MODEL-BASED INDUSTRIAL PART RECOGNITION: SYSTEMS AND ALGORITHMS

by

Charles R. Dyer   Roland T. Chin

Computer Sciences Technical Report #538

March 1984

Model-Based Industrial Part Recognition: Systems and Algorithms

*Charles R. Dyer**

Computer Sciences Department
University of Wisconsin
Madison, WI 53706

Roland T. Chin

Electrical and Computer Engineering
University of Wisconsin
Madison, WI 53706

Abstract

A comparative study and survey of the state-of-the-art in model-based industrial part recognition algorithms is presented. The goal of these methods is to recognize the identity, position and orientation of randomly oriented objects. In its most general form, this is commonly referred to as the "bin-picking" problem in which the parts to be recognized are presented in a jumbled bin. The paper is organized according to 2-D, 2 ½-D, and 3-D object representations. Three central issues common to each category, namely feature extraction, modeling, and matching, are examined in detail. An evaluation and comparison of existing industrial part recognition systems and algorithms is given, providing insights for progress toward future industrial vision systems.

Index Terms: Machine vision systems, industrial part recognition, bin picking, model-based image analysis, feature extraction, modeling, matching, 2-D, 2 ½-D, and 3-D representations.

* This work was supported in part by the National Science Foundation under Grant ECS-8301521.

Table of Contents

1. Introduction

Extensive research and development has taken place over the last thirty years in areas of scene analysis, image understanding, pattern recognition, and artificial intelligence. Areas to which these disciplines have been successfully applied include character recognition, medical diagnosis, military intelligence, and satellite image analysis; however, machine vision for automating the manufacturing process has not *fully* reached commercial reality. One reason for the slow progress on machine vision for robotics and industrial automation is that many manufacturing tasks require sophisticated visual interpretation yet demand low cost, and high speed, accuracy, and flexibility. The following delineates some of these requirements.

- SPEED: The duration required for processing an image and recognizing parts has to agree with the speed of execution of the specific task. Often, the required rate is less than fractions of a second per part.
- ACCURACY: The percentage of successful recognition of parts and the accuracy in determining parts' locations and orientations must be high.
- FLEXIBILITY: The system must be flexible enough for accommodating changes in products and competent enough to analyze objects in uncontrolled environments.

To be fully effective, machine vision systems must be able to handle complex industrial parts. This includes the recognition of various parts and the determination of their position and orientation in industrial environments. In addition, machine vision systems must be able to extract and locate salient features of parts in order to establish spatial references for assembly operations and be able to verify the success of operations.

Most industrial part recognition systems are model-based systems in which recognition involves matching the input scene with a set of pre-defined models. The goal of such systems is to pre-compile a description of each of a known set of industrial parts, and then to use these object models to recognize in an image each instance of an object, and to specify its position and orientation relative to the viewer. In an industrial environment, the following types of constraints and properties are usually found:

- the number of parts in a given domain is usually small (1-50),
- parts may be exactly specified, with known tolerances on particular features,
- parts often have distinctive features (e.g. holes and corners) which are commonly found on many different types of parts,
- defective parts occur in many different (unmodelable) ways, and
- in scenes containing multiple parts, the possible allowable configurations is very large (e.g. touching parts, overlapping parts, and parts at arbitrary orientations with respect to one another and the image sensor).

A growing number of studies have been conducted investigating various approaches to machine recognition of industrial parts. The body of literature generated from this developing field is both vast and scattered. A number of conferences and workshops have been dedicated to the topic of industrial machine vision, for example [A.1]-[A.14]. A few books have also been published on this subject; see, for example, [B.1]-[B.3]. Numerous journal publications have discussed issues involved in industrial vision system design and requirements. A significant number of research activities have been reported on the development of prototype systems for certain specific applications. These studies are concerned with providing pragmatic solutions to current problems in industrial vision tasks. Some of them show the adequacy of image processing techniques and the availability of technology needs for practical automation systems. Others are concerned with the development of various part recognition algorithms that are needed in the future. This paper attempts to provide a survey that looks at these current state-of-the-art machine vision systems and algorithms for industrial part recognition. The goal is to provide some fresh insights and up-to-date information for those interested in this new technology.

2. Related Surveys

There are a number of survey papers and tutorials which have been published recently which provide general information and updated summaries on machine vision for industrial automation.

In [Ros79b] Rosen examined the desired functions and industrial requirements for machine vision which are applicable to sensor-controlled manipulation. Industrial implementations as well as selected problems in the research stage are described. Examples are grouped into bin picking, the manipulation of isolated parts on conveyors, the manipulation in manufacturing and assembly, and visual inspection. He also comments on the fact that present machine vision techniques are sufficiently advanced to permit their uses in factories in a cost-effective way.

Myers [Mye80] presents a survey of existing systems including operational systems in manufacturing and feasibility demonstrations. He describes the work done by the General Motors Research Laboratories, one of the first to apply computer vision technology to a production line, as well as other inspection systems. Yachida and Tsuji [Yac80] survey industrial machine vision activities in Japan. A number of successful vision systems that are now operational in Japanese manufacturing were used to emphasize the commitment being made by both government and industry to research and development in the field. Chin [Chi82a] presents a bibliography on industrial vision for discrete parts.

Kruger and Thompson [Kru81] present a summary and survey of techniques and applications relevant to the field. They look at appropriate generic examples in the areas of inspection, part recognition, and discrete component assembly and discuss sample systems which exemplify the current state-of-the-art. The authors also make economic projections and give recommendations to guide future investigations. The survey concludes with some comments on the fact that the efficacy of the techniques in any application of machine vision depends on both technical factors as well as economic considerations.

Foith *et al.* [Foi81] discuss selected methods in image processing and analysis related to industrial applications and point out why practical systems perform binary image processing. A brief survey and some specific approaches used in several state-of-the-art systems are presented.

Bolles [Bol81] reviews some possible applications of image understanding research to industrial automation, and compares the characteristics of current image understanding systems with that of industrial automation systems. He points out a few ways in which current image understanding techniques may be used in the future to enhance the capabilities of industrial systems.

Binford [Bin82] presents a survey and critique of the state-of-the-art in model-based image analysis systems. Most of the surveyed systems were designed with the philosophy of general purpose vision systems. These systems are anticipated to have a significant impact on practical industrial applications.

In a recently published paper, Kinnucan [Kin83] briefly looks at the development of machine vision in the U.S. in the past twenty years and surveys the current activities of several major research laboratories and industries. He also examines current market activities of existing commercial machine vision systems.

On automated visual inspection, Jarvis [Jar80] uses three practical examples to illustrate the nature of the techniques and problems. Chin and Harlow [Chi82b] present an extensive survey and discuss in detail the inspection of printed circuit boards, photomasks, and IC chips. Porter and Mundy [Por80] provide a comprehensive list of the types of visual inspection techniques currently in use.

Other published surveys and overviews include [Pot83a], [Agi80], [Kin81], [Tro82], [Wes82], [Wes83], [Ale83], [Cas83], [Pug82], [Ros82], [Fu83], [Kel83b], [Kel83a].

3. Organization of the Paper

Tenenbaum, Barrow, and Bolles [Ten79] have identified a number of weaknesses that limit the competence of current recognition systems for complex industrial parts. One of the major limitations is the low dimensionality in spatial representation and description of parts. Simple objects presented against a high contrast background with no occlusion are recognized by extracting simple 2-D features which are matched against 2-D object models. The lack of higher dimension spatial descriptions (for example, 3-D volumetric representations) and their associated matching and feature extraction algorithms for industrial systems restrict the system's capabilities to a limited class of objects observed from a few fixed viewpoints. The ability to recognize a wide variety of rigid parts independent of viewpoint demands the ability to extract view-invariant 3-D features and match them with features of 3-D object models. Another problem is the lack of descriptions of surface characteristics of industrial parts. Without using properties of the surface, many recognition tasks cannot be accomplished by machine vision. It can be concluded that the dimensionality of spatial description and representation is highly dependent on both the particular application and its intended level of accomplishment. What is needed are many levels of spatial description (2-D, 3-D, and intermediate levels that fill the gap that exist between images and physical objects) to fulfill various tasks.

Three central issues are common to the above-discussed problems: (1) How does one extract *features* from 2-D, 3-D, or any intermediate level of spatial description? (2) How does one utilize the extracted features to form *models* at various levels of spatial description? (3) How should the *matching* be done between pictorial features and models at various levels of spatial description?

In this paper, we will survey a variety of solutions to these problems and issues. It is convenient to categorize all industrial part recognition systems into several classes before focusing on their problems, requirements, limitations, and achievements. The selected cases fall into three categories based on their dimensionality of spatial description. To be more specific, we have grouped the reported studies into three classes: 2-D, 2 ½-D, and 3-D representations, presented in Sections 5, 6, and 7, respectively. It is natural to organize the studies in this fashion since systems within each class usually use similar assumptions. The grouping is also intended to provide the readers with an easy understanding of the state-of-the-art technology related to industrial part recognition. Associated with each category, issues related to feature extraction, modeling, and matching are discussed in detail; examples are given to illustrate their contributions and limitations. Figure 1 provides a graphical summary of our organization.

3-D spatial descriptions define exact representations in "object space" using an object-centered coordinate system (either a single global coordinate frame or multiple local coordinate frames may be defined). 3-D representations are viewpoint-independent, volumetric representations that permit computations at an arbitrary viewpoint and to an arbitrary precision of detail. 2-D spatial descriptions are viewer-centered representations in "image space". Each distinct view is represented using, for the most part, shape features derived from a gray-scale or binary image of a prototype object. This class of representation is appropriate when the viewpoint is fixed and only a small number of stable object positions are possible. 2 ½-D representations have attributes of both 2-D and 3-D representations, using features defined in "surface space". These spatial descriptions are viewer-centered representations, but depend on local surface properties of the object in each view, for example range and surface orientation.

The reported studies surveyed in this paper are by no means exhaustive and have been chosen because of their general interest and availability of information. No effort has been made to include non-English references and omissions are unintentional and do not reflect any judgement by the authors. In most cases, the categorization of the cited publication into one of the categories was relatively easy. In marginal cases, the authors assigned the paper to the category they felt most appropriate. Mention should also be made that some effort has been made to critically evaluate and compare various systems and algorithms.

Many reported studies using the above image representations are worth mentioning, but it is almost impossible to discuss all of them in detail. These studies are included in the sections under "Other Studies". They are primarily to provide an annotated bibliography on industrial part recognition algorithms to make the survey complete. No attempt is made to critically evaluate these listed studies.

Related topics that are largely or entirely omitted from this paper are: (a) industrial visual inspection applications, methodologies, and systems; (b)

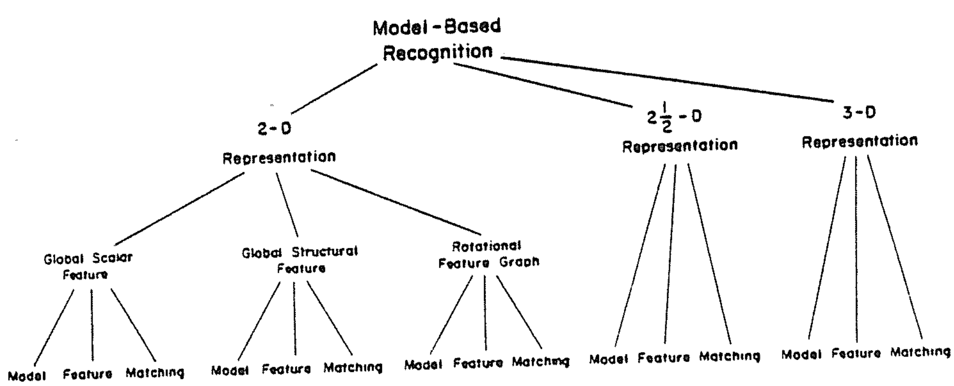


Figure 1. Organization of the survey.

machine vision applications and research activities in private industries that have not been published; (c) the role of software and hardware implementation in industrial machine vision as well as optics and imaging devices; (d) the examination of the economic, social, and strategic elements which justify the use of robot vision; (e) topics dealing with related subjects but too far removed from the main subject of model-based industrial part recognition.

4. Models, Features, and Matching

A part recognition system can be broken down into a training phase and a recognition phase as illustrated in Figure 2. The three major components of the system are *feature extraction*, *object modeling*, and *matching*.

Models: The use of models for image understanding has been studied extensively (see, for example, [Ros79a] and [Bin82]). Most of the models that have been investigated are relatively simple, however, and do not provide adequate descriptions for recognizing industrial parts in complex scenes. While many models of regions and images have been developed based on homogeneity of gray level properties (e.g., texture and color), they have not been widely used for industrial applications. For this reason, this type of model will not be discussed further here. Alternatively, models based on geometric properties of an object's silhouette are commonly used because they describe objects in terms of their constituent shape features.

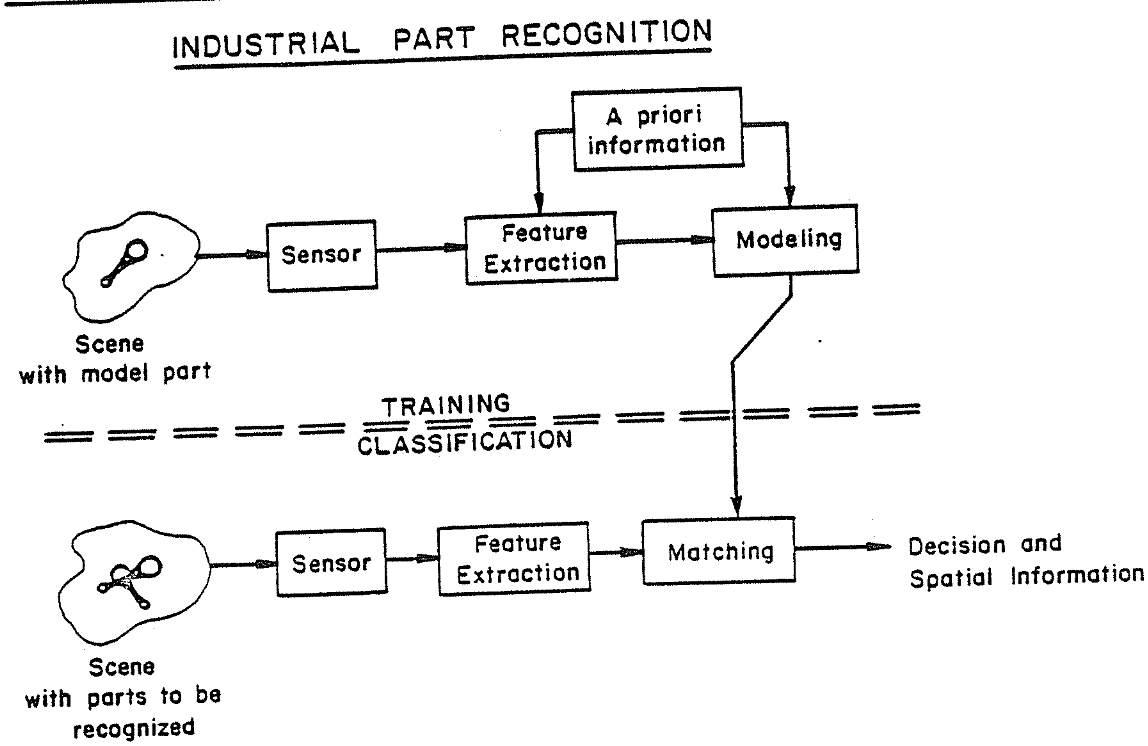


Figure 2. Components of a model-based recognition system.

2-D models have the advantage that they can be automatically constructed from a set of prototype objects (from each possible viewpoint). (In general, it is nontrivial to automatically construct 3-D representations from a set of 2-D views.) They have the disadvantage that they do not make the full 3-D description of an object explicit -- their completeness depends on the complexity of the object, and number and positions of the viewpoints used. In industrial part recognition applications, however, it is frequently the case that limited allowable viewpoints, limited possible stable configurations, and object symmetries substantially reduce the number of distinct views which must be considered.

Features: In many pattern recognition applications, the first problem is to determine which feature measurements should be taken from the input patterns. Features such as edge, corner, line, curve, texture, etc., define individual feature components of an image. These features are then used to segment the image into regions and to generate object descriptions. They are less sensitive with respect to the encountered variations of the original noisy gray-scale images. Usually, the decision of what to measure is rather subjective and dependent on the practical situations.

The features important for industrial image analysis are largely boundaries and measurements derived from boundaries. These features can be roughly categorized into three types [Foi81] : global scalar, global structural, and relational features. Examples for *global scalar features* are perimeter, centroid, distance of contour points from the centroid, curvature, area, and moments of inertia. Global scalar features are relatively easy to extract and recognition schemes associated with this type of feature are usually straight forward feature classifiers. All properties are derivable directly from a single image and their number and order in the feature list is unimportant. Examples of *global structural features* include line segment, arc segment with constant curvature, and corner, defining the object's boundary. They allow the use of syntactic pattern recognition approaches in the matching process in which structural elements are transformed into primitives forming a string grammar; recognition is performed by parsing. Examples of *relational features* include a variety of distance and relative orientation measurements inter-relating substructures and regions of the object. These features are usually configured in a graph structure forming a *relational object graph* in which nodes represent local features and arcs the spatial relations between these features. The matching process involves variations on subgraph isomorphism. This type of feature also provides local cues for the recognition of parts that are overlapping each other.

Most existing industrial vision systems and algorithms restrict their applications to industrial parts against a high contrast background with controlled lighting to eliminate shadows, highlights, and noisy backgrounds. The process of feature extraction usually begins by generating a binary image from the original gray-scale image using simple thresholding, or simply by using a sensor that produce binary images. The use of a binary representation reduces the amount of data that must be handled, but it places a serious limitation on the flexibility and capabilities of the system. After thresholding , the process continues by extracting 2-D features from the binary image. In these systems, features are functions of silhouettes. A tutorial on binary image processing to robot vision applications was given by Kitchin and Pugh [Kit83].

Most pictorial feature extraction algorithms used in these binary imaging systems are simple edge-detection and line-tracing algorithms. They detect boundaries of simple planar objects but usually fail to detect low contrast surface boundaries. Another limitation is that they attempt to deal with 3-D physical objects in terms of 2-D features. This simplification might meet the cost requirement of many industrial applications, but it lacks the capability and flexibility required by many other industrial vision tasks. Finally, current systems seldom have representations of physical surface properties such as surface reflectance and surface orientation (i.e. 2 ½-D representations). Such information is lost in reducing the gray-scale image to a binary image or to a piecewise constant image. Without using these properties of the surface, many important industrial vision tasks that are easy for humans to perform, will remain beyond the competence of vision systems.

There are a few current vision systems and algorithms that are capable of extracting useful information from images of complex industrial parts with considerable noise caused by dirt and unfavorable lighting conditions. These systems process gray-scale images with reasonable dynamic range. The most important drawback of gray-scale image processing is the slow processing rate in extracting features. Most of these systems employ sophisticated feature extraction methods, but, their matching procedures are still based on 2-D models.

Matching: After the modeling procedure the system contains a set of models that describe all aspects of parts that are to be recognized. The process of recognition then consists of matching the extracted data from the scene with those of the models. The general problem of matching may be regarded as finding features in the given image that match one model's features. Some of these methods rely on total image matching using cross-correlation type of measures applied to image intensities or coefficients of some mathematical expansions (e.g. orthogonal expansion). They can be formulated using global optimization to achieve great reliability, but are computationally intensive. In practice, however, recognition speed is of ultimate importance. Moreover, the image will be noisy, and parts within the image will be occluded, and they will be located at random positions. Matching algorithms of this type have little value in industrial part recognition systems.

Matching techniques using global, local, or relational features, or a combinational of these features provide a way to recognize and locate a part on the basis of a few key features. Matching by features becomes a model-driven process in which model features control the matching process. Several model-driven matching techniques have been developed. Most of them are invariant to translation and rotation, and are not too sensitive to noise and image distortion.

The choice of matching process is highly dependent on the type of model used for object representation. Models using global scalar features, such as area and perimeter, are usually associated with the classical feature-space classification scheme. The features of each of the model parts may be thought of as points in n-dimensional space, where n is the number of features measurements. The recognition of an unknown part with an n-dimensional feature vector involves comparing this feature vector with each of the model feature vectors. Both parallel (e.g. the nearest neighbor rule [Dud73]) and hierarchical/sequential decision rules (e.g. the decision-tree method) can be used. The computational expense associated with the parallel classification increases steeply with dimension, but optimal results are achievable. There are numerous advantages to hierarchical classification. Most importantly, the decision procedure can be designed to be both inexpensive and effective, but the overall accuracy is not as great as with the classical decision rules.

Models can be constructed using abstracted and precise geometric representations such as arcs and lines. (In this paper, this type of features is referred to as the global structural feature.) Recognition uses a hypothesis-verification procedure. The structural features are used to predict where objects are located in the scene. Then, additional features are measured, based on the prediction hypothesized by the model, in order to verify and fine tune the match.

Objects can be represented structurally by graphs. Under this model type, geometrical relations between local features (e.g. corner and hole) are of particular interest. The relational structure can be represented by a graph in which each node represents a local feature, and is labeled with a list of properties (e.g. size) for that feature; and arcs link pairs of nodes and are labeled with lists of relation values (e.g. distance and adjacency). Recognition of the object becomes a graph-matching process. This type of matching can be used to handle overlapping parts where a partially visible part corresponds to a subgraph. The matching reduces to one of finding the subgraph. Most of the techniques of this type involve tree-searching techniques which are exhaustive in nature leading to expensive implementation. An alternate graph-matching technique is the hierarchical approach by Barrow and Tenenbaum [Bar81] where the model is decomposed into independent components. This technique was designed to reduce the complexity of the matching process.

5. 2-D Image Representations

In this section we review recognition algorithms that are based on 2-D image representations. They represent objects by a set of one or more distinct views. These viewer-centered representations treat each view independently, reducing the problem to 2-D using image relations and image observables as primitives. For each viewpoint, a sufficient set of image-space-derived features and relations are extracted for describing the object.

We will classify image space models into three types based on the kinds of features which are predominantly used to define an object model. The first class of representations uses *global scalar features* of an object's size and shape (e.g. perimeter and area) organized in geometric property lists. The second type of representation uses *global structural features* which describe more complex properties of the object, usually in terms of lines and curves defining the object's boundary. The third type uses local shape features which are organized in a *relational object graph*. Nodes describe local features and arcs have associated properties which describe the relationship between the pairs of features that they connect.

This division of models based on their constituent feature types also coincides with the kinds of matching algorithms which are appropriate in each case. Global features describe a significant portion of an object's boundary or interior and thus because there are very few of them, there is relatively little "combinatorial" processing to find matches between image and model features. Of course, we are more likely to miss finding such global features because they are so large (e.g. due to occlusion). Alternatively, when many local features are used to describe an object more care must be taken in defining the search procedure used for matching image features with model features. The sequential approach is to tentatively locate local features and then use this to constrain the search for other features.

The remainder of this section covers each of these three representations in detail.

5.1. Global Scalar Features

The predominant model used to date, especially in commercial systems, is the use of a set of 2-D, global shape features describing each possible stable object view. In these systems, each feature is usually translation- and rotation-invariant, but not scale-invariant. That is, objects may be placed at any position and orientation, but the camera geometry is fixed so that the object is of known size. Furthermore, objects are not allowed to touch or overlap one another, allowing simple geometric features of the boundary of each component in an image to be extracted and compared with the models.

5.1.1. Models

The prototype system using a list of simple, global scalar features is the SRI Vision Module [Gle79]. The user interactively selects a set of features which are used to construct an object model. This modeling task is termed as the "training by showing" process. This type of model is compact and facilitates fast matching operations because of the limited number and size of the feature vectors which are extracted from a given image.

The major limitations of this type of model are (1) each possible 2-D view of an object must be described by a separate model, (2) all objects in an image must be extracted by a single predefined threshold (hence lighting, shadows and highlights must be controlled), and (3) objects may not touch or overlap one another, nor may objects have significant defects. (A defective object which is not sufficiently similar to any model can be recognized as a reject, but this may not be adequate in many applications.)

Many other systems have been developed which use variations on the global feature list approach to modeling. A method for modeling industrial parts and programs to compute stable orientations and their views have been developed by Lieberman [Lie79]. A 3-D, CAD system is used to form representations of parts using polyhedrons. Stable positions of the part are determined and their 2-D projections with respect to the camera's viewpoint are computed. A set of outlines are in turn extracted forming the polyhedral model. Figure 3 shows a set of stable orientations of a part and their corresponding models.

Birk, Kelley and Martins [Bir81] model objects by a set of coarse shape features for each possible viewpoint. For a given viewing position, the thresholded object is overlaid with a 3 by 3 grid (each grid square's size is selected by the user) centered at the object's centroid and oriented with respect to the minimum moment of inertia. A count of the number of above threshold pixels in each grid square is used to describe the object.

CONSIGHT-I [Hol79] avoids the problem of threshold selection for object detection by employing a pair of line light sources and a line camera focused at the same place across a moving conveyor belt of parts. Part boundaries are detected as discontinuities in the light line as shown in Figure 4. Features such as centroid and area are computed for each object as it passes through the line of light.

5.1.2. Features

Features and feature extraction procedures for 2-D global scalar feature representations are common to many recognition systems. The image is first analyzed and outlines of connected components (blobs) within the field of view are extracted so that each region can be analyzed. The information in a connected component is sufficient to derive a number of global scalar features such as shape and size descriptors characterizing the industrial part. For some systems, the feature set also includes position and orientation descriptors such as center of gravity, and moments of inertia which provide useful information for part manipulation.

Computation of these global scalar features using binary image processing techniques are relatively inexpensive. They can be computed "on the fly" from,

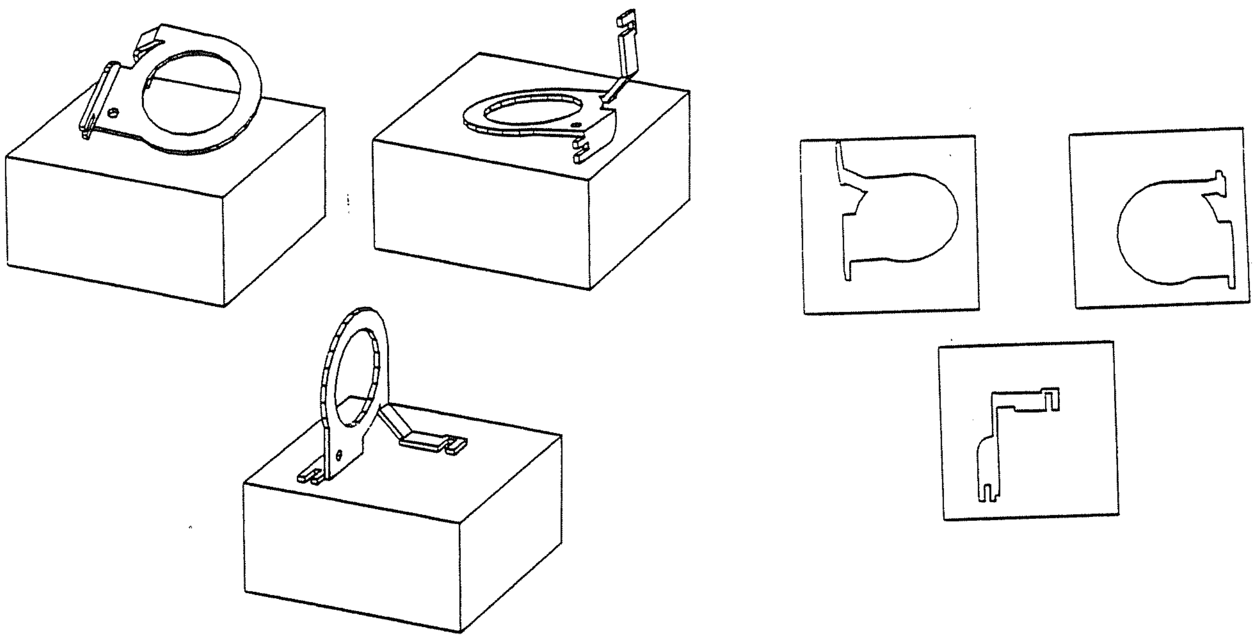


Figure 3. A set of stable orientations for a part and the corresponding silhouettes calculated for each of the orientations when viewed from directly overhead.

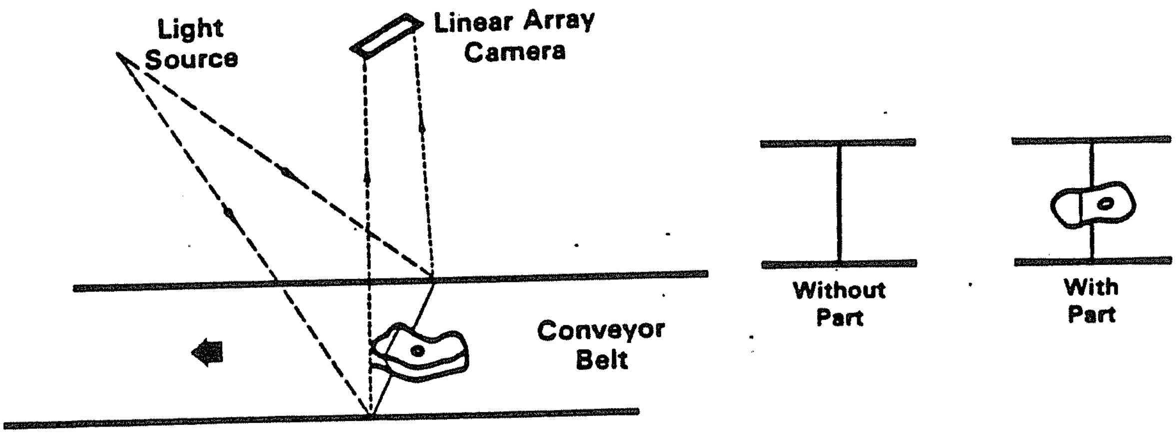


Figure 4. Basic lighting principle of the CONSIGHT-I system and the computer's view of a part.

for example, a run length encoding of the image. This is one of the major advantages of using this feature type.

In the SRI Vision Module system [Gle79], a connectivity analysis procedure generates a description of each blob on the fly as the image is processed. At the same time are computed such features as number of holes, area, perimeter, boundary chain code, compactness, number of corners, and moments of inertia. No feature is calculated if it is not needed for a particular application.

Kelley *et al.* [Kel82] have developed an experimental system to recognize randomly oriented cylindrical workpieces piled in a bin. A number of features are measured to select a holdsite location of the workpiece for manipulation by a mechanical jaw gripper. A global, coarse size descriptor is extracted by a shrinking operator applied to successive images to reduce the binary data into small clusters of pixels. These clusters are then sorted in order by size, and the largest cluster is selected as the holdsite location. Position and orientation features of the selected cluster are computed to determine the location and direction of the gripper relative to the image. An additional feature measurement, the ratio of eigenvalues, is also computed to determine the inclination of the cylinder with respect to the image plane so as to determine the appropriate gripper opening distance. The 'i-bot' system is based on this technique and is now commercially available [Zue83]. This system computes the locations and orientations of a maximum of three workpieces from a bin in two seconds.

Fourier descriptors [Zah72] have been suggested as shape descriptors for industrial part recognition by Persoon and Fu [Per77]. A finite number of harmonics of the Fourier descriptors are computed from the part boundary and compared with a set of reference Fourier descriptors. A minimum-distance classification rule is used for the recognition of various classes of parts.

In gray-scale image processing, segmentation is usually the first step to find regions of fairly uniform intensity which would greatly increase the degree of organization for generating higher level descriptions such as shape and size. Perkins [Per80] has developed a region segmentation method for industrial parts using edges. An expansion-contraction technique is used in which the edge regions are first expanded to close gaps and then contracted after the separate uniform regions have been identified. The process is performed iteratively to preserve small segments.

Baird [Bai77] has used a similar method to separate automotive parts on noisy conveyors. His method involves a smoothing and gap-filling procedure which is essentially a series of edge expansions.

5.1.3. Matching

When models are defined as feature lists, the use of a hierarchical decision tree provides a fast, convenient method for matching that can also take into account feature observation reliability and feature importance.

The SRI Vision Module uses a decision tree method for matching based on the list of global features associated with each model [Agi77]. The tree is automatically constructed from the models as follows. The feature values with the largest separation for a given feature and pair of object models are found and this feature is used to define the the root node of the tree. That is, a threshold is selected for this feature which distinguishes between these two models. Next, two children of the root node are constructed such that all models which have feature value less than or equal to the threshold are associated with the left child; the right child is assigned all models with feature value greater than the threshold. This procedure is repeated recursively, dividing a set of model candidates associated with a node into two disjoint subsets associated with its two children. A terminal node in the tree is one that contains a single model. Figure 5 illustrates such a decision tree.

The decision tree method has the primary advantage of speed, but has the disadvantage of not allowing similar models to be explicitly compared with a given list of image features.

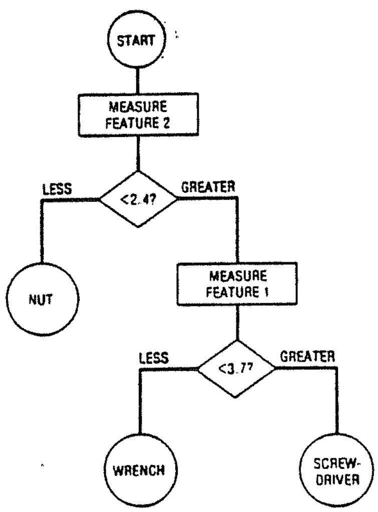


Figure 5. A decision tree classifier [Agi80].

Alternatively, the best matching model to a given list of global features extracted from a component in an image is computed using a nearest-neighbor classification method in feature space, as illustrated in Figure 6. That is, if n features are used to describe all models, then each model is represented by a point in n-dimensional feature space. Given a new feature list extracted from an image, the component is recognized as being an instance of the model which is closest in feature space.

5.1.4. Other Studies

Several systems based on the SRI Vision Module are now commercially available including Machine Intelligence Corporation's VS-100 system, Automatix's Autovision system, Unimation's Univision I, Control Automation's V-1000, Intelledex V-100 Robot Vision System, and Octek's Robot Vision Module. The VS-100 system (and the related system for Puma robots, Univision I) accepts images up to 256 by 256 and thresholds them at a user specified gray level. Up to twelve objects can be in an image and up to thirteen features can be used to model each part. Recognition times of from 250 msec (one feature) to 850 msec (eleven features) per object are typical [Ros81]. The Autovision 4 system processes images up to 512 by 256 and recognition performance is listed at "over ten parts per second for simple parts" [Vil83].

An industrial vision system, S.A.M., has been developed by Tropf *et al.* [Tro82] using binary image processing to extract global scalar features for inspection and part recognition. The system is now commercially available for flexible manufacturing assembly systems [Bru83]. A development system for machine vision based on the Machine Intelligence Corp. VS-100 has been developed and marketed [Che82]. Performance evaluation of the above vision system has also been done by Rosen and Gleason [Ros81].

An experimental system has been developed by Page and Pugh [Pag81] to manipulate engineering parts from random orientation. Simple global scalar features are used to identify gripper locations. Typical recognition times are in the range 0.5 to 3 seconds.

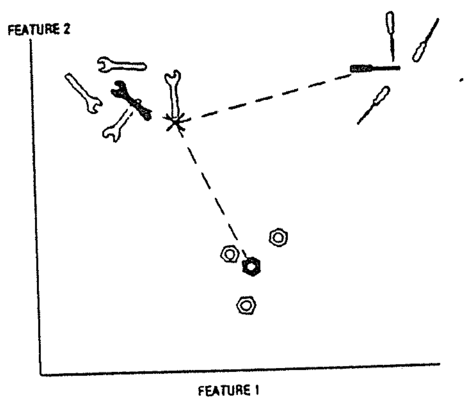


Figure 6. A nearest neighbor classifier [Agi80].

5.2. Global Structural Features

The systems described in the previous section included global shape and size features which consisted of, for the most part, simple integral or real-valued descriptors. In this section we describe models which include more complex features, for the most part structural descriptions of object boundaries. Since a boundary-based feature often includes all or a major portion of an outer or hole border, we have included it here as a global structural feature.

5.2.1. Models

Perkins [Per78] constructs 2-D models from the concurve features extracted from training images of each stable view of each part (see Section 5.2.2 for the feature extraction algorithms). These features include both global structural (e.g. the complete outer border of the object) and local (e.g. a small hole border in the object) descriptions of the object's borders. The list of concurves is not ordered, but it does comprise a more structural approach to describing objects which is not as sensitive to noise (see Section 5.2.3 for the matching algorithm).

An object-centered coordinate system is used in which the origin is defined by either (a) the center of area of the largest closed concurve, or (b) the center of a small closed concurve if it is sufficiently close to the center of the largest closed concurve. The axes are defined in terms of the direction of the least moment of inertia of the largest concurve.

For each concurve in the model, a property list is computed including type (circle, arc, line segment, complex curve, etc.), total length or radius of arcs, magnitude of total angular change, number of straight lines, number of arcs, bending energy, and compactness. In addition, rotational symmetries of the concurve and the complete object are computed as additional descriptors. Rotational symmetry is computed using a correlation-like technique which determines if a sufficient percentage of concurve "multisectors" intersect the rotated concurve. Multisectors are short line segments which are placed at equal intervals along the concurve and at orientations perpendicular to the tangent directions at these points.

Stockman *et al.* [Sto82] define models of 2-D objects by a set of vectors (with an object-centered coordinate system), where each vector is constructed from the linked output of straight edge detectors, curved edge detectors, circle detectors, and intersection detectors. The vectors comprise points which are of four types, and rules are used to permit only certain combinations of points to be linked.

Foith *et al.* [Foi81] describe the object boundary with respect to the centroid of the "dominant blob" defining the 2-D binary object. Circles of prespecified radii are centered on the centroid, their intersections with the object boundary are marked, and line segments are then drawn between these intersections and the centroid. The sequence of angles between successive line segments is used as a rotation-invariant model of the object boundary.

Shirai [Shi78] models objects using a hierarchy of features consisting of edges represented by a description of their curvature and endpoints.

5.2.2. Features

The vision system developed by Perkins [Per78] extracts concurve features in the form of line drawings from noisy gray-scale scenes. The line drawing is a compact representation that describes boundaries of complex 2-D industrial objects. First, a gray-scale image is transformed into an edge map by the Hueckel edge operator [Hue74]. Next, these edges are thinned and connected together into a highly organized chain structure by using knowledge of proximity, directional continuity, and gray-scale continuity. In the linking process, both local and global criteria are used.

Finally, the chains are transformed into a group of concurves. A concurve is defined as an ordered list of shape descriptions which are generated by fitting the chain data to straight lines, or circular arcs, or a combination of both. This curve-fitting step is quite similar to the one used by Shirai [Shi75] in his feature extraction algorithm. The fitting procedure first examines the curvature of the chain data, i.e. connected edge points. Next, it looks for abrupt changes in curvature and picks out end points (critical points) to set the bounds of each grouping. The chain of edge points in each group is fitted with a circular arc or straight line by Newton's method. An additional step is included which verifies and corrects for a poor fit. Figure 7 shows the various stages of extracting concurves from a visually noisy scene.

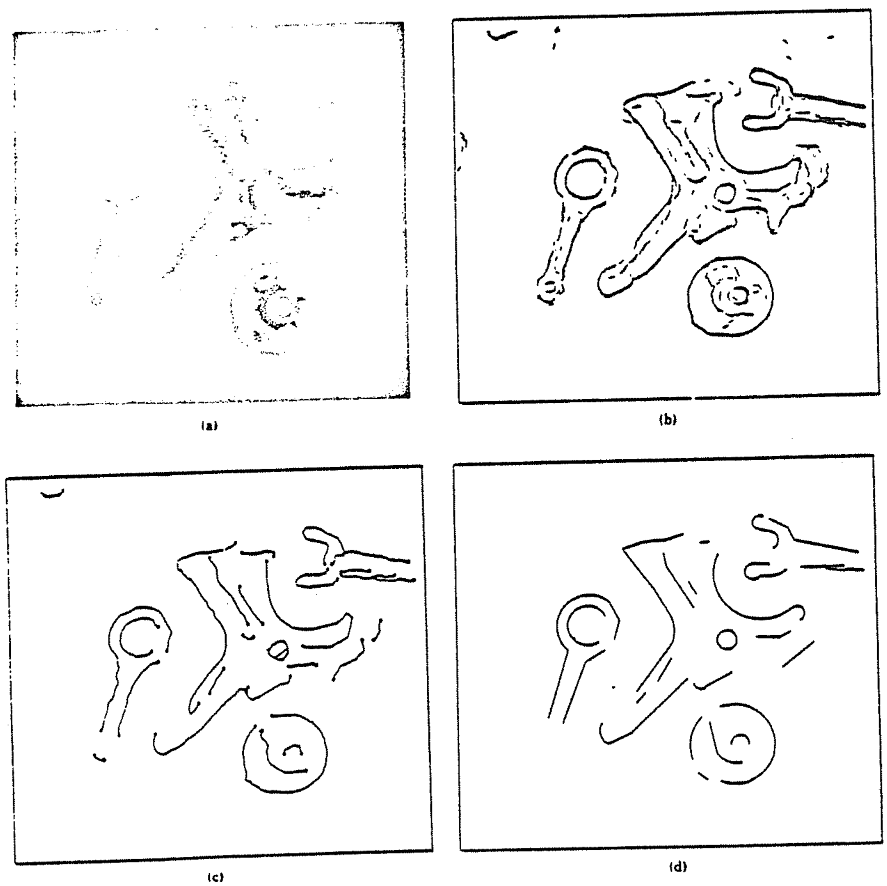


Figure 7. Concurve extraction. (a) Digitized picture. (b) Edge points. (c) Chains with critical points at the end of open chains. (d) Concurves.

In the system developed by Shirai [Shi78], recognition is based on edge cues. The system first extracts edges using a conventional gradient operator. The extracted edges are classified into three types according to their intensity profiles. Next, an edge kernel is located by searching for a set of edge points of the same type which have similar gradient directions. A tracking algorithm is applied in both directions of the kernel to find a smoothly curved edge and its end points. Several passes are applied to locate all sets of smoothly connected edges in the scene. Finally, straight lines and elliptic curves are fit to each segment and segments are merged together, if possible.

Stockman *et al.* [Sto82] use directed edge elements (vectors) as the primary features in their part recognition experiments. Straight edges are detected using the Hough transform and holes are detected by a set of circular masks. Details of the extraction procedures are presented in [Sto80]. A set of simulated carburetor covers and T-hinges are used to demonstrate the method. Matching is performed by registering the detected vectors with the model vectors.

5.2.3. Matching

In Perkin's model-based recognition system [Per78] the matching process is performed in three steps. First, global scalar features (length, area, etc.) extracted from the model and image concurves are compared. The comparison is an exhaustive matching procedure applied to all possible pairings between the model concurves and the image concurves, and the results, given in terms of likelihood measures, are arranged in an ordered list. Second, one model concurve is matched against one image concurve to determine a tentative transformation (x,y,θ) from model to image coordinates. The pair with the highest likelihood is used first; successive pairs are compared until a tentative transformation is found. In cases when the model concurve is symmetric, two matching pairs are required to determine the transformation. Third, a global check of the tentative transformation is performed by matching the complete model with the image. In this step, a set of model multisectors is first transformed using the tentative transformation determined in the previous step. The transformed multisectors of the model are then superimposed on the image for a final comparison by intersecting each multisector with the image concurves. This matching process is shown to be very successful with closed concurves and has been tested with images containing partially overlapping parts.

Stockman *et al.* [Sto82] match images to models using clustering approaches. The procedure matches all possible pairs of image features and model features on the basis of local evidence. A rotation, scaling, and translation transformation is derived to extract parameters from all possible pairs of features. Clustering is performed in the space of all possible transformation parameter sets. This method is believed to be more robust because the clustering procedure integrates all local information before any recognition decision is made. The reported results indicate that their method works well with isolated objects but the success rate for recognizing overlapping parts is low.

Shirai [Shi78] performs recognition of objects using a hierarchy of features where most of the features consist of linked edges represented by an equation and end points. The recognition involves three steps. First, the main feature is located to get clues for the object. Next, a secondary feature is searched for to verify the main feature and to determine the range of the object. Finally, the other lines of the object are located to confirm the recognition.

5.2.4. Other Studies

In a model-driven recognition experiment [Hat82], contour elements described in terms of straight lines are used as the global structural features. Matching is done by iteratively constructing the model contour from image data. Experiments on occluded part recognition have been performed by Turney *et al.* [Tur83] using edges as the features. Recognition is based on template matching between the model edge template and the edge image in the generalized Hough transform space [Bal81a]. This algorithm is shown to be more efficient than direct template matching. Dessimoz *et al.* [Des78a], [Des78b] recognize overlapping parts by first mapping the objects' boundaries into a set of curves and then matching the curves with those in the model. Tropf [Tro80], [Tro81] has developed a recognition system for overlapping workpieces using corner and line primitives and semantic labeling. Structural knowledge of workpieces is used to construct models. Recognition uses heuristic search to find the best match based on a similarity measure. Ayache [Aya83] uses binary images and polygonal approximations to each connected component. Models are automatically constructed by analyzing a prototype part in its different stable positions. The matching is done first by generating a hypothesis of the object location and then by matching model segments to scene segments. The model location is sequentially adjusted by evaluating each match until the best match is found.

Bhanu [Bha81] has developed a hierarchical relaxation labeling technique to do shape matching and has done experiments using 2-D occluded industrial parts [Bha83]. 2-D shapes are used as the global structural features and they are represented by a polygonal approximation. The technique involves the maximization of an evaluation function which is based on the ambiguity and inconsistency of classification. Ayache and Faugeras [Aya82] extended the relaxation labeling technique to match polygonal representations of object boundary segments. Umetani and Taguchi [Ume79] use "general shapes", defined as artificial and non-artificial shapes,, to study the properties and procedures for complex shape discrimination. Feature properties based on vertices, symmetry, complexity, compactness, and concavity have been investigated. These features are chosen based on some psychological experiments and a procedure to discriminate random shapes has been proposed [Ume82].

Vamos [Vam77] has proposed the use of syntactic pattern recognition for modeling machine parts from picture primitives: namely straight line, arc, node and undefined. A set of syntax rules are inferred to characterize the structural relationships of these strings of primitives describing the part. The matching process is a syntax analysis or parsing procedure involving the use of similarity measures between two grammar strings, or two graphs. Jakubowski [Jak77], [Jak83] has conducted a similar study using straight lines or curves as primitives to model machine part shapes and to generate part contours.

Takeyasu *et al.* [Tak77], [Kas77] have developed an assembly system for vacuum cleaners using integrated visual and tactile sensory feedback. First, global scalar features of various parts of the vacuum cleaner are used to locate the cleaner. Then, structural features such as circles and arcs are used in a template matching step for the assembly operation.

5.3. Relational Feature Graphs

This class of models is based on a structural description of a part in terms of locally detectable primitive features and the geometric relations between pairs of these features. Models which are based on local rather than global features have the following advantages: (a) local features may be cheaper to compute because they are simpler and can be selectively (sequentially) detected, (b) models are less sensitive to minor differences in instances of a given object type, (c) if a few local features are missing (due to noise or occlusion) it may still be possible to recognize the object on the basis of the remaining features associated with the model, and (d) since a few types of local features are often sufficient to describe a large number of complex objects, it is possible to specify only a few types of local feature detectors which are applied to the image. A disadvantage with this type of model is the fact that a large number of features must be detected and grouped together to recognize an object. Thus the combinatorial explosion issue must be handled by the matching algorithm.

5.3.1. Models

Yachida and Tsuji [Yac77] use a simple kind of feature graph representation plus the use of a two level model (for coarse-to-fine processing) to speed the search process. Each object is described by a set of models, one for each possible viewpoint. Each model contains a coarse representation of the object using global features such as area and elongatedness, plus a description of the outer boundary (in polar coordinates). Each component extracted from an image is compared with each coarse model to determine if it is sufficiently similar to warrant further comparison. Object boundaries are compared by using the cross correlation as the measure of shape match.

The fine level of representation of each model is based on a higher resolution image and consists of a list of features such as outer boundary, holes, edges and texture. Associated with each feature is an attribute list, location (relative to the object's centroid), and the expected likelihood that the feature can be extracted reliably. Features are ordered in the model by their reliability value.

Chen *et al.* [Che80] also model objects using local feature graphs. Each node is a (feature type, position) pair and arcs connect pairs of nodes when an edge connects this pair of features on the part. Feature types are corners and small holes. Feature position is specified using an object-centered coordinate system.

Bolles and Cain [Bol82] have developed a sophisticated modeling system for 2-D objects called the local-feature-focus method. Two types of local features are used: corners and regions. An object model consists of three parts. First, a polygonal approximation of the object's borders. Second, a list of local features, where each is specified by a unique name, its type, position and orientation relative to the object's centroid, and rotational symmetries about the centroid. Position and orientation values also have associated allowable tolerances. Third, for each distinct feature type, an unambiguous description of each possible instance of this feature type in all the models is determined. In this way each possible feature type can be used as a "focus feature." Each structurally different occurrence of a given feature type has an associated feature-centered subgraph description containing a sufficient set of "secondary" features (and their relative locations) to uniquely identify the given focus feature and determine the position and orientation of the object.

A semi-automatic procedure is currently used to construct these focus feature tables. First, all possible distinguishable local feature types are determined over all stable viewpoints of all objects. The extraction of features is automatically performed by analysis of computer-aided design (CAD) models of the objects. Next, rotational and mirror symmetries are determined in order to identify all structurally equivalent features. For each structurally different feature, select a set of nearby features which uniquely identifies the focus feature, and construct a graph description of these features and their relations. Feature types are ranked by the size of their associated feature graphs, i.e., in increasing order of the sum of the number of secondary features needed to describe all instances of the given focus feature.

An automatic system for transistor wire bonding has been implemented at Hitachi [Kas76]. The model consists of three sets of three 12 by 12 binary templates which are selected by the user from three different orientations of a given prototype chip. For each triple of patterns in a set, an associated distance and direction (relative to the image's x-axis) pair are computed from the same binary image of the chip used to define the templates. Chips are assumed to be of a fixed size (camera position above table is fixed); orientation of a chip is fixed with a tolerance of up to fifteen degrees in either direction. Empirically it was determined that a triple of templates is a reasonable model for rotations of up to seven degrees from the normal orientation. Therefore, in order to meet system orientation specifications, three sets of templates are selected by the user with the prototype chip positioned at orientations -10, 0, and 10 degrees from the normal orientation.

The SIGHT-I system also locates integrated circuit chips by using a set of local templates [Bai78] This model consists of the specification of the possible relative positions of the four corners of a chip, and a set of four 4 by 4 templates is used to evaluate the probability that a corner is present at a given position.

5.3.2. Features

The feature extraction process in the system developed by Yachida and Tsuji [Yac77] is divided into several stages using the idea of "planning", that is, knowledge of the structure of an object guides the feature extraction module in a top-down manner. Simple features are detected first in a coarse resolution image and then more complex features are sought based on the locations of the coarse features. Industrial parts used in their demonstration are parts of a gasoline engine. In the preprocessing stage, a low resolution version of the image is analyzed and outlines of objects are detected by thresholding. Each outline is then analyzed separately, using a high-resolution image of the region of interest to extract a finer outline of the object. Local histogramming and dynamic thresholding based on 11 by 11 windows using the method in [Chow and Kaneko] are used in this step. Next, the object's gross properties, such as size, thinness ratio, and shape are computed. This coarse description of the object is used to select candidate models for matching and to guide the extraction of finer features for final recognition. There are four features extracted in the fine-resolution processing stage and they include circle, line, texture, and small hole. Each feature is extracted from a search region around the expected location in the gray-scale image. The circle detector uses thresholding as in the preprocessing step; the line finder, using dynamic pro gramming, searches for the optimum sequence of edge points in the region that maximizes a measure of goodness; the texture detector measures edge strength per unit area and average edge direction; the small-hole detector uses neighbor merging to locate circular objects.

In [Che80] Chen *et al.* estimate the pose of workpieces using the 3-D locations of at least three noncollinear feature points. The location of features is computed using trigonometric relations between corresponding features from two stereo views of the workpiece. Local image features include small holes and corners. Corner and small hole detection are based on the diameter-limited gradient direction histograms [Bir79] in which intensity variations in several directions and various heuristic thresholds are examined. Detected features from the image are evaluated to eliminate redundant features. The resultant corner points are fine-tuned for accuracy by fitting a pair of lines in an 11 by 11 window. The intersection of the two lines yields the final corner location. Finally, the interfeature distances between every pair of features are computed. Workpiece examples used in the experiments include simple planar industrial parts and 3-D block objects.

In the recognition system developed by Bolles and Cain [Bol82] a few key features, such as hole and corner are used. The system locates holes by finding small regions in the binary image and extracts corners by moving a jointed pair of chords around the boundaries and comparing the angle between the chords to the angles defining the different types of corners. This corner finder is believed to have difficulties with rounded corners. Relational features such as distances between features are used to describe position and orientation of objects. Another set of useful features used in this system is the symmetries of the object. Both the rotational and mirror symmetries of binary patterns are extracted automatically, using the method in [Bol79a]. They are useful in the reduction of the number of features to be considered, since symmetrical objects usually have duplicate features.

In most of the recognition systems for IC alignment and bonding, multiple template matching procedures are used. Features used for template matching are distinct patterns such as corners and bonding pads. Relational features such as the distance and angle between pairs of successfully matched templates are also used. In most cases these features are extracted by thresholding; e.g., see Hitachi's transistor wire-bonding system [Kas76].

In the SIGHT-I system developed by Baird [Bai78], a coarse processing stage is applied to the gray-scale image before the relational template matching step. In this step, the approximate orientation of the chip is determined by analyzing the edge-orientation histogram to find the most prominent edge orientation. This enables the matching stage to search for corners in known orientations.

5.3.3. Matching

Yachida and Tsuji [Yac77] have developed a matching process which examines the current information obtained from the scene and the structural models of objects to propose the next matching step. The model relates features at a coarse resolution with more detailed features at a fine resolution enabling the matching to be performed using simple features as cues. Given a tentative match between an image component and an object model based on the coarse model features, the fine model features are then successively compared. The object boundary matched at the coarse level determines a tentative match angle of rotation. For a given feature extracted from the image, a measure of the dissimilarity between it and each of the model features is computed. A cumulative dissimilarity measure is kept for each active model. When a model's dissimilarity exceeds a threshold, the model is rejected as a possible match. After the current feature has been compared with each of the remaining candidate models, a next-feature proposer analyzes the features described in these candidate models and proposes the most promising feature among them as the feature to be examined next for recognizing the input object.

Chen *et al.* [Che80] estimate workpiece pose using a sequential pairwise comparison algorithm in which a feature point is matched in turn to all model feature points of the same type. The matching process starts with the selection of the feature point which has the highest confidence. The remaining feature points are then matched with all model points. In this step, feature type, interfeature distance, and edge information are used as matching criteria, and redundant matched points are deleted. If enough feature points are successfully matched with the model points, and a transformation test, used to eliminate problems due to symmetry, is passed, a match is considered to be found. Finally, the pose of the workpiece is computed from the correspondence between workpiece features and model features.

The matching procedure of the Local-Feature-Focus method [Bol82] uses a graph-matching technique to identify the largest cluster of image features matching a cluster of model features. The procedure first retrieves models of industrial parts together with the list of focus features and their nearby features. Figure 8 shows an example of a focus feature. Then, for each image, the system locates all the potentially useful local features, forms clusters of them to hypothesize part occurrences, and finally performs template matches to verify these hypotheses.

After locating all the features found in the image, the system selects one feature (the focus feature) around which it tries to find a cluster of consistent secondary features. If this attempt fail to lead to a hypothesis, the system seeks another potential focus feature for a new attempt. As it finds matching features it builds a list of possible model-feature-to-image-feature assignments. This list is transformed into a graph that is analyzed by a Figure 9 shows the possible assignments and the resulting graph. The result from the maximal-clique algorithm is used to hypothesize an object. At the final stage, two tests are used to verify the hypotheses by looking at other object features and checking the boundary of the hypothesized object. Figure 10 shows a recognition example.

In the transistor wire bonding system developed by Kashioka *et al.* [Kas76], a multiple template matching method is used to locate the position of the chip. A set of characteristic 12 by 12 binary templates is used. In the matching process, the system searches a 160 by 120 image for the local region which best matches the first template. It then searches for the best match to a second

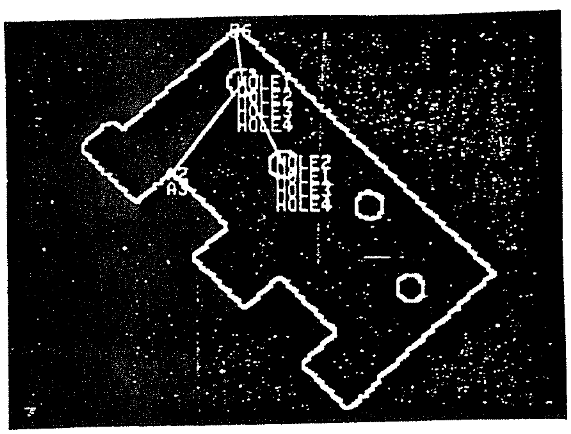


Figure 8. An example of a focus feature on a hinge. Nearby feature found around a hole and their lists of possible model features.

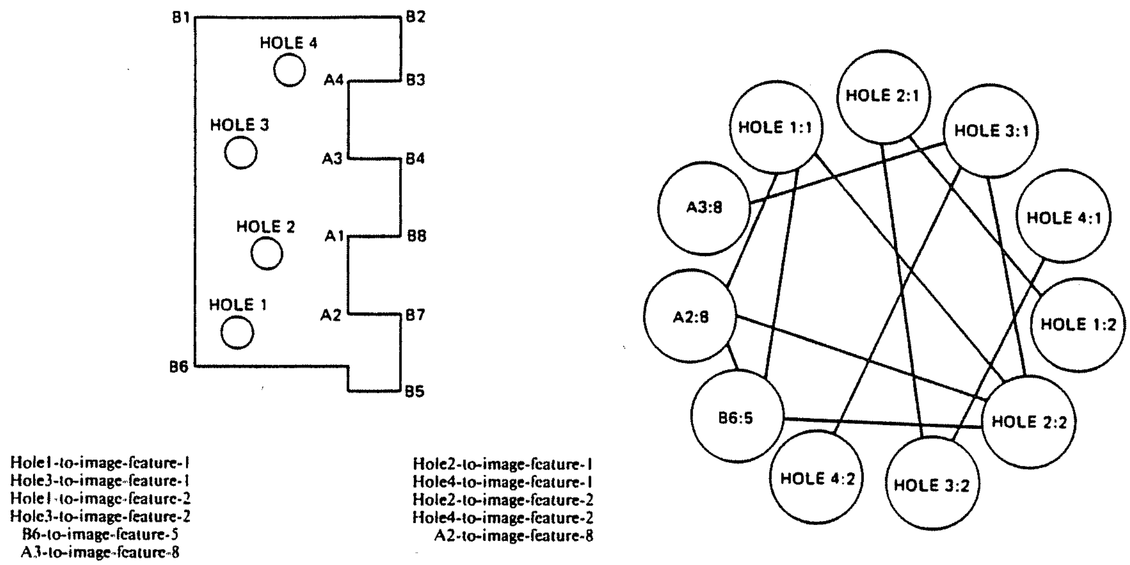


Figure 9. (a) Definitions of the model features of the hinge. (b) List of model-feature-to-image-feature assignments. (c) Graph of pairwise consistent assignments. Each node represents a possible assignment of a model feature to an image model. Two nodes are connected if the two assignments they represent are mutually consistent.

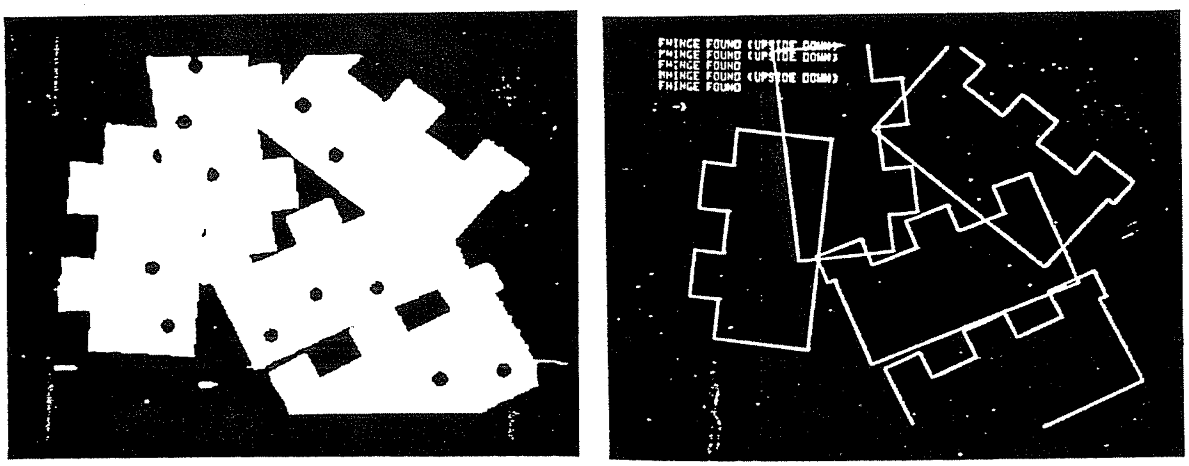


Figure 10. Image of five hinges and the recognition result.

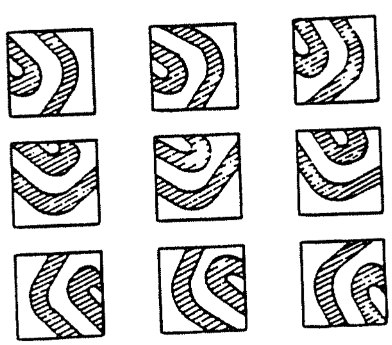
template. From these positions, a distance and a direction angle are computed and compared to the values predetermined from the geometry of the chip. If the measurements are not close to the predefined values, a third local template is used, and measurements are again computed. Locations of bonding pads are computed using the measurements obtained from the multiple local template matching. Figure 11 shows the set of templates and the matching process.

5.3.4. Other Studies

Cheng and Huang [Che82] have developed a method for the recognition of curvilinear objects by matching relational structures. The boundary of an object is segmented into curve segments and then into chords. Attributes (parallel, symmetric, adjacent, etc.) associated with the chords are used as the nodes in the relational structure representation of the object. The matching is based on a star structure [Che81] representation of the object. The recognition of overlapping tools has been shown.

Segen [Seg83] has developed a method to recognize partially visible parts by using local features computed from an object boundary. The local features used are defined at points of local maximum and minimum of contour curvature. A local feature from the image is matched with a feature from the model, and they determine a transformation (rotation and translation). All features are used in the matching and a set of transformations are generated. The algorithm then clusters together features that imply similar transformations. The center of each cluster is used to define a candidate transformation that may possibly give a partial match. Finally, these candidate transformations are tested with a point by point matching of the image contour and the transformed model contour.

Westinghouse's gray-level robot vision system uses a very simple form of relational feature graph approach. In one of their reported studies [Sch83], edges are used to form corners where a corner is defined as two intersecting edge vectors. The matching algorithm searches for four edge vectors forming two opposing corners such that the center of the line segment joining the corner pair coincides with the part center. The assumption that the object center and the two opposing corners are collinear restricts the applicability of the algorithm to a limited type of industrial parts.



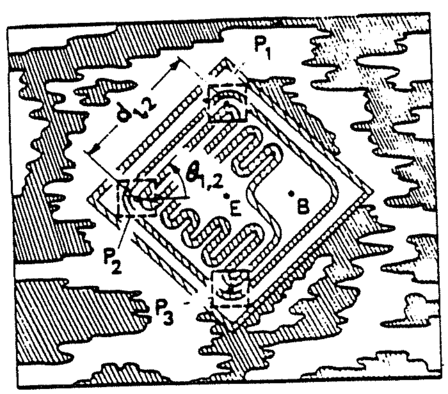


Figure 11. Nine corner templates and the recognition of the circuit position by evaluating relations between pairs of matched templates.

In semiconductor chip manufacturing, each die is being visually inspected for the bonding of the die onto the package substrate and the bonding of wires from the die pads to the physically larger package leads. The process involves the recognition of the chip boundary, the determination of the chip position and orientation, and the recognition of bonding pads. Conventionally, human operators have to perform all of these functions. Until recently, there exists a number of automatic die-bonding and wire-bonding systems operating the production lines. Most of these systems are based on relational features and associated matching algorithms. Some other IC location recognition systems include [Mes77], [Iga79], [Hor75a], [Kaw79], [Hsi79].

6. $2\frac{1}{2}$D Surface Representations

The previous section presented methods based on image intensities, deriving features from gray level or binary images to represent the projection of an object in two dimensions. In this section we present another class of methods which is also viewer-centered, but which is based on physical scene characteristics of a single view of an object. This representation maintains information in register with the original gray-scale image, and includes "intrinsic images" [Bar78], "2 ½-D sketch" [Mar78], "needle map" [Hor79], "parameter images" [Bal81b], and "surface orientation map" [Bra82]. Intrinsic scene parameters include surface range, orientation, discontinuities, reflectance, illumination, color, and velocity. Since this local information is obtained over a whole region within some boundaries, it is more robust than the edge-based techniques used with many of the 2-D representations presented in the previous section.

All of the methods in this section use scene surface properties derived from a single viewpoint to define features and construct models. Hence if multiple views of an object are required, each is modeled independently. We have included range maps as part of this class of representation despite the fact that 3-D data are used. This is because the models which use these data are viewer-centered and emphasize the description of observable surface features from a single viewpoint. Models which describe a complete (viewpoint-insensitive) 3-D object are included in the next section as 3-D representations.

Most current research is focusing on the problem of how to compute these intrinsic surface maps. See [Bra82], [Bal82], [Mar82b], [Bar78], [Jar83], for example, for surveys of many applicable techniques. This survey will not consider this "preprocessing" step.

Of particular interest for applications in industrial part recognition is the computation and use of range maps and local surface orientation (needle) maps. Jarvis [Jar83a] and Poje and Delp [Poj82] give recent overviews of range finding techniques using both active and passive methods. Active methods include ultrasonic and light time-of-flight measurement, and structured light projection using a plane or grid of light. While early methods of these types have been slow, expensive, and low accuracy, many recent improvements have been made; see, for example, [Jar83b], [Kan81], [Alt81], [Agi82], [Pip83].

Instead of extracting a range map, other researchers are focusing on obtaining local surface orientation as a descriptor of surface shape. This includes such methods as shape from shading [Hor75b], shape from texture (e.g. [Ken80], [Wit81]), and shape from photometric stereo [Woo78].

One method of computing surface orientation which shows considerable promise for industrial part recognition is called photometric stereo [Woo78]. Local surface orientation is computed using a reflectance map for each of three different incident illumination directions but from a single viewing direction.

Since an object point corresponds to the same pixel in each of these three gray level images, the surface orientation at this point can be obtained from the intersection of isobrightness contours in the reflectance maps associated with each light source. The method has been implemented very efficiently by inverting the reflectance maps into a lookup table which gives surface orientation from a triple of gray levels [Sil80]. So far the technique has been defined for objects containing Lambertian and specular surfaces [Woo78], [Ike81a], and error analysis has been performed [Ray81].

To date researchers have developed only a few model-based recognition systems which are based on features derived from one or more surface maps. Hence applications to industrial part recognition have yet to be extensively investigated. In the remainder of this section we present most of the techniques which have been studied. All of these methods are based on features derived from either a range map or a needle map. We expect that considerable future work will be devoted to expanding this class of techniques.

6.1. Models

Oshima and Shirai [Osh83] construct a relational feature graph in which nodes represent planar or smoothly curved surfaces extracted from a range map, and arcs represent relations between adjacent surfaces. Surface types include planar, ellipsoid, hyperboloid, cone, paraboloid, cylinder, and others. For each pair of adjacent regions the type of intersection (convex, concave, mixed, or no intersection), angle between the regions, and relative positions of the centroids is stored. Figure 12 illustrates this relational graph description for a scene containing three objects. If objects may be viewed from multiple viewing positions, then a separate relational graph must be constructed for each view and these models are treated independently by the matcher. Partial

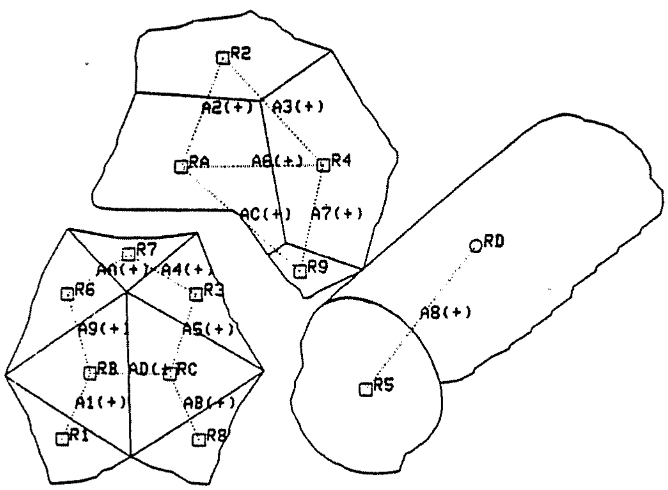


Figure 12. Relational graph description of planar and curved surfaces derived from a range map.

occlusion of certain secondary planar surfaces is allowed, although the extent is dependent on the predefined thresholds used by the matcher. Currently, curved surfaces may not be occluded in the scene.

Nevatia and Binford [Nev77] construct a relational graph description of each view of an object using a set of "generalized cones" corresponding to elongated subparts of the object. In particular, cones are represented as "ribbon-like" descriptors [Bro79] containing 2-D cross-sections of range discontinuity points. Given a set of these ribbons defining an object, a set of "joints" are constructed indicating which ribbons are adjacent to each other. A joint description includes an ordered list of ribbons connected to it and a designated dominant ribbon having the largest width. A relational graph is constructed in which joints are represented by nodes and ribbons by arcs. Figure 13 shows the spines of the ribbons detected in a scene containing a reclining doll and the resulting relational graph description. In addition, a set of coarse descriptors are associated with this object graph, including number of ribbons, number of elongated ribbons, number of joints, bilateral symmetries, and a set of "distinguished" ribbons having the largest widths. For each distinguished piece of an object a three bit description code is used to permit efficient organization and search of the set of models. The three descriptors which are encoded are the part's connectivity, type (long or wide part), and

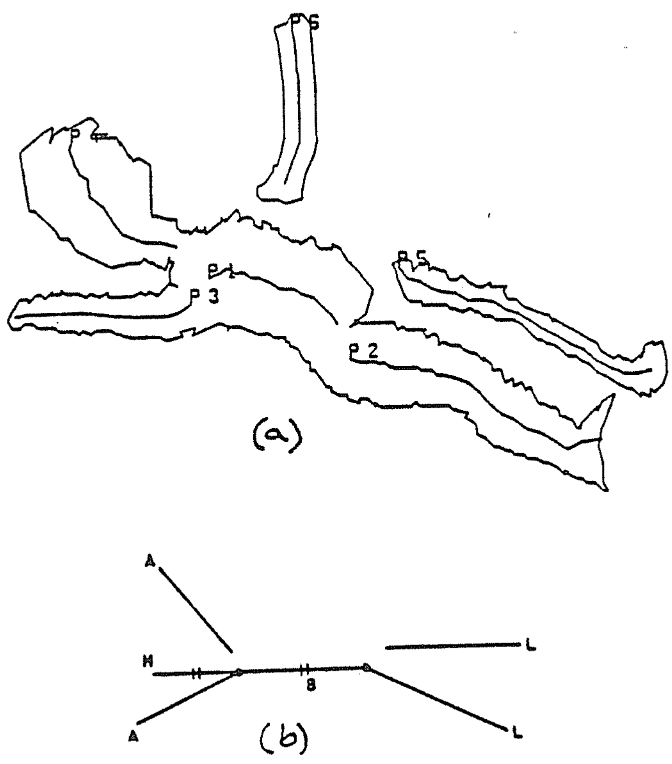


Figure 13. (a) Spines of ribbons detected in a range map for a scene containing a reclining doll. Range discontinuities detected define the boundary of the object. (b) The relational graph constructed from (a).

whether it is conical or not. Models are sorted by their description code and secondarily by the maximum number of parts attached at either end of the distinguished piece.

6.2. Features

Many researchers, including [Agi72], [Shi72], [Osh79], [Pop75], [Hen82a], [Hen82b], [Mil80], [Dud79], [Bol81], have investigated using range maps as the basis for segmenting an image into regions, grouping points into planar surfaces, cylindrical surfaces, surfaces on generalized cylinders, and other smoothly curved surfaces. For example, Oshima and Shirai [Osh79] group points into small surface elements, then the equation of the best plane surface through each of these elements is computed, and finally these surface elements are merged together into maximal planar and curved regions.

Alternatively, range maps can be segmented by locating discontinuities in depth. For example, Sugihara [Sug79] segments a range map by finding such edges. To aid this process, a "junction dictionary" is precomputed listing all possible ways junctions can occur in range maps for scenes containing only trihedral objects. The dictionary is then used to guide the search in the range map for edges of the appropriate types.

For the most part, however, the resulting surface and boundary descriptions have not been used to define object models. In addition to the following methods which have defined 2 ½-D models, several of the methods presented as 3-D multi-view feature models (see Section 7) could also be included here when considering only a single view.

Oshima and Shirai [Osh83] compute features of each region including surface type (planar, ellipsoid, cone, cylinder, etc.), number of adjacent regions, area, perimeter, compactness, occlusion, minimum and maximum extent, and mean and standard deviation of radius.

As an alternative to extracting planar and curved surfaces from range maps, some researchers have developed techniques for detecting surface boundaries by detecting and linking points at which range discontinuities occur. Nevatia and Binford [Nev77] use a range map to derive a boundary description of a given view of an object. Rather than use this global structural feature to describe an object, they immediately construct a relational graph using "ribbon-like" primitives [Bro79] to describe subparts. A ribbon is the 2-D specialization of a 3-D generalized cylinder [Bin71]. Ribbons are specified by three components: spine, cross-section, and sweeping-rule. By sweeping a planar cross-section at a constant angle along a spine according to the sweeping-rule, a planar shape is generated. They first construct a set of local ribbons restricted to having straight axes in eight predefined directions and smoothly varying cross sections. This is done by linking midpoints of cross sections (runs of object points perpendicular to the axis direction) which are adjacent and have similar cross sections. These local ribbons are then extended by extrapolating the axes of the local ribbons and constructing new cross sections. This process allows the resulting axis to curve smoothly. In general, a single part of an object may be described by (parts of) several overlapping ribbons. To reduce this redundancy, ribbons are deleted which are not as elongated or rectangular as another ribbon which overlaps it. Associated with each ribbon is a crude description of its shape given by its axis length, average cross section width, elongatedness, and type (conical or cylindrical).

6.3. Matching

Oshima and Shirai [Osh83] match an observed relational graph of surface descriptions with a set of graphs for each viewpoint of each object modeled. First, regions corresponding to maximal smooth surfaces are extracted from the range map of a given scene. A "kernel" consisting of either a single region or a pair of adjacent regions is then selected from the surface descriptors of the given scene in order to guide the search for matching model graphs. The kernel represents regions with high confidence of being found; criteria include no occlusion, planar surfaces, and large region area. Next, an exhaustive search of all model graphs is performed, selecting as candidate models all those which contain regions which match the kernel. Finally, the system performs a depth-first search which attempts to determine the correspondence between each remaining region in the current candidate model and the regions extracted from the scene. A model region and a scene region match if their region properties are similar, all adjacencies to previously matched regions are consistent, and the properties of all new relations between regions are similar. This process is repeated for other kernel regions in the scene until a globally consistent interpretation is achieved in which each scene region is found to correspond to exactly one model region. If multiple consistent interpretations are possible, then the system returns each one. Figure 14 illustrates this matching process.

Nevatia and Binford [Nev77] match a relational graph extracted from a scene with each relational graph describing a model. First, a set of candidate models is determined by comparing the properties of the distinguished ribbons in the scene with those distinguished ribbons associated with each of the models. For each such candidate model a finer match is performed by comparing other ribbons, pairing a model ribbon with a scene ribbon if their properties are similar and all connectivity relations are consistent in the

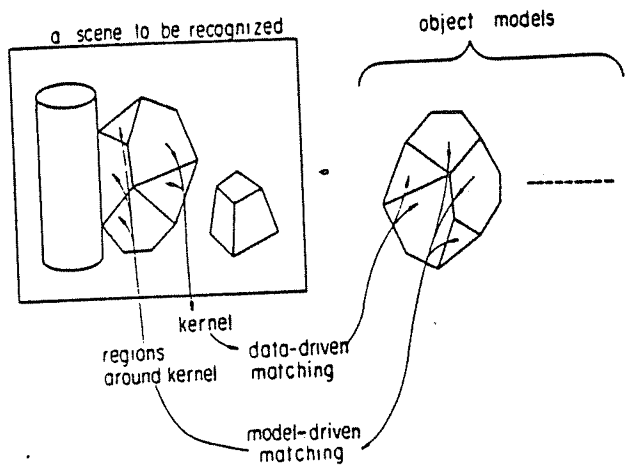


Figure 14. Matching kernel surfaces in a scene with model surfaces is used to select candidate models in [Osh83]. Neighboring surfaces are then matched in order to verify a candidate model.

current pair of matched subgraphs. The scene graph is allowed to match a model graph even if not all model ribbons are present in the scene graph (hence partial occlusion is permitted), but the scene graph may not contain extra ribbons which are not matched by any ribbon in the model graph.

7. 3-D Object Representations

If we assume that an object can occur at an arbitrary attitude, then the model must provide an explicit description of the object from all viewing angles. Image-space (2-D) and surface-space (2 ½-D) representations are viewer-centered and each distinct view is represented independently. Thus when multiple views of complicated objects are permitted (as in the general bin picking problem), a viewpoint-independent, volumetric representation may be preferred [Mar78]. In contrast to the previous representations, a single model is used to represent all possible viewing positions around a given object. In addition, in an industrial automation setting in which the vision system must be integrated with objects represented in CAD databases, an object-space model may be convenient because of its compatibility.

Researchers have investigated two main types of 3-D representations. These are (1) *exact representations* using surface, sweep, and volume descriptions, (2) *multi-view feature representations* in which a set of 2-D or 2 ½-D descriptions are combined into a single composite model. This includes the specification of a set of topologically-distinct views or a uniformly sampled set of 2-D viewpoints around an object. The first representation method completely describes an object's spatial occupancy properties, whereas the second representation only represents selected visible 2-D or 2 ½-D surface features (and sometimes their 3-D spatial relationships).

Exact representations include the class of complete, volumetric methods based on the exact specification of a 3-D object using either surface patches, spines and sweeping rules, or volume primitives. Object-centered coordinate systems are used in each case. See, for example, [Req80], [Bad78], [Bal82] for a general introduction to this class of representations. Surface model descriptions specify an object by its boundaries or enclosing surfaces using primitives such as edge and face. Baumgart's "winged edge" representation for planar polyhedral objects is an elegant example of this type of model [Bau72]. Volume representations describe an object in terms of solids such as generalized cylinders, cubes, spheres, rectangular blocks, etc. The main advantage of this class of representations is that it provides an exact description which is object-centered. The main disadvantage is that it is difficult to use in a real-time object recognition system since the processing necessary to perform either 2-D to 3-D or 3-D to 2-D projections (for matching 2-D observed image features with a 3-D model) is very costly. For example, in ACRONYM [Bro81] camera constraints are built in so as to limit the number of 3-D to 2-D projections which must be hypothesized and computed at run-time.

The second type of 3-D representation method is based on the representation of observable features in each of several views of an object. In the limiting case this includes the work on storing 2-D descriptions for each stable configuration of an object. We restriction our discussion here to coordinated representations of multiple views which permit the specification of efficient matching procedures which take advantage of intra-view and inter-view feature similarities.

One class of multi-view representations is based on the description of the *characteristic views* of an object. This requires the specification of all topologically-distinct views. Koenderink and vanDoorn [Koe76a], [Koe76b], [Koe79] are studying the properties of the set of viewing positions around an object, and the qualitative nature of how most viewing positions are stable. That is, small changes in viewing position do not affect the topological structure of the set of visible object features (i.e., point and line singularities). Based on the topological equivalence of neighboring viewpoints, they define an "aspect graph" of feature-distinct viewpoints (see Figure 15).

Fuchs *et al.* [Fuc80] have also used this idea to perform a recursive partitioning of a 3-D scene using the polygons which describe the surfaces of the constituent 3-D objects. That is, a "binary space partitioning" tree is constructed in which each node contains a single polygon. Polygons associated with a node's left subtree are those contained in one half-space defined by the plane in which the current polygon lies; the polygons in the right subtree are the ones in the other half-space. Using this structure they perform hidden surface elimination from a given viewpoint by a simple inorder tree traversal in which subtrees are ordered by their "visibility" from the given viewpoint. In this representation each leaf defines a characteristic view volume; hence the set of leaf nodes define a partition of 3-space into distinct viewing positions.

Another type of multi-view representation is to discretely sample the "viewing sphere" of all possible viewpoints (at a fixed distance) around an object, storing a viewer-centered description for each sample viewpoint. This ***discrete view-sphere representation*** has the advantage that it can be pre-computed from a complete 3-D volumetric description and provides a description which is compatible with the features extracted from a test image at run-time. Thus it is a more convenient representation, and yet provides sufficient accuracy of description except at pathological viewing positions.

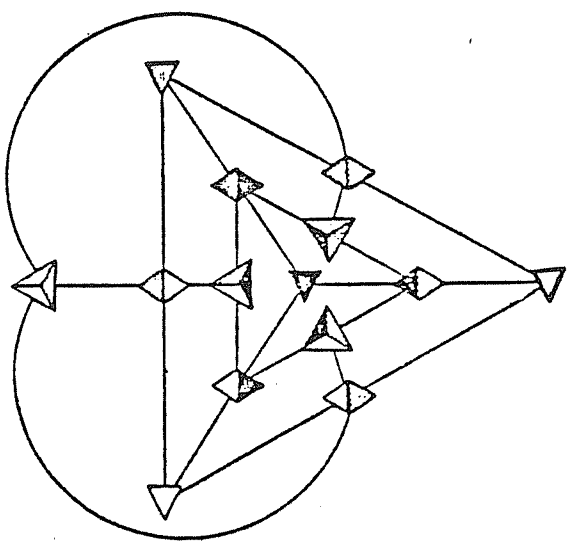


Figure 15. The aspect graph for a tetrahedron. Nodes are of three types representing whether one, two, or three faces are visible. Arcs connect adjacent viewing patches.

7.1. Models

Shneier [Shn79], [Shn81] constructs 3-D *surface* models from a set of light stripe images of the object to be modeled. Each distinctly different plane surface which is extracted is represented by a unique node in a "graph of models" which describes all models to be recognized. Associated with each node is a set of properties which describes the surface's shape, and a set of pointers to the model names in which this primitive shape is a part. Thus if two surface shape descriptions are similar in the same or different objects, they are represented by a single node in the graph. Arcs connect pairs of nodes using a set of predefined relation schemata, for example the "is adjacent to" relation. Arguments to relation schemata are surface descriptions, not actual surfaces. Relation schemata also index the models in which they occur and the primitives that form their arguments. Thus nodes and arcs in the graph of models may be shared within models and across models. This integration of multiple object models into a single graph has the advantages of being very compact and enabling a rapid indexing scheme to be used.

Goad [Goa83] builds a *multi-view* feature model of an object by constructing a list of object features and the conditions on when each is visible. The single object feature used is a straight line segment representing a portion of the object's surface at which either a surface normal or a reflectivity discontinuity occurs.

The set of possible viewing positions is represented by partitioning the surface of a unit "viewing sphere" into small, relatively uniform size, patches. The current implementation uses 218 patches. To represent the set of positions from which a given edge feature is visible, a bit map representation of the viewing sphere is used to encode whether or not the feature is wholly visible from each patch on the sphere (i.e., a line's projection is longer than a threshold). Thus each feature is stored as a pair of endpoint coordinates plus 218 bits to describe its visibility range.

The matching procedure used with this model requires a sequential enumeration of model edges which are successively matched with image edges. In order to improve the run-time efficiency of the search for a consistent set of matches which determines a unique view position, it is important to select an order which presents edges in decreasing order of expected utility. This can be done by preprocessing the list of features in the model using each edge's (a) likelihood of visibility, (b) range of possible positions of the projected edge, and (c) focusing power (i.e. if a match is made, how much information about restrictions on the camera position becomes known). Combining these factors for a given model results in a predetermined ordering of the best edge to match next at any stage of the search.

Horn [Hor82] and his colleagues use a *multi-view* feature model in which features are derived from the needle map for an object. That is, they model each viewpoint of an object by the distribution of its surface orientation normals on the Gaussian sphere, ignoring positional information by moving all surface normals to the origin. By associating a unit of mass with each point on the unit sphere, we obtain a distribution of mass called the "Extended Gaussian Image" [Smi79]. Segments of developable surfaces such as planes and cylinders map into high concentrations of points in known configurations. Figure 16 illustrates this representation for two convex objects. Ikeuchi [Ike81b], [Ike82], [Ike83a], [Ike83b] models an object by a set of 240 (normalized) extended Gaussian images, one for each possible viewing direction on a uniformly sampled viewing sphere. More specifically, a two dimensional table is constructed for each possible viewpoint, mass distribution pair; an element in this table stores the mass (surface area) corresponding to the total surface description for the given

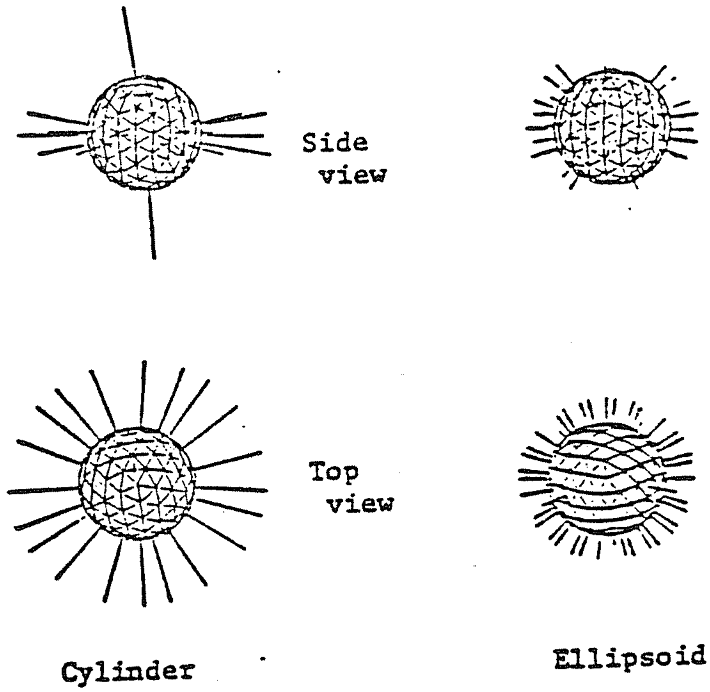


Figure 16. Two views of the extended Gaussian images for a cylinder and an ellipsoid.

viewpoint. Figure 17 shows the normalized extended Gaussian images for four viewpoints of an object. If multiple objects are to be recognized then a table is constructed for each object.

Brooks' ACRONYM system [Bro81], [Bro83a], [Bro83b] constructs *sweep* models using part/whole hierarchical graphs of primitive volume elements described by generalized cylinders. A generalized cylinder (GC) describes a volume by sweeping a planar cross section along a space curve spine; the cross section is held at a constant angle to the spine and its shape is transformed according to some sweeping rule. The user constructs a tree for each object, where nodes include GC descriptions and arcs indicate the "subpart" relation. The tree is designed to provide a hierarchical description of an object, where nodes higher in the tree correspond to more significant parts in the description. For example, the root of an "electric motor" tree describes the cylinder for the large cylindrical body of the motor. Arcs from this node point to nodes describing cylinders for the small flanges and spindle which are part of a lower priority level of description of the motor.

Each GC has its own local coordinate system and additional affixment arcs between nodes specify the relations between coordinate systems. If multiple parts of the same type are associated with a single object, they are represented by a single node in the tree with a quantity value and a set of coordinate transformations specifying the location of each part. Furthermore, in order to allow variations in size, structure, and spatial relationships in GC descriptions, any numeric slot in a node's description may be filled by an algebraic expression

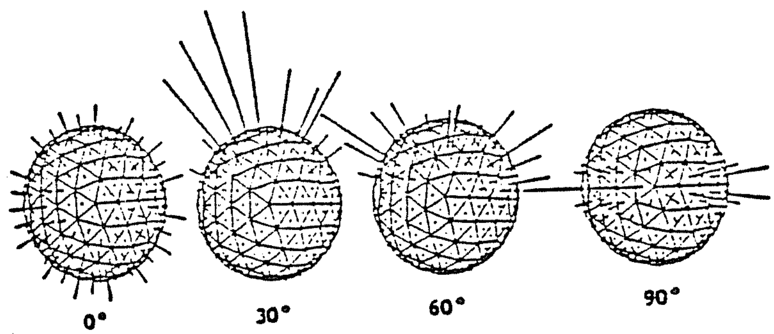


Figure 17. Four of the normalized extended Gaussian images for an object. From each such image a mass distribution feature is stored in a lookup table to represent the given viewpoint.

ranging over numeric constants and variables. Classes of objects are specified by constraints, i.e., inequalities on algebraic expressions which define the set of values which can be taken by quantifiers.

A scene is modeled by defining objects and affixing them to a world coordinate system. A camera node is also included specifying bounds on its position and orientation relative to the world coordinate system.

To aid the matching of models with image features, the user constructs from the static object graph a model class hierarchy called the restriction graph. That is, the sets of constraints on quantifiers in the object graph are used to build a specialization hierarchy of different classes of models. The root node represents the empty set of constraints for all restriction graphs. A node is added as the child of another node by constructing its constraint list from the union of its parent's constraints and the additional constraints needed to define the new node's more specialized model class. A pointer is also added to a node in the object graph which defines the volumetric structure of the new node's model (as specialized by the given constraints). Thus an arc in the graph always points from a less restrictive model class (larger satisfying set of constraints) to a more restrictive one (smaller satisfying set). Figure 18 illustrates a restriction graph for classes of motors and Figure 19 shows three instances associated with the leaf nodes' sets of constraints. During the matching process other nodes are also added to the restriction graph in order to further specialize a given model for case analysis, or to specify an instance of a match of the model to a set of image features.

Bolles *et al.* [Bol83] use a *surface* model as the primary structure for generalizing their "local-feature-focus" method to 3-D. A model consists of two parts: an augmented CAD model and a set of feature classification networks. The augmented CAD model is similar to Baumgart's [Bau72], describing edges, surfaces, and vertices and their relations with one another. The feature classification network classifies observable features by type and size. For example, surface elements that have the same normal direction, and cylinders that have a common axis. Each feature contains a pointer to each instance in all of the augmented CAD models. Figure 20 illustrates this modeling method.

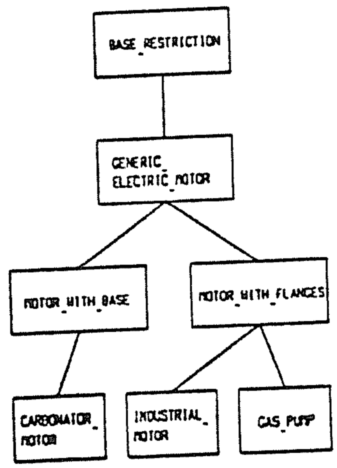


Figure 18. The restriction graph for classes of electric motors used in [Bro81].

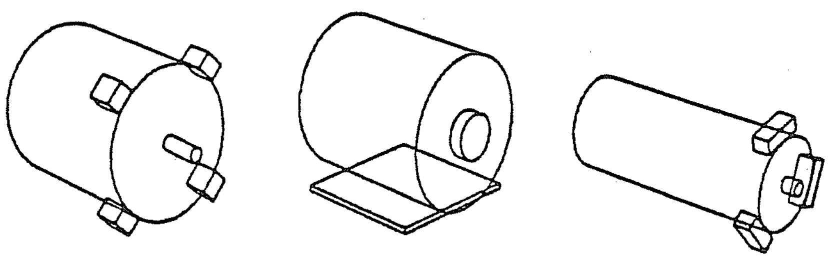


Figure 19.
Three instances of the model classes associated with the three leaf nodes in Figure 18.

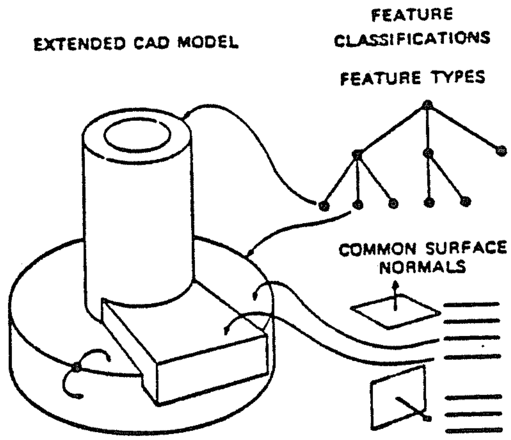


Figure 20. Bolles [Bol83] augmented CAD model and feature classification network.

Silberberg *et al.* [Sil83] model an object using a *multi-view* representation to define a Hough space of possible transformations of a set of 3-D line segments (edges) which are observable surface markings on the given object in a given viewpoint.

Chakravarty and Freeman [Cha82] define a *multi-view* model using characteristic views for recognizing curved and polyhedral objects. For a given object, they define a finite set of equivalence classes called "characteristic view partitions," which define a set of "vantage-point domains" on the sphere of possible viewpoints. Each topologically-distinct patch is described by a list of the visible lines and junctions in the given object. In order to reduce the number of patches in the partition of the view-sphere, they assume objects will occur in a fixed number of stable positions, and the set of possible camera positions is also limited. With these restrictions, two viewpoints are part of the same patch if they contain the same image junctions and lines with the same connectivity relationships, although the lengths of the lines may differ. A linear transformation describes features within a patch. An object is now modeled as a list of patch descriptors, where each list specifies the number of visible junctions of each of the five possible distinct types for this class of objects [Cha79].

7.2. Features

The principle features used in most 3-D recognition systems are based on surface properties such as faces, edges and corners. The references given in Section 6.2 for grouping range data into planar, cylindrical, and other smoothly curved surfaces are also used for 3-D surface description and modeling. Potmesil [Pot83b] constructs 3-D surface models from a series of partially overlapping range images by an iterative merging algorithm which first groups local surface patches into locally smooth surface sheets (using a quadtree representation) and then merges partially overlapping surface representations using a heuristic search procedure.

Goad [Goa83] restricts his model-based vision system to detect straight line segments (straight edges on an object). The edge detection algorithm is based on a program developed by Marimont [Mar82a]. The algorithm convolves the image with a lateral inhibition operator, detects zero-crossings in the convolved image, and then performs linking following by a simple segmentation thresholding. Three different types of objects, a universal joint casting, a keyboard key-cap, and a connecting rod, have been used in the experiments. Silberberg *et al.* [Sil83] also use line segments as the basis for constructing 3-D object models.

Ikeuchi [Ike83a] uses the complete surface orientation map in the form of the normalized Extended Gaussian Image as a global feature descriptor.

ACRONYM [Bro81a] uses "ribbons" and ellipses as low level features describing a given image. In this implementation only simple ribbons were used; a simple ribbon is defined by sweeping a symmetric width element normally along a straight spine while changing the width linearly with distance swept. Ellipses are used to describe the shapes generated by the ends of GCs. For example, ellipses describe ends of a cylinder and ribbons describe the projection of the cylinder body.

The extraction of these features is performed by the descriptive module of the ACRONYM system [Bro79]. First, an edge linking algorithm creates sets of linked edges (contours) from the image data. Linking edges into a contour is formulated as a tree searching problem searching for the best edge direction at a given point. A contour is retained only if it satisfies certain global shape criteria. Next, an algorithm fits ribbons and ellipses to the sets of contours by extracting potential boundary points of a ribbon from a histogram of the angles of the edge elements making up the contour. Finally redundant ribbons in a single area of the image are disambiguated. A graph structure, the *observation graph*, is the output of the descriptive module. The nodes of the graphs are ribbon and ellipse descriptions and the arcs linking the nodes together are relations between ribbons.

Bolles uses range data to detect surface discontinuities in an image. Two methods are used: detecting discontinuities occurring in 1-D slices of the range finder, and finding zero-crossings in the complete range map.

Chakravarty [Cha82] uses a list containing the number of occurrences of each of eight generalized junction types possible for planar and curved-surface objects. Lists are ordered by decreasing significance for recognition and organized into a hierarchical decision tree.

7.3. Matching

Shneier [Shn79] matches a set of observed planar surfaces with the graph of models for all possible objects by a two step procedure. First, for each observed surface which is sufficiently similar to a node in the graph of models, a node is created in the "scene graph" indicating this match. Since each node in the graph of models corresponds to one or more surfaces in one or more objects, each possibility is tested using a predefined set of procedures for deciding if an interpretation is possible for the observed surface, and for assigning confidences to these interpretations. A subgraph of the scene graph is created for each possible interpretation, and each surface/model-node pair is assigned to one or more such subgraphs. Next, the scene graph is traversed, deleting surfaces that are insufficiently substantiated and propagating constraints in order to remove multiple interpretations for a single surface.

Goad [Goa83] uses an elaborate sequential matching procedure with backtracking in his model-based vision system. The matching involves a search for a match between image and model edges. At any given time in the search, a hypothesis about the position and orientation of the object relative to the camera is used to restrict the search area to some reasonable bounds. The hypothesis is refine sequentially during the matching process.

The procedure starts with predicting the position and orientation of the image projection based on the current hypothesis. Then, a model edge is selected to match with image edges. If a match is found, the measured location and orientation of the new edge are used to update the hypothesis. The algorithm repeats the searching and updating until a satisfactory match of an object is found. If the algorithm fails to locate a predicted edge, the algorithm backtracks to use another image edge that has also been predicted as a good match.

Ikeuchi [Ike83a], [Ike83b] matches an observed extended Gaussian image (EGI) with each model EGI. To constrain the set of match tests which must be made for each pair, the observed EGI and model EGI mass centers are aligned, constraining the line of sight. Next, the observed and model spheres are rotated about the candidate line of sight so as to align their directions of minimum EGI mass inertia. These two constraints completely specify the alignment of the observed EGI with a model EGI. A match measure for a given pair of normalized EGI's is specified by comparing the similarity in their mass distributions; the model which maximizes this measure is the estimate of the observed line of sight. When multiple objects are present in a single scene, it is necessary to first segment the surface orientation map into regions corresponding to separate objects.

ACRONYM [Bro81a] predicts appearances of models in terms of ribbons and ellipses which can be observed in an image. Rather than exhaustively make predictions based on all possible viewing positions, viewpoint-insensitive symbolic constraints are used which indicate features which are invariant or quasi-invariant over a large range of viewing positions. To generate predictions a rule-based module is used to identify contours of model faces which may be visible. Case analysis is used to further restrict predictions and produce predicted contours in the viewer's coordinate system.

As a result of this constraint manipulation process a "prediction graph" is built in which nodes represent either specific image features or join prediction subgraphs containing lower level features. Arcs of the graph denote image relations between features, relating multiple feature shapes predicted for a single GC. Arcs are labeled either "must be," "should be," or "exclusive." Associated with a prediction graph is a node in the restriction graph which specifies the object class being predicted.

Matching is performed at two levels. First, predicted ribbons must match image ribbons, and second, these "local" matches must be globally consistent. That is, relations between matched ribbons must satisfy the constraints specified in the arcs of the prediction graph, and the accumulated constraints for each maximal subgraph matched in the observation graph must be consistent with the 3-D model constraints in the associated restriction node. Local matches of predicted ribbons with image ribbons also provides additional "back constraints" which are used to further restrict model parameters. Finally, matching is first done for GCs of highest priority in each model's object graph hierarchy in order to limit the search initially to include only the most important parts. Figure 21 illustrates the results of this method.

Bolles *et al.* [Bol83] use a matching scheme similar to that used for the "local-feature-focus" method [Bol82]. First, the system searches for features which match some model's feature, e.g., a cylinder with a given radius. This is

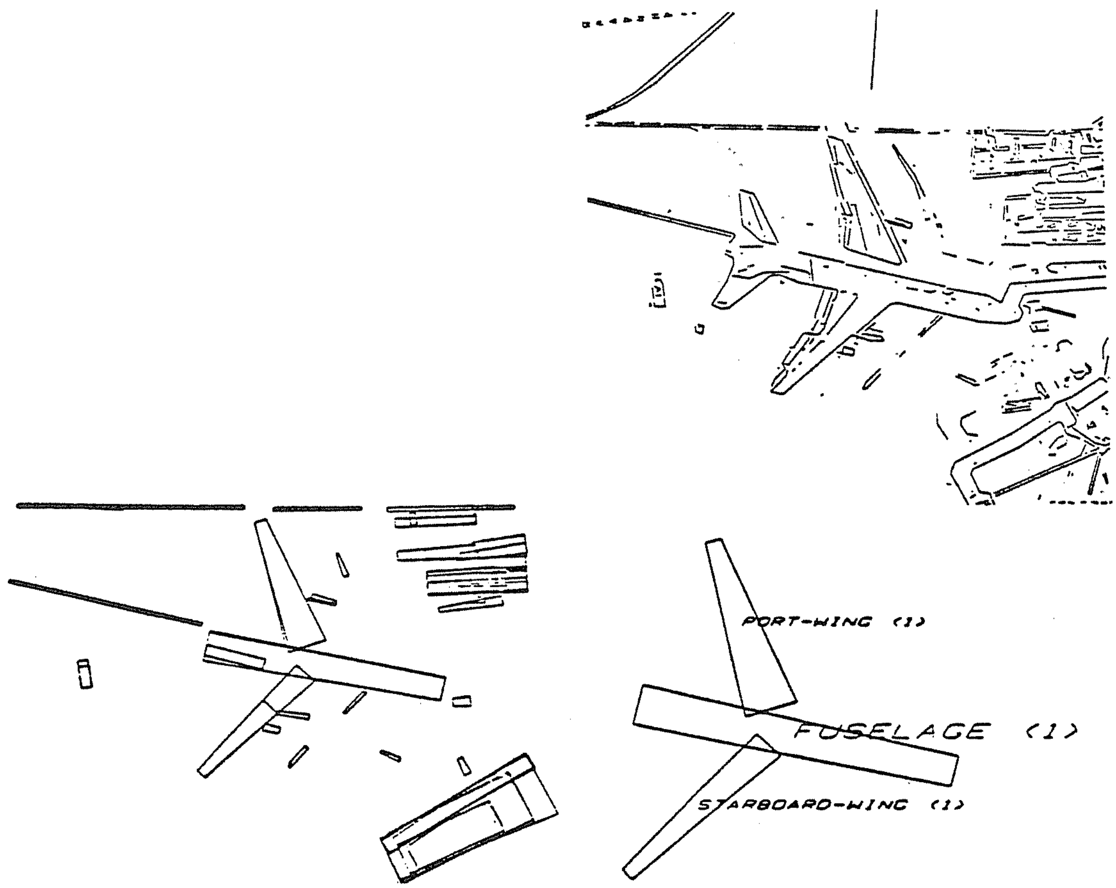


Figure 21. Results of Brooks' matching procedure. The first figure shows the output of the edge detector, the second figure shows the output of the ribbon finder. The final figure is the output of the matcher.

accomplished by grouping edges which lie in the same plane, partitioning each such set of points into line segments and arcs of circles, and associating properties with each line or arc based on the relations between the surfaces which meet to form the given segment. Second, objects are hypothesized by determining if a pair of observed segments are consistent with a given model's features.

Silberberg *et al.* [Sil83] use a generalized Hough transform to match a set of observed line segments with model lines for each viewpoint. A 3-D Hough space is used to represent a viewpoint (two dimensions for position on the view-sphere, one dimension for orientation at a viewpoint). For each viewpoint and pair of line segments, one from a model and one from the image, the model line is projected onto the image plane, incrementing the corresponding bin in Hough space if the pair of lines match. This procedure is first used with a coarsely quantized Hough space to select those viewpoints which correspond to peaks in Hough space after the voting procedure are successively refined to provide a finer resolution estimate of the exact viewing position.

Bhanu [Bha82] uses a relaxation labeling technique for identifying which model face is associated with each observed image face. Range data are first merged into planar faces using the technique in [Hen82a]. A two stage relaxation procedure is then used. In the first stage compatibilities between pairs of adjacent faces are used; in the second stage compatibilities between a face and two of its neighboring faces are used. The compatibility of a face in an unknown view with a face in a model is computed by finding transformations (scale, rotation, translation), applying them and computing feature value mismatches. The initial probabilities for a face are computed as a function of global features of the face, including area, perimeter, number of vertices, and radius.

Chakravarty [Cha82] use a multi-stage matching procedure for a given observed set of lines and junctions. First, all viewing patches having similar boundaries to the given observed image boundary are selected; second, patches which don't contain matching junction types are removed; finally, a projection is computed based on the correlated junctions, and this transformation Is verified with the original image data.

8. Concluding Remarks

An extensive review of robot vision techniques for industrial part recognition has been presented. The major motivation for using industrial machine vision is to increase flexibility and reduce cost. At the present time only very simple techniques based on 2-D global scalar features have been applied in real-time manufacturing processes. More sophisticated techniques will have to be developed in order to adequately deal with less structured industrial environments and permit more task versatility. These techniques will incorporate higher level modeling (e.g. highly organized graph models containing 2 ½-D and 3-D descriptions), more powerful feature extraction methods (e.g. global structural features of object boundaries, surfaces and volumes), and more robust matching procedures for efficiently comparing large sets of complex models with observed image features. A survey of the current state-of-the-art on these research topics has been presented for the task of industrial part recognition.

References

[A.1] Symposium on Computer Vision and Sensor-based Robots, Warren, MI, 1979.

[A.2] 1st Int. Conf. Robot Vision and Sensory Controls, Stratford-upon-Avon, England, 1981.

[A.3] 2nd Int. Conf. Robot Vision and Sensory Controls , Stuttgart, W. Germany, 1982. (SPIE Conf. 392.)

[A.4] 3rd Int. Conf. Robot Vision and Sensory Controls, Cambridge, MA, 1983.

[A.5] SPIE Conf. 155: Image Understanding Systems and Industrial Applications, San Diego, CA, 1978.

[A.6] SPIE Conf. 182: Imaging Applications for Automated Industrial Inspection and Assembly, Washington, D.C., 1979.

[A.7] SPIE Conf. 336: Robot Vision, Arlington, VA, 1982.

[A.8] SPIE Conf. 442: Robotics and Robot Sensing Systems, San Diego, CA, 1983.

[A.9] 11th Workshop on Applied Imagery Pattern Recognition: Machine Vision Systems, College Park, MD, 1982.

[A.10] SPIE Conf. 360: Robotics and Industrial Inspection, San Diego, CA, 1982.

[A.11] Workshop on Industrial Applications of Machine Vision, Research Triangle Park, NC, 1982.

[A.12] Applied Machine Vision Conference, Cleveland, OH, 1982.

[A.13] Applied Machine Vision Conference, Memphis, TN, 1983.

[A.14] Applications of Machine Vision to Robotics and Automated Inspection, Madison, WI, 1983.

[B.1] A. Pugh, ed., *Robot Vision*, Springer-Verlag, New York, 1983.

[B.2] I. Aleksander, ed., *Artificial Vision for Robots*, Kogan Page, London, 1983.

[B.3] G.G. Dodd and L. Rossol, eds., *Computer Vision and Sensor-based Robots*, Plenum, New York, 1979.

[Agi72] G.J. Agin, Representation and description of curved objects, Ph.D. Thesis (Technical Report AIM-173), Computer Science Department, Stanford University, Stanford, CA, 1972.

[Agi76] G. J. Agin and T.O. Binford, Computer description of curved objects, *IEEE Trans. Computers 25*, 1976, 439-449.

[Agi77] G.J. Agin and R.O. Duda, SRI vision research for advanced industrial automation, *Proc. 2nd U.S.A./Japan Computer Conf.* 1977, 113-117.

[Agi80] G.J.Agin, Computer vision systems for industrial inspection and assembly, *Computer*, May 1980, 11-20.

[Agi82] G.J. Agin and P. T. Highnam, A movable light-stripe sensor for obtaining three-dimensional coordinate measurements, *Proc. SPIE 360*, 1982.

[Ale83] I. Aleksander, T.J. Stonham, and B.A. Wilkie, Computer vision systems for industry: comparisons, in *Artificial Vision for Robots*, I. Aleksander, ed., Kogan Page, UK, 1983.

[Ale82] T. Alexander, Practical uses for a useless sciences, *Fortune*, May 31, 1982.

[Alt81] M.D. Altschuler, J.L. Posdamer, G. Frieder, B.R. Altschuler, and J. Taboada, The numerical stereo camera, *Proc. SPIE 283*, 1981, 15-24.

[Aya82] N.J. Ayache and O.D. Faugeras, Recognition of partially visible planar shapes, *Proc. 6th Int. Conf. Pattern Recognition*, 1982.

[Aya83] N.J. Ayache, A model-based vision system to identify and locate partially visible industrial parts, *Proc. Computer Vision and Pattern Recognition*, 1983, 492-494.

[Bad78] N. Badler and R. Bajcsy, Three-dimensional representations for computer graphics and computer vision, *ACM Comp. Graphics 12*, 1978, 153-160.

[Bai77] M.L. Baird, Image segmentation technique for locating automotive parts on belt conveyors, *Proc. 5th Int. Joint Conf. Artificial Intelligence*, 1977, 694-695.

[Bai78] M.L. Baird, Sight I: A computer vision system for automated IC chip manufacture, *IEEE Trans. Syst. Man Cybern. 8*, 1978, 133-139.

[Bal81a] D.H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition 13*, 1981, 111-122.

[Bal81b] D.H. Ballard, Parameter networks: towards a theory of low level vision, *Proc. 7th Int. Joint Conf. Artificial Intelligence*, 1981, 1068-1078.

[Bal82] D.H. Ballard and C. M. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[Bar78] H.G. Barrow and J.M. Tenenbaum, Recovering intrinsic scene characteristics from images, in *Computer Vision Systems*, A. R. Hanson and E.M. Riseman, eds., Academic Press, New York, 1978.

[Bar81] H.G. Barrow and J.M. Tenenbaum, Computational vision, *Proc. IEEE 69*, 1981, 572-595.

[Bau72] B. G. Baumgart, Winged edge polyhedron representation, Stanford A.I. Memo AIM-179, Stanford University, Stanford, Ca., 1972.

[Bau82] R.D. Baumann and D.A. Wilmshurst, Vision system sorts castings at General Motors Canada, *Sensor Review 2*, 1982.

[Bha81] B. Bhanu, Shape matching and image segmentation using stochastic labeling, USCIPI Report 1030, USC, Aug. 1981.

[Bha82] B. Bhanu, Surface representation and shape matching of 3-D objects, *Proc. IEEE Conf. Pattern Recognition and Image Processing*, 1982, 349-354.

[Bha83] B. Bhanu, Recognition of occluded objects, *Proc. 8th Intl. Joint Conf. Artificial Intelligence*, 1983.

[Bin71] T.O. Binford, Visual perception by computer, IEEE Systems Science and Cybernetics Conference, Miami, FL, Dec. 1971.

[Bin82] T.O. Binford, Survey of model-based image analysis systems, *Int. J. Robotics Research 1*, 1982, 18-63.

[Bir79] J.R. Birk, R.B. Kelly, N.-Y.Chen, and L.Wilson, Image feature extraction using diameter-limited gradient direction histograms, *IEEE Trans. Pattern Anal. Machine Intell. 1*, 1979, 258-235.

[Bir81] J.R. Birk, R.B. Kelly, and H. Martins, An orienting robot for feeding workpieces stored in bins, *IEEE Trans. Syst., Man, Cybern. 11*, 1981, 151-160.

[Bol79a] R.C. Bolles, Symmetry analysis of two-dimensional patterns for computer vision, *Proc. Int. Joint Conf. Art. Int.*, 1979, 70-72.

[Bol79b] R.C. Bolles, Robust feature matching through maximal cliques, *Proc. SPIE 182*, 1979, 140-149.

[Bol81] R.C. Bolles, Overview of applications of image understanding to industrial automation, *Proc. SPIE 281*, 1981, 134-140.

[Bol81b] R.C. Bolles and M.A. Fischler, A RANSAC-based approach to model fitting and its application to finding cylinders in range data, *Proc. 7th Int. Joint Conf. Artificial Intelligence*, 1981, 637-643.

[Bol82] R.C. Bolles and R.A. Cain, Recognizing and locating partially visible objects: the local-feature-focus method, *Intl. Journal Robotics Research 1*, 1982, 57-82.

[Bol83] R. C. Bolles, P. Horaud, and M. J. Hannah, 3DPO: a three-dimensional part orientation system, *Proc. 8th Int. Joint Conf. Artificial Intelligence*, 1983.

[Bra82] M. Brady, Computational approaches to image understanding, *ACM Computing Surveys 14*, 1982, 3-71.

[Bru83] W. Brune and K.H. Bitter, S.A.M. Opto-electronic picture sensor in a flexible manufacturing system, in *Robot Vision*, A. Pugh, ed., Springer-Verlag, New York, 1983.

[Bro79] R.A. Brooks, Goal-directed edge linking and ribbon finding, *Proc. ARPA Image Understanding Workshop*, April 1979, 72-76.

[Bro81] R.A. Brooks and T.O. Binford, Geometric modeling in vision for manufacturing, *Proc. SPIE 281*, 1981, 141-159.

[Bro83a] R. A. Brooks, Symbolic reasoning among 3-D models and 2-D images, *Artificial Intelligence 17*, 1983, 285-348.

[Bro83b] R. A. Brooks, Model-based three-dimensional interpretations of two-dimensional images, *IEEE Trans. Pattern Analysis and Machine Intelligence 5*, 1983, 140-150.

[Cas83] R.J. Casler, Vision-guided robot part acquisition for assembly packaging applications, SME technical paper, MS83-219, 1983.

[Cha79] I. Chakravarty, A generalized line and junction labeling scheme with applications to scene analysis, *IEEE Trans. Pattern Analysis and Machine Intelligence 1*, 1979, 202-205.

[Cha82] I. Chakravarty and H. Freeman, Characteristic views as a basis for three-dimensional object recognition, *Proc. SPIE 336*, 1982, 37-45.

[Che81] J.K. Cheng and T.S. Huang, Image recognition by matching relational structure, *Proc. IEEE Conf. Pattern Recognition and Image Processing*, 1981.

[Che82] J.K. Cheng and T.S. Huang, Recognition of curvilinear objects by matching relational structure, *Proc. IEEE Conf. Pattern Recognition and Image Processing*, 1982.

[Che80] N.-Y. Chen, J.R. Birk, and R.B. Kelly, Estimating workpiece pose using the feature points method, *IEEE Trans. Automatic Control 25*, 1980, 1027-1041.

[Che82] M.J. Chen and D.L. Milgram, A development system for machine vision, *Proc. IEEE Pattern Recognition and Image Processing*, 1982.

[Chi82a] R.T. Chin, Machine vision for discrete part handling in industry: a survey, *Proc. Int. Workshop on Industrial Applications of Machine Vision*, 1982.

[Chi82b] R.T. Chin and C.A. Harlow, Automated visual inspection: a survey, *IEEE Trans. Pattern Analysis and Machine Intelligence 4*, 1982, 557-573.

[Cro82] A.J. Cronshaw, Automatic chocolate decoration by robot vision, *Proc. 12th Int. Symp. Industrial Robots*, 1982.

[Des78a] J.D. Dessimoz, et al. Recognition and handling of overlapping industrial parts, *Proc. Int. Sym. Computer Vision and Sensor Based Robots*, GM Research Lab., Warren, MI, Oct. 1978.

[Des78b] J.D. Dessimoz, Visual identification and location in a multi-object environment by contour tracking and curvature description, *Proc. 8th Int. Sym. Industrial Robots*, 1978, 746-776.

[Dud73] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[Dud79] R.O. Duda, D. Nitzan and P. Barret, Use of range and reflectance data to find planar surface regions, *IEEE Trans. Pattern Analysis and Machine Intelligence 1*, 1979, 259-271.

[Foi81] J.P. Foith, C. Eisenbarth, E. Enderls, H. Geisselmann, H. Ringshauser, and G. Zimmermann, Real-time processing of binary images for industrial applications, in *Digital Image Processing Systems*, L. Bolc and Z. Kulpa, eds., Springer-Verlag, Berlin, 1981, 61-168.

[Fu83] K.S. Fu, Robot vision for machine part recognition, *Proc. SPIE 442*, 1983.

[Gle79] G.J. Gleason and G.J. Agin, A modular system for sensor-controlled manipulation and inspection, *Proc. 9th Int. Symp. Industrial Robots*, 1979, 57-70.

[Goa83] C. Goad, Special purpose automatic programming for 3D model-based vision, *Proc. Image Understanding Workshop*, 1983, 94-104.

[Hat82] W. Hattich, Recognition of overlapping workpieces by model directed construction of object contours, *Digital Systems for Industrial Automation 1*, 1982, 223-239.

[Hen82a] T.C. Henderson and B. Bhanu, Three-point seed method for the extraction of planar faces from range data, *Proc. 1982 Workshop on Industrial Applications of Machine Vision*, 1982, 181-186.

[Hen82b] T.C. Henderson, Efficient segmentation method for range data, *Proc. SPIE 336*, 1982, 46-47.

[Hol79] S.W. Holland, L. Rossol, and M.R. Ward, CONSIGHT-I: A vision-controlled robot system for transferring parts from belt conveyors, in *Computer Vision and Senor-based Robots*, (G.G. Dodd and L. Rossol, eds.,) Plenum, New York, 1979.

[Hor75a] B.K.P. Horn, A problem in computer vision: Orienting silicon integrated circuit chips for lead bonding, *Computer Graphics and Image Processing 4*, 1975, 294-303.

[Hor75b] B.K.P. Horn, Obtaining shape from shading information, in *The Psychology of Computer Vision*, P.H. Winston, ed., McGraw-Hill, New York, 1975, 115-155.

[Hor79] B.K.P. Horn, SEQUINS and QUILLS – representations for surface topography, in *Representations for 3-Dimensional Objects*, R. Bajcsy, ed., Springer-Verlag, New York, 1982.

[Hsi79] Y.Y. Hsieh and K.S. Fu, A method for automatic IC chip alignment and wire bonding, *Proc. IEEE Conf. Pattern Recognition and Image Processing*, 1979, 101-108.

[Iga79] K. Igarashi, M. Naruse, S. Miyazaki, and T. Yamada, Fully automated integrated circuit wire bonding system, *Proc. 9th Int. Symp. Exposition on Industrial Robots*, 1979, 87-97.

[Ike81a] K. Ikeuchi, Determining surface orientations of specular surfaces by using the photometric stereo method, *IEEE Trans. Pattern Analysis and Machine Intelligence 3*, 1981, 661-669.

[Ike81b] K. Ikeuchi, Recognition of 3-D object using Extended Gaussian Image, *Proc. 7th Int. Joint. Conf. Artificial Intelligence*, 1981, 595-600.

[Ike82] K. Ikeuchi and Y. Shirai, A model based vision system for recognition of machine parts, *Proc. AAAI*, 1982, 18-21.

[Ike83a] K. Ikeuchi, Determining attitude of object from needle map using extended gaussian image, MIT A.I. Memo No. 714, April 1983.

[Ike83b] K. Ikeuchi, B.K.P. Horn, S. Nagata, T. Callahan and O. Feingold, Picking up an object from a pile of objects, MIT A.I. Memo No. 718, May 1983.

[Jak77] R. Jakubowski and A. Kasprzak, A Syntactic description and recognition of rotary machine elements, *IEEE Trans. Computers 26*, 1977, 1039-1042.

[Jak82] R. Jakubowski, Syntactic characterization of machine parts shapes, *Cybernetics and Systems 13*, 1982, 1-24.

[Jar80] J.F. Jarvis, Visual inspection automation, *Computer 13*, no. 5, May 1980, 32-39.

[Jar83a] J. F. Jarvis, A perspective on range finding techniques, *IEEE Trans. Pattern Analysis and Machine Intelligence 5*, 1983, 122-139.

[Jar83b] J. F. Jarvis, A laser time-of-flight range scanner for robotic vision, *IEEE Trans. Pattern Analysis and Machine Intelligence 5*, 1983, 505-512.

[Kan81] T. Kanade and H. Asada, Noncontact visual three-dimensional ranging devices, *Proc. SPIE 283*, 1981, 48-53.

[Kas76] S. Kashioka, M. Ejiri, and Y. Sakamoto, A transistor wire- bonding system utilizing multiple local pattern matching techniques, *IEEE Trans. Systems, Man, and Cybernetics 6*, 1976, 562-569.

[Kas77] S. Kashioka, S. Takeda, Y. Shima, T. Uno, and T. Hamada, An approach to the integrated intelligent robot with multiple sensory feedback: visual recognition techniques, *Proc. 7th Int. Symp. Industrial Robots*, 1977, 531-538.

[Kaw79] S. Kawata and Y. Hirata, Automatic IC wire bonding system with TV cameras, *Proc. SME Assembly VI Conf.*, 1979.

[Kel82] R.B. Kelly, J.R. Birk, H.A.S. Martins, and R. Tella, A robot system which acquires cylindrical workpieces from bins, *IEEE Trans. Systems, Man, and Cybernetics 12*, 1982, 201-213.

[Kel83a] R.B. Kelly, H.A.S. Martins, J.R. Birk, and J-D. Dessimoz, Three vision algorithms for acquiring workpieces from bins, *Proc. IEEE 71*, 1983, 803-820.

[Kel83b] R.B. Kelly, Binary and gray scale robot vision, *Proc. SPIE 442*, 1983.

[Ken80] J.R. Kender, Shape from texture, Ph.D. Thesis, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA, 1980.

[Kin81] P. Kinnucan, How smart robots are becoming smarter, *High Technology*, September/October 1981.

[Kin83] P. Kinnucan, Machines that see, *High Technology*, April 1983, 30-36.

[Kit83] P.W. Kitchin and A. Pugh, Processing of binary images, in *Robot Vision*, A. Pugh, ed., Springer-Verlag, New York, 1983, 21-42.

[Koe76a] J. J. Koenderink and A. J. vanDoorn, The singularities of the visual mapping, *Biol. Cybernetics 24*, 1976, 51-59.

[Koe76b] J. J. Koenderink and A. J. vanDoorn, Visual perception of rigidity of solid shape, *Journal of Mathematical Biology 3*, 1976, 79-85.

[Koe79] J. J. Koenderink and A. J. vanDoorn, The internal representation of solid shape with respect to vision, *Biol. Cybernetics 32*, 1979, 211-216.

[Kru81] R.P. Kruger and W.B. Thompson, A technical and economic assessment of computer vision for industrial inspection and robotic assembly, *Proc. IEEE 69*, 1981, 1524-1538.

[Lie79] L. Lieberman, Model-driven vision for industrial automation, in *Advances in Digital Image Processing*, P. Stucki, ed., Plenum , New York, 1979.

[Mar78] D. Marr, Representing visual information, in *Computer Vision Systems*, A.R. Hanson and E.M. Riseman, eds., Academic Press, New York, 1978.

[Mar82a] D.H. Marimont, Segmentation in Acronym, *Proc. Image Understanding Workshop*, Sept. 1982.

[Mar82b] D. Marr, *Vision*, Freeman, San Francisco, 1982.

[Mes77] M. Mese, I. Yamazaki, and T. Hamada, An automatic position recognition technique for LSI assembly, *Proc. 5th Int. Joint Conf. Artificial Intelligence*, 1977, 685-693.

[Mil80] D. L. Milgram and C.M. Bjorklund, Range image processing: planar surface extraction, *Proc. 5th Int. Joint Conf. Pattern Recognition*, 1980, 912-919.

[Mye80] W. Myers, Industry begins to use visual pattern recognition, *Computer*, May 1980, 21-31.

[Nev77] R. Nevatia and T.O. Binford, Description and recognition of curved objects, *Artificial Intelligence 8*, 1977, 77-98.

[Osh79] M. Oshima and Y. Shirai, A scene description method using three-dimensional information, *Pattern Recognition 11*, 1979, 9-17.

[Osh83] M. Oshima and Y. Shirai, Object recognition using three-dimensional information, *IEEE Trans. Pattern Analysis and Machine Intelligence 5*, 1983, 353-361.

[Per77] E. Persoon and K.S. Fu, Shape discrimination using Fourier descriptors, *IEEE Trans. Systems, Man and Cybernetics 7*, 1977, 170-179.

[Per78] W.A. Perkins, A model-based vision system for industrial parts, *IEEE Trans. Computers 27*, 1978, 126-143.

[Per80] W.A. Perkins, Area segmentation of images using edge points, *IEEE Trans. Pattern Analysis and Machine Intelligence 2*, 1980, 8-15.

[Pip83] F. J. Pipitone and T. G. Marshall, A wide-field scanning triangulation rangefinder for machine vision, *Int. J. Robotics Research 2*, 1983, 39-49.

[Poj82] J.F. Poje and E.J. Delp, A review of techniques for obtaining depth information with applications to machine vision, RSD-TR-2-82, Center for Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, MI, April 1982.

[Pop75] R.J. Popplestone, C.M. Brown, A.P. Ambler, G.F. Crawford, Forming models of plane-and-cylinder faceted bodies from light stripes, *Proc. 4th Int. Joint Conf. Artificial Intelligence*, 1975, 664-668.

[Por80] G.B. Porter III and J.L. Mundy, Visual inspection system design, *Computer 13*, no. 5, 1980, 40-49.

[Pot83a] J. Pot, P. Coiffet, and P. Rives, Comparison of five methods for the recognition of industrial parts, in *Developments in Robotics*, B. Rooks, ed., IFS Publications, UK, 1983.

[Pot83b] M. Potmesil, Generating models of solid objects by matching 3D surface segments, *Proc. 8th Int. Joint Conf. Artificial Intelligence*, 1983.

[Pug82] A. Pugh, Second generation robotics, *Proc. 12th Int. Symp. Industrial Robots*, 1982.

[Ray81] R. Ray, Error analysis of surface normals determined by radiometry, in J. Birk et al., General methods to enable robots with vision to acquire, orient and transport workpieces, 7th report, Univ. of Rhode Island, 1981.

[Ros79a] A. Rosenfeld and L. S. Davis, Image segmentation and image models, *Proc. IEEE 67*, 1979, 764-772.

[Ros79b] C.A. Rosen, Machine vision and robotics: industrial requirements, in *Computer Vision and Sensor-based Robots*, G.G. Dodd and L. Rossol, eds., Plenum, New York, 1979, 3-20.

[Ros81] C.A. Rosen and G.J. Gleason, Evaluating vision system performance, *Robotics Today*, Fall 1981.

[Ros82] L. Rossol, Computer vision in industry, Meadowbrook Research Conf. Robotics: The Next Decade, Rochester, MI, May 1982.

[Sch83] B.J. Schachter, A matching algorithm for robot vision, *Proc. Computer Vision and Pattern Recognition*, 1983, 490-491.

[Seg83] J. Segen, Locating Randomly Oriented Objects from Partial View, *3rd Int. Conf. Robot Vision and Sensory Controls*, Cambridge, MA, Nov. 1983.

[Shi72] Y. Shirai, Recognition of polyhedrons with a range finder, *Pattern Recognition 4*, 1972, 243-250.

[Shi79] Y. Shirai, Three-dimensional computer vision, in *Computer Vision and Sensor-based Robots*, G.G. Dodd and L. Rossol, eds., Plenum, New York, 1979, 187-205.

[Shi78] Y. Shirai, Recognition of real-world objects using edge cue, in *Computer Vision Systems*, A.R. Hanson and E.M. Riseman, eds., Academic Press, New York, 1978, 353-362.

[Shn79] M. Shneier, A compact relational structure representation, *Proc. Workshop on Representation of Three-Dimensional Objects*, 1979.

[Shn81] M. Shneier, Models and strategies for matching in industrial vision, Computer Science Technical Report TR-1073, University of Maryland, College Park, MD, July 1981.

[Sil80] W.M. Silver, Determining shape and reflectance using multiple images, M.Sc. dissertation, MIT, Cambridge, MA, 1980.

[Sil83] T. M. Silberberg, L. S. Davis, and D. Harwood, An iterative Hough procedure for three-dimensional object recognition, Technical Report CAR-TR-20, Center for Automation Research, University of Maryland, College Park, MD, August 1983.

[Smi79] D.A. Smith, Using enhanced spherical images for object representation, MIT A.I. Memo No. 530, 1979.

[Sto80] G. Stockman, Recognition of parts and their orientation for automatic machining, handling and inspection, Final Rep. NSF-SIBR-Phase I, NTIS Order PB 80-178817, 1980.

[Sto83] G.C. Stockman, Circular scanning for part recognition and orientation, SME technical paper, MS83-214, Feb. 1983.

[Sto82] G. Stockman, K. Kopstein, and S. Benett, Matching images to models for registration and object detection via clustering, *IEEE Trans. Pattern Analysis and Machine Intelligence 4*, 1982, 229-241.

[Sug79] K. Sugihara, Range-data analysis guided by a junction dictionary, *Artificial Intelligence 12*, 1979, 41-69.

[Tak77] K. Takeyasu, M. Kasai, R. Shimomura, T. Goto, and Y. Matsumoto, An approach to the integrated intelligent robot with multiple sensory feedback, *Proc. 7th Int. Symp. Industrial Robots*, 1977, 523-530.

[Ten79] J.M. Tenenbaum, H.G. Barrow, and R.C. Bolles, Prospects for industrial vision, in *Computer Vision and Sensor-based Robots*, G.G. Dodd and L. Rossol, eds., Plenum, New York, 1979, 239-256.

[Tro80] H. Tropf, Analysis-by-synthesis search for semantic segmentation — applied to workpiece recognition, *Proc. 5th Int. Conf. Pattern Recognition*, 1980, 241-244.

[Tro81] H. Tropf, Analysis-by-synthesis search to interpret degraded image data, *Proc. 1st Int. Conf. Robot Vision and Sensory Controls*, 1981, 25-33.

[Tro82] H. Tropf, H. Geisselmann, and J.P. Foith, Some applications of the fast industrial vision system S.A.M., *Proc. Int. Workshop on Industrial Applications of Machine Vision*, 1982.

[Tro82] J.E. Trombly, Recent applications of computer aided vision in inspection and part sorting, SME technical paper MS82-128, Dearborn, MI, 1982.

[Ume79] Y. Umetani and K. Taguchi, Feature properties to discriminate complex shapes, *Proc. 9th Int. Symp. Industrial Robots*, 1979, 367-378.

[Ume82] Y. Umetani and K. Taguchi, Discrimination of general shapes by psychological feature properties, *Digital Systems for Industrial Automation 1*, 1982, 179-196.

[Vam77] T. Vamos, Industrial objects and machine parts recognition, in *Applications of Syntactic Pattern Recognition*, K.S. Fu, ed., Springer-Verlag, New York, 1977.

[Vil83] P. Villers, The role of vision in industrial robotic systems and inspections, *Proc. ELECTRO 83*, New York, 1983.

[Wes82] P.C. West, Overview of machine vision-1982, SME technical paper MS82-184, Dearborn, MI, 1982.

[Wes83] P.C. West, Overview of machine vision, in Notes for: Applications of Machine Vision to Robotics and Automated Inspection, Madison, WI, October 1983.

[Wit81] A.P. Witkin, Recovering surface shape and orientation from texture, in *Computer Vision*, M. Brady, ed., North-Holland, Amsterdam, 1981, 17-47.

[Woo78] R.J. Woodham, Photometric stereo: a reflectance map technique for determining surface orientation from image intensity, *Proc. SPIE 155*, 1978, 136-143.

[Yac80] M. Yachida and S. Tsuji, Industrial computer vision in Japan, *Computer*, May 1980, 50-63.

[Yac77] M. Yachida and S. Tsuji, A versatile machine vision system for complex industrial parts, *IEEE Trans. Computers 26*, 1977, 882-894.

[Zah72] C.T. Zahn and R.Z. Roskies, Fourier descriptors for plane closed curves, *IEEE Trans. Computers 21*, 1972, 269-281.

[Zue83] N. Zuech and R. Ray, Vision guided robotic arm control for part acquisition, *Proc. Control Engineering Conference*, 1983.