A Technical and Economic Assessment of Computer Vision for Industrial Inspection and Robotic Assembly

RICHARD P. KRUGER, SENIOR MEMBER, IEEE, AND WILLIAM B. THOMPSON, MEMBER, IEEE

Invited Paper

Abstract—The use of computer vision to detect, measure, and perhaps guide the assembly of man-made components is potentially a very significant research and development area. The efficacy of these techniques for any given application depends on both technical and economic considerations. This paper will explore both these considerations using appropriate generic examples. It is our goal to first present a concise discussion of the present state of many technical and economic factors and then extrapolate these factors into the future for the purpose of guiding further investigations.

Introduction

THIS paper will be organized in the following manner. First, there will be presentations of the terminology and techniques commonly employed when discussing computer vision. Next, examples of research that exemplify these techniques will be presented. Lastly, there will be a discussion of the future technological and economic trends that must be encouraged if progress in this area of computer vision is to continue or be accelerated. The general source material for this paper includes several surveys [1]-[3], a tutorial [4], and two proceedings dedicated to this topic [5], [6]. Other references will be cited as required. It will, however, be impossible to offer a detailed presentation of all available material. An attempt will be made to integrate the body of knowledge with generic examples and present general recommendations where future efforts should be concentrated.

Despite substantial research efforts, the study of computer vision is still in its infancy. Thus it is not possible to describe an elegant theoretical basis encompassing all of our examples. Significant reductions in complexity are possible if automated perception is limited to an industrial environment. Even here, however, we still lack a clear understanding of the fundamental problems that must be addressed if computer vision is to have a major impact on manufacturing.

The economic motivation for the use of industrial computer vision is to increase productivity through the introduction of intelligent programmable vision-based systems for inspection and/or robotic assembly. Productivity is defined as the output of goods or services produced (or inspected) per unit of labor input. Samuelson [7] quotes Solow, "Scarcely half of the increase in America's productivity per capita and in real wages can be accounted for by the increase in available capital. More than half of the increase in productivity is a residual that seems to be attributable to *technical change*, to scientific advance, to *industrial improvements*, and to improved management and *training of labor*." The key words here are underlined for emphasis. Thus capital formation is often linked to technical innovation, which can produce higher productivity. It is our basic postulate that computer vision as a tool for industrial inspection and assembly will raise both labor and general productivity through a symbiotic use of cybernetics.

An example might be used to clarify this postulate. Let us suppose that a given discrete part requires visual inspection for surface defects and spatial tolerances before being boxed for sale. This task was formerly performed by an unaided quality control inspector. Because of the volume of parts produced, he was required to inspect only 10 percent of the production. Thus 90 percent of the production run was uninspected. A vision-based system was then designed that was fast and reliable enough to inspect 100 percent of the product. The inspector is now used in equivocal situations as indicated by the inspection system. Several things have occurred from an economic/productivity standpoint. Capital formation has made it possible to purchase an automated inspection station. This technical change has increased inspection productivity tenfold and at the same time has increased the utilization of the inspector. The quality of the product available for sale has been increased and with it the reputation of the producer. Further justification of the economic rationale for automation will be discussed later in the paper. We will now present some basic terminology of computer vision and scene analysis.

Computer Vision and Scene Analysis

Computer vision is the process of producing useful symbolic descriptions of a visual environment from image data [8]. The nature of this description and the processes by which it is developed often depend on the uses to which it will be put. Thus computer vision is best understood as the perceptual component of a larger, computational problem-solving system. In the sections that follow, we will describe ways in which automated visual processing may be integrated into two classes of manufacturing processes: the inspection of parts to deter-

Manuscript received July 23, 1981; revised August 10, 1981. This work was supported by the U.S. Department of Energy under Contract W7405-ENG-36.

R. P. Kruger is with the University of California, Los Alamos National Laboratory, Los Alamos, NM 87545.

W. B. Thompson is with the University of Minnesota, Minneapolis, MN, 55455.

0018-9219/81/1200-1524$00.75 © 1981 IEEE

mine if they meet prespecified standards and the identification, manipulation, and assembly of parts using visual feedback.

For most applications, perceptual processes must do much more than produce simple, standard responses to preprogrammed stimuli. Instead, powerful techniques are required that can transform the almost overwhelming amount of data in an image into a compact and useful form in which only information of relevance is represented. This is most often accomplished by attempting to use any available information about the specific task at hand in order to reduce the complexity of the problem. This "knowledge" includes information about sensors and the shape, surface properties, and presentation of all objects possible in the scene. The number and nature of possible objects will almost always be extremely restricted in an industrial vision application. Thus, effective exploitation of these knowledge sources will have a significant impact on the performance of all such systems.

To use the knowledge sources, a model of the possible objects must be developed and represented. This model may be used in a number of ways; clearly, the more that is known about the nature of possible scenes, the easier it will be to perform various interpretation tasks. In fact, meaningful interpretation is impossible without a model. Model information is also important in directing the control of computational steps in a perceptual process. The order of operations significantly affects the ways in which knowledge sources can be used. The best control structures will effectively integrate expectations about the scene represented in the model with the results of previous computations on the image.

The complexity of most preceptual tasks requires that the problem be decomposed into manageable subunits. Thus major design decisions include the function of each module, the computational techniques and data representations imbedded in each module, and the control structures that relate modules and transfer information between them. Most computer vision systems use a hierarchical organization. "Lower" levels are designed to determine simple structural characteristics of objects using detailed image properties. "Higher" levels use this structural information to infer more complete interpretations of the scene.

The knowledge base used by lower level modules tends to consist of general information about surfaces and/or boundaries. This knowledge is often incorporated implicitly. For example, one common approach to locating object boundaries in a scene is to search for areas of rapid change in intensity in the image. This is easily done using some form of difference operator. This approach exploits the fact that discontinuities in luminance are many times caused by discontinuities in surface orientation or surface reflectance properties. Both of these effects commonly occur at object boundaries. Examples of more sophisticated lower level processing may be found in the literature [9], [10].

Higher level processes require much more specific, goal-directed knowledge. The information used is not about simple surface but about the nature of and relationships between whole objects. The set of possible objects that can be present and an abstract description of their appearance in terms of lower level primitives must be developed. The nature of this knowledge depends heavily on the particular application involved. Inspection tasks require little or no information needed for object identification—only one class of objects of known type is likely to be inspected at one time. Emphasis must be placed on high tolerance descriptions that can detect objects which depart from the desired standard. A manipulator controller designed to pick up parts off a conveyor needs to be able to determine the identity, orientation, and position of parts, but need know little of their precise shape, except perhaps at the pickup point. A visual controller for an arc welder may need to know nothing at all about the general appearance of the object it is operating on but may require a great deal of information about the visual properties of the seam between two pieces of metal.

While this hierarchical organization imposes a linear ordering on computational modules, important control structure specifications must still be determined. Modules usually operate by searching for some set of features (e.g., edges, surface, objects, etc.) in either the image or some previously computed abstraction of the image. The way in which this search is directed has a significant effect on the performance of the system. If the search is performed over the whole of the input representation or if it is constrained only by the results of computations at lower levels in the hierarchy, then the control flow is said to *bottom-up*. If the search is dependent on results of computations at higher levels, then a *top-down* control flow is involved. Because top-down controlled processes depend on the availability of higher level knowledge to guide lower level search, they are often described as *model-driven*.

Perhaps the most common organization for computer vision involves a bottom-up control flow and a hierarchy with two major components. The lower level units perform a *segmentation* operation whereby an image is partitioned into regions corresponding to object surfaces. The higher level units use the results of this segmentation to develop an *interpretation* of the scene. Objects are identified from their surface descriptions and any application specific descriptions are produced. On the other hand, some inspection systems fit the model of purely top-down control. If parts are accurately positioned and oriented, then inspection consists of an organized search for areas where the image of the part differs from what is expected. This is often best done by directing the search using specific, high-level models of the part.

In practice, most successful systems incorporate aspects of both bottom-up and top-down controls. As an example, primarily bottom-up processing may be used to locate prominent features of a part to determine its position. Then, top-down control can direct a search to determine if the part satisfies an inspection criterium [11]. Bottom-up control seems most appropriate at lower levels of analysis where the application of relatively simple feature detectors over the whole image can yield much useful information. This allows a representation of the image in terms of properties such as edge location, color, and texture, rather than just in terms of raw luminance values. Top-down control can be most effectively used at higher levels where more goal-directed analysis is required. This best mix of these two control strategies is still a matter of debate and is very much application dependent.

Industrial inspection and assembly problems are well suited to model-based analysis. The objects are man-made and manufactured from well-defined geometric descriptions. (CAD) computer-aided design/computer-aid manufacturing (CAM) technology allows the specification of objects using either volumetric or surface base models, providing compact descriptions that are independent of viewpoint and appropriate for recognition, inspection, and possible manipulation. Volumes can be characterized by attributes such as location, size, diameter, principal axis, and orientation. Surface representations can be described in terms of surface boundaries, shape, and local properties such as orientation and curvature. These object models are appropriate both for high-level tasks such as recognition and for assisting in the characterization and interpretation of lower level features such as local luminance variability. The models can be used in their original, precise geometric formulation, or abstracted into more symbolic terms.

These geometrically based models lend themselves to a particular control strategy sometimes called a hypothesis-verify paradigm. Low-level primitive features are detected from the scene. These features are matched to possible objects. Next, additional features are computed based on those suggested by the object model. This model-guided process eventually verifies object type and orientation. The Concurve system [12], to be presented, represents this analysis approach.

Objects may also be represented structurally by graphs. Recognition of the object becomes a graph-matching process. Often, only part of an object is visible at any one time. This corresponds to a subgraph, which must be isomorphically mapped into various object graphs. This mapping process comprises the recognition process. Performing the match using general tree search is combinatorially exhaustive and, furthermore, the mapping process may be 1 to n. When the attributes are matched by Boolean logic, the tree search reduces to one of finding maximal cliques of the subgraph. Nodes of the subgraph represent possible pairings of scene nodes with the stored object model graph. The links between nodes indicate compatible node pairs. An accumulation of such linkages is called a clique. The maximum such clique is chosen as the probable object in the field of view.

All tree-searching techniques, including the method of maximal cliques, still exhibit the potential for exponential growth, which can lead to impractical computational times. This difficulty may sometimes be minimized using a hierarchical approach. For example, Barrow and Tennebaum [4] suggest that an object model can often be decomposed into self-contained components. These components are then matched independently. The results are then combined and checked afterwards. In this case, complexity of the matching process increases additively. They also suggested that object models be used that contain common components. This will further limit the matching process. Such considerations seem reasonable and should bear directly upon aspects of future part design when using CAD/CAM technology. Intelligent part design can then lead to more practical/efficient automated inspection and assembly.

The examples to follow are representative of three related application areas: 1) visual inspection, 2) materials handling, and 3) visually assisted automatic parts assembly. Each may be implemented as separate operations, though the best results usually come from a more integrated approach. Inspection processes may check for defects due to previous manufacturing steps or verify that an automated assembly operation proceeds correctly. Materials handling often require simple pick and place functions that can be greatly aided by visual feedback. Finally, visually assisted robotic assembly operations require complex and sophisticated hand-eye coordination to be effective.

The sections that follow will describe examples of systems designed to perform in one of these three application areas. The list is not intended to be exhaustive, but rather to illustrate contemporary practices in industrial computer vision. Initial applications to be described will involve automatic scene segmentation and measuration for the express purpose of visual inspection and/or quality assessment. A second group of applications will use segmentation and mensuration as precursors to automated manipulation and assembly. A final section will describe the current status of intelligent programmable manipulators for automated parts assembly. It is the nature of this work to describe these diverse efforts, conceptually unify them when possible, and finally extrapolate these initial disjointed efforts into a future where they may become commonplace.

Automated Visual Inspection Systems

Current practices in automated visual inspection are illustrated by examining several quality control systems for printed circuit boards and integrated circuits. Though the details of the techniques vary widely, they have much in common. First, they all involve the inspection of modern electronics assemblies. Secondly, several of the research efforts have led to production systems. Thirdly, the inspection algorithms, although seemingly diverse, use common *a priori* knowledge. This *a priori* knowledge concerns the stratification of the various scene models possible because man-made objects are under inspection. This stratification produces much regular geometric structure, such as straight lines and right angles, which greatly reduces the possible universe of "normal" structures.

Jarvis [13], [14] has devised a bottom-up two-stage stochastic scene analysis technique for inspection of a mass-produced printed circuit board (PCB). The board image can be represented by a binary array of 0's and 1's. The 1's represent conductor, and the 0's represent substrate. Briefly, the inspection method consists of the preselection of a list of local 5-by-5 pixel template regions that describe the universe of "correct" or normal bit patterns observable on an errorless PCB. The binary templates as well as the binary PCB images are obtained from an array of solid state charge-coupled devices (CCD) packaged in a camera similar to a vidicon. It was found in the present example that 99 percent of the possible "normal" 5-by-5 template patterns could be enumerated by storing only 170 such binary template patterns. When the PCB is inspected, any 5-by-5 region that does not exactly match one of the prestored templates is marked as suspicious. Because 170 prestored templates constitute only 99 percent of the universe of all correct patterns, supplemental tests are performed on those 5-by-5 PCB regions deemed suspicious. These supplemental tests include, among others, the computation of conductor area, the length of the conductor-substrate boundary, and the ratio of area to length within each suspicious 5-by-5 region. Limits on each of these features for a normal range are prestored. Fig. 1(a) is an example output from a small section of the PCB. In this figure pixels labeled 0 correspond to interior conductor cells. Those labeled 1 are horizontal or vertical boundary cells. Those labeled 2, 3, or 4 were candidates for pattern-matching as described earlier. If the cell was labeled 2, it was successfully matched to a normal pattern. If it was labeled 3 or 4, it was unmatched or failed to pass as a normal pattern, respectively. The approach used in this example used a rapid search technique on binary data to eliminate the definite normal regions. Only a relatively few (approximately 2 percent) of the regions were subjected to the second phase process. This quality screening application was possible because of the extreme geometric regularity present in the data.

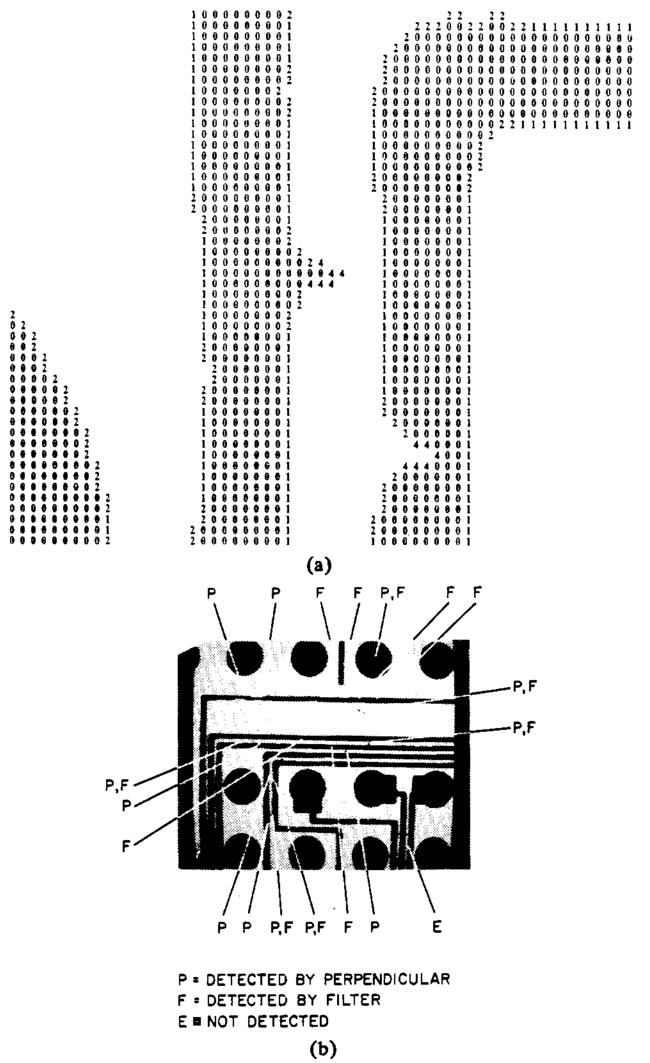


Fig. 1. (a) Example of computer output on a small section of a PCB. (b) PCB board errors as detected by Pavlidis and Krakauer.

Also, the use of a binary data representation was very advantageous. Binary data is compact, minimizing the processor overhead required to deal with the large amount of data in an image. Furthermore, processing may be done with simple logical operations rather than more costly arithmetic operations.

Pavlidis and Krakauer [15] have used data provided by Jarvis to detect flaws in these same printed circuit boards. Their technique involves the ingenious use of binary template matching using a limited number of well-chosen templates accessed via a rapid table lookup technique and an equally rapid paging algorithm designed to sequentially process numerous small portions of the entire board. The limited number of templates necessary was made possible by the *a priori* knowledge that the universe of correct boards has conductor boundaries that are both parallel to each other and exhibit constant thickness. Fig. 1(b) shows a representative output from this algorithm, with *P* and *F* indicating detected errors from various template analyses. The *E*'s denote errors detected by the method.

Chin [16] and Harlow [17] approach automated PCB inspection somewhat differently. They initially use an operator-interactive model-building graph procedure to train the inspection system. Using an interactive camera scanner/display system, the binary image edges of a prototype PCB board are

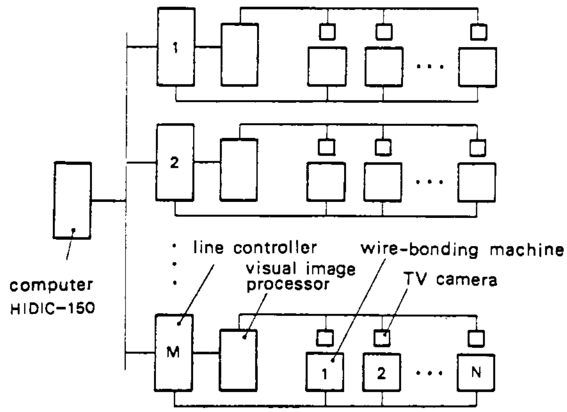


Fig. 2. Time-shared computer system for wire bonding.

detected, smoothed to reduce noise, and encoded into a compact data structure. This encoded form is stored in the computer memory. Interaction is used when necessary to form the graph model and to correct any errors in the edge data structure. In the later inspection phase, the prestored edges and the graph model are used to detect flaws in the test images. The inspection routine accesses the stored model for expected edge coordinates and directs the scanner to look for similarly located edge points in the test image. If the edge is missing or is measurably different, the model indicates an error. This example, like the previous one, attempts to reduce the scene to a binary representation as an initial step. At this juncture the similarity ends. Operator interaction to produce a global edge structure and a graph inspection model replace the noninteractive two-phase stochastic approach of the initial example.

Kashioka [18] has developed a multiplexed multiple station transistor wire-bonding system also using template-matching techniques. In semiconductor integrated circuit (IC) assembly, wires are stretched and bonded between leads on a semiconductor chip by visual detection of the appropriate base and emitter leads. The visual inspection was automated via the multiplexed computer architecture shown in Fig. 2. The system can support up to 50 bonding machines in a time-shared mode from a central computer. Each wire-bonding machine, as shown in Fig. 3, is equipped with a microscope and vidicon camera. This is used to replace the human operator. The principal goal of image processing is to locate the base and emitter regions on each chip and then instruct an x-y servomechanism in each bonding machine to stretch and bond transistor leads between these two detected features. All local 12-by-12 pixel binary regions or templates are shown in Fig. 4. The entire chip is digitized into a 160-by-120 pixel binary field. This field is sequentially examined rapidly by the binary template matching or correlation process until the base and emitter regions have been identified. As in previous applications the use of binary templates and image fields has dramatically decreased the inspection time. In this example, the binary correlator or template matching process was carried out in hardware using a simple exclusive OR circuit. Successful wire bonding has been achieved in over 99 percent of chips entered into the system. This is a far higher success-to-failure ratio than under previous manual techniques. As in previous applications, the computer system is initially trained by man-machine interactive selection of the universe of templates. Thus a flexible man-machine symbiosis is initially used to train

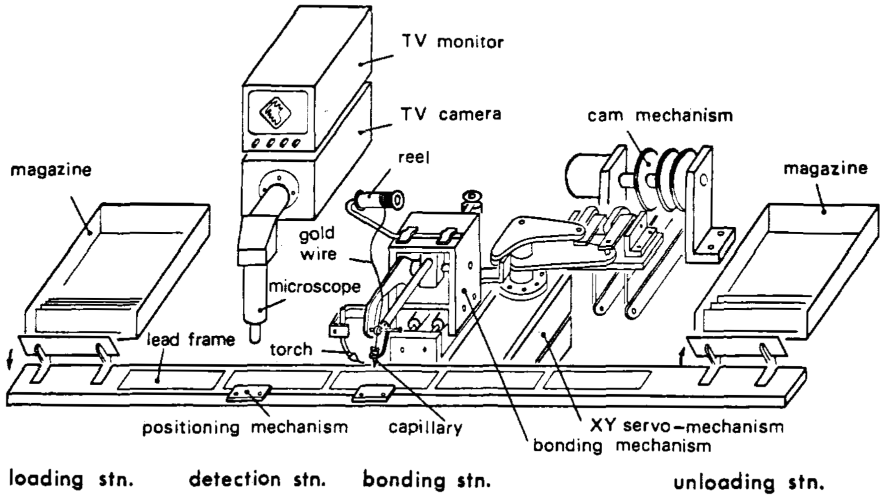


Fig. 3. Wire-bonding machine.

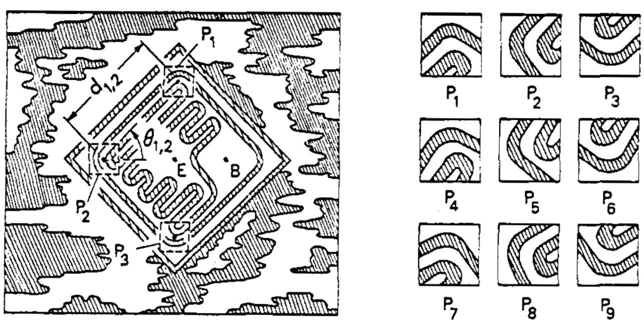


Fig. 4. Binary templates used for pattern matching with an example of use.

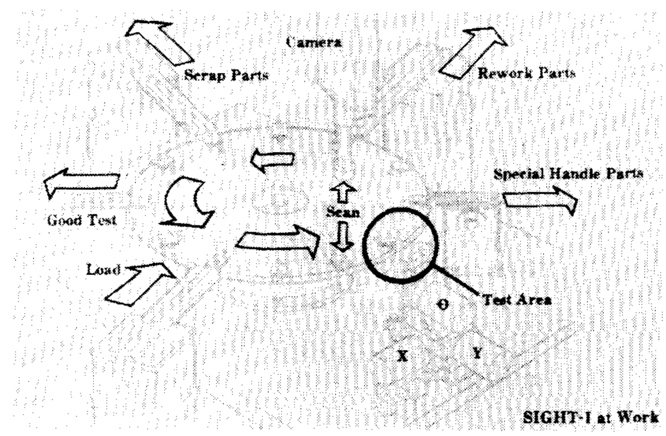


Fig. 5. Sight-1 system.

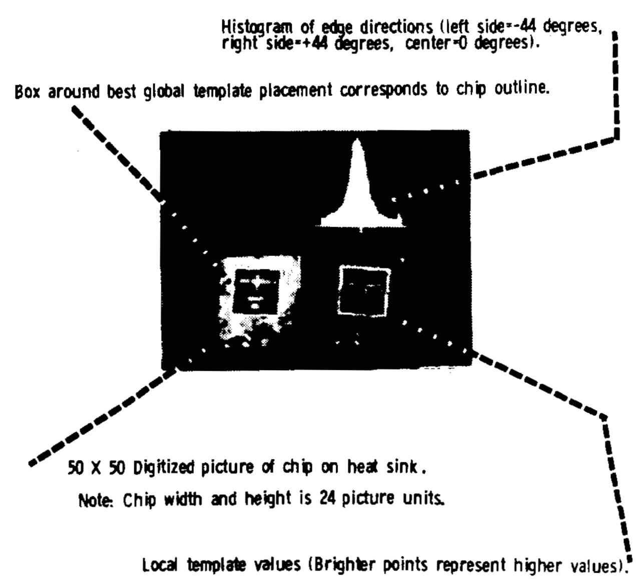


Fig. 6. Gradient histogram to determine chip orientation.

the machine for subsequent repetitive automatic operation. It should be noted that this system does not make a quality control assessment of the substrate prior to transistor bonding. The next example performs such a quality decision.

Baird [11] has developed an inspection system shown in Fig. 5 for automated visual inspection of a power transistor-pair IC chip that is a component in the ignition system of all General Motors automobiles. As in the last example, the human inspector is replaced by a camera interfaced to a minicomputer. Inspection process consists of 1) detection of the IC location and orientation on the heat-sink substrate, and 2) quality control assessment of the chip after acquisition. The inspection field in this example consists of a 50-by-50 pixel region digitized to 16 gray levels (4 bits). A gradient edge detector is used to compile the histogram of all edge directions in the inspection field as shown in Fig. 6. The peak of this histogram indicates the approximate orientation of the chip. As in a previous example, the geometric parallel-perpendicular nature of the edges in this man-made object has served to stratify and thus simplify the detection paradigm. Rotation greater than $\pm 30^\circ$ causes rejection of the IC since automatic function-test equipment cannot successfully attach to IC's rotated beyond this angular range. The next task is to locate the corners of the IC. This was accomplished by a template-matching process not dissimilar to the previous example. However, unlike the previous example the image field and the templates were not binary. If one or more corners are not found, the chip is defective and the part is rejected. Cracked, fractured, chips are eliminated by a simple contrast thresholding operation.

Recent work by Hsieh and Fu [19], [20] proposes an integrated system for automated visual inspection and wire bond-

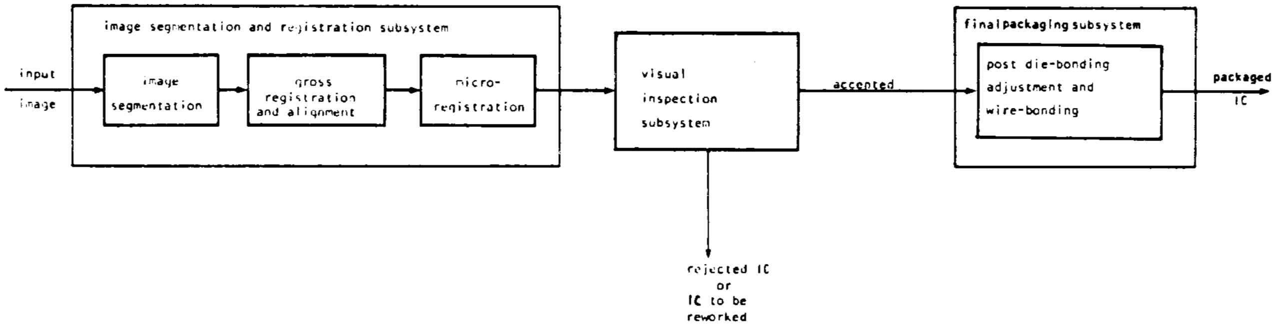


Fig. 7. Block diagram of the proposed integrated system for automatic visual inspection and wire-bonding of IC's.

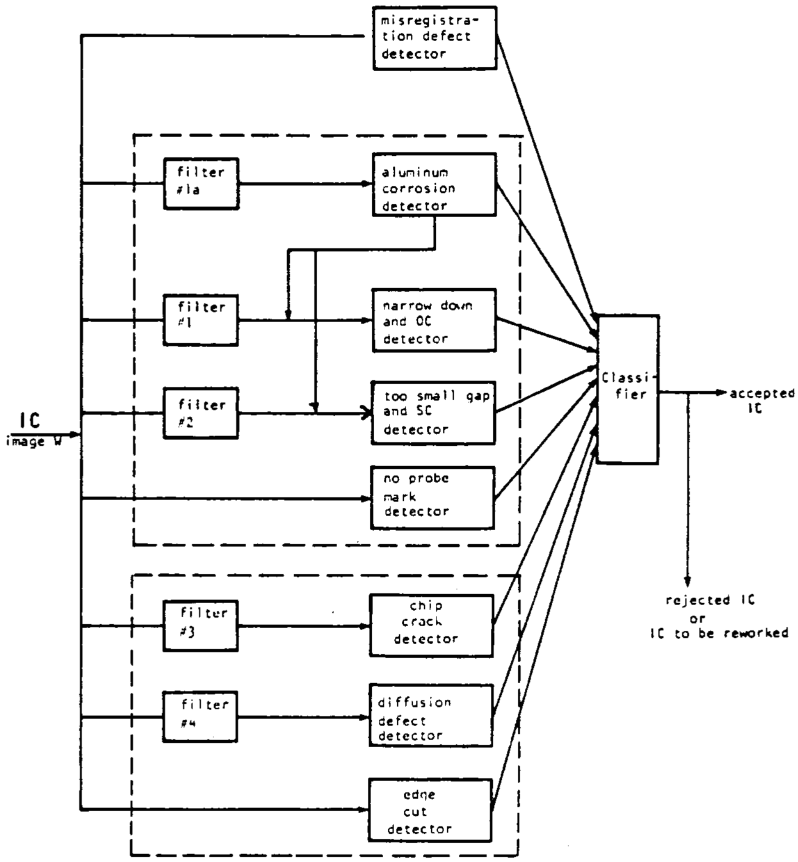


Fig. 8. Block diagram of the proposed IC visual inspection subsystem.

ing of multilayer IC's. The block diagram for their system is shown in Fig. 7. The inspection paradigm is for the most part top-down and model-driven using a treelike syntactic approach. The design and inspection specification model takes the form of a descriptive data base. The various inspection algorithms are called for based on the actions of a controller, which monitors the whole visual inspection process. The inspection process proceeds to the next suggested operation based on the accumulated information of the preceding operation. Each of six subpattern masks are independently registered. The structure of the visual inspection subsystem is shown in Fig. 8. The filters used are task- and context-dependent to reduce ambiguity. Each defect detector outputs a binary response based on whether the IC passed or failed the test. Successful inspection leads to the IC being passed on to the wire-bonding process.

Automated visual analysis has also been applied to the inspection of surface properties such as roughness, scratching and other potential defects. The best successes have come with highly specialized illumination and sensing systems, specifically tailored for a particular application. Recently, greater sophistication in the modeling of the imaging process has lead to prototype surface inspection systems with the promise of increased generality [9], [10].

Industrial Part Identification Systems

The ability to visually locate and identify industrial parts can often greatly aid the handling and manipulation of manufacturing materials. This ability is needed in inspection situations where more than one type of part is involved or where part location and orientation is not tightly controlled. Automated

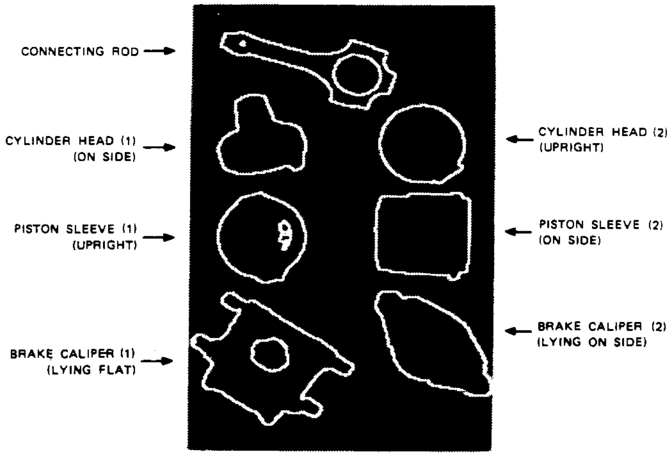


Fig. 9. Edge detected parts in their stable configurations.

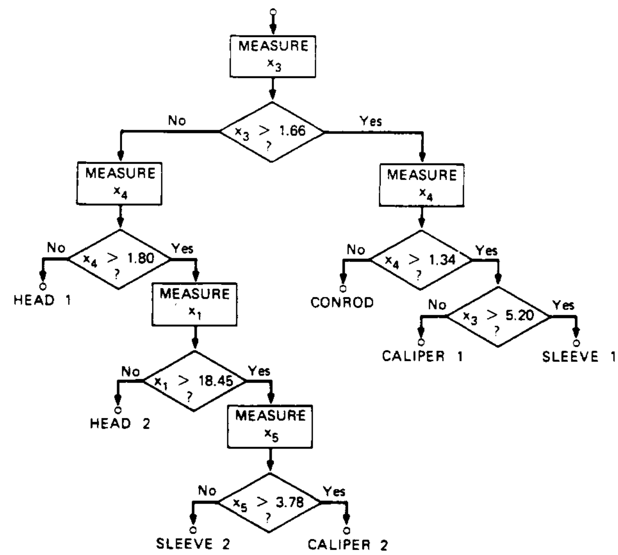


Fig. 10. Tree-parsing pattern recognition process.

pick and place operations can benefit from visual control in a similar way. In applications to follow, the scene will consist of a finite number of objects, each of which can be observed in one of several "stable orientations." In addition, objects are spatially unconstrained and in some instances may partially occlude each other. These relaxations and generalizations of the scene model present additional challenges to automation. The following examples should be considered precursors to automated product assembly, a topic to be discussed later.

Agin and Duda [21], [22] have developed a bottom-up scene analysis control structure for the visual detection of foundry castings. These unoccluded parts are placed on a conveyor belt in one of several stable orientations and scanned from above with a linear array of 128 camera-mounted diode detectors. As in some previous applications, the output of this array is binary. The belt itself serves as the y-axis scan mechanism. Edge data are extracted line by line as transitions from 0 to 1, or vice versa. Next, these transitions are ordered into discrete parts boundaries by the connectivity analyzer. The output of this process for each part in each of its "stable configurations" is shown in Fig. 9. Each discrete part has its centroid computed as a function of these boundary points. Finally, the Cartesian (x, y) boundary points are converted to polar (r, θ) coordinates relative to the part's centroid as a precursor to automated parts recognition. This latter step removes rotational variations in the scene.

The part-identification process used a sequential statistical pattern recognition approach with a treelike hierarchical structure. A sequence of decisions are made leading to a determination of part type. Seven $(n = 7)$ possible features could be extracted from each discrete object. These were

x_1 perimeter of object
x_2 square root of the area
x_3 total hole area
x_4 minimum radius
x_5 maximum radius
x_6 average radius
x_7 compactness (x_1/x_2).

However, as can be seen from Fig. 10, as few as two features need be computed to reliably detect some of these parts using this pattern recognition approach. This structural approach computes the mean or average μ_i of all x_i features over a "training set" of parts data. The two μ's with the largest mean separation for a given feature (i) and pair of parts (c) are selected at the first node on the parsing tree (in this case x_3). These criteria are then recursively applied to the remaining mean differences until all parts are successfully identified. Once the training phase is complete, the decision tree serves as the structure for object identification in the on-line analysis. The technique is quite rapid because of the economies introduced both in image acquisition and classification.

This philosophy has been incorporated into the SRI [22] Vision Module and is used in a variety of parts-identification and manipulation tasks. A commercial version recently became available from Machine Intelligence Corporation of Mountain View, CA. The Vision Module is one of the first attempts to produce an easy-to-use system that can be applied to different applications without extensive reprogramming. The system is "trained" to recognize objects by presenting it with a series of prototypes. Appropriate features and classification rules are then developed. The Vision Module has limitations. First of all, it requires special lighting to produce very high contrast scenes so that objects can be separated from the background by means of a global binary threshold. In addition, recognition must be done using relatively primitive shape features based only on the object's outline. Finally, the generality of the Vision Module reduces performance. However, the system performs its recognition task in less than 1 s. This time is considered acceptable for a production environment.

Yachida and Tsuji [23] have developed a flexible programmable parts-recognition system for characterization of the nonoccluded parts of a small gasoline engine. Fig. 11 shows their processing paradigm. The first phase of boundary detection and object isolation from binary parts images in their denumerable stable orientations resembles Duda's approach. However, the recognition system based on the stored polar coordinate boundary representations differs to a large degree. A uniform data structure is used to describe each part stable-orientation model M_i that consists of a part name, a face (stable orientation) name, and a list of primitive features F_{ij} where $j = 1, \cdots, n_i$. The n_i vary with each model M_i. These features are further classified into those necessary for recognition or assembly. F and I denote recognition and assembly features, respectively. Each primitive feature also has associated with it an operator-entered estimate of its probability of detection. Primitive features F_{ij}, such as hole detectors, line

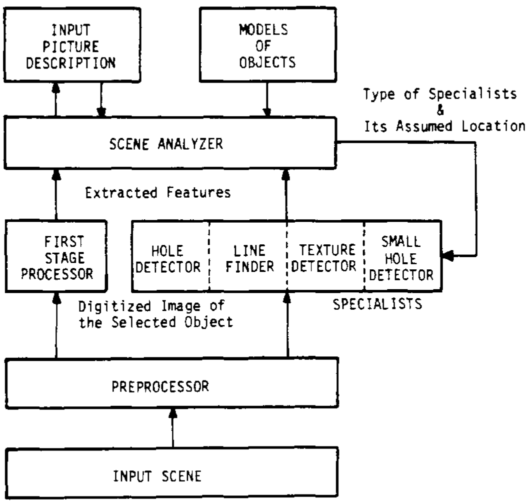


Fig. 11. Trainable operator interactive part-recognition system.

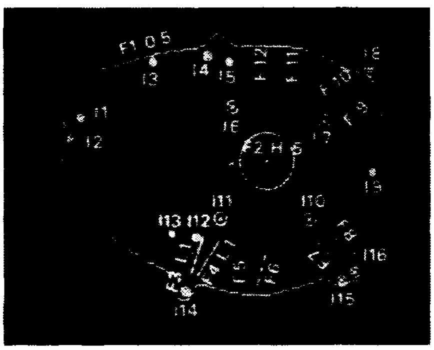


Fig. 12. Recognition and identification features of stable orientation M_i.

finders, and texture detectors are used to analyze the basic stored outlines. Recognition of unknowns is accomplished by a structured step-by-step analysis of the previously stored models. An example of recognition and identification features for a stable part orientation is shown in Fig. 12.

System training involves interactive man-machine examination of the identification task. Once the polar form outline of a new M_i has been automatically extracted, it is shown to the operator by a video display along with the most similar previously detected outlines that at this point have an identical description. The human operator then adds additional features from the primitive group until the new M_i in unique. Through this man-machine symbiosis the system is trained to analyze the scene. If the scene, i.e., the list of models M_i, is changed or augmented, one training session retrains the machine to the task at hand.

Perkins [24] has developed a scene segmentation and interpretation algorithm for industrial parts recognition that differs from previous models in at least two respects. First, it initially operates on multiple gray level rather than binary images, and second it can identify parts that have been partially occluded by other parts. The major sequence of processing operations is shown in Figs. 13 and 14. The 5-bit (32 gray level) 256-by-256 pixel image shown in Fig. 15 is reduced to an edge gradient image of 13(b). Edges with similar gradient magnitudes are linked together to form chains shown in Fig. 13(c). These chains are then characterized as either straight lines or circular arcs collectively referred to as scene concurves as shown in Fig. 14. This successive compression has reduced an array of

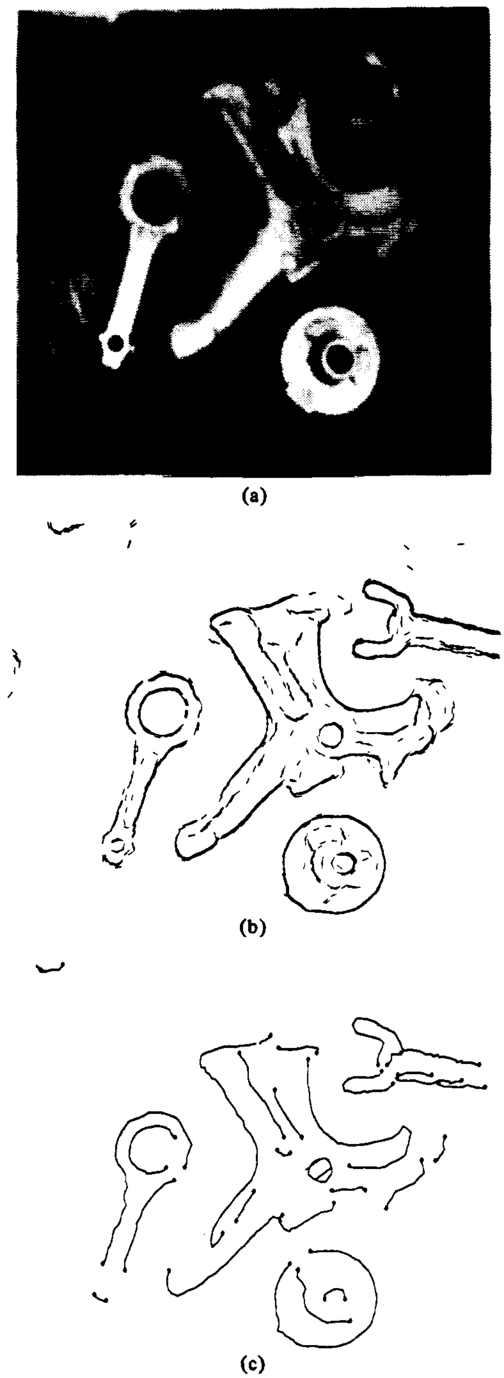


Fig. 13. (a) Digitized parts scene. (b) Hueckel gradient operator. (c) Chain encoded and linked edges.

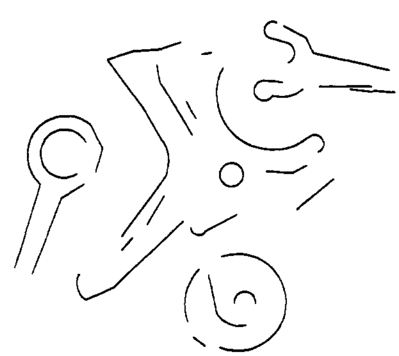


Fig. 14. Concurve representation of Fig. 13.

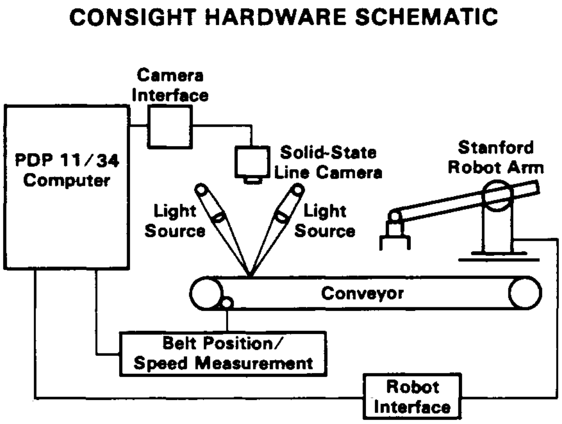


Fig. 15. Consight system.

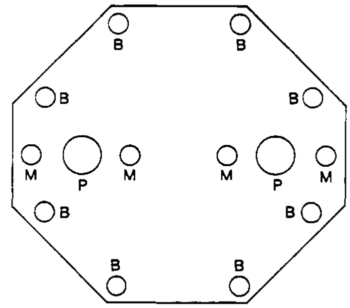


Fig. 16. Complete model of the engine cover.

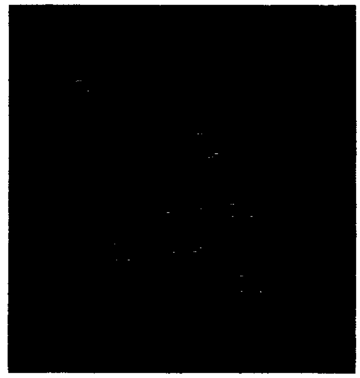


Fig. 17. Assignment based on local information.

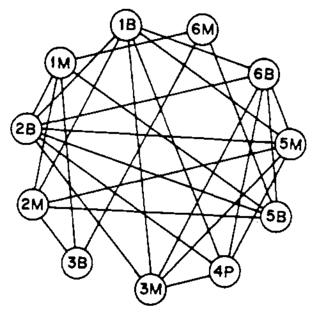


Fig. 18. Compatibility graph for the assignments.

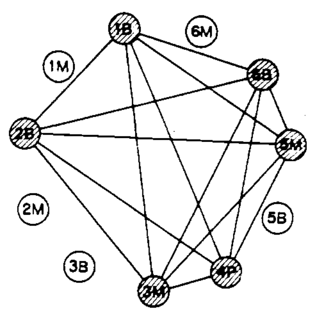


Fig. 19. Largest maximal clique in the graph.

65 000 pixels to perhaps 50 descriptive global concurves by a bottom-up analysis approach. A similar reduction process is also undertaken for each discrete part during the previously completed computer interactive training of the system. In operation, the system must match the previously stored concurve models with the concurves visible in the given scene under analysis. There are three steps to this recognition process. First, a preset control structure selects the order in which combinations of model and scene concurves are to be matched. Next, one model and one scene concurve are matched. Third, the stored model is spatially transformed and rotated to fit the appropriate scene concurves. When partial part occlusion is encountered, fewer concurves will match. However, with high reliability, more matches will occur with the proper part than with any other prestored part model. The ability to match rapidly predicted and extracted image features supports a technique that integrates the prediction and validation phases of object recognition.

Holland [25] has recently developed Consight, a part-identification system, shown in Fig. 15. The Consight system uses a unique form of illumination to reduce the image to contour information. A series of light stripes are projected onto automotive parts. The two light sources superimpose on the conveyor belt but separate when a part passes by because of their different angular inclinations. The degree of separation is proportional to the object thickness, and the point of separation denotes the object boundary. Once the boundary is found, techniques similar to those described above can be used to locate and identify parts. A robot arm can then be positioned to pick up the parts. Consight is particularly interesting because it is the best current example of using controlled illumination to eliminate much of the complexity normally associated with low-level primitive detection.

Bolles [26] has applied the graph theory technique of "maximal cliques" to model relationships between detected local features of industrial parts. Fig. 16 shows a model of a small engine cover. There are three types of holes: pipe holes (P), manifold-bolt holes (M), and bolt holes (B). The M- and B-type holes are indistinguishable when detected by a local operator. Fig. 17 is the detected and labeled output of a local circle detector primitive from a partially occluded view of the engine cover. Based on the model, a node compatibility graph, shown in Fig. 18, can be constructed. In this case, two nodes of the given scene are compatible with the stored model when 1) their labels are compatible, and 2) the distance between holes in the model correspond to the distance between holes in the scene.

A maximal clique is a clique that cannot be extended to include other nodes of the graph. The largest maximal clique is the maximal clique that contains the most nodes of the graph. Fig. 19 shows the largest maximal clique for the example presented. It represents the largest set of mutually consistent assignments of model and scene local features. As mentioned

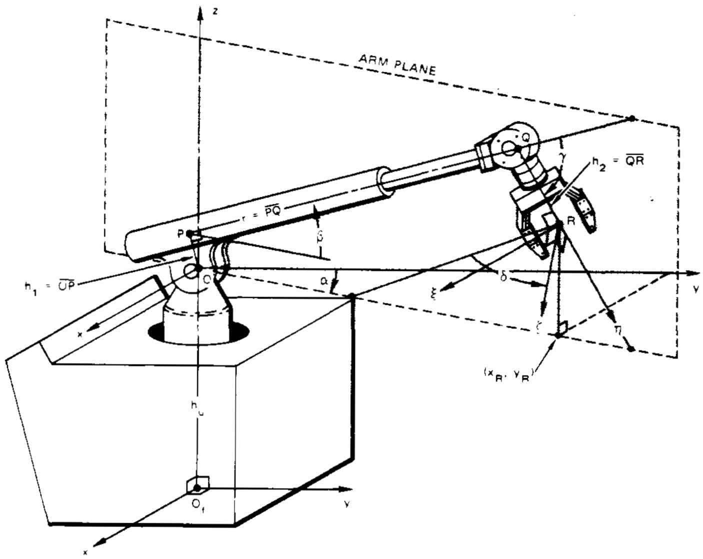


Fig. 20. Diagram of Unimate mechanical manipulator.

earlier, as the number of nodes increases linearly, the computation time increases exponentially. To minimize this disadvantage, it is necessary to minimize both the number of local features detected and the number of possible labels these features can exhibit. The *a priori* grouping of nodes into a hierarchical tree can also limit this growth.

These examples of programmable scene descriptions using industrial parts have both similarities and differences. All of the previous examples were able to identify the part under investigation. Most were able to recognize the position and orientation of a part. Some used interactive man-computer training sessions to previously train their respective recognition systems. Each reduced the huge information content of the scene to a much smaller subset of descriptive edge or edge-derived primitive local features. Most used these locally extracted primitive features in conjunction with prestored object models to derive a consistent explanation of the scene using a flexible control structure. Many of the systems used special purpose lighting procedures to reduce the scene information content prior to digitization. The use of such techniques to specify the knowledge base is a most useful preprocessing step in most current systems. None of these examples performed a visual quality inspection. However, such an inspection would logically follow between the parts recognition phases and the automated assembly operation.

All of the previously described techniques are characterized as having been developed for a specific, narrowly defined class of applications. This of course allows the maximum exploitation of any available knowledge about that application. The problem is that the systems are not easily generalizable. Recently, several groups have been investigating ways of increasing the sophistication of the general-purpose knowledge used at the lower levels of the analysis hierarchy [8]-[10]. Emphasis is placed on determining surface properties for reasonably broad categories of surfaces. This approach may be used in a wide variety of application areas without the need for extensive reengineering of lower level components.

For example, Mundy and Porter [9] recently described techniques useful for detecting defects in metal surfaces. They first develop a formal model for the way in which light is scattered from a surface. The model accounts for the incident angle of illumination, the angle of viewing, and the orientation, roughness, and reflectance of the surface. The model shows that by taking several views from particular positions and with carefully controlled illumination, it is possible to separate and independently measure the different surface properties. This shape-from-shading concept has also been explored by Barrow and Tenenbaum in a more general context [4]. These techniques, along with range sensing, hold a promise of an extension from two- to three-dimensional part characterization and recognition.

Automated Assembly of Discrete Components Using Programmable Manipulators

The research already presented has dealt with the task of scene segmentation for the purpose of recognition of discrete parts as a precursor to automated inspection and assembly. This section explores the use of programmable manipulators with visual feedback for automated assembly of discrete components.

Fig. 20 shows an example of degrees of freedom available in a commerically available Unimate, Inc. industrial robot [27]. These robots are currently being used in repetitive assembly line applications such as simple pick-and-place operations, and automatic spot welding or spray painting of automobile bodies. These robots often do not "sense" the environment around them either with visual or tactile feedback. As a result, they are preprogrammed internally with hardwired logic to perform stylized repetitive tasks with parts prepositioned and oriented within very narrow spatial tolerances. The goal of current manipulator research is to provide for more manipulator flexibility through the use of tactile and/or visual sensors controlled by programmable, thus more flexible, computer systems. The internal manipulator computer control system will then supply the intelligent interpretation of visual and tactile sensor inputs.

Much progress towards these goals has been achieved by various researchers at SRI International [28]-[30] Fig. 21 shows a diagram of a Unimate manipulator that has been interfaced to an LSI-11 microcomputer. The added flexibility

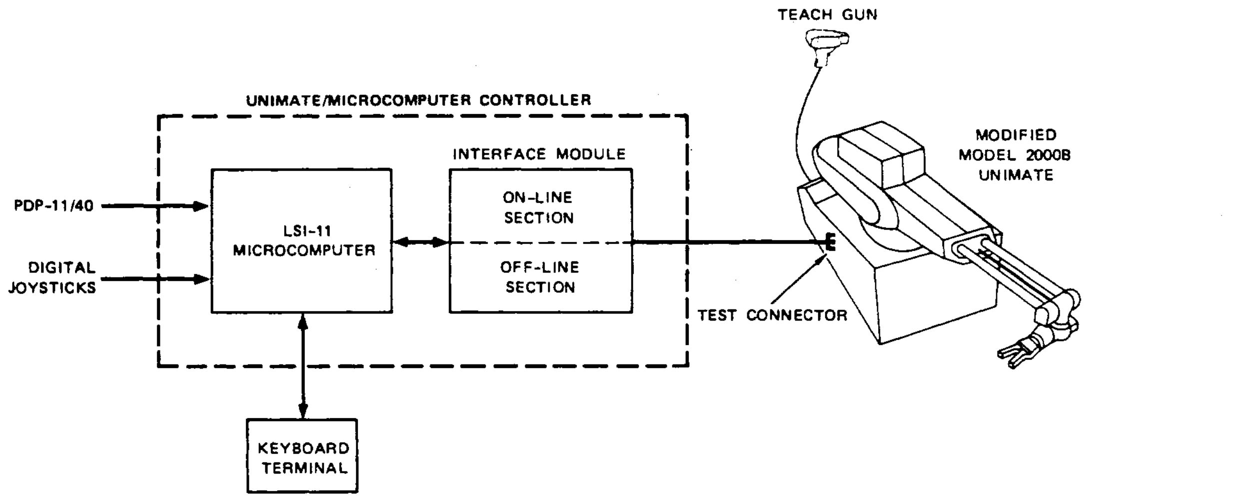


Fig. 21. Unimate manipulator interfaced to a conveyor belt and a microcomputer.

made possible by these modifications includes the incorporation of a position and velocity encoded conveyor belt for material handling, and a computer-controlled operator interactive teach gun used as a binary joystick to control and train the manipulator in its internal coordinate system. Thus, for some applications, the human operator initially trains the manipulator, which "remembers" the paths necessary to perform certain materials-handling or spot-welding tasks. This computer control effectively replaces the previously used fixed logic with programmable, i.e., flexible logic that can allow the easy retraining of the manipulator for new tasks. This increased flexibility has facilitated the experimentation with visual and tactile sensors.

Visual control of the Unimate manipulator at SRI is centered about tasks like the automatic bolting of eight bolts of a compressor head to the block of a small engine without placing bolts in the holes to be later used for the manifold attachment. For this experiment a pneumatic wrench and a 100-by-100 pixel solid-state video camera were mounted on the Unimate control arm. The image is digitized from above by the camera, and the binary image is obtained by thresholding. All candidate hole boundaries are detected by connectivity analysis described in the previous section, and prespecified size and second moment roundness constraints eliminate other features. Scene context is used to eliminate those holes to be used later for manifold attachment. In practice, the device is pretrained under joystick control to position itself in a starting position directly above the work piece. From this position successive visual and mechanical iterations locate and precisely align the target hole to be filled. After the bolt has been inserted, visual confirmation is obtained by taking another binary image from the starting position. The target hole will have disappeared if the operation is successful.

The programmable universal manipulator for assembly (PUMA), manufactured by Unimation Inc., is a commercial version of this prototype. It consists of a supervisory microprocessor and five axis control microprocessors to program one of five possible robot axes of motion. The PUMA is designed to be a relatively inexpensive but extremely flexible manipulator appropriate for the precise handling and assembly of small light parts.

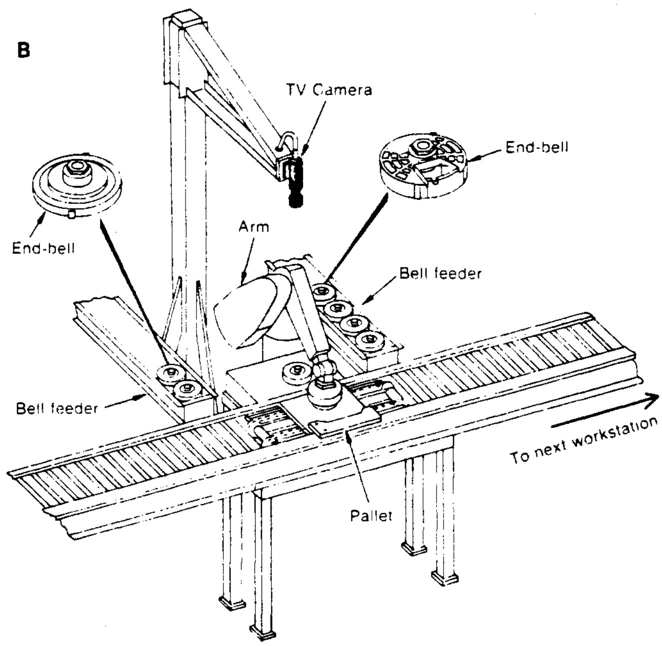


Fig. 22. Prototype Westinghouse PUMA station with visual control.

Sugarman [31] has reported that Westinghouse plans to field a six-robot-prototype system to assemble small electric motors. Each robot will be supplied with special parts-feeding machinery and pallets to precisely orient parts. All but two of the work stations will use pick-and-place robots that have no sensory feedbacks. Two of the work stations will use PUMA five-axis robots with visual feedback to inspect parts for defects and correct assembly and then place them on pallets for transfer to the next work station. The computer control for this system is both distributed and hierarchical, and uses multiple microprocessors. The PUMA-based-work station is shown in Fig. 22.

Much of the cost of a robotics assembly operation is in purchasing elaborate jigs for holding objects in precisely controlled locations. These expenses often exceed the cost of the industrial robots themselves. A vision system capable of

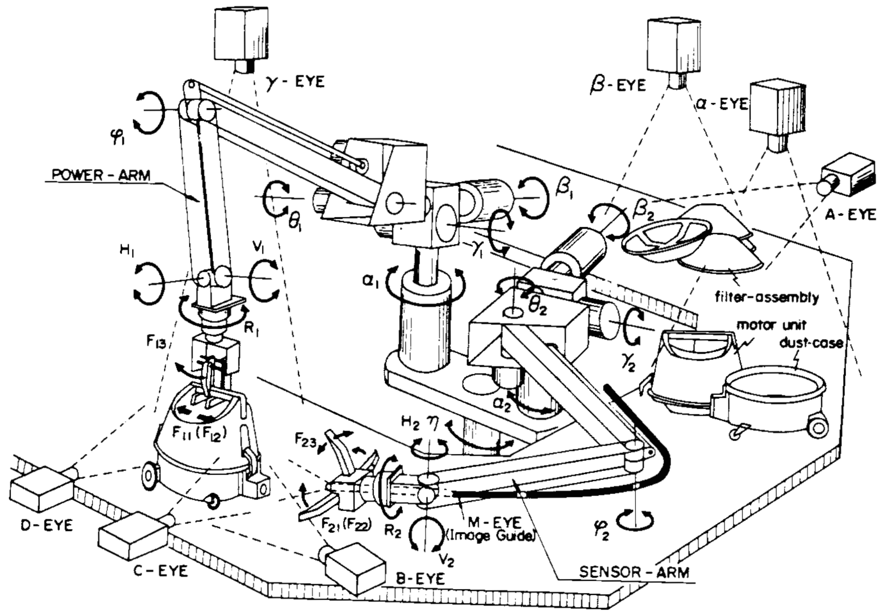


Fig. 23. Hitachi experimental system with visual and tactile sensors.

accurately determining the location of objects without elaborate fixtures would allow much more conventional parts manipulation.

A common means of part storage and transport involves the use of bins. Rosen [1] defines "bin picking" as the acquisition of a part at an initial location followed by presentation at a second location in a predetermined position and orientation. Two approaches to this problem are currently under investigation. Birk *et al.* [32] at the University of Rhode Island have developed a vacuum gripper with six degrees of freedom integrated with a TV camera for vision. Stable holding points are determined using an algorithm that thresholds both a pixel's intensity and gradient in the scene. The intensity threshold is set high to isolate flat surfaces. The gradient threshold is set low to eliminate curved surfaces and edges. The centroid of pixels that passes these thresholds is used to locate the pickup point for the part. Once the object is picked up, a binary image is formed. This image is used to compute the center of gravity and the axis of the minimum moment of inertia. This is sufficient information to determine the parts orientation as acquired.

SRI [33] approached the same problem using a two-step method. First, several identical parts were removed from the bin using an inexpensive limited sequence robot. These parts were then placed on a flat surface with controlled illumination. Then the parts were recognized using measures similar to those previously discussed [22]. Both of these methods have yet to be implemented in a production environment.

An advanced two-armed robot under development at Hitachi [34] assembles vacuum cleaners using both tactile and visual robot sensors. This system, shown in Fig. 23, does not use pallets or fixtures for parts location. Multiple cameras are used to control the arm(s) locations relative to the vacuum cleaner parts. Template matching, using salient parts features, is used to provide spatial cues to the cameras. The computer control is once again distributed and hierarchical.

The robotic systems previously described used varying degrees of part-feeding mechanisms. These fixtures and jigs add cost to the manufacturing process and eventually to the product. The potential use of vision to simplify or eliminate these mechanisms will measurably increase its cost effectiveness and use in inspection and assembly tasks.

Future Technical and Economic Considerations That Impact Industrial Automation

A question centers about the need for automation. It has been generally shown that today's production worker is less likely to be acquiescent when called upon to perform dull, repetitive, arduous, or dangerous jobs. In the latter two instances, government regulation has begun to limit the options of both worker and employer. In the former instances, the result has often been worker dissatisfaction resulting in deliberate sabotage and high absenteeism.

From an economic viewpoint, there is a need for greater reliability and worker productivity that will maximize long-term profitability. Rising labor costs in most industrialized nations, coupled with the growth of multinational enterprise, have effectively transferred labor-intensive standardized manufacturing into third-world markets where labor is cheap. Allan [35] points out that the hourly labor costs in the U.S. have risen from \$3.80 to \$14.00/h during the past 20 years. However, the cost of automation has remained relatively constant at \$4.60/h. He also points out that robots compare favorably with manual labor in fixed automation batch processes that constitute 75 percent of all U.S. manufacturing operations. Rosen [1] points out that 17 percent of all production workers are involved in assembly tasks, while 10 percent are involved in inspection tasks. Thus the potential market for automation is viable from several viewpoints.

An estimated 40 000 [35] industrial robots of all types are in operation worldwide. Of this number, 30 000 [35] are being used in Japan. Why has Japan led the way in industrial automation efforts? Yachica and Tsuji [36] enumerate the following possible reasons.

First, there is a positive and cooperative attitude between labor, business, and government towards new technological development. There is also a similar joint perception of the urgent need to automate medium- and low-volume production. Japanese industry is experienced in and does not fear the introduction of computers and microprocessor-based automation in their factories. Lastly, there is a tradition of technology transfer from basic research to practical systems in Japanese industry. These attitudes, along with a 10-year, 12-million-dollar per year effort funded by Japan's Ministry of International Trade and Industry in Pattern Information Processing, are beginning to yield impressive results. To these reasons can be added yet another. Most Japanese companies are funded primarily by long-term bank loans arranged with the aid of the government's finance ministry. The typical U.S. corporation is 50 percent funded through the issuance of stock. This figure rarely reaches 16 percent in Japan [37]. Thus there is a greater need to maintain a substantial flow of dividends for a typical U.S. corporation. This can and often does inhibit the introduction of capital intensive productivity enhancing technology because its introduction can adversely affect short-term profitability.

For these reasons, American industry continues to stress short-term economic goals. Such areas as cash management, quarterly profit goals, and the need for a short payback period for capital improvements constrain the vision of too many managers. The elevation of financial planners to positions of primacy in many corporations has encouraged this trend. Strategic planning of products and processes is too often neglected. Much emphasis is often placed on extracting the last bit of profit from mature products using increasingly outmoded physical plants and manufacturing processes. At times, government tax policies have encouraged this trend.

Allan [35] and Myer [38] point out other potential economic and technical obstacles to automation in this country. First, automation efforts must compete for investment funds with other projects. They must also compete for the support of manufacturing engineers and factory management. It may also be necessary to redesign the work methods and processes to isolate the essential visual and manipulative requirements from the total manufacturing process in order to minimize costs and increase efficiency. This would seem to suggest a long-term strategy where automation is planned into new production facilities, not added on, if possible, later. There is also a tradeoff between structuring the hardware to fit one application (fixed automation) and developing computer architectures and languages/algorithms powerful enough to solve broad classes of recognition problems without expensive system redesign. Finally, the addition of vision to a Unimate or similar industrial robot can increase the cost of such a system by as much as 50 percent. Rosen [1] has indicated that vision must not add more than 25 percent additional cost to a robot if it is to be cost effective.

There are economic policies that can circumvent the economic obstacles. Capital intensive products requiring technical innovation remain the primary mainstay of the industrialized nations. Tax structures, when properly formed, can encourage capital intensive manufacturing processes. For instance, government can effectively encourage the purchase of productivity enhancing machinery and subsidize research and development efforts through the use of investment tax credits and rapid depreciation allowances. It can also encourage capital formation by encouraging saving and investment through its taxing policies. Governments increasingly penalize the use of manual labor through a myriad of health and safety requirements, workman's compensation, and social security payments.

There is a need to use less marginal low-level manual labor. The goal of automation should then be to push the man higher up the cognitive ladder by multiplying his capabilities and by relieving him of undesirable tasks. Many of the applications previously discussed have attempted to achieve this goal. Often the system has been initially trained to perform some function by a man–machine training session that is symbiotic to both. This is often referred to as "training by example." Once training is completed, the capabilities of the man are multiplied manyfold by this flexible form of automation.

Technical advances will also assist the growth of visually aided automation. There has always been a need for a flexible form of automation that could handle small or medium production runs of several similar products without requiring a massive reinvestment in "hard" automation equipment. The recent introduction of inexpensive microcomputers and microprocessors has made possible cost-effective reprogramming, i.e., restructuring of hardware heretofore impossible. These inexpensive digital devices have, and promise to have, the effect of allowing increased intelligence closer to the point of need than was possible in the past.

Most current inspection/assembly systems operate on binary imagery and/or require highly structured scene presentations and lighting. Gray-level imagery requires more software/hardware computing resources but have much greater potential flexibility. There is a need and promise for hardware fast enough to operate at a speed of one scene/second on gray scale imagery. The development of both very large-scale integration (VLSI) technology and of more capable microcomputers in parallel configurations will help accomplish this goal. Fast raster-based pipeline preprocessing hardware to compute low-level features on local regions of an entire scene have and will encourage fast acquisition of low-level scene primitives. This along with the continuing rapid decline in the cost of digital hardware and its increasing reliability and speed also bode well for the future. However, our understanding of effective algorithms for gray-scale processing is still extremely limited.

The decline of hardware costs will be accompanied by a rise in the cost of software and firmware to control these system. It is perhaps ironic that the field of industrial automation, like other areas of computer technology, will itself become labor intensive. That is, the cost of software and firmware development will exceed the cost of the hardware itself. If this is or soon will be the case, at least man will have become the dominant cost consideration on a higher cognitive plane where productivity is enhanced.

Future inspection and assembly software systems must be easily modified to adapt to new situations. Current approaches are limited in this respect. Agin [39] indicates that what is needed are control structures that will specify tests to be performed and potential alternate paths of action. Computer languages now exist for controlling manipulators [40]. It is possible that several of the programming languages developed for artificial intelligence research might be adapted to CAM environments [41]. It is likely that a language and programming environment specifically intended for the design and implementation of industrial vision systems will be a major contribution. This will reduce system software/firmware costs by encouraging modular implementation of basic knowledge.

An understanding of the computation techniques employed in machine vision is possible only if the problem is decomposed into manageable computational modules. The computational modules may then be controlled by the nature of the decomposition and the ways in which modules interact. These adaptive control structures will be critical in coordinating the proper use of these units in the future.

Lower level primitive detection will be constrained by what is computable from the scene itself. Higher level symbolic representations must contain the information required to support the ultimate inspection or assembly goal. In between these levels, the control structure must provide for the timely acquisition of necessary information between intermediate levels of representation.

For optimal efficiency, the choices of control structures and implementation hardware must be carefully coordinated. Most systems installed in the U.S. have used relatively low-cost general-purpose computers. The Japanese, on the other hand, are making extensive use of special-purpose hardware to implement some operations. While special hardware holds the potential of dramatic increases in speed, such benefits are only possible for some control strategies. Top-down control and the use of higher level knowledge becomes more difficult when special hardware is used.

Finally, CAD/CAM technology useful for part design and manufacturing must be integrated with computer vision technology for inspection and assembly. This will facilitate the design and manufacture of parts that are symbiotic with subsequent automated inspection and assembly paradigms.

Kelly [42] states that the CAD/CAM model presently used for part definition, communication and storage, usually consists of several two-dimensional drafting views of a three-dimensional part. As a result, the predominant use of the computer in CAD/CAM systems is that conventional drafting methods have been automated. He further points out that three-dimensional models based on volumes or surfaces will soon replace these conventional models in future CAD/CAM systems. These advanced models will allow proposed parts to be realistically simulated using shaded graphics. Machine vision algorithm design factors such as scene illumination, viewing perspective, number of stable part orientations, surface texture, and uniqueness could all be investigated before a parts assembly is actually fabricated. One could then envision using three-dimensional CAD/CAM models as a product synthesis paradigm, while using machine vision models as the corresponding product analysis paradigm. In this environment it is also probable that joint analysis/synthesis algorithms, based on optimization criteria, would be developed. These would be similar in concept to joint circuit analysis/synthesis algorithms that have existed for some time. "Training by being told" would replace the "training by example" paradigms described in several examples presented earlier.

Conclusions

Industrial computer vision systems must analyze objects in a manner of value for inspection or assembly operations. This means that a description of the objects must be produced. A particular analysis technique may be characterized by the nature of the formal description produced, the computational techniques employed, and the degree to which the formal description accurately reflects pertinent information about the object. One major advantage of studying computer vision within the context of the industrial environment is that the nature of the desired descriptions can be determined. For inspection tasks, output can consist of a list of deviations between the observed object and a model of the "correct" object. For assembly tasks, the identity, orientation, and spatial position of parts will also be available.

The current state of the art precludes the construction of one general-purpose computer vision system with applicability to all industrial vision tasks. In fact, such a system may never be cost effective. Industrial computer vision systems are likely to be installed in the work place to the degree that they can be organized around generic classes of applications. This has and will lead to a number of intelligent systems. Current systems use no common primitives for formal representations of object properties. There is also no common programming language for these applications. This situation will likely improve as computer vision becomes more integrated into the production process.

It is likely that programmable automation will be most useful in environments where product changeovers are frequent, part-transfer fixtures are inexpensive, production volume is small per product, and scene lighting can be rigorously controlled. It also seems likely that distributed and hierarchical computing will continue to be attractive. This latter consideration once again reflects on the need for more robust control structures.

Flexible programmable industrial automation of visual tasks can have a long-range positive future economic impact for industrialized nations by allowing the repatriation of labor-intensive standardized product manufacturing currently lost to less expensive labor markets. Productivity can be raised via the judicious and cost-effective use of flexible computer vision devices. There must, however, be increased communication between managers, quality control, and production engineers on one hand, and computer-vision-based image processing and digital design specialists on the other. The design of future computer-vision systems for industrial applications will necessitate a formal integration of computational and control operations to achieve accurate and efficiently derived knowledge about the scene. Significant performance improvements in industrial vision systems will occur only when these concepts are integrated with other aspects of the parts design and manufacturing process.

Acknowledgement

The authors wish to thank Dr. Glen Wade for the invitation to write this paper and all the authors who supplied necessary illustrative figures. We also wish to thank the reviewers for their valuable comments.

References

[1] C. A. Rosen, "Machine vision and robotics: Industrial requirements," in *Computer Vision and Sensor Based Robots*, G. G. Dodd and L. Rossol, Eds. New York: Plenum Press, pp. 3–20.

[2] J. L. Mundy and J. F. Jarvis, "Automatic visual inspection," in *Applications of Pattern Recognition*, K. S. Fu, Ed. New York: CRS Press, 1980.

[3] Y. Y. Hsieh and K. S. Fu, "A method for automatic visual inspection and wire bonding of integrated circuits," Purdue University, Lafayette, IN, Rep. TR-EE-80-45, Oct. 1980.

[4] H. G. Barrow and J. M. Tenenbaum, "Computational vision," *Proc.* IEEE vol. 69, pp. 572–595, May 1981.

[5] W. B. Thompson, Ed., "Machine perception," Computer Mag., May 1980.

[6] R. P. Kruger, Ed., "Imaging applications for automated industrial inspection and assembly," *Proc. SPIE*, vol. 182.

[7] P. A. Samuelson, *Economics*, 10th ed.. New York: McGraw-Hill, 1976.

[8] D. Marr, "Representing visual information," in *Computer Vision Systems*, A. R. Hanson and E. M. Riseman, Eds. New York: Academic Press, 1978, pp. 61–80.

[9] J. L. Mundy and G. B. Porter, III, "Visual inspection of metal surfaces," in *Proc. 5th Int. Joint Conf. Pattern Recognition*, Dec. 1980.

[10] R. J. Woodham, "Reflectance map techniques for analyzing surface defects in metal castings," Ph.D. dissertation, M.I.T., Cambridge, MA, June 1978.

[11] M. L. Baird, "Sight-1: A computer vision system for automated IC chip manufacture," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-8, 1978.

[12] W. A. Perkins, "A model based vision system for industrial parts," *IEEE Trans. Comput.*, vol. C-27, 1978.

[13] J. F. Jarvis, "A method for automating the visual inspection of printed wiring boards," in *Proc. SITEL-LILG Seminar Pattern Recognition* (Univ. Leige, Sart-Telman, Belgium), Nov. 1977.

[14] ——, "Automated visual inspection of printed wiring boards by local pattern matching," *IEEE Trans. Patt. Anal. Machine Intelligence*, vol. 2, pp. 77–82, Jan. 1980.

[15] L. Krakauer and T. Pavlidis, "Visual printed wiring board fault detection by a geometrical method," in *Proc. COMSAC* (Chicago, IL), pp. 260–265, Nov. 1979.

[16] R. Chin, C. A. Harlow, and S. J. Dwyer, "Automated Inspection Techniques," in *Proc. Assembly IV Conf. Exposition* (Detroit, MI), Nov. 1977.

[17] C. A. Harlow, S. E. Henderson, D. A. Rayfield, R. J. Johnston, and S. J. Dwyer, "Automated inspection of electronic assemblies," *Comput.* pp. 36–45, 1975.

[18] S. Kashioka, M. Ejiri, and Y. Sakamoto, "A transistor wire bonding system utilizing multiple local pattern machine," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-6, 1976.

[19] Y. Y. Hsieh and K. S. Fu, "An automatic visual inspection system for integrated circuit chips," *Comput. Graph. Image Processing*, vol. 14, pp. 293–343, 1980.

[20] Y. Y. Hsieh and F. S. Fu, "A method for automatic IC chip alignment and wire bonding," in *Proc. IEEE Computer Soc. Conf. Pattern Recognition and Image Processing* (Chicago, IL), 1979.

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

[22] G. J. Gleason and G. J. Agin, "A modular system for sensor-controlled manipulation and inspection," in *Proc. 9th Int. Symp. Industrial Robots*, SME and Robot Institute of America, Mar. 1979.

[23] M. Yachida and S. Tsuji, "A versatile machine vision system for complex industrial parts," *IEEE Trans. Comput.*, vol. C-26, 1977.

[24] W. A. Perkins, "A model-based vision system for industrial parts," *IEEE Trans. Comput.*, vol. C-27, 1978.

[25] S. W. Holland, L. Rossol, and M. R. Ward, "Consight-I: A vision-controlled robot system for transferring parts from belt conveyors," *Computer Vision and Sensor-Based Robots*, G. G. Dodd and L. Rossol, Eds. New York: Plenum Press, 1979, pp. 81–100.

[26] R. C. Bolles, "Robust feature matching through maximal cliques," *SPIE* vol. 182, Imaging Applications for Automated Industrial Inspection and Assembly, 1979.

[27] C. Rosen, D. Nitzan, G. Agin, R. Blean, R. Duda, and G. Gleason, "Fourth report exploratory research in advanced automation," SRI Rep., June 1975.

[28] C. A. Rosen and D. Nitzan, "Use of sensors in programmable automation," *Comput.*, pp. 12–23, Dec. 1977.

[29] J. W. Hill and A. J. Sword, "Manipulation based on sensor-directed control; An integrated end effector and touch sensing system," in *Proc. 17th Annu. Human Factors Soc. Conv.* (Washington, DC), Oct. 1973.

[30] C. Rosen, D. Nitzan, G. Agin, J. Hill, R. Blean, G. Gleason, D. Hill, D. McGhie, and W. Park, "Seventh report machine intelligence research applied to industrial automation," SRI Rep., Aug. 1977.

[31] R. Sugarman, "The blue collar robot," *IEEE Spectrum*, vol. 17, pp. 53–57, Sept. 1980.

[32] J. R. Birk, R. B. Kelley, and A. Martins, "An orienting robot for feeding workpieces stored in bins," *IEEE Trans. Syst. Man, Cybern.*, vol. 11, Feb. 1981.

[33] C. A. Rosen *et al.*, "Exploratory research in advanced automation, Reps. 1 through 7, prepared for National Science Foundation by Sanford Research Institute, Dec. 1973–Aug. 1977.

[34] S. Kashioka, S. Takeda, Y. Shima, T. Uno, and T. Hawada, "An approach to the integrated intelligent robot with multiple sensory feedback visual recognition techniques, in *Proc. 7th Int. Symp. Industrial Robots*, pp. 531–538, 1977.

[35] R. Allan, "Busy robots spur productivity," *IEEE Spectrum*, pp. 31–36, Sept. 1979.

[36] M. Yachida and S. Tsuji, "Industrial computer vision in Japan," *Comput. Mag.* pp. 50–60, May 1980.

[37] E. F. Vogel, *Japan as Number 1*. New York: Harper and Row, 1979, p. 135.

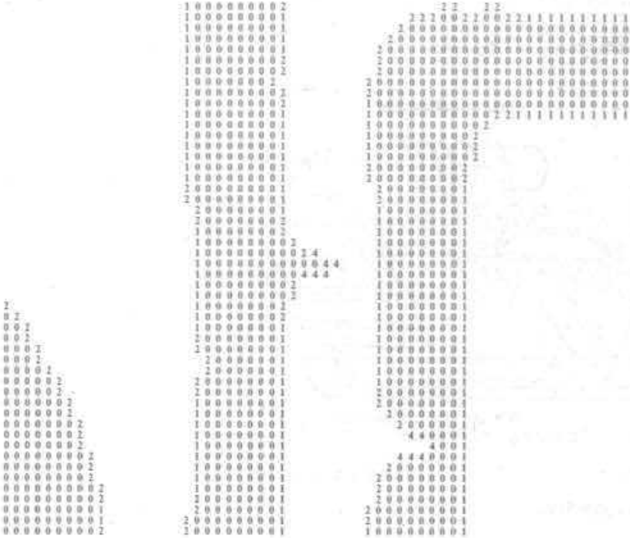
[38] W. Myer, "Industry begins to use visual pattern recognition," *Comput. Mag.*, pp. 21–31, May 1980.

[39] G. Agin, "Computer vision systems for industrial inspection and assembly," Comput. Mag. pp. 11–21, May 1980.

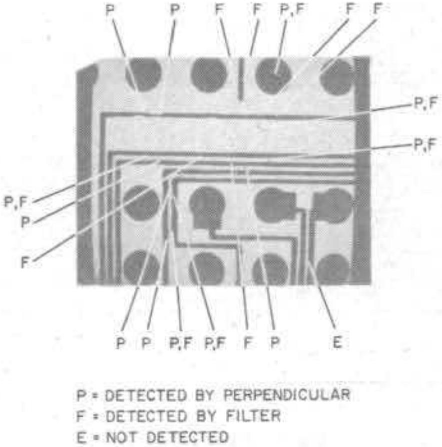
[40] "Users' guide to VAL," Unimation, Inc., Danbury, CT, 1979.

[41] C. Rieger, J. Rosenberg, and H. Samet, "Artificial intelligence programming languages for computer aided manufacturing," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-9, pp. 205–226, Apr. 1979.

[42] J. C. Kelly, R. A. Crowe, G. L. Finlay, G. C. Jackson, B. E. Koopman, and R. L. Wickwar, "An evaluation report on the initial graphics exchange specification (IGES)," Livermore National Laboratory, Livermore, CA, UCID-18864, Jan. 1981.



(a)



(b)

Fig. 1. (a) Example of computer output on a small section of a PCB. (b) PCB board errors as detected by Pavlidis and Krakauer.

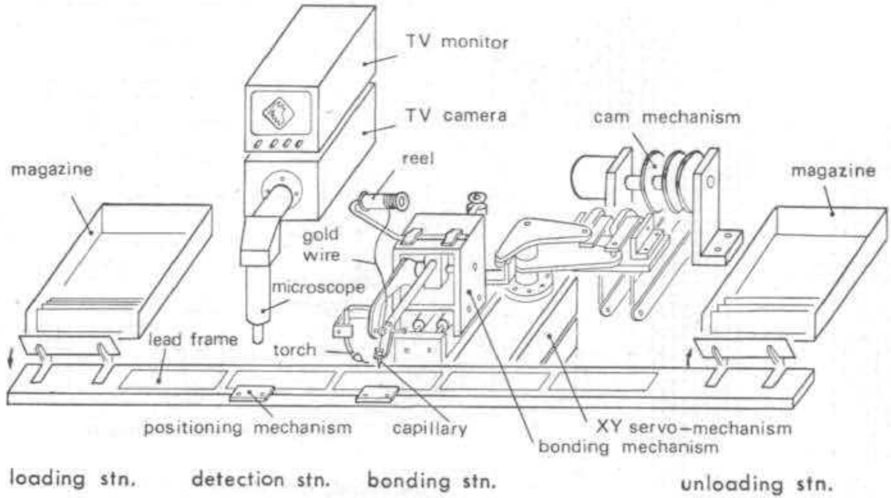


Fig. 3. Wire-bonding machine.

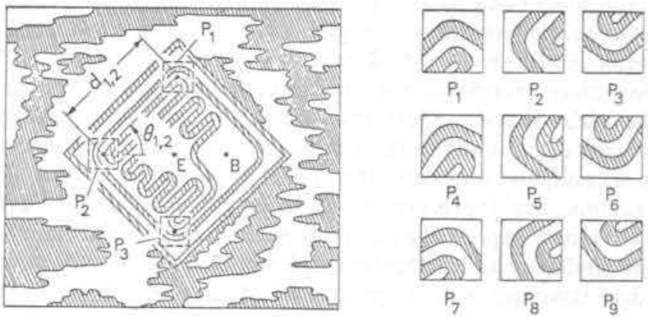


Fig. 4. Binary templates used for pattern matching with an example of use.

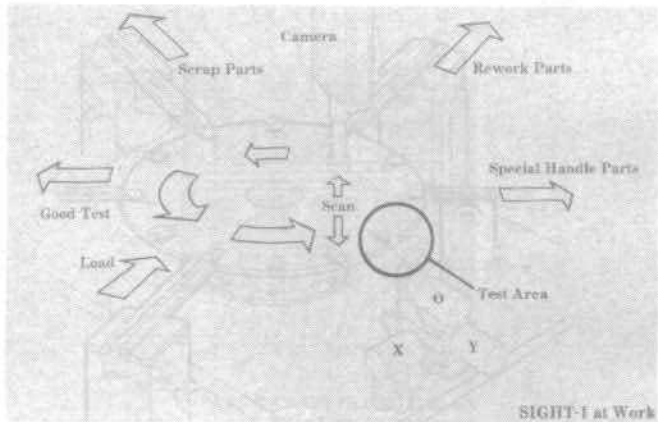


Fig. 5. Sight-1 system.

the machine for subsequent repetitive automatic operation. It should be noted that this system does not make a quality con-

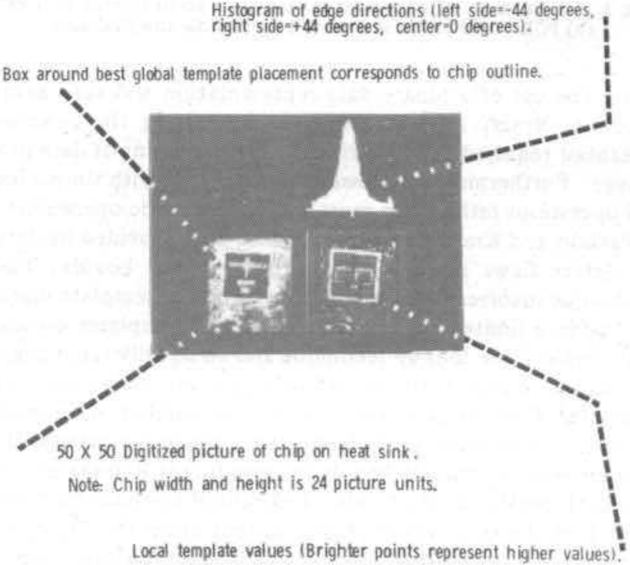


Fig. 6. Gradient histogram to determine chip orientation.

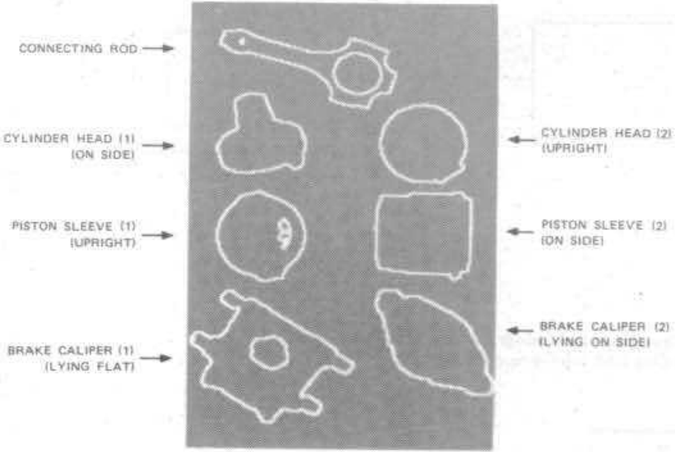


Fig. 9. Edge detected parts in their stable configurations.

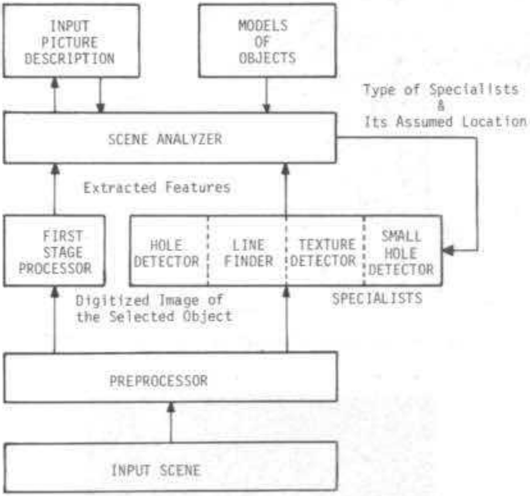


Fig. 11. Trainable operator interactive part-recognition system.

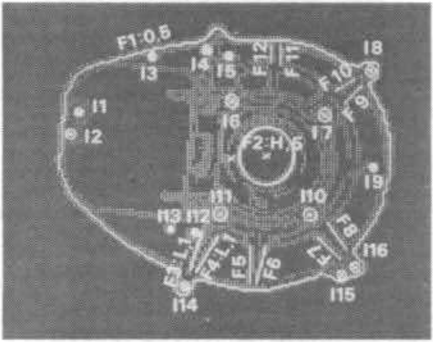
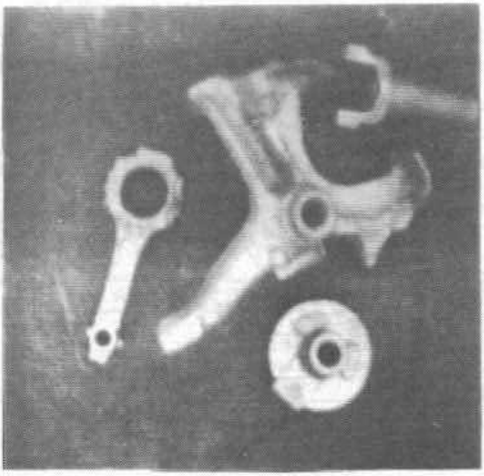
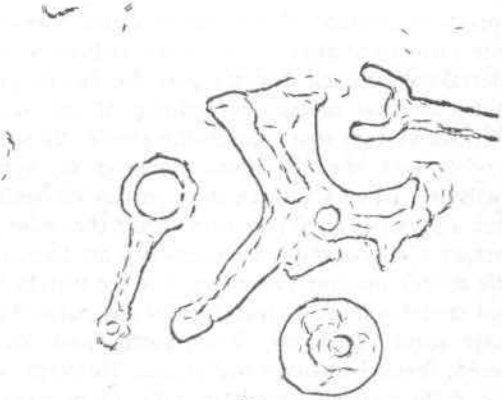


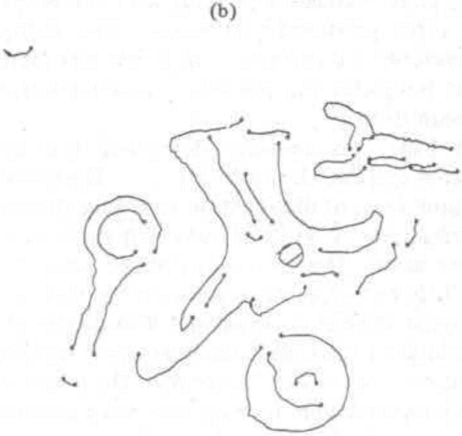
Fig. 12. Recognition and identification features of stable orientation M_l.



(a)



(b)



(c)

Fig. 13. (a) Digitized parts scene. (b) Hueckel gradient operator. (c) Chain encoded and linked edges.

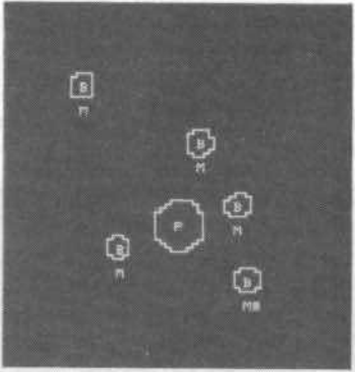


Fig. 17. Assignment based on local information.