1983年美国摄影-光学仪器工程师学会会议录
第442卷：机器人学和机器人传感系统

Third in the SPIE Critical Reviews of Technology Series

PROCEEDINGS

Of SPIE-The International Society for Optical Engineering

Volume 442

Robotics and Robot Sensing Systems

David Casasent, Ernest L. Hall
Chairmen/Editors

August 25, 1983
San Diego, California

Third in the SPIE Critical Reviews of Technology Series

Proceedings of SPIE—The International Society for Optical Engineering

Volume 442

Robotics and Robot Sensing Systems

David Casasent, Ernest L. Hall
Chairmen/Editors

August 25, 1983
San Diego, California

An SPIE Critical Reviews of Technology conference brings together an invited group of experts in a selected technology. Each presents an extended critical review paper that characterizes the state of the art of the technology by assessing key developments from past research and identifying important areas of current work. The conference is designed to present an overview of the technology and to result in a proceedings of value to individuals working in this or related technologies.

Published by
SPIE—The International Society for Optical Engineering
P.O. Box 10, Bellingham, Washington 98227-0010 USA
Telephone 206/676-3290 (Pacific Time) • Telex 46-7053

SPIE (The Society of Photo-Optical Instrumentation Engineers) is a nonprofit society dedicated to advancing engineering and scientific applications of optical, electro-optical, and photo-electronic instrumentation, systems, and technology.

The papers appearing in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and are published as presented and without change, in the interests of timely dissemination. Their inclusion in this publication does not necessarily constitute endorsement by the editors or by SPIE.

Please use the following format to cite material from this book:
Author(s), "Title of Paper," *Robotics and Robot Sensing Systems,* David Casasent, Ernest L. Hall, Editors, Proc. SPIE 442, page numbers (1983).

Library of Congress Catalog Card No. 83-051055
ISBN 0-89252-477-4

Printed in the United States of America.

Conference Committee

Chairmen
David Casasent
Carnegie-Mellon University

Ernest L. Hall
University of Tennessee

Session Chairmen
Session 1—Robotic Sensing I
Ernest L. Hall, University of Tennessee

Session 2—Robotic Sensing II
David Casasent, Carnegie-Mellon University

INTRODUCTION

No prior technology topic has captured the interest and attention of people as much and as fast as the area of robotics. Effective industrial automation, inspection, parts handling and cost-efficient production necessitate the use of robotics. Scientists, researchers, and managers in industry, government and university are intensely interested in this technology. New commercial products and systems emerge monthly and a wealth of robotics institutes have been established in many major universities. At the August 1982 SPIE Symposium in San Diego, a three-day conference was held on this topic. The 45 papers from this conference are available as SPIE Volume 360 entitled *Robotics and Industrial Inspection.* At the November 1983 SPIE Cambridge meeting, another 105 papers on this topic will be presented (SPIE Volume 449, entitled *Intelligent Robots: 3rd International Conference on Robot Vision and Sensory Controls*).

This research area is clearly most appropriate for this present Critical Review of Technology set of papers. The ten papers included herein were all invited papers by experts and leaders in various aspects of robotics. Although they do not represent all aspects of this technology, they provide a review of ten different aspects of this highly multidisciplinary technology. A history of robot vision or computer vision is advanced in the first paper. Image sensors are then surveyed. Binary and gray-scale data handling is the next topic discussed. Shape discrimination and distortion-invariant pattern recognition are treated in the next two papers. Force sensors are the next topic surveyed. This is followed by the control aspects of robots, a specific case study for the nuclear industry, and finally survey reviews on three-dimensional image processing and object measurements.

David Casasent
Carnegie-Mellon University

Ernest L. Hall
University of Tennessee

ROBOTICS AND ROBOT SENSING SYSTEMS

Volume 442

Contents

Session 1

Robotic Sensing I

Chairman
Ernest L. Hall
University of Tennessee

Robot Vision for Machine Part Recognition †

K. S. Fu
School of Electrical Engineering
Purdue University
W. Lafayette, Indiana 47907
U.S.A.

I. Introduction

Most industrial applications of computer vision can be categorized into two groups. They are (1) visual inspection and (2) machine parts recognition. There are several review articles for automatic visual inspection [1,30,31]. This paper gives a brief review of robot vision system for machine part recognition. A robot vision system for machine parts recognition contains four sub-systems: (1) sensing, (2) segmentation, (3) description, and (4) recognition. A block diagram of such a system is shown in Fig. 1.

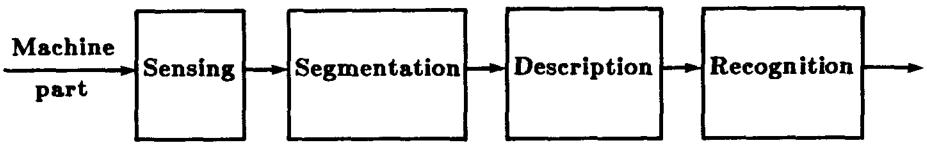

Fig. 1 Computer vision system for machine part recognition

II. Visual Sensing

In a robot vision system, visual information is converted into electrical signals by the use of visual sensors. The most commonly used visual sensors are vidicon cameras and solid state diode arrays. The output of a vidicon camera can be digitized into pixels with two or more gray levels for digital processing. on the other hand, an array of binary outputs can be directly obtained from a solid state diode array. More sophisticated visual sensors include laser, CCD and CID cameras. The vision system may use one camera or multiple cameras. Stereo vision can be accomplished using two cameras. Multiple views of an object or a (time) sequence of object images can be obtained by a multi-camera system.

Illumination of a scene can often be controlled in an industrial environment. A well-designed lighting system illuminates the scene so that the complexity of the resulting image is minimized, while the information required for recognition is enhanced. Commonly used lighting system for industrial applications include diffuse-lighting [74], backlighting [30], spatially modulated lighting [21,54], and directional lighting [30]. It should be pointed out that ordinary light sources may not be the best choice for lighting. Light sources with special spectral ranges or lasers have been used in some applications.

III. Segmentation ††

There is no universal method of segmenting an image into subimages. During the past decade, many segmentation techniques have been proposed [32]. These segmentation techniques can be categorized into three classes, (1) characteristic feature thresholding or clustering, (2) edge detection, and (3) region extraction. One way to define image segmentation is as follows. Let X denote the array of picture points (or pixels), i.e., the set of pairs

$$\{i,j\}, \quad i=1,2,\ldots N, \; j=1,2,\ldots M$$

where N and M are the number of pixels in the x and y directions respectively. Let Y be an nonempty subset of X consisting of contiguous pixels. Then a uniform predicate P(Y) is one which assigns the value true or false to Y, depending only on properties related to the brightness of the pixels of Y. A segmentation of X for a uniform predicate P is a partition of X into disjoint nonempty subsets $X_1, X_2, \ldots, X_n$ such that:

† This work was supported by the NSF Grant ECS 81-19886.

†† For detailed discussions and references on segmentation, see [32,36,39,68].

$$\text{(i)} \quad \bigcup_{i=1}^{n} X_i = X$$

$$\text{(ii)} \quad X_i,\ i=1,2,\ldots,n, \text{ is connected.}$$

$$\text{(iii)} \quad P(X_i) = \text{TRUE for } i=1,2,\ldots,n \tag{1}$$

$$\text{(iv)} \quad P(X_i \cup X_j) = \text{FALSE for } i \neq j \text{ where } X_i \text{ and } X_j \text{ are adjacent.}$$

Segmentation can also be considered basically a process of pixel classification [68]. An image is segmented into subimages by assigning the individual pixel to classes. Thus, many pattern classification methods [28] can be applied to image segmentation. For example, when an image is segmented by thresholding its gray level, the pixels are classified into "dark" and "light" classes so that dark objects are distinguished from their light background. Similarly, in edge detection, the pixels are classified into "edge" and "not edge" by various edge operators proposed.

(A) Characteristic feature thresholding or clustering

Characteristic feature thresholding is a technique widely used in image segmentation. In its most general form, thresholding is described mathematically as:

$$S(x,y) = k \quad \text{if } T_{k-1} \leq f(x,y) < T_k \qquad k=0,1,2,\ldots,m \tag{2}$$

where (x,y) is the x and y co-ordinate of a pixel; $S(x,y)$, $f(x,y)$ are the segmented and the characteristic feature (e.g. gray level) functions of (x,y) respectively; $T_0,\ldots,T_m$ are threshold values with T_0 equal to the minimum and T_m the maximum; m is the number of distinct labels assigned to the segmented image. A threshold operator T can be viewed as a test involving a function T of the form

$$T(x,y,N(x,y),f(x,y))$$

where $N(x,y)$ denotes some local property of the point (x,y), e.g., the average gray level over some neighborhood. Weszka [90] divided thresholding into three types depending on the functional dependencies of the threshold operator T. When T depends only on $f(x,y)$, the threshold is called global. If T depends on both $f(x,y)$ and $N(x,y)$, then it is called a local threshold. If T depends on the coordinate values x,y as well as on $f(x,y)$ and $N(x,y)$, then it is called a dynamic threshold.

There are a number of global threshold selection schemes. Some are based on the characteristic feature (e.g. gray level) histogram, others are based on local properties such as the gradient or Laplacian of an image. For an image consisting of object and background where the percent of the object area is known, Doyle suggested the "p-tile" method which chooses as a threshold the gray level which most closely corresponds to mapping at least (1-p)% of the gray levels into the object. If, for example, dark objects occupy 20% of the image area, then the image should be thresholded at the 80th percentile, or, more precisely, at the largest gray level allowing at least 20% of the pixels to be mapped into the object. This method is not applicable if the object area is unknown or varies from image to image. Dynamic thresholding is quite powerful in segmentation due to the fact that it allows the use of concepts such as proximity of points sharing a given property in order to separate objects from the background. Figure 2 shows an example of the segmentation results obtained by dynamic thresholding [56]. In this example, the images were thresholded to determine border points. A gradient operation was applied to the images to determine edges. The boundary of objects shown in Fig. 2 consist of points which passed both the border and edge tests. Thresholding was applied to both original and gradient images.

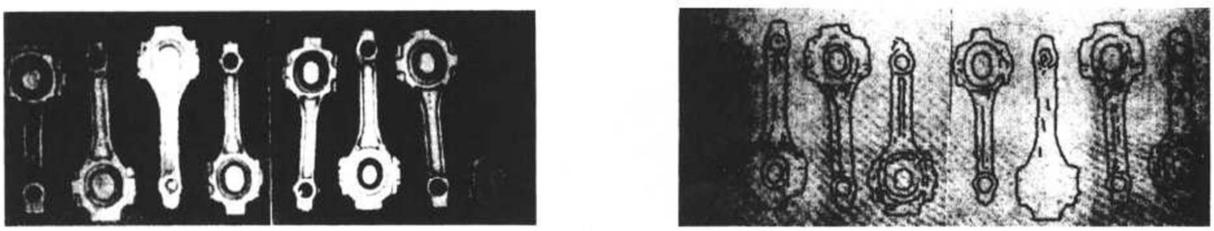

Fig. 2 Image of industrial parts and segmentation result (From Nakagawa and Rosenfeld [1979])

Clustering of characteristic features applied to picture segmentation is the multidimensional extension of the concept of thresholding. Typically, two or more characteristic features are used and each class of regions is assumed to form a distinct cluster in the space of these characteristic features. A clustering method is used to group the points in the characteristic feature space into clusters. These clusters are then mapped back to the original spatial domain to produce a segmentation of a picture. The characteristic features that are commonly used in picture segmentation by clustering not only include gray values through different filters, it may include any feature that one thinks is helpful to his segmentation problem; for example, texture measures defined on a local neighborhood may be used. The reason one wants to use two or more characteristic features to perform image segmentation is that sometimes there are problems not resolvable with one feature that can be resolved with two or more features [32].

(B) Edge detection

Edge detection is an image segmentation technique based on the detection of discontinuity. An edge or boundary is the place where there is a more or less abrupt change in gray level. To produce a closed boundary, the edge elements extracted have to be connected together to form a closed curve. Edge element extraction methods can be classified as (i) high-emphasis spatial frequency filtering, (ii) gradient operators, and (iii) functional approximation. Edge element combination consists of eliminating false edge elements and merging the edge elements into boundaries, and is generally carried out by three classes of techniques: (i) heuristic search, (ii) relaxation, and (iii) line and curve fitting. Many techniques incorporate edge element extraction as part of the process of edge or boundary detection, so there is no need to separate edge element extraction from edge element combination.

a) Edge Element Extraction

(a.1) High-Emphasis Spatial Frequency Filtering. Since high spatial frequencies are associated with sharp changes in intensity, so one can enhance or extract edges by performing high-pass filtering: i.e., take the Fourier transform of the picture, say $F(f(x,y)) = F(u,v)$ where $f(x,y)$ and $F(u,v)$ are the original gray level function and its Fourier transform respectively, F is the Fourier operator. Multiply F by the linear spatial filter $H : E(u,v) = F(u,v)\cdot H(u,v)$ and take the inverse transform $e(x,y) = F^{-1}(E(u,v))$ where $e(x,y)$ is the filtered picture of $f(x,y)$ and $E(u,v)$ its Fourier transform, and F^{-1} is the inverse Fourier transform operator. The problem here is filter design.

(a.2) Gradient Operators. The gradient operator is defined as

$$\nabla f(x,y) = \frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j} \tag{3}$$

where $|\nabla f(x,y)| = \left(\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right)^{1/2}$

and the direction of $\nabla f(x,y)$ is $\tan^{-1} \dfrac{\left(\frac{\partial f}{\partial y}\right)}{\left(\frac{\partial f}{\partial x}\right)}$

f is the original gray level function; $\hat{i}$ and $\hat{j}$ are unit vectors in the positive x and y directions respectively.

Quite a few proposed edge detection techniques are based on the digital approximations or variations of equation (3) which will produce a high magnitude where there is an abrupt change in gray level and a low magnitude where there is little change in gray level. Roberts' cross operator is based on a 2x2 window

$$g(i,j) = \left[(f(i,j) - f(i+1,j+1))^2 + (f(i+1,j) - f(i,j+1))^2\right]^{1/2} \tag{4}$$

where $f(i,j)$ and $g(i,j)$ are the gray level function and magnitude of gradient of point (i,j) respectively. The operator requires that there is a distinct change in intensity between two adjacent points in the gray value function, so only very sharp edges with high contrast between the surfaces which form the edges will be detected. This method cannot detect ill defined edges (edges which are formed by a gradual change in intensity across the edge). Since the computation is based on a small window, the result is quite susceptible to noise. Kirsch's, Sobel's, and Prewitt's operators are based on a 3x3 neighborhood. The main difference between these operators are the weights assigned to each element of the 3x3 template. An adaptive local operator was proposed by Rosenfeld et al. [68].

(a.3) Functional Approximations. Edge detection can be considered as an approximation problem. For every point (x',y') in an image, Hueckel used a circular neighborhood D about (x',y') and asked the question "Are the intensities (x,y) in D the noisy form of an ideal edge which is characterized by a step function?" Let

$$F(x,y,c,s,p,b,d) = \begin{cases} b & \text{if } cx+sy \leq p \\ b+d & \text{if } cx+sy > p, \end{cases} \qquad (5)$$

where the x-y co-ordinate system has its origin at the center of the circular region; F is the step function. The task of the operator is to best approximate a given empirical edge element whose gray values are $f(x,y)$ by an ideal edge element characterized by a step function F. As a measure of closeness, E (the square of the Hilbert distances between f and F) was chosen as

$$E = \;_{D}\,[f(x,y) - F(x,y,c,s,p,b,d)]^2 \, dxdy \qquad (6)$$

Hueckel's operator is an efficient solution to the minimization of E. The minimization procedure was facilitated by choosing orthonormal functions (e.g., Fourier functions) over D. The results of the minimization were the best edge and a measure of the goodness of the edge. This technique was later extended to detect lines.

Persoon's operator was defined over a window of size 5x5 pixels and the two columns to the left and to the right of the central one were approximated by linear functions. Deviations from the actual gray levels for the left and right linear function were computed and the right gradient (0°) was defined as a function of the two deviations and the average gray values corresponding to the left and right two columns. The picture was then rotated 7 times over 45° and seven additional gradients were computed. The maximum value of the 8 gradients was taken as an indication of the goodness of the edge which was perpendicular to the direction of the gradient. This edge detector solves some of the problems related to edge direction and noise but takes more computation time than some simpler edge operators.

b) Edge Element Combination (Streak or Boundary Formation)

(b.1) Heuristic Search and Dynamic Programming. Heuristic search is a technique using state space search methods where heuristic information is used to limit the space to be searched. Martelli formulated the edge detection problem as a heuristic search for the shortest path on a graph. The graph nodes (or states) were edge elements defined by two neighboring pixels, e.g., the points A=(i,j), B=(i,j+1) defined the directed edge element AB. The direction of the edge was obtained with the convention of moving clockwise around the first pixel. He then stated that an edge was a sequence of adjacent edge elements that started in the top row ended in the bottom row contained no loops and had no element whose direction was "up." So an edge was a path in the graph that represented the state-space and the problem of finding the best edge in a picture reduced to the problem of finding an optimal path in the graph. He then embedded properties of edges into an evaluation function and the edge which minimized this function was sought. Some of the drawbacks of this approach are that the algorithm is sequential in nature and the proposed approach does not provide for backtracking, so that once a mistake is made in the midst of the process the detected edge could be far off from the desired edge. The construction of a proper evaluation function is another problem.

Montanari proposed using dynamic programming techniques to perform edge detection. A figure of merit representing the heuristic information was used to determine the relative value of different paths but was not used to guide the search as in the heuristic search mentioned above. This figure of merit determined the best path once they had all been enumerated. Montanari discussed finding a smooth, dark curve of fixed length. The curve was embedded in a noisy background, but since the merit function did not guide the search, the computation time was independent of the noise level (which would not be the case if the merit function guided the search as in heuristic search).

(b.2) Relaxation. Rosenfeld and Riseman et al. used a relaxation technique to connect edge elements. The technique is an iterative process where the probability that a candidate edge element is a true edge element is estimated at each iteration. Some of the advantages of this approach are that it is a parallel process and it utilizes spatial information. Some of the disadvantages are that the construction of the compatibility function which updates the probabilities of edge elements is not trivial and the convergence rate of the process is sometimes slow.

(b.3) Line and Curve Fitting. Another technique of connecting edge elements together is to fit lines or curves through the edge elements. Duda and Hart [23] proposed an efficient solution to the Hough transform which is an ingenious way of detecting colinear points. Suppose we have a set of n points $\{(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)\}$ and we want to find a set of straight lines that fit them. We transform the points (x_i,y_i) into the sinusoidal curves in the θ-ρ plane defined by

$$\rho = x_i \cos\theta + y_i \sin\theta \tag{7}$$

It is obvious that curves corresponding to colinear points have a common point of intersection. This point in the θ-ρ space say (θ_0, ρ_0), defines the line passing through the colinear points. The implementation is to quantize the θ-ρ space into an array of cells and plot these sinusoidal curves on this array of cells. The number of curves that pass through every cell in the array is recorded. If the count in a given cell (θ_i, ρ_i) is k, then precisely k figure points lie (to within quantization error) along the line whose normal parameters are (θ_i, ρ_i). The Hough transform concept can be extended to curves. Some of the limitations are that the results are sensitive to the quantization of both θ and ρ, and the technique finds colinear points without regard to continuity. Thus the position of a best-fit line could be distorted by the presence of unrelated points in another part of the picture.

There are other techniques of edge detection such as template matching which can be applied not only in edge detection but in other areas as well, e.g., object extraction. Template matching works well in a very constrained environment but fails where there is great variation of the patterns to be matched.

(C) Region Extraction

Another way of doing image segmentation instead of finding boundaries of regions, is to divide the image into regions. Region extraction techniques can be broken down into three categories, 1) region merging, 2) region dividing, and 3) a combination of region merging and dividing.

(1) Region Merging. Muerle and Allen used regional neighbor search method to merge regions of similar properties. Brice and Fennema [17] formed connected components of equal intensity, refined with some heuristics. Pavlidis [58] partitioned a picture into a collection of 1-dimensional strips, divided the strips into segments and merged the segments with similar approximation coefficients. Feldman and Yakimovsky used semantics to do region merging. They tried to maximize the probability that all regions and borders were correctly interpreted. Rosenfeld et al. used a relaxation approach, also called iterative probabilistic process to do scene labelling.

Tenenbaum and Barrow proposed IGS (Interpretation Guided Segmentation) as an approach to region merging. The program iteratively processed the scene until its components are semantically consistent. Gupta and Wintz used a minimum distance classifier which interpreted each initial region as belonging to one of a small predetermined number of different classes. Neighboring regions were merged based on their class membership. Jarvis used a shared near neighbor clustering technique to do region merging. Tsuji and Fujiwara used linguistic techniques to perform region merging.

(2) Region Dividing. One way of doing picture segmentation by region extraction is the region dividing approach. Robertson et al. used a mean vector of gray levels of a multispectral image to perform region dividing. Klinger proposed to use regular decomposition for image segmentation

(3) Region Merging and Dividing. Horowitz and Pavlidis approached the problem using a "split and merge" principle. Regions were described in terms of an approximating function. They merged adjacent regions having similar approximations and split those regions that had large approximating errors.

IV. Description

The description problem in a robot vision system is one of extracting important properties or features for recognition purpose. General image properties include shape, texture and color. Descriptors for machine part recognition are however primarily based on shape information. There are three basic approaches to shape description. Both the skeleton and the (outer) boundary of an object can be used to represent its shape. Ideally speaking, the shape descriptors selected should be independent of translation, rotation and scaling and contain discriminating information for shape recognition.

(A) Functional shape descriptors

Moments and Fourier descriptors have been suggested as shape descriptors [40, 63,100]. Let C denote the (outer) boundary of an object, which is a simple closed contour with representation $(x(\ell), y(\ell)) = C(\ell)$ where ℓ is the arc length along the contour. A point moving along the boundary generates the complex function $u(\ell) = x(\ell) + jy(\ell)$ which is periodic with period L. The Fourier expansion of $u(\ell)$ can be written as

$$u(\ell) = \sum_{-\infty}^{\infty} a_n e^{jn(2\pi/L)\ell}$$

with the Fourier descriptors

$$a_n = \frac{1}{L}\int_0^L u(\ell)e^{-j(2\pi/L)n\ell}d\ell$$

A truncated sequence $\{a_n | n = -M,\ldots,0,1,\ldots,M\}$ can be used as shape features for recognition purpose. The surface area bounded by the object contour and its skeleton can be computed from a given set of Fourier descriptors [63].

Chen and Shi† suggest the following shape features in terms of Fourier descriptors:

(1) Roundish degree

$$F_1 = |a_1| \;/\; \sum_{n=1}^{\infty} (|a_n| + |a_{-n}|)$$

when $F_1 = 1$, C is a circle; $0 < F_1 < 1$ otherwise.

(2) Slim degree

$$F_2 = 2|a_{-1}| \;/\; (|a_1| + |a_{-1}|)$$

$F_2 = 0$ when C is a circle and $0 < F_2 < 1$ otherwise.

(3) Diffusion

$$F_3 = \frac{(\text{perimeter})^2}{4\pi\ (\text{AREA})}$$

where

$$\text{AREA} = \pi \sum_{-\infty}^{\infty} n|a_n|^2$$

(4) Concavity

$$F_4 = \sum_{n=1}^{\infty} n^3(|a_n|^2 - |a_{-n}|^2) \;/\; (|a_1|^2 + |a_{-1}|^2)$$

$F_4 = 1$ when C is a circle and $F_4 < 1$ when C is concave. For other cases, $F_4 > 1$.

(B) Heuristic shape descriptors

Many intuitively appealing measurements have been proposed as shape descriptors. Agin and Duda [2] have used the following shape features for recognition of foundry castings:

x_1 = perimeter of figure
x_2 = square root of 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)

Both Agin and Duda and Chen and Shi use a tree classifier for shape recognition based on their suggested shape features.

† Proc. 5th Int'l. Conf. Pattern Recognition, 1980.

(C) Syntactic shape description

The basic idea of syntactic or structural pattern recognition is to represent a pattern in terms of its components and the relations among them. The simplest components are called primitives [27,29,42,43]. For shape description in terms of object median curve or boundary, straight line segments or curve segments are often suggested as primitives. Length, slope and curvature can be used as the attributes of the primitives. The median curve or contour of an object is then represented as a sequence of the primitives. A set of structural or syntax rules can be inferred to characterize the structural interrelationships of these sequences (or strings of primitives) describing the objects of interest.

Vamos [87] has proposed to use a context-free grammar for building machine parts from picture primitives. As shown in Fig. 3(a), the grammar has four primitives: straight line, arc, node and undefined. These primitives are then assembled by the syntax rules into generalized picture primitives, as shown in Fig. 3(b), and further refined into final objects. A typical sequence of steps is shown in Fig. 4. Using segments of straight lines or curves as primitives, Jakubowski [43] has suggested the use of extended context-free grammar to characterize machine parts shapes. Regular right part grammars are employed to generate contours of machine parts.

A method recently proposed for syntactic shape recognition is the use of attributed grammars [59,81,98,99]. In this method, a primitive is define by a symbol and its associated attributes. The rules governing the construction of the objects from the primitives consists of syntax rules which provide the basic structural description as well as semantic or attribute rules which assign meaning to that description.

V. Recognition and Interpretation

Recognition of a pattern usually means to assign the pattern to a particular class. With the additional structural and semantic information, an interpretation of the pattern (or a scene) can often be made. There are three major approaches to pattern recognition: (i) template-matching, (ii) decision-theoretic or statistical approach, and (iii) structural or syntactic approach. They are briefly reviewed in this section.

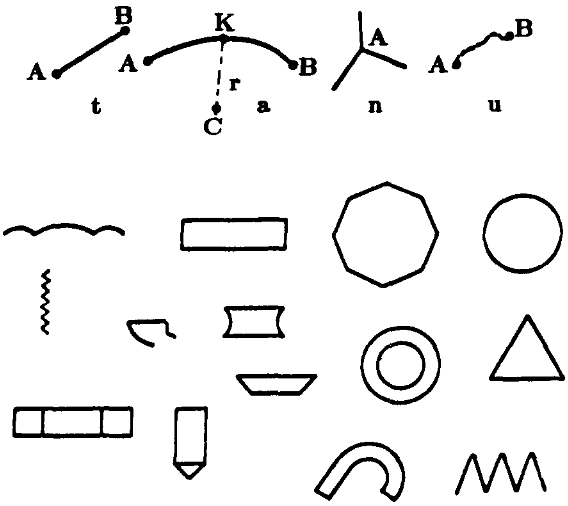

Fig. 3 (a) Picture primitive, and (b) Generalized picture primitive (From Vamos [1977])

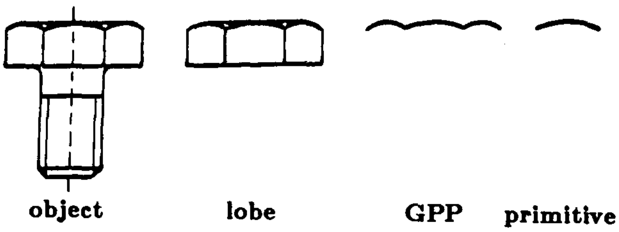

Fig. 4 Picture hierarchy (From Vamos [1977])

(A) Template-matching

In the template-matching approach, a set of templates or prototypes, one for each pattern class, is stored in the machine. The input pattern with unknown classification is matched or compared with the template of each class, and the classification is based on a preselected matching criterion or similarity measure (e.g., correlation). In other words, if the input pattern matches the template of ith pattern class better than it matches any other templates, then the input pattern is classified as from the ith pattern class. Usually, for the simplicity of the machine, input patterns and the templates are represented in their raw-data form, and the decision-making process is nothing but matching the unknown input to each template. The disadvantage of this approach is that it is sometimes difficult to select a good template for each pattern class, and to define an appropriate matching criterion. This difficulty is especially remarkable when large variations and distortions are expected in the patterns under study. Recently, the use of flexible template-matching or "rubber mask" techniques has been proposed.

(B) Decision-theoretic approach

In the decision-theoretic approach, a pattern is represented by a set of N features or an N-dimensional feature vector, and the decision-making process is based on a similarity measure which, in turn, is expressed in terms of a distance measure or a discriminant function. In order to take noise and distortions into consideration, statistical and fuzzy-set methods have been proposed. The characterization of each pattern class could be in term of an N-dimensional class-conditional probability density function or a fuzzy set, and the classification (decision-making) of patterns is then based on a (parametric or nonparametric) statistical decision rule or (fuzzy) membership function.

A block diagram of a decision-theoretic pattern recognition system is given in Figure 5.

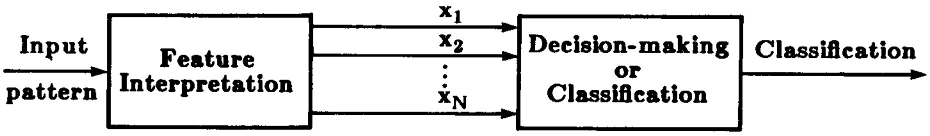

Fig. 5 Block diagram of decision-theoretic pattern recognition system

The feature vector $X = [x_1,x_2,...,x_N]$ representing the input pattern can be classified according to the discriminant functions $D_1(X)$, $D_2(X)$,..., $D_m(X)$ where m is the number of possible pattern classes. if $D_i(X) = \max_{k=1,...,m} \{D_k(X)\}$ then X is classified as from the ith class. Linear, piecewise linear and quadric discriminant functions are often used in practice. When a set of prototype or reference patterns can be selected, one for each class, we can use the minimum-distance classification rule. Let $\{R_1,R_2,...,R_m\}$ be the set of reference patterns where R_i is the reference feature vector for the ith class. Calculate the distance between an input feature vector X and R_k, $k=1,...,m$. X will be classified as in the same class as R_i if the distance between X and R_i is the smallest.

Sometimes, a tree classifier can be used for efficient classification. At each node of the tree classifier, only a small number of features needs to be used [28].

(C) Structural or syntactic approach [29]

In the structural or syntactic approach, a pattern is often represented as a string, a tree or a graph of pattern primitives and their relations. A set of structural or syntax rules (or a grammar) is used to characterize the pattern structure and to provide a compact representation. The decision-making process is in general a syntax analysis or parsing procedure. Special cases include the use of similarity (or distance) measures between two strings, two trees, or two graphs. A block diagram of a structural/syntactic pattern recognition system is given in Figure 6.

Conventional parsing requires an exact match between the unknown input sentence and a sentence generated by the pattern grammar. Such a rigid requirement often limits the applicability of the syntactic approach to noise-free or artificial patterns. Recently, the concept of similarity measure between two sentences and between one sentence and a language has been developed. Parsing can be performed using a selected similarity (a distance measure or a likelihood function), and an exact match becomes unnecessary. Such a parsing procedure is called "error-correcting" parsing.

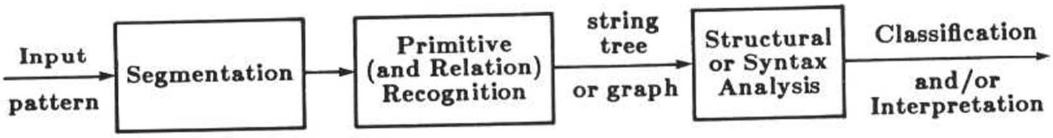

Fig. 6 Block diagram of syntactic pattern recognition system

VI. Application Examples

Two application examples are given in this section for illustrative purpose.

Example 1: The SRI vision system [2] for the recognition of foundry castings uses a diode array to obtain binary image of the parts. Since a given part can appear in a number of different stable states, and since each stable state typically presents a different image, it is treated as a different part to be recognized. Thus, even if only one kind of part will present, the system still has to solve a multi-class recognition problem, with one class for each stable state.

The seven shape features $x_1, x_2, \ldots, x_7$ described in Section IV (A) are extracted from the part image (Fig. 7). A binary tree classifier, as shown in Fig. 8, is design for recognition.

Fig. 7 Images of foundry castings and their boundaries (From Agin and Duda [1975])

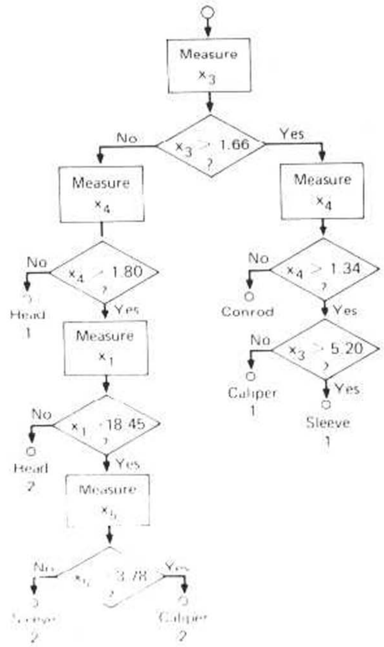

Fig. 8 Tree classifier for recognition of foundry castings (From Agin and Duda [1975])

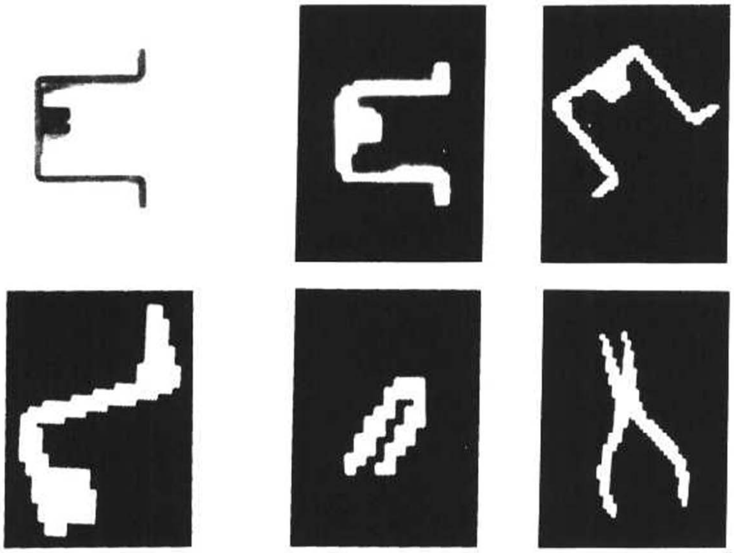

Fig. 9 Industrial parts and their thresholded images (From Persoon and Fu [1977])

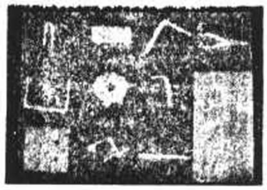

Fig. 10 Ten classes of industrial parts (From Persoon and Fu [1977])

Example 2: Fourier descriptors have been suggested for recognition of industrial parts [63]. The visual information is obtained for a TV camera interfaced to a mini-computer. Each part image is digitized into a 60x60 array of pixels with 128 gray levels (7 bits). The gray level histogram of this array is computed and used to determine a threshold. This threshold allow us to convert the image array into a binary array representing the silhouette of the part (Fig. 9).

The Fourier descriptors (15 harmonics) are then 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 ten classes of industrial parts (Fig. 10). Overlapping parts can be detected as follows: either the boundary obtained from overlapping parts does not match closely with any one of the reference or training patterns or, in case it matches closely, the area of the silhouette is not as expected from a single part. In such a case, a robot arm will try to separate the parts.

VII. Concluding Remarks

We have briefly reviewed major robot vision techniques for machine part recognition. The principal motivation of having vision for robot is increased flexibility and lower cost. Due to speed requirement in a real time manufacturing process, at present only very simple techniques have been actually applied. More sophisticated techniques can certainly be applied to the problems such as analysis of part image sequence and bin-picking [12]. Implementation of robot vision algorithms on microprocessors and VLSI architectures should be investigated particularly from the viewpoint of cost-effectiveness.

BIBLIOGRAPHY

1. Agin, G. L. [1980]. "Computer Vision Systems for Industrial Inspection and Assembly," Computer, vol. 13, no. 5, pp. 11-20.
2. Agin, G. L. and Duda, R. O. [1975]. "SRI Vision Research for Advanced Industrial Automation," Proc. 2nd USA-Japan Computer Conference, Aug. 26-28, Tokyo, Japan.
3. Agrawala, A. K. and Kulkarni, A. V., [1977]. "A Sequential Approach to the Extraction of Shape Features," Computer Graphics and Image Processing, vol. 6, pp. 538-557.
4. Alt, F. L. [1962]. "Digital Pattern Recognition by Moments," J. ACM, vol. 9, pp. 240-258
5. Argentiero, P. et al. [1982]. "An Automated Approach to the Design of Decision Tree Classifiers," IEEE Trans. PAMI, vol. PAMI-4, no. 1, pp. 51-57.
6. Baird, M. [1977]. "Image Segmentation Technique for Locating Automotive Parts on Belt Conveyors," 5th Int'l Joint Conf. Artificial Intelligence, Cambridge, Ma, pp. 694-695.
7. Bejczy, A. K. [1976]. "Issues in Advanced Automation for Manipulator Control." Proc. Joint Automatic Control Conf., pp. 700-711.
8. Belforte, G. et al. [1980]. "Some New Systems for Recognition and Positioning of Mechanical Parts," Proc. 10th Int'l Symp. on Ind. Robots, pp. 203-214.
9. Bernasconi, M., Delazer, R., and Gini, M. [1980]. "Precision Measurement by Stereo Vision System," Proc. 10th Int'l Symposium on Industrial Robots, pp. 349-360.