure of similarity between the set of observed features and the set of salient features for a given object in the database is used to determine the identity of an object in the scene or reject the object(s) in the scene as unknown. The similarity measure is shown to have nice properties with respect to object model updates. The recognition technique avoids the inherently exponential time complexities of those matching procedures which find a mapping from every observed patch to some object patch. The recognition procedure is shown to have polynomial time complexity and to be fairly accurate in correctly identifying objects.

Chapter 7 outlines the overall scheme of the thesis, summarizing the important points from each chapter. The contributions and their advantages and disadvantages are presented, as well as suggestions for further research to extend the results of this thesis.

CHAPTER II
PRELIMINARIES

The success of an object recognition system can be compromised by excessive noise in its input. Therefore, images are usually smoothed before recognition is attempted. Recognition may also be improved by removing extraneous pixels, such as those which do not belong to any object. This chapter presents smoothing and background removal techniques to enhance real range images. A number of features of range images are defined which will be useful for recognition: edges, surface normals, surface area, and shape features. Also, we formally define range images and describe the database of range images used in this thesis.

2.1 -- Range Images

A range image is a function f(r,c) where r is the row in the image, c is the column in the image, and f(r,c) is some number which corresponds to depth at position (r,c). The vector (r,c,f(r,c)) can be converted into a real world spatial coordinate system by some procedure depending on the ranging system used. In the simplest cases this conversion will be a linear transformation. We denote a pixel as p; if the row r and column c are specified we write $p_{r,c}$. The notation $\bar{p}$ denotes the vector (r,c,f(r,c)).

The range images in our experiments are 128x128 pixels in size. Each f(r,c) is represented by an 8-bit range value; smaller range values represent more distant depth, and larger range values correspond to pixels closer to the sensor.

We use range images from two sources; real range data obtained from the laser scanner at ERIM (Environmental Research Institute of Michigan) located in Ann Arbor, Michigan, and synthetic data obtained by software we have written.

The range sensor at ERIM operates on the continuous wave technique. Figure 2-1 shows a block diagram of the components of this system. The 128x128 spatial resolution describes a footprint of approximately 6.4"x6.4", and the 256 grey values correspond to an approximate depth range of 8". Specifications of the ERIM scanner dictate that to (approximately) convert one of these range images into (x,y,z) coordinates we use x and y increments of 0.05" and z increment of 0.032". A complication arises due to the design of the sensor: the laser is emitted from a fixed location and it is incremented in angles of this beam, not x-y displacements, that correspond to the pixels. The effect is that the range sensor produces data in an elliptical coordinate system. These points may be transformed into standard Cartesian coordinates by an appropriate procedure based on the parameters of the laser sensor system. However, we find the distortion is not significant enough to degrade the performance of our procedures. Hence throughout this thesis we treat the vector (r,c,f(r,c)) as if it were the correct Cartesian 3D coordinate. In Figure 2-2 we show the reflectance images and corresponding range images obtained from ERIM for two objects: an aftershave bottle and a cup.

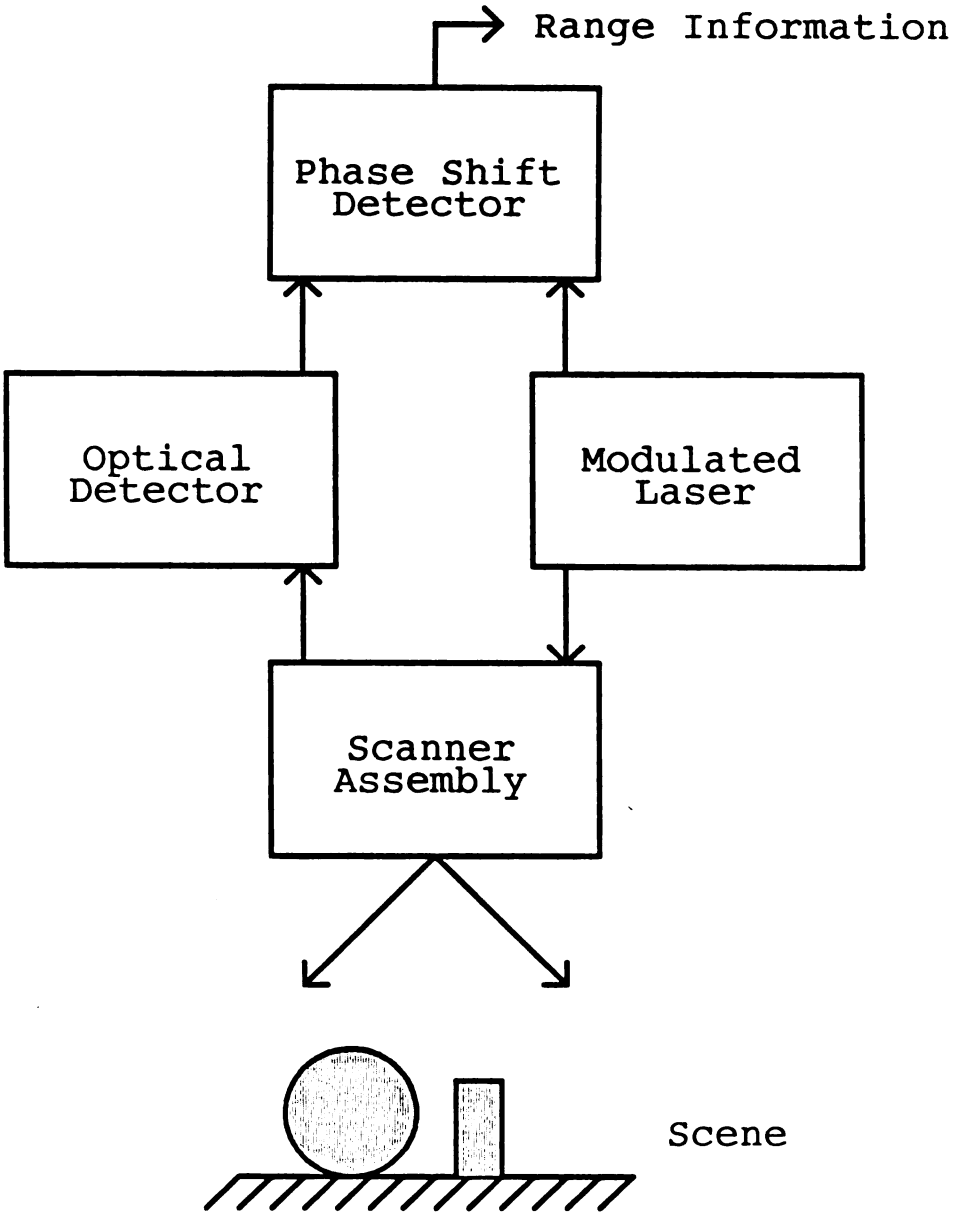


Figure 2-1
ERIM Laser Scanner

Figure 2-2
Reflectance and Range Images

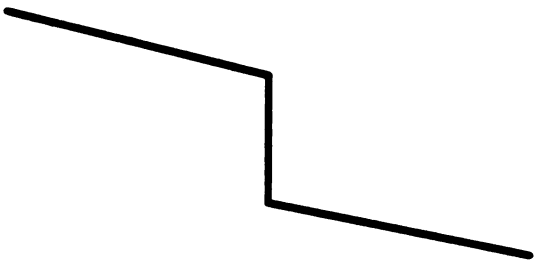
In our synthetic data, a pixel whose corresponding depth value is undefined (i.e. the imaginary sensor hits no object) is assigned a depth value of 0 and this value is ignored in further processing. Unlike the real data, we are able to generate data such that (r,c,f(r,c)) is an exact 3D spatial coordinate on an object surface. To simplify software requirements, these images are also generated with the same x, y, z increments as provided by the ERIM system. The theory and basic approach behind our synthetic range image generation is presented in Appendix A.

Both the synthetic and real range data contain noise: in synthetic images, the quantization to 256 range values introduces some degradation, and real images suffer from sensor noise as well. For known planar surfaces it is possible to measure the noise by 1) finding the best-fitting plane for the surface, and 2) deriving the mean and standard deviation of the (signed) distances from the observed range

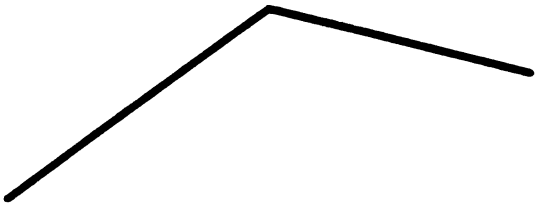
values to estimated range values derived from the planar fit. The mean should be approximately zero, and the standard deviation indicates the severity of the noise. Note that since the x, y, and z increments are fixed, a reasonable measure of the noise is this standard deviation. For convenience, let us define a constant g=0.032", the distance corresponding to one grey level in our range images. For the rest of this thesis we denote the noise level of a surface as a multiple of g.

A large amount of the literature dealing with reflectance images involves detecting edges as a means of delineating object boundaries. Edges in reflectance images are almost always formed by large jumps in grey values, and correspond to surface boundaries or changes in albedo. In range images, edges have also played a big role in the literature; however, there are at least three types of edges that are used: jump edges, crease (or roof) edges, and smooth edges.

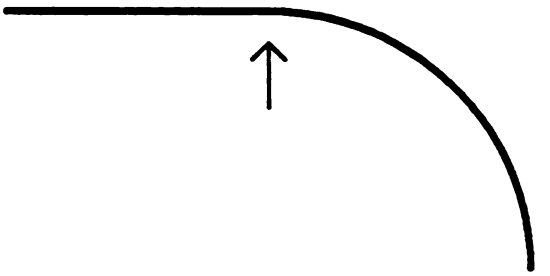
As with reflectance images, jump edges in range images are formed where depth values are discontinuous. Such edges occur when an object occludes another object or when part of an object occludes itself. Crease edges correspond to surface creases; that is, points over which surface normals are discontinuous. The third type of edge is called a smooth edge and is characterized by continuity of surface normals but discontinuity of curvature (rate of change of surface normals). Only jump and crease edges have been widely used in the literature. Ponce and Brady [Pon85] have had some success in detecting smooth edges (smooth joins in their terminology). In Figure 2-3 we observe 2D instances of jump, crease, and smooth edges.



Jump Edge



Crease Edge



Smooth Edge

Figure 2-3
Types of Edges

2.2 -- Noise Reduction

In the real range images we have obtained, the background plane (the table on which the object is placed) typically has noise level of 1.2g. However, due to the larger specularity of the objects in our database, the noise on object surfaces averages about 4g.

Before analyzing the real range images, it would be helpful to reduce the noise level of the images somewhat. Hurt and Rosenfeld [Hur84] investigate several techniques for reducing noise in two and three dimensional intensity imagery without destroying edge information: we have looked at three techniques covered in their work -- median filter, nearest neighborhood smoothing, and maximum likelihood smoothing -- as well as the donut filter discussed in [Sve85]. The median filter replaces a grey value with the median grey value of its neighbors, where the neighborhood is nxn, where n is odd. This filter reduces noise but also seems to blur the edges. The donut filter operates in two stages. An intermediate image is formed by replacing each pixel with the minimum grey value in its 3x3 neighborhood. The last stage replaces each pixel of the intermediate image with the maximum grey value in its 3x3 neighborhood. This procedure really only cleans the obvious low or high peaks in the image; noise is not really reduced.

The nearest neighborhood smoothing technique operates on all pixels in parallel:

For each pixel (r,c):

1. Identify the 5 neighbors in the 3x3 neighborhood of (r,c) having corresponding grey values f_1, f_2, f_3, f_4, f_5 closest to f(r,c).
2. Replace f(r,c) with $(f_1 + f_2 + f_3 + f_4 + f_5)/5$.

The maximum likelihood smoothing operates in much the same

way: after the 5 "nearest neighbors" are identified, the number of these neighbors falling in each 2x2 subneighborhood is counted; the grey values falling in the 2x2 window(s) with maximal count are averaged and the central pixel is replaced with this average. The performances of these two procedures, as reported in [Hur84], are good at reducing noise without reducing edge information. We have arbitrarily chosen from these two approaches the nearest neighborhood technique. In fact, we apply this filter twice. Working with planar surfaces having noise level 4g, two applications of this filter reduces the noise to less than 2g. Figure 2-4 shows the effect of applying this technique to a range image of a toy part. Notice that the grainy quality of the original image is reduced by the smoothing, and yet the important edge information has been retained.

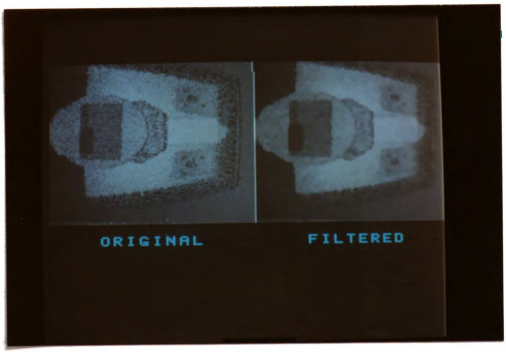


Figure 2-4
Demonstration of NN Smoothing on Toy Part

2.3 -- Background Removal

The flat table surface where the object(s) is resting also provides depth values of its own at those locations where the sensor beam hits no object. Since this background contains no useful information, it is easier to identify these background pixels and remove them from future processing at the start than to analyze the entire scene, objects and background, and then identify the background in the recognition stage. Furthermore, the number of background pixels can be much larger than the number of object pixels, hence range image processing will proceed quicker if only object pixels are analyzed.

From the laser scanner system at ERIM we also obtained a range image of the table alone. This provides a reference range image $f_{ref}(r,c)$ which, when smoothed as discussed in Section 2.2, may be used to reject pixels that fall on the background surface. However, we expect that the background surface in other range images may differ by an additive constant. Because the range scanner uses phase modulation to determine depth, the range values are calculated modulo an adjustable depth range. For example, a point that lies 12.2" from the scanner and a point that lies 20.2" from the scanner will have the same grey value (if the 256 grey values cover an 8" depth range). During the range image acquisition of objects, the sensed images had a wrap-around effect -- the grey values for the closer object points would bypass the depth level for grey value 255 and "wrap-around" to produce a grey value close to 0. An adjustment in the sensor circuitry allows the grey values to be shifted by an additive constant (modulo 256), so that no wrap-around occurs (assuming the object is not too large for the 8" depth range to begin with). The effect of such an adjustment is that the background surface grey values are also shifted by a constant. Hence, to use the reference range image we need to compute a shift factor before comparing reference and other

real data.

Our process for extracting object pixels from a real range image is as follows. In every real range image we obtained, the bottom right corner of the image shows background surface. Therefore, after smoothing the image as in Section 2.2 (to obtain smoothed range image $\bar{f}$), we perform the following:

1. Set v^* to be the average grey value over the 5x5 neighborhood of the pixel (r,c)=(123,123) in $\bar{f}$.
2. Set $s = v^* - \bar{f}_{ref}(123,123)$
3. For all (r,c):
   If $| \bar{f}(r,c) - \bar{f}_{ref}(r,c) - s | \leq 3$ then set $\bar{f}(r,c) = 0$.

Experience shows that single pixels or small patches of background pixels may not be removed by this step, particularly around the extreme border of the image. We clean these up by removing connected nonbackground components containing fewer than 40 pixels. The remaining pixels are object pixels. Denote the number of object pixels by N_p. Figure 2-5 shows the result of smoothing and background removal for cup and aftershave bottle range images. We observe an noticeable improvement in image quality as a result of these two simple operations.

2.4 -- Surface Normals

The orientations of visible surfaces of an object are useful features for its recognition because they encode the shape of the object. This orientation is represented by the unit vector normal to the tangent plane at a pixel p, and is denoted $\bar{n}_p$. We estimate this unit normal at a pixel by finding the best fitting plane (by linear least squares method) over an mxm neighborhood of the pixel. A problem

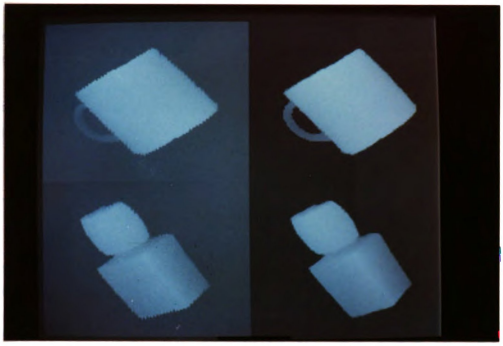


Figure 2-5
Background Removal for Cup and Bottle

with the normal vector estimation is the need to define a neighborhood. Even though we can ignore background pixels, there will be distortion of the "true" normals in the vicinity of jump and crease edges; namely, a smoothing effect, so that normals will not be wholly discontinuous in passing over an edge but will gradually change. We can expect that although large neighborhoods (e.g. 9x9, 7x7) would give reliable normals within a face, edge effects would propagate out from the boundaries a substantial distance; smaller neighborhoods would produce normals which are more susceptible to noise and quantization of range values. We have experimented with varying neighborhood sizes and have found 5x5 neighborhoods to provide good tradeoff with respect to minimizing edge effect propagation and noise (quantization) effects.

Figure 2-6 shows a sampling of surface normals derived from the aftershave bottle range image. This figure should be interpreted as a picture of a 3D scene consisting of unit length surface normal vector "needles" sticking out of the (invisible) surface of the bottle. Note the prominent edge effects on the edge of the main body of the bottle, where the two planar faces meet.

2.5 -- Jump Edge Detection

Computing surface normals in the vicinity of a jump or crease edge can give nonsensical results because pixels on either side of such an edge generally belong to different object faces. Although these erroneous normals could be treated as outliers, the detection of jump edges is not a difficult task and once known they may be used to restrict the neighborhoods used to calculate normals. Further, jump edges can be used to detect surfaces that are not spatially contiguous, which will be useful in the recognition stage. On the other hand, crease edge detection is not an easy problem.

Our detection of pixels that appear to fall on jump edges is based on the idea that if a jump edge intersects a row (column) of the range image, then there will be a discontinuity in the sequence of depth values along the row (column). To decide if a pixel $p_{r,c}$ falls on a jump edge:
Define $\Omega_4(p_{r,c})$ to be the 4-neighbors (N,S,E,W) of pixel $p_{r,c}$.

Define Δ:

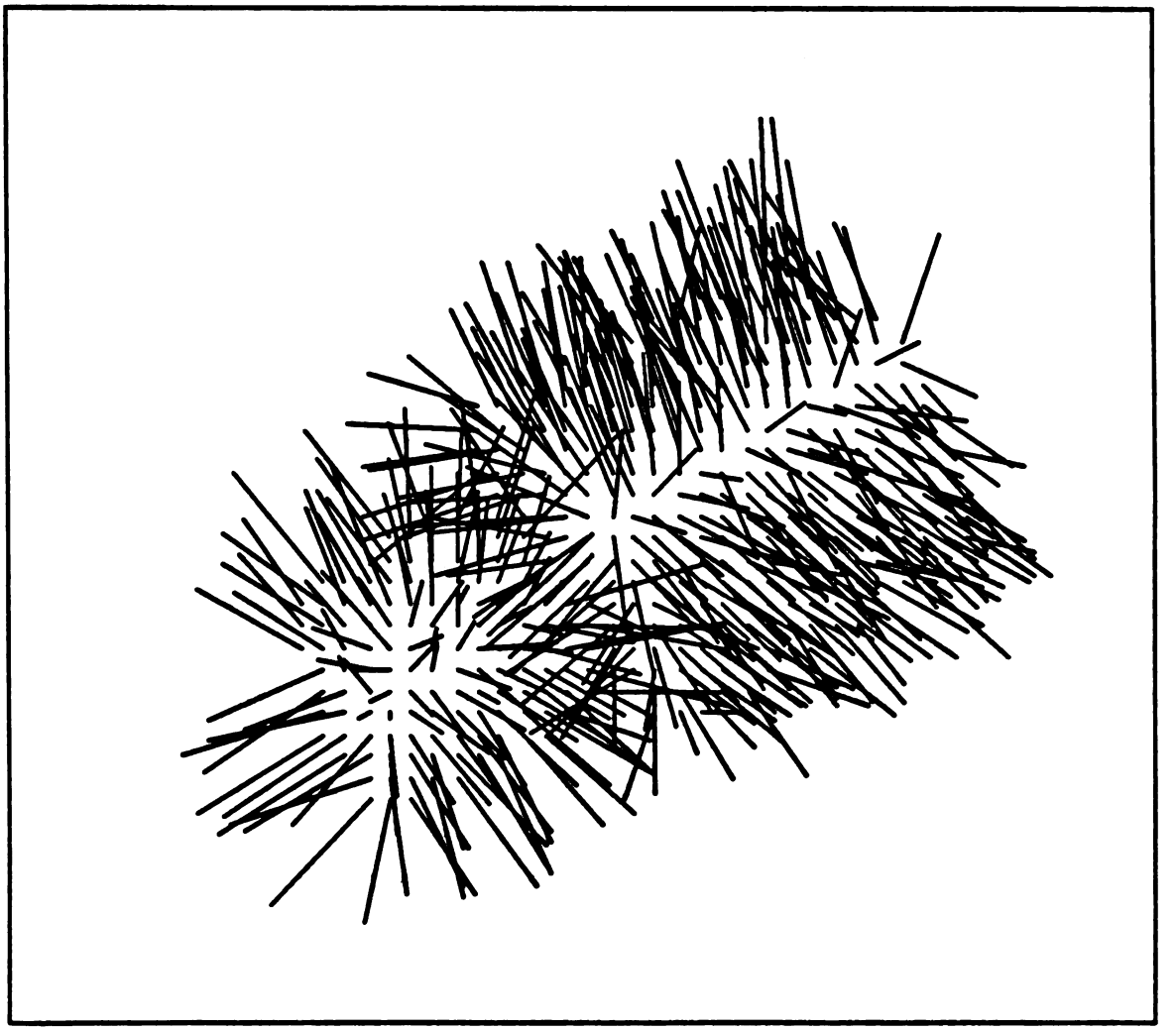


Figure 2-6
Surface Normals of a Bottle

$$\Delta = \max_{(r',c') \in \Omega_4(p_{r,c})} (|f(r,c)-f(r',c')|)$$

$$= |f(r,c)-f(r+a,c+b)| \text{ for some } (a,b) \in \Omega_4(p_{r,c}).$$

If (a,b) is not uniquely defined, pick it arbitrarily. Define a sequence $(d_i)_{i=-n}^{n}$ as follows:

$$d_i = |f(r+ia,c+ib)-f(r+(i+1)a,c+(i+1)b)|$$

(note that $d_0=\Delta$). The value n is user-specified. Conclude that a pixel $p_{r,c}$ falls on a jump edge if

$\Delta \geq w * \max(d_i: \ i=-n,\ldots,1, \ i\neq 0)$ or

$\Delta \geq w * \max(d_i: \ i=-1,\ldots,n, \ i\neq 0)$

where w is some user-specified constant. This says that the grey level difference Δ is larger than the surrounding grey level differences by at least a factor of w.

To illustrate this procedure, let w=3 and n=4, and observe the grid of range values shown in Figure 2-7. To decide if the middle pixel (range value 60) falls on a jump edge, we investigate its neighbors indicated by bold squares and find that the neighbor with range value 138 has the greatest absolute difference from 60 (Δ=78). Thus the pixels indicated by thin boxes are used to derive (d_i). Since $\max(d_i: i=-1,\ldots,4; i\neq 0)=14$ and $3*14=42$ is less than $\Delta=78$, we conclude that the middle pixel falls on a jump edge.

The algorithm for jump edge detection is:

(1) Calculate jump edge pixels for w=4 and n=4. This identifies those pixels with strong evidence of falling on a jump edge.

(2) Remove connected components of these jump edge pixels having fewer than 5 pixels. Spots of noise can produce false detections in (1).

(3) LOOP

46	47	47	47	48	48	133	138	141	143	144
46	47	47	47	47	49	136	137	140	142	142
46	46	47	47	47	50	138	139	139	140	143
46	46	46	47	47	51	139	139	139	139	143
46	46	46	46	46	53	139	139	139	139	144
46	46	46	46	46	60	138	137	140	140	143
45	46	46	46	46	115	137	137	138	140	142
46	46	46	46	47	122	136	137	137	140	141
46	46	46	46	47	126	136	136	138	139	141
46	46	46	46	48	126	136	136	138	138	139
46	46	46	46	48	127	135	137	137	138	138

Figure 2-7
Jump Edge Demonstration

Determine if pixels adjacent to previously detected jump edge pixels are classified as jump edge pixels when w=3 (and n=4). Hence the sensitivity level is increased to try to extend the current jump edge pixel set.

UNTIL no new jump edge pixels are found.

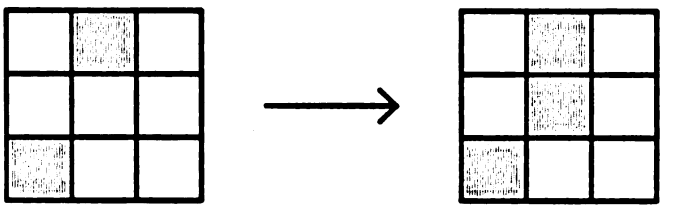
(4) Perform gap-filling:

(a) Creating bridge connections between previously unconnected jump edge pixels in a 3x3 neighborhood (e.g., see Figure 2-8(a)).

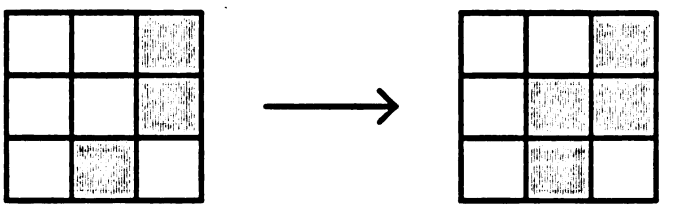
(b) Filling diagonal gaps in the edges (e.g., see Figure 2-8(b)).

The values of 4 and 3 for w were determined empirically by observing the performance of this technique for different w's. The value n=4 is a compromise between having too few d_i's to make a good decision and sampling so far away from the pixel of interest that another jump edge pixel may be encountered to adversely influence the detection scheme. Each of the above 4 steps has time complexity $O(N_p)$, where N_p is the number of object pixels. Therefore, this jump edge procedure has time complexity $O(N_p)$.

Note that any gradient operator such as the Sobel operator could be used to find jump edges. However, such a technique requires thresholding the gradient image and the resulting jump edges could be several pixels thick (depending on the neighborhood size used in the gradient operation). The technique described above will, under all but certain pathological cases, provide an edge exactly two pixels thick: one pixel on each side of the jump edge. This provides a crisp localization of jump edges which cannot be guaranteed by thinning the thresholded gradient image.



(a)



(b)

Figure 2-8
Gap-Filling Diagram

Figure 2-9 shows in part (a) the result of applying our jump edge detection technique and in part (b) the result of applying the Sobel operator to a cup image. We observe that the intensity of the Sobel values diminish towards the side of the cup along the jump edge formed by the lip of the cup. To capture the jump edge pixels toward the sides of the cup with a threshold operation on the Sobel-based image, we need a low threshold which may produce a thick boundary in other parts of the resultant thresholded image.

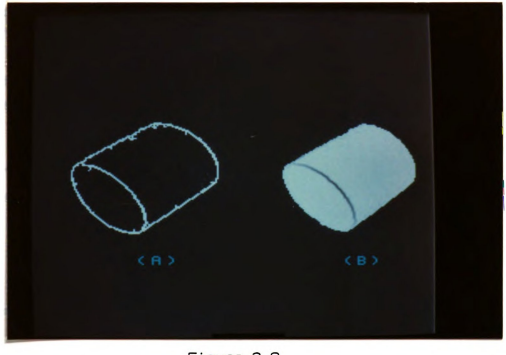


Figure 2-9
Output of Edge Detectors

Given the binary jump edge image, the surface normal vector for a pixel p is derived over the restricted 5x5 neighborhood of p defined as follows: identify the connected components of non-jump edge pixels in the 5x5 neighborhood, call these sets of pixels U_1, U_2,..., U_m. Identify those U_i which contain pixel(s) that are 8-adjacent to p. The union of these U_i's form the restricted neighborhood of p over which the normal vector at p is calculated. Figure 2-10

shows an example of this neighborhood. Jump edge pixels are marked with *'s and the restricted neighborhood of the center pixel p is the shaded portion.

2.6 -- Surface Area Estimation

Information about size, whether it be absolute size of an object component or relative size between two or more object components, is crucial for object recognition. In our surface-based approach, we use surface area to provide this "size" information. Given a region (surface patch) P in a range image obtained by segmentation, to derive its area we compute a "neighborhood area value" for each pixel (r,c) in P; their sum gives us an approximation of the surface area of P.

To derive the neighborhood area value for pixel (r,c), define the diagonal neighbor set $\Omega_d(p_{r,c})$ as (see Figure 2-11)

$$\Omega_d(p_{r,c}) = \{(r+i, c+j): \; i=-1,1; \; j=-1,1\}.$$

We derive four 3D points $\bar{\lambda}(a,b)$, $(a,b) \in \Omega_d$, where

$$\bar{\lambda}(a,b) = (\bar{p}_{r,c} + \bar{p}_{a,b}) / 2,$$

with the exception that if (a,b) is not in P then the z component of $\bar{\lambda}(a,b)$ is f(r,c). The areas of the two 3D triangles defined by the triples of points $(\bar{\lambda}(r-1,c-1), \bar{\lambda}(r-1,c+1), \bar{\lambda}(r+1,c-1))$ and $(\bar{\lambda}(r-1,c+1), \bar{\lambda}(r+1,c-1), \bar{\lambda}(r+1,c+1))$ are added to obtain the contribution of pixel (r,c) to the surface area of P; that is, the neighborhood area value of (r,c).

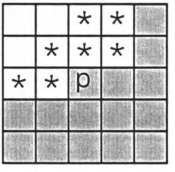


Figure 2-10
Restricted Neighborhood Example

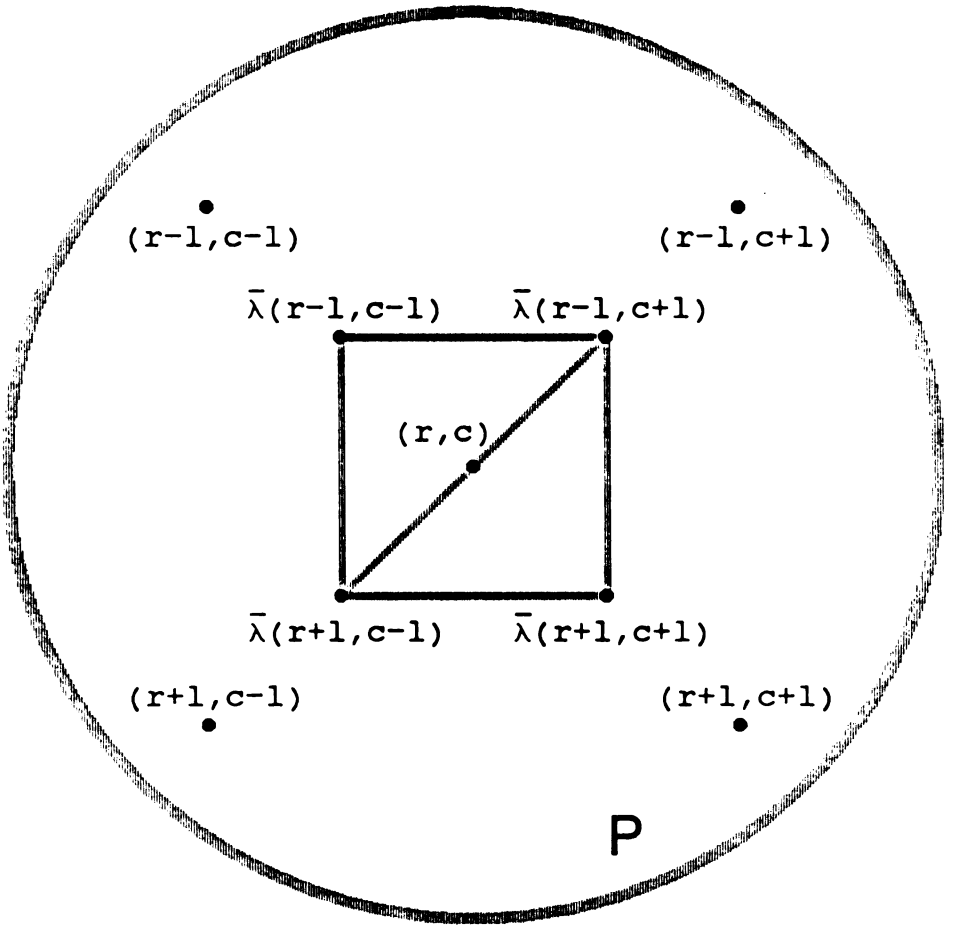


Figure 2-11
Neighborhood for Surface
Area Estimation

Unfortunately, noise even of level 1g has a "crinkling" effect which causes a substantial overestimation of the surface area. Table 2-1 shows the effect of noise on the estimation of surface area of a plane with true area 4.76 square inches and a spherical patch with true area 4.90. Ten noisy surfaces were generated for each noise level, and the table reports the average estimated area. Note that the values under noisy conditions are considerably greater than the true area. The overestimation of area for the spherical patch under noise level 0g occurs due to quantization

Table 2-1
Effect of noise and smoothing on
estimated surface area.

	Patch type	
	Planar	Spherical
Noise level		
0g	4.76	5.32
1g	5.63	6.02
2g	7.41	7.66
2g & smoothed	4.76	4.98

effects. Therefore, it is important to use smoothing to eliminate noise. This may have an adverse effect on curved surfaces, however. For example, the degree of smoothing needed to reduce the noise level on a spherical surface may also slightly flatten the surface, causing the area to be underestimated. However, it is not as serious to underestimate as it is to overestimate the surface area; occlusion in a scene will also cause an underestimation of surface area, so a recognition system will have to be able to handle underestimation of surface area anyway. Table 2-1 also shows the estimated areas after smoothing for the planar and spherical patches with 2g additive noise; these are very close to the true values, demonstrating that smoothing can help recover the correct surface area. Here we have applied

the Gaussian smoothing technique, which we will now define.

The Gaussian smoothing technique is based on the "computational molecules" of Terzopoulos [Ter83] which were utilized by Brady et al. [Bra85] for range image smoothing. This smoothing technique proceeds by performing a weighted averaging based on the 3x3 Gaussian filter mask shown in Figure 2-12(a), which is broken up into the four "computational molecules" shown in Figure 2-12(b). These computational molecules are used for the purpose of handling edge effects. Application of this filter to an image n times corresponds to smoothing with a Gaussian filter whose standard deviation is $\sqrt{n}$ [Bur83].

Smoothing a patch P by this technique is basically a convolution of the patch with the 3x3 mask shown in 2-12(a), with adjustments in the mask for those instances where the mask would fall partially outside of P. Suppose we wish to smooth the image in a neighborhood of a pixel $\bar{p}_{r,c}$ in patch P. We define four values δ_1, δ_2, δ_3, and δ_4 where:

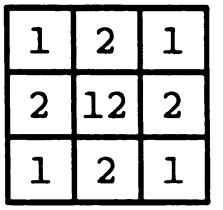
$$\delta_1 = \begin{cases} 1 \text{ if } (r-1,c-1) \text{ and } (r+1,c+1) \text{ are in P,} \\ 0 \text{ otherwise;} \end{cases}$$

$$\delta_2 = \begin{cases} 1 \text{ if } (r-1,c) \text{ and } (r+1,c) \text{ are in P,} \\ 0 \text{ otherwise;} \end{cases}$$

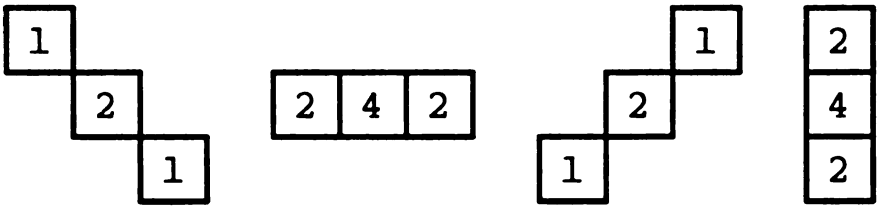
$$\delta_3 = \begin{cases} 1 \text{ if } (r-1,c+1) \text{ and } (r+1,c-1) \text{ are in P,} \\ 0 \text{ otherwise;} \end{cases}$$

$$\delta_4 = \begin{cases} 1 \text{ if } (r,c-1) \text{ and } (r,c+1) \text{ are in P,} \\ 0 \text{ otherwise;} \end{cases}$$

We define the smoothed pixel $\bar{p}^{sm}_{r,c}$ at (r,c) to be:



(a)



(b)

Figure 2-12
Computational Molecules

$$\bar{p}^{sm}_{r,c} = (\delta_1[\bar{p}_{r-1,c-1} + 2\bar{p}_{r,c} + \bar{p}_{r+1,c+1}] + \delta_2[2\bar{p}_{r-1,c} + 4\bar{p}_{r,c} + 2\bar{p}_{r+1,c}] + \delta_3[\bar{p}_{r-1,c+1} + 2\bar{p}_{r,c} + \bar{p}_{r+1,c-1}] + \delta_4[2\bar{p}_{r,c-1} + 4\bar{p}_{r,c} + 2\bar{p}_{r,c+1}]) / (4\delta_1+8\delta_2+8\delta_3+4\delta_4)$$

if $(\delta_1+\delta_2+\delta_3+\delta_4) \neq 0$, and

$\bar{p}^{sm}_{r,c} = \bar{p}_{r,c}$ otherwise.

One application of this filter does not sufficiently smooth a patch. For the kinds of noise level we have, we apply this filter 20 times in order to be reasonably certain that the resulting surface is smooth. Figure 2-13 shows a spherical surface with 2g additive noise; Figure 2-14 shows the same surface after Gaussian smoothing is applied 20 times. The elimination of "crinkling" on the surface is obvious, although smooth pits and hills remain, possibly an effect of the initial grey level quantization.

2.7 -- Morphological Features

When the distinction between object and background has been made, we can construct a binary image, with 0's where background pixels occur and 1's where object pixels occur. This binary image provides a silhouette image of the object(s) present in the range image. We may think of a silhouette image as a projection of points of a 3D object onto a 2D image plane. The identification of objects from silhouettes is often an easy task for humans, particularly for objects which have a distinctive shape. Hence we derive various morphological features from silhouette images to help identify objects in a range image. We define three morphological features:

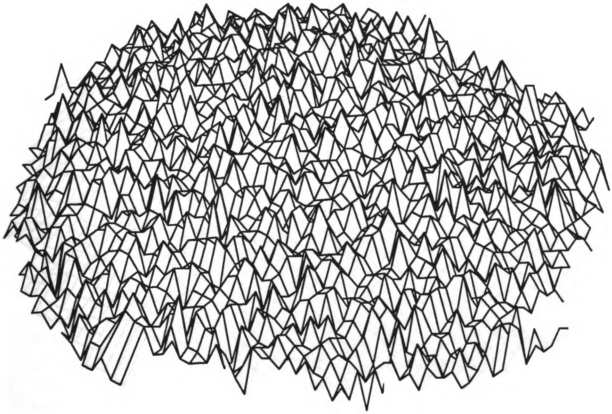


Figure 2-13
Contour Plot of Noisy Surface

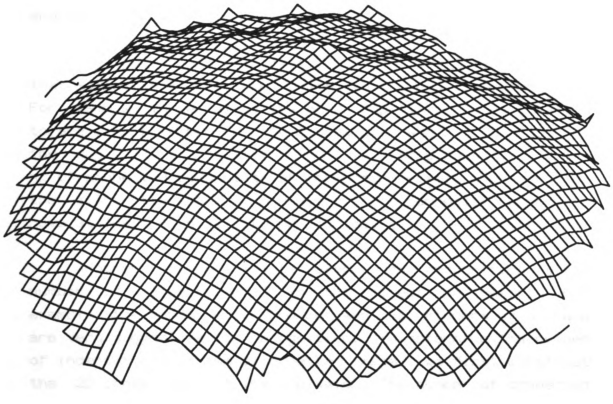


Figure 2-14
Contour Plot of Smoothed Surface

(1) Perimeter (two dimensional);

(2) Background component count; and

(3) Convex hull based background component count.

The perimeter is derived by identifying those "1" pixels which lie on the outside border of the object and tracing around the object along these pixels.

The background component count is useful in the sense that some objects may have "holes" which help in recognition. For example, in certain views of a cup it would be possible to see a hole formed by the cup handle and the main body of the cup. To obtain this feature we simply count the number n_b of connected components of background pixels that occur in the silhouette image. We require that connected components have at least 40 pixels to be counted.

Another feature which is useful for recognition is the number of "indentations" in the object perimeter. For example, the space occurring between the fingers of a hand are very suggestive for recognition. To determine the number of indentations in the silhouette image, we first construct the 2D convex hull of its 1-pixels. The number of connected background components n_c which occur within this convex hull (again having 40 or more pixels) reflects the number of silhouette indentations. Specifically, the number of connected silhouette indentations is $n_c - n_b + 1$.

Figure 2-15 shows a stage in computing the number of convex hull background components for the range image of a hand. The convex hull of the hand pixels is colored white, the connected background components occurring within this convex hull are colored blue, and the object (hand) pixels are colored red. The number of convex hull based background components is six for this image (not seven, since one component has less than 40 pixels).

Figure 2-15
Silhouette Indentations

Component counts must be used with some caution in object recognition, since these features are very general properties. In particular, they are scale invariant. These issues are considered in more detail in Section 6.3. The time complexity of deriving these three features is $O(N_p \log N_p)$.

2.8 -- Range Image Database

The principal set of images to which we will apply the techniques developed in this thesis are a set of 31 range images, 19 of which were obtained from ERIM and 12 of which were generated with our software. These images represent views of 10 objects (6 real and 4 synthetic). Table 2-2 shows a summary of these objects and views: a description of the objects (a name), an abbreviation for each object used throughout the thesis, the type of object (real or

synthetic), and the number of views taken. The names of the synthetic objects suggest their shape. Synthetic range images were generated with added Gaussian noise with noise level 1.5g. We refer to the set of images for a given object

Table 2-2

Object and Range Image Database

Name	Abbreviation	Type	Number of views
Aftershave Bottle	AS	Real	4
Cup	HC	Real	4
Block	GB	Real	3
Tunnel	TN	Synthetic	3
Cobra Sculpture	CB	Real	3
Mushroom	MH	Synthetic	3
Plug	PL	Synthetic	3
Diesel	DS	Synthetic	3
Toy Part	TY	Real	2
Human Hand	HN	Real	3

by the abbreviation and a number: for example, the four images of the aftershave bottle are denoted AS1, AS2, AS3, and AS4. The preprocessed views of these objects are shown in Appendix B. Part (a) of Figures B-1 through B-31 show the range images for the ten objects.

2.10 -- Summary

This chapter establishes a foundation on which to develop procedures for range image analysis and object recognition. Noise reduction (smoothing) and background pixel removal are important stages in the investigation of real range images and provide more reliable and quicker results. When dealing with real range images this thesis will assume that smoothing and background pixel removal have been applied: the depth function f(r,c) will refer to the

smoothed grey values. Synthetic images are not smoothed and do not require background removal. Jump edge detection is useful for delineating image regions which do not correspond to contiguous surfaces in 3D, and will be useful both in the analysis and recognition phases.

We have also defined the concepts of surface normal and surface area estimation and corresponding numerical techniques for their computation on discrete images. Some morphological features useful for object recognition have also been defined.

CHAPTER III
RANGE IMAGE SEGMENTATION

A segmentation of a scene into regions which ideally correspond to natural object faces is essential for surface-based object recognition. Properties of these regions are used to construct a representation of the scene which can be used for recognition. Image segmentation is well suited to range images, in which pixels fall into natural groups corresponding to faces of objects in the range image. This chapter proposes clustering as a means of segmenting range images. Section 3.1 briefly discusses image segmentation techniques in the literature. Section 3.2 considers issues for the specific problem of range image segmentation by clustering, which is followed in Section 3.3 by an evaluation of a number of clustering techniques on range images. Section 3.4 discusses the postprocessing stage necessary to clean segmentations, and evaluates our findings.

3.1 -- Image Segmentation Techniques

Image segmentation techniques fall into two classes: edge detection approaches, and region growing or clustering approaches. Edge detection techniques take as image segments those regions bounded by closed boundary contours, which are derived by any of a number of edge operators. This approach depends on flawless edge detection (e.g. missing portions of edges must be filled in). As seen in Section 2.5, jump edges

are not difficult to detect in range images; however, most natural object faces are bounded not by jump edges but by crease edges, and there are no reliable crease edge detectors available. Thus we rule out the possibility of creating a range image segmentation system solely based on edge detection. However, we do make use of jump edge information in Section 3.4 to help clean the segmentation derived by the CLUSTER technique discussed in Section 3.3.

The clustering approach involves aggregating feature vectors (corresponding to pixels) that are similar and separating those feature vectors that are dissimilar. Clustering differs from region extraction in that the latter uses the adjacency information available in an image to produce connected segments, whereas clustering techniques treat feature vectors simply as patterns in a high-dimensional space. Feature vectors may include the (x,y) coordinates so that clusters will tend to be connected, but are not guaranteed to be connected. Consequently, image segmentation by region extraction guarantees that each segment is connected in the image; image segmentation by clustering does not. Clustering has been extensively used in the literature for segmenting reflectance images but not for range images. Segmentation of reflectance images by clustering has been performed by region splitting/merging ([Nar77], [Fuk80]), by thresholding features of the image ([Sch79], [Har75]), and by grouping patterns in a multidimensional feature space ([Col79], [Gol78]).

3.2 -- Clustering for Range Image Segmentation

The basic purpose of segmenting range images is to identify connected regions of the range image such that 1) each region is contained within a larger region of the range image which corresponds to a natural object face, and 2) the regions are not too small to make merging or classifying them

impossible or unreliable. The first condition asserts one implicit assumption about the range images we expect to see: that the complexity of the surface primitives is limited to quadrics, so that a face in a range image can be approximated by a function

$$0 = a_1 r^2 + a_2 c^2 + a_3 f^2(r,c) + a_4 rc + a_5 r f(r,c) + a_6 c f(r,c) + a_7 r + a_8 c + a_9 f(r,c) + a_{10}.$$

The second condition implies that a segmentation into singleton pixel patches trivially satisfies condition (1) but is clearly of no use: the larger the segments that do not violate condition (1), the better the detection of natural object faces.

An important issue in any clustering application is that of deciding what features should be used. For each pixel of a range image there are many candidates; e.g., $\bar{p}$, $\bar{n}_p$ values, the coefficients for the best fitting quadric surface, and curvature measures. In our clustering experiments, we used the following features:

(1) The spatial coordinates $\bar{p}=(r,c,f(r,c))$, and

(2) The estimated unit surface normal vector $\bar{n}_p$.

To avoid preference of any features over other features due to unequal scaling, we normalize each feature to have mean zero and unit variance.

These features are in a sense the minimum feature set we could justify; they all play distinct and important roles. The r and c features emphasize coherence between neighboring pixels. That is, two pixels that are adjacent in the image are more likely to belong to the same object face than two pixels which are widely separated in the image. The depth value $f(r,c)$ is important in the detection of jump edges, and the normal vector is needed to detect crease edges since $\bar{p}$ does not experience unusual change over such a boundary.

The idea of obtaining 10 features for each pixel by finding the best fitting quadric surface function over a neighborhood of the pixel is very appealing [Hal82]. The clusters in this 10-dimensional space should ideally form a perfect segmentation, given that object faces are quadric surfaces. Also, the center of each cluster could supply a function for each face of the range image, from which could be derived the type of surface (plane, sphere, cylinder, etc.), its appropriate parameters (e.g. radius of curvature), and its location (or translation) in space. We have attempted this approach and found that the coefficients a_i cannot be reliably estimated, most likely a side effect of noise and quantization of depth values, and too many parameters. Consider, for example, what happens when we generate a range image of a hemisphere containing 5013 pixels, representing a sphere with radius 2 and center at (0,0,2): that is, satisfying the equation $x^2/3 + y^2/3 + z^2/3 - 4z/3 = 0$. For this surface $a_4 = -4/3$, $a_8 = a_9 = a_{10} = 1/3$, and other a_i coefficients are 0. Consider the five surface patches of this range image shown in Figure 3-1. Since the basic system of equations is homogeneous, we add the constraint that $a_8+a_9+a_{10}=1$. If we perform a least squares fit to obtain a_1 through a_{10} for each patch, we obtain 5 sets of (a_i). If the values we supply to the least squares routine are exact (i.e. not quantized to 256 grey levels), then we obtain Table 3-1, where each row corresponds to one set of (a_i) coefficients. We observe that the coefficients so obtained are essentially correct. However, if we supply the least squares routine with values which are quantized to 256 grey levels, we obtain Table 3-2, which shows estimated coefficients which are often much different from the theoretical coefficients. Thus, unless the range data is practically noise-free, the use of quadric fitting appears to be inappropriate.

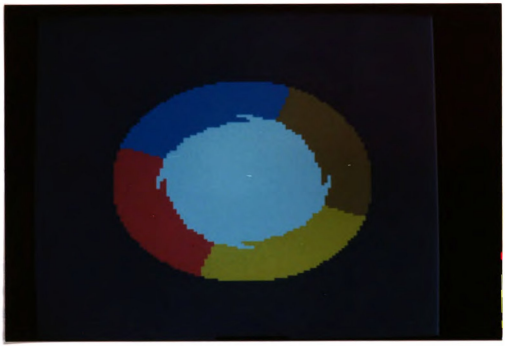


Figure 3-1
Surface Patches of Hemisphere

Table 3-1
Quadric Fit Coefficients for Exact Sphere Data

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
0.03	0.00	0.00	-1.35	0.00	0.00	0.00	0.33	0.33	0.33
-0.00	-0.00	0.00	-1.33	0.00	0.00	0.00	0.33	0.33	0.33
0.00	0.00	0.00	-1.33	0.00	0.00	0.00	0.33	0.33	0.33
-0.00	0.00	-0.00	-1.33	0.00	0.00	0.00	0.33	0.33	0.33
0.00	0.00	0.00	-1.33	0.00	0.00	0.00	0.33	0.33	0.33

If we repeat this process and fit a surface-parameterization function

$$z = f(r,c) = a_1 + a_2r + a_3c + a_4rc + a_5r^2 + a_6c^2$$

to obtain patterns in 6-dimensional space, the results obtained are still unusable. Even though this surface parametrization form is commonly used (e.g. [Bes84]), it has

Table 3-2
Quadric Fit Coefficients for Quantized Sphere Data

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
4.92	0.00	0.00	-3.49	0.00	0.00	0.00	0.22	0.22	0.57
0.67	0.43	0.19	-1.42	0.03	-0.04	-0.02	0.37	0.31	0.32
0.67	0.19	-0.42	-1.42	-0.03	-0.02	0.04	0.31	0.37	0.32
0.67	-0.19	0.42	-1.42	-0.03	0.02	-0.04	0.31	0.37	0.32
0.67	-0.42	-0.19	-1.42	0.03	0.04	0.02	0.37	0.31	0.32

the inherent disadvantage that even simple surfaces such as spheres and cylinders do not have a consistent parametrization over the entire surface. For example, if we take the above sphere function and patch information and derive coefficients a_1-a_6 using exact data, we obtain Table

Table 3-3
Six-coefficient Fit for Exact Sphere Data

a_1	a_2	a_3	a_4	a_5	a_6
15.766	-0.027	-0.026	-0.000	-2.271	-2.271
-15.071	-25.996	-24.147	-8.361	-8.550	-8.125
-9.433	-20.790	21.035	7.204	-7.430	-7.287
-9.489	19.216	-22.553	6.980	-6.988	-7.963
-4.399	17.877	16.159	-5.809	-6.787	-6.330

3-3. Each row of coefficients may be a plausible parametrization of one of the five patches, but none will satisfy the entire surface with good accuracy. Therefore, we conclude that fitting quadric surfaces to real data is not useful for characterizing surfaces.

A classical unsolved problem of clustering is the issue of determining how many clusters are present in a given set of patterns [Dub80]. This problem is critical to our application: choosing too few clusters would result in a segment containing portions of two distinct faces, and too many clusters could compromise the efficiency of further analysis. Many clustering techniques will give the user as many clusters as requested. It is possible to define measures which in some sense evaluate the merit of a given clustering. For example, compactness and isolation measures are popular [Bai82]. A priori knowledge can play a role in segmenting range images if the complexity of the objects one expects to encounter is known, such as the number of faces on the most complex object. A reasonable upper bound B on the number of segments may be used. One may then either choose a clustering of exactly B segments and concentrate on developing a technique that reliably merges segments belonging to the same face, or use some criterion to determine the best number of clusters from 1 to B; a combination of utilizing a measure and a merging routine may be best.

3.3 -- Clustering Techniques

Different clustering techniques produce clusters with different characteristics. For example, the single link cluster algorithm finds "straggly" clusters and the complete link cluster algorithm finds ellipsoidal clusters. Some cluster algorithms are designed to avoid such characteristic tendencies, for example the mutual nearest neighborhood algorithm of Gowda and Krishna [Gow78]. What characteristic of clusters, if any, we expect to find to be best for range image segmentation is not immediately obvious. Some idea about the shape of clusters can be obtained by projecting our 6D data (formed by spatial coordinates and unit surface normal vectors) down to 2D. Figure 3-2 shows a projection of

subsampled 6D data derived from the AS3 range image (Figure B-3(a)) using the principal component method. The percent variance retained is 87%. The points have been labeled such that labels 1 and 4 correspond to the two flat faces of the bottle, label 3 corresponds to those faces whose surface normals are parallel to the axis of symmetry of the bottle, and label 2 corresponds to the bottle cap. We observe that points with labels 1, 2, and 4 tend to be aggregated in the plane. We have applied a number of clustering techniques to range images to evaluate the effectiveness of the techniques in finding reasonable surface patches. These techniques are described in the following subsections.

3.3.1 -- Single Link Clustering

The a priori expectation that segments should be spatially connected prompted us to try the single link clustering technique, where a minimal spanning tree (MST) is constructed such that nodes of the graph correspond to pixels in the image and edges of the graph connect 8-neighbors of the range image. Edge weight is taken to be the Euclidean distance in 4D space between corresponding 4-tuples formed by (z,n_x,n_y,n_z), with the multiplicative factor of $\sqrt{2}$ for diagonal neighbors. This technique generates clusters by progressively cutting the longer edges in the MST.

One problem of the single link technique is that it tends to create many small single pixel "outlier" clusters. To remedy this situation and derive clusters of significant size, we merge all derived clusters with less than a certain number of pixels (in our case 40) into a single "scratch cluster". However, even this measure is insufficient to provide a good segmentation. To illustrate, Figure 3-4 shows the application of the single link technique to the synthetic box range image shown in Figure 3-3, deriving 10 clusters. Note that most of the surface area belongs to a single large

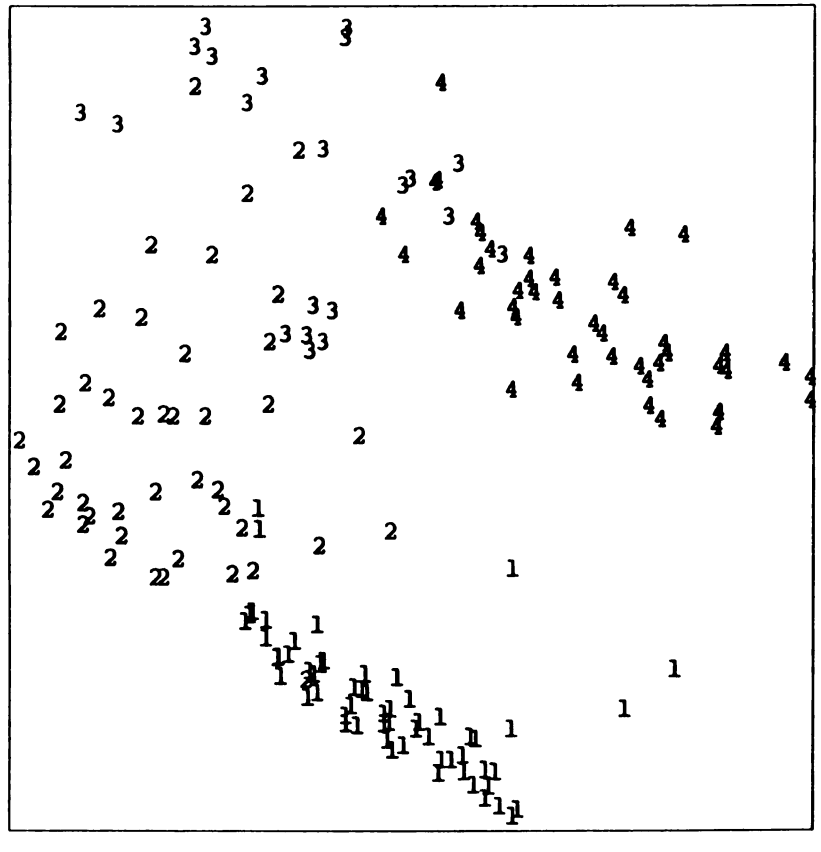


Figure 3-2
Projection of 6D Image Data to 2D

Figure 3-3
Box Range Image

Figure 3-4
Single Link Clustering of Box

cluster (colored red). The inferior segmentation may follow from the choice of features, rather than an insufficiency in the single link technique. For example, perhaps the z feature should be ignored.

A related clustering technique which is also based on the MST was proposed by Zahn [Zah71]. In this technique, a value of inconsistency is derived for each edge of the MST, which reflects how unusual the given edge length is compared to the "neighboring edges" in the MST. The cluster technique proceeds by progressively cutting edges with larger inconsistency values. Applying this cluster technique to the box range image gives us the segmentation shown in Figure 3-5. This result is very poor; the boundaries of these segments tend to consist of pixels of equal depth values.

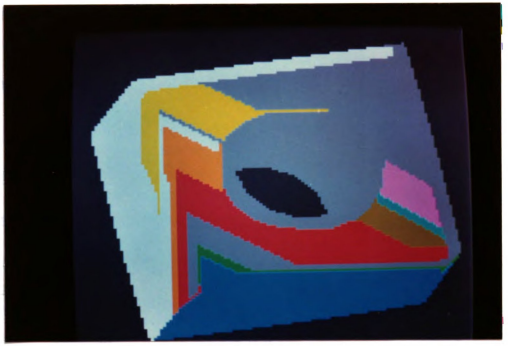


Figure 3-5
Inconsistent Edge Clustering of Box

3.3.2 -- Mutual Nearest Neighbor Clustering

The mutual nearest neighbor clustering algorithm [Gow78] uses a dissimilarity measure called the mutual neighbor value (mnv) along with the Euclidean distance dissimilarity function to sequentially merge clusters (initially individual patterns) until a specified number c of clusters is obtained. The measure mnv between two patterns (pixels) a and b is defined as

$$mnv(a,b) = NF(a,b) + NF(b,a),$$

where NF(a,b) is defined to be that integer satisfying the statement "a is the NF(a,b)'th nearest neighbor of b". The user specifies a threshold k for mnv. Clustering proceeds by ordering all pairs of patterns with mnv=2, and merging the corresponding clusters through this sequence of pairs. The same procedure is performed for mnv=3,4,...,k, and terminates when either exactly c clusters have been obtained, or all the above operations have been carried out.

When applied to the box range image (Figure 3-3), with k=20 we obtained a clustering consisting of one extremely large (blue) cluster containing most pixels of every face, and five small clusters that correspond mainly to edge pixels. This is seen in Figure 3-6.

3.3.3 -- Centralized Cluster Techniques

Another type of clustering scheme tends to produce globular clusters. Two such algorithms are the complete link (COMLNK) algorithm and the CLUSTER algorithm [Dub76]. The CLUSTER algorithm is a heuristic cluster scheme which attempts to minimize a square error criterion. The user specifies an upper bound on the number c of clusters desired, and CLUSTER finds a sequence of clusterings of 2 through c

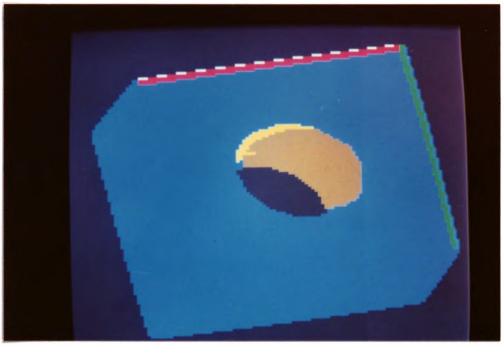


Figure 3-6
Mutual Nearest Neighbor Clustering of Box

clusters. Unlike COMLNK, this sequence is not hierarchically structured.

The memory and time requirements of these algorithms demand that these algorithms be applied to subsampled images. In our studies the range images are 128x128 so that up to 16384 pixels would have to be clustered (but usually fewer, since background pixels are removed) By sampling every f_sth row and f_sth column, the number of patterns to be clustered can be reduced to program limits. Since both algorithms generate clusters that are reasonably represented as hyperellipsoids about a cluster center, it is possible to assign the remaining pixels to clusters having the closest cluster center.

Figure 3-7 shows the 10-cluster segmentation of the box range image using CLUSTER, and Figure 3-8 shows its 10-cluster segmentation using COMLNK. The quality of these two results are surprisingly similar. But although their

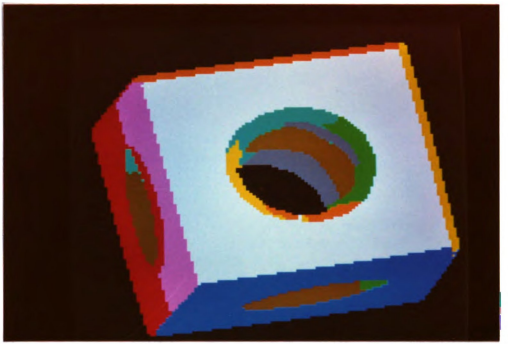


Figure 3-7
Clustering of Box by CLUSTER

outputs are essentially equal, we note that CLUSTER is considerably faster to execute and has lower memory requirements so that finer subsamplings of the image may be clustered. We have applied the various segmentation schemes discussed to several other range images, with similar results. Hence our range image segmentation technique of choice is CLUSTER; all further discussion in this thesis deals with CLUSTER. A brief outline of the CLUSTER algorithm is given in Appendix C.

The segmentation shown in Figure 3-9 was made by subsampling with frequency f_s=4. To verify the clustering, we performed two further clusterings: one utilized more sparsely subsampled image points (f_s=6), shown in Figure 3-9, and the other chose 900 points picked at random from the

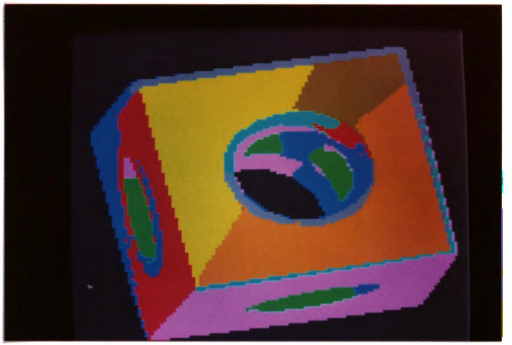


Figure 3-8
Complete Link Clustering of Box

image, and is shown in Figure 3-10. Comparing the clusterings formed from f_s=4, f_s=6, and the random sampling, we observe that the clusters obtained are very consistent in the sense that a cluster for one data set is usually also a cluster for the other data sets. This supports our belief that the clusters generated by subsampling are not radically different from those which would be obtained by applying CLUSTER to the entire image.

3.3.4 -- Determination of Number of Segments

Now we address the issue of determining the appropriate number of clusters. CLUSTER supplies for each cluster m the average within-cluster interpoint distance, CLAVGD(m), defined as:

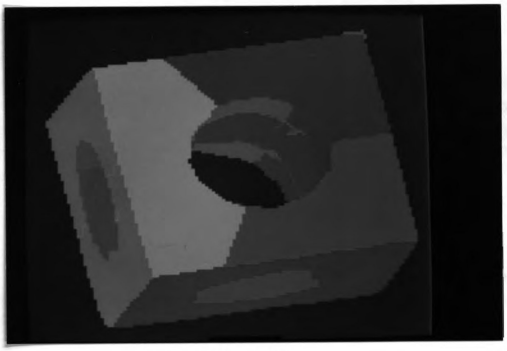


Figure 3-9
Clustering from Sparser Subsampling of Box

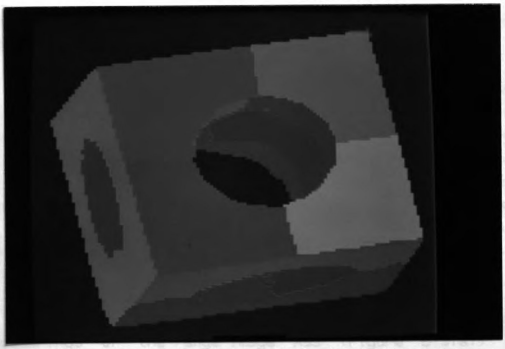


Figure 3-10
Clustering from Random Subsampling of Box

$$CLAVGD(m) = 1/|G_m| \sum_{x \in G_m} d(x,c(m))$$

where $c(m)$ is the center of cluster m, G_m is the set of points belonging to cluster m, d is the Euclidean distance measure, and $|G_m|$ denotes the cardinality of G_m. Coggins [Cog82] defined a statistic $S(m)$ which reflects the isolation and compactness of cluster m, defined as:

$$S(m) = [\min_{\ell : \ell \neq m} d^2(c(m),c(\ell))] \; / \; CLAVGD(m).$$

He derived an overall merit function S_{ave} which is a weighted average of $S(m)$'s, where the weights are the cardinalities of clusters:

$$S_{ave} = (\sum_m |G_m|)^{-1} (\sum_m |G_m| S(m))$$

Those clusterings with larger values of S_{ave} are preferred over those with smaller values.

In practice we observe that S_{ave} often hits an early global maximum (e.g. a 3-cluster solution). Thus the simple technique of accepting that clustering giving the largest value of S_{ave} is unsuitable. Instead, we set an upper bound on the number of segments expected (12 in our experiments) and identified the clustering which gave the largest number of clusters out of the set of clusterings whose S_{ave} values were local maxima. This is the criterion by which we choose a segmentation of a range image using CLUSTER.

For an example, Table 3-4 shows the S_{ave} values for clusterings of the range image AS3 (Figure B-3(a)) with numbers of segments 2 through 12. The local maxima occur at clusterings 4, 8, and 10. The above procedure selects the clustering with 10 segments.

Table 3-4

Compactness Criterion S_{ave} Vs. Number of Clusters

Number of clusters	S_{ave}
2	40.837
3	52.008
4	55.675
5	53.377
6	41.434
7	30.348
8	33.516
9	30.050
10	31.514
11	30.891
12	27.086

3.4 -- Postprocessing

The segmentation from CLUSTER is not guaranteed to provide connected clusters. Furthermore, there is a weakness in this approach which, fortunately, can usually be countered by using jump edge information. This weakness consists of the fact that, given two object faces which are essentially parallel to each other but are separated by a jump edge, only the z-components of pixels passing over the jump edge from one patch to the other will exhibit any change: the fact that the other five spatial coordinates remain fairly constant gives CLUSTER strong reason to ignore the jump edge. Hence, the first step in our postprocessing is to "zero out" pixels in the segmentation label image which lie on a jump edge. By doing this, we separate the incorrectly merged patches along the jump edge -- that is, disconnect that particular patch.

Next we deal with clusters that are disconnected; here we simply detect all connected components of all clusters, and make these components clusters in their own right. In a typical real image, the number of resulting connected components is often as large as 100; this is caused by many instances of clusters containing 1 to 4 pixels. Since these "noise clusters" are essentially useless, we merge them with larger neighboring clusters. Hence, given a cluster C_1 containing fewer than some prespecified number of pixels (40 in our experiments), we merge C_1 with that cluster C_2 having the largest number of pixels adjacent to pixels of cluster C_1.

Once these postprocessing steps of removing jump pixels and detecting and merging connected components is made, we hopefully have a reasonable segmentation for further analysis. Note that the number of segments finally obtained can be different from that specified by the S_{ave} statistic.

Our range image segmentation procedure can be summarized as follows:

(1) Obtain subsampled 6D feature vectors from range image; normalize features to mean zero and unit variance.
(2) Apply CLUSTER to get 2 to 12 cluster segmentations.
(3) Use S_{ave} to determine the number of clusters.
(4) Use jump edge information to find clusters broken by depth discontinuities.
(5) Identify connected cluster components.
(6) Merge smaller components.

The results of applying this segmentation and postprocessing operations to our database of 31 range images are shown in part (b) of Figures B-1 through B-31. We note that segments do not generally cross over natural object faces; however, large object faces or object faces with high surface curvature are usually oversegmented. Of 149 natural object faces over the 10 objects, 69 of these, or 46%, are correctly detected as single surface patches. Merging techniques for recovering from this oversegmentation are discussed in Chapters 5 and 6; the technique presented in Section 5.4 is solely data-driven, whereas the technique developed in Section 6.3 involves some knowledge about the objects to be recognized.

3.5 -- Summary

This chapter has described cluster techniques for segmenting a range image into reasonable surface patches. We use the CLUSTER algorithm to segment range images because it gives reasonable results and is computationally expedient.

Although range image segmentation by region extraction is popular in the literature, we have avoided this approach on the grounds that:

* One or more parameters are required to govern operations such as merging and when to stop growing regions. These parameters would depend on factors such as the amount of noise present in the image and the radius of curvature of surfaces on objects in the scene.
* Adjacent natural object faces which are separated by smooth edges may be incorrectly merged.

The CLUSTER algorithm does not require setting any parameters; nevertheless it performs well in the sense that

clusters (patches) do not generally cross over natural object face boundaries.

CHAPTER IV
SURFACE CLASSIFICATION

This chapter deals with deriving symbolic information about range image segments. A segment, or surface patch, is classified as planar, convex, or concave to characterize the "sense" of the corresponding object face. These classifications provide attributes of surface patches which are useful for object recognition.

There are two issues to be considered:

(1) How can we recover from oversegmentation of natural object faces? and,

(2) How do we utilize these patches to achieve object recognition?

To deal with these issues, we need to derive more information about each patch and about relationships between pairs of patches. This chapter treats the problem of classifying surface patches, and Chapter 5 treats the problem of classifying boundaries between surface patches. Section 4.1 briefly surveys the general topic of surface classification; in Section 4.2 we introduce a number of techniques for patch classification, and demonstrate in Section 4.3 how these techniques can be integrated to form a global decision about the surfaces. Finally, Section 4.4 presents some results and conclusions.

4.1 -- Background

We need to answer the following two questions about surface patches:

(1) What kind of surfaces do these patches represent? This question involves determining whether a surface comes from, for example, a plane or sphere or cylinder. The orientation and displacement of planes (corresponding to planar patches) has been useful in object recognition [Gri84] for polyhedral objects.

(2) What are the parameters that specify each patch/surface? For example, we would like to derive the radius and translation of the parent sphere to which a spherical patch belongs.

To answer question (1) we need to assume a priori what kinds of surfaces we expect to encounter. In many cases the set of primitive surface types is restricted to a single surface type, such as planes [Mil80] or generalized cylinders [Kua84]. These approaches can handle essentially any complexity of object, but do so only by potentially involving a large number of patches. Planar surface extraction is a popular tool for range image analysis [Dud79],[Mil80]. We present some of the work related to planar region extraction, shape classification, and derivation of surface parameters.

A popular school of thought in range image processing contends that certain sets of mathematical features or properties of surfaces motivated by differential geometry should suffice to provide surface classification. For example, the characteristics known as mean curvature and Gaussian curvature are invariant to translation, rotation, and surface parameterization. Besl and Jain [Bes84] use the signs of mean and Gaussian curvature to assign each pixel to one of eight different surface types, and detect critical points, such as jump and roof edges, as well as several other characteristics, such as direction of principal curvature, to

generate a rich representation of a range image. A general matching procedure is sketched but not implemented. Brady [Bra84] proposed a "curvature patch" representation. The direction of principal curvature and the directions in which the normal curvature is zero, as well as the principal curvature and geodesic curvatures at each zero normal curvature direction are collected at each pixel. These local features are grown to produce contours on the surface, called "lines of curvature". It is proposed that surface regions are delineated by these contours.

Duda et al. [Dud79] perform a sequential process of planar region extraction from range and reflectance images of a scene by first extracting horizontal planes (which have one degree of freedom), then vertical planes (with two degrees of freedom), and finally arbitrarily slanted planes (having three degrees of freedom). This approach is most appropriate for man-made scenery such as an office scene. By appropriate transformation, pixels in the range image are converted to real-world (x,y,z) coordinates, where z is height and x and y are floor positions. A histogram of z values indicate horizontal planes as peaks. Vertical planes can be detected by strong linear clusters in the x-y plane and are detected by Hough transform. Finally, reflectance data is used to detect the remaining planar regions under the reasonable assumption that reflectance is more or less constant over a planar surface. As each potential plane is postulated, points on this plane are derived by "slicing" the 3D coordinates into a "sandwich region" consisting of pixels that fall within a small distance W of the postulated plane. This slice is then "cleaned" and the larger components tested via a planarity test for final acceptance. The planarity test is based on the distances of the pixels in the slice from the postulated plane. The distribution of distances tends to be uniform if the region is not planar, and essentially normal with mean 0 if the region is planar. Coefficients corresponding to uniform and normal components

of a mixture distribution are estimated, and planarity is accepted if the uniform coefficient is sufficiently smaller than the normal coefficient. This approach is designed to work well in situations where the scene is expected to contain mainly horizontal and vertical planes.

Milgram and Bjorklund [Mil80] extract planes by a region-growing process. They find a best-fitting plane at each pixel over a 5x5 neighborhood, and store its orientation, altitude, and residual (goodness of fit) as features. Planar regions are grown by (1) ignoring pixels whose residuals are too large, and (2) otherwise merging the pixel with none, one, or both of its upper and left-hand neighbors (with merging of corresponding regions) based on how well the orientation and altitudes match. The result of this process is a segmentation of the range image into planar components with associated normals, altitudes, sizes, and total residuals, which are then matched with an a priori reference plane list to ascertain sensor position in the real world. The process was applied to four real range images of a single scene of a building site, and to two synthetic images. Three thresholds based on error-of-fit, orientation, and proximity of a point to a given planar region were defined to drive the plane-growing technique, and consistency of the results over the different views and similarity with ground truth were pointed out. The effects of varying the three thresholds were not discussed.

A region-growing technique is also used by Faugeras et al. [Fau83] to segment range data into planar patches. Their approach begins with a triangulation of the points on the object surface; each triangle is a planar region with associated points, border, and parameters of the plane (e.g. normal vector and distance from the origin). Pairs of adjacent regions are merged such that the induced error is minimal over all region pairs and the error does not exceed a pre-specified threshold. When no merging can take place

under this criterion, the process stops. This technique is applied to a Renault part and detects 500 and 60 regions under two different error thresholds. A procedure to segment a range image into quadratic patches is described but no examples are given.

Surface patches are classified into various shape primitives by Ittner and Jain [Itt85]. The surface primitives are: a plane, a sphere of radius 3 units, a cylinder of radius 3 units and height 6 units, and a cone with base radius 3.5 units and height 8 units. The classification is based on the fact that the distribution of curvature for a surface depends on parameters of the surface, such as radius of a sphere or cylinder. Reference cumulative distribution functions of six curvature measures for each primitive are formed by synthetically generating 100 surfaces of each type, degraded by noise, and observing these measures on synthetic surfaces. Given a surface patch to be classified, CDF's of the six curvature measures for this patch were constructed. A Kolmogorov-Smirnov 2-sided distance measure is used for each curvature measure to compare the corresponding empirical CDF with the four reference CDF's for the four primitive shapes. The lowest distance value indicates the best match-primitive for that curvature measure. A majority vote over the six best match-primitives is used to decide the final classification. Furthermore, if the majority is not strong enough, the patch is segmented by CLUSTER [Dub76] and the resulting subpatches reclassified. A disadvantage of this approach is the need to synthetically generate many instances of a given surface in order to derive curvature distributions. This is a difficult task unless the surface is simple (e.g., plane, sphere, cylinder). For particular, surfaces found on the Cobra sculpture (Figure B-15(a)) would be difficult to characterize.

Oshima and Shirai [Osh79] classify surface regions as curved, planar, or undecided based on two statistics derived from the regions. A 240x400 range image is broken up into overlapping 8x8 surface elements, to produce a 60x80 image of surface elements. A plane is fit to each surface element by the least squares method. Surface elements are merged based on errors of fit of neighboring surface elements to produce approximately planar regions called elementary regions. Each elementary region is classified by calculating two statistics D and E: D is the average squared angle between the planes fit to surface elements that make up the elementary region and the best fitting plane to the elementary region; E is approximately the diameter of the largest sphere that can be enclosed by the elementary region. A value $g = aD-E$ is defined, where a is some constant, and classification is made as follows:

If $E < E_t$ then the region is undefined;
otherwise, if $g < -g_t$ the region is planar;
if $g > g_t$ the region is curved.

This procedure classifies those patches which are too small or slender (small E) as undefined. Classification is based on the general idea that large D indicates curvature, unless E is also large. Curved regions are merged to form global curved regions by thresholding the angle between corresponding normals of the two neighboring patches that are classified as curved. Adjacent curved patches are merged if they appear to be smoothly joined, and the final curved patches are classified by finding the best-fitting quadric surface function. Because smoothly joined curved patches are merged, this technique is not ideal for applications which involve sculptured objects. Difficulties with quadric fits to real data have been documented in Section 3.2.

Hall et al. [Hal82] and Muller and Mohr [Mul84] use the theory of quadric surfaces to classify a surface into one of several curvature classes. They make use of four quantities that are invariant to translation and rotation, and three other quantities whose signs are similarly invariant. In [Hal82], a decision tree tests whether a given invariant quantity is less than, equal to, or greater than zero to determine the surface type. The one example performed is classification of a side of a coffee cup. It is classified as a hyperboloid, which in the decision tree requires no decisions that certain invariants are zero. They do not specify how, with noisy real data, to decide that an invariant is "equal to zero", though this seems to be an important part of their classification procedure. In [Mul84], a Hough transform technique is used to find the parameters of the surface (e.g., radius and center of a sphere), once the type of surface is determined. The only example worked out is a synthetic scene with unspecified (probably zero) noise degradation. We have shown in Section 3.2 how small noise degradation can drastically affect a quadric fit.

4.2 -- Statistics for Classification

Although classification of surface patches into quadric surfaces would be very useful for the task of identifying natural object faces and recognizing 3D objects, we have seen that no techniques are available which are insensitive to noise. Also, we do not assume that objects are constructed only from simple volumetric primitives such as spheres and cylinders. A classification of surface patches as planar, convex, and concave appears to be well suited for surfaces of arbitrary objects.

We have combined three techniques for classifying a patch as planar, convex, or concave:

(1) A rank-based trend test,

(2) Eigenvalue analysis, and

(3) Differences of normals.

4.2.1 -- Nonparametric Trend Test for Planarity

The trend test for planarity operates under the observation that as two points move in opposite directions from each other from a given point on a curved surface, the distance of closest approach of the line connecting the points to their starting point will tend to increase monotonically. For planar (or noisy planar) surfaces, this distance should meander about zero. A nonparametric rank-based test can be applied to determine whether there is a significant trend in these distances. Figure 4-1 illustrates this concept, and shows a slice of a range image patch.

First we define $(\theta_i^r, \theta_i^c)_{i=0}^{3}$:

$$\theta_i^r = \sin(i\pi/4)/\max(\sin(i\pi/4), \cos(i\pi/4))$$
$$\theta_i^c = \cos(i\pi/4)/\max(\sin(i\pi/4), \cos(i\pi/4))$$

The pairs (θ_i^r, θ_i^c) correspond to increments for stepping across an image in horizontal (0,1), diagonal slope-negative (1,1), vertical (1,0), and diagonal slope-positive (1,-1) directions. Also, denote by L(r,c) the label of pixel $p_{r,c}$, or the segment to which it belongs after clustering and cleaning is performed. Hence pixels $p_{r,c}$ and $p_{r',c'}$ belong to the same patch if and only if L(r,c)=L(r',c').

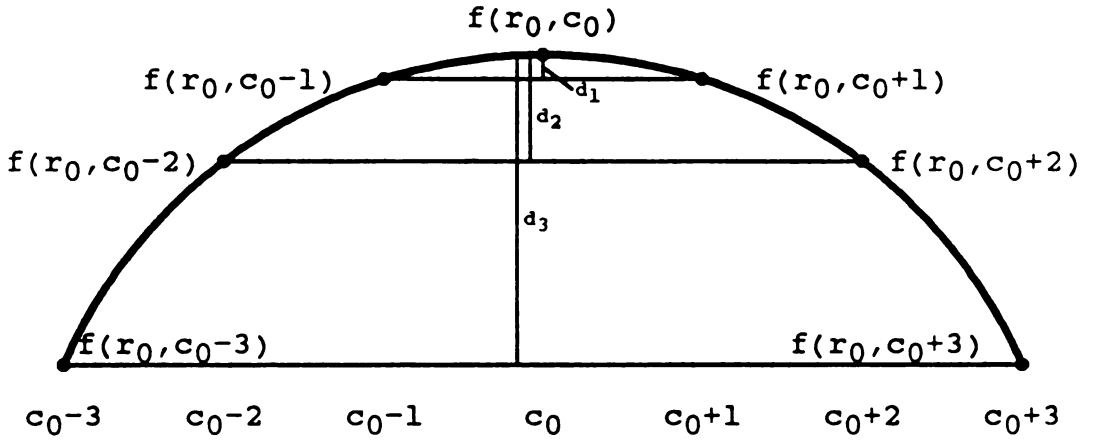


Figure 4-1
Trend Test Illustration

Suppose we are investigating patch P, having label b, with respect to direction index j. For every pixel $p_{r,c}$ in P define

$$E_j(p_{r,c}) = \min\{ i : L(r+i\theta_j^r, c+i\theta_j^c) \neq b \text{ or } L(r-i\theta_j^r, c-i\theta_j^c) \neq b \} - 1.$$

$E_j(p_{r,c})$ measures the number of pixels you can travel in opposite directions (along direction j) from $p_{r,c}$ before encountering the boundary. Let

$$E_j^* = \max_{p_{(r,c)} \in P} (E_j(p_{r,c})) = E_j(p_{r_0,c_0})$$

where (r_0,c_0) is chosen arbitrarily if not uniquely defined. Figure 4-2 illustrates the derivation of E_j^* for a sample patch, for j=0, 1, 2, and 3. The four thin lines indicate the lines of greatest span in the four directions. For example, one may travel up and down 5 pixels from the center pixel of the vertical line before hitting the boundary; since there is no pixel from which one may travel up and down more than 5 pixels before hitting the boundary, $E_j^*=5$ for direction j=2.

Define $\Psi_j = \max(\lfloor E_j^*/10 \rfloor , 1)$

and $\eta_j = \min(E_j^* , 10)$

where $\lfloor x \rfloor$ represents the greatest integer less than or equal to x. The number Ψ_j represents the number of distances which will be used by the trend test: exact distributions of the statistic to be defined have been published for $\Psi_j \leq 10$, hence the use of the factor 10 in the above definitions. The number η_j is the number of steps of length Ψ_j that may be taken in opposite directions (along direction j) from $p_{r0,c0}$ without stepping outside of P. Now consider the points δ_i, $i = -\eta_j, \ldots, \eta_j$, where

$j=0$: $E_j^*=5$

$j=1$: $E_j^*=3$

$j=2$: $E_j^*=5$

$j=3$: $E_j^*=3$

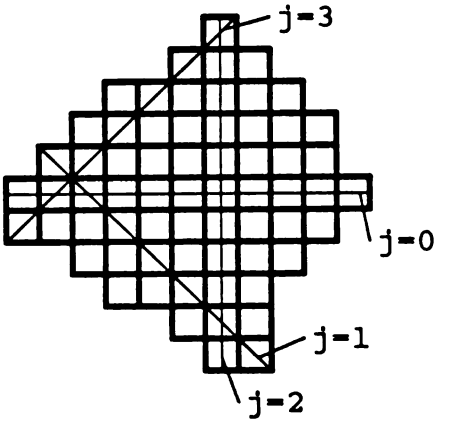


Figure 4-2
Derivation of E_j^*

$$\gamma_i = \bar{p}_{r_0+i\psi_j\theta_j^r, c_0+i\psi_j\theta_j^c} \quad .$$

These points form a cross section of the range image with length $(2\eta_j+1)$ centered at (r_0,c_0) and cut in direction (index) j. Derive a sequence of distances d_i, $i=1,\ldots,\eta_j$ such that d_i is the distance of closest approach of the line connecting points γ_{-i} and γ_i to the point γ_0, multiplied by -1 if the line passes below γ_0 (that is, if

$$f(r_0,c_0) > [\, f(r_0-i\psi_j\theta_j^r, c_0-i\psi_j\theta_j^c) + f(r_0+i\psi_j\theta_j^r, c_0+i\psi_j\theta_j^c) \,]/2.$$

This negative factor is necessary because without it we could observe a definite trend in the absolute distances created by lines passing above and below γ_0 alternately, clearly an instance of no trend. As an example, suppose $j=0$ and $\eta_j=3$. Then the γ_i's all have r-coordinates equal to r_0, thus forming a horizontal cross section of the range image. This situation is illustrated in Figure 4-1. Since the r-coordinate is fixed, each γ_i can be plotted by its c-coordinate and its range value. The lines connecting γ_{-i} and γ_i for $i=1,2,3$ are shown and the distances of closest approach of these lines to γ_0, namely d_1, d_2, and d_3, are indicated in Figure 4-1.

Construct a rank-ordering of these distances, randomly breaking ties if they occur, to obtain a rank sequence (r_i). The statistic S_j defined as

$$S_j = \sum_{i=1}^{\eta_j} i * r_i$$

is well-known in nonparametric statistics literature as Spearman's rank correlation statistic [Con80]; tables exist for various η_j. Under the null hypothesis of no curvature, the rank order of the (d_i) sequence should be independent of the sequence $(1,2,\ldots,\eta_j)$. We select thresholds on S_j that

give a two-sided trend test with size about .05. If S_j is too large or too small, we reject the surface patch as planar in direction j.

We derive a planarity confidence statistic. However, if for any direction j we find $\eta_j<6$, the patch is too narrow in that direction and the trend test cannot be applied; in such a case the statistic is left undefined. Otherwise we proceed as follows:

Define $\bar{S}_j = E(S_j) = \eta_j(\eta_j+1)^2/4$;
$u_{2.5}$ = the 2.5% upper quantile of statistic S_j;
u_0 = the maximum possible value of S_j;

Derive $v_0 = |\bar{S}_j - S_j|$;
$v_1 = u_{2.5} - \bar{S}_j$;
$v_2 = u_0 - \bar{S}_j$.

The theory of linear fractional transformations [Con78] provides a function T_j of v_0 with range [-1,1] such that T_j=-1 gives strong evidence of curvature (in direction j) and T_j=1 gives strong evidence of planarity (in direction j), given by:

$$T_j = \frac{v_2(v_0-v_1)}{2v_0v_1-v_2(v_0+v_1)} \qquad j=0,1,2,3.$$

Thus we obtain up to four measures which may indicate curvature in a surface patch with respect to specific directions.

Since we are using signed distances in the trend test, we can base our decision about convexity and concavity on the extremity of the distribution in which the statistic S_j lies. Small values of S_j will indicate convexity and large values of S_j will indicate concavity. The time complexity of deriving S_j for all surface patches is $O(N_p)$, where N_p is the number of object pixels.

4.2.2 -- Eigenvalue Planarity Test

The planarity test using eigenvalue analysis is based on the fact that the smallest eigenvalue of the covariance matrix computed from 3D points lying on a plane is zero (or close to zero in noisy situations). This test proceeds by computing the three eigenvalues e_1, e_2, and e_3 ($e_3 \leq e_2 \leq e_1$) for the 3D points in a surface patch. The time complexity of this operation is $O(N_p)$. The "planarity confidence" statistic E is given by:

$$E = \frac{e_3 - ce_1}{e_3(2c-1) - ce_1}$$

where c is a value of e_3/e_1 derived for a synthetic noisy plane with noise level 2g and size 20x20 pixels. Note that E lies between -1 and 1, with -1 corresponding to $e_3/e_1=1$ which indicates nonplanarity, 0 corresponding to $e_3/e_1=c$, the threshold value, and 1 corresponding to the case $e_3/e_1=0$ which indicates planarity (or linearity). To illustrate, below we present the three eigenvalues for a noisy planar patch (noise level 0.5g) and a convex spherical patch:

Noisy plane: $e_1=0.7816$, $e_2=0.6238$, $e_3=0.0008$
Convex sphere: $e_1=0.6987$, $e_2=0.6987$, $e_3=0.1538$

Corresponding values of E are, respectively, 0.960 and -0.705. We expect that the corresponding value of E will decrease as noise is increased for a planar patch. If noise is held constant and the patch size increases for a planar patch, however, E will increase.

4.2.3 -- Differences of Normals

The difference of normals test decides between convexity and concavity, and is useful if the other tests indicate nonplanarity. Given two pixels p and q on a surface patch, we define a difference of normals $\nu(p,q)$ as

$$\nu(p,q) = \| \bar{n}_p - \bar{n}_q \| * s(\bar{p},\bar{q})$$

where $\| \|$ denotes the Euclidean norm, $\bar{n}_p$ denotes the estimated unit surface normal vector at $\bar{p}$, and $s(\bar{p},\bar{q})$ is a "sign" factor which is 1 or -1 to indicate whether the vectors $(\bar{p}+\bar{n}_p)$ and $(\bar{q}+\bar{n}_q)$ point away from each other (convexity) or toward each other (concavity), respectively:

$$s(\bar{p},\bar{q}) = \begin{cases} 1 \text{ if } d(\|(\bar{p}+t*\bar{n}_p)-(\bar{q}+t*\bar{n}_q)\|)/dt \Big|_{t=0} \geq 0 \\ -1 \text{ otherwise.} \end{cases}$$

Thus if the two rays $\bar{n}_p$ and $\bar{n}_q$ emanating normal to pixels $\bar{p}$ and $\bar{q}$ initially approach each other, the curvature is classified as negative, indicating concavity; otherwise the curvature is positive, indicating convexity. See Figure 4-3.

For each pixel p in the patch we compute $\bar{\nu}(p)$ defined as

$$\bar{\nu}(p) = (1/|\Omega(p)|) * \sum_{\substack{q \in \Omega(p) \\ q \neq p}} \nu(p,q);$$

where $\Omega(p)$ is a neighborhood of pixel p. In our experiments we define $\Omega(p)$ to be the set of pixels whose corresponding rows and columns differ from the row and column of p by either 0 or 5. Hence we subsample the image by every 5th row and every 5th column to get $\Omega(p)$. We perform this subsampling because noise degradation of $\bar{\nu}$ is apparent for

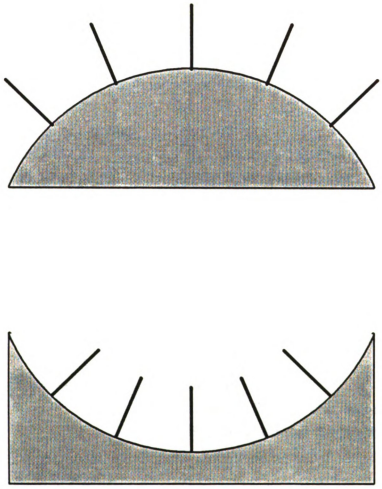


Figure 4-3
Convex vs Concave Illustration

smaller sampling intervals. After computing $\bar{\nu}$ for all pixels in the patch, the median value D of $\bar{\nu}$ is found. The time complexity of finding D for each patch is $O(N_p \log N_q)$, where N_q is the maximum of the number of pixels in the patches. If $D<0$, then most differences of normals within the patch indicated concavity and hence we can conclude that the patch is concave. If $D>0$, we conclude that the patch is convex.

4.3 -- Tree Decision Procedure

We have obtained a set of statistics for each patch in our segmentation: T_0, T_1, T_2, and T_3 by the trend test, E by eigenvalue analysis, and D by difference of normals. Note that one or more of the trend values may be undefined due to insufficient region span in various directions. Each T_j value implies something about the sense of the surface in question. Specifically:

If T_j is defined:
- $T_j \geq 0 \Rightarrow$ planar;
- $T_j < 0$ and $S_j < \bar{S}_j \Rightarrow$ convex;
- $T_j < 0$ and $S_j > \bar{S}_j \Rightarrow$ concave;

(where $\bar{S}_j$ is the expected value of S_j defined in Section 4.2.1)

$E > 0 \Rightarrow$ planar;
$E < 0 \Rightarrow$ nonplanar;

$D > 0 \Rightarrow$ convex;
$D < 0 \Rightarrow$ concave;

Given these individual classifications of a patch, which are not necessarily in agreement, we need to construct a procedure for eliciting an overall classification. Define the following:

T(null) = the number of undefined trend values
T(plan) = the number of planar trend decisions
T(cvex) = the number of convex trend decisions
T(ccav) = the number of concave trend decisions
TM ϵ {plan,cvex,ccav} the trend decision corresponding to that trend value with greatest magnitude (undefined if T(null)=4).
EI ϵ {plan,nonp}, the decision based on eigenvalues.
DN ϵ {cvex,ccav}, the decision based on difference of normals.

Our classification scheme is designed on the general strategy to base the decision on the trend test if T(null) is not too large; otherwise, information provided by the eigenvalue-based decision EI and the difference of normal decision DN are included. The tree decision procedure is as follows:

```
if T(null) ≠ 4
|   if T(null) ≤ 1
|   |   if ∃ type ∈ (plan,cvex,ccav) such that T(type)≥3
|   |      class = type
|   |   else
|   |      if T(DN)=2
|   |         class = DN
|   |      elseif EI=plan and T(plan)=2
|   |         class = plan
|   |      else
|   |         if TM = plan
|   |            if ∃ type ∈ (cvex,ccav) such that T(type)≥2
|   |               class = type
|   |            else
|   |               class = plan
|   |         else
|   |            class = TM
|   else
|      if ∃ type ∈ (plan,cvex,ccav) such that T(type)=2
|         class = type
|      else
|         if T(DN)>0
|            class = DN
|         elseif (EI = plan) and T(plan)=1
|            class = plan
|         elseif EI = plan
|            class = plan
|         else
|            class = DN
else
   if EI = plan
      class = plan
   else
      class = DN
```

Figure 4-4 shows a segmentation of the bottle range image in Figure B-3(a) into patches numbered 1 through 6. Table 4-1 reports the four trend statistics, the eigenvalue-based statistic, the difference of normals statistic, and the final classification based on our tree classifier for each of these patches. Each row corresponds to a patch; the four trend values are signed "+" or "-" corresponding to decisions of convexity or concavity, respectively. No sign corresponds to a decision of planarity. Those trend values which do not exist due to insufficient size of η_j are indicated by ":::".

Figure 4-4
Patches for Classification Example

Table 4-1
Statistics for Patch Classification of Bottle

Patch	T_0	T_1	T_2	T_3	E	D	Class
1	0.610	0.223	0.555	0.536	0.925	+0.011	Planar
2	-1.000	+0.828	-0.361	:::	0.037	-0.221	Concave
3	0.745	-0.644	0.026	0.925	0.921	-0.028	Planar
4	+0.747	+0.672	+0.594	+0.787	0.042	+0.367	Convex
5	:::	:::	:::	:::	0.839	+0.123	Planar
6	+0.241	:::	+0.389	+0.558	0.133	+0.325	Convex

The classifications are correct for the bottle; the surface of the bottle corresponding to patch 2 has slightly raised spines at the corners of the bottle, so that in a direction parallel to the planar sides of the bottle we could say that there is a concave nature to the patch, which might explain the concave classification of that patch.

To evaluate the performance of our classification technique, we generated synthetic range images consisting of five 30x30 pixel surface patches with noise level bg. The five surfaces were (with R in inches):

(1) Convex spherical, with radius of curvature R;

(2) Concave spherical, with radius of curvature R;

(3) Planar;

(4) Convex cylindrical, with radius of curvature R;

(5) Concave conical, with radius of curvature at widest part R.

We applied three techniques to determine the surface type for each surface patch:

(A) The full tree decision technique described above;

(B) A decision based solely on the eigenvalue and difference of normals: if EI=plan then conclude that the patch is planar, otherwise conclude DN;

(C) A decision based solely on the trend values: if there exists a maximum value in the set (T(plan),T(cvex),T(ccav)), say T(type), then conclude that the patch has sense type; otherwise, conclude TM.

This experiment was performed for (b,R)= (1,1), (1,3), (2,2), (3,1), and (3,3). Each experiment involved generating 100 synthetic range images containing the five types of patches as described above (where no smoothing operation was performed), and classifying each type of patch under the three decision techniques (A), (B), and (C). For each patch type and each decision procedure we obtain a 3-tuple (n_{plan}, n_{cvex}, n_{ccav}) which gives the number of times (out of 100 trials) that the given patch type was classified as planar, convex, or concave, respectively, under the given decision. Sample contour plots of convex spherical patches with parameters (b,R)= (1,1), (1,3), (2,2), (3,1), and (3,3) are shown in Figures 4-5 through 4-9, respectively.

Table 4-2

Results of Classification for (b,R)=(1,1)

patch	n_{plan}	n_{cvex}	n_{ccav}	
(1)	0	100	0	\
(2)	0	0	100	\
(3)	100	0	0	decision (A)
(4)	0	100	0	/
(5)	0	0	100	/
(1)	0	100	0	\
(2)	0	0	100	\
(3)	100	0	0	decision (B)
(4)	58	42	0	/
(5)	0	0	100	/
(1)	0	100	0	\
(2)	0	0	100	\
(3)	100	0	0	decision (C)
(4)	1	99	0	/
(5)	0	0	100	/

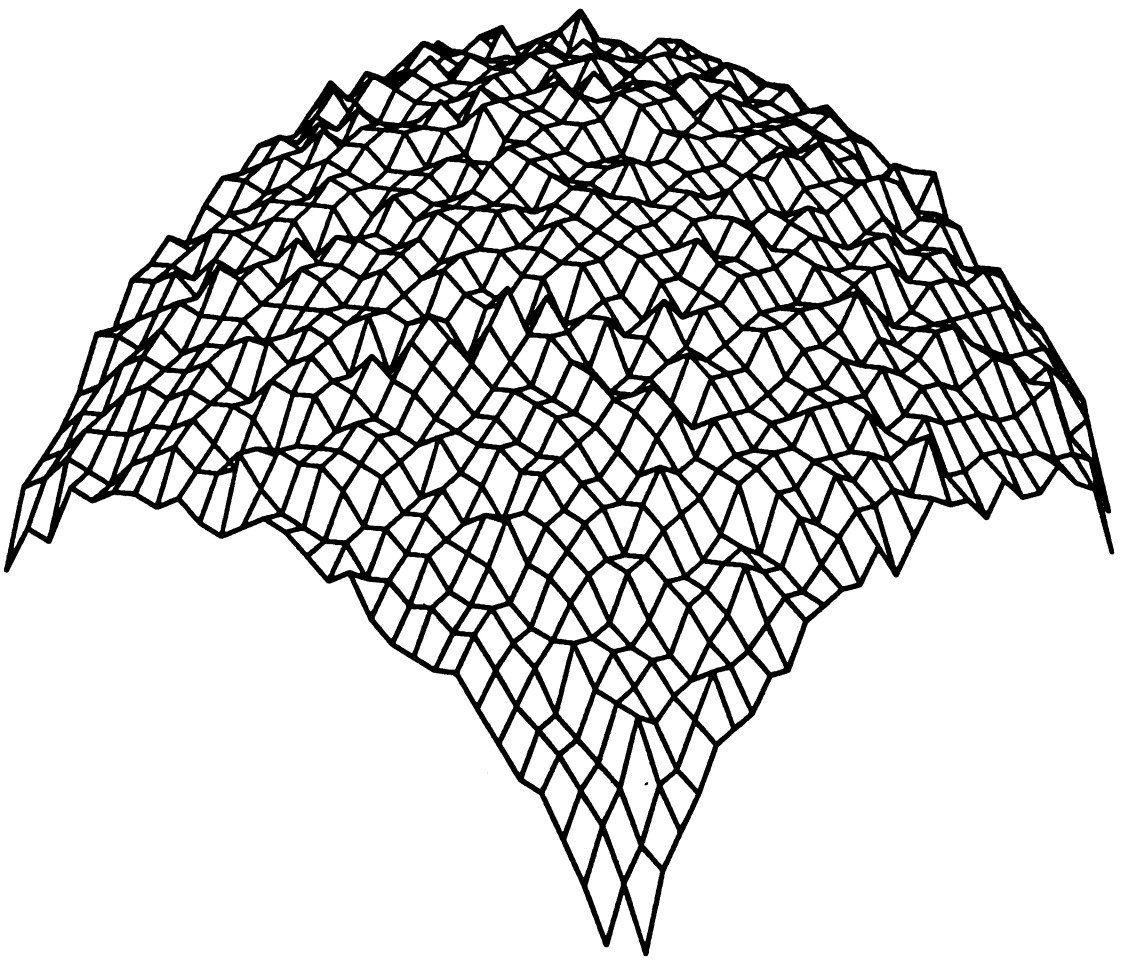


Figure 4-5
Convex Sphere with Parameters (b,R)=(1,1)

Table 4-3

Results of Classification for (b,R)=(1,3)

patch	n_{plan}	n_{cvex}	n_{ccav}	
(1)	18	82	0	\
(2)	17	0	83	\
(3)	100	0	0	decision (A)
(4)	74	26	0	/
(5)	0	0	100	/
(1)	100	0	0	\
(2)	100	0	0	\
(3)	100	0	0	decision (B)
(4)	100	0	0	/
(5)	100	0	0	/
(1)	45	55	0	\
(2)	43	0	57	\
(3)	100	0	0	decision (C)
(4)	93	7	0	/
(5)	8	0	92	/

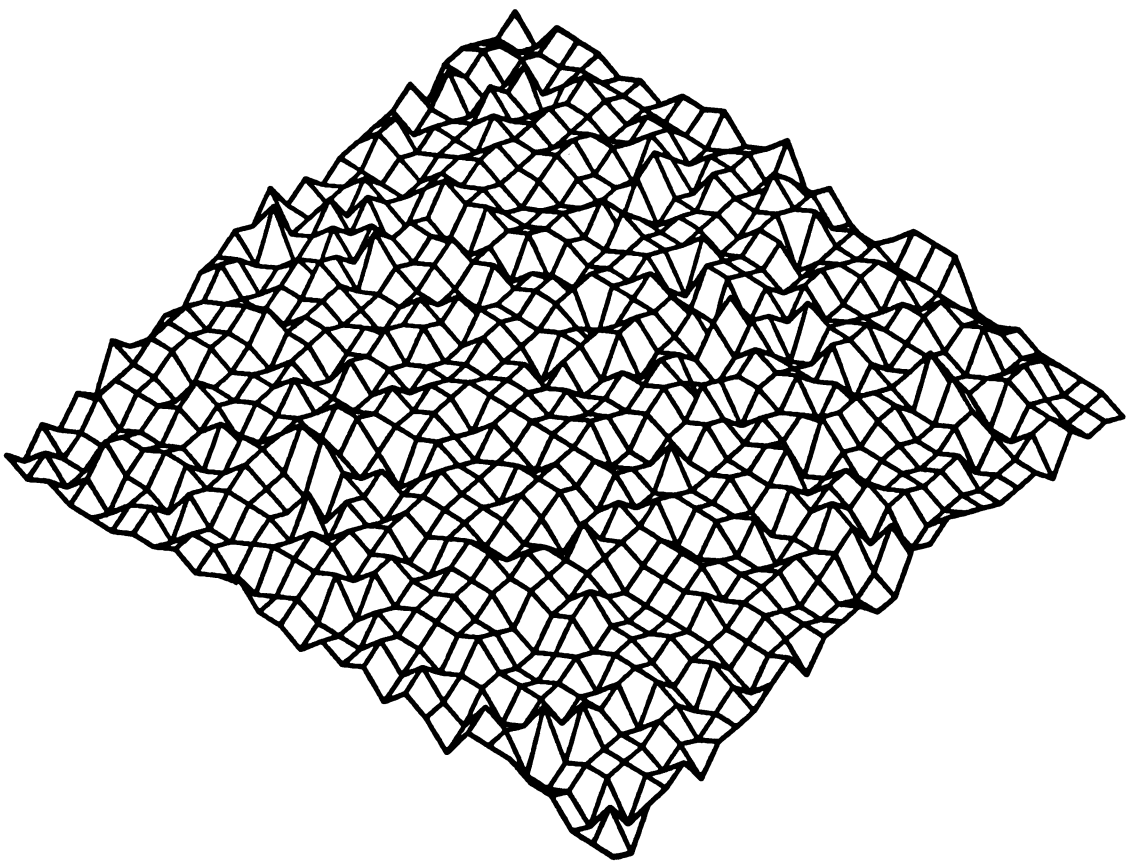


Figure 4-6
Convex Sphere with Parameters (b,R)=(1,3)

Table 4-4

Results of Classification for (b,R)=(2,2)

patch	n_{plan}	n_{cvex}	n_{ccav}	
(1)	41	59	0	\
(2)	32	0	68	\
(3)	100	0	0	decision (A)
(4)	88	12	0	/
(5)	25	0	75	/
(1)	100	0	0	\
(2)	100	0	0	\
(3)	100	0	0	decision (B)
(4)	100	0	0	/
(5)	100	0	0	/
(1)	61	39	0	\
(2)	62	0	38	\
(3)	100	0	0	decision (C)
(4)	96	4	0	/
(5)	65	0	35	/

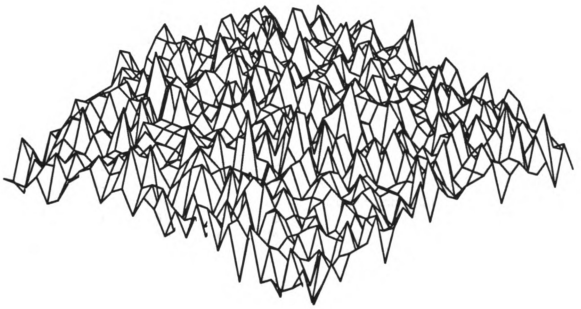


Figure 4-7
Convex Sphere with Parameters (b,R)=(2,2)

Table 4-5

Results of Classification for (b,R)=(3,1)

patch	n_{plan}	n_{cvex}	n_{ccav}	
(1)	2	98	0	\
(2)	1	0	99	\
(3)	99	0	1	decision (A)
(4)	75	25	0	/
(5)	23	0	77	/
(1)	0	100	0	\
(2)	0	0	100	\
(3)	100	0	0	decision (B)
(4)	3	97	0	/
(5)	0	0	100	/
(1)	3	97	0	\
(2)	3	0	97	\
(3)	100	0	0	decision (C)
(4)	91	9	0	/
(5)	46	0	54	/

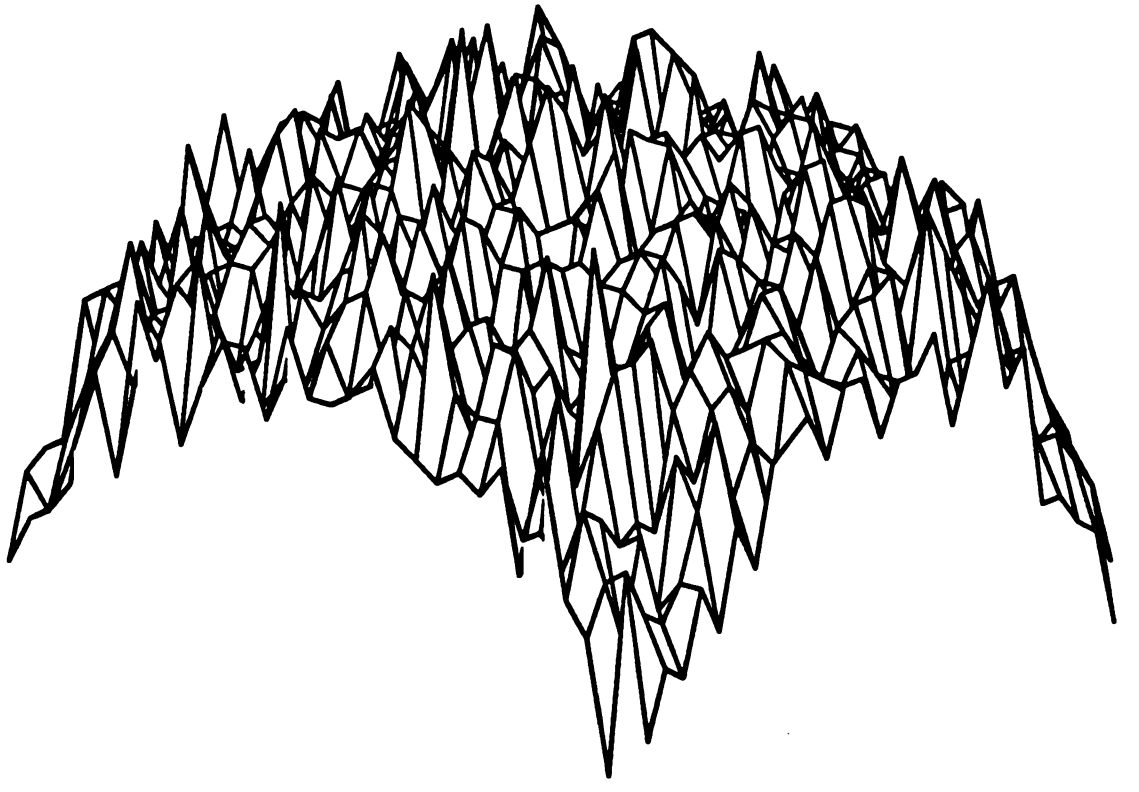


Figure 4-8
Convex Sphere with Parameters (b,R)=(3,1)

Table 4-6

Results of Classification for (b,R)=(3,3)

patch	n_{plan}	n_{cvex}	n_{ccav}	
(1)	94	6	0	\
(2)	97	0	3	\
(3)	100	0	0	decision (A)
(4)	96	4	0	/
(5)	78	0	22	/
(1)	100	0	0	\
(2)	100	0	0	\
(3)	100	0	0	decision (B)
(4)	100	0	0	/
(5)	100	0	0	/
(1)	100	0	0	\
(2)	99	0	1	\
(3)	100	0	0	decision (C)
(4)	100	0	0	/
(5)	99	0	1	/

From these results we can make a number of observations.

(1) All of the decision procedures have some trouble with the cylinder and conical surface. This is probably due to the fact that these surfaces have directions of zero curvature as well as directions of nonzero curvature. For the trend test, this means that these patches should be classified as planar on the basis of at least one direction.

(2) The eigenvalue test tends to classify those curved surfaces having a moderate radius of curvature (2" or 3") as planar. Therefore, decision procedure (B) has problems with the corresponding sets of experiments.

(3) The performance of the trend decision procedure degrades with increasing noise level. This could

Figure 4-9
Convex Sphere with Parameters (b,R)=(3,3)

be expected since the noise will tend to randomize the rank ordering of the distance sequence. The cases where the noise level is 3g is really an extreme case. Due to smoothing operations applied to our real range images, we usually work with noise levels of 2g or less in actual applications.

(4) With the exception of the experiment using (b,R)=(3,1), decision procedure (a) appears to perform better than the other two decision procedures, justifying our tree decision procedure. In the exceptional experiment, our emphasis on basing the decision primarily on the trend results works against us, due to the high noise (detrimental to the trend test) and the low radius of curvature (beneficial for the eigenvalue test).

The results of our classification procedure on our database of 31 range images are illustrated in Appendix B. Part (c) of each of Figures B-1 through B-31 shows a segmentation of the range images pictured in parts (a) with corresponding classifications indicated within the segments: "+", and "-" indicate a classification of convexity and concavity, respectively. If no mark occurs, then the patch was classified as planar. We observe that the classification is generally accurate, except when dealing with smaller patches, in which case a classification of planarity is common. From the 196 surface patches shown in part (c) of Figures B-1 through B-31, 69 out of the 77 of these which fall within a planar object face are classified as planar, 37 out of the 98 of these which fall within a convex object face are classified as convex, and 13 out of the 21 of these which fall within a concave face are classified as concave. Overall, 61% of the surface patches are classified correctly.

A note about the trend test: a certain amount of noise degradation is actually better than no noise in real data, due to the distortion induced by the range scanner system. In some cases this distortion is sensed by the trend test regardless of noise, particularly for larger planar patches.

4.4 -- Summary

In this chapter we have looked at the problem of classifying surface patches as planar, convex, or concave. We defined a trend test, an eigenvalue-based test, and a difference of normals test. The trend test is a nonparametric statistical rank order test and has performed well, with degrading performance for patches with high noise level and small numbers of pixels. We have designed a decision technique to make a global decision about a patch given the individual decisions provided by the trend, eigenvalue, and difference of normals tests, and have shown that the accuracy of the resulting classification is, in general, greater than that attained by decisions based only on the trend test or only on the eigenvalue and difference of normals tests.

CHAPTER V
BOUNDARY CLASSIFICATION AND MERGING

Surface patch classification is not enough to provide information for merging or recognition. For example, even though two adjacent surfaces are both planar, whether or not they should be merged depends on how they intersect each other. If the intersection forms a distinct crease on the object surface, then they belong to different surface patches, and the existence of the crease is a salient object feature. If their angle of intersection is very small, on the other hand, it would appear that they belong to the same large planar surface. This chapter addresses the task of classifying the boundaries between adjacent surface patches for the purpose of guiding a merging procedure. We define normal edges in Section 5.2, and develop a test for crease edge detection in Section 5.3. A merging procedure based on patch and boundary classifications is defined in Section 5.4. Section 5.5 introduces a linear goodness of fit feature for patch/patch and patch/background boundaries which is useful for object recognition.

5.1 -- Background

Edge detection has been a fundamental tool in image segmentation and object recognition. The same importance is attached to the detection of edges in range images. Jump edge detection has been discussed in Section 2.5. Although jump edges are readily detected by standard gradient operators, crease edges are considerably more difficult to detect. Most approaches detect edges by applying a certain process to all pixels in an image; the following summarizes edge detection research using this idea.

Ponce and Brady [Pon85] develop a technique to detect significant surface changes in range imagery. The two models of surface change considered are crease and jump edges, and the approach is based on smoothing locally cylindrical surface functions by convolving the surface with Gaussian filters of various scales. This "scale space smoothing" produces a sequence of progressively smoother images. In each smoothed image they detect zero crossings of the Gaussian curvature, retaining those points and their associated direction of principal curvature. Expressions are derived for the location of curvature extrema as a function of the spread σ of the Gaussian filter, for the cases of crease and jump edges, given that the object surfaces are planar. The behavior of the location of curvature extrema is different in the two types of edges; hence it is possible, given the sequence of smoothed images at different σ values, to ascertain the type of edge by observing the relative location over the various smoothed images. Artifact edges may be detected and rejected when they do not conform to the expected behavior of the location under the assumptions that the edge is a jump or crease edge.

Inokuchi et al [Ino82] use a ring operator to classify a pixel as belonging to one of: jump edge, convex crease edge, concave crease edge, and planar region. A circular path of pixels around the pixel of interest is used to define a periodic function. This function is decomposed into basis waveform components by a discrete Fourier Transform, and the amplitudes of the first three component waveforms are thresholded to arrive at the various classifications. Given a polyhedral scene, it is possible to obtain orientation information for those pixels classified as planar. By placing a point in amplitude-phase space for each "planar" pixel, individual planar faces may appear as clusters in this space.

Mitiche and Aggarwal [Mit83] design a crease edge detector which has low sensitivity to noise. The NxN neighborhood for each pixel is divided horizontally and vertically. Best-fitting planes are found for each half-neighborhood to derive surface normals and goodness-of-fit values. The difference in normals is thresholded to detect regions that are essentially flat. The most likely direction (horizontal or vertical) for an edge is derived via differences of normals and the goodness-of-fit values for remaining pixels. Probabilistic merit assessment removes those remaining points that are merely "near-edge" points.

Gil et al. [Gil83] consider using registered intensity and range images to derive a combined edge map. They assume the scene is composed only of planar surfaces and design a technique to detect jump and crease edges. Their approach is to first derive an angle of curvature value at each pixel in the image. Construct the 8-neighborhood of each pixel $\bar{p}_0$, $(\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_8)$, where $\bar{p}_1$ is that pixel lying directly above $\bar{p}_0$ and the rest of the sequence is obtained by travelling clockwise from $\bar{p}_1$ around $\bar{p}_0$. The angles θ_m, $m=1,\ldots,4$ are defined to be the angle formed by the intersection of the

lines $\bar{p}_m\bar{p}_0$ and $\bar{p}_0\bar{p}_{(m+4)}$. The angle of curvature at $\bar{p}_0$ is then $\max(\theta_1,\theta_3)$ if $|\theta_1-\theta_3|\geq|\theta_2-\theta_4|$, and $\max(\theta_2,\theta_4)$ otherwise. However, if there is a "significant range discontinuity" at $\bar{p}_0$ (the meaning of this is not defined), the angle of curvature is set to be 180 degrees plus the jump distance. Values below a threshold are set to zero, and values above the threshold are unchanged. A thinned version is produced by suppressing pixels which are not local maxima nor important for connectivity of the semithresholded pixels. This edge map is combined with a similar map derived from an intensity image.

These approaches are all fundamentally different from our procedure, since no prior information regarding location of edges is used by the edge detecting routines; individual pixels are tested for a "crease" quality, and those satisfying such a criterion must be connected to form crease edges. In our approach, edges are supplied a priori by boundaries between surface patches, and the task is to classify them.

5.2 -- Normal Edge Detection

The primary purpose for defining normal edges is to facilitate the reconstruction of large planar patches, and to provide an edge classification of some sort in the situation that a boundary between patches P_i and P_j is really a crease edge but is too short in number of pixels to make a reasonable decision based on the crease edge detection technique described in Section 5.3.

Large object patches tend to be broken up into subpatches. If a large planar surface is split into several smaller planar subpatches it would be a simple matter to compare the orientations of the subpatches and verify that they are nearly identical to assert that the subpatches are

in fact part of the same large plane. Unfortunately, in real images we observe a curvature tendency for large planar patches due to the range sensing technique. By computing the unit surface normals on the flat background portion of our real range images (the portion that is normally removed), we observe that these unit normal vectors at two different locations on a plane can differ by as much as 20°.

For each surface patch P_i we derive an average unit normal vector $\hat{n}_i$ as

$$\hat{n}_i = \bar{N}_i / |\bar{N}_i|, \text{ where}$$

$$\bar{N}_i = \sum_{p \in P_i} \bar{n}_p.$$

Recall that $\bar{n}_p$ is the estimated unit surface normal at pixel p. We derive a normal angle $N(i,j)$ for each pair of adjacent patches P_i and P_j as

$$N(i,j) = \cos^{-1}(\hat{n}_i \cdot \hat{n}_j) * s(\bar{p},\bar{q})$$

where $s(\bar{p},\bar{q})$ is the sign factor defined in Section 4.2.3, and where $\bar{p}$ and $\bar{q}$ are two arbitrarily chosen adjacent pixels belonging to, respectively, patches P_i and P_j. We call the boundary between patches P_i and P_j a normal edge if $N(i,j) > 20°$. We consider the existence of a normal edge to be essential for the existence of a crease edge, the detection of which is described next.

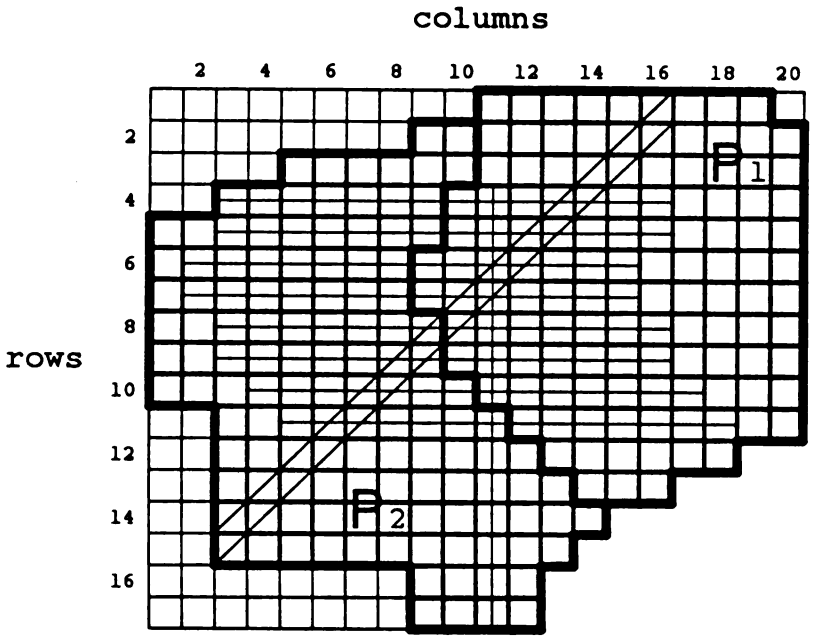
5.3 -- Crease Edge Detection

A crease edge occurs at a bend in the surface: The rate of change of surface normals (curvature) is discontinuous over such an edge. We have designed a test which checks for a crease as follows:

Let a_1 and a_2 be labels of two patches P_1 and P_2 having a common boundary along which a normal edge has been detected. For $j=0,\ldots,3$ define the set of border pixels

$$B_j = \{p_{r,c}:\ L(r+\theta^r_j i, c+\theta^c_j i)=b_1 \text{ for } i=0,\ldots,(n-1) \text{ and } L(r-\theta^r_j i, c-\theta^c_j i)=b_2 \text{ for } i=-n,\ldots,-1 \text{ for } (b_1,b_2) = (a_1,a_2) \text{ or } (a_2,a_1)\ \},$$

where (θ^r_j,θ^c_j) defines one of four directions (horizontal, diagonal slope-negative, vertical, and diagonal slope-positive) as defined earlier in Section 4.2.1. The set B_j consists of those pixels $p_{r,c}$ belonging to either P_1 or P_2 such that the pixels encountered in traveling n-1 steps from $p_{r,c}$ in direction (θ^r_j,θ^c_j) all belong to the same patch as $p_{r,c}$, and the pixels encountered in traveling n steps from $p_{r,c}$ in the opposite direction $(-\theta^r_j,-\theta^c_j)$ all belong to the other patch. This gives a chain of 2n pixels split down the middle by the P_1-P_2 border. Figure 5-1 shows B_j for two patches (shown bordered by heavy lines) when n=7. The thin lines passing through the common boundary indicate instances in which the above conditions are satisfied. For example, the top horizontal line indicates that the set of pixels $\{p_{4,i}:\ i=10,\ldots,16\}$ is contained in patch P_1 and the set of pixels $\{p_{4,i}:\ i=3,\ldots,9\}$ is contained in patch P_2, therefore pixel $p_{4,10}$ is in B_0.



$B_0 = \{ p_{4,10}, p_{5,10}, p_{6,9}, p_{7,9}, p_{8,10}, p_{9,10}, p_{10,11}, p_{11,12} \}$

$B_1 = \{\}$

$B_2 = \{ p_{11,11} \}$

$B_3 = \{ p_{8,9}, p_{9,9} \}$

Figure 5-1
Derivation of B_j Sets

Let $\ell_j=|B_j|$. Suppose $\ell_k=\max(\ell_0,\ell_1,\ell_2,\ell_3)$. If $\ell_k<3$, the test for a crease should not be made because the sample size is too small. Otherwise, for each $p_{r,c} \in B_k$: Define $\gamma(t)$, for $t \in [-n,n-1]$:

$$\gamma(t) = (\bar{p}_{r+\theta_k^r\lfloor t\rfloor,\, c+\theta_k^c\lfloor t\rfloor})(1-t+\lfloor t\rfloor) + (\bar{p}_{r+\theta_k^r(\lfloor t\rfloor+1),\, c+\theta_k^c(\lfloor t\rfloor+1)})(t-\lfloor t\rfloor) .$$

$\gamma(t)$ is the piecewise linear curve whose corner points are

$$\bar{p}_{r+i\theta_k^r,\, c+i\theta_k^c} , \quad i=-n,\ldots,n-1.$$

Define distances $d_1^{r,c}$, $d_2^{r,c}$, and $d_3^{r,c}$ which roughly measure the relative "warp" at three locations in the cross section: $d_2^{r,c}$ measures the warp over the boundary of P_1 and P_2; $d_1^{r,c}$ and $d_3^{r,c}$ measure the warps within P_1 and P_2. We expect that $d_2^{r,c}$ will be larger than $d_1^{r,c}$ and $d_3^{r,c}$ if there is a crease at the boundary of P_1 and P_2.

These distances are defined as follows:

- $d_1^{r,c}$: distance from the point $(\gamma(-n)+\gamma(-2))/2$ to the intersection point of the perpendicular bisector (plane) of the line segment with endpoints $\gamma(-n)$ and $\gamma(-2)$ with the curve γ.
- $d_2^{r,c}$: distance from the point $(\gamma[-(n-1)/2]+\gamma[(n-3)/2])/2$ to the intersection point of the perpendicular bisector (plane) of the line segment with endpoints $\gamma[-(n-1)/2]$ and $\gamma[(n-3)/2]$ with the curve γ.
- $d_3^{r,c}$: distance from the point $(\gamma(1)+\gamma(n-1))/2$ to the intersection point of the perpendicular bisector (plane) of the line segment with endpoints $\gamma(1)$ and $\gamma(n-1)$ with the curve γ.

As an example, suppose $n=7$, $k=0$, and $p_{r,c} \in B_0$. The points $\gamma(i)$, $i=-7,\ldots,6$ form a horizontal cross section of the range image and thus may be plotted by their c-coordinates and range values. The function γ is formed by connecting these points by lines. This is illustrated in Figure 5-2, which shows a crease edge between $\gamma(-1)$ and $\gamma(0)$. γ is indicated by the solid piecewise linear line connecting points along the slice. The derivation of distances $d_1^{r,c}$, $d_2^{r,c}$, and $d_3^{r,c}$ is pictured, and the fact that $d_2^{r,c}$ is larger than $d_1^{r,c}$ and $d_3^{r,c}$ reflects the presence of the crease edge.

For our experiments we use $n=7$: if it were larger, the crease detection at a pixel of B_j would be more reliable, but at the expense of obtaining smaller B_j sets, making detection of crease edges between regions less reliable.

We define

$$\beta = |\{p_{r,c} \in B_k: \ d_2^{r,c} > \max(d_1^{r,c}, d_3^{r,c})\}|.$$

We treat β as a Binomial, $Bin(\ell_k, 1/3)$, random variable, and decide there is a crease edge if β is too large under a binomial test with size .05. The time complexity of determining whether or not a crease occurs over all pairs of adjacent surface patches is $O(N_s N_p)$, where N_s is the number of surface patches and N_p is the number of object pixels.

We have tried to justify this procedure with theory. However, since our patch boundaries are not arbitrarily chosen, but are instead implicitly generated by the segmentation process, it is difficult to specify the null hypothesis. In fact, it appears that the null distribution of β depends on the noise level and amount of surface curvature. Hence it is possible that the false alarm error rate of this test will be unacceptably large.

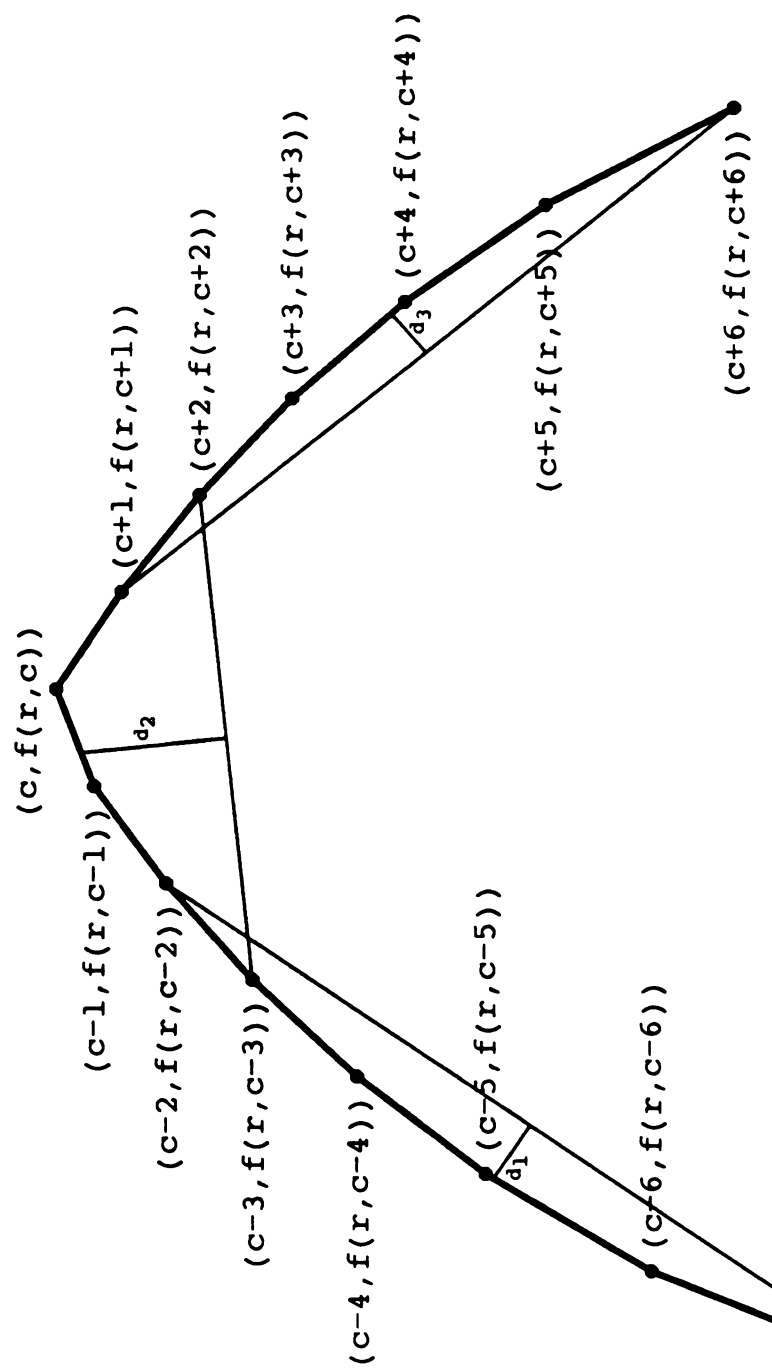


Figure 5-2
Crease Detection Illustration

We tested the false alarm error rate for two types of range images: planar surfaces and spherical (radius 2") surfaces, under two noise levels, 1g and 2g. Table 5-1 shows the resulting false alarm error rates (number of false alarms/number of boundaries tested) We observe that with noise the segmentation technique discovers many false crease

Table 5-1
False Alarm Error Rate for Crease Edge Detection

	Patch type	
	Planar	Spherical
Noise		
2g	12/80 (.15)	84/160 (.53)
1g	0/97 (.00)	44/160 (.28)

edges. It should be no surprise, then, when crease boundaries are detected where common sense says there is no such surface anomaly.

5.4 -- Merging Patches

We now describe a procedure for recovering from some of the oversegmentation typically produced by our segmentation scheme. We have at our disposal classifications of the surface patches and of the boundaries between these patches. We utilize a two-step merging procedure.

First, adjacent patches whose boundaries are neither jump, normal, nor crease edges are merged. We will call this type of boundary a null edge. This merges two patches P_i and P_j if their difference in orientation N(i,j) is less than a threshold (20°) and are not separated by a jump edge. This step tends to recover large planar patches. Next, these patches and their boundaries are reclassified, and two adjacent patches are merged if their boundary is a normal

edge, but not a crease edge, and the classifications of the individual patches are both concave or both convex. A normal edge between a planar patch and another patch might indicate a possible crease edge, and thus no merging is performed in such a case. This second step is designed to recover the larger curved patches.

To illustrate our merging procedure, Figure 5-3 shows the sequence of steps used for merging the patches obtained from the bottle range image. Figure 5-3(a) shows the original set of patch boundaries derived from AS3. Figure 5-3(b) shows the initial boundary classifications: the boundaries colored white, yellow, black, and red correspond to jump, crease, normal, and null edges, respectively. The first merge joins those pairs of patches whose boundaries are null edges; namely, the patches belonging to the planar sides of the bottle. Figure 5-3(c) shows the boundaries derived from these merged patches. Within each patch an indication of the patch classification is made: "+" and "-" correspond to, respectively, convex and concave classifications, and if neither occurs then the patch is classified as planar. Note that the curved head of the bottle has a normal edge forming the boundary between two patches both classified as convex. Therefore, the second merging step merges these patches; the edge map for the final merged patches is shown in Figure 5-3(d). Observe that a crease edge was also detected in the curved bottle head, and thus the merging procedure has almost but not totally recovered from the initial oversegmentation to provide segments, four of which correspond to natural object faces.

When we apply this merging scheme to our database of range images, we obtain the final segmentations shown in part (c) of Figures B-1 through B-31; the corresponding classifications of these patches are shown as "+" or "-" if the patch is classified as convex or concave, respectively, and planar if not marked. In general, we observe that:

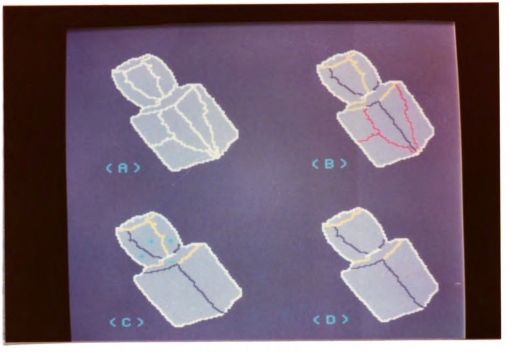


Figure 5-3
Example of Merging Procedure

(1) Planar patches are usually totally recovered. Of 63 planar natural object faces over the 10 objects, only 5 are oversegmented.

(2) Curved surfaces tend to remain oversegmented to some extent.

Recall that, after initial segmentation, 46% of natural object faces were not oversegmented. After merging, 88 out of 149 natural object faces, or 59%, were represented by single surface patches. Thus, merging appears to improve range image segmentations.

5.5 -- Linear Boundary Fits

Knowing whether or not a boundary is linear is useful for object recognition. A feature which may be derived from boundaries is the degree to which a 2D line fits the boundary pixels in the image plane. This may be applied to the boundaries occurring between surface patches and the background region as well as to those between pairs of adjacent surface patches.

We treat the case of linear fits to boundaries between pairs of surface patches first. Given two adjacent surface patches P_i and P_j, we derive the set B_{ij} of boundary pixel coordinates (r_k,c_k) consisting of all pixels of P_i which are adjacent to some pixel of patch P_j. A line is fit to these 2D points by a least squares technique. We obtain the goodness of fit of this line as the average squared error ϵ_{ij}, where the error contribution of each pixel (r_k,c_k) in B_{ij} is the distance of closest approach of the fitted line to the pixel:

$$\epsilon_{ij} = 1/|B_{ij}| \sum_{(r_k,c_k) \in B_{ij}} D^2_{ijk}$$

where

$$D_{ijk} = \left| \frac{a_{ij}r_k + b_{ij}c_k + d_{ij}}{\sqrt{a^2_{ij} + b^2_{ij}}} \right|$$

and $a_{ij}r + b_{ij}c + d_{ij} = 0$ is the equation of the best fitting line to B_{ij}.

The value ϵ_{ij} is the linear fit feature for the boundary between patches P_i and P_j.

The definition of the boundary between a given patch and the background requires a different treatment. To illustrate, consider the segmentation for range image AS1 shown in Figure B-1(b). The red patch corresponding to the flat side of the bottle has a boundary with the background which spans three distinct linear sides of the object surface. Fitting a line to this group of boundary pixels would certainly indicate a poor linear fit by the previously described technique. However, we would like to classify the boundary as "piecewise linear"; we need some procedure for detecting the "corners" in the polygonal boundary.

To accomplish this, we propose deriving a segmentation of the boundary pixels occurring between image and background, and using this segmentation to identify appropriate components for a given patch. For each pixel (r_i,c_i) which is on the perimeter of the silhouette image (defined in Section 2.7) and which we call a perimeter pixel, we derive a 4D spatial vector consisting of the image coordinates (r_i,c_i) and the 2D estimated unit normal vector computed by fitting a line to the set of perimeter pixels (r_j,c_j) satisfying $|r_i-r_j|\leq 5$ and $|c_i-c_j|\leq 5$. The CLUSTER algorithm described in Chapter 3 is applied to these 4D spatial vectors (where each feature has been normalized to have unit variance) with 10 clusters requested. We clean the resulting segmentation of perimeter pixels: detect connected components, and eliminate components which have fewer than 5 pixels. We end with some number A of connected segments (clusters) C_i, $i=1,\ldots,A$ of perimeter pixels.

Suppose we are in the process of finding the linear fit for the boundary B_j occurring between patch P_j and the background region. For $i=1,\ldots,A$ we derive $B_j \cap C_i$. If $|B_j \cap C_i|<5$, set the linear fit ϵ_i^* to be -1. Otherwise, fit a line to this set $B_j \cap C_i$ and derive the average squared error ϵ_i^*. Define the linear fit to the patch/background boundary of P_j, ϵ_j, to be $\max(\epsilon_i^*)$.

Although we save the actual values of the errors of fit for these boundaries, we use the value 1 as the threshold for deciding between linearity and nonlinearity in the recognition process. That is, if an ϵ-value is less than or equal to 1, we conclude that the boundary is linear (piecewise linear for patch/background boundaries). As an example, Table 5-2 shows the errors of linear fit for the boundaries found in Figure B-1(c).

Table 5-2

Errors of Linear Fit For AS1

Patch	Color	Error of fit with background
1	White	2.34
2	Red	0.45
3	Blue	0.34
4	Yellow	0.10

Adjacent patches	Error of fit of boundary
1 & 4	34.18
2 & 3	0.11
3 & 4	147.30

As desired, the boundary between patches 1 and 4 and the boundary between patch 1 and the background are correctly classified as nonlinear and the boundary between patch 2 and the background is correctly classified as (piecewise) linear. However, the boundary between patch 4 and the background is incorrectly classified as linear.

5.6 -- Summary

In this chapter we have developed techniques for classifying the boundary between two adjacent surface patches. These boundaries are classified as jump, normal, crease, or null edges. A procedure is outlined whereby this boundary information, along with the surface patch classifications, is used to merge adjacent surface patches to help recover from oversegmentation. This procedure performs well for planar object faces; unfortunately, a potentially high false alarm error rate for crease edge detection tends to prevent the full reconstruction of curved surfaces.

A potential technique for alleviating this difficulty involves the assumption that boundaries between natural object faces are "smooth". That is, the true boundary in 3D is, say, a 2nd order function of x, y, and z. By fitting an appropriately smooth function to a boundary between two surface patches to construct a new boundary in the image plane, the performance of the crease edge detector may become better behaved.

We have also derived features called linear fits which indicate how well a line can be fit to a patch/patch boundary or a patch/background boundary. These features will be useful for object recognition.

CHAPTER VI
OBJECT RECOGNITION

A computer vision system must eventually utilize knowledge of objects to recognize objects in the external world. The optimal stage in the processing sequence for introducing this external knowledge is not obvious. Psychophysical research into the human visual system seems to indicate that much analysis of visual stimuli occurs precognitively [Mar82], suggesting that we initiate this integration of knowledge after our analysis has derived what it can from the data without a priori knowledge -- in our case, surface patches and boundaries, and corresponding classifications of these patches and boundaries. The information derived from analysis is structured to form an object representation; the formal construction and use of these representations will be called the model (or model scheme). These two tasks, model construction and manipulation, are important in all object recognition systems. This chapter first reviews pertinent topics related to modeling and object recognition, then surveys the CV literature dealing with these areas. Finally, an evidence-based object recognition technique is defined. Patch and boundary information is combined to form a representation of a range image. A list of salient features of the various objects in the database forms the core of an object recognition system, which looks for instances of these features in the representation. Occurrences of these salient features are interpreted as evidence for or against the

hypothesis that a given object occurs in the scene. A measure of similarity between the set of observed features and the set of salient features for a given object in the database is used to determine the identity of an object in the scene or reject the object(s) in the scene as unknown.

6.1 -- Preliminary Issues

Certain essential questions which must be answered when designing an object recognition system are:

(1) What objects do we wish to recognize?
(2) What viewing conditions will we assume?
(3) What primitives will constitute a representation of our model?
(4) How is the model constructed from the primitives?
(5) How are the observed representations compared with database representations to achieve recognition?

These issues will now be discussed in more detail.

6.1.1 -- Domain of Objects

The choice of the class of objects to recognize is extremely important, for it has a strong influence on the model design itself. If only polyhedral objects will be recognized (an object domain widely used in the literature), a model could be designed from simple planar surface patch primitives. However, for sculptured objects (e.g. shoes, car parts) there is no simple family of surface patch primitives from which to construct a representation (unless one is willing to put up with a large number of primitives to make up a representation, as in approximating a sculptured object with small planar facets). If the objects can be articulated (that is, can assume a range of different shapes, as with a pair of scissors), then ideally the representation should be invariant to these articulations. The larger the

domain of objects which can be represented by a given model, the more useful the object recognition system will be -- but only if the recognition is computationally feasible and successful.

Not only must we choose a family of objects to represent, but also we must consider the viewing conditions of these objects. For example, it is possible to assume that objects will always be in stable positions, say resting on a flat surface. This allows the model to ignore those potential views of an object which turn out to be physically impossible [Sto84]. Also, if the image may only contain a portion of the object, as would happen if the object was too large for the field of view, then the resulting model should be designed so as not to be rendered useless by missing data. The most difficult of the situations is a jumble of objects. On the one hand, all possible object poses may occur when objects rest on other objects rather than on a flat surface. On the other hand, objects may occlude other objects.

6.1.2 -- Primitives

The analysis stage provides information about the sensed scene, and the recognition stage must take this information and derive a representation from it. In this sense the choice of primitives for the model is often strongly suggested by the form of the analysis output. Or, conversely, a priori specification of a model can influence what primitives should be derived in the analysis stage. The most common primitives in the literature are [Bes85]: points, lines, surfaces, and volume elements; point and line primitives are usually used together and constitute wireframe models, surfaces are used for boundary or surface-based models, and volume elements are used in volumetric models. Further discussion of characteristics of these modeling schemes is provided in section 6.2.

Having decided upon the primitives, the means of combining these primitives to construct the representations of the model is the next task. One useful dichotomy of approaches to this task is the distinction between unified models [Lu85] and characteristic view models [Cha82]. A unified model corresponds to a model design for which an object has a single representation, so that in some sense the representation stores information about all views of the object. A characteristic view model allows multiple representations for a given object, such that each representation corresponds to a distinct class of views of the object. Views within a class are mapped to the same representation. An advantage of characteristic views is the quicker recognition times than could be achieved with unified view models, since what is required for recognition is an exact match of the observed representation to some characteristic view representation rather than a mapping from the observed representation into many "subrepresentations" of a unified model; however, this economy in recognition time sometimes requires that objects be assumed to be in stable positions so that the number of characteristic views is not too large. A disadvantage is the need to construct all characteristic views; this is not an easy task, and is generally done by hand, requiring some degree of faith that all characteristic views have been discovered [Cha82].

6.1.3 -- Models

Another dichotomy involves means by which the primitives are combined to make a representation. This involves defining attributes and relations between primitives. It is usually possible to define an adjacency relation on the primitives, and such a relation forms the core of many object representation designs. For example, if the primitives are object surface patches, then adjacency of two patches would

require that the patches have some boundary in common. Generally, such a relation includes one or more attributes, values that represent a distance between primitives, such as a measure of the angle formed at the boundary of two surface patches. Of course, other relations can be defined, such as "parallel to", to enhance the information content of a representation, as well as to define the number of attributes for each relation.

6.1.4 -- Matching

The final important problem is matching an observed representation to the database of model representations to perform recognition. Some approaches [Ree85] derive a vector of numbers from a representation, so that identification consists merely of finding the closest model representation vector to the object representation vector. However, reducing an information-rich representation to a handful of numbers is perhaps better for screening out models which cannot possibly be the correct match than for recognizing objects. More commonly, recognition involves matching relational structures, which requires subgraph isomorphism detection, preferably with provisions for errors in the representations. An unfortunate problem is that the task of subgraph isomorphism detection is NP-complete [Gar79]; therefore most approaches use heuristics to speed up recognition, often resulting in sub-optimal techniques which, nevertheless, may perform well at recognition.

6.2 -- Background

Approaches to modeling 3D objects can broadly be partitioned into two schools: that of Computer Vision and that of CAD/CAM. Whereas Computer Vision tends to construct modeling schemes for which representations can be reliably obtained from sensed imagery and successfully used to achieve object recognition, CAD/CAM tries to construct modeling schemes best suited for internal manipulation of representations and display with an ultimate goal of guiding the manufacture of objects from such models. The issues of interest to these schools are very different, although convergence of these schools is increasingly noticeable [Yor81]. We deal solely with the modeling literature involved in Computer Vision.

One way of partitioning the literature dealing with modeling and recognition is by the dimensionality of the primitives used: approaches can use 3D volumetric primitives, 2D surface primitives, or 1D and 0D (line and point) primitives. The following subsections look at representative work from each of these three approaches. Finally, a survey of approaches to the matching problem is presented.

6.2.1 -- Volumetric Models

One technique for representing 3D objects is as a union of 3D volume elements sometimes known as voxels. Some drawbacks are immediately apparent. First, the voxels must be small in size to obtain reasonable resolution, and therefore a large number of them are required to represent a given object. Second, a single view of an object is insufficient for determining a set of voxels of the object; theoretically, the voxels can extend to infinity behind what is visible in the single view. Hence, multiple views from

widely varying vantage points are required to provide an unambiguous approximation of an object as a collection of voxels.

Martin and Aggarwal [Mar83] work with 2D projections of objects (silhouettes). By back-projecting a silhouette into 3D, a volume of infinite extent in the direction perpendicular to the silhouette plane is derived. An approximation of the 3D object may be obtained by finding the intersection of several of these infinite volumes corresponding to several silhouettes. A hierarchical linked list structure provides the object representation: at the top level of the hierarchy, voxels are classified according to their z values (voxels with the same z value are grouped together). Each "constant-z" group (planar slice of the object) is then broken into groups of constant x value (lines through the slice), and finally a linked list enumerates voxels belonging to each line. The claim is that the representation is both efficient and economical. Note that this technique is not guaranteed to detect indentations in an object.

Wang et al. [Wan84] extend the above work to perform 3D object recognition based on the hierarchical volumetric representation. First, the 3D reconstruction of the object is rotated so that its three principal axes line up with the x,y,z coordinate axes. Next, this rotated reconstruction is projected onto the xy, xz, and yz planes to form three silhouettes of the object. A Fourier transform technique is used to evaluate the shape of the silhouettes and any holes in these silhouettes. Also used are the pqr-th principal moments of the silhouette contour functions where pqr={000,002,020,200}. These features are compared to model features, and a decision is made based on the closest match. The number of initial views (silhouettes) of the object to be recognized is increased from a starting value of 2 views until consistent results are obtained.

Chien and Aggarwal [Chi85] modify the scheme of [Mar83] by representing silhouettes and reconstructed objects as generalized quadtrees and octrees, respectively. The term "generalized" means that area and volume elements are parallelepipeds, not necessarily squares or cubes. As in [Wan84], the reconstructed object is rotated according to its principle axes and projected onto the principal planes to form silhouettes, which are in turn represented as quadtrees. A dissimilarity measure between the quadtrees corresponding to a 3-view reconstruction and a priori models are used to select a set of potential matches; fine matching then derives a dissimilarity between the octree representations to select the best match.

6.2.2 -- Wireframe Models

The human visual system generally has no difficulty in recognizing objects depicted in crude line drawings. Hence there is evidence that recognition can be achieved from only edge and point information in a scene. This has been the approach taken by many researchers. However, due to the difficulties involved in manipulating and matching general spatial curve functions, much of the research in this area makes the assumption that edges will be linear or circular. By far the most common objects recognized by edge/point models are polyhedral objects, not only because the edges are linear, but also because the points of interest are well defined as intersections of these edges.

Hebert and Kanade [Heb85] utilize jump edge information to identify potential matches for an unknown polyhedral object. The model uses characteristic views in the sense that a view-sphere (which enumerates all possible viewing angles of a scene) is discretized into approximately 2000 uniform cells and the corresponding jump edge information

visible from a given vantage point corresponding to a cell is coded in each cell (by hand!!). The model-designer must also enter a "sequence of exploration" which specifies which edges to match first for optimal speed and accuracy. Once a small set of potential matches is found, higher level information in the form of planar face parameters is used to identify the correct object/position. Since, aside from a qualitative classification indicating the jump edge configuration there is also stored at each view-sphere cell certain attributes (angles between edges, bounds in length of line segments), the models tend to be large; a U-shaped polyhedral object requires about 200K words of memory.

Magee et al. [Mag85] make use of registered reflectance and range images. The intensity image guides range sensing in the sense that since range sensing takes so much more time per pixel than reflectance sensing, the reflectance information is used to derive points of interest for which range values should be sensed. These points are identified by a gradient operation of the intensity image. Hough transform techniques identify linear and circular edges in the intensity image, and two graph structures, one corresponding to the detected lines, another corresponding to the detected circles, are formed. The nodes of the line-based graph correspond to intersections of linear segments, whereas the nodes in the circle-based graph correspond to circle centers. Edges have attributes of length in 3D space. A suboptimal matching strategy is used to derive a measure of fit with theoretical model graphs of every object in the domain. The object domain consists of synthetic objects formed by polyhedra and generalized cones in one experiment, and assorted types of (real) wrenches in another experiment.

Characteristic views are promoted in [Cha82] as an efficient means of object recognition via edge configurations. Objects are polyhedra and the assumptions are made that these objects are in a stable position and that scene illumination is "as favorable as possible for the recognition task", apparently meaning that all edges should be detectable. Characteristic views consist of topologically equivalent 2D line configurations; these views can be partitioned into groups based on the numbers and types of line junctions that occur in the views, which reduces search time in detecting a match. Recognition proceeds by extracting the 2D line configuration from the input image and labeling the junction points. Initial matching is made by comparing the silhouette line structure with those for characteristic views; further matching proceeds by use of junction types. A junction-to-"characteristic view junction" transformation provides 3D coordinates of the junctions. Although they claim that nonpolyhedral objects could be recognized as well, they provide no examples, and it is not clear how the resulting edges should be processed.

Stockman and Esteva [Sto84] use inverse perspective computations which map pairs of points of a 2D image of an object to pairs of points on a 3D model of the object to determine a correct object pose. It is assumed that the object models are fixed and known, the object is at rest on a planar surface, and the face of the object in contact with the plane is known. Three pose parameters <r,x,y> (rotation angle, x translation, y translation) are the free variables to be found. The existence of a plausible <r,x,y> can be determined by appropriate camera matrix operations with a point-pair from the image of the object and a point-pair from the object model. For all pair-pair combinations which do possess an associated <r,x,y>, this triple is stored as a 3D pattern vector. When all possible pair-pairs have been investigated in this manner, these pattern vectors are clustered, and the largest cluster indicates the most likely

candidate for the correct <r,x,y> pose parameters and whether recognition is satisfactory.

6.2.3 -- Surface Representations

Another approach to 3D object modeling is based on the fact that an object surface viewed by the sensor may be segmented into regions which correspond to natural object faces. By suitably describing each region and relations between regions, it should be possible to unambiguously identify objects. Problems arise when certain views of an object provide too little information to make a definite identification. For example, looking at the aftershave bottle from below will show only the planar base surface. Although it would be ideal to work with general quadric functions, matters of manipulability and parameter stability under noise degradation make it necessary to work with simple surface primitives, such as planar, spherical, cylindrical, and conical surfaces.

Lu et al. [Lu85] use an attributed hypergraph representation to represent and recognize 3D objects in range images. The primitives derived in preliminary analysis consist of object surface patches: planes, cylindrical faces, and conical faces. The hypergraph representation at the most primitive level has elementary area attributed graphs to represent polygonal planar faces. Nodes represent linear boundary segments, and edges represent the intersections of these segments and have attributed angle values. These elementary area attributed graphs and other non-polygonal surface patches are considered as hypervertices of the hypergraph; there is a relation between adjacent faces which is a principle angle, the angle formed by surface normals for planes and axes of symmetry for non-planes. Hyperedges then consist of groups of hypervertices which form polygonal blocks, cylindrical and conical surfaced blocks.

Object recognition involves finding hypergraph monomorphisms (subgraph isomorphisms). Certain methods for accelerating recognition are used: sorting numerical attribute values, looking for special features of the objects, and using a distance measure between attributed graphs. Unfortunately, there is no allowance for error in the image analysis stage: one extraneous over-segmentation could easily confound the monomorphism search. It appears that the power of this procedure is derived from the polyhedral faces and polygonal blocks, which are a rich source of matchable information in the form of elementary area attributed graphs; no examples involving nonpolyhedral surfaces were presented.

Barrow and Popplestone [Bar71] use reflectance images with a dynamic range of 16 grey levels to identify surfaces in a scene. Preliminary surfaces are identified as regions of pixels whose grey levels span at most three grey values. These surfaces are merged if the average contrast over the common boundary is less than some threshold, and remaining small patches are removed. The picture is described in terms of properties and relations between the regions. Some properties are compactness and shape measures derived from the Fourier analysis of the region boundary. The relations are: "bigger than", "adjacent to", "distant from", "convex boundary", "above", and "below". The models are characteristic-view models. The area information is based only on pixel counts and thus topologically equivalent views of an object can have different characteristic-view models. Recognition is based on identifying the best match of a subset of the picture with a subset of the model regions (the hierarchical synthesis method described in more detail in Section 6.2.4).

6.2.4 -- The Matching Problem

Barrow et al. [Bar72] evaluate four methods for matching relational structures. The basic problem is finding a monomorphism (a homomorphism which is 1-1) from a given relational structure R which is an internal representation to a relational structure S derived from an image. These four methods are described in detail in the following paragraphs.

The TREE SEARCH technique generates a tree of partial mappings, where "branching out" from a node (partial mapping) involves adding one more correspondence to the partial mapping. Each node is evaluated through a merit function measuring how well the corresponding relational structures agree under the partial mapping. At each stage of tree construction, the node representing the most promising partial mapping is expanded. The resulting mapping, which may itself be a partial mapping, is optimal with respect to the merit function.

A RELATIONAL COMPOSITION SEARCH involves the creation of "intermediate" relations from the primitive relations used in the relational structures R and S. These can be constructed by compositions, intersections, and inverses, of the primitive relations. These new relations can be designed to possess fewer members than the primitive relations. This smaller relation size can reduce search time for a monomorphism. As an example, consider a drawing of a face complete with head, eyes, and mouth, and the three relations:

$F = \{\langle x,y\rangle : x \text{ is inside of } y\}$

$G = \{\langle x,y\rangle : x \text{ is to the left of } y\}$

$H = \{\langle x,y\rangle : x \text{ is above } y\}$.

Our relational structure for the face is:

$F=\{\langle \text{mouth,head}\rangle, \langle \text{lefteye,head}\rangle, \langle \text{righteye,head}\rangle\}$

$G=\{\langle \text{lefteye,righteye}\rangle\}$

$H=\{\langle \text{lefteye,mouth}\rangle, \langle \text{righteye,mouth}\rangle\}$.

The intermediate relation given by $F \cdot G \cdot H^{-1} \cap F \cap F \cdot G^{-1} \cdot H^{-1}$,

where "·" represents composition, has one member: <mouth,head>. If this relation were to have members in relational structure S, assuming primitives F, G, and H are used, these members would be good cues for locating faces in the image from which S was derived. If no such members exist, one can conclude that no complete faces are in the image.

CLASSIFICATION REFINEMENT requires the nodes of the relational graphs to be classifiable into two or more groups based on a unary relation, or property, of the nodes. This initial classification is then refined by considering the binary relations. To illustrate, suppose the initial classification partitions the nodes of each relational structure into A = "circular objects" and B = "non-circular objects", and consider binary relation F. Then A can be refined into A_1 and A_2, where

$$A_1 = \{ \pi_1[A \times A \cap F] \}$$
$$A_2 = \{ \pi_1[A \times B \cap F] \}$$

where π_1 is the projection function mapping an ordered pair onto its first element. Note that A_1 and A_2 may not be disjoint. As other relations are considered, some of the classes are, hopefully, singleton sets, whereby some unambiguous correspondences may be deduced. The refinement technique concludes with one or more consistent monomorphisms from R to S if any exist.

HIERARCHICAL SYNTHESIS decomposes the matching problem into many subproblems to reduce the overall computational time. A brute force search for a relational structure R with n elements within another structure S with N elements ($N >> n$) requires testing $O(N^n)$ combinations. However, if we decompose R into k substructures R_i of r_i elements each (so that $\Sigma r_i = n$), and perform a brute force search to find all occurrences of R_i in S, for all i, and test combinations of these occurrences to identify R, then the number of combinations to be tested will be

$$N^k + \sum_{i=1}^{k} N^{r_i} << \prod_{i=1}^{k} N^{r_i} = N^n.$$

We accept the conjecture that the number of occurrences of a substructure in S will be O(N) if this substructure has nontrivial complexity. In addition, some of the R_i's could in turn be decomposed into R_{ij}'s and so on to even lower levels to give a hierarchical decomposition of the original structure R.

These four techniques were applied to various relational structure matching problems in vision: a representative of CPU time required for execution of these procedures is as follows:

(1) Tree search: $\sim 10^{10}$ secs
(2) Relational composition: 65 secs
(3) Classification refinement: 408 secs
(4) Hierarchical synthesis: 22.5 secs

They conclude that hierarchical synthesis seems to be the most promising technique.

Shapiro and Haralick [Sha82] consider using clustering to prune the search tree for matching a simple graph representation of an observed object to a large database of model objects. A simple graph is the representation consisting of unlabeled nodes and edges, with no attributes on the nodes and edges. A distance metric for the dissimilarity between two graph representations is defined, and is used to construct a dissimilarity matrix on all model representations. These representations are then clustered using this dissimilarity matrix, and for each cluster an "average" or "representative" graph (the cluster center, roughly speaking) is found. The graph for an observed object can then be compared to each representative graph to obtain a best match among the cluster centers. Objects are then recognized by comparing the observed graph to each element

graph in the best-match cluster of model graphs. This two-level matching reduces the amount of computational time. It is very important that the distance measure be a metric, hence the use of simple graphs. Extensions of this technique to include more complex representations that involve attributes of nodes and edges require deriving a suitable distance metric for these representations.

6.3 -- Representation Derivation

From the segmentation, classification, and merging processes developed in Chapters 3, 4, and 5, we can obtain a segmentation of a range image into surface patches, with additional information consisting of the "sense" of the surface and the relationships between pairs of patches. We now derive a wealth of information from this segmentation. A representation consists of three classes of information:

(1) Morphological information which characterizes the object shape in 2D;

(2) Patch information which describes the 3D surface patches derived in Chapters 3 through 5;

(3) 2D boundary information which describes relationships between pairs of patches.

6.3.1 -- Initial Representation

The morphological information derived from a range image was defined in Section 2.7: we denote this information as PERIM, the perimeter of the silhouette image, BGCOMP, the number of connected background components in the range image, and CHCOMP, the number of connected background components within the convex hull of the silhouette image.

Patch information consists of the sense (planar, convex, concave) derived in Chapter 4, surface area, span, and linear fit of the boundary of the patch with the background pixels (-1 if no such boundary exists). The surface area (or size) attribute is an approximation of the true object surface area corresponding to the patch and a procedure for computing it was presented in Section 2.6.

The span of a patch, P, is defined to be

$$\max(\|\bar{p}_{r1,c1} - \bar{p}_{r2,c2}\|,\ (r1,c1)\epsilon P,\ (r2,c2)\epsilon P\ ),$$

where $\bar{p}_{r,c}$ is defined in Section 2.1. However, because computation of this value can be $O(|P|^2)$, where $|P|$ is the number of elements in patch P, we approximate the span by subsampling the patch every 4th row and every 4th column (i.e., we constrain r1,r2,c1,c2 to be multiples of 4). If no pixels of P are found in this subsampling, we conclude that P is very small and set the span to 0. The linear fit was defined in Section 5.5.

The next set of information involves the relationships between pairs of patches. We derive one (or two) vectors of information for each pair of patches, consisting of: boundary type (adjacent, jump, or remote), normal angle, minimum distance between patches, and maximum distance between patches, along with boundary angle and linear fit of pixels along the boundary (if the patches are adjacent) or jump gap (if the boundary is a jump edge). Given two patches in the segmentation, there are four possible conditions:

(a) They could be adjacent;

(b) They could be separated by a jump edge;

(c) Both (a) and (b) might hold; and

(d) They may be separated by other patches;

Cases (a), (b), and (d) are referred to as, respectively, adjacent, jump, and remote boundary types; case (c) is represented as two vectors of information. The normal angle

between a pair of patches is defined in Section 5.2; the linear fit for the case of adjacent patches is defined in Section 5.5. The boundary angle between two adjacent patches P_i and P_j is the average angle of intersection along the boundary, and is given by

$$\frac{\left(\sum_{\bar{p} \in P_i,\ \bar{q} \in P_j,\ \bar{p},\bar{q}\ \text{adjacent}} s(\bar{p},\bar{q}) \times \text{angle between } \bar{n}_p \text{ and } \bar{n}_q \right)}{\left(\sum_{\bar{p} \in P_i,\ \bar{q} \in P_j,\ \bar{p},\bar{q}\ \text{adjacent}} 1 \right)}$$

where $s(\bar{p},\bar{q})$ is the sign factor defined in Section 4.2.3. The minimum and maximum distances between two patches P and Q are defined by

$$\min \{ \| \bar{p}_{r1,c1} - \bar{p}_{r2,c2} \|,\ (r1,c1) \epsilon P,\ (r2,c2) \epsilon Q \}$$
$$\max \{ \| \bar{p}_{r1,c1} - \bar{p}_{r2,c2} \|,\ (r1,c1) \epsilon P,\ (r2,c2) \epsilon Q \}.$$

As with the span derivation for patches, we subsample every 4th row and 4th column to improve the execution time. The jump gap is defined to be the maximum z distance between pixels on either side of the jump edge forming the border of the two patches.

Combining all this information, we obtain an initial object representation R_0. We establish the following notation for patches P and Q,

PERIM	perimeter of the silhouette image;
BGCOMP	number of connected background components;
CHCOMP	number of connected background components within the convex hull of the silhouette image;
SENSE(P)	sense (planar, convex, concave) of P;
SIZE(P)	surface area of P;

SPAN(P)	span of P;
FIT1(P)	linear fit of the boundary of P with the background;
B_TYPE(P,Q)	neighborhood type (adjacent, jump, remote) of P and Q. This is not, strictly speaking, a function, since P and Q may be both adjacent and jump neighbors.
N_ANGLE(P,Q)	normal angle between P and Q;
MIN_DIST(P,Q)	minimum distance between P and Q;
MAX_DIST(P,Q)	maximum distance between P and Q;
B_ANGLE(P,Q)	boundary angle between P and Q (if appropriate);
FIT2(P,Q)	linear fit of the boundary between P and Q (if appropriate);
JUMPGAP(P,Q)	jump gap between P and Q (if appropriate);

6.3.2 -- Modified Representations

Although the 3D surface patches in R_0 are contained in natural object faces, often a natural object face will be broken into more than one surface patch. It would be useful to recover from this oversegmentation to obtain representations which better represent the object in the range image.

Knowledge about the n objects in our database is used to perform a merging of patches of R_0 to produce n new representations $(R_1,\ldots,R_n)$. For each object in the domain, there is a minimum expected boundary angle. For example, if object i is a cube then no boundary angles should be less than 90° and pairs of adjacent patches whose boundary angles are much less than 90° are probably part of the same face of the cube and should be merged. Noise prevents us from using the theoretical thresholds. Instead, we give plenty of

margin for error.

Given angle threshold t_i for object i, we construct a new representation R_i from $R_\emptyset$ by merging pairs of adjacent patches whose (absolute) boundary angle is less than t_i (irrespective of their sense). Table 6-1 shows the set of 10 threshold angles used for our object database containing 10

Table 6-1
Thresholds for Knowledge-based Merging

Object	Angle
Aftershave bottle	50°
Cup	50°
Block	30°
Tunnel	50°
Cobra sculpture	0°
Mushroom	50°
Plug	50°
Diesel	30°
Toy part	30°
Human hand	0°

objects. If a pair of patches happen to be related as both adjacent and jump neighbors, the merging process is not performed regardless of the boundary angle. This merging process requires that the representation R_0 be modified: suppose that patches P_1 and P_2 are to be merged to make patch P; let Q be a patch which is not P_1 or P_2.

(1) If SIZE(P_1)/SIZE(P_2)≥5 then SENSE(P)=SENSE(P_1); If SIZE(P_2)/SIZE(P_1)≥5 then SENSE(P)=SENSE(P_2); If neither of the above two conditions are satisfied, then SENSE(P) = "convex" if B_ANGLE(P_1,P_2)≥0 and "concave" otherwise.

(2) SIZE(P) = SIZE(P_1) + SIZE(P_2).

(3) SPAN(P) = max (SPAN(P_1), SPAN(P_2), MAX_DIST(P_1,P_2)).

(4) FIT1(P) = max (FIT1(P_1), FIT1(P_2)).

(5) N_ANGLE(P,Q) = (N_ANGLE(P_1,Q) + N_ANGLE(P_2,Q))/2.

(6) MIN_DIST(P,Q) = min (MIN_DIST(P_1,Q), MIN_DIST(P_2,Q))

(7) MAX_DIST(P,Q) = max (MAX_DIST(P_1,Q), MAX_DIST(P_2,Q))

(8) If B_TYPE(P_1,Q) = B_TYPE(P_2,Q) = "adjacent", then B_TYPE(P,Q)="adjacent", and B_ANGLE(P,Q) = (B_ANGLE(P_1,Q) + B_ANGLE(P_2,Q))/2. Otherwise, if one of B_TYPE(P_1,Q) or B_TYPE(P_2,Q) is "adjacent" (say P_1), then B_TYPE(P,Q)="adjacent" and B_ANGLE(P,Q)=B_ANGLE(P_1,Q).

(9) FIT2(P,Q) = max(FIT2(P_1,Q), FIT2(P_2,Q)).

(10) If B_TYPE(P_1,Q) = B_TYPE(P_2,Q) = "jump", then B_TYPE(P,Q)="jump", and JUMPGAP(P,Q) = max (JUMPGAP(P_1,Q), JUMPGAP(P_2,Q)). Otherwise, if one of B_TYPE(P_1,Q) or B_TYPE(P_2,Q) is "jump" (say P_1), then B_TYPE(P,Q)="jump" and JUMPGAP(P,Q)=JUMPGAP(P_1,Q).

(11) If B_TYPE(P_1,Q) = B_TYPE(P_2,Q) = "remote", then B_TYPE(P,Q)="remote".

Of course, the morphological information PERIM, BGCOMP, and CHCOMP are not affected.

At the end of this process, we have n representations of our observed scene, $(R_1,\ldots,R_n)$, which are modified versions of the initial representation R_0 under the hypotheses that the object in the scene is Object 1, ..., Object n, respectively. Table 6-2 shows the original representation obtained from the cup image segmentation in Figure B-8(c). Figure 6-1 shows the result of merging this cup representation with an angle threshold of 50°, and the corresponding revised representation is shown in Table 6-3. Figures 6-2 and 6-3 show the results of merging the cup segmentation in Figure B-7(c) and the Cobra sculpture segmentation in Figure B-15(c) with an angle theshold of 50°. Note that the oversegmentation of the cup passed to the recognition stage is corrected by the threshold, whereas the cobra sculpture segmentation loses information if patches are merged because the boundary angles between the concave patches at the sides of the cobra head and the planar patch in front of the head are essentially zero. The existence of smooth joins between patches for certain objects motivates us to adopt a 0° threshold for those objects.

6.4 -- Evidence-Based Recognition

Many techniques for recognizing objects map quantitative information to a model representation. This procedure typically involves graph-matching and has exponential time-complexity; however, considerable speed-up has been achieved by clever use of constraints, knowledge, and thresholds on maximum matching error [Tom84]. Another type of approach to object recognition involves reducing a representation to symbolic information, or evidence; a collection of evidence can be used to determine the likely contents of an observed scene [Coh85].

Table 6-2
Original Cup Representation

Morphological Information

PERIM	13.84
BGCOMP	1
CHCOMP	0

Patch Information

Color	Index	SENSE	SIZE	SPAN	FIT1
White	1	concave	1.285	1.478	0.79
Red	2	convex	6.766	3.577	1.12
Blue	3	planar	3.099	3.074	0.19
Yellow	4	planar	3.146	3.009	1.53
Brown	5	concave	1.749	1.669	0.04
Green	6	planar	1.643	2.246	0.11

Boundary Information

Patches		BTYPE	N_ANGLE	B_ANGLE	JUMPGAP	MINDIST	MAXDIST	FIT2
1	2	jump	38.490		3.207	1.780	4.204	-1
1	3	jump	91.045		2.040	1.109	3.846	-1
1	4	rem	18.662			2.672	4.123	-1
1	5	adj	-39.863	-24.812			2.262	2.97
1	6	rem	-79.107	0.000		1.061	3.202	-1
2	3	adj	52.831	20.130			3.782	3.63
2	4	adj	54.000	17.909			4.028	1.09
2	5	jump	9.939		3.435	2.602	4.332	-1
2	6	jump	43.210		2.971	0.583	4.278	-1
3	4	rem	106.579			1.815	4.139	-1
3	5	rem	51.791			2.147	4.251	-1
3	6	rem	16.688			2.577	4.302	-1
4	5	rem	57.342			1.946	4.049	-1
4	6	jump	96.523		2.167	0.000	3.674	-1
4	6	adj	96.523	96.523			3.674	0.26
5	6	adj	-39.297	-21.944			3.031	2.04

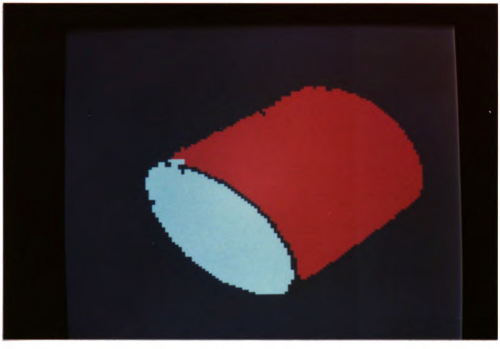


Figure 6-1
Knowledge-based Merging of Cup from Figure B-8(c)

Evidence-based reasoning has played an important role in the design of programs which make decisions under uncertainty, particularly expert systems in Artificial Intelligence. A technique which is very popular is called the parallel certainty inference technique, and is used by expert systems such as MYCIN [Sho75] and PROSPECTOR [Dud79b]. It has the same flavor as the Bayesian inference technique and the Dempster-Shafer theory [Sha76]. A decision is to be made from observations about which of n hypotheses $H_1,\ldots,H_n$ is true. The parallel certainty inference technique operates in two stages. First, a conclusion (typically a number corresponding to strength of belief, but it may also be symbolic [Coh85]) is derived about each H_i under the assumption that H_i is true. Second, these conclusions are compared to obtain a final decision. Evidence-based recognition also uses this technique. In fact, the knowledge-based merging we have already described is a part of the first stage of this technique.

Table 6-3
Revised Cup Representation

Morphological Information

PERIM	13.84
BGCOMP	1
CHCOMP	0

Patch Information

Color	Index	SENSE	SIZE	SPAN	FIT1
White	1	concave	4.677	3.031	0.79
Red	2	convex	13.011	4.028	1.54

Boundary Information

Patches		BTYPE	N_ANGLE	B_ANGLE	JUMPGAP	MINDIST	MAXDIST	FIT2
1	2	adj	96.520	79.450			4.332	0.26
1	2	jump	96.520		3.435	0.000	4.332	-1

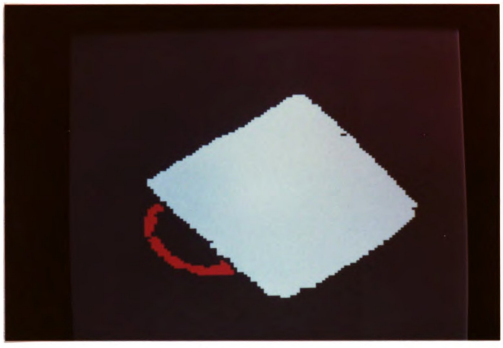


Figure 6-2
Knowledge-based Merging of Cup from Figure B-7(c)

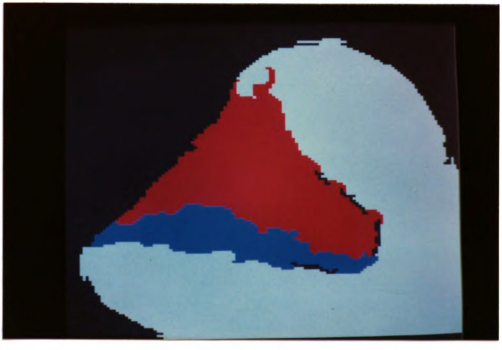


Figure 6-3
Knowledge-based Merging of Cobra from Figure B-15(c)

Consider that an object has a set of salient features, or set of evidence features, that help to identify the object. We develop the concept of an evidence space, in which objects in the model database are represented by vectors in this space. A similarity function to compare observed evidence to these model vectors is used for recognition.

Given n objects in the model database and m evidence features to be used for recognition, define an evidence feature rule base to consist of:

(1) a set of m evidence conditions $(\Xi_j)_{j=1}^{m}$, and
(2) an mxn evidence weight matrix $\mathbf{E}$, where $\mathbf{E}_{ji}$ represents the degree to which satisfaction of the evidence condition Ξ_j (i.e., observing the jth evidence feature) supports the hypothesis that the observed object is object i (denote this hypothesis by H_i).

The value $\mathbf{E}_{ji}$ is a number in the range [-1,1]; although any number in this range is theoretically fair game, we restrict our particular system to five possible values, to reduce subjectivity in setting the values:

$$\mathbf{E}_{ji} = \begin{cases} 1.0 & \text{if condition } \Xi_j \text{ strongly substantiates } H_i; \\ 0.5 & \text{if condition } \Xi_j \text{ tends to support } H_i; \\ 0.0 & \text{if condition } \Xi_j \text{ provides no information about } H_i \\ -0.5 & \text{if condition } \Xi_j \text{ tends to refute } H_i; \\ -1.0 & \text{if condition } \Xi_j \text{ strongly refutes } H_i. \end{cases}$$

Denote the jth evidence rule as $\langle \Xi_j, \varepsilon_j \rangle$ where ε_j is the jth row of $\mathbf{E}$. For each object i define an evidence m-vector ξ_i which corresponds to the ith column of $\mathbf{E}$, and is defined such that $\xi_{ij} = \mathbf{E}_{ji}$.

Given an observed representation R, we derive an instance m-vector ι^R:

$$\iota^R_j = \begin{cases} 1.0 & \text{if evidence condition } \Xi_j \text{ is satisfied;} \\ 0.0 & \text{otherwise.} \end{cases}$$

6.4.1 -- Evidence-Based Similarity Measure

We approach the task of recognition by deriving a similarity between an evidence vector ξ_i and the instance vector ι^R. We would like the "1" entries of ι^R to correspond to positive (or zero) entries in ξ_i, and negative entries in ξ_i to correspond to "0" entries in ι^R when object i is in a range image. A correlation measure between ι^R and ξ_i may appear to be a useful similarity measure for this purpose. The difficulty with applying a standard rank correlation measure such as Spearman's rank correlation coefficient [Con80] is that the measure interprets positive and negative elements of ξ_i the same, whereas their comparative effects on a similarity measure for evidence-based decision making should be different. For example, suppose there were 6 evidence features and 2 objects, and:

$$\iota^R = (\ 1.0,\ \ 1.0,\ \ 1.0,\ \ 0.0,\ \ 0.0,\ \ 0.0)$$
$$\xi_1 = (\ 1.0,\ \ 0.5,\ -0.5,\ -1.0,\ -1.0,\ -1.0)$$
$$\xi_2 = (\ 1.0,\ \ 0.5,\ \ 0.0,\ \ 0.0,\ -0.5,\ -0.5)$$

The Spearman rank correlation coefficient applied to ι^R and ξ_1 gives a value of 0.93, and when applied to ι^R and ξ_2 gives a value of 0.81. Notice that all positive evidence features for each object were observed; however, negative evidence was observed for object 1, but none was observed for object 2, and yet the correlation returns a larger similarity to object 1. This comes from not observing the strongly refuting evidence in ξ_1.

We now define a similarity measure which does behave as we would expect given the nuances belonging to "positive" and "negative" evidence. We also present a number of important properties of our measure.

Construct two m-vectors from ξ_i and ι^R: Φ_i, called the observed evidence vector, and ξ_i^+, called the positive evidence vector, where

$$\phi_{ij} = \iota_j^R \xi_{ij} \text{ , and}$$
$$\xi_{ij}^+ = \max(\xi_{ij}, 0) \text{ .}$$

The vector Φ_i can be written as $(\Phi_i^+ + \Phi_i^-)$, where

$$\phi_{ij}^+ = \iota_j^R \max(\xi_{ij}, 0)$$

indicates observed positive evidence for object i, and

$$\phi_{ij}^- = \iota_j^R \min(\xi_{ij}, 0)$$

indicates observed negative evidence for object i.

Define the similarity $\tau_i^m(R)$ of the observed object representation R (having instance vector ι^R) to model object i based on m evidence features to be:

$$\tau_i^m(R) = \frac{\|\Phi_i^+\|^2}{\|\Phi_i\| \ \|\xi_i^+\|} - \frac{\|\Phi_i^-\|^2}{\|\Phi_i\|^2} \text{ if } \|\Phi_i\| \neq 0, \text{ and } 0 \text{ otherwise.}$$

We will write τ_i^m and ι instead of $\tau_i^m(R)$ and ι^R when R is understood.

Theorem 1: $-1 \leq \tau_i^m \leq 1$.

A value of τ_i^m close to 1 means that there is a good match

between the observed evidence features and the positive evidence features corresponding to object i, whereas a value close to -1 means that there is a good match between the observed evidence features and the negative evidence features corresponding to object i. Note that $\tau_i^m=1$ when all positive evidence features and no negative evidence features for object i occur in the observed representation. Similarly, $\tau_i^m=-1$ when all observed evidence is nonpositive for object i. Hence we can design an object recognition system which is based on detecting large values of τ_i^m.

Proof of Theorem 1:

$$\tau_i^m \leq \frac{\|\phi_i^+\|^2}{\|\phi_i\| \, \|\xi_i^+\|} \leq 1, \text{ since}$$

$$\|\phi_i^+\|^2 = \sum_{j=1}^{m} (\phi_{ij}^+)^2$$

$$= \sum_{j=1}^{m} (\iota_j^R)^2 (\max(\xi_{ij}, 0))^2$$

$$= \sum_{j=1}^{m} (\iota_j^R) \xi_{ij} (\max(\xi_{ij}, 0))^2$$

$$= \sum_{j=1}^{m} \phi_{ij} \xi_{ij}^+$$

$$\leq \sqrt{\sum_{j=1}^{m} \phi_{ij}} \sqrt{\sum_{j=1}^{m} \xi_{ij}^+} \qquad \text{(Schwarz Inequality)}$$

$$= \|\phi_i\| \|\xi_i^+\|.$$

Also,

$$\tau_i^m \geq \frac{-\|\varphi_i^-\|^2}{\|\varphi_i\|^2}$$

$$= \frac{-\|\varphi_i^-\|^2}{\|\varphi_i^+ + \varphi_i^-\|^2}$$

$$\geq \frac{-\|\varphi_i^+ + \varphi_i^-\|^2}{\|\varphi_i^+ + \varphi_i^-\|^2} = -1. \qquad \text{QED.}$$

6.4.2 -- Properties of the Similarity Measure

Our recognition system can be expanded to accommodate more objects in the database: each added object will require the development of a number of new evidence features. It is useful to know how the behavior of the system will change under an added evidence feature.

Theorem 2: If evidence rule (m+1) is added to the system, then the new similarity τ_i^{m+1} is related to τ_i^m as follows:

(a) If $\xi_{i(m+1)}=0$, then $\tau_i^{m+1} = \tau_i^m$.

(b) If $\xi_{i(m+1)}<0$, then $\tau_i^{m+1} \leq \tau_i^m$ if $\iota_{(m+1)}=1$;
Otherwise, $\tau_i^{m+1} = \tau_i^m$.

(c) If $\xi_{i(m+1)}>0$, then $\tau_i^{m+1} \geq \tau_i^m$ if $\iota_{(m+1)}=1$;
Otherwise, $\tau_i^{m+1} < \tau_i^m$.

This theorem establishes some of the desirable properties expected of an evidence-based decision system. If new positive evidence for the hypothesis H_i is observed, then the strength of belief in H_i should increase (i.e., $\tau_i^{m+1} \geq \tau_i^m$), and if new negative evidence for H_i is observed, the strength of belief in H_i should decrease (i.e., $\tau_i^{m+1} \leq \tau_i^m$). The property that $\tau_i^{m+1} \leq \tau_i^m$ when a new positive

evidence feature for H_i is added but is not observed is not as intuitively obvious. What this property roughly means is that if a larger proportion of object i's positive evidence features are satisfied by representation R than that of object j, belief in H_i should be stronger than belief in H_j.

Lemma 1: If $B \geq A$ and $C \geq A$, then the function f defined by

$$f(x) = \frac{A+x}{\sqrt{B+x}\ \sqrt{C+x}}$$

is a nondecreasing function of x, for $x \geq 0$.

Proof:

$$\frac{d}{dx}\left(\frac{A+x}{\sqrt{B+x}\ \sqrt{C+x}}\right) = (B+x)^{-1/2}(C+x)^{-1/2} - (1/2)(A+x)(B+x)^{-3/2}(C+x)^{-1/2} - (1/2)(A+x)(B+x)^{-1/2}(C+x)^{-3/2}$$

$$\geq (B+x)^{-1/2}(C+x)^{-1/2} - (1/2)(B+x)(B+x)^{-3/2}(C+x)^{-1/2} - (1/2)(C+x)(B+x)^{-1/2}(C+x)^{-3/2}$$

$$= 0 \quad \text{for } x \geq 0.$$

Proof of Theorem 2:

$$\tau_i^{m+1} = \frac{\|\Phi_i^+\|^2}{\|\Phi_i\|\ \|\xi_i^+\|} - \frac{\|\Phi_i^-\|^2}{\|\Phi_i\|^2}.$$

We can write

$$\|\Phi_i^+\|^2 = \sum_{k=1}^{m} (\phi_{ik}^+)^2 + (\iota_{(m+1)}\max(\xi_{i(m+1)}, 0))^2$$

$$= a + \delta a,$$

$$\|\Phi_i\|^2 = \sum_{k=1}^{m} (\phi_{ik})^2 + (\iota_{(m+1)}\xi_{i(m+1)})^2$$

$$= b + \delta b,$$

$$\|\xi_i^+\|^2 = \sum_{k=1}^{m} (\xi_{ik}^+)^2 + (\max(\xi_{i(m+1)},0))^2$$

$$= \quad c \quad + \quad \delta c \ , \text{ and}$$

$$\|\phi_i^-\|^2 = \sum_{k=1}^{m} (\phi_{ik}^-)^2 + (\iota_{(m+1)}\min(\xi_{i(m+1)},0))^2$$

$$= \quad d \quad + \quad \delta d \ ,$$

so that

$$\tau_i^{m+1} = \frac{a + \delta a}{\sqrt{b + \delta b}\,\sqrt{c + \delta c}} - \frac{d + \delta d}{b + \delta b} \ .$$

Note that a, b, c, d, δa, δb, δc, and δd are all nonnegative quantities, and that

$$\tau_i^m = \frac{a}{\sqrt{b}\,\sqrt{c}} - \frac{d}{b} \ ;$$

the values δa, δb, δc, and δd correspond to effects of the (m+1)st evidence rule.

(a): If $\xi_{i(m+1)}=0$ then $\delta a=\delta b=\delta c=\delta d=0$, so that

$$\tau_i^{m+1} = \frac{a}{\sqrt{b}\,\sqrt{c}} - \frac{d}{b} = \tau_i^m .$$

(b): If $\xi_{i(m+1)}<0$ and $\iota_{(m+1)}=1$, then

$\delta a=\delta c=0$, $\delta b>0$, and $\delta d=\delta b$.

Note that $d = \sum_{k=1}^{m} (\phi_{ik}^-)^2 \leq \sum_{k=1}^{m} (\phi_{ik})^2 = b$.

Hence $$\frac{d}{b} \leq \frac{d + \delta b}{b + \delta b} = \frac{d + \delta d}{b + \delta b} .$$

Therefore,

$$\tau_i^{m+1} = \frac{a}{\sqrt{b+\delta b}\,\sqrt{c}} - \frac{d+\delta d}{b+\delta b} \leq \frac{a}{\sqrt{b}\,\sqrt{c}} - \frac{d}{b} = \tau_i^m.$$

If $\iota_{(m+1)}=0$, then $\delta a=\delta b=\delta c=\delta d=0$, and as in (a) we have $\tau_i^{m+1}=\tau_i^m$.

(c) If $\xi_{i(m+1)}>0$, and $\iota_{(m+1)}=1$, then $\delta a=\delta b=\delta c=\xi^2_{i(m+1)}$, and $\delta d=0$. Furthermore, we know that $b\geq a$ and $c\geq a$, since

$$b = \sum_{j=1}^{m} (\phi_{ij})^2 \geq \sum_{j=1}^{m} (\phi^+_{ij})^2 = a;$$

$$c = \sum_{j=1}^{m} (\xi^+_{im})^2 = \sum_{j=1}^{m} \max(\xi_{ij},0)^2 \geq \sum_{j=1}^{m} \iota_j (\max(\xi_{ij},0))^2$$

$$= \sum_{j=1}^{m} (\iota_j \max(\xi_{ij},0))^2 = \sum_{j=1}^{m} (\phi^+_{ij})^2 = a.$$

Therefore by Lemma 1 we know that

$$\frac{a}{\sqrt{b}\,\sqrt{c}} \leq \frac{a+\delta a}{\sqrt{b+\delta b}\,\sqrt{c+\delta c}};$$

also,

$$\frac{d+\delta d}{b+\delta b} = \frac{d}{b+\delta b} \leq \frac{d}{b}.$$

Hence

$$\tau_i^{m+1} = \frac{a + \delta a}{\sqrt{b + \delta b}\,\sqrt{c + \delta c}} - \frac{d + \delta d}{b + \delta b}$$

$$\geq \frac{a}{\sqrt{b}\,\sqrt{c}} - \frac{d}{b} = \tau_i^m.$$

If $\iota_{(m+1)}=0$, then $\delta a=\delta b=\delta d=0$ and $\delta c>0$ so that

$$\tau_i^{m+1} = \frac{a}{\sqrt{b}\,\sqrt{c+\delta c}} - \frac{d}{b} < \frac{a}{\sqrt{b}\,\sqrt{c}} - \frac{d}{b} = \tau_i^m. \qquad \text{QED.}$$

The behavior of τ_i^{m+1} as a function of $\xi_{i(m+1)}$ is established in Corollary 1.

Corollary 1: Given the conditions of Theorem 3, then:

(a) If $\xi_{i(m+1)}<0$ and $\iota_{(m+1)}=1$, then τ_i^{m+1} is a decreasing function of $\xi^2_{i(m+1)}$.

(b) If $\xi_{i(m+1)}>0$ and $\iota_{(m+1)}=1$, then τ_i^{m+1} is an increasing function of $\xi^2_{i(m+1)}$.

(c) If $\xi_{i(m+1)}>0$ and $\iota_{(m+1)}=0$, then τ_i^{m+1} is a decreasing function of $\xi^2_{i(m+1)}$.

This corollary simply asserts that belief in H_i will increase monotonically as a function of the weight assigned to new observed positive evidence, will decrease monotonically as a function of the (absolute) weight assigned to new observed negative evidence, and will decrease monotonically as a function of the weight assigned to new unobserved positive evidence.

Proof of Corollary 1:

We use the same notation used in the proof of Theorem 2. Let PT2 be shorthand for "the proof of Theorem 2".

(a) If $\xi_{i(m+1)}<0$ and $\iota_{(m+1)}=1$ then from PT2(b) we have

$$\tau_i^{m+1} = \frac{a}{\sqrt{b+\delta b}\,\sqrt{c}} - \frac{d+\delta d}{b+\delta b}$$

which is a decreasing function of $\xi^2_{i(m+1)}(=\delta b=\delta d)$.

(b) If $\xi_{i(m+1)}>0$ and $\iota_{(m+1)}=1$ then from PT2(c) we have

$$\tau_i^{m+1} = \frac{a+\delta a}{\sqrt{b+\delta b}\,\sqrt{c+\delta c}} - \frac{d}{b+\delta b}$$

which is an increasing function of $\xi^2_{i(m+1)}(=\delta a=\delta b=\delta c)$, by Lemma 1.

(c) If $\xi_{i(m+1)}>0$ and $\iota_{(m+1)}=0$ then from PT2(c) we have

$$\tau_i^{m+1} = \frac{a}{\sqrt{b}\,\sqrt{c+\delta c}} - \frac{d}{b}$$

which is a decreasing function of $\xi^2_{i(m+1)}(=\delta c)$. QED.

It would also be useful to estimate the change in the similarity measure under new evidence. This estimate would also apply to the situation in which noise or faulty segmentation causes one evidence feature to be missed or incorrectly observed, and also indicates the stability of τ under changes in the degrees of evidence values.

Theorem 3: Assume the conditions of Theorem 2. If evidence feature (m+1) is added and $\iota_{(m+1)}=1$, then an upper bound on the difference between τ_i^m and τ_i^{m+1} is given by:

$$|\tau_i^{m+1} - \tau_i^m| \leq \frac{1}{\sqrt{b+1}\ \sqrt{c}} + \frac{1}{b+1}$$

where b and c are as defined in Theorem 2:

$$b = \sum_{j=1}^{m} (\phi_{ij})^2 \quad \text{and} \quad c = \sum_{j=1}^{m} (\xi_{ij}^{+})^2.$$

If $\iota_{(m+1)}=0$ and $\xi_{i(m+1)}>0$, then an upper bound on the difference between τ_i^m and τ_i^{m+1} is given by:

$$|\tau_i^{m+1} - \tau_i^m| \leq 1 - \frac{\sqrt{c}}{\sqrt{c+1}}$$

Note that b increases as the number of observed evidence features with nonzero degree of evidence values for object i increases, and c increases as the number of positive evidence features for object i increases. Thus the sizes of fluctuations in the measure of similarity resulting from adding evidence features to the evidence rule base will tend to decrease as the number of observed evidence features with nonzero degree of evidence values for object i increases, and as the number of positive evidence features for object i increases.

Proof of Theorem 3:

We use the same notation a, b, c, d, δa, δb, δc, δd as in Theorem 2.

Then

$$\tau_i^m = \frac{a}{\sqrt{b}\sqrt{c}} - \frac{d}{b};$$

Suppose $\iota_{(m+1)}=1$:
If $\xi_{i(m+1)}>0$, then by corollary 1, the maximum change from τ_i^m to τ_i^{m+1} will occur for $\xi_{i(m+1)}=1$. In this case we have $\delta a=\delta b=\delta c=1$ and $\delta d=0$. Hence the maximum change occurs for $\tau_i^{m+1}= T^+$, where

$$T^+ = \frac{a+1}{\sqrt{b+1}\sqrt{c+1}} - \frac{d}{b+1}$$

$$< \frac{a+1}{\sqrt{b+1}\sqrt{c}} - \frac{d}{b+1},$$

If $\xi_{i(m+1)}<0$, then by corollary 1, the maximum change from τ_i^m to τ_i^{m+1} will occur for $\xi_{i(m+1)}=-1$. In this case we have $\delta a=\delta c=0$ and $\delta b=\delta d=1$, so that the maximum change occurs for $\tau_i^{m+1}= T^-$, where

$$T^- = \frac{a}{\sqrt{b+1}\sqrt{c}} - \frac{d+1}{b+1} .$$

From Theorem 2 we know that $T^- \leq \tau_i^m \leq T^+$.
Hence

$$|\tau_i^{m+1} - \tau_i^m| \leq T^+ - T^- < \frac{1}{\sqrt{b+1}\sqrt{c}} + \frac{1}{b+1}.$$

Suppose $\iota_{(m+1)}=0$ and $\xi_{i(m+1)}>0$:
By corollary 1 the maximum difference between τ_i^{m+1} and τ_i^m occurs for $\xi_{i(m+1)}=1$.
If $\xi_{i(m+1)}=1$, then $\delta a=\delta b=\delta c=1$ and $\delta d=0$.
Also, from the proof of Theorem 2 we know that $b \geq a$ and $c \geq a$.

Thus

$$\tau_i^m - \tau_i^{m+1} = \left(\frac{a}{\sqrt{b}\sqrt{c}} - \frac{d}{b}\right) - \left(\frac{a}{\sqrt{b}\sqrt{c+1}} - \frac{d}{b}\right)$$

$$= \frac{a}{\sqrt{b}}\left(\frac{1}{\sqrt{c}} - \frac{1}{\sqrt{c+1}}\right)$$

$$< \frac{\sqrt{a}}{\sqrt{c}} - \frac{\sqrt{a}}{\sqrt{c+1}}$$

$$< 1 - \frac{\sqrt{c}}{\sqrt{c+1}} . \qquad \text{QED.}$$

6.4.3 -- Recognition Technique

The final issue to be discussed is the actual procedure for making a decision about what object is observed. We have imposed a few restrictions on the assignment of evidence weights $\mathbf{E}_{ji}$ to allow the option of rejecting a representation when the observations do not strongly support any hypothesis. We require that for evidence feature (rule) j no more than one $\mathbf{E}_{ji}$ is equal to 1.0; when $\mathbf{E}_{ji}$=1.0, we call evidence feature j major evidence for object i -- the condition Ξ_j is ideally a very specific condition which constitutes strong supporting evidence only for object i. Furthermore, we require that for all objects i there exist an evidence condition Ξ_j such that $\mathbf{E}_{ji}$=1.0. That is, every object has at least one major evidence feature. It is difficult to find a specific condition to serve as major evidence for some objects, so a more general condition will have to be used.

A set of n representations were obtained in Section 6.3, such that the ith representation R_i is generated from the original representation $R_\emptyset$ under the hypothesis that object i is present in the scene. For each i, calculate $\tau_i^m(R_i)$ as follows:

(1) Derive the instance vector ι^{R_i} by determining the presence or absence of condition Ξ_j in R_i for each j=1,...,m.

(2) Derive $\tau_i^m(R_i)$ from ι^{R_i} and ξ_i.

Next, define $\hat{i}$ to be that value of i such that

$$\tau_{\hat{i}}^m = \max(\tau_i^m: i=1,..,n)$$

Conclude that object $\hat{i}$ occurs in the scene if and only if:

(1) $\hat{i}$ is unique; and

(2) $|(j:$ condition Ξ_j occurs in $R_{\hat{i}}$, $\xi_{j\hat{i}} = 1.0)| \geq 1$.

Condition (2) requires that at least one instance of major evidence for object $\hat{i}$ be observed. If condition (2) is not satisfied, then reject the object; that is, decide that it does not belong to the database.

We note here the time complexity of this object recognition technique. Let m be the number of evidence features, n the number of objects in the database, and N the number of patches in an initial representation. The two main stages of recognition are 1) deriving representations R_1, ..., R_n, and 2) deriving similarities $\tau_1^m(R_1)$, ..., $\tau_n^m(R_n)$. The derivation of each one of the n representations require as many as O(N) merges, where each merge requires the inspection of $O(N^2)$ patch relations, thus giving this stage a time complexity of $O(nN^3)$. The derivation of similarities involves n representations, each of which must be tested for the absence or presence of m evidence features, which in turn

require the inspection of (worst case) $O(N^2)$ patch relations, thus giving a time complexity of $O(nmN^2)$. Therefore, the time complexity of our recognition technique is $O(max(m,N)nN^2)$. In practice, we typically find $m>>N$. The value of m is 31 for the evidence rule base developed for our 10 object database. Over the 31 range images, N takes a minimum of 4 and a maximum of 25; all but 3 of the images have $N<13$.

6.5 -- Evidence Features and Results

So far, no mention has been made about the specific objects in our database. The specific objects and format of evidence is independent of the general design of the evidence system presented in Section 6.4. The specific format of evidence features based on the representations derived in Section 6.3 will now be introduced. Define three types of evidence features: 0th level evidence dealing with morphological features, 1st level evidence dealing with properties of patches, and 2nd level evidence dealing with pairs of patches.

A 0th level evidence condition is composed of any combination of morphological features defined in Section 2.7:

(1) PERIM bounds (a real interval);
(2) BGCOMP bounds (an integer interval); and
(3) CHCOMP bounds (an integer interval).

The condition will be satisfied if the morphological features observed for the range image satisfy the specified bounds. For example,

$$\begin{cases} \text{PERIM} \in (20'', \infty); \\ \text{BGCOMP} \in (1,1); \\ \text{CHCOMP} \in (0,0) \end{cases}$$

specifies the condition that the silhouette image has perimeter at least 20", and has no holes or indentations.

A 1st level evidence condition for patch P is composed of any combination of the following components defined in Section 6.3:

(1) SENSE(P) specification (planar, convex, concave);
(2) SIZE(P) bounds (a real interval);
(3) SPAN(P) bounds (a real interval);
(4) FIT1(P) bounds (a real interval); and
(5) OCCUR bounds (an integer interval).

The occurrence bounds specify the possible number of distinct patches P that must satisfy conditions described in (1), (2), and (3). An example of a 1st level evidence condition is:

$$\begin{cases} \text{SENSE(P)} = \text{planar}; \\ \text{SIZE(P)} \in (4.0, 5.5); \\ \text{SPAN(P)} \in (0.0, \infty); \\ \text{FIT1(P)} \in (-1.0, \infty); \\ \text{OCCUR} \in (1,2). \end{cases}$$

In order to be satisfied, this evidence condition requires that the number of patches in the representation that are planar with size in the interval (4.0,5.5) be either 1 or 2.

The 2nd level evidence conditions are considerably more detailed than 0th or 1st level evidence conditions: corresponding components are:

(a) Patch P conditions;
(b) Relationship(P,Q) conditions;
(c) Patch Q conditions; and
(d) Occurrence bounds,

where (a) and (c) consist of any combination of SENSE, SIZE, SPAN, and FIT1 specifications as used in 1st level evidence conditions, (b) consists of any combination of the following properties defined in Section 6.3:

(1) B_TYPE(P,Q) specification (adjacent, jump, remote);
(2) N_ANGLE(P,Q) bounds (a real interval);
(3) MINDIST(P,Q) bounds (a real interval);
(4) MAXDIST(P,Q) bounds (a real interval);
(5) B_ANGLE(P,Q) bounds (a real interval); and
(6) JUMPGAP(P,Q) bounds (a real interval),

and (d) is the integer bound on the number of occurrences of distinct pairs of patches (P,Q) that satisfy specifications (a), (b), and (d).

Appendix D gives the set of evidence features and corresponding evidence weights for the recognition of 10 objects. The evidence rules are stated verbally, with a header which specifies if the rule is major evidence for some object or is simply a general rule. For example, rule 22 is a major evidence feature for the plug object. This rule is based on the existence of two small planar patches at the ends of the cylinders sticking out of the main body of the plug, which are both parallel to a planar face of the main body. With the added distance and area information, this rule is specifically tailored to the plug object and is a good example of a major evidence rule. On the other hand,

rule 1 is a good example of a general rule, since the condition that the perimeter is greater than 20" is not a very specific condition and is satisfied by both the hand and the toy part objects. Evidence feature conditions are based on bounding intervals which specify the acceptable range of values of a given feature, such as surface area or boundary angle, which one might expect to observe under the imprecise conditions caused by image degradation.

In creating the evidence conditions, the bounds on angles are set to be (true angle - 10°, true angle + 10°) when the true angle > 90°, decreasing this range to about ±5° for angles closer to 0°. Distance ranges were tailored to the amount of uncertainty in the evidence feature being considered. For example, the distance of the jump gap when looking over the rim of the cup to its inside surface will vary depending on the angle at which the cup is held, and therefore the JUMP_GAP range is 1". On the other hand, given two planar faces which are parallel such as encountered on the block, the MIN_DIST between these faces could be specified with a smaller interval about the expected distance (e.g., evidence condition 14 in Appendix D).

Under the expectation that a surface area is unlikely to be grossly overestimated but that the surface could be occluded, lower bounds for surface areas are generally set much smaller than the true area of the object face, and upper bounds for surface areas are close to the true surface area. Also, the bounds on features for general evidence conditions are usually much larger than those for major evidence conditions which are more specifically tailored to a single object.

6.5.1 -- Results on the Range Image Database

For each of the 31 representations derived from the range images in our database (Figures B-1(a) to B-31(a)), a decision is made about what object is present in the image. Table 6-4 shows the similarities obtained from these segmentations. Appendix E provides the results of applying the recognition scheme to 31 range images. These results are presented as follows:

(a) The vector of similarities $r_i^m(R_i)$, given in the order (AS,HC,GB,TN,CB,MH,PL,DS,TY,HN);

(b) Those objects for which representation R_i satisfied at least some major evidence condition for object i;

(c) The final decision about the object, if the object was not rejected;

(d) If (d) specified recognition of object $\hat{i}$, then the evidence features satisfied by $R_{\hat{i}}$ are listed.

Note that bottle image AS4 was rejected; otherwise, all other images were recognized correctly. The difficulty with the AS4 classification occurred because the side of the bottle was classified as concave rather than planar.

This technique has also been applied to objects which are not in our database to test the reject option. Table 6-5 shows the set of similarities obtained from the four alien objects pictured in Figure 6-4. These objects were all rejected. Note that the maximum similarity value for object (b) was 0.48; however, the maximum similarity for PL3 was 0.50. This indicates that forming a reject option by thresholding the maximum similarity value would probably not be very successful.

Table 6-4

Recognition of Database Range Images

	AS	HC	GB	TN	CB	MH	PL	DS	TY	HN
AS1	0.91	-0.34	-0.36	0.38	-0.26	-0.65	0.29	-0.33	0.28	-0.30
AS2	0.82	-1.00	0.20	0.27	-1.00	-1.00	0.20	0.24	0.00	-1.00
AS3	0.82	-1.00	0.20	0.27	-1.00	-1.00	0.20	0.24	0.00	-0.30
AS4	0.41	0.23	-1.00	0.27	0.33	0.33	0.20	-1.00	0.28	0.28
HC1	-0.75	0.56	-0.75	-0.80	-0.26	-0.78	-0.50	-0.73	-0.44	0.28
HC2	-0.75	0.56	-1.00	-0.45	-0.26	-0.78	-0.82	-1.00	-0.44	0.40
HC3	-1.00	0.51	-1.00	0.00	0.00	-1.00	-1.00	-1.00	-0.68	0.00
HC4	-0.75	0.56	-1.00	-0.45	0.33	-0.78	-0.82	-1.00	-0.44	0.28
GB1	0.08	-0.34	0.63	0.38	-1.00	-1.00	-0.71	-0.47	0.39	-1.00
GB2	-0.05	-0.74	0.60	-0.27	-1.00	-0.83	-0.34	-0.69	0.28	-1.00
GB3	0.08	-0.34	0.63	-0.68	-0.70	-1.00	0.29	0.34	-0.30	-1.00
TN1	-0.43	-0.70	-0.36	0.65	-0.47	0.05	-0.10	-0.73	0.39	0.28
TN2	-1.00	0.23	-0.71	0.65	-1.00	-0.65	-0.71	-0.69	0.39	-1.00
TN3	-0.21	-0.70	-1.00	0.60	-0.26	0.47	0.29	-1.00	0.28	0.28
CB1	-0.67	-0.07	-0.01	-0.02	0.70	-0.78	-0.75	-1.00	0.48	-0.01
CB2	0.41	0.23	-0.84	0.27	0.70	0.33	0.20	-0.82	-0.44	-0.01
CB3	0.41	0.23	-1.00	0.27	0.51	0.33	0.20	-1.00	0.28	0.28
MH1	-0.67	-0.34	-1.00	-0.31	-0.26	0.82	-0.10	-1.00	0.28	0.28
MH2	-0.67	-0.34	-1.00	-0.31	-0.26	0.82	-0.10	-1.00	0.28	0.28
MH3	-1.00	-1.00	-0.36	-1.00	-1.00	0.75	-0.36	0.24	0.28	0.00
PL1	-0.05	-0.74	-0.88	-0.63	-0.70	-0.78	0.65	-0.86	-0.68	-0.01
PL2	-0.01	-1.00	0.28	0.11	-1.00	-0.39	0.54	-0.47	0.28	-1.00
PL3	-0.62	-0.70	-0.56	0.27	-0.79	-0.39	0.50	-0.86	0.13	-0.44
DS1	-0.67	0.32	0.16	-0.45	-0.81	-0.70	-0.10	0.64	0.48	-0.30
DS2	-1.00	-1.00	0.28	-1.00	-1.00	-1.00	-1.00	0.69	0.39	-1.00
DS3	-1.00	0.23	0.28	-1.00	-1.00	-1.00	0.20	0.54	0.00	-1.00
TY1	-0.83	-0.83	-0.58	-0.09	-0.46	-0.87	-0.88	-0.54	0.83	0.48
TY2	-0.70	-0.74	-0.61	-0.34	-0.46	-0.62	-0.61	-0.58	0.78	0.48
HN1	-0.82	-0.70	-0.90	-0.73	-0.46	-0.88	-0.84	-0.89	0.23	0.92
HN2	-0.82	-0.27	-0.85	-0.51	-0.30	-0.68	-0.56	-0.87	-0.11	0.68
HN3	-1.00	-0.34	-0.84	-0.31	0.33	-1.00	-1.00	-0.82	0.48	0.62

To determine the robustness of this procedure to different segmentations, we repeated the experiments by taking 2 fewer clusters than indicated by the S_{ave} values (Section 3.3.4). Table 6-6 shows the sets of similarities obtained from these segmentations. Here we note that the prevalent problem is the number of rejections. Note that the

Table 6-5
Similarities to Alien Objects

	AS	HC	GB	TN	CB	MH	PL	DS	TY	HN
(a)	0.03	-0.34	-0.70	0.38	-1.00	-1.00	-0.72	-0.69	0.28	-0.30
(b)	-0.36	0.32	-0.74	0.46	-0.46	-0.41	-0.52	-0.73	0.48	-0.01
(c)	-0.36	0.32	-0.08	-0.45	-0.25	-0.41	0.32	-0.05	0.39	0.28
(d)	-1.00	0.23	0.22	-1.00	0.35	-1.00	0.19	0.24	0.00	0.00

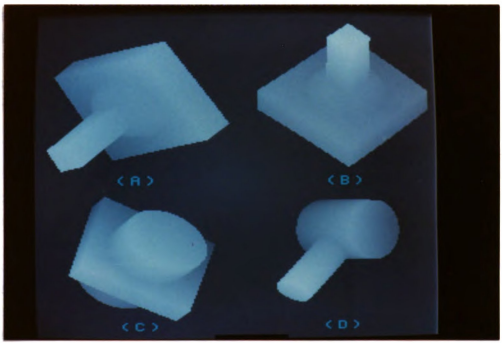


Figure 6-4
Alien Objects

bottle image AS4 is now correctly identified, because the side of the bottle was correctly classified as planar. Three rejections (GB1, GB3, and TN2) are due to to initial under-segmentation of the object surface. For example, the initial segmentations of block images GB1 and GB3 merged the distinguishing 45° slanted surface (the blue patch in Figure B-9(c)) with its neighboring patch (colored green in Figure B-9(c)). We conclude that there is some degradation of performance; this suggests that the choice of number of clusters is important.

6.5.2 -- Effects of Perturbing Evidence Weights

We also consider the sensitivity of this recognition procedure to the values of the evidence weights $\mathbf{E}_{ji}$. Insensitivity to arbitrary perturbations of these values is desirable since these values are very subjective in nature. For example, it has been shown that the certainty factors used by MYCIN can be modified by plus or minus 0.2 without significantly affecting performance [Buc84].

We define a perturbation of the degrees of evidence matrix $\mathbf{E}$ used by our recognition system with two parameters: a probability of distortion ρ_d, and a restriction rule governing the major evidence weights. The restriction rules are important because of the significant role played by the major evidence; we expect that the assignment of major evidence weight 1.0 is not quite as subjective as the assignment of lower evidence weights. The restriction rules we have used are:

(A) Major evidence weights cannot be perturbed, and other evidence weights may not be perturbed to take the value 1.0.

(B) Major evidence weights cannot be perturbed, but other evidence weights may be perturbed to 1.0.

Table 6-6

Recognition Under Sparser Segmentations

	AS	HC	GB	TN	CB	MH	PL	DS	TY	HN
AS1	0.93	-0.34	-0.36	0.38	-0.25	-0.41	0.32	-0.33	0.39	0.17
AS2	0.93	-0.34	-0.10	0.46	-0.25	-0.21	0.37	-0.05	0.48	-0.01
AS3	0.85	-1.00	0.28	0.27	-1.00	-0.66	0.26	0.34	0.39	-0.30
AS4	0.93	-0.34	-0.36	0.38	-0.25	-0.41	0.32	-0.33	0.39	-0.01
HC1	-0.63	0.51	-1.00	-0.68	-0.25	-0.66	-0.72	-1.00	-0.68	0.28
HC2	-0.66	0.73	-0.70	-0.12	-0.25	-0.70	-0.77	-0.74	-0.35	0.39
HC3	-1.00	0.51	-1.00	0.00	0.00	-1.00	-1.00	-1.00	-0.68	0.28
HC4	-0.76	0.56	-1.00	-0.45	0.35	-0.78	-0.83	-1.00	-0.44	0.48
GB1	0.03	-0.34	-0.71	0.38	-0.25	-1.00	-0.72	-0.69	0.28	-1.00
GB2	0.60	-1.00	0.66	-0.27	-1.00	-0.63	0.13	-0.30	0.55	-0.30
GB3	0.03	-0.34	0.35	-0.45	-0.25	-0.78	0.32	0.42	0.28	-1.00
TN1	-0.23	-0.70	-0.36	0.65	-0.46	0.03	-0.12	-0.73	0.39	0.28
TN2	-0.63	0.23	-0.71	0.38	-1.00	-0.66	-0.72	-0.69	0.28	-0.30
TN3	-0.23	-0.70	-0.36	0.65	-0.25	0.55	0.32	-0.73	0.39	0.28
CB1	-0.63	0.32	0.24	-0.68	0.14	-0.66	0.26	-0.33	0.39	0.28
CB2	0.38	0.23	-1.00	0.27	0.87	0.32	0.19	-1.00	-0.44	0.28
CB3	-0.63	0.32	-1.00	0.38	0.50	-0.66	-0.72	-1.00	-0.68	0.28
MH1	-0.68	-1.00	-1.00	-1.00	-1.00	0.77	-0.12	0.00	0.28	0.28
MH2	-0.43	-0.34	-1.00	-0.31	-0.25	0.84	0.03	-1.00	0.39	0.39
MH3	-0.68	-1.00	-0.36	-1.00	-1.00	0.71	0.21	-0.69	0.39	0.28
PL1	-0.36	-0.70	-0.56	0.38	-0.77	-0.21	0.49	-0.12	-0.26	-0.44
PL2	-0.09	-1.00	-0.67	-0.63	-1.00	-0.63	0.51	-0.65	-0.01	0.40
PL3	-0.63	0.32	-0.36	-0.68	-0.64	-0.66	0.26	-0.73	0.39	0.39
DS1	-0.63	0.32	0.16	-0.45	-0.80	-0.70	-0.12	0.64	0.48	-0.30
DS2	-1.00	-1.00	0.28	-1.00	-1.00	-1.00	-1.00	0.69	0.39	-1.00
DS3	-1.00	0.23	0.28	-1.00	-1.00	-1.00	0.19	0.54	0.28	-0.30
TY1	-0.36	-0.70	-0.56	-0.27	-0.69	-0.29	-0.43	-0.53	0.78	0.48
TY2	-0.63	-0.70	-0.75	-0.45	-0.64	-0.41	-0.52	-0.73	0.68	0.39
HN1	-0.76	-0.07	-1.00	-0.45	-0.46	-0.78	-0.72	-1.00	-0.44	0.88
HN2	-0.68	-0.74	-1.00	0.00	-0.83	-0.41	0.26	-1.00	-0.68	0.46
HN3	-0.63	0.32	-0.75	0.38	-0.69	-0.66	-0.72	-0.23	0.48	-0.72

(C) Major evidence weights of 1.0 may both be created and changed.

Only one of these restriction rules will be in effect for a given experiment. To perturb the degrees of evidence in our evidence feature rule base, we consider each degree of evidence value E_{ji}. With probability p_d we try to perturb E_{ji}: by this we mean that

(a) If E_{ji} = -1.0, then it is replaced by -0.5;

(b) If E_{ji} = -0.5 or 0.0 then with equal probability it will be decreased by 0.5 or increased by 0.5;

(c) If E_{ji} = 0.5 and we are using restriction rule (B) or (C), then it will be perturbed by either 0.5 or -0.5 with equal probability;

(d) If E_{ji} = 0.5 and we are using restriction rule (A), then it will be replaced by 0.0;

(e) If E_{ji} = 1.0 and we are using restriction rule (A) or (B), then it is not changed;

(f) If E_{ji} = 1.0 and we are using restriction rule (C), then it will be replaced by 0.5.

We have experimented with perturbations of the degrees of evidence under all three restriction rules and with various probabilities of distortion. Note that under restriction rule (B) for a given j the number of i for which E_{ji}=1.0 may become larger than 1, and under restriction rule (C) an object i may not have any major evidence, contrary to specifications which we defined earlier. We do not try to avoid these potential violations. Table 6-7 shows the results of recognizing objects in our database of range images under these perturbations. For each p_d and restriction rule we report the number of misclassifications and the number of rejections out of the 31 range images, and also the number of misclassifications and rejections for the set of four alien objects.

Table 6-7
Recognition Under Perturbations of **E**

Rule, ρ_d	database images		alien images	
	misclassify	reject	misclassify	reject
(A) 0.3	1	1	0	4
(A) 0.6	2	4	1	3
(A) 1.0	2	5	1	3
(B) 0.3	3	0	3	1
(B) 0.6	6	0	3	1
(C) 0.5	7	5	3	1

We observe that modifying major evidence weights, either by changing them or by allowing other evidence weights to be changed to 1.0, has very serious effects on the performance of the system; it appears that the creation of major evidence features cannot be taken too lightly. On the other hand, when the major evidence features are fixed under rule (A), we find that the system performance is not bad, especially for ρ_d=1.0, although all supportive evidence weights (0.5) are reduced to 0.0. The overall impression obtained from these results is that the recognition relies heavily on the major evidence.

Given that the major evidence features form the crux of the recognition system, it is natural to ask how recognition would work if we removed the general rules from the rule base. We find that AS4 is still rejected, and that images GB1 and GB3 are misclassified as the aftershave bottle. All other images are correctly classified. The difficulty with

the block images was the fact that the sole major evidence feature for the aftershave bottle is also satisfied by the GB1 and GB3 segmentations. Since all positive evidence and no negative evidence for the bottle was found for GB1 and GB3, the similarity measure with the aftershave bottle was equal to 1.0. Therefore, it appears that general evidence features may be superfluous to an extent, but may be important in the correct classification of objects whose major evidence features may be observed in other objects of the database.

6.5.3 -- Effects of Object Distortion

We may also inquire about the effects of object distortion on recognition. Experiments based on both synthetic and real range images were made. First two distortions of the plug object (Figure 6-5) provided two new plug-like objects. The first distortion was created by expanding the object along the direction of axis of symmetry of its component cylinders by 10% (Figure 6-6); the second was created by expanding the object along the other two object axes by 10% (Figure 6-7). Whereas the undistorted plug image was correctly identified, the two distorted views were rejected. These distorted objects were rejected because the major evidence features for the plug involve tight bounds. This is not the case for the cup object, since the major evidence features for the cup are not very specific with respect to scaling, but concentrate instead on symbolic conditions such as the jump edge between a convex and a concave face. We also took a real range image of a different cup (Figure 6-8) than the cup shown in HC1 through HC4. This new cup image was identified as the cup in our object database.

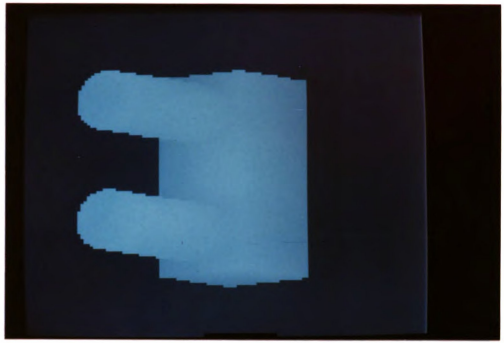


Figure 6-5
Plug Object

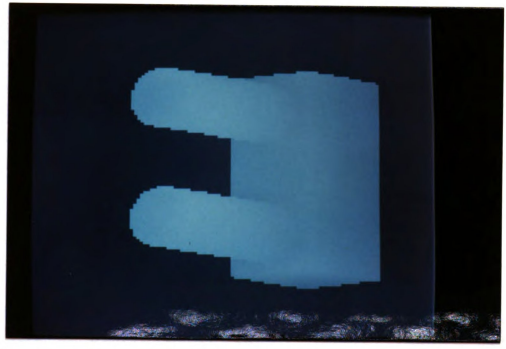


Figure 6-6
Distorted Plug 1

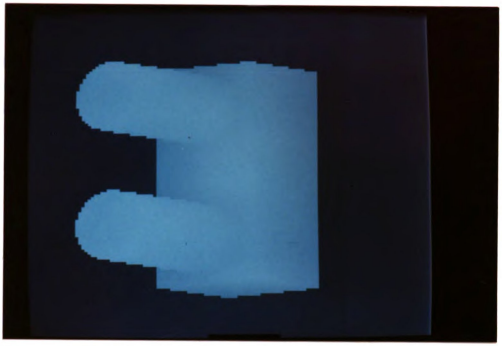


Figure 6-7
Distorted Plug 2

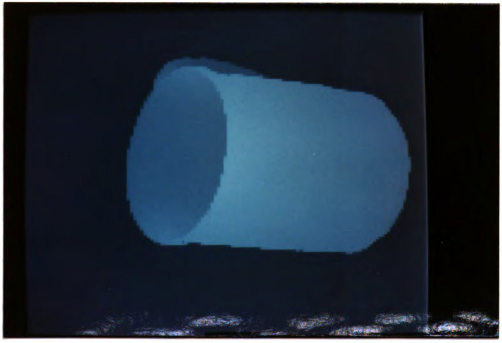


Figure 6-8
Range Image of a Different Cup

6.5.4 -- Multiple Object Scenes

Experiments up to now have dealt solely with range images containing single objects. The procedure we have discussed for deducing the identity of an object in a range image does not consider, after one object is found, whether others may also be found in the image. A reasonable approach to multiple object detection would be to implement the following control loop:

(0) Generate representation R_0 from the range image;
(1) Recognize a single object in R_0 as discussed. If no object is recognized, STOP.
(2) Remove those patches (and boundaries) from R_0 which were involved in the major evidence features detected in step (1) to obtain a reduced version of R_0, and go to (1).

To illustrate this procedure, a real range image of a roll of masking tape placed on top of the block object is considered (see Figure 6-9). When we apply our current evidence feature rule base to this range image, we find the block object is identified. Two evidence features for the tape roll, one relating the flat side of the roll to the concave inside cylindrical surface, the other relating the flat side of the roll to the convex outside surface, were added to the evidence rule base.

Figure 6-10 shows the segmentation and classification of patches for the tape-on-block range image. Applying our augmented evidence features to this data, the roll of tape is identified first; we remove the surface patches providing the major evidence for identification of the tape roll (the flat side and the concave inside surface) to obtain the set of surface patches shown in Figure 6-11. The recognition procedure next identifies the block object. When the surface patches contributing to the major evidence for the block are

Figure 6-9
Tape-on-Block Range Image

removed, we are left with the patches shown in Figure 6-12. The recognition routine fails to find an object in this final collection of surface patches, so the procedure terminates, having identified the tape roll and the block.

6.6 -- Summary and Discussion

In this chapter we have defined an object representation scheme based on the range image surface patches and patch classifications derived in Chapters 3, 4, and 5. We have demonstrated a knowledge-based merging process which helps to recover from oversegmentation of objects. A recognition procedure based on observing supporting and refuting evidence has recognized 30 out of 31 objects. We have shown that performance is fairly insensitive to perturbation of non-major evidence weights, and that general rules do not play a very large role in correct recognition. We have also demonstrated a prototype technique for identifying multiple objects in a scene.

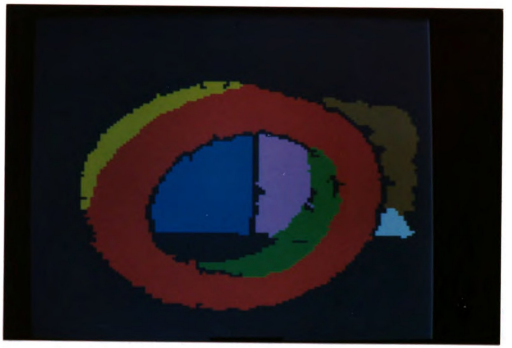


Figure 6-10
Segmentation of Tape-on-Block Range Image

Figure 6-11
Result of Removing Tape Patches

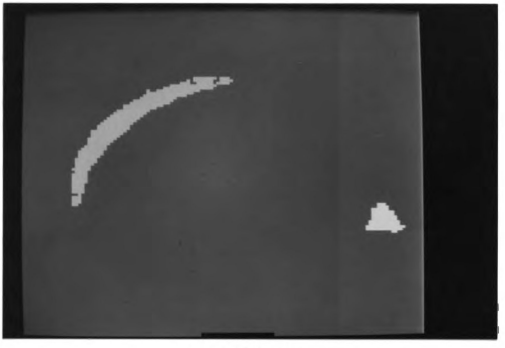


Figure 6-12
Result of Removing Tape and Block Patches

One difficulty with our technique is its inability to handle unremarkable objects; that is, objects which have no specific distinguishable features. The main difficulty lies in our selection of representation features: for example, if representations included further surface information such as classifications as cylinders or spheres, and parameters of these corresponding surface types, then the class of unremarkable objects would be diminished in size.

We have not addressed the task of determining the spatial location of objects, but have only concentrated on identification. Our recognition technique may be able to map specific patches in the range image to model object faces (namely, those patches involved in observations of major evidence) and this could provide a unique location of an object in a post-identification processing stage. However, some major evidence features do not involve patches (e.g. morphological features) and therefore would not provide such

information.

A final weakness is the method of construction of the evidence rule base. Our rule base was constructed by manually looking for "salient" features of objects and then deciding on what feature bounds to use, a very subjective and empirical approach. Ideally, the rule base would be constructed automatically from a number of object models, or perhaps would be obtained by presenting the system with "training range images" of objects in the database and allowing the system to derive the evidence features from these samples.

CHAPTER VII
SUMMARY, DISCUSSION, AND FUTURE RESEARCH

7.1 -- Summary

The goal of our work has been to develop a general 3D object recognition system using range images as input data. Range images are first enhanced to reduce noise and remove known background areas. The pixels in the image are then partitioned by clustering the set of six-dimensional feature vectors obtained by considering the three spatial coordinates and three unit surface normal coordinates corresponding to each pixel. A clustering scheme which minimizes a squared error criterion tends to produce regions which do not cross boundaries between natural object faces.

The initial partition of the image into surface patches was refined with techniques developed to classify both patches and boundaries between adjacent patches. A nonparametric statistical trend test was used to test for curvature in a surface patch, and was found to perform well. Its performance quickly diminished with increasing noise degradation and decreasing patch size. Two other tests for determining patch type were developed for small patches. Boundaries between patches were classified as jump, crease, or normal edges.

A merging procedure based on patch and boundary classification refines the initial cluster-induced surface segmentation. This technique performs best at reconstructing planar patches. Oversegmented curved surfaces are repaired to some extent. An error of linear fit was defined for boundaries between patches and between a patch and the background. The degree of linearity of a boundary is a useful feature for recognition.

The final stage is object recognition. A merging of patches based on knowledge about the objects in the database is performed to produce a representation relative to each object in the database. An evidence feature rule base is generated by hand and supplies salient information about patches and pairs of patches. A measure of similarity between observed features derived from the observation and supporting features present in the evidence feature rule base is developed; a similarity value is derived for each object in the object database. The maximum similarity value is used to identify the object in the range image. This system correctly identified objects in 30 out of 31 range images. Its performance did not degrade much under various perturbations of the system such as modifying the evidence weights, eliminating general evidence features, or when encountering objects not included in its object database.

7.2 -- Computational Complexity

The time complexities of processes defined in this thesis are reviewed, and some representative CPU times for these processes are reported. Values used in expressing complexities are defined as follows:

n_p the number of nonbackground pixels in a range image;
n_s the number of 6D patterns obtained by subsampling the range image;

n_c the number of clusters requested of CLUSTER;

n_q the number of pixels belonging to the largest surface patch;

N the number of patches;

n_o the number of objects in the database; and

n_r the number of rules or evidence features used for recognition.

The complexities of various stages of our range image processing paradigm are:

Jump edge detection: $O(n_p)$;

CLUSTERing of image: $O(n_c n_s)$;

Classifying boundaries as crease or normal: $O(Nn_p)$;

Classifying patches as planar, convex, or concave:

- Trend test: $O(n_p)$;
- Eigenvalue: $O(n_p)$;
- DON method: $O(n_p \log n_q)$;

Deriving morphological features: $O(n_p \log n_p)$;

Object recognition: $O(\max(n_r, N) n_o N^2)$

Table 7-1 shows the CPU time required for the sequence of processes executed in analyzing a cup image and a cobra image. The various components are denoted:

JUMP jump edge detection;

CLUSTER application of the CLUSTER technique, including postprocessing;

MERGE1 first boundary classification, merging of null edges, and reclassification of boundaries;

CLASS classification of patches produced by the first merge step;

MERGE2 merging over non-crease edges;

BOUND final boundary classification;

MORPH derivation of morphological features;

IDEN identification of object (merging and evidence-based recognition).

The cup image had 5772 nonbackground pixels and 5 surface patches were present in the initial representation for object recognition; The cobra image had 11419 nonbackground pixels and 8 surface patches were present in the initial representation for object recognition.

Table 7-1
Sample CPU Times for Range Image Analysis
on a Harris 500 Supermini Computer

Analysis Stage:	Timings (sec) Cup image	Cobra image
JUMP	43	92
CLUSTER	279	472
MERGE1	140	191
CLASS	75	131
MERGE2	11	12
BOUND	37	104
MORPH	146	131
IDEN	11	22
Total:	742	1155

7.3 -- Discussion

There are three main contributions of this thesis.

(1) A complete 3D object recognition system is developed.
(2) Nonparametric statistical techniques are used to classify surface patches.
(3) Evidence features are used for object recognition.

The complete object recognition system described in this thesis is different from other complete systems in the sense that it is able to treat both articulable and arbitrary-shaped objects. Some systems have handled articulable objects [Tom84] while assuming that objects are constructed from simple primitives, and some have handled arbitrary objects [Gri84], which are approximated by a large number of simple primitives and therefore require rigid geometry for recognition. Our recognition system does not require rigid geometry constraints to recognize objects, but may require local rigidity. Therefore, articulable objects may be treated by the system, as has been demonstrated with the hand object HN. Also, since surface patches are only classified as planar, convex, or concave, no implicit assumption about forms of surface functions has been made, allowing treatment of arbitrary objects.

Nonparametric statistical techniques are used in a trend test which ascertains the sense (planar, convex, or concave) of a surface patch. The contribution of the trend test is that it is the first application of nonparametric techniques to the task of determining object surface type. Some advantages of this nonparametric test are given below.

(1) Besides setting a significance level for the tests, no arbitrary thresholds based on (say) goodness of fit of a plane to the surface are required.
(2) The trend test performs well for surface patches with a moderate number of pixels.

The nonparametric trend test has the following disadvantages:

(1) Small patches or short boundaries cause the corresponding sample sizes to be too small for a reasonable application of the tests.
(2) The trend test deals exclusively with four fixed directions; these directions may not optimally span a

given patch.

Using evidence features for object recognition has the following advantages:

(1) Recognition proceeds by searching only for salient information.
(2) The recognition procedure has polynomial time complexity. Object recognition procedures which use tree searching have potentially exponential search times. However, lower computation times may not be realized for simple objects.
(3) Both symbolic and numeric information are used.
(4) Since the object models consist of lists of salient features of objects, the evidence feature rule base is compact. About 3 rules were required, on the average, for each object in our object database.

Some disadvantages of the evidence-based recognition are:

(1) The generation of the evidence feature rule base is not a well-defined procedure; the concept of a salient feature needs to be formalized.
(2) There is difficulty with objects which have no salient features.
(3) The object modeling scheme is not complete. A given model may correspond to more than one object. For example, a left shoe and a right shoe will have the same model under our modeling scheme.

7.4 -- Future Research

The following suggestions for future research will extend the results presented in this thesis.

(1) Small surface patches tend to be classified as planar patches. By requesting a fixed number of clusters from the segmentation process, we find that, almost all patches in range images with few object pixels will be too small to have a chance of being correctly classified. Setting the requested number of clusters to be a function of the number of object pixels may provide better results.

(2) The crease edge detection technique suffers from a high false alarm rate under noise degradation. This is a result of using the CLUSTER algorithm to identify surface patches. Patch boundaries will tend to occur when the difference between estimated unit surface normals of two adjacent pixels is large, a condition which is produced by noise degradation as well as by true surface creases. Fitting a smooth curve to boundaries obtained by CLUSTER may modify the set of boundary pixels enough to obtain more reliable crease edge detection.

(3) Knowledge-based merging currently makes use only of angle information. It may be possible to use other types of knowledge about the objects in the database, such as maximum and minimum patch sizes, or the number of planar, convex, and concave object faces.

(4) The rule base structure for our recognition process is very simple: evidence features are either 0th level, 1st level, or 2nd level evidence conditions. A more useful rule base structure would include the possibility of conjunctions of these evidence conditions. In

particular, it may be easier to develop major evidence for unremarkable objects. For example, an evidence condition indicating that there are 2 convex-hull-based background components (a 0th level evidence condition) and that no object patch has surface area greater than 4 (a 1st level evidence condition) would provide a stronger major evidence feature for the mushroom object than is currently used (Rule 21).

(5) The recognition technique utilizes only a bottom-up approach to recognition, since the only control process which occurs is the detection of evidence conditions triggered by the observed representation, which in turn constructs a similarity measure. It may be better to introduce a top-down component to the recognition process: that is, upon observing a major evidence feature for object i, which provides identification of two (or more) object faces in the observed representation, a search for other patches which belong to object i can be initiated.

APPENDICES

APPENDIX A
SYNTHETIC RANGE IMAGE GENERATION

This section describes the principles and techniques underlying the synthetic range image generation software written for use in preliminary tests of the range image analysis techniques presented in this thesis. The general approach is a Constructive Solid Geometric approach: objects to be defined are unions and differences of primitive 3D shapes. The set of primitives we have implemented is (sphere, cylinder, box), and the addition of other primitives (such as cones, elliptical cylinders, etc.) is very straightforward.

For simplicity of notation and operation, we embed points in $\mathbb{R}^3$ into $(\mathbb{R}^3 \times (1))$ by the mapping $(x,y,z)^t \rightarrow (x,y,z,1)^t$. This allows us to perform rotation and translation by multiplying by a 4x4 rotation/translation (RT) matrix. As a brief review, such RT matrices have the general form

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ \hline 0 & 0 & 0 & 1 \end{array}\right) = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & 1 \end{array}\right)$$

where A_{11} is a rotation matrix and A_{12} is a translation vector.

We define the mapping $C:(\mathbb{R}^3 x(1))\rightarrow\mathbb{R}^3$ such that

$$C[(x,y,z,1)^t] = (x,y,z)^t$$

and extend the notation to an operation on a set of points S in $\mathbb{R}^3x(1)$ in the obvious way:

$$C[S] = \{(x,y,z)^t: \ (x,y,z,1)^t \epsilon S\}.$$

User input consists of a sequence $(D_i)_{i=1}^n$ of primitive object definitions and a global RT matrix G. For each i, $D_i = \langle \mathbf{P}_i, \bullet_i, H_i \rangle$ where:

$\mathbf{P}_i$ is a set of 4-dimensional points $(x,y,z,1)^t$ such that $C[\mathbf{P}_i]$ defines a 3D primitive with user-specified shape parameters (e.g. radius for sphere, radius and height for cylinder) in canonical position (i.e., no translation or rotation);

$\bullet_i$ is a set operation, either union ($\cup$) or set difference ($\setminus$);

H_i is a RT matrix of the primitive.

For a RT matrix H and a set of points $\mathbf{P}$ in $\mathbb{R}^3x(1)$, we define

$$H\mathbf{P} = \{H(x,y,z,1)^t: \ (x,y,z,1)^t \epsilon \mathbf{P}\}.$$

The range image which is returned corresponds to the object

$$C[G(\cdots(((\emptyset \bullet_1 H_1\mathbf{P}_1)\bullet_2 H_2\mathbf{P}_2)\bullet_3 H_3\mathbf{P}_3)\bullet_4 H_4\mathbf{P}_4)\cdots\bullet_n H_n\mathbf{P}_n)].$$

We can think of

$$C[(\cdots(((\emptyset \bullet_1 H_1\mathbf{P}_1)\bullet_2 H_2\mathbf{P}_2)\bullet_3 H_3\mathbf{P}_3)\bullet_4 H_4\mathbf{P}_4)\cdots\bullet_n H_n\mathbf{P}_n)]$$

as the "raw object definition". Only G need be changed in order to obtain different views of the object. The range image consists of a grid of numbers, with rows corresponding to increments in the y direction and columns corresponding to increments in the x direction. The range value r(x,y) is the largest z value such that $(x,y,z)^t$ lies on the object surface.

The basic algorithm proceeds as follows:

(1) Derive D_i's and G from user input.

(2) Step along (x,y) grid. For each (x,y):

(a) For i=1,...,n find all z's such that $(x,y,z)^t$ lies on the surface of $C[GH_iP_i]$. For the convex primitives we work with, there can be at most two such values. Using these values, we create the one-dimensional set O_i as follows: If there are no z's, set $O_i=\emptyset$; if there is exactly one z (this situation occurs with probability zero), set $O_i=(z)$. If two solutions z_1 and z_2 exist $(z_1<z_2)$ then set $O_i=[z_1,z_2]$.

(b) Derive the one-dimensional set

$$S = (\cdots(((\emptyset \bullet_1 O_1) \bullet_2 O_2) \bullet_3 O_3) \cdots \bullet_n O_n)$$

and let r(x,y) be $z = \sup(S)$ ($z=-\infty$ if $S=\emptyset$).

(3) Convert r(x,y)'s to 8-bit integers; $z=-\infty$ is mapped to 0. [This is not necessary; we perform this step to emulate the image format obtained with the real range image sensor system.]

Step 2b above is tedious but straightforward to implement. However, the derivation of z values where the 3D ray defined by constant x and y coordinates hits the surface of $C[GH_iP_i]$ is not so obvious and is treated in some detail below. Each type of primitive demands a slightly different approach to this task; I present the derivation for the case of a cylindrical primitive (consisting as it does of both planar and curved surfaces); the extension to further primitives should then be reasonably clear. The basic idea behind this derivation is that we can apply an inverse RT transformation to x, y, and $C[GH_iP_i]$ in order to work with the primitive object in its canonical position.

Let primitive P_i be a cylinder. The shape parameters of a cylinder are a radius r and a height h. Suppose our canonical form of a cylinder is one whose axis of rotational symmetry lies on the z axis and whose two planar end-surfaces have equations z=h/2 and z=-h/2. We are given (x,y) and wish to find all z's (if any exist) such that $(x,y,z)^t$ falls on the surface of $C[GH_iP_i]$. Note that this is equivalent to the point $C[H_i^{-1}G^{-1}(x,y,z,1)^t]$ falling on the surface of $C[P_i]$, the canonical primitive. Let

$$H_i^{-1}G^{-1} = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Denote

$$(x',y',z')^t = C[H_i^{-1}G^{-1}(x,y,z,1)^t] = \begin{pmatrix} \alpha_1x + \alpha_2y + \alpha_3z + \alpha_4 \\ \beta_1x + \beta_2y + \beta_3z + \beta_4 \\ \gamma_1x + \gamma_2y + \gamma_3z + \gamma_4 \end{pmatrix}.$$

The equation of the curved surface of $C[P_i]$ is

$$(x')^2 + (y')^2 = r^2.$$

That is,

$$(\alpha_3z + \eta_1)^2 + (\beta_3z + \eta_2)^2 = r^2$$

where

$$\eta_1 = \alpha_1x + \alpha_2y + \alpha_4$$

$$\eta_2 = \beta_1x + \beta_2y + \beta_4.$$

We get real solution(s) of z (if they exist) by solving the quadratic

$$(\alpha_3^2 + \beta_3^2)z^2 + 2(\alpha_3\eta_1 + \beta_3\eta_2)z + (\eta_1^2 + \eta_2^2 - r^2) = 0.$$

For each solution obtained, find

$$z' = \gamma_1x + \gamma_2y + \gamma_3z + \gamma_4;$$

if $-h/2 \leq z' \leq h/2$, then $(x,y,z)^t$ is a surface point of the curved surface of $C[GH_iP_i]$.

If two surface points have not been found, we test for $(x,y,z)^t$ falling on the end-surfaces of the cylinder: solve

$$z' = h/2 = r_1x + r_2y + r_3z + r_4$$

to get

$$z = (h/2 - r_1x - r_2y - r_4)/r_3;$$

find x' and y', and if

$$(x')^2 + (y')^2 \leq r^2$$

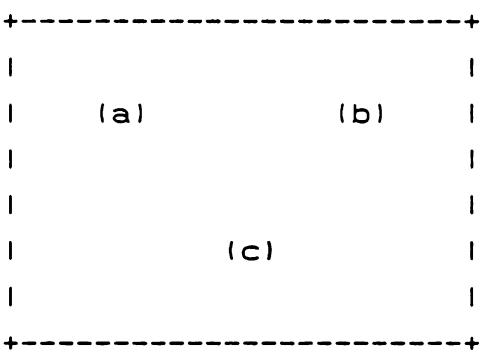
we conclude that $(x',y',z')^t$ falls on the end-surface $z' = h/2$. An identical procedure is carried out for the end-surface $z' = -h/2$. This completes the processing for the cylinder primitive for a given ray (x,y).

Corresponding implementation of the sphere and box primitives are corollaries of the implementation of the cylinder primitive. If the canonical position of the sphere primitive has its center at the origin, the equation $(x')^2 + (y')^2 + (z')^2 = r^2$ provides a quadratic in z whose real solutions give corresponding surface points. If the canonical position of the box primitive is such that its faces have equations of the form x=constant, y=constant, or z=constant, then six tests, one for each face and similar to the end-surface tests for the cylinder, are carried out.

APPENDIX B

RANGE IMAGES AND RESULTS

This appendix shows the 31 range images in our database and results obtained by applying techniques of this thesis to them. The format for each figure is:



where

(a) shows the range image after smoothing and removing background pixels.

(b) shows the surface patches resulting from segmentation.

(c) shows the surface patches after merging based on classifications of patches and boundaries between patches.

Figure B-1
Aftershave Bottle View 1

Figure B-2
Aftershave Bottle View 2

Figure B-3
Aftershave Bottle View 3

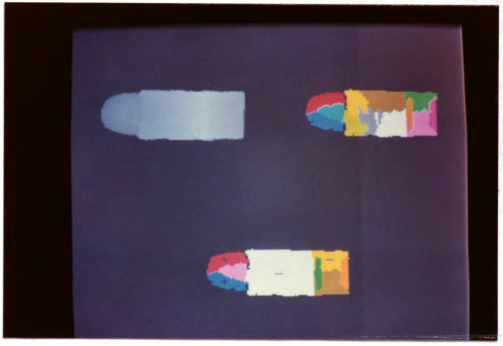


Figure B-4
Aftershave Bottle View 4

Figure B-5
Cup View 1

Figure B-6
Cup View 2

Figure B-7
Cup View 3

Figure B-8
Cup View 4

Figure B-9
Block View 1

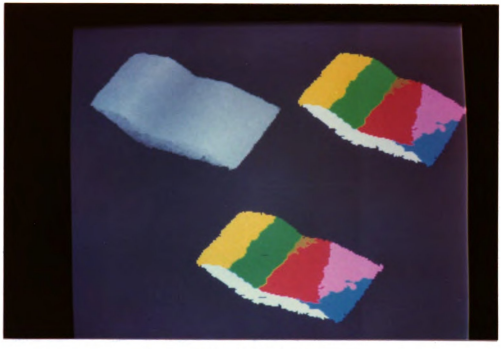


Figure B-10
Block View 2

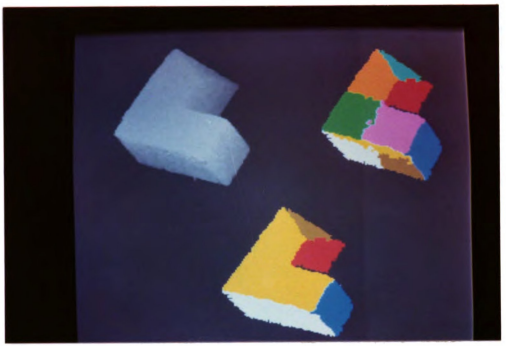


Figure B-11
Block View 3

Figure B-12
Tunnel View 1

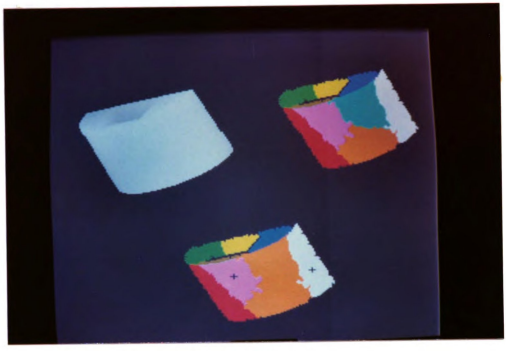


Figure B-13
Tunnel View 2

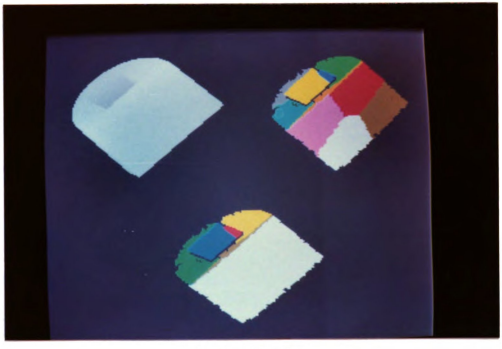


Figure B-14
Tunnel View 3

Figure B-15
Cobra Sculpture View 1

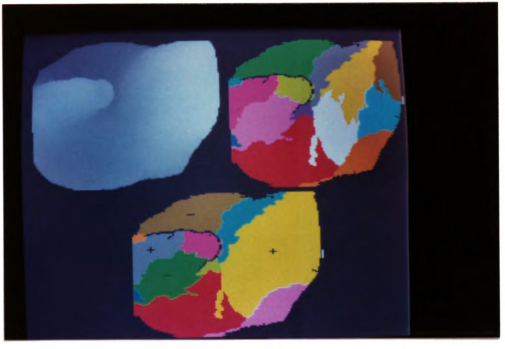


Figure B-16
Cobra Sculpture View 2

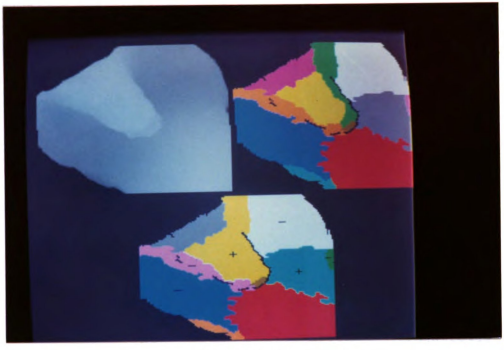


Figure B-17
Cobra Sculpture View 3

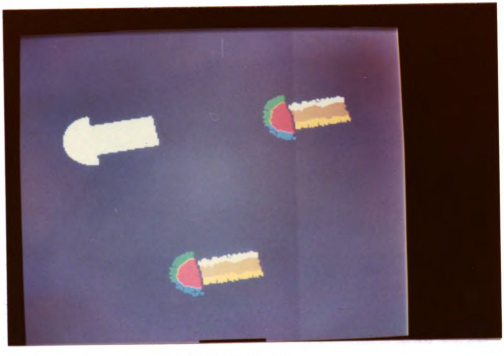


Figure B-18
Mushroom View 1

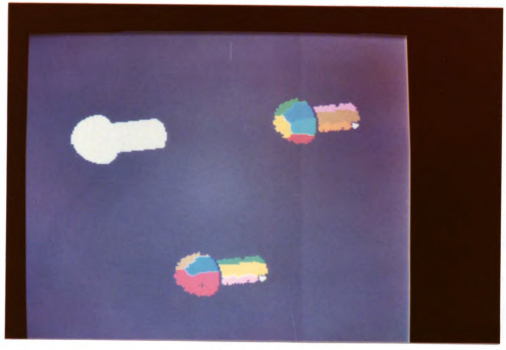


Figure B-19
Mushroom View 2

Figure B-20
Mushroom View 3

Figure B-21
Plug View 1

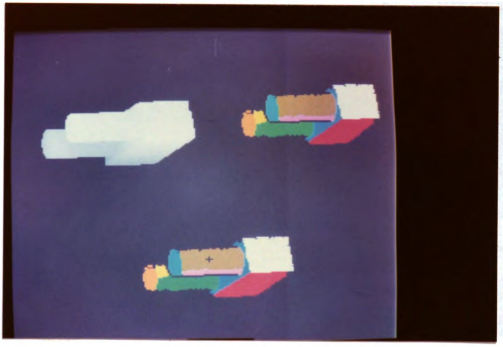


Figure B-22
Plug View 2

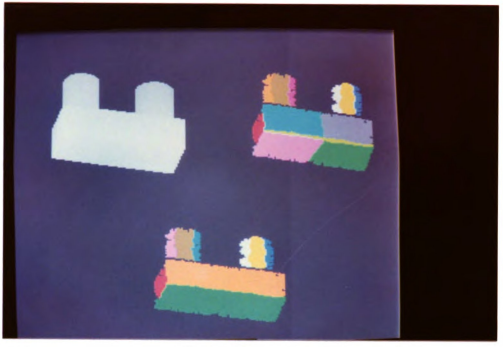


Figure B-23
Plug View 3

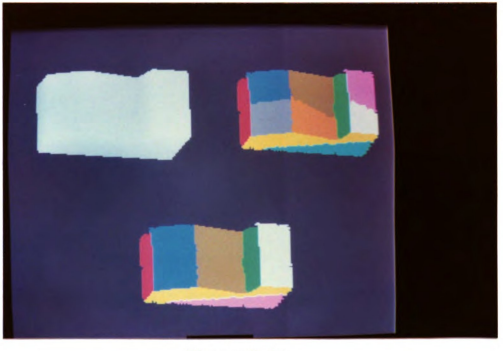


Figure B-24
Diesel View 1

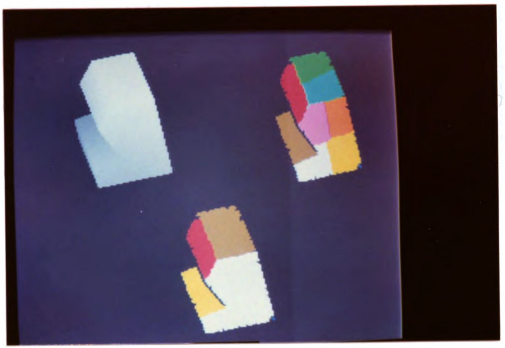


Figure B-25
Diesel View 2

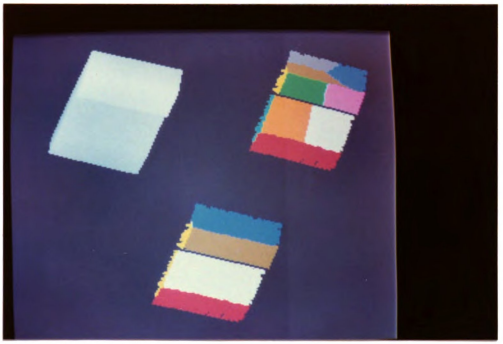


Figure B-26
Diesel View 3

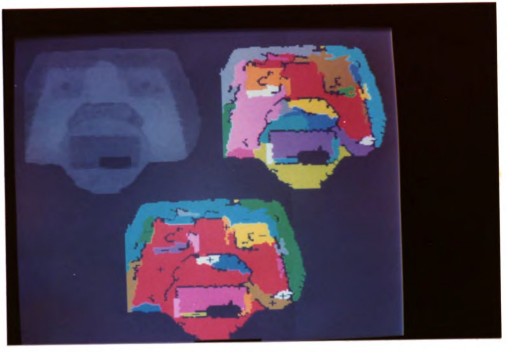


Figure B-27
Toy Part View 1

Figure B-28
Toy Part View 2

Figure B-29
Human Hand View 1

Figure B-30
Human Hand View 2

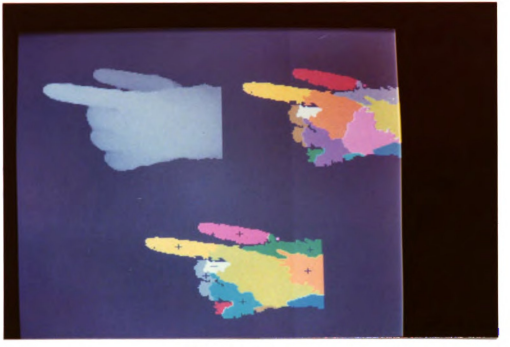


Figure B-31
Human Hand View 3

APPENDIX C
CLUSTER ALGORITHM

The following gives a brief outline of the CLUSTER algorithm used in this thesis to perform segmentation of range images.

(a) Input pattern vectors, consider these to form one cluster.

(b) Create clusterings containing 2, 3, ..., C clusters:

  (b1) Look for that pattern with greatest distance from the cluster center of its parent cluster. Let this pattern form a new cluster seed.

  (b2) Suppose there are D clusters: A pattern is moved from its parent cluster to each of the remaining (D-1) clusters, one by one, attempting to reduce squared error. A move will be permanent if squared error is reduced. Otherwise, restore the pattern to its parent cluster. Perform this procedure for all patterns in the data. Repeat this process until no moves will reduce the squared error.

(c) Create clusterings containing C, C-1,...,2 clusters:

  (c1) For each pair of clusters, consider the result of merging the clusters. If lower squared error would result, mark that pair for possible merging. After all pairs have been considered,

merge that marked pair which provides the largest decrease in squared error. Repeat this step to produce C-2, ...,2 clusters.

(d) Compare clusterings containing 2, 3,..., C clusters generated in latest execution of steps (b) and (c) to current set of best clusterings (lowest squared error). If there are some clusterings which have lower squared error than achieved during earlier executions of (b) and (c), then store the better clusterings and repeat steps (b) and (c). Otherwise, report the current set of best clusterings and terminate execution.

Part (b) has time complexity $O(n_s C)$, where n_s is the number of patterns to be clustered, and part (c) has time complexity $O(C^2)$. Therefore, since $C < n_s$ in general, the time complexity of this cluster algorithm is $O(n_s C)$.

APPENDIX D

EVIDENCE FEATURE RULES

This appendix provides the specifications of the evidence features which were used in our experiments. Each rule shows:

(0) The rule index, j.

(1) A purpose identifier, which is enclosed in <<< >>> brackets and specifies the object (if any) for which the evidence feature is a major evidence. If the evidence feature is not major evidence for any object, the rule has purpose identifier <<< general rule >>>.

(2) A verbal description of the evidence condition. Distances are given in inches, areas are given in square inches, and angles are given in degrees.

(3) Evidence weights $_{ji}$ for i=1,...,10, corresponding to the ten objects in our database, listed in the order: AS,HC,GB,TN,CB,MH,PL,DS,TY,HN.

```
Rule 1:          <<< general rule >>>
the perimeter is greater than 20.
-0.5 -0.5 -0.5 -0.5  0.5 -1.0 -0.5 -0.5  0.5  0.5

Rule 2:          <<< general rule >>>
there are 2 to 3 faces with area between 6 and 10;
-1.0  0.5  0.5 -1.0  0.5 -1.0  0.5  0.5  0.0  0.0

Rule 3:          <<< general rule >>>
there is exactly one convex face with area larger than 10;
-1.0  0.5 -1.0  0.5  0.5 -1.0 -1.0 -1.0  0.5  0.5

Rule 4:          <<< general rule >>>
there are exactly two convex faces with areas between
1.0 and 3.0;
-1.0  0.0 -1.0  0.0  0.0  0.5  0.5 -1.0  0.0  0.5

Rule 5:          <<< general rule >>>
there is a nonlinear background boundary
 0.5  0.5 -0.5  0.5  0.5  0.5  0.5 -0.5  0.5  0.5

Rule 6:          <<< general rule >>>
there are two edges with angles of 80 to 100 degrees;
 0.5  0.0  0.5  0.5  0.0  0.5  0.5  0.5  0.5  0.0

Rule 7:          <<< general rule >>>
there is one edge with angle of -100 to -80 degrees,
and the edge is linear;
-0.5  0.0  0.5  0.5 -0.5 -0.5 -0.5  0.5  0.5  0.0
```

Rule 8: <<< general rule >>>
there is a jump edge between two planar faces,
the jump gap is between 1 and 3;
0.0 0.0 0.5 0.5 0.0 0.5 0.5 0.5 0.5 0.0

Rule 9: <<< Aftershave bottle >>>
there are 1 or 2 planar faces, area between 4.0 and 5.5,
and boundary with background is linear;
1.0 -0.5 0.5 0.5 -0.5 -1.0 0.5 0.5 0.0 -0.5

Rule 10: <<< Cup >>>
there is a jump edge between a convex surface of area
6 to 16 and a concave surface of area 2 to 8, and the
jump gap is between 2.5 and 3.5;
-1.0 1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -0.5

Rule 11: <<< Cup >>>
there is a jump edge from a convex surface of area
greater than 10 to a planar patch -- jump gap 3 to 4;
-1.0 1.0 0.0 -1.0 0.5 -1.0 -1.0 -1.0 -1.0 0.0

Rule 12: <<< Cup >>>
there is a jump edge from a convex surface of area
greater than 10 to a patch of area less than 1.5;
-1.0 1.0 -1.0 0.0 0.0 -1.0 -1.0 -1.0 -1.0 0.0

Rule 13: <<< Cup >>>
there is a circular patch: area is between 7.0 and
9.0, and maximum distance within the patch is between
3.0 and 3.4;
-1.0 1.0 -1.0 -1.0 -0.5 -1.0 -1.0 -1.0 -1.0 0.0

Rule 14: <<< Block >>>
there are two parallel planar faces with minimum
distance between 1.3 and 1.5;
-1.0 -1.0 1.0 -1.0 -1.0 -1.0 0.5 -1.0 0.0 0.0

Rule 15: <<< Block >>>
there are two adjacent planes whose normal vectors form
an angle of 40-50 degrees, one plane with area in
(1.0,3.1) and the other with area in (2.0,6.1);
-1.0 -1.0 1.0 -1.0 0.0 -1.0 -1.0 0.0 0.5 0.0

Rule 16: <<< Block >>>
there are two planar faces separated by a distance of
3 to 3.5 whose normals form an angle of 130-140 degrees.
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

Rule 17: <<< Block >>>
there are two remote planes w/ normal angle 40-50 degrees,
areas in (2.0,3.1) and (2.0,5.0), and min distance 1.5-2.5.
-1.0 -1.0 1.0 -1.0 -1.0 -1.0 -1.0 0.0 -0.5 -0.5

Rule 18: <<< Tunnel >>>
there are two parallel planar faces with minimum
distance between 1.9 and 2.1;
-0.5 -1.0 0.0 1.0 -0.5 0.5 0.5 -1.0 0.0 0.0

Rule 19: <<< Tunnel >>>
there is a convex face with area in (10,21) which is a
minimum distance (1.4,1.8) from a planar patch of area
less than 4;
0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0

Rule 20: <<< Cobra sculpture >>>
there are pairs of concave patches, size of each greater than 4, which have a minimum distance of 1.5 to 9;
-1.0 -1.0 -1.0 -1.0 1.0 -1.0 -1.0 -1.0 -1.0 0.0

Rule 21: <<< Mushroom >>>
there is no patch larger than 4;
-0.5 -0.5 -0.5 -0.5 -0.5 1.0 -0.5 0.0 0.0 0.0

Rule 22: <<< Plug >>>
there are two instances of a planar patch of area less than 4 and a planar patch of area less than 1.2 being parallel with distance between 2 and 2.4;
-1.0 -1.0 -1.0 0.0 -1.0 -1.0 1.0 -1.0 -1.0 -1.0

Rule 23: <<< Plug >>>
there are two planar patches of area less than 1 which are parallel with distance between 1.4 and 1.8;
-1.0 -1.0 -1.0 -1.0 -1.0 -1.0 1.0 -1.0 0.0 0.0

Rule 24: <<< Plug >>>
there are two planar faces each with areas in (5.0,6.5) with intersection angle of 80 to 95 degrees and linear;
0.0 -1.0 0.5 0.0 -1.0 -1.0 1.0 -1.0 -1.0 -1.0

Rule 25: <<< Plug >>>
there are two planar faces with area in (0.5,1.0) which intersect with another patch at an angle 80-95 degrees;
-1.0 0.0 0.0 -1.0 0.0 0.0 1.0 0.0 0.0 0.0

Rule 26: <<< Diesel >>>
there are two planes, one with area less than 2.6 and the other with area less than 4.5, whose normal vectors form an angle -45 to -55 and lie 1.7 to 2.3 apart;
-1.0 -1.0 0.5 -1.0 -1.0 -1.0 -1.0 1.0 0.5 -1.0

Rule 27: <<< Diesel >>>
there are two planes, both with area in (3.0,4.5) whose normal vectors form an angle 60-70 and are adjacent;
-1.0 -1.0 0.5 -1.0 -1.0 -1.0 -1.0 1.0 0.5 0.0

Rule 28: <<< Diesel >>>
there are two planes, one with area less than 3.3 and the other with area less than 4.5, whose normal vectors form an angle 20-30 degrees, and lie 2.0 to 2.4 apart;
-1.0 -1.0 0.5 -1.0 -1.0 -1.0 -1.0 1.0 0.0 0.0

Rule 29: <<< Toy part >>>
there are at least 4 jump edges with gap less than 1;
-1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 1.0 0.5

Rule 30: <<< Human hand >>>
there are two to five finger-shaped patches: area is between 1.2 and 2.1, and maximum distance within the patch is between 2.0 and 3.0;
0.0 -1.0 -1.0 -1.0 0.0 -1.0 0.0 -1.0 -0.5 1.0

Rule 31: <<< Human hand >>>
there is 1 background component but more than 3 CH components, perimeter is at least 25;
-1.0 -0.5 -1.0 -1.0 -0.5 -1.0 0.0 -1.0 -1.0 1.0

APPENDIX E

RECOGNITION RESULTS

This supplement provides the results of applying the recognition scheme for each of the 31 range images in our range image database. These results are presented as follows:

(a) The vector of similarities $\tau_i^m(R_i)$, given in the order (AS,HC,GB,TN,CB,MH,PL,DS,TY,HN);

(b) Those objects for which representation R_i satisfied at least some major evidence condition for object i;

(c) The final decision about the object, if the object was not rejected;

(d) If (d) specified recognition of object $\hat{i}$, then the evidence features satisfied by $R_{\hat{i}}$ are listed.

results for AS1

Similarity vector:

0.91 -0.34 -0.36 0.38 -0.26 -0.65 0.29 -0.33 0.28 -0.30

Major evidence occurs for: AS

Decide the object is the Aftershave bottle

Reasoning: 5,9

results for AS2

Similarity vector:

0.82 -1.00 0.20 0.27 -1.00 -1.00 0.20 0.24 0.00 -1.00

Major evidence occurs for: AS

Decide the object is the Aftershave bottle

Reasoning: 9

results for AS3

Similarity vector:

0.82 -1.00 0.20 0.27 -1.00 -1.00 0.20 0.24 0.00 -0.30

Major evidence occurs for: AS

Decide the object is the Aftershave bottle

Reasoning: 9

results for AS4

Similarity vector:

0.41 0.23 -1.00 0.27 0.33 0.33 0.20 -1.00 0.28 0.28

Major evidence occurs for no objects.

Reject the object.

results for HC1

Similarity vector:

-0.75 0.56 -0.75 -0.80 -0.26 -0.78 -0.50 -0.73 -0.44 0.28

Major evidence occurs for: HC

Decide the object is the Cup

Reasoning: 2,5,13

results for HC2

Similarity vector:

-0.75 0.56 -1.00 -0.45 -0.26 -0.78 -0.82 -1.00 -0.44 0.40

Major evidence occurs for: HC,HN

Decide the object is the Cup

Reasoning: 3,5,11

results for HC3

Similarity vector:

-1.00 0.51 -1.00 0.00 0.00 -1.00 -1.00 -1.00 -0.68 0.00

Major evidence occurs for: HC,TN

Decide the object is the Cup

Reasoning: 3,12,19

results for HC4

Similarity vector:

-0.75 0.56 -1.00 -0.45 0.33 -0.78 -0.82 -1.00 -0.44 0.28

Major evidence occurs for: HC

Decide the object is the Cup

Reasoning: 3,5,10

results for GB1

Similarity vector:

0.08 -0.34 0.63 0.38 -1.00 -1.00 -0.71 -0.47 0.39 -1.00

Major evidence occurs for: AS,GB

Decide the object is the Block

Reasoning: 6,9,14,15

results for GB2

Similarity vector:

-0.05 -0.74 0.60 -0.27 -1.00 -0.83 -0.34 -0.69 0.28 -1.00

Major evidence occurs for: AS,GB

Decide the object is the Block

Reasoning: 9,14,15

results for GB3

Similarity vector:

0.08 -0.34 0.63 -0.68 -0.70 -1.00 0.29 0.34 -0.30 -1.00

Major evidence occurs for: AS,GB

Decide the object is the Block

Reasoning: 2,9,15,17

results for TN1

Similarity vector:

-0.43 -0.70 -0.36 0.65 -0.47 0.05 -0.10 -0.73 0.39 0.28

Major evidence occurs for: TN

Decide the object is the Tunnel

Reasoning: 5,7,18

results for TN2

Similarity vector:

-1.00 0.23 -0.71 0.65 -1.00 -0.65 -0.71 -0.69 0.39 -1.00

Major evidence occurs for: TN

Decide the object is the Tunnel

Reasoning: 3,8,19

results for TN3

Similarity vector:

-0.21 -0.70 -1.00 0.60 -0.26 0.47 0.29 -1.00 0.28 0.28

Major evidence occurs for: TN

Decide the object is the Tunnel

Reasoning: 5,18

results for CB1

Similarity vector:

-0.67 -0.07 -0.01 -0.02 0.70 -0.78 -0.75 -1.00 0.48 -0.01

Major evidence occurs for: GB,CB

Decide the object is the Cobra sculpture

Reasoning: 1,2,5,9,15,20

results for CB2

Similarity vector:

0.41 0.23 -0.84 0.27 0.70 0.33 0.20 -0.82 -0.44 -0.01

Major evidence occurs for: CB

Decide the object is the Cobra sculpture

Reasoning: 2,3,5,8,9,20

results for CB3

Similarity vector:

0.41 0.23 -1.00 0.27 0.51 0.33 0.20 -1.00 0.28 0.28

Major evidence occurs for: CB

Decide the object is the Cobra sculpture

Reasoning: 5,7,20

results for MH1

Similarity vector:

-0.67 -0.34 -1.00 -0.31 -0.26 0.82 -0.10 -1.00 0.28 0.28

Major evidence occurs for: MH

Decide the object is the Mushroom

Reasoning: 4,5,21

results for MH2

Similarity vector:

-0.67 -0.34 -1.00 -0.31 -0.26 0.82 -0.10 -1.00 0.28 0.28

Major evidence occurs for: MH

Decide the object is the Mushroom

Reasoning: 4,5,21

results for MH3

Similarity vector:

-1.00 -1.00 -0.36 -1.00 -1.00 0.75 -0.36 0.24 0.28 0.00

Major evidence occurs for: MH

Decide the object is the Mushroom

Reasoning: 4,21

results for PL1

Similarity vector:

-0.05 -0.74 -0.88 -0.63 -0.70 -0.78 0.65 -0.86 -0.68 -0.01

Major evidence occurs for: AS,PL

Decide the object is the Plug

Reasoning: 5,9,23,25

results for PL2

Similarity vector:

-0.01 -1.00 0.28 0.11 -1.00 -0.39 0.54 -0.47 0.28 -1.00

Major evidence occurs for: AS,TN,PL

Decide the object is the Plug

Reasoning: 8,9,18,25

results for PL3

Similarity vector:

-0.62 -0.70 -0.56 0.27 -0.79 -0.39 0.50 -0.86 0.13 -0.44

Major evidence occurs for: PL,TY

Decide the object is the Plug

Reasoning: 4,5,24

results for DS1

Similarity vector:

-0.67 0.32 0.16 -0.45 -0.81 -0.70 -0.10 0.64 0.48 -0.30

Major evidence occurs for: DS

Decide the object is the Diesel

Reasoning: 5,7,9,27,28

results for DS2

Similarity vector:

-1.00 -1.00 0.28 -1.00 -1.00 -1.00 -1.00 0.69 0.39 -1.00

Major evidence occurs for: DS

Decide the object is the Diesel

Reasoning: 26,27

results for DS3

Similarity vector:

-1.00 0.23 0.28 -1.00 -1.00 -1.00 0.20 0.54 0.00 -1.00

Major evidence occurs for: DS

Decide the object is the Diesel

Reasoning: 9,28

results for TY1

Similarity vector:

-0.83 -0.83 -0.58 -0.09 -0.46 -0.87 -0.88 -0.54 0.83 0.48

Major evidence occurs for: TN,TY

Decide the object is the Toy part

Reasoning: 1,3,6,7,8,19,29

results for TY2

Similarity vector:

-0.70 -0.74 -0.61 -0.34 -0.46 -0.62 -0.61 -0.58 0.78 0.48

Major evidence occurs for: TY

Decide the object is the Toy part

Reasoning: 1,5,7,8,29

results for HN1

Similarity vector:

-0.82 -0.70 -0.90 -0.73 -0.46 -0.88 -0.84 -0.89 0.23 0.92

Major evidence occurs for: TY,HN

Decide the object is the Human hand

Reasoning: 1,5,8,29,30,31

results for HN2

Similarity vector:

-0.82 -0.27 -0.85 -0.51 -0.30 -0.68 -0.56 -0.87 -0.11 0.68

Major evidence occurs for: HN

Decide the object is the Human hand

Reasoning: 1,5,18,21,31

results for HN3

Similarity vector:

-1.00 -0.34 -0.84 -0.31 0.33 -1.00 -1.00 -0.82 0.48 0.62

Major evidence occurs for: HN

Decide the object is the Human hand

Reasoning: 1,8,30

LIST OF REFERENCES

[Bad79] Badler, N., J. O'Rourke, H. Toltzis, A spherical representation of a human body for visualizing movement, Proc. IEEE, Vol. 67, pp1397-1403, 1979.

[Bai82] Bailey, T.A.,Jr., and Dubes, R., Cluster validity profiles, Pattern Recognition, Vol. 15, pp61-83, 1982.

[Bam83] Bamba, T., H. Maruyama, E. Ohno, and Y. Shiga, A visual sensor for arc-welding robots, Robot Vision, Alan Pugh, (ed.), Springer-Verlag, New York, pp169-178, 1983.

[Bar71] Barrow, H.G. and R.J. Popplestone, Relational descriptions in picture processing, Machine Intelligence 6 (eds. B. Meltzer and D. Michie), Edinburgh University Press, pp377-396, 1971.

[Bar72] Barrow, H.G., A.P. Ambler, and R.M. Burstall, Some techniques for recognising structures in pictures, Frontiers of Pattern Recognition, S. Watanabe, Ed., Academic Press, New York, pp1-29, 1972.

[Bar80] Barnard, S. and W. Thompson, Disparity analysis of images, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 2, pp333-340, 1980.

[Bes84] Besl, P. and R. Jain, Surface characterization for three-dimensional object recognition in depth maps, Tech. Rep. RSD-TR-20-84, Univ. of Mich., College of Engineering, 1984.

[Bes85] Besl, P. and R.C. Jain, Three-dimensional object recognition, Computing Surveys, Vol. 17, pp75-145, 1985.

[Boy79] Boyse, J.W., Data structures for a Solid Modeller, IEEE Workshop on the Representation of Three-Dimensional Objects, ppE1-E27, Philadelphia, 1979.

[Boy86] Boyter, B.A. and J.K. Aggarwal, Recognition of polyhedra from range data, IEEE Expert, Vol. 1, pp47-59, 1986.

[Bra84] Brady, M., Representing shape, 1st IEEE Conf. on Robotics, pp256-265, 1984

[Bra85] Brady, M., J. Ponce, A. Yuille, and H. Asada, Describing surfaces, MIT AI Memo 822, January 1985.

[Buc84] Buchanan, B.G. and E.H. Shortliffe, Rule-based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, Reading, Mass., 1984.

[Bur83] Burt, P.J., Fast algorithms for estimating local image properties, Computer Graphics and Image Processing, Vol. 21, pp368-382, 1983.

[Cha82] Chakravarty, I. and H. Freeman, Characteristic views as a basis for three-dimensional object recognition, Proc. of the Society for Photo-Optical Instrumentation Engineers Conference on Robot Vision, Vol. 336, pp37-45, 1982.

[Chi85] Chien, C.H. and J.K. Aggarwal, Reconstruction and matching of 3-D objects using quadtrees/octrees, Proceedings of the 3rd Workshop on Computer Vision: Representation and Control, Traverse City, Michigan, pp49-54, October 1985.

[Cog82] Coggins, J., A Framework for Texture Analysis Based on

Spatial Filtering, PhD Dissertation, Dept. of Computer Science, Michigan State University, 1982.

[Coh85] Cohen, Paul R., Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach, (Research Notes in Artificial Intelligence 2), Pitman Publishing Inc., Boston, 1985.

[Col79] Coleman, G.B., and H.C. Andrews, Image segmentation by clustering, Proc. IEEE, Vol. 67, pp773-785, 1979.

[Con78] Conway, J.B., Functions of One Complex Variable, 2nd edition, Springer-Verlag, New York, 1978.

[Con80] Conover, W.J., Practical Nonparametric Statistics, 2nd edition, John Wiley & Sons:New York, 1980.

[Dub76] Dubes, R., and A.K. Jain, Cluster techniques: the user's dilemma, Pattern Recognition, Vol. 8, pp247-260, 1976.

[Dub80] Dubes, R. and A.K. Jain, Clustering methodologies in exploratory data analysis, in Advances in Computers, (M.C. Yovits, ed.), pp113-228, Academic Press, New York, 1980.

[Dud79] Duda, R.O., D. Nitzan, P. Barrett, Use of range and reflectance data to find planar surface regions, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 1, pp259-271, 1979.

[Dud79b] Duda, R.O., J. Gaschnig, P. Hart, Model design in the Prospector consultant system for mineral exploration, in D. Michie (Ed.) Expert Systems in the Microelectronic Age, Edinburgh: Edinburgh University Press, pp153-167, 1979.

[Fau83] Faugeras, O.D., M. Hebert, E. Pauchon, Segmentation of range data into planar and quadratic patches, CVPR-83,

pp8-13, 1983.

[Fuk80] Fukada, Y., Spatial clustering procedures for region analysis, Pattern Recognition, Vol. 12, pp395-403, 1980.

[Gar79] Garey, M.R. and D.S. Johnson, Computers and Intractability, W.H. Freeman and Company, San Francisco, 1979.

[Gen77] Gennery, D., A stereo vision system for an autonomous vehicle, 5th IJCAI, pp576-582, 1977.

[Gil83] Gil, B., A. Mitiche, and J.K. Aggarwal, Experiments in combining intensity and range edge maps, Computer Vision, Graphics, Image Processing, Vol. 21, pp395-411, 1983.

[Gol78] Goldberg, M., and S. Shlien, A clustering scheme for multispectral images, IEEE Trans. Systems, Man, and Cybernetics, Vol. 8, pp86-92, 1978.

[Gow78] Gowda, K.C., and G. Krishna, Agglomerative clustering using the concept of mutual nearest neighborhood, Pattern Recognition, Vol. 10, pp105-112, 1978.

[Gri80] Grimson, W., A computer implementation of a theory of human stereo vision, MIT AI Memo No. 565, 1980.

[Gri84] Grimson, W. and T. Lozano-Perez, Model-based recognition and localization from tactile data, 1st IEEE Conf. on Robotics, pp248-255, 1984.

[Gu84] Gu, W.K. and T.S. Huang, Connected line drawing extraction from a perspective view of a polyhedron, 1st IEEE Conference on Artificial Intelligence Applications, pp192-198, 1984.

[Hal82] Hall, E.L., J.B.K. Tio, C.A. McPherson, F.A. Sadjadi,

Measuring curved surfaces for robot vision, Computer, Vol. 15, pp42-54, 1982.

[Har75] Haralick, R.M., and I. Dinstein, A spatial clustering procedure for multi-image data, IEEE Trans. on Circuits and Systems, CAS-22, pp440-450, 1975.

[Heb85] Hebert, M. and T. Kanade, The 3D-profile method for object recognition, CVPR-85, pp458-463, 1985.

[Hor84] Horaud, P. and R.C. Bolles, 3DPO's strategy for matching three dimensional objects in range data, 1st IEEE Conf. on Robotics, pp78-85, 1984.

[Hur84] Hurt, S.L. and A. Rosenfeld, Noise reduction in three-dimensional digital images, Pattern Recognition, Vol. 17, pp407-421, 1984.

[Ike81] Ikeuchi, K., Determining surface orientations of specular surfaces by using the photometric stereo method, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 3, pp661-669, 1981.

[Ino82] Inokuchi, S., T. Nita, F. Matsuda, Y. Sakurai, A three dimensional edge-region operator for range pictures, 6th ICPR, pp918-920, 1982.

[Ino84] Inokuchi, S., K. Sato, F. Matsuda, Range-imaging system for 3-D object recognition, 7th IJCPR, pp806-808, 1984.

[Itt85] Ittner, D.J., and A.K. Jain, 3-D surface discrimination from local curvature measures, CVPR-85, pp119-123, 1985.

[Jar76] Jarvis, R.A., Focus optimisation criteria for computer image processing, Microscope, Vol. 24, pp163-180, 1976.

[Jar83] Jarvis, R.A., A perspective on range finding techniques

for computer vision, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 5, pp122-137, 1983.

[Jul71] Julesz, B., Foundations of Cyclopean Perception, The University of Chicago Press, 1971.

[Kre82] Kreis, Th., H. Kreitlow, W. Jen, Detection of edges by video systems, Proc. 2nd Int'l Conf. on Robot Vision and Sensory Control, pp9-17, Stuttgart, Germany, 1982.

[Kro86] Krotkov, E. and J-P. Martin, Range from focus, International Conference on Robotics and Automation, pp1093-1098, San Francisco, April 1986.

[Kua84] Kuan, D.T., and R.J. Drazovich, Model-based interpretation of range imagery, AAAI-84, pp210-215, 1984.

[Lu85] Lu, S.W., A.K.C. Wong, M. Rioux, Recognition of 3-D objects in range images by attributed hypergraph monomorphism and synthesis, Int'l Federation of Automatic Control, Barcelona, pp389-394, November 1985.

[Mag85] Magee, M.J., B.A. Boyter, C-H. Chien, J.K. Aggarwal, Experiments in intensity guided range sensing recognition of three-dimensional objects, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 7, pp629-637, 1985.

[Mar79] Marr, D. and T. Poggio, A computational theory of human stereo vision, Proc. R. Soc. Lond., Vol. B-204, pp301-328, 1979.

[Mar80] Marr, D. and E. Hildreth, Theory of edge detection, Proc. R. Soc. Lond., Vol. B-207, pp187-217, 1980.

[Mar82] Marr, D., Vision, New York:W.H. Freeman and Company, 1982.

[Mar83] Martin, W.N. and J.K. Aggarwal, Volumetric description of objects from multiple views, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 5, pp150-158, 1985.

[Mil80] Milgram, D.L., and C.M. Bjorklund, Range image processing: planar surface extraction, 5th ICPR, pp912-919, 1980.

[Mit83] Mitiche, A. and J.K. Aggarwal, Detection of edges using range information, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 5, pp174-178, 1983.

[Mul84] Muller, Y., and R. Mohr, Planes and quadrics detection using Hough transform, 7th ICPR, pp1101-1103, 1984.

[Nar77] Narendra, P.M., and M. Goldberg, A non-parametric clustering scheme for LANDSAT, Pattern Recognition, Vol. 9, pp207-215, 1977.

[Nev76] Nevatia, R., Depth measurement by motion stereo, CGIP, Vol. 5, pp203-214, 1976.

[Nev77] Nevatia, R. and T.O. Binford, Description and recognition of curved objects, Artificial Intelligence, Vol. 8, pp77-98, 1977.

[Nim82] Nimrod, N., A. Margalita, H. Mergle, A laser based scanning rangefinder for robotic applications, Proc. 2nd Int'l Conf. on Robot Vision and Sensory Control pp241-252, Stuttgart, Germany, 1982.

[Ode80] Odenthal, J.P., A linear photodiode array employed in a short range laser triangulation, obstacle avoidance sensor, Master's thesis, Rensselaer Polytechnic Institute, Dept. of Computer Science, 1980.

[Osh79] Oshima, M. and Y. Shirai, A scene description method using three-dimensional information, Pattern Recognition, Vol. 11, pp9-17, 1979.

[Osh83] Oshima, M. and Y. Shirai, Object recognition using three-dimensional information, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 5, pp353-361, 1983.

[Pon85] Ponce, J. and M. Brady, Toward a surface primal sketch, 2nd IEEE Int'l Conf. on Robotics, pp420-425, 1985.

[Pos82] Posdamer, J.L. and M.D. Altshuler, Surface measurement by space-encoded projected beam systems, CGIP, Vol.18, p.1, 1982.

[Rad84] Radack, G.M. and N.I. Badler, Local matching of surfaces using critical points, Tech. Rep. MS-CIS-84-13, Dept. of Computer and Information Science, Univ. of Penn., 1984.

[Ree85] Reeves, A.P., R.J. Prokop, R.W. Taylor, Shape analysis of three dimensional objects using range information, CVPR-85, pp452-457, 1985.

[Req80] Requicha, A.A.G., Representations for rigid solids: theory, methods, and systems, Computing Surveys, Vol. 12, pp437-464, 1980.

[Req82] Requicha, A.A.G. and H.B. Voelcker, Solid modeling: a historical summary and contemporary assessment, IEEE Computer Graphics and Applications, Vol.2, pp9-24, 1982.

[Ros78] Rosenberg, D., M.D. Levine, S.W. Zucker, Computing relative depth relationships from occlusion cues, Proc. 4th Int. Joint Conf. on Pattern Recognition, Kyoto, Japan, pp765-769, 1978.

[Sch79] Schachter, B.J., L.S. Davis, A. Rosenfeld, Some experiments in image segmentation by clustering of local feature values, Pattern Recognition, Vol. 11, pp19-28, 1979.

[Sha76] Shafer, Glenn, A Mathematical Theory of Evidence, Princeton University Press, 1976.

[Sha82] Shapiro, L.G. and R.M. Haralick, Organization of relational models for scene analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 4 pp595-602, 1982.

[Sho75] Shortliffe, E.H. and B.G. Buchanan, A model of inexact reasoning in medicine, Mathematical Biosciences, Vol. 23, pp351-379, 1975.

[Sto84] Stockman, G. and J.C. Esteva, Use of geometric constraints and clustering to determine 3D object pose, 7th ICPR, pp742-744, Montreal, Canada, 1984.

[Sto85] Stockman, G. and G. Hu, 3-D surface sensing using a projected grid, Tech. Rep. MSU-ENGR-85-024, Michigan State University, Dept. of Computer Science, 1985.

[Sve85] Svetkoff, D.J., P.F. Leonard, R.E. Sampson, R. Jain, Techniques for real-time, 3D, feature extraction using range information, Tech. Rep. 521-43, Environmental Research Institute of Michigan, Ann Arbor, Michigan, 1985.

[Ter83] Terzopoulos, D., The role of constraints and discontinuities in visible-surface reconstruction, Proc. 7th IJCAI, Karlsruehe, pp1073-1077, 1983.

[Tom84] Tomita, F. and T. Kanade, A 3D vision system generating and matching shape descriptions in range images, IEEE 1st

Conference on Artificial Intelligence Applications, pp186-191, 1984.

[Wan84] Wang, Y.F., M.J. Magee, J.K. Aggarwal, Matching three-dimensional objects using silhouettes, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 6, pp513-518, 1984.

[Wil80] Williams, T., Depth from camera motion in a real world scene, IEEE Transactions on Pattern Analysis and Machine IntelligenceVol. 2 , pp511-516, 1980.

[Yor81] York, B.W., A.R. Hanson, E.M. Riseman, 3D object representation and matching with B-splines and surface patches, Proc. 7th Int. Joint Conf. on Artificial Intelligence, pp648-651, 1981.

[Zah71] Zahn, C.T., Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans. Comp., Vol. C-20, pp68-86, 1971.