RSD-TR-19-84

AN OVERVIEW OF THREE-DIMENSIONAL OBJECT RECOGNITION[1]

J.

Paul Besl

Ramesh Jain

Department of Electrical Engineering and Computer Science

The University of Michigan

Ann Arbor, Michigan 48109-1109

December 1984

CENTER FOR RESEARCH ON INTEGRATED MANUFACTURING

Robot Systems Division

COLLEGE OF ENGINEERING

THE UNIVERSITY OF MICHIGAN

ANN ARBOR, MICHIGAN 48109-1109

[1] This work was supported in part by IBM Corporation, Data Systems Division, Kingston, New York under the monitorship of Dr. Jack Contino and in part by the Air Force Office of Scientific Research under contract F49620-82-C-0089.

TABLE OF CONTENTS

Abstract

A general purpose computer vision system must be capable of recognizing three-dimensional (3-D) objects. This paper proposes a precise definition of the 3-D object recognition problem and discusses general concepts associated with this problem. The relevant literature is then reviewed and summarized. Since depth maps, or range images, are often considered as input rather than intensity images, techniques for obtaining, processing, and characterizing range data are also surveyed.

Index Terms: 3-D object recognition, computer vision, depth maps, 3-D object representation, 3-D object reconstruction, surface characterization, surface matching

1. Introduction

Vision is a complex perceptual process which involves the physical elements of illumination, geometry, reflectivity, and image formation as well as the intelligence aspects of recognition and understanding. Human beings have little trouble understanding stationary or moving, color or black-and-white three-dimensional scenes in the real world, in movies and television, or in photographs. The ultimate goal of computer vision researchers is to endow computers with human-like visual capabilities so that machines can sense the environment in their field of view, understand what is being sensed, and take appropriate actions as programmed. Object recognition is critical to understanding sensor data.

Most of the computer vision research performed during the last twenty years has concentrated on using digitized gray-scale intensity images as sensor data. It has been extremely difficult to program computers to understand and describe these images in a general purpose way. One particular problem is that digitized intensity images are rectangular arrays of numbers which indicate only the *brightness* at individual points on a regularly spaced rectangular grid and contain no *explicit* information which is relevant to *depth perception.* Yet human beings are able to correctly infer depth relationships quickly and easily among image regions in such images whereas automatic inference of such relationships has proven to be quite difficult. In recent years digitized range data has become available from both active and passive sensors, and the quality of this data has been steadily improving. Range data is usually produced in the form of a rectangular array of numbers, referred to as a *depth map or range image*, where the numbers quantify the distances from the sensor plane to the surfaces within the field of view along parallel rays on a regularly spaced rectangular grid. Not only are depth relationships between depth map regions explicit, the three-dimensional *shape* of depth map regions approximates the three-dimensional shape of the corresponding object *surfaces* in the field of view. Therefore, the process of recognizing objects by their shape *should* be less difficult in depth maps than in intensity images due to the explicitness of the information. For example, since correct depth map information depends only on geometry and is independent of illumination and reflectivity, intensity image problems with shadows and surface markings do not occur. In our literature review, object recognition papers are categorized according to whether or not the use of range data is discussed.

This paper closely examines the three-dimensional object recognition problem. An outline of the covered material is given below:

(1) Autonomous single arbitrary view three-dimensional object recognition is defined as a worthwhile goal which computer vision systems might achieve.

(2) The necessary components for an object recognition system are discussed from a general qualitative point of view. The characteristics of an ideal

system for solving the particular well-defined object recognition problem stated in (1) are proposed.

(3) The existing literature and subject matter relevant to this problem are reviewed. The 3-D object recognition systems presented by different authors are discussed with respect to the general purpose goals set forth in (1). Techniques for obtaining and processing range data are also surveyed since these methods are fairly new compared to the corresponding techniques for intensity image data.

2. Problem Definition

Three-dimensional object recognition is a somewhat nebulous term. A brief survey of the literature on this subject is sufficient proof of this statement [30] [57] [73] [113] [129] [130] [150]. Therefore, we first attempt to give a reasonably precise definition to the object recognition problem. The problem which we present is more general and more useful than many other recognition problems addressed in the literature, and yet it should still be solvable. First, we present a brief qualitative summary concerning desirable visual capabilities which motivates the detailed definition that follows.

The *real world* which humans commonly perceive both visually and tactilely is primarily composed of solid *objects*. When humans are given a new object they have never seen before, they are typically able to gather information about that object from many different *viewpoints*. The process of gathering detailed object information and then storing that information in some format is referred to as *model formation*. Once human beings are familiar with many objects, they can identify those objects from an *arbitrary single stationary viewpoint* without further investigation in most cases. In particular, humans can *identify, locate, and qualitatively describe the orientation* of objects in black-and-white photographs. The black-and-white photograph capability is significant because only the spatial variation of a *single* parameter within a framed rectangular region corresponding to a fixed single view of the real world is involved whereas human color vision generally involves a three-parameter color variation within a large, almost hemispherical solid angle corresponding to a continually changing viewpoint. Since we are interested in an automatic, computerized recognition process, we must restrict allowable input data to be compatible with digital computers. The term *digitized sensor data* will be used to refer to any input matrix of numerical values (which can represent intensity, range, or some other scalar parameter) and associated auxiliary information concerning how the matrix of values was obtained.

The above paragraph motivates the following definition of the autonomous single arbitrary view three-dimensional object recognition problem:

(1) Given any labeled rigid solid object, that *object may be examined* in any way desired as long as the object is not deformed.

(2) A *model* for the labeled object *may be formed* using information from this examination in any way desired and given that object's label.

(3) (i) Given digitized sensor data corresponding to one particular, but arbitrary, field of view of the real world as it existed at the time of data acquisition;
(ii) given any data stored previously during the model formation process; and
(iii) given a list of labels of distinguishable solid objects; answer the following questions for each object in the list using the capabilities of a single autonomous processing unit:

a) Does the given labeled object appear in the digitized sensor data?

b) If it does, how many times does it occur?

c) For each occurrence of a given object, determine the location of that object within the sensor data and, if it is possible using the particular type of sensor data, determine the three-dimensional location of that object with respect to some convenient coordinate system.

d) Also, if possible, determine the three-dimensional orientation of each occurrence of a given object with respect to some convenient coordinate system.

(4) Finally, if there exist regions within the sensor data which do not correspond to any of the objects in the list, characterize these regions in a way that they might be recognized if they occur again in any future images.

We will refer to the problem of successfully completing these assigned tasks using real world sensor data while obeying the given constraints as the 3-D object recognition problem. This problem is *not* successfully addressed in many of the object recognition systems discussed in the literature review; more constrained problems are usually addressed which are limited to particular applications. If our stated 3-D object recognition problem is solved successfully by some system, that system would be extremely useful in a wide variety of applications, including automatic inspection and assembly and autonomous vehicle navigation. The problem is posed so that it is feasible to use computers to solve the problem, and it is also clearly solvable by human beings.

How do we know whether or not a particular approach solves the problem and how can we compare different approaches to see if one is better than another? The performance of object recognition systems could be measured using the number of errors made by a system in performing the assigned problem tasks on particular standardized sets of digitized sensor data which challenge the capabilities mentioned in the problem definition. The following list enumerates *some* of the possible types of errors that can be committed by such systems:

(1) Miss error: The presence of an object is not detected when it is definitely present,

(2) False alarm error: The presence of an object is indicated when it is not really there,

(3) Count error: The non-zero number of occurrences of a particular object may be wrong,

(4) Location error: Object occurrences may be identified correctly, but the location of the object in the data may be wrong,

(5) Orientation error: The object occurrences and positions may be determined correctly, but the orientation may be wrong.

Different object recognition systems could be compared and the term "successful" could be made quantitative by establishing a performance index which quantitatively combines the number, the type, and the magnitude of the various errors. Currently published information make it practically impossible to quantitatively compare existing object recognition systems since different researchers do not evaluate their systems in any consistent manner. Hence, subsequent comparisons in the literature review will be subjective and qualitative.

3. General Object Recognition System Concepts

The common components of all object recognition systems are discussed below. By identifying these components, comparisons can be made between different systems by comparing and contrasting these components. For instance, two systems might be equally general with respect to their object models but may differ significantly in their sensor-data processing or matching algorithms. Characteristics of the ideal system are proposed.

3.1. Object Recognition System Components

The specific tasks to be performed by an object recognition system are given in the problem definition above. We also suggest that we could measure how well these tasks are performed. But how can these tasks be accomplished?

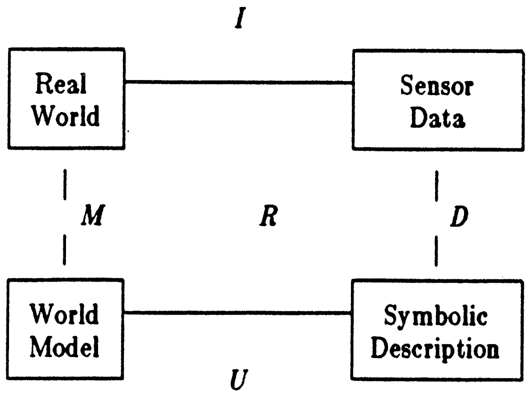
First, how can one recognize something unless one knows what one is looking for? Therefore, even though model formation is not specifically required by our problem definition, it is practically demanded by the circumstances. The literature survey reinforces this basic idea since all known investigators in this subject area have utilized some sort of modeling process. Many different types of models, both view-independent and view-dependent, have been used for modeling real world objects for recognition purposes. We will survey different view-independent object representations because representation is such a critical factor in object recognition system design. We will not survey *view-dependent* techniques as a separate topic because these types of representations are clearly not advisable for single *arbitrary* view recognition. For now, we shall assume that the necessity of model formation has been established and that some representation is required to store object model data. Thus, a *world model* of some sort is a necessary object recognition system component.

The next issue can be stated as follows: Once one knows what one is looking for, how can one go about finding it in the digitized sensor data? In other words, a method for matching the model data to the sensor data must be considered to determine how recognition will take place. A simple-minded straightforward blind search approach would entail transforming all possible combinations of all possible known object models in all possible distinguishable orientations and all possible distinguishable locations into the digitized sensor data format and computing a matching error quantity to be minimized. The minimum matching error configuration of object models would correspond to the recognized scene. All the tasks mentioned in the statement of the object recognition problem would be accomplished except possibly the characterization of unknown regions not corresponding to known objects. Of course, this would take an extremely large amount of processing time even for the simplest scenes. Therefore, a better algorithm is required. Note that since the world model will usually contain more object information than the sensor data, we are usually prohibited from transforming sensor data into *complete* model data and matching in the model data format. (However, this does not prevent one from matching with *partial* model data.) As a result of these problems and the natural desire to reduce the large dimensionality of the input sensor data, it is often advantageous to work with an intermediate domain which is computable from both the sensor data and the model data. For lack of a better term, we will refer to this domain as the ***symbolic scene description domain.*** In the literature review, it is seen that sensor data is usually processed until it reaches some form of symbolic scene description. The model data can also be transformed into an equivalent symbolic scene description. Some sort of matching procedure can then be carried out on the quantities in this intermediate domain, which are often referred to as features. The best matching results should occur when the hypothetical object model configuration accurately represents the real world scene represented in the sensor data. Thus, we propose that a matching procedure and intermediate symbolic scene description mechanisms are necessary object recognition system components.

The interaction of the individual object recognition system components is diagrammed in Figure 1. The key domains are the real world, the digitized sensor data domain, the modeling domain, and the intermediate symbolic scene description domain. The key functions are the image formation function which might create intensity or range data or both, the description function which acts on the sensor data and extracts features, the modeling function which provides object models for real world objects, the understanding, or recognition, function which involves some sort of matching algorithm, and the rendering function which can produce synthetic sensor data. It is proposed that any object recognition system can be discussed within the framework of this system model. This description is basically in agreement with the ideas brought forth by Brooks [29], except for the rendering function. The rendering function is an important feedback link because it can allow an autonomous system to check on its own understanding of the sensor data.

3.2. Characteristics of Ideal Object Recognition System

What are the characteristics of the ideal system which handles the object recognition problem as we have defined it? Of course, one would like it to



I = Image Formation Function

M = World Modeling Function

D = Description Function

U = Understanding Mapping

R = Model Rendering Function

Figure 1. General Object Recognition System Format

have human-like performance, but we can be more specific than that. Below, we summarize some of the capabilities that might be realized by object recognition systems in the near future:

(1) It must be able to handle arbitrary viewing directions for the sensor data without preference to horizontal, vertical, or any other directions. This implies the usage of a good view-independent modeling scheme within such a system which is compatible with recognition processing requirements.

(2) It must handle arbitrarily complicated real world objects without preference to either curved or planar surfaces.

(3) It must handle arbitrary combinations of a relatively large number of objects in arbitrary orientations and locations without being sensitive to superfluous occlusions (i.e., if occlusion does not affect human understanding of a scene, then it should not affect the ideal automated object recognition system either).

(4) It must be able to handle a certain amount of noise in the sensor data without a significant degradation in system performance.

(5) It should be able to analyze scenes quickly and correctly.

(6) It should not be too difficult to modify the world model data to handle new situations and new objects.

(7) It is desirable for the system to be able to express its confidence in its own understanding of the sensor data.

4. Literature Review

The existing literature and subject matter relevant to the three-dimensional object recognition problem is now reviewed within the framework established by the previous section. Individual works will be considered within the context of a particular general category. It is difficult to establish a natural order to these general categories; therefore, we have chosen one convenient ordering for our discussion. The topics which we consider relevant to the object recognition problem are the following:

(1) 3-D Object Representation Schemes

(2) 3-D Surface Representation Schemes

(3) 3-D Object and Surface Rendering Algorithms

(4) Intensity and Range Image Formation

(5) Intensity and Range Image Processing

(6) 3-D Object Reconstruction Algorithms

(7) 3-D Surface Characterization

(8) 3-D Object Recognition Systems using Intensity Images

(9) 3-D Object Recognition Systems using Range Images

The treatment of some of the above topics is necessarily very brief. They were included to put the main discussion contained in the last three topic areas into perspective. We have attempted to collect all published (English language) works in these last three areas (7-9); we apologize to any authors inadvertently omitted.

This paper is not intended as an overview of computer vision. Gevarter [53] presents a short easily understandable introduction and summarizes the state of the art. The book by Ballard and Brown [7] is a reasonable place to start for the more serious reader who is not already familiar with the field. There are also several overview papers which survey computer vision and treat 3-D issues but consider *only intensity images* as input [5] [11] [27] [28] [127]. Binford [18] has written a survey of model-based intensity-image analysis systems. Dyer and Chin [43] have reviewed 2-D and 3-D object recognition literature with an emphasis on industrial systems.

4.1. 3-D Object Representation Schemes

In order to recognize a particular object, you need to know what that object "looks like." How can a computer "understand" what an object looks like? Computers can understand 3-D object structure and appearance through use of object models which are independent of viewer position and orientation. How can such a model be stored in a computer? There are many different answers to this question, and each different answer gives rise to a different object representation scheme.

We now briefly review the basic categories of 3-D object representation. This overview will make it easier to understand the limitations of some of the object recognition systems reviewed later. First, we discuss object representations used primarily by systems where the goal is to create realistic digital images from models (i.e., existing computer *graphics* representations). Then we look at other representations used by systems where the goal is to understand digital images using models (i.e., existing computer *vision* representations). The two types of systems mentioned here perform basically opposite operations. Both systems need the same kind of object information, but the utilization of that information is quite different.

The 3-D object representations commonly used by contemporary computer-aided-design (CAD) geometric solid object modeling systems can usually be categorized as one of the following:

1) **Wireframe Representation:**
The wireframe representation of a three-dimensional object usually consists of a 3-D vertex point list and an edge list of vertex pairs, or can be formatted as such. This representation is quite common because it is so simple. However, it can also be an ambiguous representation for determining quantities such as the surface area and volume of an object. See Figure 2 for the block-within-a-block example of this ambiguity. The single wireframe model can be interpreted as three different solid objects.

2) **Constructive Solid Geometry Representation (CSG):**
The CSG representation of an object is specified in terms of a set of 3-D volumetric primitives (blocks, cylinders, cones, and spheres are typical examples of bounded primitives) and a set of Boolean operators: union, intersection, and difference. See Figure 3 for an example of a CSG description of an object. The storage data structure is a binary tree where the terminal nodes of the tree are instances of primitives and the branching nodes represent Boolean set operations. CSG trees define object volume and surface area unambiguously and are capable of representing complex objects with a very small amount of data. However, free form sculptured surfaces, such as the head surface shown in Figure 4, are not easily represented using CSG modelers. A general purpose modeling system should be able to

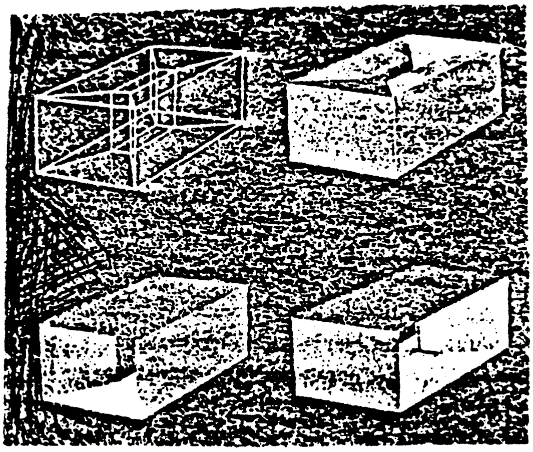


Figure 2. Wireframe Ambiguity (from [123])

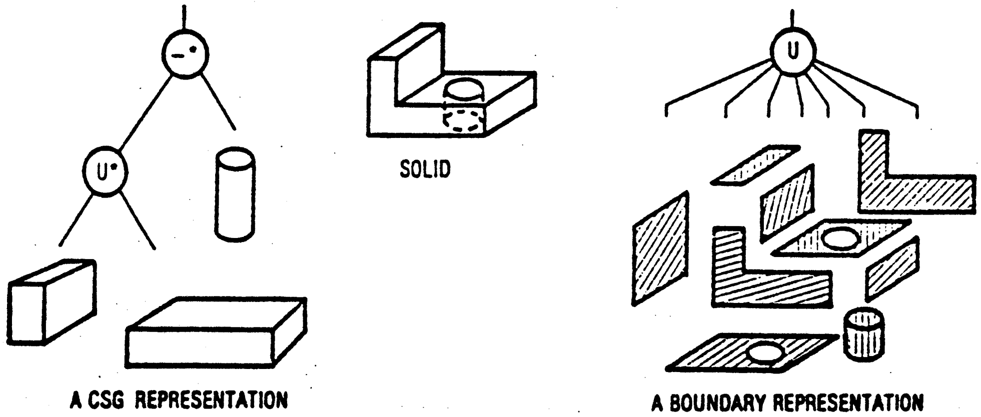


Figure 3. CSG and Surface Boundary Representations for Solid (from [122])

represent such surfaces.

3) **Spatial Occupancy Representation:**

Spatial occupancy representations define *non-overlapping* regions of 3-D space occupied by a particular object and unambiguously define an object's volume. The following single primitive representations of this type are commonly used:

1) Voxel Representation:

Voxels are small volume elements of discretized 3-D space which are usually fixed-size cubes. Objects are represented by the list of voxels occupied by the object. This

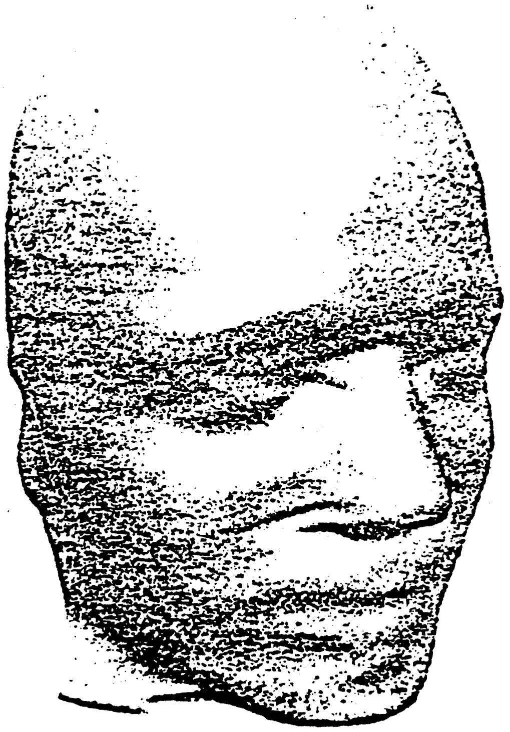


Figure 4. Intensity Image of Head Surface (from [142])

representation tends to be memory intensive.

2) Oct-tree Representation: [98]

An oct-tree is a hierarchical representation of spatial occupancy. Volumes are decomposed into cubes of different sizes where the cube size depends on the distance from the root node. Each branching node of the tree structure represents a cube and points to eight other nodes which describe object volume occupancy in the corresponding octant cube of the branching node cube. This representation offers the advantages of the voxel description but is much more compact. The basic idea of oct-trees is displayed by considering the 2-D analog of oct-trees, usually referred to as quad-trees, as shown in Figure 5.

3) Tetrahedral Cell Decomposition Representation.

Decomposition of 3-D space regions into tetrahedral elements is very similar to the lower-dimensional analog of decomposing flat surfaces into triangles. (The tetrahedron is a 3-simplex where as the triangle is a 2-simplex.) Tetrahedral decompositions define volume and surface area and are

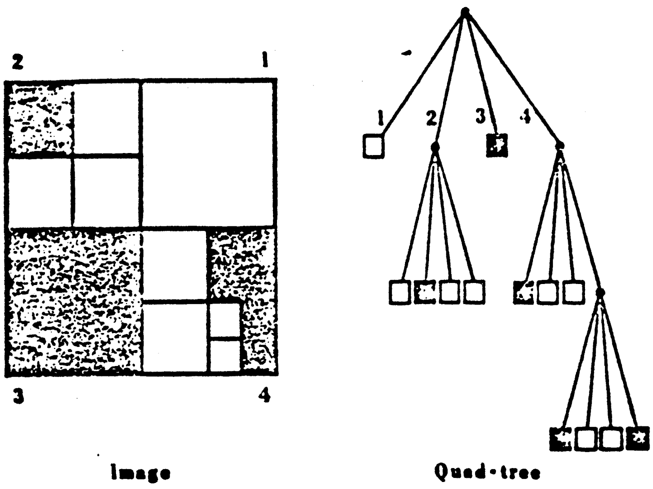


Figure 5. Hierachical Approach to Spatial Occupancy (from [62])

useful for many mathematical purposes.

Voxels and oct-trees are useful for a number of computer graphics applications whereas tetrahedral models are often useful for finite element applications. Many other spatial occupancy schemes are possible.

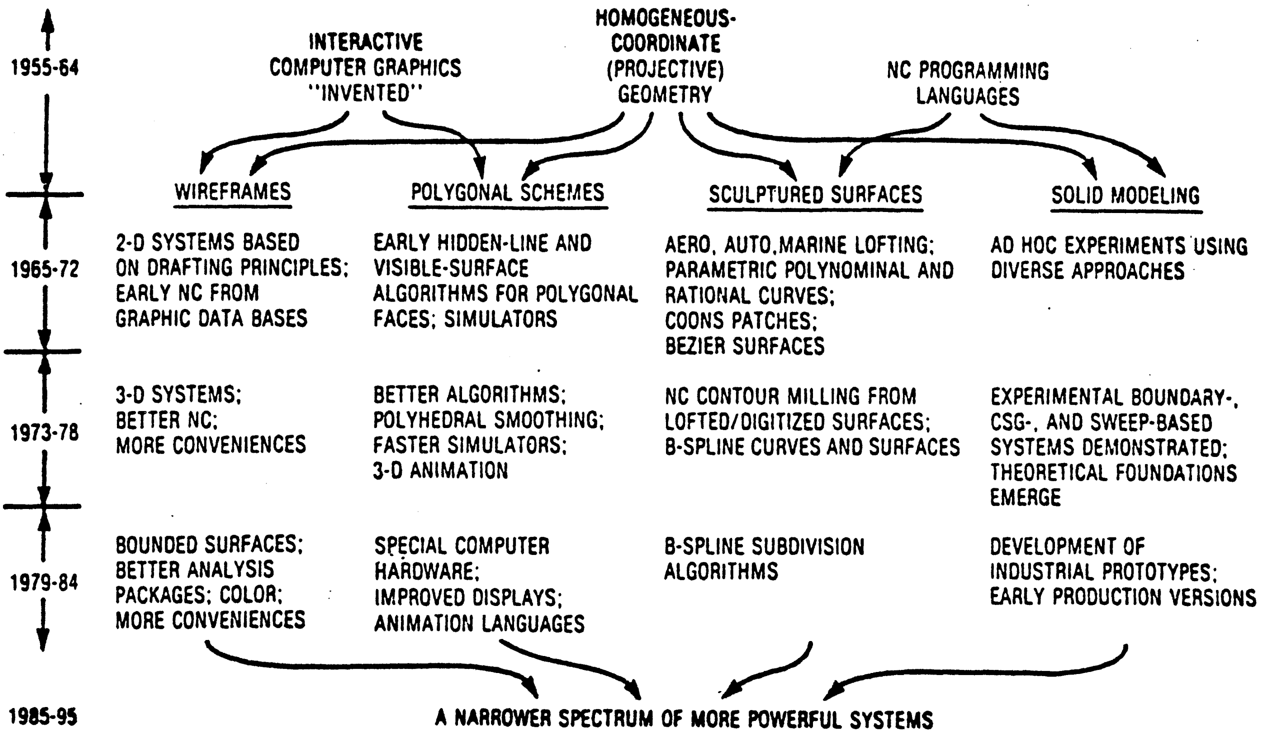
4) **Surface Boundary Representation (B-Rep):**

Surface boundary representations define a solid object by defining the three-dimensional surfaces which bound that object. Figure 3 shows an example of the boundary representation concept. The simplest boundary representation is the triangle-faced polyhedron which can be stored as a list of 3-D triangles. Arbitrary surfaces can be approximated to any desired degree of accuracy by utilizing many faces. The head surface image shown in Figure 4 was generated from a list of triangles and quadrilaterals (the union of two triangles) which describe the surface. A slightly more compact representation allows the replacement of adjacent, connected, coplanar triangles with arbitrary n-sided planar polygons. This sort of representation is popular because model surface area and volume are well-defined and all object operations can be carried out using piecewise-planar algorithms. The next usual incremental step in generality is obtained using quadric surface based boundary representations. Many other techniques for representing surfaces with higher order polynomials or splines exist and are discussed in the next section. For surfaces more complicated than planar polygons, some data structure or algorithm must be used to determine where surface intersections occur. That is, there must be some mechanism for bounding the extent of individual surface patches which bound an object. In addition, there might also exist mechanisms to describe the structural relationships between bounding surfaces. There are many ways to do this; Lin and Fu [90] have even proposed a syntactic approach which uses a context-free 3-D plex grammar for this purpose. The common element of surface boundary representations is a list of the bounding object surfaces.

Any representation which adequately represents a real world object should be usable by some type of graphics algorithm to render synthetic sensor data images very similar to real world sensor data. The above object modeling schemes have been used successfully to provide very realistic shaded image renderings of real world scenes. On the other hand, any representation chosen for computer vision object recognition should also be suitable for matching algorithm purposes. It is desirable for each real world object shape to have a unique description within the framework of a given representation to be suitable for matching. It turns out that several of the representations above do not yield unique numerical descriptions of object shapes. That is, it is often possible to reorder or reorganize points, edges, faces, and/or primitives of a given representation to obtain an identical shape. Model-based matching algorithms for computer vision systems must be made insensitive to this non-uniqueness if the modeling scheme suffers from this problem. Much more can be said about the above representation schemes, but a full discussion of the details and the relative merits of these representations is not the intention of this paper. We refer the reader to [4] [33] [122] [123] [124] for more details. For convenience, we include two self-contained figures from [122]: Figure 6 summarizes the history of approaches to 3-D object representation, and Figure 7 is a table of commercially available solid modelers.

The 3-D object representations mentioned in the computer vision literature can usually be categorized as one of the above schemes or as one of the following:

1) **Generalized Cone, Generalized Cylinder, or Sweep Representation:** Generalized cones or generalized cylinders are often called sweep representations because object shape is represented by a space curve which acts as the spine or axis of the cone, a two-dimensional cross-sectional figure, and a sweeping rule which defines how the cross-section is to be swept and possibly modified along the space curve. These ideas are shown for a simple cylinder in Figure 8. This representation is well-suited to many real world shapes. However, it becomes just about impossible to use this representation for objects that are not suited to this sort of description; consider the body of an automobile or the head image in Figure 4. Therefore, this scheme by itself is not general purpose. Despite this difficulty, many papers prefer the generalized cone object representation for vision purposes



INTERACTIVE COMPUTER GRAPHICS "INVENTED"
HOMOGENEOUS-COORDINATE (PROJECTIVE) GEOMETRY
NC PROGRAMMING LANGUAGES
1955-64
1985-95
A NARROWER SPECTRUM OF MORE POWERFUL SYSTEMS

	WIREFRAMES	POLYGONAL SCHEMES	SCULPTURED SURFACES	SOLID MODELING
1965-72	2-D SYSTEMS BASED ON DRAFTING PRINCIPLES; EARLY NC FROM GRAPHIC DATA BASES	EARLY HIDDEN-LINE AND VISIBLE-SURFACE ALGORITHMS FOR POLYGONAL FACES; SIMULATORS	AERO, AUTO,MARINE LOFTING; PARAMETRIC POLYNOMINAL AND RATIONAL CURVES; COONS PATCHES; BEZIER SURFACES	AD HOC EXPERIMENTS USING DIVERSE APPROACHES
1973-78	3-D SYSTEMS; BETTER NC; MORE CONVENIENCES	BETTER ALGORITHMS; POLYHEDRAL SMOOTHING; FASTER SIMULATORS; 3-D ANIMATION	NC CONTOUR MILLING FROM LOFTED/DIGITIZED SURFACES; B-SPLINE CURVES AND SURFACES	EXPERIMENTAL BOUNDARY-, CSG-, AND SWEEP-BASED SYSTEMS DEMONSTRATED; THEORETICAL FOUNDATIONS EMERGE
1979-84	BOUNDED SURFACES; BETTER ANALYSIS PACKAGES; COLOR; MORE CONVENIENCES	SPECIAL COMPUTER HARDWARE; IMPROVED DISPLAYS; ANIMATION LANGUAGES	B-SPLINE SUBDIVISION ALGORITHMS	DEVELOPMENT OF INDUSTRIAL PROTOTYPES; EARLY PRODUCTION VERSIONS

Figure 6. Historical Summary of Approaches to Object Representation (from [122])

MODELER	VENDOR/DISTRIBUTOR	CORE SOFTWARE	GENRE
CATIA	IBM	DASSAULT (FRANCE)	B-REP
CATSOFT	CATRONIX		CSG
DDM-SOLIDS	CALMA		B-REP
EUCLID	MATRA DATAVISION/ DEC	CNRS (FRANCE)	B-REP
GEOMOD-II	SDRC/ GENERAL ELECTRIC CAE	SDRC	B-REP
ICEM SOLID MODELLING	CDC	SYNTHAVISION (MAGI)	CSG
ICM GMS	ICM		B-REP
MEDUSA	PRIME	CIS/CV (UK)	B-REP
PADL-1,2	U. ROCHESTER		CSG
PATRAN-G	PDA ENGINEERING		CELL DECOMP.
ROMULUS	EVANS & SUTHERLAND	SHAPEDATA (UK)	B-REP
SOLIDESIGN	COMPUTERVISION		B-REP
SOLIDS MODELING-II	APPLICON	SYNTHAVISION (MAGI)	CSG
SYNTHAVISION	MAGI		CSG
TIPS-1	CAM-I	HOKKAIDO U.	CSG
UNIS-CAD	SPERRY UNIVAC	BAUSTEIN GEOMETRIE (T. U. BERLIN)	B-REP
UNISOLIDS	MCAUTO	PADL-2 (U. ROCHESTER)	CSG

Figure 7. Solid Modelers Available in US in May 1983 (from [122])

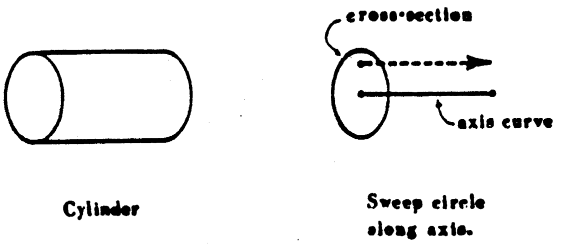


Figure 8. Generalized Cylinder Concept (from [62])

[2] [31] [83] [105] [106] [108].

2) **Multiple 2-D Projection Views Representation:**
For particular applications, it is convenient to store a library of two-dimensional silhouette projections to represent three-dimensional objects. For recognition of 3-D objects with a small number of stable orientations on a flat light table, this representation is ideal if silhouettes of different objects are different enough. This technique has also been used to recognize aircraft against the well-lit sky background in any orientation [150]. It is not a general purpose technique, however, for it is possible for many different 3-D object shapes to possess the same set of silhouette projections. A more detailed approach of a similar nature is the *characteristic views* technique used in [35]. All of the infinite 2-D projection views of an object are grouped into topological equivalence classes of which there are a finite number. Different views within an equivalence class are related by a linear transformations. This representation is general purpose in nature since it specifies the 3-D structure of an object even though it may turn out to be a very verbose form. Figure 9 shows several representative characteristic views for a particular non-convex polyhedron. Similar ideas are presented in a different form in [81]. Aspect is defined to be the topological structure of singularities [82] in a single view of an object. For almost any vantage point, small movements do not affect aspect. When a change in aspect occurs, this is referred to as an event. An object can be described by a graph, referred to as the *visual potential*, where the aspects form the nodes of the graph and events form the arcs of the graph. One can then measure visual object complexity by using a measure for the diameter of the visual potential, or aspect graph. Figure 10 shows the visual potential for a tetrahedron; in this case, there are three types of aspect: one, two, or three faces are visible. Scott [132] has looked at these ideas with the aim of implementing a graphics system which understands what it is displaying in terms of the projection topology and the visual potential neighborhood of a given view.

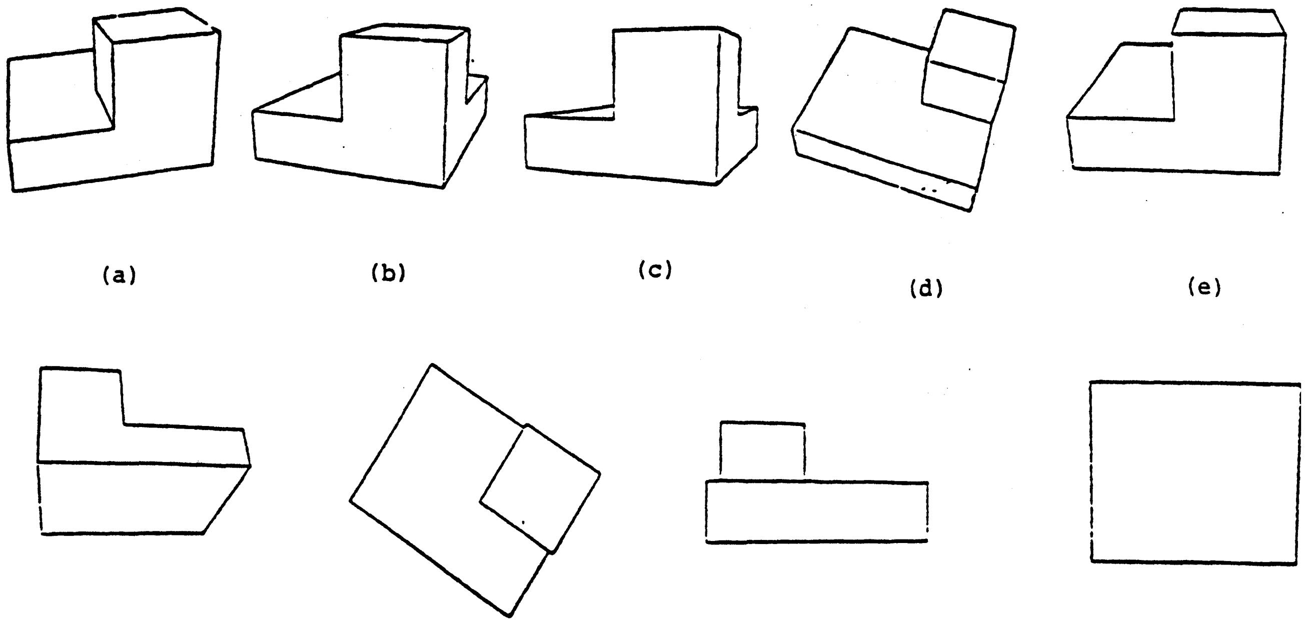


Figure 9. Representative Characteristic Views for Polyhedron (from [35])

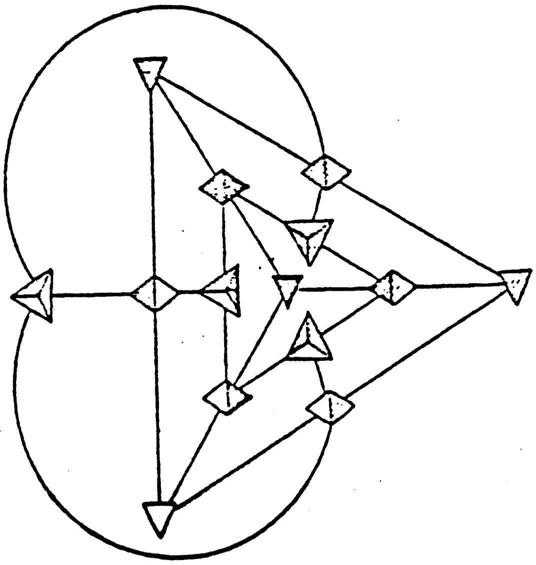


Figure 10. Visual Potential of Tetrahedron (from [81])

3) **Skeleton (Stick Figure) Representation:**

Some researchers have found it convenient to describe shape using skeleton models [146]. This representation is very ambiguous and is not suitable for general purpose 3-D object shape description. Skeletons are similar to the generalized cone descriptions but are more abstract. They can be viewed as a subset of the generalized cone information; skeletons consist only of the spines, or axis curves, of generalized cones.

4) **Generalized Blob Representation:**
Generalized blobs have been used by some investigators as a 3-D object shape description scheme [103]. Objects are described by sticks, plates, and blobs. This representation is certainly not detailed enough for general purposes.

5) **Spherical Harmonic Representation:**
Some objects can be represented by specifying the radius from a point as a function of latitude and longitude angles about that point. This representation may be useful in very restricted situations, but it is not general purpose.

6) **Overlapping Sphere Representation:**
O'Rourke and Badler [112] have proposed the use of *overlapping spheres* as a solid object representation. Many spheres are required to yield relatively smooth surfaces. This single primitive representation seems better suited to molecular display algorithms than object modeling. Although it is a general purpose technique, it is rather awkward for precisely representing the majority of man-made objects.

Most of the object representation methods discussed in the computer vision literature are not capable of describing arbitrary solid objects. These methods specialize in particular shapes or are symbolic or abstract in nature. They may be suitable to particular applications, but they are not general purpose.

All of the above categories have been listed to put the object representation methods prevalent in the literature into perspective. Each method has its own advantages and disadvantages with respect to different applications, but some are inherently more limited than others. The stated object recognition problem requires a representation which can model arbitrarily complicated objects with fine detail. Note that each of the above representations could allow object models to be built by the human designer or by automatic methods if possible.

4.2. 3-D Surface Representation Schemes

In the previous section, we discussed surface boundary representations of three-dimensional objects. Since surfaces of objects become sampled depth map surface regions under the depth map projection, we feel that surface boundary representations will be extremely important to object recognition in depth maps. These representations all employ lists of surfaces. But how are these surfaces represented? We saw previously that polyhedral models can be represented as a list of planar polygons so we will not discuss this surface representation further. How can smooth surfaces be represented?

A general surface in three dimensions can be written as

$$S = \left\{ (x,y,z) : F(x,y,z) = 0 \right\}.$$

This is referred to as an *implicit* representation of a surface. If the gradient vector ∇F exists, is continuous, and is non-zero for every point (x,y,z) , then S is a smooth surface. The implicit surface representation is very useful for *low order* polynomials of the spatial variables. Planar surfaces are precisely represented with only four coefficients (which describe three degrees of freedom):

$$F_{plane}(x,y,z) = Ax + By + Cz + D$$

(A,B,C) specify the direction of the single normal to the surface whereas D specifies the distance of the plane from the origin of the coordinate system if A,B,C are properly normalized. Quadric surfaces require ten coefficients in general (which describe nine degrees of freedom):

$$F_{quadric}(x,y,z) = Ax^2 + By^2 + Cz^2 + Gxy + Hyz + Izx + Ux + Vy + Wz + D$$

Only three coefficients are needed to describe the *shape* of a quadric surface of a given type whereas six parameters are needed to locate and orient the surface in space. If a quadric surface is properly translated and rotated, at least six of the ten coefficients will be zero. All quadric surfaces can be then classified as one of the six following types using the three or four *non-zero* coefficients in that particular coordinate system:

(1) Ellipsoid $(A>0,B>0,C>0,D=-1)$,
(2) Elliptic Paraboloid $(A>0,B>0,W=-1)$,
(3) Hyperbolic Paraboloid $(A>0,B<0,W=-1)$,
(4) Hyperboloid of One Sheet $(A>0,B>0,C<0,D=-1)$,
(5) Hyperboloid of Two Sheets $(A>0,B<0,C<0,D=-1)$,
(6) Quadric Cone $(A>0,B>0,C<0)$.

Unfortunately, the implicit surface representation is not generally useful for arbitrary surface descriptions unless surfaces are decomposed into locally homogeneous patches. The reason is that it becomes more and more difficult to deal with polynomial surface functions as the order of the polynomial increases.

Imagine trying to fit a polynomial surface of some order to the surface shape in Figure 4.

The standard alternative approach is to use an *explicit* parametric surface representation:

$$S = \left\{ (x,y,z) : x = h(u,v),\ y = g(u,v),\ z = f(u,v),\ (u,v) \in D \subseteq \mathbf{R}^2 \right\}.$$

where f,g,h are smooth scalar functions of two variables. A significantly less general, but still very useful parametric description of surfaces is given via the Monge patch representation:

$$S = \left\{ (x,y,z) : x = u,\ y = v,\ z = f(u,v),\ (u,v) \in D \subseteq \mathbf{R}^2 \right\}.$$

Gray level surfaces in intensity images and depth map surfaces in range images are often analyzed using this common representation.

There are many different types of parametric surface representations discussed in the computer graphics and computer-aided-design literature. The differences between these surface representations are due to different choices of representations for the individual $f(u,v)$, $g(u,v)$, $h(u,v)$ functions. Coon's patch and tensor product composite surfaces are useful when the parameter domain D is rectangular. Coon's patches are represented using one-dimensional boundary curves and blending functions. Tensor product surfaces and surface patches are represented as a "quadratic" form $S(u,v) = B_u(u)[Q]B_v^{\ T}(v)$ where the $[Q]$ matrix is a function of a set of control points and the B vectors consist of one-dimensional basis function components. The following is a list of the various types of tensor product surfaces (the first six surfaces are described in Faux and Pratt [48] while the last surface type is described by Tiller [142]):

(1) Ferguson bicubic surface patches,

(2) Bezier bicubic surface patches,

(3) Rational Biquadratic surface patches,

(4) Rational Bicubic surface patches,

(5) Parametric Spline surfaces,

(6) B-Spline surfaces,

(7) Rational B-Spline surfaces.

The use of homogeneous coordinates allows rational and non-rational tensor product surfaces to be put in the same form. Rational B-spline surfaces are quite general in that they can precisely represent quadric primitives, free-form surfaces, and polyhedral objects using one mathematical form. (As a result, this representation has been adopted as an IGES standard for 3-D surfaces for the CAD/CAM industry.) They are fairly complicated, however, and it requires substantial effort to implement them.

York et al. [157] [158] discuss the use of Coon's surface patches with cubic B-spline boundary curves as a surface representation for computer vision. The use of this surface type within the VISIONS system Long Term Memory (LTM) layered network database is presented in detail. A preliminary example of matching using the shape features of a 3-D circle is presented in [157]. Although the scope of these papers is very limited, it is one of the few works to discuss CAD surface representations in the computer vision context.

Sometimes it is necessary to represent surfaces over arbitrary domains $\dot{D}$. Barnhill [10] surveys the use of triangular interpolants and distance-weighted interpolants to create smooth surface descriptions from arbitrarily located point data. Hence, another type of surface representation is one where a set of data points is given along with an interpolation scheme. These surface descriptions can be categorized according to whether the smooth surface passes through the given points or whether the surface approximates the given points while minimizing some error criterion. There are too many different techniques of this sort for us to attempt to survey them here.

We have seen that there are many different ways to represent objects and surfaces. Three-dimensional object recognition must use one of these known techniques to describe objects or invent new ones. Each known method has its own advantages and disadvantages. One needs to be aware of all these modeling issues when constructing a object recognition system in order to make informed, intelligent decisions.

4.3. 3-D Object and Surface Rendering Algorithms

Once an object and/or surface representation has been selected as a model data storage mechanism for an object recognition system, it would be very convenient to have some technique to transform the model data into synthetic sensor data and/or a symbolic scene description. This would allow the introduction of a feedback loop which could be used to evaluate a computer vision system's understanding automatically in terms of confidence factors. How can this rendering task be accomplished?

The computer graphics literature discusses many techniques for generating line drawings and shaded images in color or black and white from geometric models. These topics are discussed in the computer graphics textbooks [50] and [107]. These techniques are usually divided into display space algorithms, object space algorithms, or hybrid algorithms. Some kind of sorting of graphic primitives is usually required which makes these algorithms compute-intensive. However, techniques are continually improving and may impact computer vision research. If the model data is adequate for general purpose vision, relatively realistic intensity images or range images and the corresponding symbolic scene descriptions can be generated automatically upon request within a vision system by using an appropriate algorithm.

Much research work is specifically interested in depth map sensor data. The z-buffer or depth-buffer algorithm from computer graphics can be used to generate synthetic depth maps from arbitrary polyhedral object models. It is easy to implement such an algorithm in software. In fact, hardware implementations of this algorithm are commercially available for generating intensity images. Therefore, it may not be unreasonable to assume the availability of extremely fast rendering algorithms in the design of future object recognition systems if needed.

4.4. Intensity and Range Image Formation

In order to use sensor data to yield information about the real world, it is important to understand the image formation process. This process has been studied in detail by both computer vision and computer graphics researchers. Ballard and Brown [7] contains a good treatment of this subject. At each point in an intensity image, the brightness value *encodes information* about surface geometry (shape, orientation, and location), surface reflectance characteristics and texture, scene illumination, the distance from the camera to an object surface, the characteristics of the intervening medium, and the camera characteristics which include spatial resolution, noise parameters, dynamic range, brightness resolution, and lens parameters. Over the years, increased understanding of intensity image formation [69] and the constraints of the physical world has led to important computer vision research developments, such as shape from binocular stereo [56], shape from motion [77] [147], shape from shading [71], shape from photometric stereo [36] [155], shape from texture [153], and shape from contours [80]. (These methods will be referred to collectively as shape from (xxx) techniques.) These developments are directed toward the goal of correctly inferring the three-dimensional structure of a scene from only brightness values. The great difficulty in reaching that goal is certainly related to the large number of factors encoded in each brightness value during the intensity image formation process.

Range image formation is also generally a very complicated process. At each point in a range image, the depth value *encodes information* about surface geometry, the distance from the camera to an object surface, and the rangefinder characteristics which include spatial resolution, range resolution, dynamic range, noise parameters, and other rangefinder parameters which depend on the type of rangefinder used. One important difference is that scene illumination and surface reflectance are *not directly* encoded in range values. Moreover, rangefinders directly produce the depth information which the shape from (xxx) techniques mentioned above seek to produce. Even though rangefinders are sometimes regarded as specialized non-vision instruments since they do not address vision as humans experience it, they are receiving a great deal of attention and can be very useful sensors in many situations. Since rangefinders are not nearly as common as cameras and digitization equipment, we discuss different techniques for sensing depth. This review is a condensation of the material found in [7] [78] [79] [115].

Rangefinders can be classified as using either *active* or *passive* methods. Active methods are regarded as such because they project energy onto a scene to measure range. Ultrasound and radio wave techniques can be used for range determination, but do not currently possess high enough resolution for most range imaging purposes. Lasers can be used as pulsed-mode or modulated continuous-wave range sensors. A pulsed-mode time-of-flight laser rangefinder determines distance by measuring the elapsed time between pulse transmission and signal reception and therefore requires signal processing electronics with 70 picosecond time resolution to obtain a depth resolution of 1 centimeter. A laser rangefinder of this type is discussed in [89]. A schematic for a rangefinder of this type is shown in Figure 11. Amplitude-modulated continuous-wave laser rangefinders determine distance by measuring the phase difference between the received wave and a reference signal. A laser rangefinder of this type is discussed in [141]; the range ambiguity problem with this sort of sensor is also discussed. A diagram for a rangefinder of this sort is shown in Figure 12. Both types of laser rangefinders tend to be fairly expensive, spatial resolution is typically about 128x128 pixels, depth resolution is usually about 1 cm for objects in the 1-4 meter range, and these instruments are often very slow compared to TV cameras. The state-of-the-art in close-range high-resolution laser rangefinders can be summarized perhaps by listing the specifications of the Environmental Institute of Michigan laser rangefinder discussed in [141]:

(1) Source: Gallium Arsenide Laser Diode, 20mW

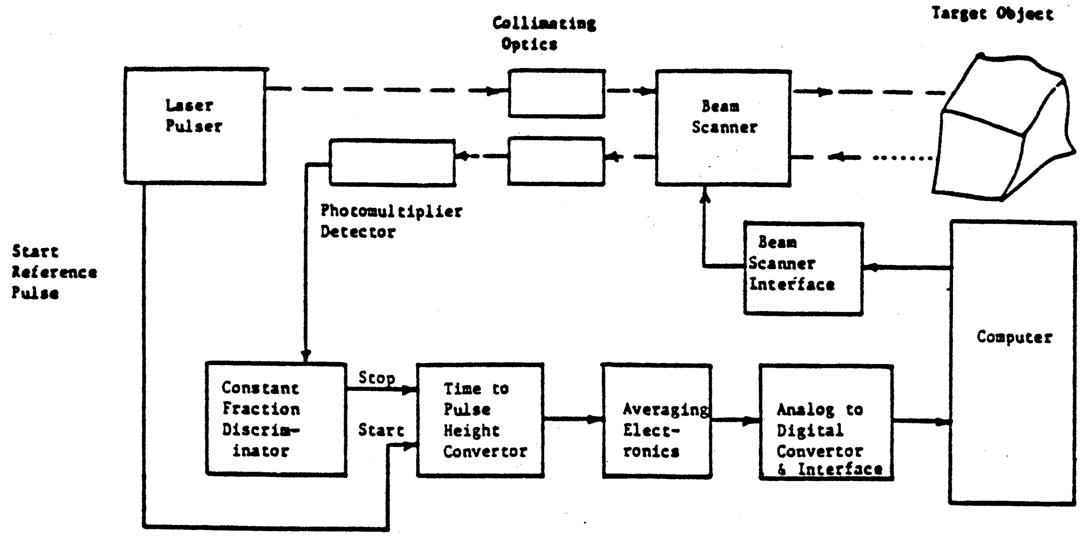


Figure 11. Pulsed-Mode Time-of-Flight Laser Rangefinder Schematic (from [78])

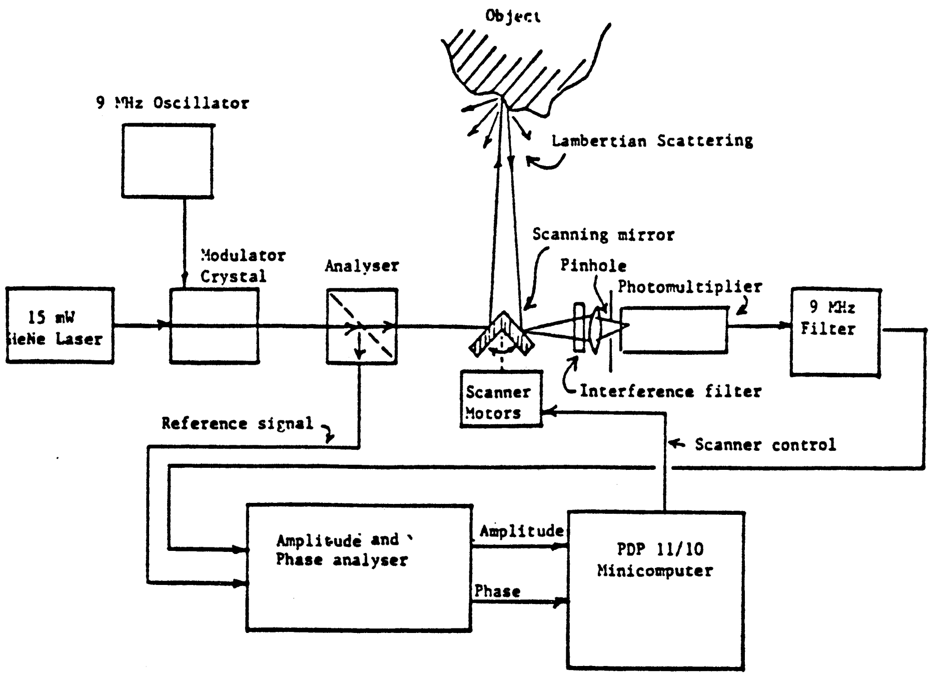


Figure 12. Continuous-Wave Phase Detection Laser Rangefinder Diagram (from [78])

(2) Dynamic Range (Ambiguity Interval): 6 inches to 3 feet.

(3) Range Resolution: 0.001 inch to 0.1 inch (13 bits quantization)

(4) Field of View: 1.6 degrees square to 35 degrees square

(5) Frame Rate: 1 frame per second typical

(6) Scan Control: Programmable (capable of 512x512)

It is important to note that the accuracy of both types of laser rangefinders depends on the return signal power, which in turn depends upon (1) transmitted power, (2) the inverse fourth power of the distance to the object, and (3) the object's surface reflectance. For this reason, the laser power is sometimes large enough that human eye damage may result if safety precautions are not exercised.

Lasers are also used in triangulation based rangefinders where a spot or line of light is projected onto a scene. Cameras or infrared sensors are used to detect the light, signal or image processing techniques are used to determine the position of the spot or pieces of the line, and trigonometry is used to estimate the distance to the detector. See Figure 13 for an example of triangulation rangefinder geometry. Depth resolution depends on how well positions, distances, and angles can be measured. Triangulation methods always suffer from the "missing parts" or shadowing problem due to the separation of the source and detectors whereas laser range sensors with coaxial source and

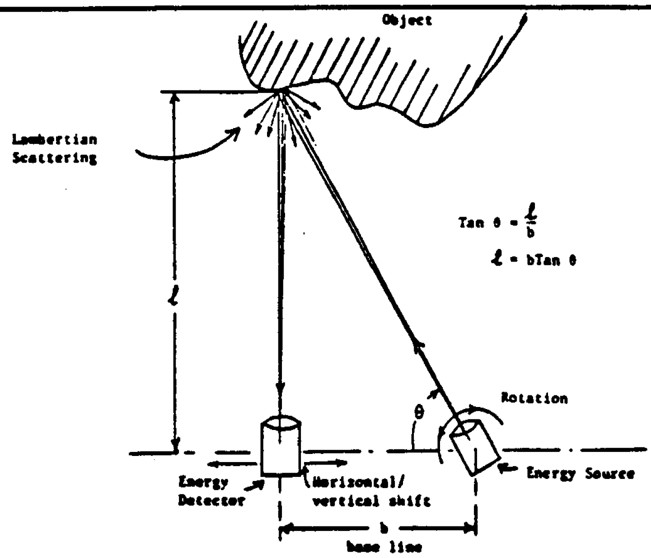


Figure 13. Simple Triangulation Rangefinding Geometry (from [61])

detector are not subject to this problem. Figure 14 show a detailed categorization of the rangefinder discussed in [115]. This gives one an idea of the variety of laser rangefinders.

Non-coherent white light is also used for range finding in much the same way that laser light is used. A spot, line or stripe of light, or even an entire

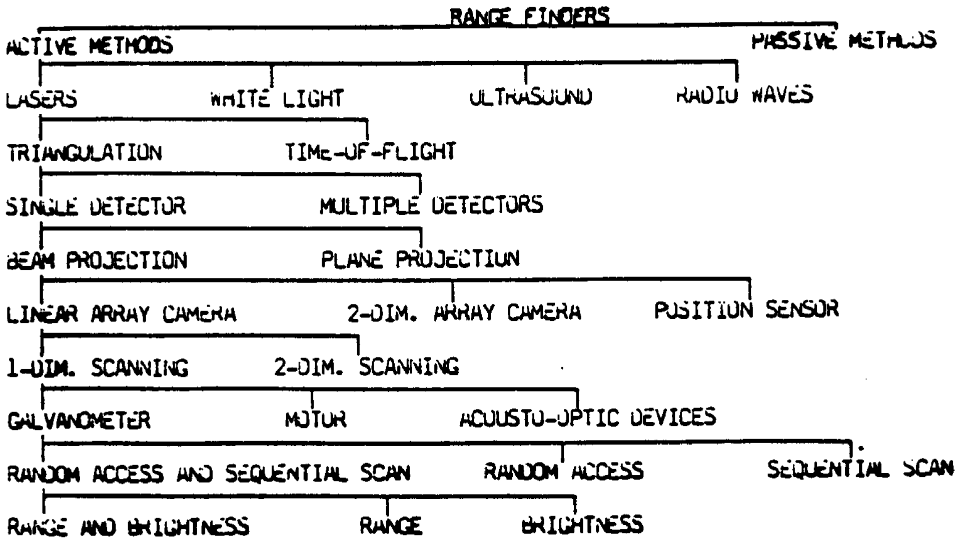


Figure 14. A Classification of Rangefinders (from [90])

grid of light [118] [143] can be projected onto a scene and the reflected light is detected by cameras. Image processing techniques are then needed to isolate the bright pixels in the image, and depth is determined by triangulation. Other patterns and Moire fringe techniques [70] have also been used for range finding.

An interesting approach for range imaging using white light has been developed by Inokuchi et al. [76]. A scene is not scanned as in spot and line/stripe approaches. Instead, the scene is sequentially illuminated with a series of Gray code bit-mask patterns, and a stack of binary images is generated. This set of images is combined and transformed into a depth map. See Figure 15 which shows the Gray-code patterns and the overall system configuration. The range image shown in the paper looks quite good. Using only n binary images, this technique can generate the equivalent range information generated by a light stripe rangefinder which has scanned and processed 2^n light stripe images. Altschuler et al. [3] devised a similar instrument which used non-coded binary patterns. Rangefinders will probably become more common as this sort of innovation continues.

Passive range finding techniques are considered as such because they do not project energy onto a scene. Focusing techniques use the shallow depth of field of large aperture lenses to determine the depth of different parts of a scene. The shape from (xxx) methods mentioned above are other types of passive range finding techniques.

The major passive range finding technique is stereo [9] [56] [110] [156]. The correspondence problem of matching scene points in different images must be solved to obtain good depth values from stereo rangefinders. Even then,

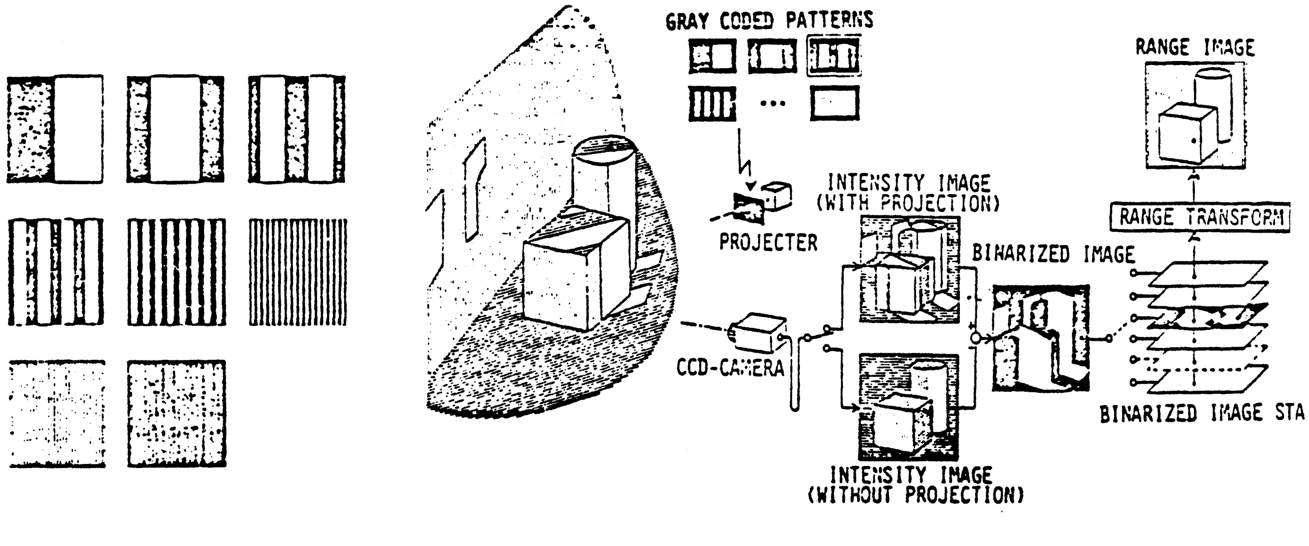


Figure 15. Gray Code Patterns and Rangefinder System Configuration (from [59])

stereo and other passive techniques generally only provide depth at isolated points in the field of view. Surfaces can be interpolated from these points to obtain entire depth maps for a scene [144]. In general, passive range finding techniques are usually computationally intensive, but it is expected that these methods will also improve with time.

As a result of the difficulties involved in passive techniques, depth maps are usually obtained from active sensors in practice. As active rangefinding techniques improve, high spatial resolution (512x512pixels), high depth resolution (16 bits) sensors should become available at relatively reasonable costs. This type of data could be extremely useful in many applications, including automatic inspection and assembly.

Range images and intensity images both contain a great deal of scene information, and some attempts have been made to combine data from these two sources [154]. In addition, laser rangefinders can produce reflectance images where pixel values represent the surface reflectance at each point. These images are similar to intensity images except that no shadows can occur, and it can also be advantageous to form registered sets of these two type of images of 3-D scenes [41] [109]. Combined use of range, intensity, and reflectance images for computer vision has not really received a great deal of attention yet. This paper will not examine such combinations although multiple sensor data integration is certainly an important research area.

4.5. Intensity and Range Image Processing

The field of intensity image processing is a relatively mature field compared to the computer vision or image understanding field. Textbooks on the subject [55] [120] [126] discuss the topics of enhancement, restoration, coding, and segmentation of digital intensity images. Since range images or depth maps have the exact same mathematical representation as intensity images, many intensity image processing techniques are directly applicable to range images. Image segmentation is a key low-level process in image understanding systems. Most segmentation work for single intensity images is based on simple thresholding, correlation, histogram transformations, filtering, edge detection, region growing, texture discrimination, or some combination of the above. The technical literature on these subjects and their applications is so vast that we shall not attempt to survey it. Instead, we will focus on the relatively small amount of literature concerned with range image processing and make comparisons with intensity image processing.

Duda et al. [41] discussed the use of registered range and reflectance data to find planar surface regions in 3-D scenes. A sequential planar region extraction procedure is described which utilizes both range and reflectance images obtained from a modulated-continuous-wave laser rangefinder which is described in [109]. A priori scene assumptions concerning man-made horizontal and vertical surfaces motivate the procedure. First, horizontal surface regions of significant size are segmented and removed from the images using a filtered range (z) histogram. Second, major vertical surfaces are extracted from the remaining scene data utilizing a Hough transform method. Third, arbitrary planar surfaces are identified with the help of reflectance (histogram) data. In this way, all planar surfaces are segmented and labeled; all unlabeled regions correspond to depth discontinuities, non-planar regions, or very small planar regions not found in the three main processing steps. The paper mentions many adjustable algorithm parameters which were assigned values on an ad hoc basis. The overall technique seems to work quite well on their three test scenes, but it is not a theoretically unified approach and relies heavily on its horizontal and vertical surface assumptions. Planar region extraction is not a common processing step in intensity image processing.

Milgram and Bjorklund [100] also discuss planar surface extraction in range images created by a laser rangefinder. The spherical coordinate transform is used to convert slant range, azimuth angle, and elevation angle sensor data into standard Cartesian data before processing. For each Cartesian coordinate range pixel, the two orientation angles, the position variable, and the goodness of fit are computed for the best fit plane within a 5 x 5 window. This data is used to form connected components which satisfy planarity constraints. After the region growing processing is complete, a "sensed plane list" is built which constitutes the scene description. This list is compared with a reference plane list to determine sensor position with respect to a stored scene model. Experimental results are discussed for four real world range images and two synthetic range images displaying different viewpoints of the same building site. This appears to be a much better, more straightforward approach to planar surface extraction than that of Duda et al. However, there was still no effort to handle curved surfaces. Figure 16 contains a block diagram of the system which was planned for vehicle navigation.

Henderson [62] [63] has developed a method for finding planar faces in one or more range images. First, 3-D object points are computed using one or

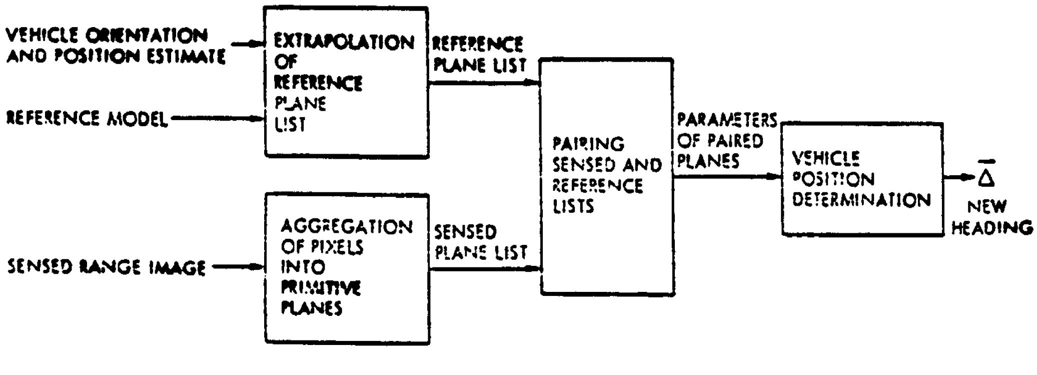

VEHICLE ORIENTATION AND POSITION ESTIMATE
REFERENCE MODEL
SENSED RANGE IMAGE
EXTRAPOLATION OF REFERENCE PLANE LIST
REFERENCE PLANE LIST
AGGREGATION OF PIXELS INTO PRIMITIVE PLANES
SENSED PLANE LIST
PAIRING SENSED AND REFERENCE LISTS
PARAMETERS OF PAIRED PLANES
VEHICLE POSITION DETERMINATION
$\bar{\Delta}$
NEW HEADING

Figure 16. Block Diagram of Matching System (from [100])

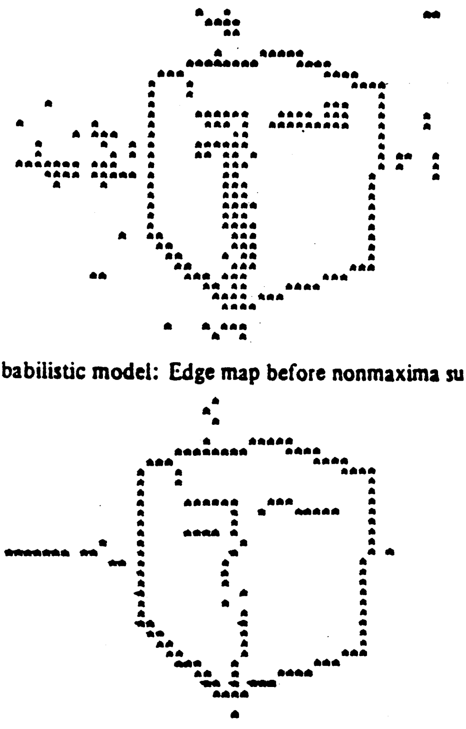
more depth maps. For multiple depth maps, points are transformed into one object-centered coordinate system using transformation data recorded during range image formation [64]. These points are stored randomly in a list with no topological connectivity information and are then organized into a 3-D binary tree which can be done in O(NlogN) time where N is the number of points. Second, each point's neighbors are determined with the aid of the 3-D tree and the results are stored in a 3-D spatial proximity graph. Third, a spiraling sequential planar region-growing algorithm known as the three-point seed method [64] is used to create convex planar faces using the spatial proximity graph as input. The union of these faces form the polyhedral object representation which is extracted from the range data. This efficient method can be used for either range data segmentation or object reconstruction. It can work on dense range data or a sparse collection of points. Curved surfaces are approximated by many polygons. This type of processing has no good analogies in image processing.

Wong and Hayrapetian [154] suggest the use of range image histograms to segment registered intensity images. All pixels in the intensity image which correspond to pixels in the range image with depth values not in a certain range are set to zero segmenting all objects in that particular range. This seems to be useful in only a few limited applications.

Hebert and Ponce [61] propose a method of segmenting depth maps into plane, cylindrical, and conical primitives. First, surface normals are computed at each depth pixel using the best-fit plane in 3x3 windows. These normals are mapped to the Gaussian sphere where planar regions become very small clusters, cylinders become unit radius semicircles, and cones become smaller radius semicircles. (This type of orientation histogram is often referred to as the extended Gaussian image, or EGI.) The Hough transform is used to detect these circles and clusters. Regions are then refined into labeled, connected components. Although still rather restricted, this technique handles at least certain types of curved surfaces in addition to handling planes.

Inokuchi et al. [74] present an edge-region segmentation ring operator for depth maps. This ring operator computes a complete one-dimensional periodic function of depth values which surround a pixel. This function is transformed to the frequency domain using an FFT algorithm for either 8 or 16 values. By examining the 0th, 1st, 2nd, and 3rd frequency components of the ring surrounding each pixel, planar region, jump-boundary-edge, convex-roof-edge, and concave roof-edge pixels are distinguished. These pixel types are grouped together and the resulting regions and edges are labeled. Experimental results are shown for one synthetic block world scene range image. This technique does appear to compute roof-edges fairly well at the boundaries of planar surfaces. However, it is not stated what happens when images contain curved surfaces. Two years earlier, Inokuchi and Nevatia [75] discussed another roof-edge detector which applied a radial line operator at jump-boundary-edge corners and followed roof-edges inward.

Mitiche and Aggarwal [102] have developed a roof-edge detector which is insensitive to noise due to use of probabilistic model which attempts to account for range measurement errors. The computational procedure goes as follows: 1) Jump-boundary-edges are extracted from a depth map. 2) For each direction (usually four) in the image at each pixel, a roof-edge is hypothesized. For each hypothetical roof-edge, two planes are fit to the immediate neighborhood of the pixel and the dihedral angles between these planes are recorded. This is referred to as the "computation of partitions." 3) All pixels at which all angles are less than a threshold are discarded. For each remaining pixel, a Bayesian likelihood ratio is computed and the most likely partition (edge-direction) is chosen. If the angle for that direction is less than a threshold, that pixel is also discarded. This is referred to as the "dismissal of flat surfaces." 4) All remaining pixels are passed through a non-maxima suppression algorithm which theoretically leaves only the desired edge pixels. Experimental results from the paper are shown in Figure 17 for a 64x64 depth map of a cube with added noise. The results look reasonable considering the large amount of noise added, but the system is internally constrained by its model to look for only horizontal and vertical edges (a domain specific constraint). It is suggested that the repeated, time-consuming laser range sampling used to achieve good distance accuracy may not be necessary to detect edges if the noise can be successfully



Probabilistic model: Edge map before nonmaxima suppression.

Probabilistic model: Final edge map.

Figure 17. Edge Maps for Depth Map of Cube with Noise (from [102])

removed at higher levels of processing.

Lynch [94] presented a range image enhancement technique for range data acquired by systems with a shallow (nearly horizontal) line of sight. The system of interest in this paper is a 94 GHz (3.2mm) radar which might hang from the bottom of an aircraft. *In this special case*, a strong depth gradient always exists in a range image. This gradient makes it very difficult for humans to interpret such a range image using a typical 256-gray level display device. Two *one-dimensional high-pass* filters (a normalized filter and a differenced filter) are derived and discussed. They are applied to an example scene creating a "feature" image which is more easily interpreted by a human observer than the original range image. Then a Sobel edge operator is applied to the original and the two filtered images which supposedly shows the "quantitative" improvement in the image quality due to the "enhancement" processing. Unless one has a similar application for human interpretation of range data, these ideas are not likely to be useful since the depth map is severely distorted by these operations.

Sugihara [140] proposes a range image feature extraction technique for edge junctions similar to the junction features used by Waltz [151] and others for intensity image understanding. A junction dictionary of possible 3-D edge junctions is implemented as a directed graph data structure and can be useful in aiding 3-D scene understanding. Unlike intensity image edges, range data edges can be classified as convex, concave, obscuring, or obscured without additional information from surrounding image regions; junction knowledge is not necessary for line categorization. This categorization can therefore be used to help predict missing edges. Some junctions are only possible when two or more bodies are in a scene; this enables junction information to be used to segment the range image into different bodies. This paper notes that junction points are *local curvature maxima* points in the depth map surface. A system is described which uses depth-discontinuity contours (edges) and junctions for complete scene segmentation. It is limited by the constraint that every vertex can be connected to only three faces.

Although we will not deal with them, several papers have discussed range image processing techniques for the detection of cylinders in range data [2] [26] [106] [116].

We have seen that the literature of range image processing places a definite emphasis on planar region extraction, specified shape extraction, and edge detection of both kinds, roof and jump-boundary. Planar region and specified shape extraction and roof-edge detection are not emphasized very much in the image processing literature.

4.6. 3-D Object Reconstruction Algorithms

In this section, some of the literature concerning three-dimensional object reconstruction is reviewed. Although our direct concern is object recognition, object reconstruction can involve many similar ideas. For example, one approach to the object recognition problem is to reconstruct as much of an object's surface as possible from a single view and perform matching of that object surface with object models. We survey intensity image methods first, followed by methods which use depth maps. We do not consider any object reconstruction schemes based on the shape from (xxx) methods mentioned earlier in this survey.

Baker [6] presents a scheme for building object models from many intensity images taken from known different rotated views. One main premise is that "effective vision requires flexible, domain-free, three-dimensional modeling." The authors are in agreement with that fundamental premise. His method tracks edge curvature irregularities using correlation techniques over a series of rotated views and creates a wire-mesh exoskeleton (wireframe) representation. An example of this process is shown in Figure 18. Experimental results are shown in the paper for two complex smooth surfaced objects. The results are difficult to visualize due to the wireframe line drawings, but they are definitely quite detailed. The author suggests a preliminary matching process which uses the maximum breadth axis of the object and a list of the n-th (e.g., n=6) most convex and concave points (points of *high surface curvature*). His algorithm successfully matched two descriptions of the same object which were analyzed in two different orientations. The main ideas of this paper are much more general than those expressed in many of the earlier papers.

Bocquet and Tichkiewitch [19] have an expert system approach to the object reconstruction problem. Their system accepts input in the form of mechanical drawings done from three orthogonal views. The drawings are digitized, and line and arc segments are given internal representations. The list of segments is then structured into a 2-D relational database which keeps track of all closed contours. A set of production rules (the knowledge base) are used to infer 3-D surfaces from the 2-D data. The hypothetical surfaces generated for one view are projected into the next view so that the corresponding 2-D segments can be checked. The system can then make adjustments and continue, or it can backtrack. When a surface representation for the object is obtained which is compatible with the given views, the representation is drawn from various viewpoints as the system output. Figure 19 contains several diagrams which graphically describe the overall structure of the system. Input drawings and very good results are shown for one machined part. When no production rules apply, the system requests the operator for a new one. If rules have been found for all segments of a contour, but no surface is generated, the system explains the problem and allows rules to be changed or added by the human operator. Presumably, a fairly robust system will result after many objects have been successfully reconstructed. This system could *possibly* be generalized to work from high quality edge maps rather than drawings. Much more object detail is potentially available from this approach than those which construct volume descriptions from multiple silhouette boundaries (see, for example, Martin and Aggarwal [96]).

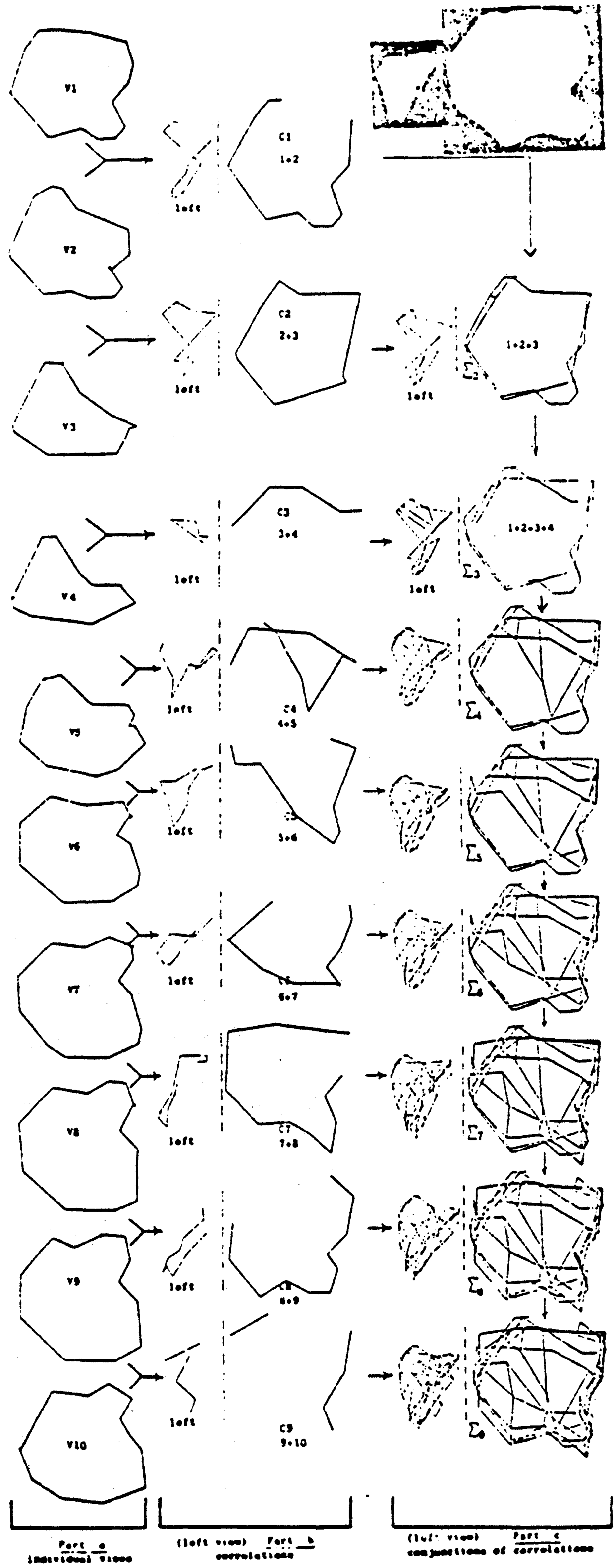
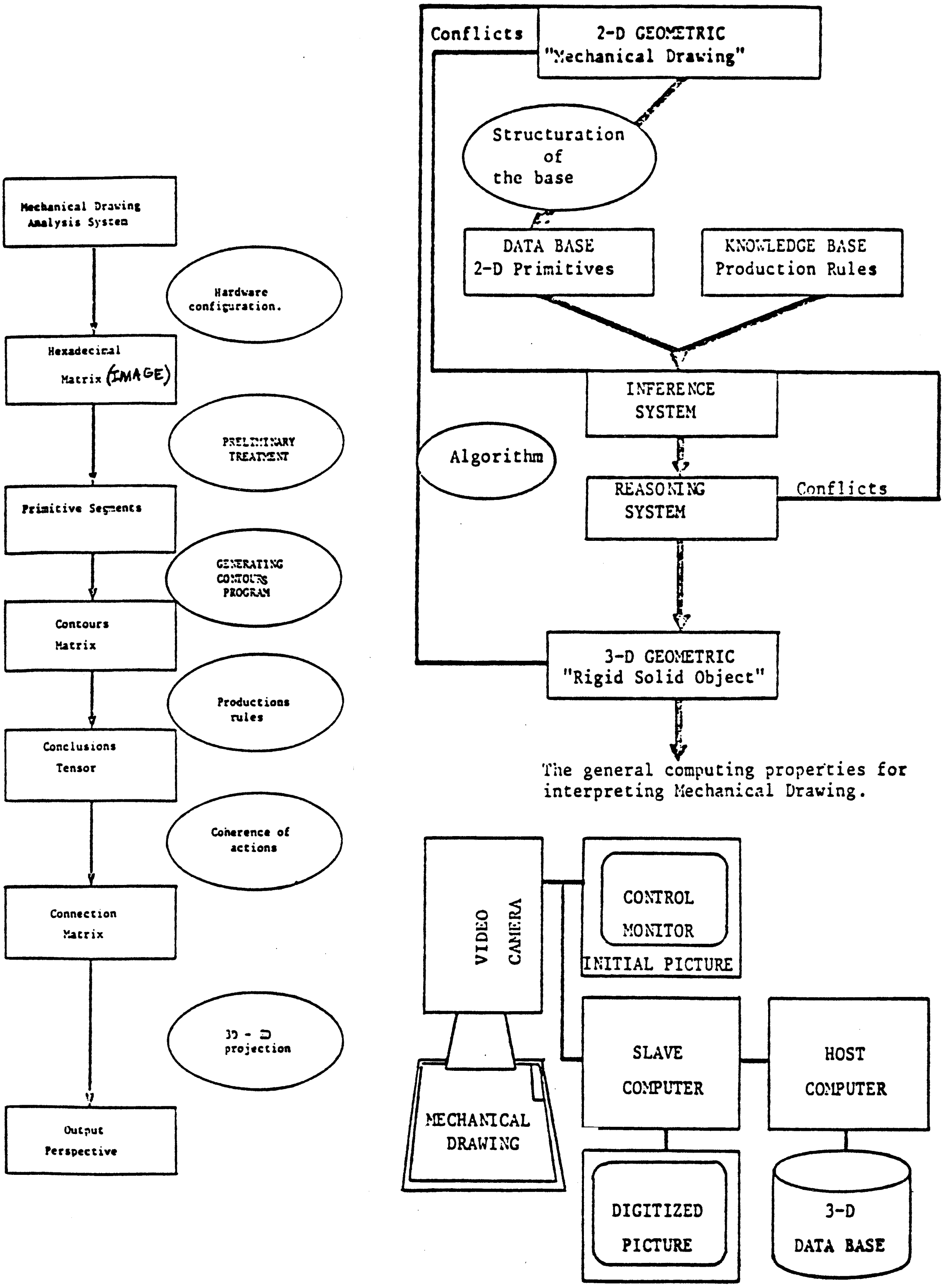


Figure 18. Progression of Modeling in 20 Degree Increments (from [6])



The general computing properties for interpreting Mechanical Drawing.

General System

Figure 19. Diagrams of Object Reconstruction Expert System (from [19]))

The previous paper was preceded by the work of Lafue [85]. He also wrote a program for interpreting orthographic view drawings as 3-D objects. Heuristics are used to resolve ambiguities resulting during the aggregation of points, edges, and faces into polyhedra. Instead of an expert systems approach, Lafue used a mini theorem prover to choose the right geometry for each set of local alternatives. This program was also written to interact with the user to get help when it was needed.

Shapira and Freeman [134] describe a procedure for reconstructing objects bounded by planar or quadric surfaces from a set of photographs of a scene taken from different viewpoints. Line and junction labelings of the type used by Waltz [151] and others are used to extract model surface descriptions of objects. Vertices must be formed by exactly three edges. No object shape restrictions are used and the method is designed to handle a limited amount of imperfections in the line-junction feature data. Experimental results are shown for one high contrast scene with several simple objects; the results look reasonable.

Abe, Itho, and Tsuji [1] proposed a system to build 3-D qualitative object models of objects with cylinder-like bodies given several 2-D intensity image views and verbal explanations of object structure. Their system mainly consists of a language interpreter, an image-to-language communications subsystem, and a model generator. It also includes an image processing subsystem and question generator for interacting with the human operator. Figure 20 shows a block diagram of the system organization. The image inputs for each view and the language input are processed and stored in separate internal frame representations. These frame representations are matched for each view to generate a consistent labeling. Different views of an object are constrained by language input and combined using a graph matching process to create a 3-

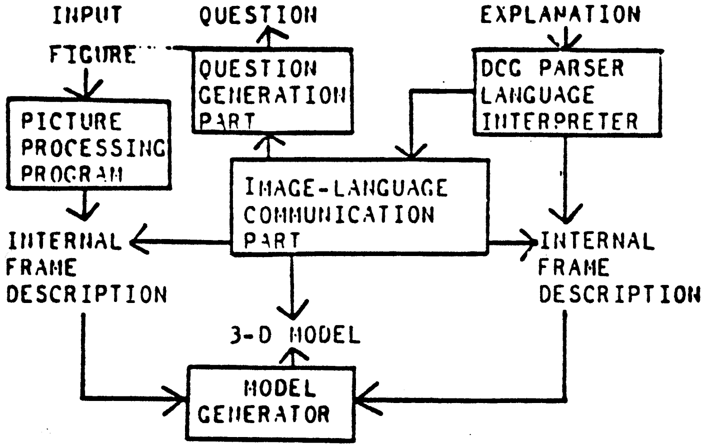

INPUT
FIGURE
QUESTION
EXPLANATION
PICTURE PROCESSING PROGRAM
QUESTION GENERATION PART
DCG PARSER LANGUAGE INTERPRETER
IMAGE-LANGUAGE COMMUNICATION PART
INTERNAL FRAME DESCRIPTION
INTERNAL FRAME DESCRIPTION
3-D MODEL
MODEL GENERATOR

Figure 20. System Organization (from [1])

D model. This paper is preliminary and does not present any real results; the authors state that they had many difficulties during their experiments. The work is an attempt to generalize methods for learning 2-D object shape which they had previously developed. The concepts are interesting, but the authors present no justification that the method will necessarily even work.

Herman et al. [65] [66] have implemented the 3-D MOSAIC scene understanding system. This system can incrementally derive a 3-D description of a complex urban scene from multiple intensity image stereo views and task-specific knowledge. A partial 3-D wireframe description is derived from each stereo view. Figure 21 shows an example of such a wireframe description. The wireframe descriptions from different directions are aligned and then processed sequentially. Close parallel edge segments are combined into single edges. Each vertex is assumed to correspond to a corner of an object; therefore, adjacent corner edges correspond to a corner of a planar face. Compatible corners of faces are merged into complete faces. Faces which are probably flat roofs are converted into buildings by adding more faces between the roof and the ground. A complete set of similar kinds of rules are applied to the sequence of images as the processing continues. Some of these rules are shown in Figure 22. Finally, a complete scene description is generated which is rendered as a gray scale image. This technique can work well for particular domains with predominantly block shapes. In this case, all objects are assumed block-shaped, all surfaces are assumed horizontal or vertical, all parallelograms are considered as rectangles unless there is evidence to the contrary, and even the ground plane is assumed known. Unfortunately, there seems to be little general purpose theory available from such an approach.

We have overlooked all techniques which obtain object structure explicitly from the motion associated with corresponding image sequences as these are

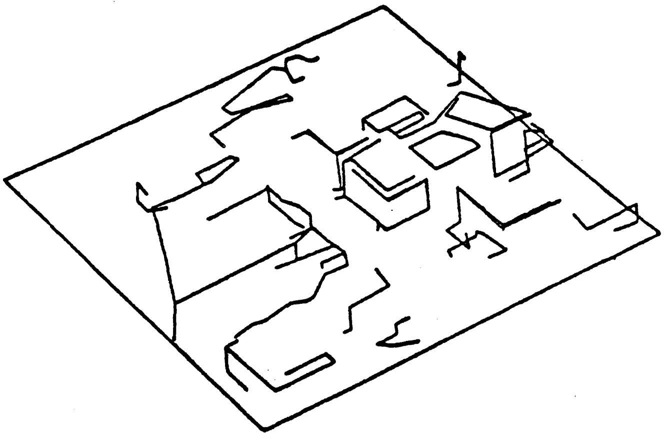


Figure 21. Perspective View of 3D Wire Frames (from [66])

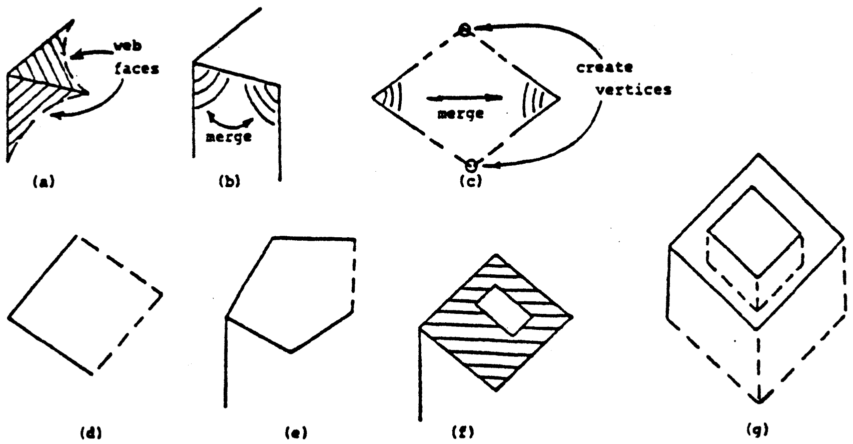


Figure 22. Obtaining Surface-Based Description from Wire Frames (from [66])

shape for motion techniques. This completes the survey of intensity image based object reconstruction papers. Next we discuss methods based on range data.

Vemuri and Aggarwal [149] have implemented an algorithm for "reconstructing" 3-D "objects" using range data from a single view. The algorithm proceeds as follows:

(1) The range image is partitioned into overlapping KxK window neighborhoods. The overlap is two pixels.

(2) For each neighborhood, the standard deviation of the Euclidean distance between the range points in the KxK window is computed. If this is less than a preset threshold, a tension spline tensor product surface patch is fitted to the window. If not, the window is discarded.

(3) The principal surface curvatures (minimum and maximum) are computed at each point in the remaining patches. If the magnitude of either curvature value exceeds another preset threshold, the point is labeled as an edge pixel.

In this way, the depth map is approximated by a set of continuous surface patches, and the edges within those patches are determined. This surface patch model is then passed through a graphics algorithm with a light source model to obtain a shaded image. Experimental results for one synthetic and one real range image are displayed in the paper. Figure 23 shows the synthetic range image results for a car shape. The algorithm's results consist of a shaded image and an edge map. The authors fail to point out that the shaded image could have been generated directly from the depth map itself without any intermediate surface-fitting. No 3-D object reconstruction appears to have been done in this work; a set of surface patches has been fitted to data. They

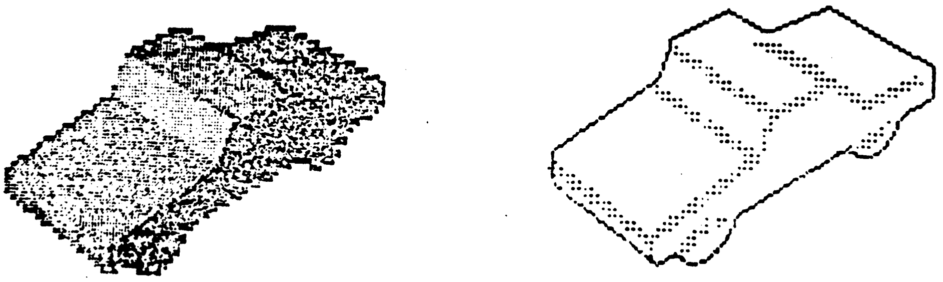


Figure 23. Surface Reconstruction; Jump Boundaries and Internal Edges (from [149])

do mention their intention to merge surface patches with similar properties into regions and form a region adjacency graph for recognition purposes. It is interesting to note the use of principal surface curvatures for edge detection.

Potmesil [117] [118] describes a method to generate models of solid objects by matching 3-D surface segments which are obtained using a white-light grid-projecting triangulation-based range finder. First, depth maps are obtained from a sufficient number of views to determine object shape and to allow sufficient size surface regions to be imaged in at least two views. The range data for an object is fit with a sheet of parametric bicubic surface patches. The rectangular patches are recursively merged (four at a time) into a quadtree hierarchical structure so that each projected surface is represented by a tree of surfaces where the root node is very coarse and other nodes become increasingly more detailed as one moves down the tree. The bottom of the tree contains the original patches. The surface information in this structure is queried via ray casting techniques so that the particular surface representation details can be modified at any time leaving the rest of the system intact. This is an important idea for a flexible system. *Surface matching* is defined as "finding a spatial registration of two surface descriptions that maximizes their shape similarities." Given a particular surface in one coordinate system and a set of surfaces, each in their own coordinate system, the algorithm computes the registration transformation to the appropriate other surface which provides the best surface segment match. Evaluation points on surfaces are used to compute (1) positional differences, (2) orientation differences, and (3) curvature differences. The evaluation points are selected at each level in the quadtree representation at surface control points or at points of *maximum curvature*. A heuristic search algorithm is used to control generation of registration transformations. The initial guess corresponds to the alignment of surface normals at the top of the quadtree. When sufficiently good segment matches are found, a merging algorithm generates a new surface for each matched surface segment. A complete object model is created by sequentially matching and merging segments. Experimental results of this technique for a balsa car model are shown in

Figures 24 and 25. 36 gray-scale images of grids from 6 views are generated which results in 18 3-D surface segments. These are matched and merged into 6 depth map surfaces corresponding to 6 physically different viewpoints shown in Figure 24. Then these 6 surfaces are matched and merged into one complete object model, and four views from arbitrary directions are displayed in Figure 25; the reconstruction results look quite good. It would seem to be a trivial step to generalize this approach for object recognition since surface matching is already implemented, but this was only suggested by the paper as a potential use of the surface matching approach. This paper is the summary of a Ph.D. thesis [119].

Dane and Bajcsy [37] present an object-centered 3-D model builder which utilizes 3-D surface point information obtained from many views. In the first stage of analysis, points for each view are grouped according to the following directly observed properties of the data: (1) number of data points in a local area, (2) average and deviation of depth values, (3) average and deviation of X component of normal, and (4) average and deviation of Y component of

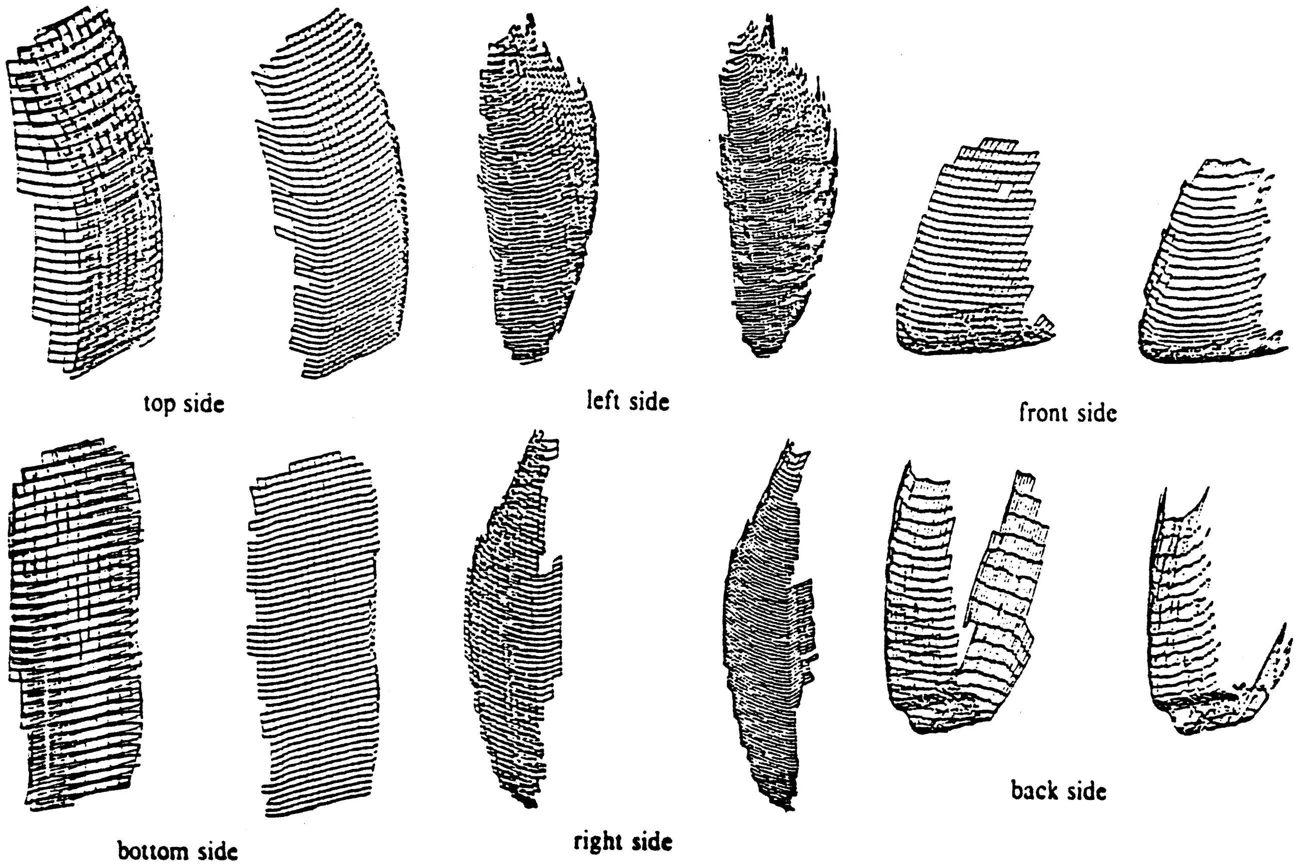


Figure 24. 18 Surface Segments merged into 6 New Segments (from [117])

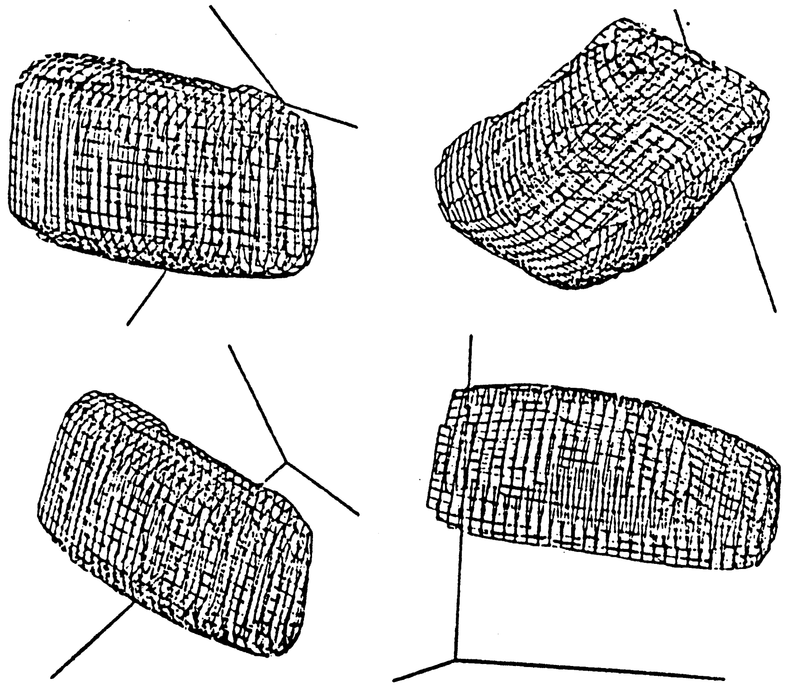


Figure 25. Four Views of Surface Segments Matched into One Model (from [117])

normal; and the following derived properties of the data: (a) local curvature in a X-Z plane, (b) local curvature in a Y-Z plane, (c) surface orientation continuity, and (d) surface depth discontinuity. It is not stated in this paper how these derived properties are computed or how the properties directly affect the grouping process. In any case, the points are grouped using these concepts and either a planar or a quadric surface primitive is fitted using a least squares technique. The second stage of analysis determines edges and corners which are stored in an edge graph structure. The view analysis is followed by view integration where surface primitives are transformed using known transformations into a common global coordinate system, identical surfaces are identified, and surface parameters are modified for overall object compatibility. The resulting object description is standardized by placing the origin at the center of gravity of the object and aligning the x,y,z directions with the principal axes of the object. The algorithm was tested with nine objects and only made one error. 36 views were used to define the object although not all points in all views were used in every case. Data acquisition is not described as the 3-D data points are assumed available from another existing object model. This paper is the summary of a Ph.D. thesis [38].

We note here that the work of Henderson [62][63] mentioned previously describes a technique to automatically reconstruct a polyhedral model of an object from points obtained from several views.

Faugeras et al. [47], Boissonnat [20], and Boissonnat and Faugeras [21] all describe an efficient (O(NlogN)) way of building a polyhedral approximation of 3-D points obtained from a triangulation laser range finder. The 3-D algorithm is presented as a generalization of the triangularization algorithm for a 2-D polygon. The basic approach is a graph-guided divide-and-conquer procedure.

(1) A planar graph G=(V,A) is constructed using the given 3-D points where arcs connect nearest neighbors.

(2) Three non-neighboring points P, Q, R are selected for initialization.

(3) The shortest most planar cycle within these points is found and labeled PQR. This divides graph G into two disconnected subgraphs and the surface of the object into two surfaces.

(4) For each subgraph, the point most distant from the plane PQR is found. The resulting hexahedron is now a first order approximation to the object surface.

(5) Subgraphs for each triangular face are determined and the previous step is applied recursively until all points are exhausted.

This processing is slightly modified to insure that bad edges do not remain in the approximating polyhedron as the algorithm proceeds. Good results for fairly complicated objects are shown in Figure 26. The input and output for

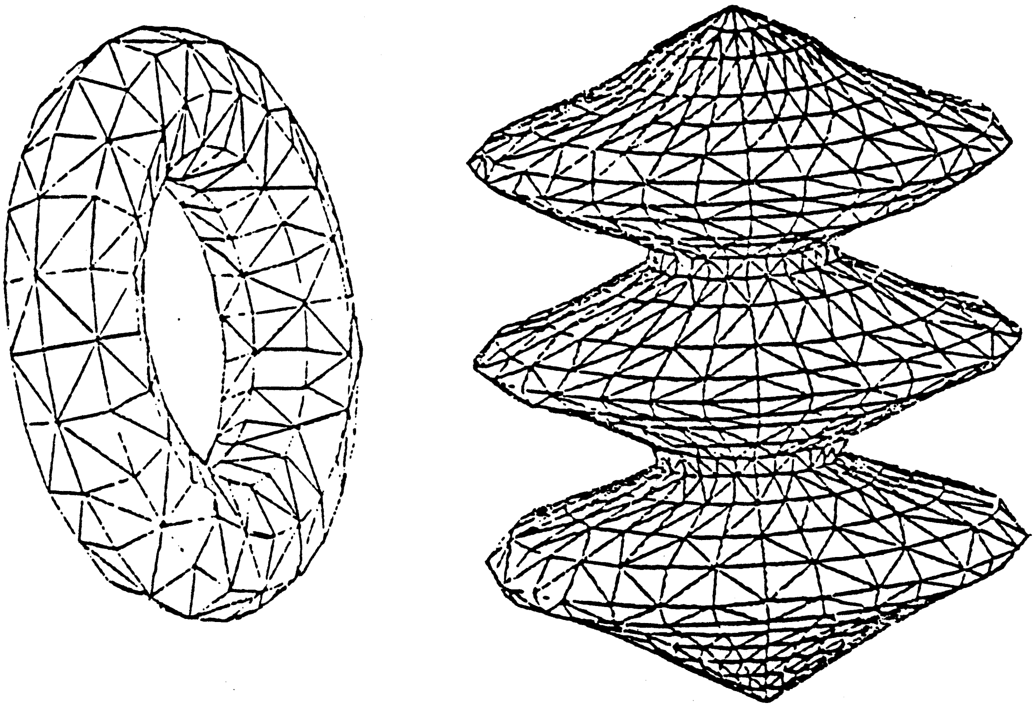


Figure 26. Objects Reconstructed from Point Information (from [20])

this approach are very similar to the approach in Henderson [62] although the processing algorithms are different.

Little [93] has discovered a method to reconstruct convex polyhedral object models from their corresponding Extended Gaussian Images (EGI). Given a uniformly spaced grid map of surface normals of a depth map, we can divide up the spherical solid angle into bins and then form an orientation histogram by computing the number of grid points which have normal vectors that fall into each bin. This orientation histogram is referred to as a discrete EGI. It can be shown that the continuous analog of this discrete EGI uniquely determines a convex polyhedron via a non-constructive proof [101]. Little has posed the object reconstruction problem as a iterative constrained minimization problem and solved it. His results answer what was previously an open question concerning the inversion of EGI's which is certainly of mathematical interest. Unfortunately, the class of convex polyhedral objects is so small compared to the class of objects of general interest that these reconstruction results are not directly useful for general purpose object reconstruction.

This concludes the review of object reconstruction using range data. We have seen that the range data methods are somewhat more quantitative than the intensity data methods. This is expected considering that reasoning and inference are extremely important to intensity image 3-D understanding due to the lack of explicit range information.

4.7. 3-D Surface Characterization

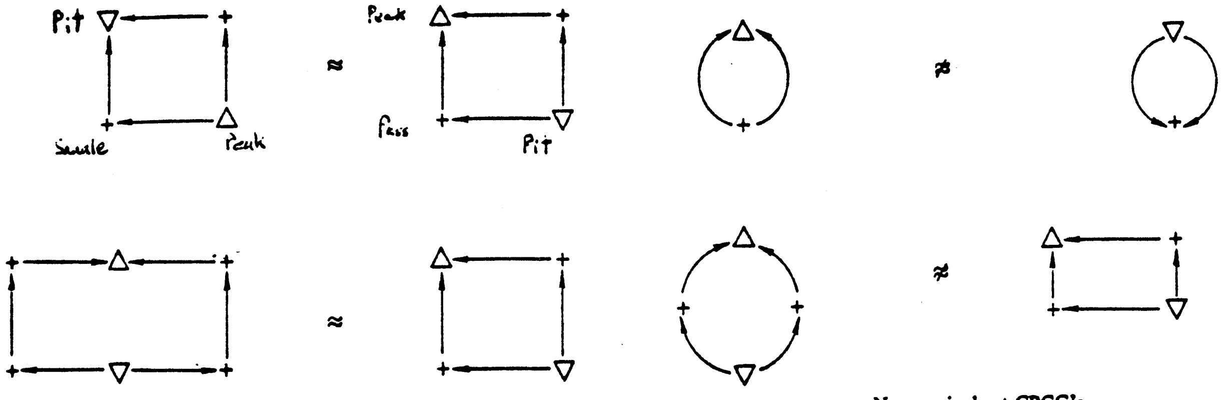
In this section, the existing literature concerning three-dimensional surface characterization is reviewed. We use the term *surface characteristic* to denote a descriptive quantitative feature which can be computed from a surface, but which need not retain enough information for surface reconstruction from the description. Surface characterization is an important topic which does not seem to have attracted much attention. The quality of features used to identify surfaces will be critical to the performance of object recognition systems using range data.

Nackman [104] discusses surface description using two-dimensional critical point configuration graphs (CPCG). Non-degenerate critical points of surfaces are local maxima, local minima, or saddle points. Most surfaces have the property that the critical points of the surface are isolated. Surfaces which do not have this property can be very closely approximated by ones that do. By identifying all the critical points of a surface as the nodes of a graph and their connecting ridge and course lines as the arcs of a graph, a surface can be characterized by this graph known as the critical point configuration graph. Slope districts are bounded by graph cycles. Several theorems relating to these graphs are proved:

(1) Only eight types of critical points are possible: peaks (local maxima), pits (local minima), and six types of passes (saddle points).

(2) Only four types of slope districts are possible.

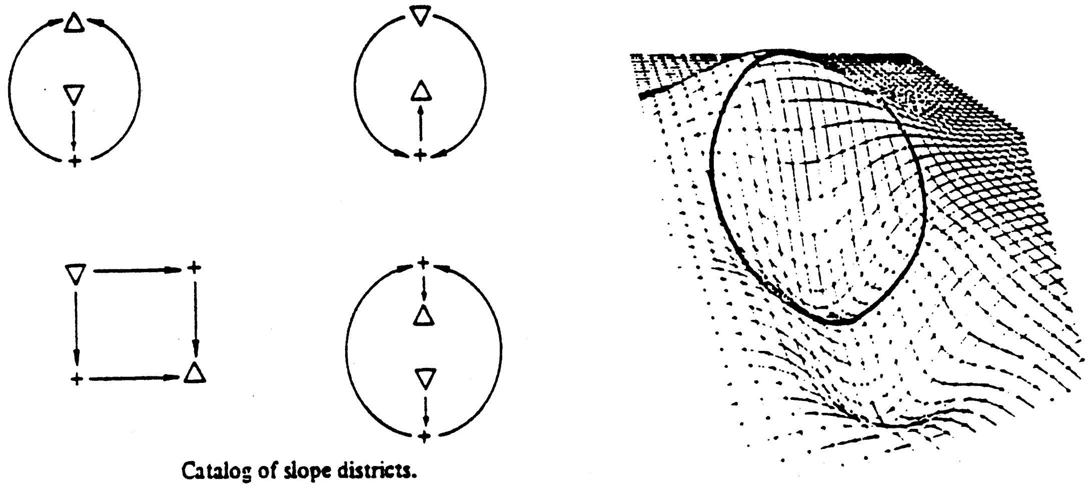
Hence, surfaces have a well-defined characterization as the union of a very small finite set of slope district types. Figure 27 shows equivalent and non-equivalent critical point configuration graphs. Figure 28 shows the catalog of the four distinct types of slope districts which are not equivalent and an example of a slope district with one pass. This characterization is a generalization of the techniques used to describe one-dimensional functions f(x). In that domain, only two types of non-degenerate critical points exist: local maxima and local minima. These points are distinguished based on the sign of the second derivative at the point. Between these critical points are intervals of



Equivalent CPCG's.

Nonequivalent CPCG's.

Figure 27. Equivalent and Non-equivalent CPCG's (from [104])



Catalog of slope districts.

Figure 28. Slope District Catalog and Example with One Pass (from [104])

constant sign of the first derivative. Slope districts are generalizations of these intervals for surfaces. The paper also mentions the use of curvature districts determined by the *sign* of the mean and Gaussian curvature of a surface for surface characterization. Surface curvature generalizes the notion of the 1-D second derivative.

Lin and Perry [91] have investigated surface shape description using surface triangularization. Differential geometry based shape measures can be very useful if they can be computed from sensor data. When a surface is decomposed into a network of triangles, many features can be easily computed. Discrete coordinate-free formulas for surface area, Gaussian curvature, aspect ratio, volume, and the Euler-Poincare characteristic are given in this paper. The formula for Gaussian curvature is significant in that estimates of second partial derivatives are not needed and it is independent of coordinate system reflecting the invariant properties of the Gaussian curvature. Integral Gaussian curvature, integral mean curvature, surface area, volume, surface area to volume ratio, integral curvature to the n-th power, and genus, or handle number, are all mentioned as scalar values which characterize the shape of a surface. No experimental results of any sort are given in the paper. Integral curvature features for solder joint description in gray scale images have been used with some success [14].

Sethi and Jayaramamurthy [133] have investigated surface classification using characteristic contours. Surface input is a needle map of surface normals. A characteristic contour is defined as the set of points in the needle map where surface normals are at a constant inclination to a reference vector. The following observations are made concerning these contours:

(1) The characteristic contours of spherical/ellipsoidal surfaces are concentric circles/ellipses.

(2) The characteristic contours of cylindrical surfaces are parallel lines.

(3) The characteristic contours of conical surfaces are intersecting lines.

These contours are computed for all normals within a 12x12 scanning window. The identity of the underlying surface for each window is computed using the Hough transform on the contours. A consistency criterion is used to fight noise effects and multiple surface types within a given window. Experimental results are discussed for synthetic 40x40 needle maps of adjacent cones and cylinders. The algorithm correctly classifies these simple shapes on man-made data. This approach does not appear to be useful for general purpose surface characterization.

Laffey, Haralick, and Watson [84] [60] discuss topographic classification of digital surfaces. They review seven previous papers on the subject by various authors, and their ten topographic labels are a superset of all labels used in the previous papers: peak, pit, ridge, ravine (valley), saddle, flat (planar), slope, convex hill, concave hill, and saddle hill. At each pixel in an image, a local *facet-model* two-dimensional cubic polynomial fit is done to estimate the first and second partial derivatives of the surface at that pixel. Once the derivatives have been estimated, the magnitude of the gradient vector, the eigenvalues of the 2x2 Hessian matrix, and the directional derivatives in the direction of the Hessian matrix eigenvectors are computed. These five scalar values are used to do a table lookup on the pixel classification. The table is shown in Figure 29. The asterisk (*) means the appropriate value does not matter. This pixel by pixel classification could be used to form groups of pixels of a particular type. No experimental results are given in the paper; only their future research directions were outlined. This work is proposed for use with intensity image gray level surfaces, not with depth maps.

There is one recent paper that we would like to discuss which deals with curve characterization rather than surface characterization. Marimont [95] presents a representation for image curves and an algorithm for its computation. His representation "is designed to facilitate matching of image curves with model *plane curves* and the estimation of their orientation in space despite the presence of noise, variable resolution, or partial occlusion." This multiple scale representation is curvature-based. First, a list of scales is determined. For each scale, the points, or knots, which are the zeros and the extrema of curvature, are stored in a knot list with a tangent direction and a curvature value for each knot. These knots have the following nice properties:

(1) The zeros of curvature of a 3-D plane curve always project to the zeros of curvature of the corresponding 2-D image curve.

Pixel Classification Scheme

$\|\nabla f\|$	λ_1	λ_2	$\nabla f\cdot\omega^{(1)}$	$\nabla f\cdot\omega^{(2)}$	Label
0	-	-	0	0	Peak
0	-	0	0	0	Ridge
0	-	+	0	0	Saddle
0	0	0	0	0	Flat
0	+	-	0	0	Saddle
0	+	0	0	0	Ravine
0	+	+	0	0	Pit
+	-	-	-,+	-,+	Concave Hill
+	-	*	0	*	Ridge
+	*	-	*	0	Ridge
+	-	0	-,+	*	Concave Hill
+	-	+	-,+	-,+	Saddle Hill
+	0	0	*	*	Slope
+	+	-	-,+	-,+	Saddle Hill
+	+	0	-,+	*	Convex Hill
+	+	*	0	*	Ravine
+	*	+	*	0	Ravine
+	+	+	-,+	-,+	Convex Hill

Figure 29. Pixel Classification Scheme (from [84])

(2) The sign of the curvature value at each point does not change within an entire hemisphere of viewing solid angle. Therefore, the *pattern of sign changes* along a curve is invariant under projection except in the degenerate case when the viewing point lies in the plane of the curve.

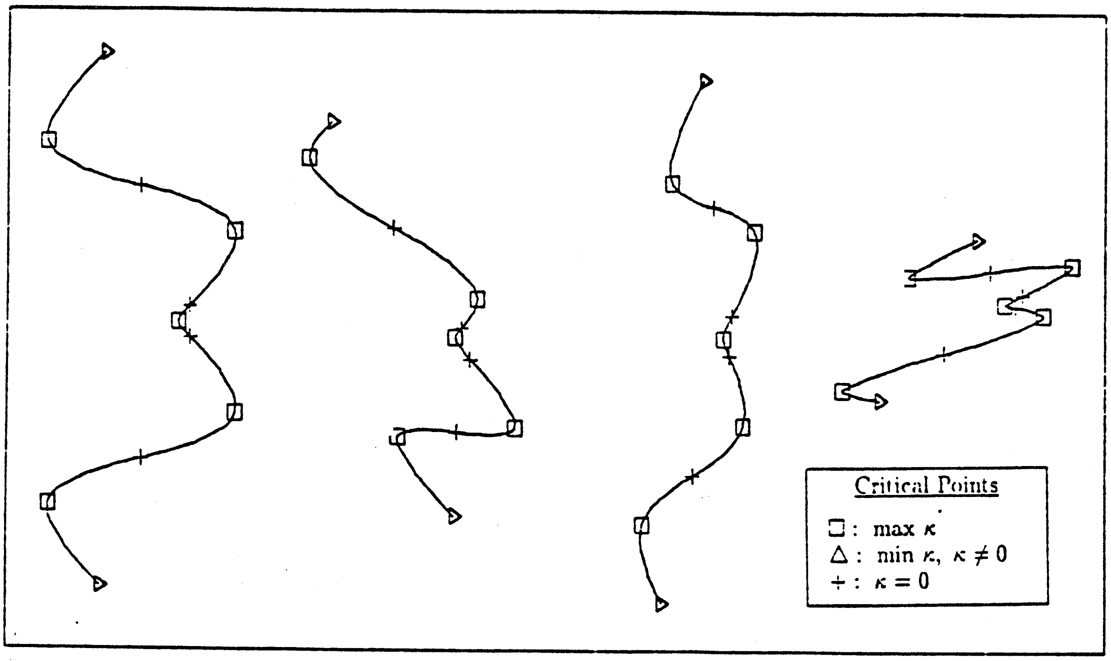
(3) Curvature is a local property which makes it much more suitable for handling occlusion than global curve properties.

(4) Points of maximum curvature of 3-D plane curves project very close to points of maximum curvature of 2-D image curves. The relationship between these points is stable and predictable depending upon viewpoint. Moreover, the relative invariance of these points increases as the curvature increases such that ideal corners almost always project to ideal corners.

This provides sufficient motivation for the use of such a representation. The stability of these curvature critical points under orthographic projection is shown in Figure 30. The processing algorithm can be outlined as follows:

(1) The image curve data is smoothed at multiple scales by gaussian filters and fitted at each scale with a continuous curve parameterization in the form of composite monotone curvature splines.

(2) Curvature critical points are extracted at each scale and stored in a list.



The stability of critical points under orthographic projection. Left. the critical points of a plane curve. On the right, the curve is projected orthographically at various orientations and the critical points of the resulting curves are marked. The stability of their critical points aids in matching the curves to models and estimating their orientation.

Figure 30. Stability of Curvature Critical Points from Different Views (from [95])

(3) Dynamic programming procedures are used to construct a list of critical points which is consistent across the range of scales.

(4) The integrated critical point information is used to match the image curve against the computed critical point information for the 3-D plane curve.

Several example curves are shown in the paper which aid understanding of the approach. Some are shown in Figure 30. No experimental results for matching were available at the time of publication; future research directions were outlined.

Langridge [86] discusses a preliminary investigation into the problem of detecting and locating discontinuities in the first derivatives of surfaces determined by arbitrarily spaced data. Neighbor computations, smoothing, quadratic variation, and the biharmonic equation are all dealt with in this paper. The techniques are useful for detecting roof-edges in range data. Results are shown for two simple synthetic surfaces.

Medioni and Nevatia [99] have written a paper on the description of 3-D range data surfaces using curvature properties. The features used for shape description are the following: (1) the zero-crossings of the Gaussian curvature, (2) the zero-crossings of the maximum principal curvature, and (3) the maxima of the maximum principal curvature. These features are computed by smoothing the depth map with a large window and using one-dimensional windows to compute directional derivatives. They seem to be limiting themselves to the use of generalized cone object model surfaces judging from this paper. Their work is in a beginning stage judging from the experimental results shown in the paper, and it appears to be directed towards object reconstruction rather than object recognition.

Besl and Jain [13] have implemented a surface characterization algorithm which computes surface curvature, critical points, and depth-discontinuity edges as output. Differential geometry tells us that local surface shape is uniquely determined by the first and second fundamental forms [92]. Gaussian and mean curvature combine these first and second fundamental forms (in two different ways) to obtain scalar surface features which invariant to rotations, translation, and changes in parameterization. Therefore, visible surfaces in depth maps will have the same mean and Gaussian curvature from any viewpoint. The two principal curvatures of a surface can be directly computed from Gaussian and mean curvature and vice versa. It turns out that are six fundamental surface types which can be characterized using only the *sign* of the mean curvature (H) and Gaussian curvature (K):

(1) H zero + K zero = plane surface,

(2) H negative + K zero = ridge surface,

(3) H positive + K zero = valley surface,

(4) H negative + K positive = peaked surface,

(5) H positive + K positive = cupped surface, and

(6) K negative = saddle surface.

Gaussian and mean curvature can be computed directly from a smoothed depth map using window operators which give least squares estimates of first and second partial derivatives [12] [23] [59]. Examples of the output of the surface characterization algorithm are shown in Figures 31 and 32 for a torus and an arbitrary surface respectively. These figures show the smoothed depth map, the first derivatives, the square root of the first fundamental form matrix determinant (edge map), the second derivatives, the quadratic variation, the sign (positive=white, negative=black, zero=gray) and magnitude of the Gaussian curvature and mean curvature, the zero-crossings of the first derivatives, the set of all critical points binary image, the magnitude of the second fundamental form matrix determinant, and the set of all non-degenerate critical points binary image. 5x5 window operators were used for derivative estimation. The combination of these images provides a great deal of surface information which can be used to identify surfaces.

4.8. 3-D Object Recognition using Intensity Images

In this section, the existing literature about three-dimensional object recognition systems using intensity images as input will be reviewed. We devote the first part of this section to the ACRONYM vision system; other systems are discussed subsequently.

Brooks [30] [31] [32] explains how model-based three-dimensional interpretations of two-dimensional images are possible using the ACRONYM system. This system is one of the most frequently mentioned vision systems in the computer vision literature. This is probably due to the flexibility and modularity in its design, its use of view-independent volumetric object models, and its domain-independent qualities. Figure 33 shows a block diagram of the ACRONYM system and a diagram of the hierarchical geometric reasoning processing. The system is based on the prediction-hypothesis-verification paradigm. The three main data structures of the system are the following:

(1) *Object Graph:* The nodes of the object graph are generalized cone volume models. The arcs of the object graph correspond to the spatial relationships between the nodes (translation and rotation) and the subpart relations.

(2) *Restriction Graph:* The nodes of the restriction graph are constraints on the volume models for a given object class. The directed arcs of the restriction graph represent subclass inclusions.

(3) *Prediction Graph:* The nodes of the prediction graph are "invariant" and quasi-invariant observable image features of the objects. The arcs of the prediction graph specify the image relationships between the invariant features. These arcs are of the following types: must-be, should-be, and exclusive.

Also, each data object of the system is referred to as a *unit*. Units have *slots* associated with them. For example, a cylinder has a length slot and a radius

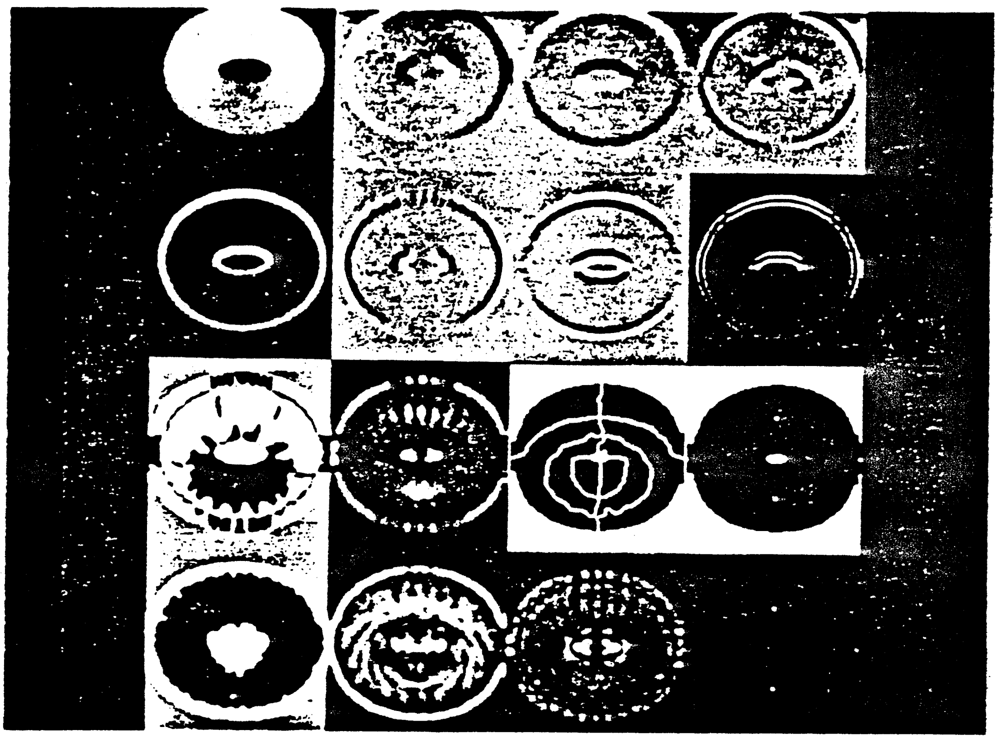


Figure 31. Surface Characterisation of Depth Map of Torus (from [13])

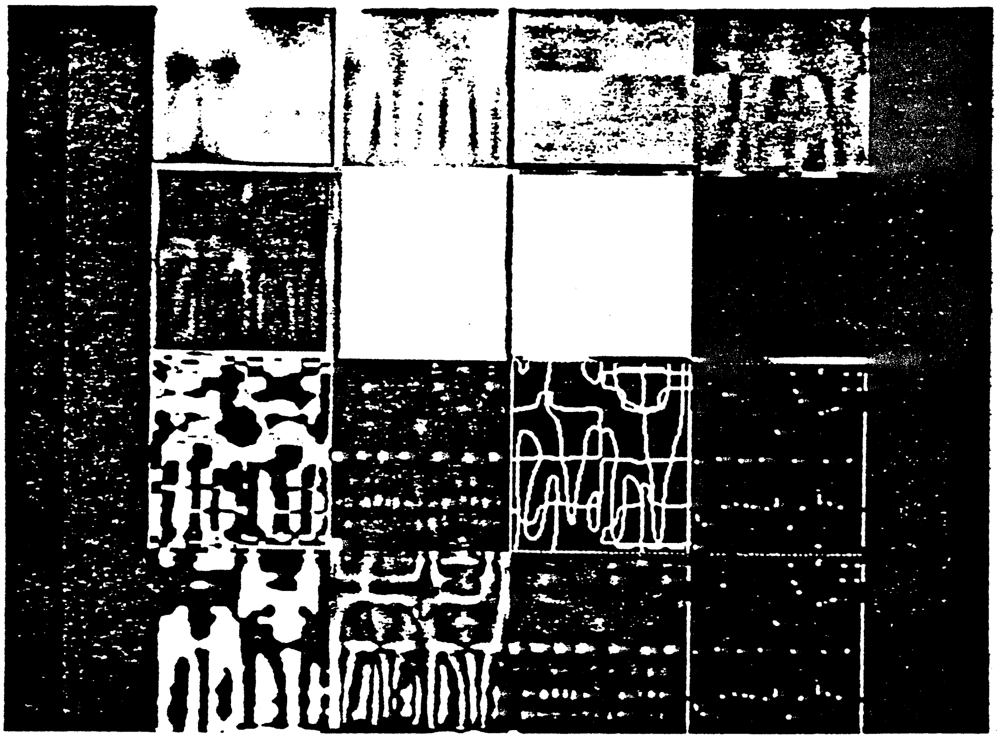
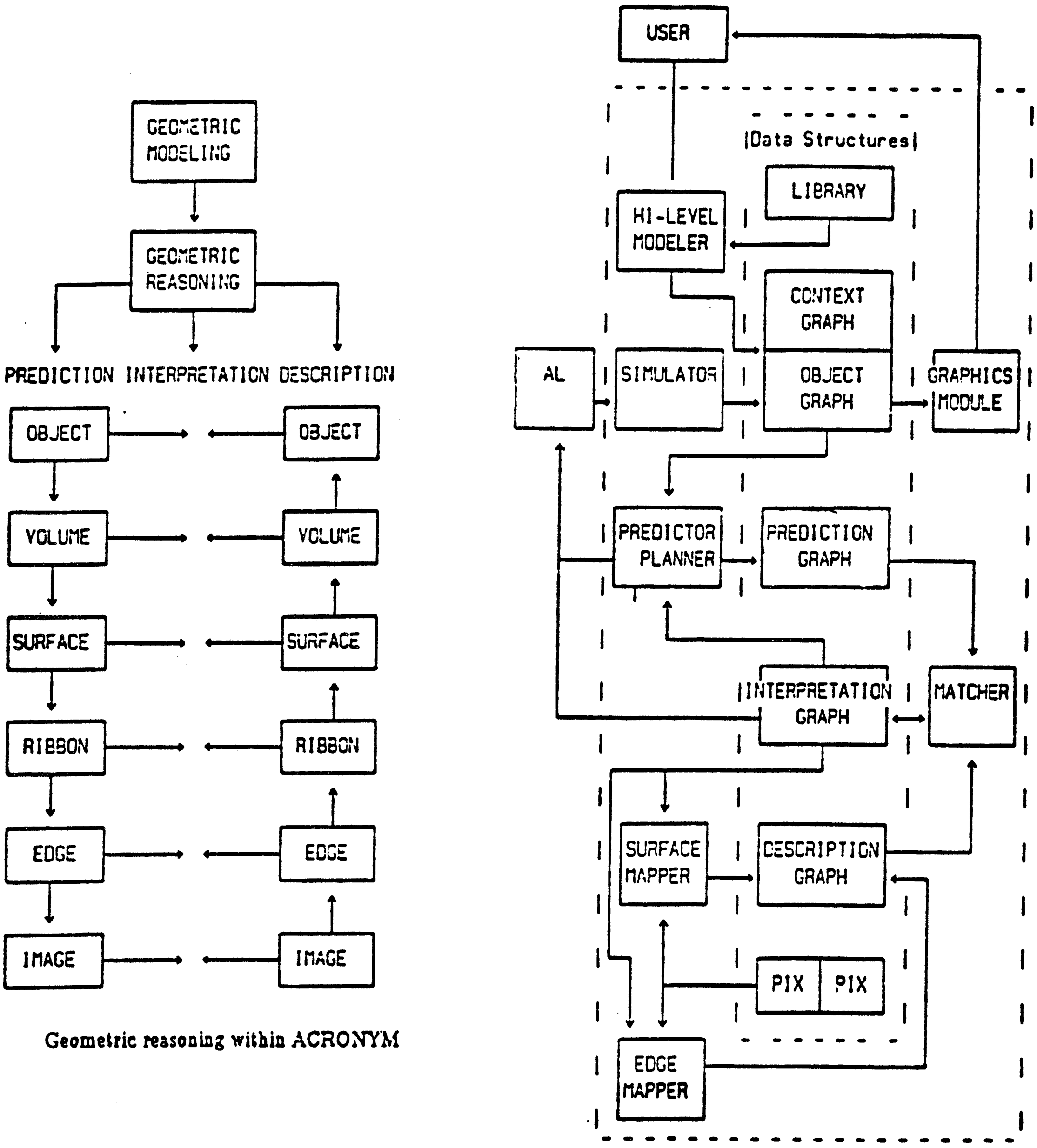


Figure 32. Surface Characterisation of Depth Map of Surface (from [13])



Geometric reasoning within ACRONYM

Figure 33. The ACRONYM System (from [32])

slot. Slots accept numeric *fillers* or *quantifier* expressions.

The ACRONYM system operates approximately as follows:

(1) An a priori world model is given to the system as a set of objects and object classes. An object class is represented as an object with constraints on the dimensions and configurations of the subparts. Each object or object class is a hierarchy of generalized cones, each with its own local coordinate system. An object graph, a restriction graph, and a prediction graph are formed based on knowledge of the world model and a set of production rules.

(2) The system is given an N x N intensity image, a camera model, and the three graph data structures created above.

(3) The image is processed in two steps. First, an edge operator is applied to the image. Second, an edge linker is applied to the output of the edge operator and is directed to look for ribbons and ellipses. Ribbons and ellipses are the 2-D image projections of the elongated bodies and the ends of the generalized cylinder models respectively. All of the higher level 3-D geometric reasoning in ACRONYM is then done based entirely on the 2-D ribbon and ellipse input.

(4) ACRONYM then searches for instances of object models in terms of the ribbons and ellipses. The heart of the system is a non-linear constraint manipulation system (CMS) that generalizes the linear SUP-INF methods of Presburger arithmetic [17] [136]. Constraint implications are propagated "downward" during prediction and "upward" during interpretation. The interpretation matching process is described by Brooks as follows:

> "Matching does not proceed by comparing image feature measurements with predictions for those measurements. Rather the measurements are used to put constraints on parameters of the three-dimensional models, of which the objects in the world are hypothesized to be instances. Only if constraints are consistent with what is already known of the model in three dimensions, then these local matches are retained for later interpretation." [31]

Interpretation proceeds by the combination of local matches of ribbons into clusters. Two consistency checks are performed on the ribbon clusters: (a) each match must satisfy constraints of the prediction graph, (b) the accumulated matching constraints must be consistent with the hypothesized object model.

(5) The final output of the system is the labeled ribbons of the consistent image interpretation. Since orientation and translation constraints have also been propagated during matching, this information should also be available for the labeled ribbons. In this way, objects and' instances of object classes are recognized in a single view intensity image.

Some miscellaneous details of the system are mentioned in [30]. The system is implemented in MACLISP. The predictor subsystem of ACRONYM consists of about 280 production rules. During a typical prediction phase, on the order of 6000 rule firings occur. Rotations and translations operations are treated as strings of matrix operators where the string length is typically ten or more.

Despite the detailed 3-D concerns in the ACRONYM design, no correct 3-D interpretation results have ever been published to our knowledge. Aerial images of jets on runways and jets near airport terminals have been successfully interpreted using ACRONYM. It seems that there are other much less complicated schemes that could yield the same results on aerial images of this type. Binford once wrote that

> "there is no profound reason why ACRONYM could not recognize aircraft in images taken at ground level, although it will probably break when tested on such images because of bugs or missing capabilities that were not exercised previously." [18]

The curious reader is left wondering if this complicated system is as robust as it might seem. The ribbon finding mechanism is usually blamed for the system's difficulties. There are no processing connections between the final decision making mechanism and the original data.

The paragraph above provides a reminder that any open-loop system is only as robust as its most limited component. Even the best possible geometric reasoning system cannot be successful if its input is consistently unreliable and no feedback paths exist. This could possibly be provided by rendering algorithms that relate object models to sensor data.

Of course, there have been many other 3-D object recognition schemes based on intensity images. Mulgaonkar, Shapiro, and Haralick [103] have devised a 3-D scene analysis system that recognizes 3-D objects from a single perspective view using geometric and relational reasoning. *Generalized blobs* (sticks, plate, and blobs) are used to represent the 3-D geometry of objects. Object recognition algorithms work with intensity images that have already been smoothed, thresholded, and segmented to produce a 2-D convex polygon decomposition of image regions. The system can only handle one object in a given view currently. The system performs 2-D image to 3-D object matching directly using constraint propagation and backtracking. All objects are assumed to be in an upright position. The "connect/support" and "triples" relations between 3-D and 2-D primitives plays a fundamental role in the recognition process. Thresholds are employed for measures of circularity and relational error. Seventeen out of twenty-two cases (77%) exhibited successful recognition in the experimental results that used eleven different objects. Camera parameters were estimated from the match to within ten degrees on tilt and twenty degrees on pan. The use of structural relationships is an important feature of this work. One main problem with this method is the object representation method. Detailed geometry is not easily represented and would therefore not be very useful for many applications.

Fisher [49] discusses his data-driven object recognition program called IMAGINE. Surfaces are used as geometric primitives. There are three major stages in the operation of this program:

(1) Image surface regions determined by their region boundary are matched to model object surfaces with the goal to estimate surface orientation parameters. Specific object surfaces are hypothesized.

(2) Hypothesized object surfaces are related to object models constrained by the structural relationships implied by the objects. Specific objects are hypothesized.

(3) Hypothesized objects are verified using consistency checks against constraints due to adjacency and ordering.

The program has four specific goals:

(1) Locate instances of 3-D objects in 2-D images.

(2) Locate image features corresponding to all features of the model OR explain why the image features are not present.

(3) Verify that all features are consistent with the geometrical and topological predictions of the model.

(4) Extract translation and rotation parameters associated with all objects in the scene.

The input to this program is pre-segmented surface regions which have the property that all boundaries between regions correspond to surface or shape discontinuities. The only information used by IMAGINE is the 2-D boundary shape of the segmented surface regions. The object models of the program are surface boundary models where all surfaces are planar or have only a single axis of curvature. Sub-component hierarchies for objects are allowed which determine the joint connections of sub-parts. Model surface to image region matching is performed using a set of heuristics that generate rotation, slant and tilt, distance, and x-y translation in a plane hypotheses. These heuristics gave reasonable results in 94 of 100 test cases. Given the hypothesized surfaces and their position and orientation in space, a set of ten rules is applied to generate object model hypotheses. Another set of rules is applied for object verification that allows for occlusion. Fisher provides his own list of program criticisms which include the following:

(1) The heuristic parameter estimation techniques require mostly planar surfaces.

(2) The program's surface modeling does not account for surface shape internal to the region boundary.

(3) Surface segmentation is currently done *by hand* with the assumption that adequate techniques will soon be available.

(4) Its object models are non-generic.

The program did however achieve its goals of recognizing and locating a PUMA robot and "understanding" its 3-D structure in a test image. Some valuable ideas concerning occlusion were also presented in the paper.

General convex polyhedra are a special object class. Underwood and Coates [148] developed a technique that reconstructs object shape from multiple view intensity images. Edges and planar surface regions extracted from the images are input to the reconstruction algorithm which constructs internal models. No information about viewing parameters for the different views is given. The internal object model used by the algorithm is topological and considers only relationships between surfaces. Different view models are matched and a complete topological object model is constructed using a graphical learning tree. These object models can then be used for object recognition. Experimental results for 20 test views matched against an object library of 19 objects yielded an 18 of 20 success rate. Extensions to more general objects are suggested in the paper. The limitations of the implementation are mainly due to incomplete use of spatial information.

Lee and Fu [87] [88] propose a design for a general computer vision system that would be capable of object recognition in a single image. The system design and system flow chart are shown in Figure 34. They are interested in creating a system that allows for the proper interaction of top-down (model-guided) analysis and bottom-up (data-driven) analysis. The proposed system consists of the following six components:

(1) General Purpose Primitive and Relation Extractor which uses no higher level knowledge: Input = Input Image + Requests for More Evidence from

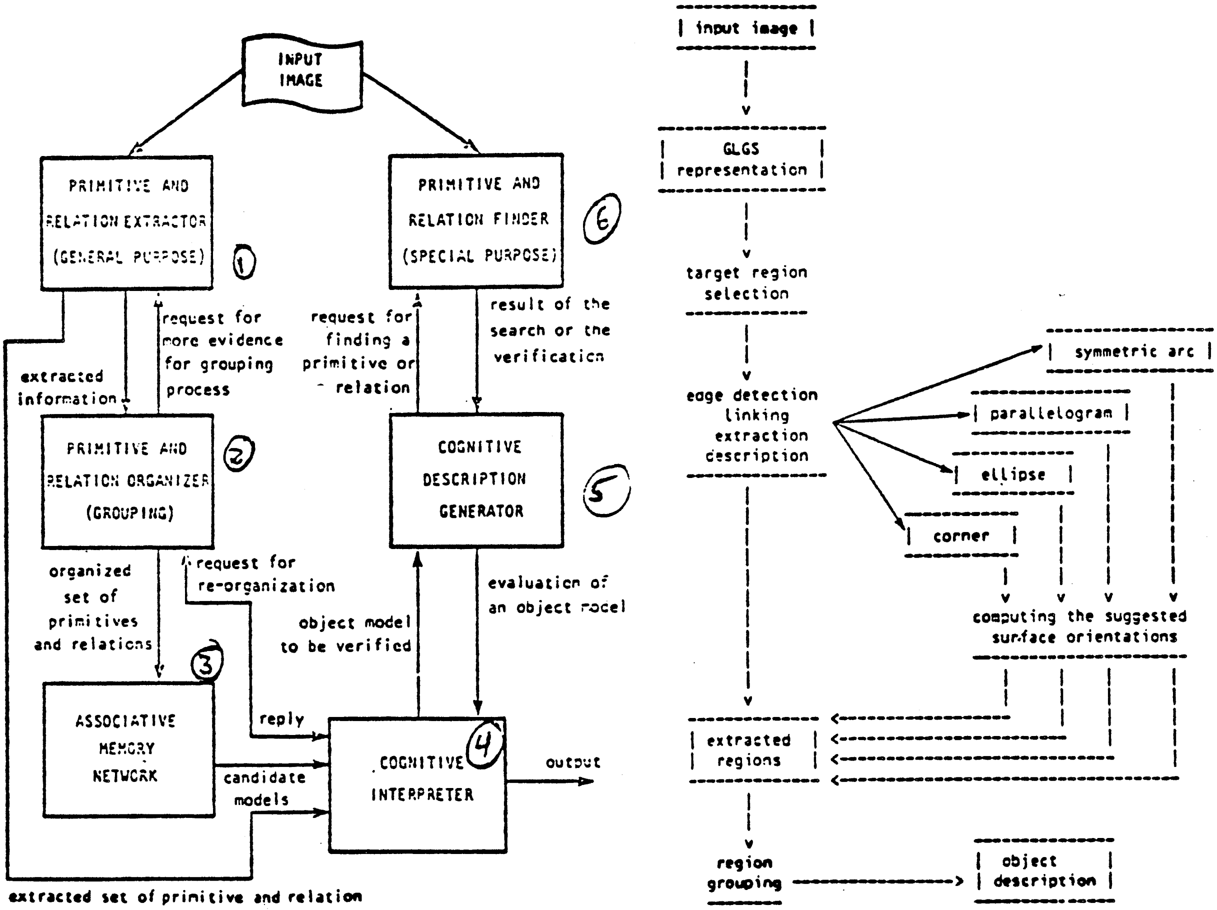

INPUT IMAGE
PRIMITIVE AND RELATION EXTRACTOR (GENERAL PURPOSE)
1
PRIMITIVE AND RELATION FINDER (SPECIAL PURPOSE)
6
request for more evidence for grouping process
request for finding a primitive or relation
result of the search or the verification
extracted information
PRIMITIVE AND RELATION ORGANIZER (GROUPING)
2
COGNITIVE DESCRIPTION GENERATOR
5
organized set of primitives and relations
request for re-organization
object model to be verified
evaluation of an object model
3
ASSOCIATIVE MEMORY NETWORK
reply
candidate models
COGNITIVE INTERPRETER
4
output
extracted set of primitive and relation
input image
GLGS representation
target region selection
edge detection linking extraction description
symmetric arc
parallelogram
ellipse
corner
computing the suggested surface orientations
extracted regions
region grouping
object description

Figure 34. Design of Vision System and Flow Chart (from [87])

Grouping Processor (2), Output = Extracted Primitive and Relation Information for (2);

(2) Primitive and Relation Organizer for Grouping Process: Input = Output from (1) + Requests for Reorganization from the Cognitive Interpreter (4), Output = Organized Set of Primitives and Relations;

(3) Associative Memory Network with Knowledge of World Model Objects: Input = Output from (2), Output = Candidate Object Models Compatible with Input Primitives and Relations;

(4) Cognitive Interpreter: Input = Output from (1) + Replies from (2) + Object Models from (3) + Object Evaluations from Cognitive Description Generator (5), Output = Object Models to be Verified for (5) + Requests for Reorganization for (2) + Final Output Image Description for System User when decision-making processing terminates;

(5) Cognitive Description Generator: Input = Object Models to be Verified from (4) + Results of Verification Search from Special Purpose Image Processor (6); Output = Request for Finding a Particular Primitive or Relation for (6) + Evaluation of an Object Model for (4);

(6) Special Purpose Primitive and Relation Finder: Input = Input Image + Requests from (5) Output = Results from Verification Searches.

Note the verification feedback from the original image and the *multi-level control-logic* interactivity of the different components in this design. The processing can be considered as three basic processes: object description generation, model retrieval, model verification. The two papers by Lee and Fu concentrate on object description generation which is now briefly described. First, images are converted to the Gray Level Geographic Structure (GLGS) representation. Target regions are selected corresponding to the maximum of the "conspicuousness" function. Extracted edges in target regions are classified as one of the following: 1) parallelogram, 2) ellipse, 3) skewed-symmetric arc, or 4) corner. Regularity constraints are then applied.

(1) Parallelograms are always projections of rectangles.

(2) Ellipses are always projections of circles.

(3) Skewed-symmetric arcs are always projections of a symmetric planar curve.

(4) Corners are always intersections of orthogonal line segments.

These regularity constraints and the so-called "least-slant-angle" preference rule are used to compute 3-D surface orientations of the selected target regions. Local interpretations of regions are propagated to neighboring regions stored in the edge-region adjacency graph. Constraints and consistency checks interact to yield an rough object description in terms of visible surface orientations. Experimental results are shown for a car and a machine shop tool. The final output is a line drawing where each surface is drawn with a slant and tilt vector for its normal. These results looks fairly good for these two objects. This system is extremely limited due to its use of only four geometric primitive extracted edges and its use of the right angle assumption throughout. It also cannot handle curved surfaces in consistent manner. The general preliminary thought has been followed by a non-general implementation.

Chakravarty and Freeman [35] have developed a technique that uses characteristic views as a basis for three-dimensional object recognition using intensity image data. The set of all possible perspective projection views of an object is partitioned into a much smaller set of characteristic views which form topological equivalence classes. Different views within an equivalence class can be obtained from one another using linear transformations. The number of characteristic views is reduced still further by allowing objects to have *only stable orientations* as positioned on a planar surface. Matching is performed using line-junction labeling constraints on detected edges. The method requires silhouette determination to guide the matching process (this is a disadvantage for occlusion handling), and it produces position and orientation information as output in addition to identifying objects. A system structure diagram is shown in Figure 35.

Some 3-D object recognition techniques are based purely on object silhouettes. These methods cannot distinguish between objects that have the same set of silhouettes, of course. McKee and Aggarwal [97] have worked on recognizing three-dimensional curved objects from a partial silhouette description. Three-dimensional object models are not used, however. The system learns the global silhouette boundary description during the training process for each view of an object and stores the description in an object-view library. The recognition algorithm accepts a partial boundary description and produces a list of all the objects in the library that could have produced the view. This

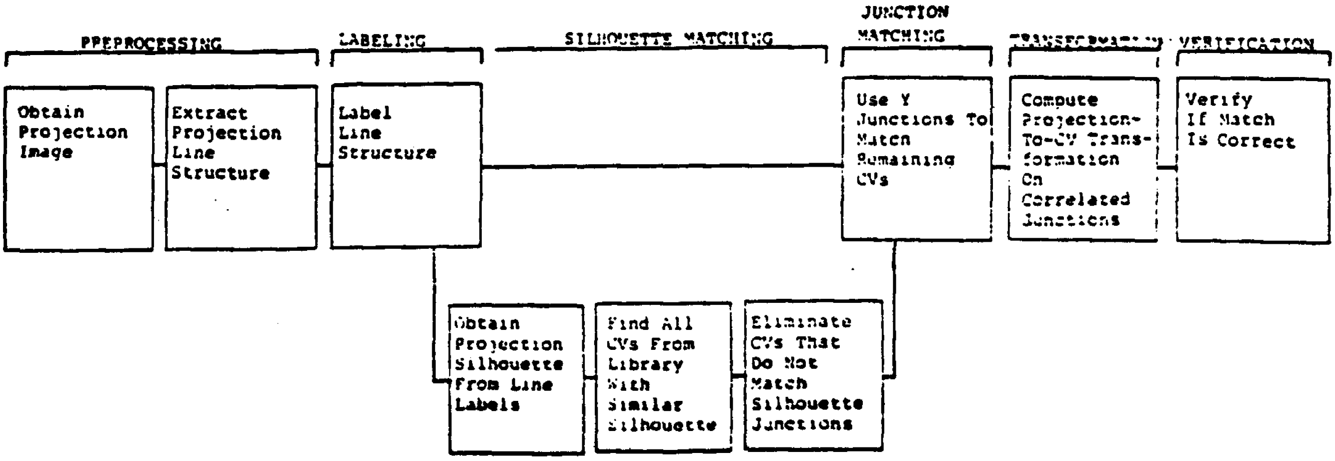

PREPROCESSING
LABELING
SILHOUETTE MATCHING
JUNCTION MATCHING
TRANSFORMATION
VERIFICATION
Obtain Projection Image
Extract Projection Line Structure
Label Line Structure
Use Y Junctions To Match Remaining CVs
Compute Projection-To-CV Transformation On Correlated Junctions
Verify If Match Is Correct
Obtain Projection Silhouette From Line Labels
Find All CVs From Library With Similar Silhouette
Eliminate CVs That Do Not Match Silhouette Junctions

Figure 35. Recognition Scheme using Characteristic Views (from [35])

work did not include view-independent processing of any sort and had problems with noisy edges.

Wallace and Wintz [150] have used global 2-D shape descriptors to recognize three-dimensional aircraft shapes by matching against a stored library of shape descriptors. One shape descriptor set is computed and compressed for each viewing angle in a finite set that covers the entire spherical solid angle giving the system view-independence. Given an arbitrary view of some known aircraft, 2-D shape descriptors are computed for the silhouette and matched against *each* precomputed view description in the library of shape descriptors for *each* possible aircraft. Since the entire outline of an aircraft is available at sufficient resolution in this application, global Fourier Boundary Shape descriptors can be used which provide excellent results. Research still continues for a similar technique for partial shape description and recognition.

Global silhouette shape *moment-based* description techniques have also been used [42] and continue to be used [121] for aircraft shape description. Libraries can be used for 3-D recognition in much the same way as mentioned above.

Wang, Maggee, and Aggarwal [152] also match three-dimensional objects using silhouettes, but their method is somewhat different. For each prototype object, the principal axes, the principal moments, and the Fourier Boundary Shape descriptors of the three primary silhouettes (silhouettes as viewed from each of the three principal axes) are computed and stored in a library. (They use 3-D models constructed using the approach developed by Martin and Aggarwal [96], but any 3-D model reconstruction approach could be used in practice. We note that the approach of Martin and Aggarwal can be duplicated readily using commercial solid modelers (such as GEOMOD [52]); object models are constructed by the intersection of extended silhouette profiles.) For each unknown object, at least three silhouettes from different views are required. These silhouette boundaries are combined to produce an object from which the principal moments and Fourier shape descriptors are computed. These quantities are then matched against the stored library quantities. The convergence of the descriptors as indexed by the number of silhouettes is dependent on the object and the viewing locations. Note that more input data and more computation is required, but less searching is needed to identify objects.

Bolle et al. [22] [23] [34] describe an approach to intensity-image-based object recognition where objects are modeled as consisting of Lambertian surface patches of planes, cylinders, and spheres. The assumption is that 85% of manufactured parts are well-represented by combinations of such models [58]. This work is innovative in its use of quadric picture functions. These functions are analytically computed quadric intensity functions that combine the primitive Lambertian surface shapes with primitive point-source-at-infinity illumination model parameters. Given an intensity image, the image is partitioned into small square windows which are fitted to quadric surfaces. Each window is classified as a piece of a sphere, cylinder, plane, or none of the above using an asymptotically Bayesian recognizer (yields minimum-probability-of-error as the window size gets large), and the 3-D surface parameters are estimated. When one of the surfaces is present in the image, the surface parameters for that surface cluster together in the parameter space. These clusters can be detected which infer the existence of a surface of the given type at the appropriate location and orientation in space. Some work has been done for handling windows with two different surface types present. Experimental results using 65x65 windows are shown for synthetic and real spheres and cylinders. Although this approach contains many good concepts, the implementation of only a few surface types is very limited and will not be very useful if 65x65 windows are needed. Once again we have seen the use of image window quadric surface fitting so that surfaces can be matched to quadric object surfaces.

Fang et al. [44] and Stockman and Esteva [139] address a constrained 3-D object recognition problem. Although they refer to their work as 3-D, they are technically addressing a 2-D estimation problem using 3-D techniques. They address the problem of determining the (x,y) location and the *single* rotation angle of a 3-D polyhedral object sitting stably on a flat plane using a single view intensity image and polyhedral object models. They extract important edges and points as primitive features from the input image. Geometric constraints and model matching of grouped primitives are used to determine possible translation and rotation parameters which are then accumulated in a 3-D histogram. Histogram clusters are detected which identify a particular object at a particular (x,y) location rotated by some particular angle. Perfect experimental recognition results for five toy objects were obtained in [44]. This sort of transformation clustering technique could possibly be useful for general 3-D object recognition, but this is not shown in this paper. Smooth objects are certainly not handled well by this method.

Tropf and Walter [145] discuss a augmented transition network (ATN) model for the recognition of randomly oriented 3-D solid objects with known geometry using single images. ATN models were developed in the field of natural language understanding and are used in this work to control an analysis-by-synthesis search procedure that is based on hypothesis generation and verification. The method is explained in the paper with the following example:

(1) Assume that only point primitives (such as edge-less corners) are used, and assume that an object is described using a set of points that are rigidly connected to each other.

(2) A parallel projection of the points is created from an arbitrary view. You pick a point from your projected data and hypothesize that it is a point P1 on some particular known object from your object library.

(3) That object has a second point P2 which is a distance R from P1. In 3-D, P2 must lie on the surface of a sphere of radius R; in the 2-D projected image, P2 must lie within a circle of radius R. Next, you choose a second

point within that circle if one exists. (Otherwise, try another second point of the object. If none of those work, go on to the next object.)

(4) Now it is assumed that one knows the axis P1-P2 in 3-D space (to within a small near-far ambiguity that can be checked). Next, we consider a third point P3 on the object not lying on P1-P2. It must lie on some 3-D circle surrounding the P1-P2 axis and must therefore lie on a known ellipse in the projected image.

(5) You now pick a point in the image closest to the ellipse that will fix the object in space. Object verification is the next step which completes the example case.

In some respects, this is similar to the RANSAC approach [26] because it uses a few randomly selected data points to estimate model parameters and relies on verification for a better fit. 3-D polyhedral object models and hidden-line algorithms are used by the system. The ATN itself consists of states, arcs, a dictionary, named registers, actions, and conditions. It is claimed that the ATN approach differs from block world approaches in that it can cope with heavily distorted data. No experimental results are given as the system was being implemented at the time the paper was written.

Douglass [39] [40] developed a system, written in SIMULA, for interpreting outdoor 3-D scenes using a 3-D model-building approach. Heuristic visual inference routines interpret perspective, shadows, highlights, occlusions, shading, texture gradients, and monocular motion parallax from multiple images. The placement routine at the heart of the system forms intensity image segments into 3-D surfaces that are iteratively refined by the various parts of the system. Image segmentation is performed using a "recognition cone" and a region growing algorithm.

Work by Goad [54], Silberberg et al.[137], and Schneier [135] is reviewed in [43]. Goad [54] uses a multiple-view object feature model that incorporates 218 different 3-D views of each object. The features are line segments which are stored as a pair of endpoints and a 218-bit bit-string that describes the visibility of that feature in each of the discrete views. Edges for objects are ordered by their expected utility for matching purposes. Silberberg et al. [137] use a generalized Hough transform to match observed line segments with model line segments for each viewpoint. Schneier [135] uses a "graph of models" where each node represents a 3-D surface primitive and contains a set of properties which describe that surface shape and a set of pointers to the model names in which the surface is used. The arcs between the nodes describe relationships between surfaces and contain pointers to the model names where those relationships occur. The integration of multiple objects into a single shared data structure provides a compact representation that can be easily indexed.

We conclude this section on an historical note. The pioneering work of Roberts [125] was a 3-D intensity image based object recognition system. The objects were constrained to be blocks, wedges, prisms, or combinations thereof.

The Roberts' cross operator was used to detect edges, and collinear segments were merged into lines to produce a line drawing of the scene. Regions were then classified to be triangles, quadrilaterals, and hexagons. These regions were matched to faces of the prototypes objects. Possible object part model matches were rendered using a hidden-line algorithm to verify the correct object match! Recognized object parts were cut away from the image, and the same process was repeated until all detected edges and vertices were explained. After identifying an object, the system could draw the object from any view to demonstrate its understanding of the object shape. This work was followed by the more advanced work of Guzman [57], Waltz [151], and others which concentrated on line-edge and edge-junction labeling for detecting polygonal regions. This early work addressed many of the fundamental problems encountered in computer vision but was essentially limited to high quality images of block world scenes. The algorithms were not robust enough to handle scenes from the real world with noise, curved objects, etc.

4.9. 3-D Object Recognition using Depth Maps

In this final section of the literature review, the existing literature concerning three-dimensional object recognition using depth maps is reviewed. It is our opinion that this research area has a great deal of potential for many applications.

Nevatia and Binford [105] is probably the first paper published concerning object recognition in range data. The emphasis in this paper is on the analysis of scenes containing curved objects, which are represented as sub-part hierarchies of generalized cones (cylinders). The data-driven recognition processing of this approach can be summarized as follows:

(1) Range image edge and region features are extracted and organized to create image object descriptions which are structured and symbolic.

(2) Important features of these object descriptions are used to index into a library of object models to retrieve a set of models which are similar to the objects in the image.

(3) The image object description is compared to each of the retrieved models and the best match is chosen.

(4) Verification is performed to see if the differences in the best retrieved object model and the image object description are reasonable. (This step was not implemented.)

Experimental results are discussed for a doll, a horse model, a glove, a ring, and a snake-like object. Objects with different structure were easily distinguished and even moderate amounts of occlusion were handled successfully. This work does not seem to have directly evolved into any more recent range data systems.

Kuan and Drazovich [83] have developed a system which attempts to extend the principles of the ACRONYM approach to range imagery. They use generalized cylinder object models with model priorities and subpart attachment relations to yield multi-level coarse-to-fine object descriptions. They use a model-driven prediction module which predicts the following features at different levels to enable coarse-to-fine multi-level interpretation:

(1) Object Level: These features include spatial relationships among object components, overall dimensions, extreme points, side view characteristics, and occlusion relationships among object components.

(2) Cylinder Level: This level is the most important level because cylinders are the basic symbolic entity of the object description. These features include cylinder contour, cylinder position and orientation, parallel edge relationships, edge types, cylinder length, extent of overlap with other cylinders, and overall cylinder visibility and obscuration.

(3) Surface Level: These features include information as to whether the surface is planar or curved, surface edge boundary information, and spatial surface relationships.

(4) Edge Level: These features include information as to whether the edge is occluding, convex, or concave.

These predictions give guidance to the low-level feature extraction processes and they also provide mechanisms for feature-to-model matching and interpretation. In contrast to the ACRONYM system, actual measured features are used for matching based on maximizing likelihood rather than creating constraints for later constraint propagation processing. The major components of their system are diagrammed in Figure 36. Experimental results are discussed for *one* synthetic 64x64 range image of a missile launcher decoy and the *two* object models shown in Figure 36. The system identified the decoy correctly in this single test. The overall approach seems quite reasonable; it is unfortunate that more experimental results were not discussed. The system does however inherit the limitations of ACRONYM's object models.

Smith and Kanade [138] discuss a program designed to produce object-centered three-dimensional object descriptions from depth maps. Conical and cylindrical surfaces are used as primitives. The object descriptions derived from their bottom up approach could be used for matching and object recognition. Coherent relationships between sub-cylinders of parts are used to aid the extraction of object surfaces. An example of this coherency is the relationship between the handle of a pan and the main body of a pan. Experimental object *description* results are shown in the paper for cups, pans, and toy shovels which exhibit this coherency. Results for one scene are shown in Figure 37.

Gennery's [51] main concern for object recognition was obstacle avoidance for autonomous vehicle navigation. His algorithm can be summarized as follows: First, find the ground surface which is usually flat. Second, segment objects above the ground by clustering all range data points more than a certain threshold distance above the ground. Next, ellipsoids are fit to these clusters and then clusters are adjusted according to the ellipsoid fits. He argues

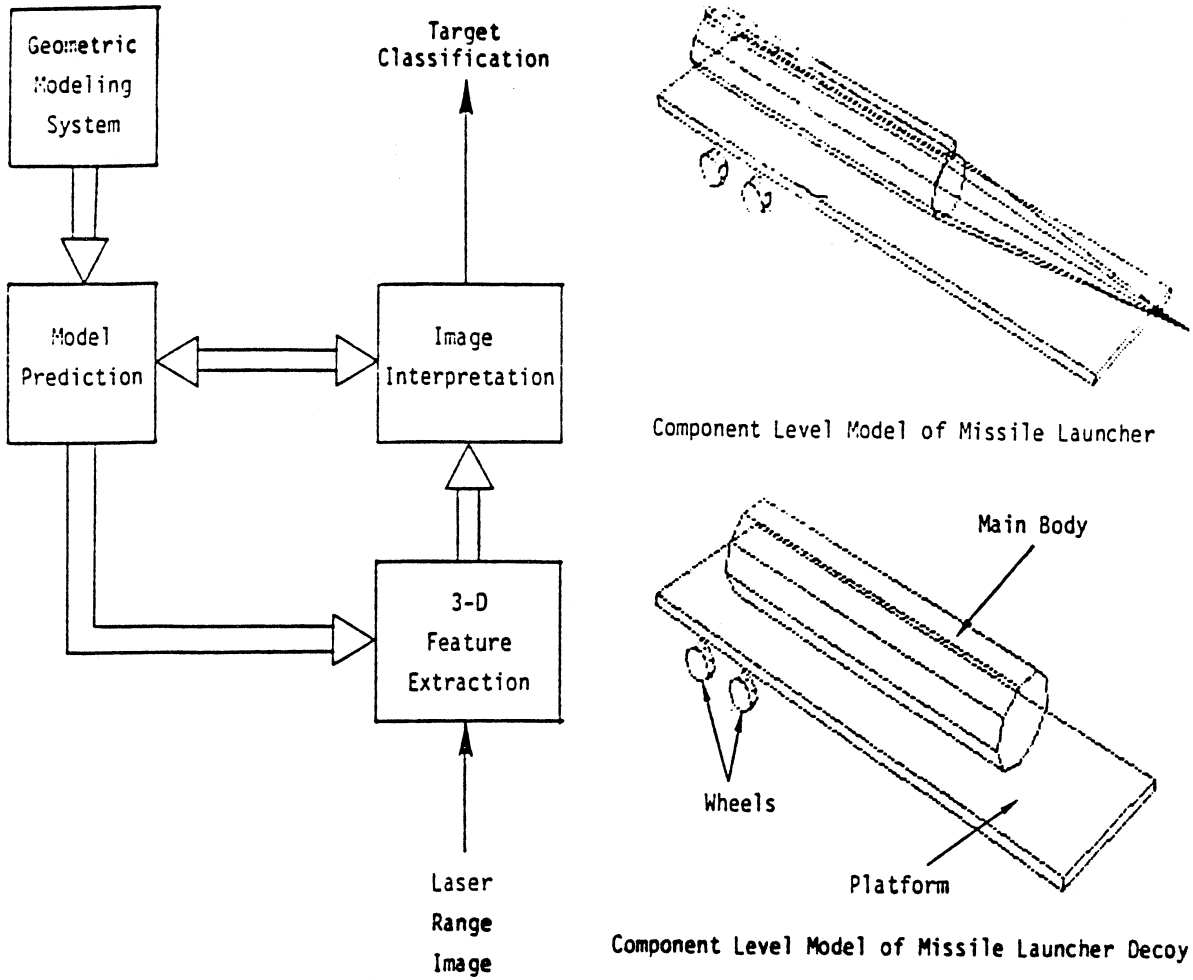


Figure 36. Object Classification System and Two Models (from [83])

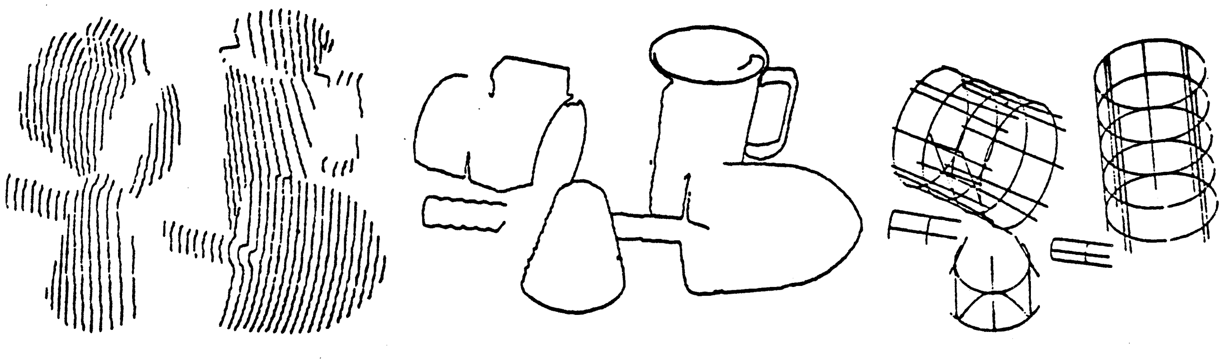
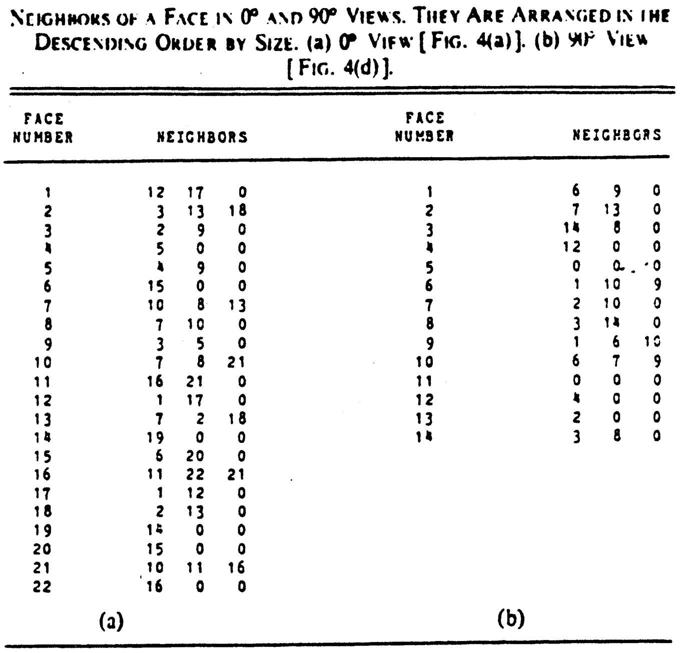


Figure 37. Light Stripes, Contours, and Object Descriptions (from [138])

that although ellipsoids are very crude object representations, a large scene containing many objects is fairly well described by sets of ellipsoids for the purposes of navigation. Experimental results are shown for a pair of stereo pictures from the Viking Lander which landed on Mars.

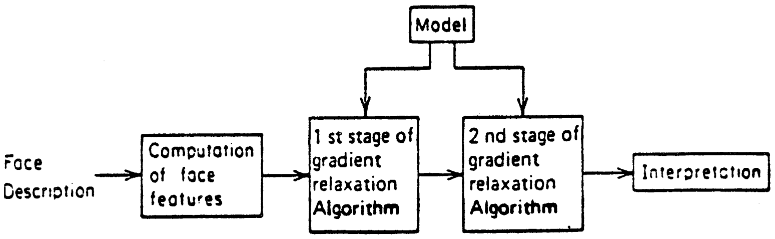
Bhanu [15] [16] presents a complete 3-D scene analysis system for recognizing 3-D objects in depth maps. The system uses the object representation and surface extraction method discussed by Henderson [62]. It constructs object models from physical prototypes using multiple view depth maps. The results for a complex curved-surface automobile part are shown. 8314 3-D object surface points are obtained by transforming points from 14 individual views into a common object centered coordinate system. These surface points are used to fit a convex-faced polyhedron using a two step algorithm: 1) the three-point seed algorithm is used to group all points into face regions using convexity and narrowness tests (four threshold values needed for this), 2) the face regions are then approximated by 3-D planar convex polygons. For the auto part, 85 flat faces are computed to describe the curved surface part. Object recognition is accomplished after model determination as follows. A depth map from an arbitrary view (same scale) is acquired using a range-finder. The object points are segmented from the background and a polygonal face approximation of the object surface is computed using the same technique mentioned above for model determination. This generates approximately 10-25 faces for unknown views of the auto part. These faces are used to perform object matching using a relaxation-based scheme called stochastic face labeling. The face features of area, perimeter, peround, length of maximum, minimum, and mean radius vectors from the face centroid, number of vertices, and angle between maximum and minimum radius vectors are used to compute the initial stochastic labeling probabilities. (A feature weighting vector is also used.) In addition, a face neighbor table is computed where neighbors are ranked according to area. An example of a face neighbor table is shown in Figure 38. A first stage iteration is performed which involves maximizing the first stage compatibility measure which is defined in terms of a one-largest-area-neighbor compatibility function. Using the labels at the end of the first iteration, a second stage iteration is then performed which involves maximizing the second stage compatibility measure which is defined in terms of a two-largest-area-neighbor compatibility function. The iterations are indicated in the diagram in Figure 38. Both compatibility functions use the following quantities: the distance between neighboring face centroids, the ratio of the areas of neighboring faces, the difference in face orientations, and the rotation angle for the maximum intersection area of coplanar faces. (These quantities are also weighted.) At the end of the second stage, translation and rotation information concerning the object can be computed. It is implied, but not shown, that object recognition is possible by choosing the object among several different prototype object models which maximizes the compatibility measures. The method appears to be general in that it handles arbitrary viewpoints. However, it does seem to rely very heavily on the consistency of the output from the face-finding algorithm. It also seems that some face adjacency information is being ignored in the two



NEIGHBORS OF A FACE IN 0° AND 90° VIEWS. THEY ARE ARRANGED IN THE DESCENDING ORDER BY SIZE. (a) 0° VIEW [FIG. 4(a)]. (b) 90° VIEW [FIG. 4(d)].

FACE NUMBER	NEIGHBORS			FACE NUMBER	NEIGHBORS		
1	12	17	0	1	6	9	0
2	3	13	18	2	7	13	0
3	2	9	0	3	14	8	0
4	5	0	0	4	12	0	0
5	4	9	0	5	0	0	0
6	15	0	0	6	1	10	9
7	10	8	13	7	2	10	0
8	7	10	0	8	3	14	0
9	3	5	0	9	1	6	10
10	7	8	21	10	6	7	9
11	16	21	0	11	0	0	0
12	1	17	0	12	4	0	0
13	7	2	18	13	2	0	0
14	19	0	0	14	3	8	0
15	6	20	0				
16	11	22	21				
17	1	12	0				
18	2	13	0				
19	14	0	0				
20	15	0	0				
21	10	11	16				
22	16	0	0				

(a) (b)


Model
Face Description
Computation of face features
1 st stage of gradient relaxation Algorithm
2 nd stage of gradient relaxation Algorithm
Interpretation

Block diagram of the 3-D shape matching algorithm.

Figure 38. Face Neighbor Table and Diagram of Matching Algorithm (from [15])

stage matching algorithm. No recognition test results are given in the paper.

Ballard and Sabbah [8] have investigated viewer independent shape recognition by factoring an image object description into an object-centered view-independent description and a view-dependent view transformation. They emphasize a decoupling of the three subgroups of scale, orientation, and translation parameters. It is assumed that a planar surface patch (polyhedral) description of an object is available to them both as a known prototype model and as sensor data from either processed range data or other sources. They also assume that scale is already known and that the orthographic projection approximation is valid. Their 3-D algorithm consists of two main sequential processing steps: 1) Use the Generalized Hough Transform (GHT) to compute the three 3-D rotational parameters corresponding to a given view and a given object, and 2) Determine the two 2-D translational parameters via another GHT. If the correct object is not being matched, only inconsistent interpretations will result. When an unknown view of an object is matched against the correct object, a consistent interpretation is output. They give no experimental results for real scenes, but they do show results for four synthetic experiments. It is assumed there will only be one object per view (image). Their approach is interesting because they determine how an object is oriented (rotation parameters) before they determine where it is (translation parameters).

Bolles et al. [24] present a system for recognizing and locating three-dimensional parts in range data which extends previous "local-feature-focus" ideas [25]. Some of their object recognition ideas are quite different from most other researchers:

(1) They prefer to use moderately complex parts instead of polyhedra or quadric surface models because they have found that the abundance of features are helpful for object recognition. They point out that most industrial parts are moderately complex; very few ideal spheres, cylinders, and polyhedra are used untouched as industrial pieces.

(2) They also express that only very few features should be used for matching, hopefully only 2 or 3 if possible. For example, if a dihedral edge is found in range data, all six degrees of freedom (3 position and 3 orientation) of that edge are determined except for one (the position along the edge). A preliminary planning system should do as much processing as is required up front to select the best features since this computation only needs to be done once.

The recognition process is partitioned into five steps:

(1) Primitive Feature Detection: Range edges are detected and linked using two separate techniques: one based on discontinuities and the other based on significant second derivative zero-crossings.

(2) Feature Cluster Formation: For example, coplanar edges can be grouped together. Circular arcs can be isolated among the coplanar edges.

(3) Hypothesis Generation about Possible Objects and Locations: For example, the system can hypothesis objects which contain the circular arc as an edge which are appropriately positioned in space.

(4) Hypothesis Verification of Best Object Hypotheses: Objects are checked to see if additional features of each object can be found in the image data or the primitive features already extracted.

(5) Parameter Refinement to Obtain More Precise Information: If additional features are predicted and found, this information can be averaged with the existing information to yield more precise locations.

3DPO uses an extended CAD model to represent objects. A volume-surface-edge-vertex model is extended via the addition of redundant pointers and other data structures to support matching. Figure 39 represents these ideas. The ideas expressed in this paper seem reasonable. Unfortunately, the system was still in development at the time and no real experimental recognition results

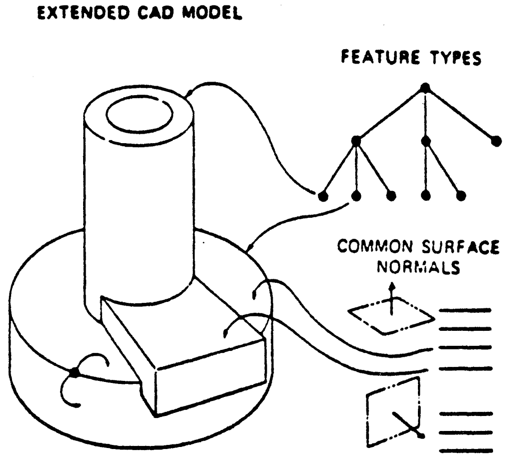
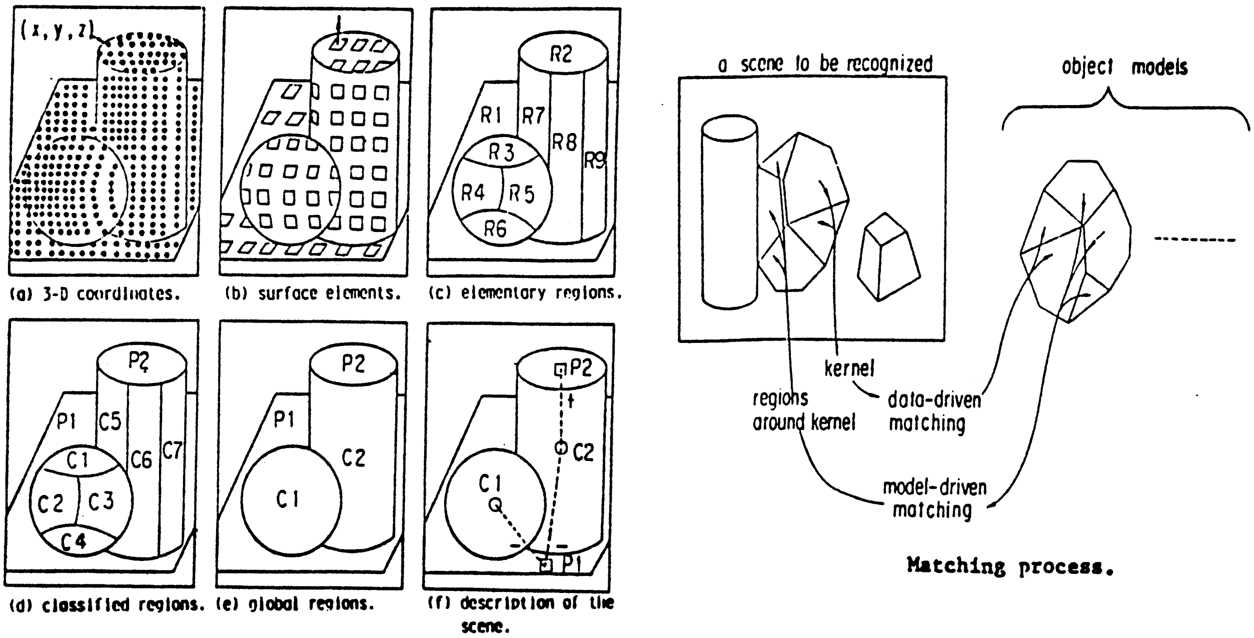


Figure 39. Extended CAD Model adds Redundancy for Matching (from [24])

were available. However, several figures are shown with hypothesized wireframe objects overlaid on the range sensor's light stripe image which look very reasonable.

Oshima and Shirai [113] [114] have also discussed object recognition using three dimensional information. Their object recognition system is based on depth maps obtained from a light stripe range finder. The range data is processed as follows: 1) Points (range pixels) are grouped into planar surface elements, 2) Surface elements are merged into elementary regions which are classified as planar or curved, 3) Curved elementary regions are merged into consistent global regions which are fitted with quadric surfaces, and 4) A scene description is generated using global region properties and their relationships with each other. This process in shown on the left in Figure 40. The region properties are based on the best-fit planar region and its boundary and include the following quantities: perimeter, area, peround, minimum, maximum, and mean region radii about the region centroid, and the standard deviation of the radii of the boundary. The region relationships are characterized by distance between region centroids, the dihedral angle between best-fit planes, and the type of intersection curve between the regions. There is a learning process which must be executed *for each view* of each object which is to be recognized. The recognition process compares unknown scene data against learned scene data. Matching is restricted using a algorithmically selected kernel region which has a principal part and a subordinate part. The kernel is matched against each learned scene, and each good match is processed further until a consistent scene description is generated. The matching process is indicated on the right in Figure 40. Two experiments were performed, one using simple objects bounded by only planar or quadric surfaces and the other using machined parts. No bad scene interpretations resulted using an empirically



(a) 3-D coordinates. (b) surface elements. (c) elementary regions.

(d) classified regions. (e) global regions. (f) description of the scene.

Matching process.

Figure 40. Surface Characterization and Matching Process (from [113])

determined set of thresholds for the matching algorithm. The technique is worthwhile because it can handle many objects at once. However, matching will be significantly slowed down when many possible views are allowed because of the view-dependent nature of the stored scene models. Thus, it appears to be inadequate for single arbitrary view object recognition.

Sato and Honda [130] have investigated pseudodistance measures for recognition of objects which can placed on a turntable in a stable vertical orientation. A fixed set of horizontal cross-section boundaries is determined for each object to be recognized using a laser projection system and image processor as described in [131]. Boundary-based Fourier shape descriptors are computed for each horizontal cross-section. The object representation then consists of N sets of M complex Fourier coefficients. Pseudodistance measures between two objects representations are defined for elongatedness, horizontal strain, section shape, torsion, and displacement. Experimental distance results are shown in the paper for four wood animal models and a doll in three different positions. Two positions of the doll are shown in Figure 41. The boundaries described by Fourier descriptors are shown below each image. Using a weighted sum of pseudodistance measures and, for example, a minimum distance classifier, unknown curved shapes can be classified. One problem with this method as currently implemented is that disjoint parts of the doll's cross-sections had to be linked manually to create a simple closed curve usable by the Fourier Descriptor algorithm. This system is totally inadequate for single arbitrary view object recognition because of its need for 360 degree view.

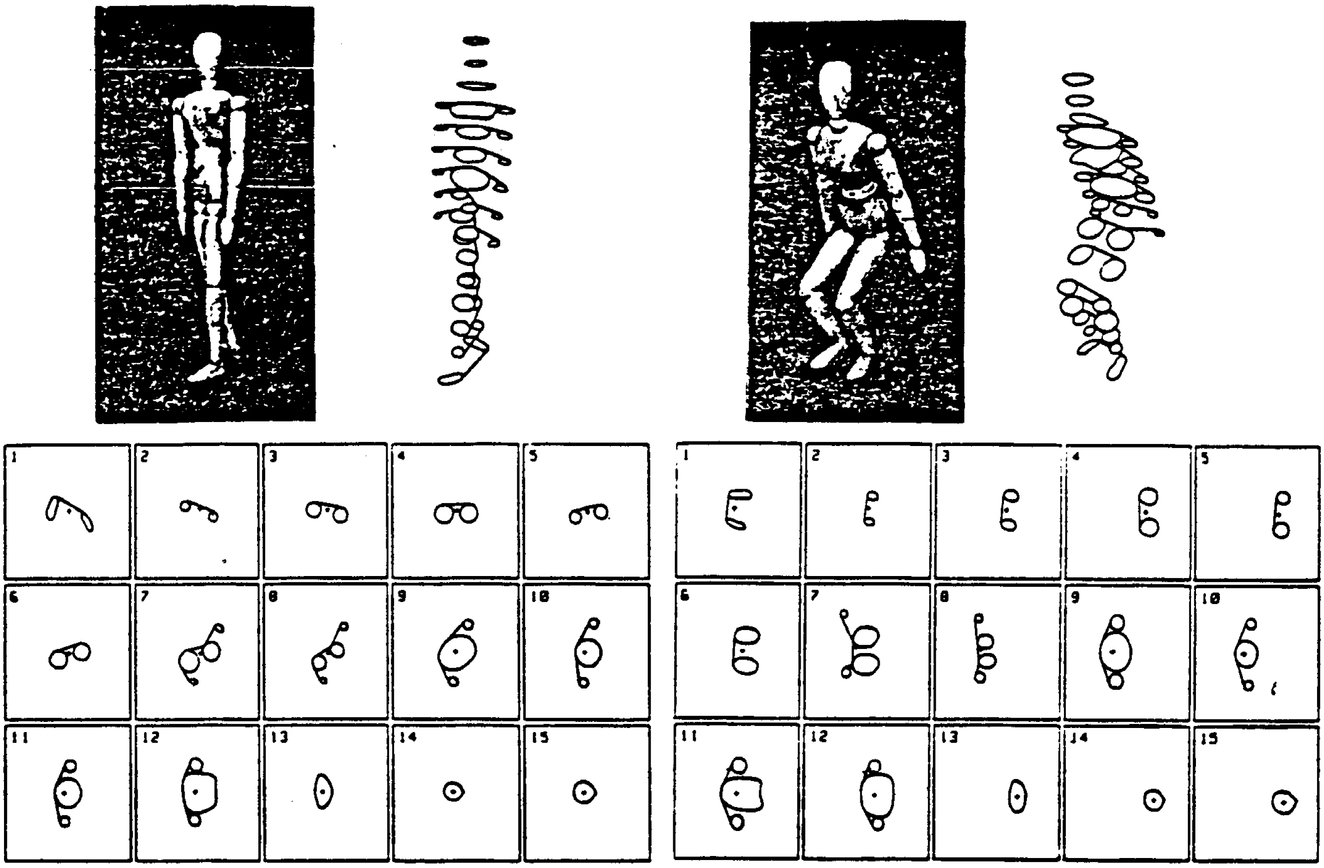
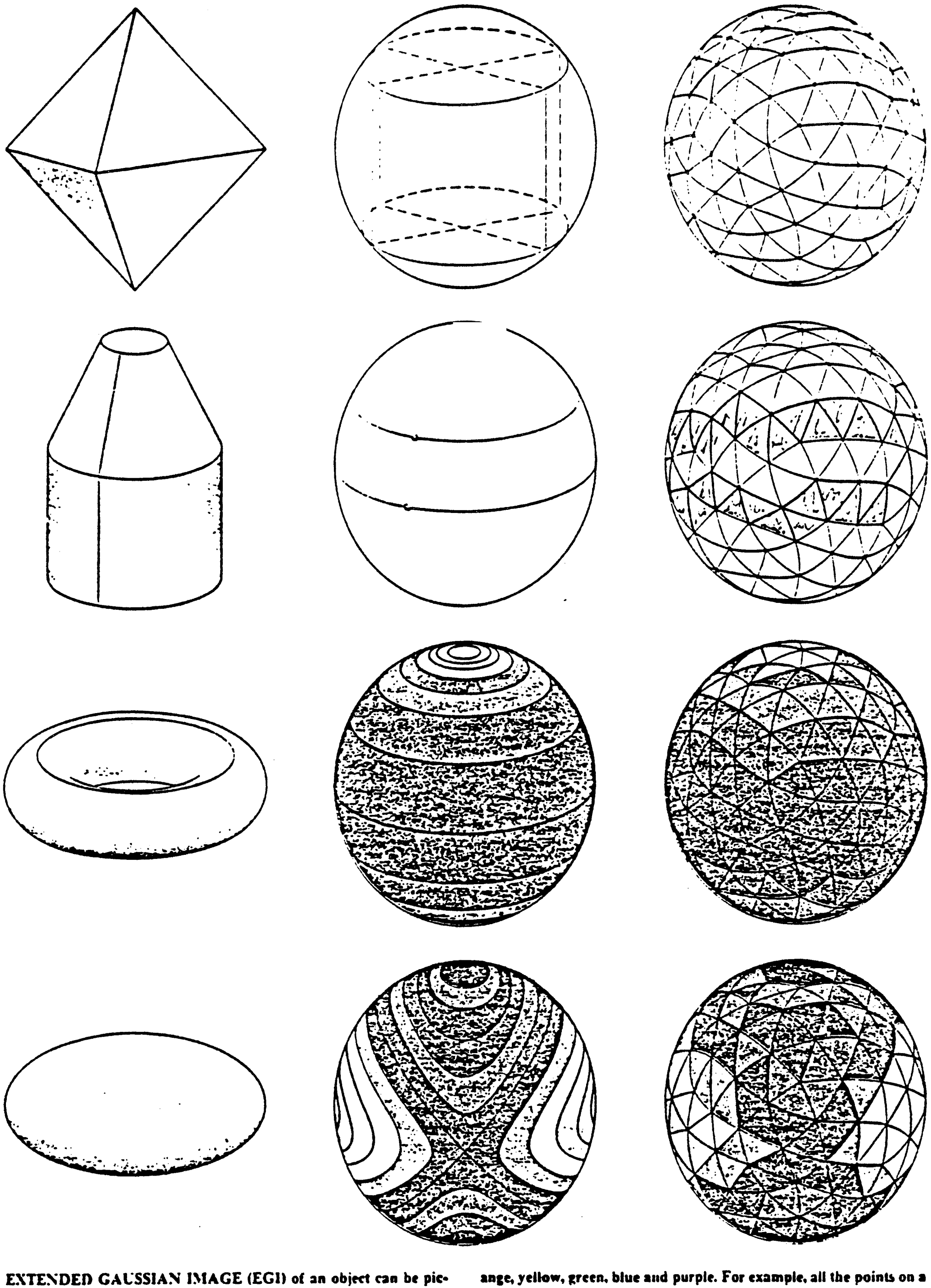


Figure 41. Cross-Sections of Doll used for Shape Description (from [130])

Faugeras [45][46] have devised a 3-D object recognition algorithm using geometrical matching between primitive surfaces. The primitive surfaces currently implemented in the INRIA computer vision system are planes, but quadric surface algorithms are presented in these papers. Each geometric primitive has an associated parameter vector which determines its degrees of freedom. For a plane, there are three independent degrees of freedom: two independent direction parameters and one distance-from-the-origin parameter. Range data is processed to obtain lists of planar regions which correspond to some object. Object models are created and stored as polyhedra which have planar region lists also. Matches between the extracted primitives list and model primitives lists are hypothesized and verified using an approach which minimizes the mean square error criterion over all plane-to-plane transformation matches. Techniques are used to incrementally drop and add primitives in the lists. The rotation and translation matching are decoupled into two separate independent least squares problems. *Quaternions* are used to convert the non-linear 3-D rotation problem into a four-dimensional eigenvalue problem which can be solved directly. The translation problem permits a standard linear least squares solution. The rotation and translation matching errors are combined to provide a quantitative measure of the goodness of the match between the data and a hypothetical model; the best match represents the recognized object. Local consistency tests are used to avoid a full computation on strongly inconsistent plane lists. These ideas result in a computationally efficient method of identifying objects and determining their translation and rotation parameters. Experimental results are shown in [45] for the same automobile part used by Bhanu [15] and Henderson [62]. The precision of rotation angle and translation vector results are stated to be 0.04 radians (2.3 degrees) and 3mm respectively where the accuracy of the range data itself is 1mm. These results are very good. For polyhedral objects, this approach is similar to the EGI approach which is described next in that surface normals are used for matching. This technique has the property that the computation of rotation angles is direct whereas the EGI approach involves a matching process for each possible discrete rotation angle. It also differs from the EGI approach in that it uses the distance from the origin rather the area of polyhedral faces.

Horn and Ikeuchi [67] [68] [72] [73] discuss the use of extended Gaussian images (EGI) for object recognition and object attitude determination. 3-D object models can be used to compute the prototype surface normal vector orientation histograms for various shapes. Extended Gaussian Images are shown for four shapes in Figure 42. (The Gaussian spheres are tessellated into 240 triangles in this figure.) Depth maps or needle maps computed for real world scene data are processed to create an orientation histogram for the visible half of the Gaussian sphere for pre-segmented objects. The scene object histogram and the prototype object histograms are compared in all possible ways to compute the best match. (For a sphere tessellated with 240 triangles, 720 comparison computations must be done in general.) The best match determines which object is represented by the segmented data and how that object is oriented in space. The extended Gaussian image technique appears to be ideal for *convex* object recognition without occlusion because it uniquely determines convex polyhedra [101]. However, it will almost certainly be limited for arbitrary complicated objects which have concave regions and holes. Simple cases exist where certain concave objects cannot be distinguished from certain rectangular blocks using EGI's. See Figure 43 for two simple objects with identical EGI's. Nonetheless, this method can be very useful in many constrained situations such as bin-picking.

Besl and Jain [13] propose an approach to depth map object recognition which combines surface information, critical point information, and depth-discontinuity edges. No pre-decided surface shapes are used. Their approach is motivated by a theorem from differential geometry which states that the coefficients of the first and second fundamental (differential) forms of a smooth surface uniquely characterize the shape of that surface. Gaussian curvature and mean curvature are isolated as important features because they combine the information of the two fundamental forms and they are invariant to rotations and translations and to changes in surface parametrization. These surface curvature characteristics generalize the notion of curvature for plane curves



EXTENDED GAUSSIAN IMAGE (EGI) of an object can be pictured as a distribution of material over the surface of the Gaussian sphere. The material is initially spread evenly over the surface of the object. Each patch of material on the surface is then moved onto the sphere and compressed or spread out like clay to fit into the corresponding patch on the sphere. The EGI is shown in the middle column of the illustration for various objects. The regions of highest density are shown in red, and regions of lower density are shown in orange, yellow, green, blue and purple. For example, all the points on a face of a polyhedron have the same orientation, and so all the material from that face is concentrated at one point on the Gaussian sphere. The surfaces of a cone and a cylinder are each mapped into a circle on the Gaussian sphere; a line on the cone and a line on the cylinder parallel to the axis of rotation are each mapped into a point. The computer "perceives" the objects as they are shown in the column at the right; there the EGI is quantized on a tesselated Gaussian sphere.

Figure 42. Extended Gaussian Images for Four Objects (from [67])

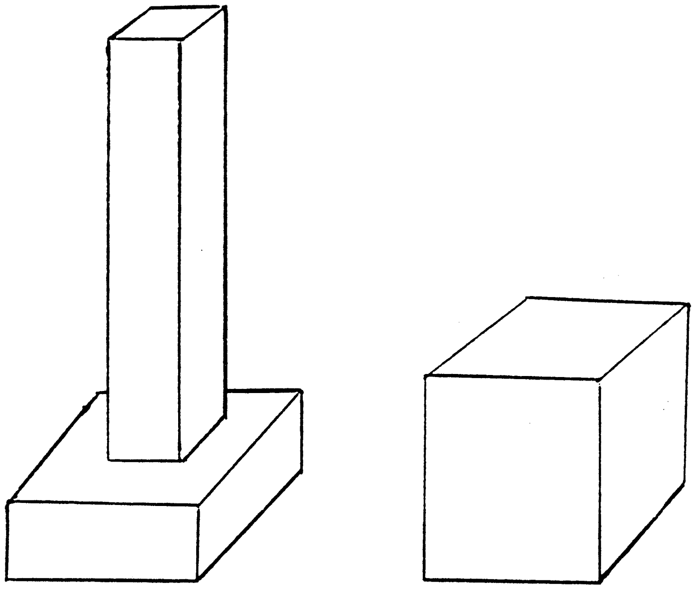


Figure 43. Two Objects with the Same EGI Representation

[111]. Their method consists of the following steps:

(1) Depth maps are first smoothed to remove noise present in the input data. This smoothed data is then interpreted as samples of an underlying surface.

(2) The smoothed image is convolved with window operators which provide least-squares estimates of the first and second partial derivatives of the underlying surface.

(3) These derivative estimates are used to determine Gaussian curvature, mean curvature, critical points, and depth-discontinuity edges. Roof-edges are detected by high mean curvature regions. A critical points image is generated using the intersection of the zero-crossing images of the two first partial derivatives.

(4) The sign of the surface curvature values can be used to place every pixel in one of six classes: pit region, peak region, saddle region, valley region, ridge region, and flat (planar) region. Critical points are similarly classified as peaks, pits, saddles, ridges, valleys, or flats.

(5) Critical points with positive Gaussian curvature are used as starting points for a region growing algorithm which produces a view-independent shape descriptor which can be used to match against a library of precomputed matching representations of various objects. The depth map is segmented via the matching process. Depth-discontinuity edges are used to verify surface region segmentation.

(6) Possible matches of individual matched are projected back into the depth map format for verification. Best-match depth map surface regions are extracted when found and matching continues on the remaining depth map regions until all objects are explained.

(7) The entire scene model description is then processed by a depth buffer algorithm to create a synthetic scene depth map. Occlusion relationships are checked for correct interpretation. The system outputs the final description listing each distinguishable object, the number of occurrences of each object, and the location and orientation of each object instance. Regions of the depth map which could not be interpreted as an object are characterized and stored for future reference.

This proposed approach plans to capitalize on the structural scene information available in the 6-level image created by the sign bits of the surface curvature values. Experimental surface curvature results are shown in [13] which indicate the robustness of the computed features, but no matching or recognition results have been obtained yet.

5. Review Summary

We have given one precise definition of the 3-D object recognition problem, examined the qualitative requirements of systems which address this problem, reviewed a variety of related topics, and surveyed 3-D object recognition systems discussed in the literature.

We first reviewed representations for objects and surfaces. No single object or surface representation seems to be preferred by all researchers for computer vision purposes. We have seen that generalized cones have received a great deal of attention perhaps due to their compact representation of a wide variety of objects. Nonetheless, generalized cones lack the generality of surface boundary representations, which have also been used by computer vision researchers. For the problem we have posed, a surface boundary representation of some sort is probably the right choice for general purpose object models. Future research shall eventually decide this issue. The choice of a particular surface representation for computer vision purposes is not easily decided. No matter which object or surface representation is used by a system, it will be important to have a good graphics module for fast renderings.

The topics of image formation and image processing were discussed. It was assumed that the reader was already somewhat familiar with intensity image formation and conventional image processing techniques. Therefore, we looked at various ways to form range images, or depth maps, of real world scenes and process them. The fact that shape information is directly available in depth maps has influenced the type of processing which is done. Many papers have concentrated on looking for particular shapes such as planes, cylinders, cones, and spheres. Unfortunately, this approach has not led to any general purpose approaches to the problem.

At this point, we digressed slightly to cover the topic of 3-D object reconstruction. This may have seemed slightly contradictory because object reconstruction algorithms must utilize data from multiple views. However, these papers discussed many issues of interest for recognizing objects such as surface matching and polyhedral depth map approximations.

Surface characterization ideas were presented next. These papers emphasized the use of surface curvature, critical points, slope districts, curvature districts, and pixel-by-pixel classification. The ideas were based on continuous surface properties and adapted to discrete surfaces.

Finally, we discussed object recognition techniques. Let us consider what can be learned from the research papers we have reviewed. We have seen the importance of *symbolic scene descriptions for matching which are invariant to object rotations and translations* as in the case of the Extended Gaussian Image. We have seen the limitations of relying on too much sensor data as in the case of Sato and Honda. The importance of incorporating view-independent object models in the design of a system is apparent in Oshima and Shirai's work. We have talked about the constraining power of knowing one single dihedral edge between planar faces in the case of Bolles et al. We have seen 3-D feature clustering techniques used by several researchers. The use of the prediction-hypothesis-verification paradigm is widespread. Constraint propagation and consistency checking are also common ideas. Some investigators have stressed the importance of multi-level logic and modular communicating structures in their systems. Lee and Fu, for example, have pointed out the necessity of an interactive combination of top-down (model-driven) and bottom-up (data-driven) approaches. We have also seen that open-loop systems are only as robust as their weakest component and that rendering algorithms can be used to provide prediction of image features, verification of object features, and feedback on final decisions. Hierarchical geometric models have been used by many researchers. The importance of efficient indexing into an object model library has been brought out. The concepts of visual potential (aspect graph) and characteristic views have been introduced which allow object shape from all views to be described by a finite list. Global and local shape description methods have been explored by many. Edge-based and region-based segmentation operators have been used, but few operators rely on both edges and regions simultaneously. We have seen that it may not be advisable to decide ahead of time what a surface is going to look like and then go and fit that surface to every window in the image to find it because of the complications which result in windows of multiple surface types not to mention the lack of generality involved. Many systems use graph structures internally for data storage where nodes and arcs are associated with geometric entities and relationships respectively.

It is interesting to note that only a few researchers have attempted to justify their approaches in logical, mathematical terms. The approach often used seems to be the following: think up something that might work and try it out. It is likely that this phenomena will gradually change as research continues.

To the best of our knowledge, there currently are no complete solutions to the object recognition problem we have stated. Much work remains for future research. Some very sophisticated systems have been developed using either intensity images or range images, but very few have even attempted to use both. How can data from images of two different types be integrated and used effectively? What are the best features to symbolically describe intensity and/or range images for matching purposes? Many modeling issues are not resolved. One of the problems with surface representations is that they usually require a larger amount of stored data than other representations. Can surfaces be represented in such a way that arbitrary view object recognition algorithms can execute in a second or two? Occlusion has always been a problem. What view-independent methods can be used to handle occlusion? Most object recognition schemes use linear time matching techniques (the unknown object features are matched against each object one-by-one). How can large object libraries be searched fast enough for practical purposes? Any recognition system of this type is bound to make mistakes occasionally. Can a system be made to learn from its mistakes? The above questions are just a few of those which need to be addressed by future research.

References

IJCAI = International Joint Conference on Artificial Intelligence
IJCPR = International Joint Conference on Pattern Recognition
PRIP = Pattern Recognition and Image Processing Conference

(1) Abe, Norihiro, Itho, Fumihide, Tsuji, Saburo, "Toward Generation of 3-Dimensional Models of Objects using 2-Dimensional Figures and Explanations in Language," *Proc. IJCAI-8,* pgs. 1113-1115, 1983

(2) Agin, Gerald J. and Binford, Thomas O., "Computer Description of Curved Objects," *Proc. IJCAI-3,* pgs. 629-640, 1973

(3) Altschuler, Martin D., Posdamer, Jeffrey L., Frieder, Gideon, Altschuler, Bruce R., and Taboada, John, "The Numerical Stereo Camera," *SPIE 3-D Machine Perception,* Vol. 283, pgs. 15-24, 1981

(4) Badler, Norman and Bajcsy, Ruzena, "Three-Dimensional Representations for Computer Graphics and Computer Vision," *ACM Computer Graphics,* Vol. 12, pgs. 153-160, 1978

(5) Bajcsy, Ruzena, "Three-Dimensional Scene Analysis," *Proc. IJCPR-5,* pgs. 1064-1074, 1980

(6) Baker, Harlyn, "Three-Dimensional Modelling," *Proc. IJCAI-5,* pgs. 649-655, 1977

(7) Ballard, Dana H. and Brown, Christopher M., *Computer Vision,* Englewood Cliffs, NJ: Prentice-Hall, 1982

(8) Ballard, D.H. and Sabbah, D., "Viewer Independent Shape Recognition", *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-5 No. 2, pgs. 653-659, March 1983

(9) Barnard, Stephen T. and Fischler, Martin A., "Computational Stereo," *ACM Computer Surveys,* Vol. 14, No. 4, pgs. 553-572, December 1982

(10) Barnhill, Robert E., "A Survey of the Representation and Design of Surfaces," *IEEE Computer Graphics and Applications,* pgs. 9-16, October 1983

(11) Barrow, Harry G., and Tenenbaum, Jay M., "Computational Vision", *Proc. IEEE,* Vol. 69, No. 5, pgs. 572-595, May 1981

(12) Beaudet, Paul R., "Rotationally Invariant Image Operators," *Proc. 4th Int'l. Joint Conf. Pattern Recognition,* Kyoto, Japan, pgs. 579-583, November 1978

(13) Besl, Paul J. and Jain, Ramesh C., "Surface Characterization for Three-Dimensional Object Recognition using Single Arbitrary View Depth Maps," RSD-TR-20-84, Center for Robotics and Integrated Manufacturing, Univ. of Mich., Ann Arbor, November 1984

(14) Besl, Paul J., Delp, Edward J., Jain, Ramesh C., "Automatic Visual Solder Joint Inspection," RSD-TR-16-84, Center for Robotics and Integrated Manufacturing, Univ. of Mich., Ann Arbor, October 1984

(15) Bhanu, Bir, "Representation and Shape Matching of 3-D Objects," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-6 No. 3, pgs. 340-350, May 1984

(16) Bhanu, Bir, "Surface Representation and Shape Matching of 3-D Objects," *Proc. PRIP,* pgs. 349-354, 1982

(17) Bledsoe, W.W., "The Sup-Inf method in Presburger Arithmetic," Dept. of Math and Comp. Sci. Memo, Univ. Texas Austin, ATP-18, Dec. 1974

(18) Binford, Thomas O., "Survey of Model-Based Image Analysis Systems," *The International Journal of Robotics Research,* Vol. 1, No. 1, pgs. 18-64, Spring 1982

(19) Bocquet, J.C. and Tichkiewitch, S., qAn 'Expert System' for Reconstruction of Mechanical Object from Projections," *Proc. IJCPR,* pgs. 491-496, 1982

(20) Boissonnat, J.D., "Representation of Object Triangulating Points in 3-D Space," *Proc. IJCPR,* pgs. 830-832, 1982

(21) Boissonnat, J.D. and Faugeras, O.D., "Triangulation of 3-D Objects," *Proc. IJCAI-7,* pgs. 658-660, 1981

(22) Bolle, Ruud M. and Cooper, David B., Cernushi-Frias, B., "Three-Dimensional Surface Shape Recognition by Approximating Image Intensity Functions with Quadric Polynomials," *Proc. PRIP,* pgs. 611-617, 1982

(23) Bolle, Ruud M. and Cooper, David B., "Bayesian Recognition of Local 3-D Shape by Aproximating Image Intensity Functions with Quadric Polynomials," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-6 No. 4, pgs. 418-429, July 1984

(24) Bolles, Robert C.; Horaud, Patrice; Hannah, Marsha Jo, "3DPO: A Three-Dimensional Part Orientation System," *Proc. IJCAI-8,* pgs. 1116-1120, 1983

(25) Bolles, Robert C. and Cain, Ronald A., "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method" *The International Journal of Robotics Research,* Vol. 1, No. 3, pgs. 57-82, Fall 1982

(26) Bolles, Robert C. and Fischler, Martin A., "A RANSAC-based Approach to Model Fitting and Its Application to Finding Cylinders in Range Data," *Proc. IJCAI-7,* pgs. 637-643, 1981

(27) Brady, Michael, "Computational Approaches to Image Understanding," *ACM Computing Surveys,* Vol. 14, No. 1, pgs. 3-71, March 1982

(28) Brady, Michael, "Preface - The Changing Shape of Computer Vision" *Artificial Intelligence,* Vol. 17, pgs. 1-15, August 1981

(29) Brooks, Rodney A., "Representing Possible Realities for Vision and Manipulation," *Proc. PRIP,* pgs. 587-592, 1982

(30) Brooks, Rodney A., "Model-Based Three-Dimensional Interpretations of Two-Dimensional Images," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-5 No. 2, pgs. 140-149, March 1983

(31) Brooks, Rodney A., "Symbolic Reasoning among 3-D Models and 2-D Images," *Artificial Intelligence,* Vol. 17, pgs. 285-348, August 1981

(32) Brooks, Rodney A., Greiner, Russell, and Binford, Thomas 0., "The ACRONYM Model-Based Vision System," *Proc. IJCAI-6,* pgs. 105-113, August 1979

(33) Brown, Christopher M., "Some Mathematical and Representational Aspects of Solid Modeling," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-3 No. 4, pgs. 444-453, July 1981

(34) Cernushi-Frias, B., Cooper, David B., and Bolle, Ruud M., "Estimation of Location and Orientation of 3-D Surfaces Using a Single 2-D Image," *Proc. PRIP,* pgs. 605-610, 1982

(35) Chakravarty, Indranil and Freeman, Herbert, "Characteristic Views as a basis for three-dimensional object recognition," *SPIE Robot Vision,* Vol. 336, pgs. 37-45, May 1982

(36) Coleman, E. North and Jain, Ramesh, "Obtaining Shape of Textured and Specular Surface using Four-Source Photometry," *Computer Graphics and Image Processing,* Vol. 18, No. 4, pgs. 309-328, 1982

(37) Dane, Clayton and Bajcsy, Ruzena, "An Object-Centered Three-Dimensional Model Builder," *Proc. IJCPR,* pgs. 348-350, 1982

(38) Dane, C., "An Object-Centered Three-Dimensional Model Builder," Ph.D. Thesis, CIS Dept., Moore School of Electrical Engineering, Univ. of Penn., Phila, PA, 1982

(39) Douglass, Robert J., "Interpreting 3-D Scenes: A Model-Building Approach," *Computer Graphics and Image Processing,* Vol. 17, pgs. 91-113, October 1981

(40) Douglass, Robert J., "Recognition and Depth Perception of Objects in Real World Scenes," *Proc. IJCAI-5,* pg. 657, 1977

(41) Duda, R.O., Nitzan, D., Barrett, P., "Use of Range and Reflectance Data to Find Planar Surface Regions," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-1 No. 3, pgs. 254-271, July 1979

(42) Dudani, S.A., Breeding, K.J., McGhee, R.B., "Aircraft Identification by Moment Invariants," *IEEE Trans. Computers,* Vol. C-26, No. 1, pgs. 39-46, January 1977

(43) Dyer, Charles R. and Chin, Roland T., "Model-Based Industrial Part Recognition: Systems and Algorithms," Computer Sciences Technical Report #538, Univ. of Wisconsin, Madison, March 1984

(44) Fang, T.J., Huang, Z.H., Kanal, L.N., Lambird, B., Lavine, D., Stockman, G., Xiong, F.L., "Three Dimensional Object Recognition using a Transformation Clustering Technique," *Proc. IJCPR,* pgs. 678-681, 1982

(45) Faugeras, O.D., "New Steps Toward a Flexible 3-D Vision System for Robotics," *IJCPR,* pgs. 796-805, 1984

(46) Faugeras, O.D. and Hebert, M., "A 3-D Recognition and Positioning Algorithm using Geometrical Matching between Primitive Surfaces," *Proc. IJCAI-7,* pgs. 996-1002, 1983

(47) Faugeras, O.D., Hebert, M., Mussi, P., Boissonnat, J.D., "Polyhedral Approximation of 3-D Objects without Holes," *Proc. PRIP,* pgs. 593-598, 1982

(48) Faux, I.D. and Pratt, M.J., *Computational Geometry for Design and Manufacture,* Chichester: Ellis Horwood, 1979

(49) Fisher, Robert B., "Using Surfaces and Object Models to Recognize Partially Obscured Objects," *Proc. IJCAI-8,* pgs. 989-995, 1983

(50) Foley, James D. and Van Dam, Andries, *Fundamentals of Interactive Computer Graphics,* New York: Addison-Wesley, 1982

(51) Gennery, Donald B., "Object Detection and Measurement using Stereo Vision," *Proc. IJCAI-6,* pgs. 320-327, 1979

(52) *GEOMOD 2.5 User Manual and Reference Manual,* Structural Dynamics Research Corporation, Milford, Ohio, 1984

(53) Gevarter, William B., "Machine Vision: A Report on the State of the Art," *Computers in Mechanical Engineering (CIME),* pgs. 25-30, April 1983

(54) Goad, C., "Special Purpose Automatic Programming for 3D Model-Based Vision," *Proc. Image Understanding Workshop,* pgs. 94-104, 1983

(55) Gonzalez, R.C. and Wintz, P., *Digital Image Processing,* Reading, MA: Addison-Wesley, 1978

(56) Grimson, W.E.L., "A Computer Implementation of a Theory of Human Stereo Vision," MIT AI Lab Memo No. 565, January 1980

(57) Guzman, Adolfo, "Computer Recognition of Three-dimensional Objects in a Visual Scene," MAC-TR-59 (Ph.D. Thesis), Project MAC, MIT, Cambridge, MA, 1968

(58) Hakala, D.G., Hillyard, R.C., Malraison, P.F., "Natural Quadrics in Mechanical Design," SIGGRAPH '81 Seminar: Solid Modeling, Dallas, Texas, August 1981

(59) Haralick, Robert M., "Digital Step Edges from Zero-Crossings of Second Directional Derivatives," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-6 No. 1, pgs. 58-68, January 1984

(60) Haralick, R.M., Laffey, T.J., Watson, L.T., "The Topographic Primal Sketch," *The International Journal of Robotics Research,* Spring 1983

(61) Hebert, Martial and Ponce, Jean, "A New Method for Segmenting 3-D Scenes into Primitives," *Proc. IJCPR,* pgs. 836-838, 1982

(62) Henderson, T.C., "Efficient 3-D Object Representations for Industrial Vision Systems," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-5 No. 6, pgs. 609-617, November 1983

(63) Henderson, T.C., "Efficient Segmentation method for Range Data," *SPIE Robot Vision,* Vol. 336, pgs. 46-47, May 1982

(64) Henderson, T.C. and Bhanu, B., "Three-Point Seed Method for the Extraction of Planar Faces from Range Data," *Proc. IEEE Workshop on Industrial Applications of Machine Vision,* Research Triangle Park, North Carolina, pgs. 181-186, May 1982

(65) Herman, Martin, Kanade, Takeo, and Kuroe, Shigeru, "The 3-D MOSAIC Scene Understanding System," *Proc. IJCAI-8,* pgs. 1108-1112, 1983

(66) Herman, Martin and Kanade, Takeo, "The 3-D MOSAIC Scene Understanding System," *Proc. Image Understanding Workshop,* pgs. 137-148, October 1984

(67) Horn, Berthold K.P. and Ikeuchi, Katsushi, "The Mechanical Manipulation of Randomly Oriented Parts," *Scientific American,* pgs. 100-111, July 1984

(68) Horn, Berthold K.P., "Extended Gaussian Images," AI Memo 740, MIT AI Lab, July 1983

(69) Horn, Berthold K.P., "Understanding Image Intensities," *Artificial Intelligence,* Vol. 8, pgs. 201-231, 1977

(70) Idesawa, Masanori and Yatagai, Totyohiko, "3-D Shape Input and Processing by Moire Technique," *Proc. IJCPR-5,* pgs. 1085-1090, 1980

(71) Ikeuchi, Katsushi and Horn, Berthold K.P., "Numerical Shape from Shading and Occluding Boundaries," *Artificial Intelligence,* Vol. 17, pgs. 141-184, August 1981

(72) Ikeuchi, K., Horn, B.K.P., Nagata, S., Callahan, T., Feimgold, O., "Picking up an object from a pile of objects," AI Memo 726, MIT AI Lab, May 1983

(73) Ikeuchi, Katsushi, "Recognition of 3-D Objects using the Extended Gaussian Image," *Proc. IJCAI-7,* pgs. 595-600, 1981

(74) Inokuchi, Seiji; Nita, Takeshi; Matsudaw, Fumio; Sakurai, Yoshifumi; "A Three-Dimensional Edge-Region Operator for Range Pictures," *Proc. IJCPR-6,* pgs. 918-920, October 1982

(75) Inokuchi, Seiji and Nevatia, R., "Boundary Detection in Range Pictures," *Proc. IJCPR-5,* pgs. 1031-1035, 1980

(76) Inokuchi, Seiji; Sato, Kosuke; Matsuda, Fumio; "Range Imaging System for 3-D Object Recognition," *IEEE Int'l Conf. on Pattern Recognition,* pgs. 806-808, August 1984

(77) Jain, Ramesh, "Dynamic Scene Analysis," *Progress in Pattern Recognition,* Vol. 2, A. Rosenfeld and L. Kanel, Eds., North-Holland, 1983

(78) Jarvis, R.A., "A Perspective on Range Finding Techniques for Computer Vision", *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-5 No. 2, March 1983, pgs. 122-139

(79) Jarvis, R.A., "A Laser Time-of-Flight Range Scanner for Robotic Vision", *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-5 No. 5, September 1983, pgs. 505-512

(80) Kanade, Takeo, "Recovery of the Three-Dimensional Shape of an Object from a Single View," *Artificial Intelligence,* Vol. 17, pgs. 409-460, August 1981

(81) Koenderink, J. J. and van Doorn, A.J., "Internal Representation of Solid Shape with Respect to Vision," *Biological Cybernetics,* Vol. 32, pgs. 211-216, 1979

(82) Koenderink, J. J. and van Doorn, A.J., "The Singularities of the Visual Mapping," *Biological Cybernetics,* Vol. 24, pgs. 51-59, 1976

(83) Kuan, Darwin T. and Drazovich, Robert J., "Model-Based Interpretation of Range Imagery," *AAAI-84*, pgs. 210-215, 1984

(84) Laffey, Thomas J., Haralick, Robert M., Watson, Layne T., "Topographic Classification of Digital Image Intensity Surfaces," *IEEE Proc. Workshop on Computer Vision: Repres. and Control*, pgs. 171-177, August 1982

(85) Lafue, Gilles, "Recognition of Three-Dimensional Objects from Orthographic Views," *ACM Computer Graphics*, Vol. 10, No. 2, pgs. 103-108, 1976

(86) Langridge, D.J., "Detection of Discontinuities in the First Derivatives of Surfaces," *Computer Vision, Graphics, and Image Processing*, Vol. 27, pgs. 291-308, October 1984

(87) Lee, Hsien-Che and Fu, King-sun, "A Computer Vision System for Generating Object Description," *Proc. PRIP*, pgs. 466-472, 1982

(88) Lee, Hsien-Che and Fu, King-sun, "Generating Object Descriptions for Model Retrieval," *IEEE Trans. Pattern Anal. & Machine Intell.*, Vol. PAMI-5 No. 5, pgs. 462-471, September 1983

(89) Lewis, R.A. and Johnston, A.R., "A Scanning Laser Rangefinder for a Robotic Vehicle," *Proc. IJCAI-5*, pgs. 762-768, 1977

(90) Lin, W.C. and Fu, K.S., "A Syntatic Approach to 3-D Object Representation," *IEEE Trans. Pattern Anal. & Machine Intell.*, Vol. PAMI-6 No. 3, pgs. 351-364, May 1984

(91) Lin, C. and Perry, M.J., "Shape Description using Surface Triangularization," *IEEE Proc. Workshop on Computer Vision: Repres. and Control*, pgs. 38-43, August 1982

(92) Lipschutz, Martin M., *Differential Geometry*, Mc-Graw Hill, 1969

(93) Little, James J., "An Iterative Method for Reconstructing Convex Polyhedra from Extended Gaussian Images," *Proc. AAAI-83*, pgs. 247-250, 1983

(94) Lynch, David K., "Range Enhancement via One-Dimensional Spatial Filtering," *Computer Graphics and Image Processing*, Vol. 15, No. 2, pgs. 194-200, February 1981

(95) Marimont, David, H., "A Representation for Image Curves," *AAAI-84*, pgs. 237-242, 1984

(96) Martin, Worthy N. and Aggarwal, J.K., "Volumetric Descriptions of Objects from Multiple Views," *IEEE Trans. Pattern Anal. & Machine Intell.*, Vol. PAMI-5 No. 2, pgs. 150-158, March 1983

(97) McKee, J.W. and Aggarwal, J.K., "Computer Recognition of Partial Views of Three-Dimensional Curved Objects," Technical Report No. 171, Univ. of Texas, Austin, May 1975

(98) Meagher, Donald J., "Efficient Synthetic Image Generation of Arbitrary 3-D Objects," *IEEE Conf. Proc. Patt. Rec. and Image Proc.*, pgs. 473-478, 1982

(99) Medioni, G. and Nevatia, R., "Description of 3-D Surfaces Using Curvature Properties," *Proc. Image Understanding Workshop*, pgs. 291-299, October 1984

(100) Milgram, D.L. and Bjorklund, C.M., "Range Image Processing: Planar Surface Extraction," *Proc. IJCPR-5*, pgs. 912-919, 1980

(101) Minkowski, H., "Allgemeine Lehrsatze uber die konvexen Polyeder," Nachrichten von der Koniglichen Gesellschaft der Wissenschaften, Mathematisch-Physikalische Klasse, Gottingen, pgs. 198-219, 1897

(102) Mitiche, Amar & Aggarwal, J.K., "Detection of Edges Using Range Information", *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-5 No. 2, March 1983, pgs. 174-178

(103) Mulgaonkar, Prasanna G., Shapiro, Linda G., Haralick, Robert M., "Recognizing Three-Dimensional Ojbects Single Perspective Views Using Geometric and Relational Reasoning," *IEEE Conf. Proc. Patt. Rec. and Image Proc.,* pgs. 479-484, 1982

(104) Nackman, Leo R., "Two-dimensional Critical Point Configuration Graphs," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-6 No. 4, pgs. 442-449, July 1984

(105) Nevatia, Ramakant and Binford, Thomas O., "Description and Recognition of Curved Objects," *Artificial Intelligence,* Vol. 8, pgs. 77-98, 1977

(106) Nevatia, Ramakant and Binford, Thomas O., "Structured Descriptions of Complex Objects," *Proc. IJCAI-3,* pgs. 641-647, 1973

(107) Newman, William M. and Sproull, Robert F., *Principles of Interactive Computer Graphics, 2nd Ed.,* New York: McGraw-Hill, 1979

(108) Nishihara, H.K., "Intensity, Visible-Surface, and Volumetric Representatio
Vol. 17, pgs. 265-284, August 1981

(109) Nitzan, D., Brain, A.E., Duda, R.O., "The Measurment and Use of Registered Reflectance and Range Data in Scene Analysis," *Proc. IEEE,* Vol. 65, pgs. 206-220, February 1977

(110) O'Brien, Nancy and Jain, Ramesh, "Axial Motion Stereo," *Proc. 2nd IEEE Workshop on Computer Vision: Representation and Control,* May 1984

(111) O'Neill, Barrett, *Elementary Differential Geometry,* New York: Academic Press, 1966

(112) O'Rourke, Joseph and Badler, Norman, "Decomposition of Three-Dimensional Objects into Spheres," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-1 No. 3, pgs. 295-305, July 1979

(113) Oshima, Masaki and Shirai, Yoshiaki, "Object Recognition Using Three-Dimesional Information," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-5 No. 4, pgs. 353-361, July 1983

(114) Oshima, Masaki and Shirai, Yoshiaki, "Object Recognition Using Three-Dimesional Information," *Proc. IJCAI-7,* pgs. 601-606, August 1981

(115) Parthasarathy, S. & Birk, J. & Dessimoz, J., "Laser Rangefinder for robot control and inspection", *Proc. SPIE,* Vol. 336, Robot Vision, pgs. 2-11, May 1982

(116) Popplestone, R.J., Brown, C.M., Ambler, A.P., Crawford, G.F., "Forming models of Plane-and-Cylinder Faceted Bodies from Light Stripes," *Proc. IJCAI-4,* pgs. 664-668, 1975

(117) Potmesil, Michael, "Generating Models of Solid Objects by Matching 3D Surface Segments," *Proc. IJCAI-8,* pgs. 1089-1093, 1983

(118) Potmesil, Michael, "Generation of 3D Surface Descriptions from Images of Pattern-Illuminated Objects," *Proc. PRIP,* pgs. 553-559, August 1979

(119) Potmesil, Michael, "Generating Three-Dimensional Surface Models of Solid Objects from Multiple Projections," IPL-TR-033, Ph.D. Thesis, Image Proc. Lab, RPI, Troy, NY, October 1982

(120) Pratt, William K., *Digital Image Processing,* New York: Wiley Interscience, 1978

(121) Reeves, A.P., Prokop, R.J., Andrews, S.E., Kuhl, F.P., "Three Dimensional Shape Analysis using Moments and Fourier Descriptors," *Proc. IJCPR,* pgs. 447-450, August 1984

(122) Requicha, A.A.G. and Voelcker, H.B., "Solid Modeling: Current Status and Research Directions," *IEEE Computer Graphics and Applications,* pgs. 25-37, October 1983

(123) Requicha, A.A.G. and Voelcker, H.B., "Solid Modeling: A Historical Summary and Contemporary Assessment," *IEEE Computer Graphics and Applications,* pgs. 9-24, March 1982

(124) Requicha, Aristides A.G., "Representations for Rigid Solids: Theory, Methods, and Systems," *ACM Computing Surveys,* Vol. 12, No. 4, pgs. 437- 464, December 1980

(125) Roberts, L.G., "Machine Perception of Three-Dimensional Solids," in *Optical and Electro-Optical Information Processing,* J.T. Tippett et al., Eds., Cambridge, MA: MIT Press, pgs. 159-197, 1965

(126) Rosenfeld, A. and Kak A., *Digital Picture Processing,* Vols. 1 and 2, New York: Academic Press, 1981

(127) Rosenfeld, Azriel, "Image Analysis: Problems, Progress, and Prospects," *Pattern Recognition,* Vol. 17, No. 1, pgs. 3-12, 1984

(128) Sadjadi, Firooz A. and Hall, Ernest L., "Three-Dimensional Moment Invariants," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-2 No. 2, pgs. 127-136, March 1980

(129) Sadjadi, Firooz A. and Hall, Ernest L., "Object Recognition by Three-Dimensional Moment Invariants," *Proc. PRIP,* pgs. 327-336, 1979

(130) Sato, Yukio and Honda, Ikuji, "Pseudodistance Measures for Recognition of Curved Objects," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-5 No. 4, pgs. 362-373, July 1983

(131) Sato, Y., Kitagawa, H., Fujita, H., "Shape Measurement of Curved Objects using Multiple Slit Ray Projections," *IEEE Trans. Pattern Anal. & Machine Intell.,* Vol. PAMI-4, pgs. 641-646, November 1982

(132) Scott, Richard, "Graphics and Prediction from Models," *Proc. Image Understanding Workshop,* pgs. 98-106, October 1984

(133) Sethi, I.K. and Jayaramamurthy, S.N., "Surface Classification using Characteristic Contours," *Proc. IJCPR,* pgs. 438-440, August 1984

(134) Shapira, Ruth and Freeman, Herbert, "Reconstruction of Curved Surface Bodies from a Set of Imperfect Projections," *Proc. IJCAI-5,* pgs. 628-634, 1977

(135) Schneier, M.O., "Models and Strategies for Matching in Industrial Vision," Computer Science Technical Report TR-1073, Univ. of Maryland, College Park, July 1981

(136) Shostak, R.E., "On the sup-inf method for proving Presburger formulas," *Journal of ACM,* Vol. 24, pgs. 529-543, 1977

(137) Silberberg, T.M., Davis, L.S., and Harwood, D., "An Iterative Hough Procedure for 3-D Object Recognition," Technical Report CAR-TR-20, Univ. of Maryland, College Park, August 1983

(138) Smith, David R. and Kanade, Takeo, "Autonomous Scene Description with Range Imagery," *Proc. Image Understanding Workshop,* pgs. 282-290, October 1984

(139) Stockman, George and Esteva, Juan Carlos, "Use of Geometrical Constraints and Clustering to Determine 3-D Object Pose," *Proc. IJCPR,* pgs. 742-744, August 1984

(140) Sugihara, Kokichi, "Range-Data Analysis Guided by Junction Dictionary," *Artificial Intelligence,* Vol. 12, pgs. 41-69, 1979

(141) Svetkoff, Donald J., Leonard, Patrick F., Sampson, Robert E., and Jain, Ramesh, "Techniques for Real-Time 3D Feature Extraction Using Range Information," *SPIE Robot Vision,* November 1984

(142) Tiller, Wayne, "Rational B-Splines for Curve and Surface Representation," *IEEE Computer Graphics and Applications,* pgs. 61-69, September 1983

(143) Tio, J.B.K., McPherson, C.A., Hall, E.L., "Curved Surface Measurement for Robot Vision," *Proc. PRIP,* pgs. 370-378, 1982

(144) Terzopoulos, Demetri, "Multilevel Computational Processes for Visual Surface Reconstruction," *Computer Vsion, Graphics, and Image Processing,* Vol. 24, pgs. 52-96, 1983

(145) Tropf, H. and Walter, I., "An ATN Model for 3-D Recognition of Solids in Single Images," *Proc. IJCAI-8,* pgs. 1094-1098, 1983

(146) Udupa, K. Jayaram and Murthy, I.S.N., "New Concepts for Three-Dimensional Shape Analysis," *IEEE Trans. Computers,* Vol. C-26, No. 10, pgs. 1043-1049, October 1977

(147) Ullman, S., *The Interpretation of Visual Motion,* Cambridge, Mass.: MIT Press, 1979

(148) Underwood, Stephen A. and Coates, Clarence L.,Jr., "Visual Learning from Multiple Views," *IEEE Trans. Computers,* Vol. C-24, No. 6, pgs. 651-661, June 1975

(149) Vemuri, B.C. and Aggarwal, J.K., "3-Dimensional Reconstruction of Objects from Range Data," *Proc. IJCPR,* pgs. 752-754, August 1984

(150) Wallace, T.P. and Wintz, P.A., "An Efficient Three-Dimensional Aircraft Recognition Algorithm using Normalized Fourier Descriptors," *Computer Graphics and Image Processing,* Vol. 13, pgs. 96-126, 1980

(151) Waltz, D.L., "Generating Semantic Descriptions from Drawings of Scenes with Shadows," AI-TR-271, MIT AI Lab, Cambridge, Mass., November 1972

(152) Wang, Y.F., Maggee, M.J., Aggarwal, J.K., "Matching Three-dimensional Objects Using Silhouettes," *IEEE Trans. Pattern Anal. & Machine Intell.*, Vol. PAMI-6 No. 4, pgs. 513-517, July 1984

(153) Witkin, Andrew P., "Recovering Surface Shape and Orientation from Texture," *Artificial Intelligence,* Vol. 17, pgs. 17-45, August 1981

(154) Wong, Robert Y. and Hayrepetian, Karineh, "Image Processing with Intensity and Range Data," *Proc. PRIP,* pgs. 518-520, 1982

(155) Woodham, Robert J., "Analysing Images of Curved Surfaces, " *Artificial Intelligence,* Vol. 17, pgs. 117-140, August 1981

(156) Yakimovsky, Y. and Cunningham, R., "A System for Extracting Three-Dimensional Measurements from a Stereo Pair of TV Cameras," *Computer Graphics and Image Processing,* Vol. 7, pgs. 195-210, 1978

(157) York, Bryant W., Hanson, Allen R., Riseman, Edward M., "3D Object Representation and Matching with B-Splines and Surface Patches," *Proc. IJCAI-7,* pgs. 648-651, August 1981

(158) York, Bryant W., Hanson, Allen R., Riseman, Edward M., "A Surface Representation for Computer Vision," *Proc. IJCPR-5,* pgs. 124-129, 1980