A Vision System for an Automatic Assembly Machine of Electronic Components

BUM-JAE YOU, MEMBER, IEEE, YOUNG SEOK OH, AND ZEUNGNAM BIEN, MEMBER, IEEE

Abstract—**A vision system for automatic assembly of electronic components is developed. The vision system presents information about positions, orientations, and quality of rectangular-shaped electronic components in real time. The orientation is detected by the windowed Hough transform along with a simple edge-detection method, whereas the position of each component is determined by employing the projection method with dynamic thresholding. In addition, real-time implementation of the vision system is described in which multiple central processors are employed for parallel processing of the algorithms.**

I. INTRODUCTION

SINCE the early 1970's, vision has been widely used for automatic location, classification, and inspection of parts and components in electronic manufacturing [1]–[2]. Examples can be found easily in manufacturing facilities for semiconductors such as the die bonder, wire bonder, and wafer inspection machine. In these systems, the vision system determines the posture of a component and inspects each component in real time [3]–[4] before automatic assembly would take place. Usually, such a vision system includes cameras and microscopes as sensing devices and video-rate image processors and central processors as processing hardware together with various pattern recognition algorithms, but it should be noted that the hardware and software of the vision system in industrial use are often configured to be dependent on specific applications in consideration of cost and processing time.

In the literature, various kinds of pattern-recognition algorithms are reported for vision processing. However, some of these known methods are not appropriate for a vision-based automation system in the factory manufacturing environment in which real-time aspects and reliability are two critical issues of the system design. To be more specific, well-known methods in [5]–[9] may be referred to for detecting the posture, that is, the position and its orientation, of an object. Hu [5] utilized the concepts of central moments and principal axis of an object by analyzing the global shape of the object. Perkins [6] used a group of concurves such as straight lines and circular arcs of an object and matched the concurves with image models to locate and classify the object. Yachida and Tsuji [7] and Dessimoz [8], respectively, adopted the polar representation of boundaries with respect to the centroid of the object and a curvature function of boundaries of the object. Under several hypotheses based on *a priori* information about objects, Ayache and Faugeras [9] used polygonal approximation for a compact description of object boundaries. It should be noted that these methods can be effective for detecting postures of complicated industrial parts in automatic manufacturing cells such as gasoline engines, pumps, bearings, or tools [10]–[12], but they are computationally inefficient because time-consuming preprocessing must be conducted to find global area, momentum, boundary points, a curvature function, concurves, and approximated polygonal or recognition hypotheses of an object. On the other hand, it should also be noted that although there are numerous types of parts or components in general, the task of the vision system is often specifically limited to recognizing the posture of objects of very simple shape (as in electronic manufacturing factories where, naturally, quick recognition is most desirable for enhancing productivity). In this respect, those methods reported in [5]–[9] may not be suitable for high-speed assembly in which the objects to be recognized are simple in shape and arranged in regular fashion, as in automatic assembly machines of microelectronic components [13]–[14] such as a die bonding machine for semiconductor chips. To our knowledge, most of those available methods have not yet been tested in a practical setting because of real-time operation and reliability as well as from a system operation point of view.

In this paper, our experiences in developing a visual pattern-recognition system for assembly of small electronic parts are reported. The vision system is designed to perform automatic assembly tasks with a fast processing capability. We shall first present pattern-recognition softwares that are relatively fast and insensitive to noise. These algorithms are developed by employing the Hough transform technique [15] together with the projection method and modified edge preserving smoothing algorithms [16]. Next, we shall describe the structure of the vision system and, in particular, a new type of video-rate frame grabber as the image memory for one frame, which is designed by employing dual-ported video RAM's [17]. We shall also briefly comment on an on-line processor, including digital binarization circuits and a window counter specially designed to process images in real time. Finally, we shall discuss experimental results for illustration.

II. PATTERN-RECOGNITION ALGORITHMS

Our vision system is designed for automatic assembly machines of small simple-looking electronic components such as rectangular-shaped integrated-circuit chips with quality marks, e.g., a die bonding machine [13] or something similar. The die bonding machine transfers good dies from a wafer to a

Manuscript received May 16, 1989; revised March 23, 1990.

The authors are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Seoul, Korea.

IEEE Log Number 9038042.

0278-0046/90/1000-0349$01.00

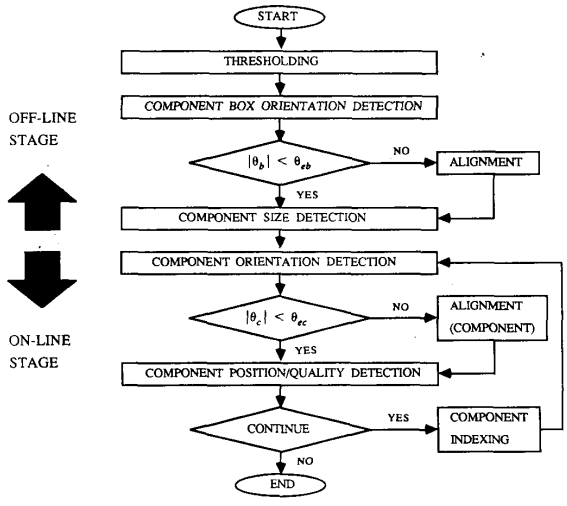


Fig. 1. Pattern recognition flowchart for an automatic assembly machine.

sequence of lead frames one at a time while conducting automatic correction of positional deviation and orientation error. In a modern electronic wristwatch maker, one may find a die attachment machine that is used for attaching integrated-circuit watch chips onto a printed circuit board for an electronic watch. In this latter case, the machine is fed with rectangular components contained in a regular array, which is called a waffle pack.

In order to conduct recognition of these types of components more efficiently, we divide the image processing into two stages, i.e., one for off-line processing to determine process parameters with high precision and the other for on-line processing for fast repetitive recognitions, as is described in Fig. 1.

In the off-line stage, a threshold level is determined to obtain binary images based on the concept of a separability function [18]. Then, the orientation of the component container (e.g., wafer or waffle pack) is determined. In addition, to use the windowing technique for fast image processing in the on-line stage, the size of a typical component is calculated accurately.

In the on-line stage, the position and orientation of a component are determined while quality inspection of the component is simultaneously executed in real time repeatedly.

Our pattern-recognition algorithms for the orientation and position of a component are derived based on the following two assumptions:

1) Each component is rectangular-shaped.
2) If the size of a component is $S_x \times S_y$ and the positional deviations of the component relative to the origin of the image plane are denoted as D_x and D_y along x and y axes, respectively, then

$$-S_x/2 < D_x < +S_x/2$$

$$-S_y/2 < D_y < +S_y/2.$$

It is noted that the second assumption is given to determine the position of recognition window and thus speed up the recognition time. The size of the window depends on the amount of positional deviation of a component with respect to a reference point where the positional deviation is mainly caused by the feeding mechanism of the machines. Thus, the assumption is realizable since the positional deviations of the feeding mechanism can be made much smaller than the size of the component in high-precision assembly machines such as the die bonder, the die attachment machine, or an automatic insertion machine for parts.

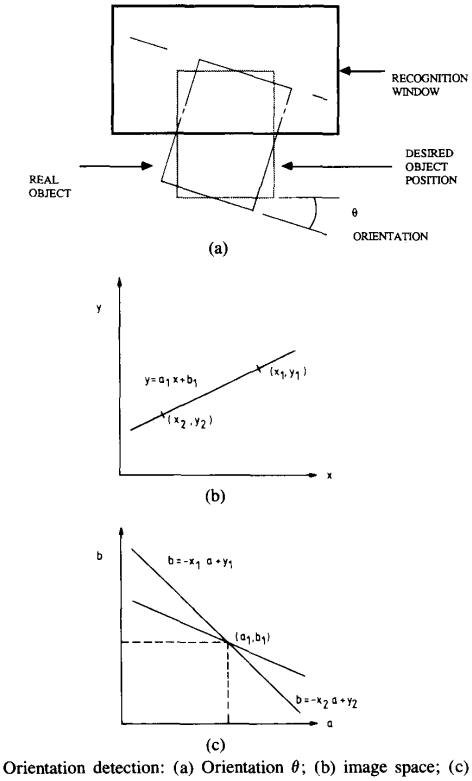


Fig. 2. Orientation detection: (a) Orientation θ; (b) image space; (c) parameter space.

A. Orientation Detection

The boundary of a rectangular-shaped component consists of several line segments. Thus, the orientation of the component can be determined by calculating the line equation for a major segment among the boundary line segments. The orientation is calculated from the slope of the line.

It is noted that the image of the boundary of an object is often more sensitive to noise and that, as depicted in Fig. 2(a), there are, in a recognition window, not only main edge points made by the major segment but also edge points formed by other nondominant boundary segments. Hence, to find the equation of the major line segment, the main edge points must be separated from all the other edge points in the recognition window, and if necessary, a compensation for the major line must be performed as in the case that the line is scattered in pieces caused by the nondominant edges. The recognition window is set up on the upper region of the component in

order to reduce the processing time. We propose that the corresponding line equation is obtained via the Hough transform method [16] since the method is relatively robust to nondominant edges, image noise, and gaps in curves. It is remarked that the least-square-error method [19] is found to be unsuitable for the above task of orientation detection due to its rather high sensitivity to noise.

The Hough transform method finds an equation of a line by converting each pixel in the image space into a line in the two-dimensional parameter space with coordinates that denote the slope and the y-intercept value of the straight line. For example, suppose that a line L in the x-y image space is given and described by the line equation of $y = a_1 \cdot x + b_1$, as is shown in Fig. 2(b). Then, the line L is mapped to the set of lines $\{b = -x_i \cdot a + y_i | (x_i, y_i) \in L\}$, all of which pass through the point (a_1, b_1) in the a–b parameter space, as is shown in Fig. 2(c). In the case in which the line L in the image space is contaminated by noise, the Hough transform renders useless a group of clustered points with scattered points in the a–b space. The slope of the line is obtained by identifying the most clustered point through which most of the lines pass through in the parameter space.

To use the Hough transform for determining the orientation of a component in the machine, several preparatory actions should be taken for repetitive operations. The first step is to specify the range of the slope a and that of the y intercept b of the boundary. This means that we input to the machine the data A_m with $0 < A_m < 1$ and $\{B_L, B_H\}$ with $-S_y/2 < B_L < B_H < +S_y/2$ so that only $-A_m < a < A_m$ and $B_L < b < B_H$ are handled in the machine.

The second step is to quantize the slope a and the y intercept b to construct a digitized parameter space. In this step, the resolution of orientation is determined, that is, if the slope and the y intercept are equally divided into N_a and N_b parts, respectively, the digitized parameter space is represented as (a_i, b_j), $0 \le i \le N_a$, $0 \le j \le N_b$, and the accumulator array for the parameter space, in which each array cell is a kind of counter indicating the number of clustered points at (a_i, b_j), is denoted as $A(a_i, b_j)$, $0 \le i \le N_a$, $0 \le j \le N_b$. In addition, the resolutions of the slope and the y intercept Δa and Δb become the following:

$$\Delta a = 2 \cdot \frac{A_m}{N_a}$$

$$\Delta b = \frac{B_H - B_L}{N_b}. \tag{1}$$

Here, the values N_a and N_b are specified in consideration of operational requirements of the automatic assembly machine. In addition, if the depth of an accumulator array cell is one byte, we find that the size of the memory, which is M_s for the parameters $|a| < A_m$ and $B_L < b < B_H$, becomes

$$M_s = N_a \cdot N_b \text{ (bytes)}. \tag{2}$$

The final preliminary action to take is to set the recognition window for fast processing. Let the range of the slope of the line in the angle be $|\theta| < \theta_m$. Then, the size of the recognition window $W_x \times W_y$ becomes

$$W_x = 2 \cdot S_x$$

$$W_y = 2 \cdot (S_y/2 + S_x \cdot (\tan \theta_m)/2). \tag{3}$$

After these pre-steps are executed, the following algorithm, which is known as the windowed Hough transform, is applied repeatedly in the coordinate frame with its origin at the center of the recognition window.

Step 1) Initialize all the accumulator array cells $A(a_i, b_j)$, $0 \le i \le N_a$, $0 \le j \le N_b$.

Step 2) Input a binary image.

Step 3) Find candidates for the main edge points of the component in the recognition window. We say that a candidate for the edge points is found at $(x, y + 1)$ if

$p(x, y) = 1$ (background) and $p(x, y + 1) = 0$ (components)

for

$$W_{ix} < x < W_{ix} + W_x \quad \text{and} \quad W_{iy} < y < W_{iy} + W_y$$

where

$p(x, y)$	pixel value at (x, y) in image plane
x	column address in image plane
y	row address in image plane.

Step 4) Transform the candidates for the main edge points in the image plane into lines in the parameter space, that is, increase the value of $A(a_i, b_j)$ by 1 whenever as line $b = -a \cdot x + y$ passes through a point (a_i, b_j) in the parameter space.

Step 5) Find an accumulator array cell $A(a_m, b_m)$ that contains the maximum value among the array $A(a_i, b_j)$, $0 \le i \le N_a$, $0 \le j \le N_b$. Then, the orientation θ of the component can be determined to be

$$\theta = \arctan(a_m).$$

B. Position Detection

Once the orientation of a component is identified and corrected to be in line with the image coordinate frame, the center of a component can now be found from the positions of the upper and left boundaries of the component. To detect the boundaries, the projection method is applied to the windowed binary images, as is shown in Fig. 3. In Fig. 3, the window that is depicted as a dotted line denotes the desired position at which the component is placed. The window that is depicted as a solid line represents the real component, and the recognition window is shown in bold solid line.

The coordinate of the upper left corner on the recognition window W_{ix} and W_{iy} is taken to be

$$W_{ix} = 128 - S_x$$

$$W_{iy} = 128 - S_y$$

where S_x and S_y are the horizontal and vertical magnitudes in pixel numbers of the component. In addition, the size of

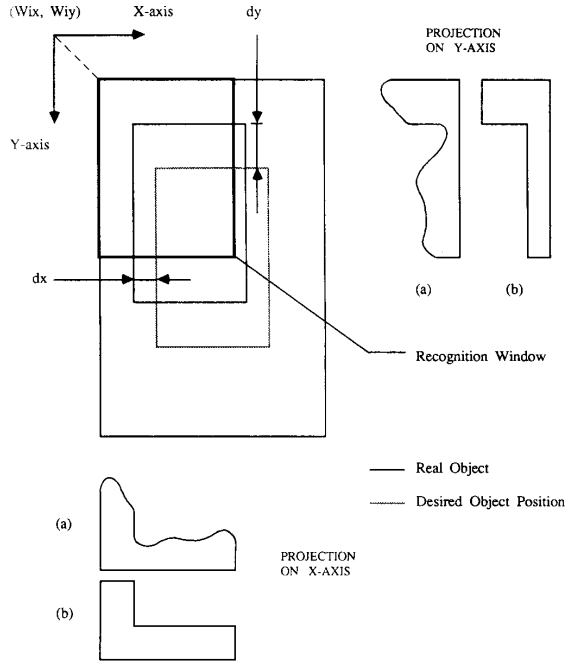


Fig. 3. Position recognition windows.

the recognition window W_x and W_y is determined in such a way that

$$W_x < 1.5 \cdot S_x$$
$$W_y < 1.5 \cdot S_y \tag{4}$$

respectively.

Since projection in the image plane is equivalent to integration of the intensity of all the pixels in a specified direction, the effect of noise due to quantization and incomplete threshold is naturally minimized. When we use other coding methods of binary images, such as chain coding or run-length coding techniques, it should be noted that two adjacent components may be misjudged to be one component if any noise exists between the two components. In addition, the processing time of the projected image is reduced since two dimensional images are converted into one dimensional array data. Further, any markings on the component can be easily distinguished from the boundary of the component if the projection is adopted. The proposed algorithm is as follows:

Step 1) Input a binary image.

Step 2) Project the binary image in the recognition window on the x and y axes of the image plane, respectively, to obtain

$$P_x(k) = \sum_{i=W_{iy}}^{W_{iy}+W_y} p(k, i), \quad W_{ix} \le k \le W_{ix} + W_x$$

$$P_y(k) = \sum_{i=W_{ix}}^{W_{ix}+W_x} p(i, k), \quad W_{iy} \le k \le W_{iy} + W_y.$$

The above projections that are graphically shown in Fig. 3(a), are modified by using the smoothing algorithms of [17], and the projections in Fig. 3(b) are obtained.

Step 3) Find differences of the projections as follows:

$$DP_x(k) = P_x(k+1) - P_x(k), \quad W_{ix} \le k \le W_{ix} + W_x$$

$$DP_y(k) = P_y(k+1) - P_y(k), \quad W_{iy} \le k \le W_{iy} + W_y.$$

Step 4) Find two sets $T_{x\min}$ and $T_{y\min}$, whose elements are indices k_i and k_j, indicating that $DP_x(k_i)$ is one of the x_n lowest values and that $DP_y(k_j)$ is one of the y_n lowest values, respectively, where $3 \le x_n \le 5$ and $3 \le y_n \le 5$:

$$T_{x\min} = \{k_{x_1}, k_{x_2}, \cdots, k_{x_n}\}$$

$$T_{y\min} = \{k_{y_1}, k_{y_2}, \cdots, k_{y_n}\}.$$

Step 5) If $P_x(k_{x_i}) > S_x - P_{ex}$ and $P_y(k_{y_j}) > S_y - P_{ey}$, where

$$0 < P_{ex} < 0.1 \cdot S_x, \quad k_{x_i} \in T_{x\min}, \quad 0 \le x_i \le x_n$$

$$0 < P_{ey} < 0.1 \cdot S_y, \quad k_{y_i} \in T_{y\min}, \quad 0 \le y_j \le y_n$$

then go to *Step 6)*. Otherwise, remove k_{x_i} or k_{y_j} from the sets $T_{x\min}$ and $T_{y\min}$, and repeat *Step 5)*.

Step 6) Determine the points

$$(k_{x_i} + 1, B_y), \quad W_{iy} \le B_y \le W_{iy} + S_y$$

$$(B_x, k_{y_j} + 1), \quad W_{ix} \le B_x \le W_{ix} + S_x$$

as a horizontal and vertical boundary, respectively.

It is remarked that when the size of any markings on the component is varying, a fixed threshold in *Step 5)* does not distinguish the boundary from the markings on the component, and for this case, the differences of projections in *Step 3)*, which are known as the dynamic thresholding method, are used.

III. Experiments

In order to test the real-time aspect and robustness of the proposed pattern recognition algorithms in a realistic assembly machine, we have conducted various experiments on the vision hardware we designed by using images of a wafer in the die bonding machine [13]. In the machine, the vision system is presented a die of the wafer one at a time and performs the tasks of detecting the position, orientation, and quality of the die within less than 0.4 s for real-time system operation. Next, if an inkdot (a quality mark) does not exist on the die, the system controller moves the pickup arm over the die, picks up the die, and bonds the die on a lead frame. An inkdot on the die is marked in the pretesting stage and represents bad quality of the die.

A. Hardware Structure of Vision System

The structure of the developed vision system is shown in Fig. 4. To transfer image data at the video rate (6.9 MHz) between processors, three 8-b vision buses in VME-I/O-BUS are incorporated.

To reduce the image data transfer time and to endow the

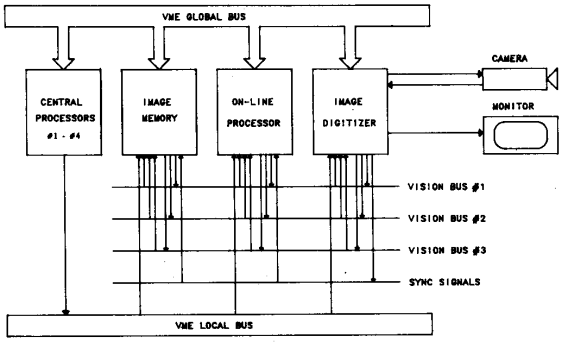


Fig. 4. Structure of designed vision system.

central processors with real-time pattern recognition capability, the image memory must be able to store the image data at the video rate and also be used as the system memory to which the central processors can be accessed whenever it is desired. For this, memory components with two channels (one for parallel interface and the other for high-speed serial interface for data transfer) are needed; therefore, a commercially available dual-ported dynamic random access memory (DRAM), which is known as a TMS4161 [15], is adopted.

It is noted that some difficulties exist in using the TMS4161 because there are three different types of addressing required by the central processors as well as refreshment, and a line address generator is also required. Those difficulties were circumvented by employing a 8203 DRAM controller, which can arbitrate the addressing requirements by refreshment and via the central processors. Further, to design the multiplexer and arbiter for the addressing requirement of the central processors and line address generator, a 26 to 16 multiplexer was used, and the necessary logic circuits for the arbiter were devised by considering the arbitration algorithm and by utilizing the 82S129 programmable read only memory (PROM). A detailed description is shown in Fig. 5.

For on-line processing, various on-line processing algorithms such as video-rate binarization of the gray level images, projection of the windowed binary images for position detection, or white/black pixel counting for quality inspection are implemented by hardware circuits for fast data processing. This specific hardware, which is composed of digital comparators, window generators, pixel counters, and an interrupt requester is called upon to be the on-line processing unit in the system.

The image digitizer, which is shown in Fig. 6(a), converts analog video signals into 256 × 256 (64K) byte digital gray-level images at a 6.9 MHz sampling rate and generates analog video signals from processed digital images. Moreover, a unit of generating synchronization signals for video signals is included in conjunction with a phase-locked loop unit. The signal specifications are shown in Fig. 6(b).

The system further includes four MC68000-based central processors with MC68881 coprocessors for controlling the overall system, performing the pattern recognition, and parallel processing of the pattern recognition algorithms by using VME-BUS.

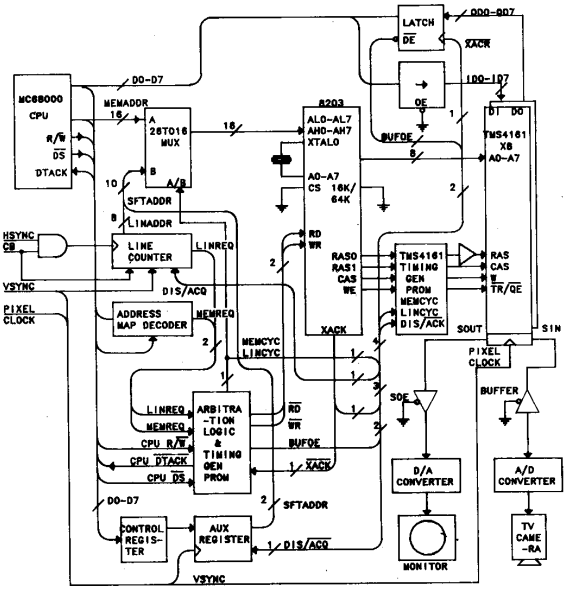
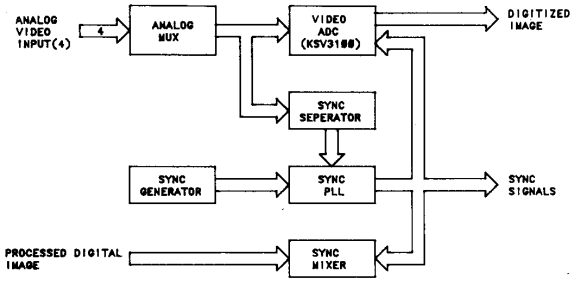


Fig. 5. Dual-ported DRAM controller.



(a)

Sync Signal	Freq.	
Pixel Clock	6.9	MHz
H - Sync	15.58	KHz
V - Sync	56.6	Hz

(b)

Fig. 6. Image digitizer: (a) Functional block diagram; (b) frequency of pixel clock and synchronization signals.

B. Experimental Results

For image formation in the die-bonding machine, we noticed that normal illumination, typically measured as 400 lx, is found to be too dark to obtain good images. Moreover, as the magnification ratio of the microscope increases, the scope of the camera becomes narrower, and the lighting condition becomes darker. To alleviate such a troublesome lighting condition, we adopted with care an optical system with a CCD camera, which is supplemented with two fiberoptic light sources with two lenses and a microscope with 0.5–20 magnification. The lenses are needed for concentrating, without blurring effect, the fiberoptic lights that can be brighter than 3000 lx.

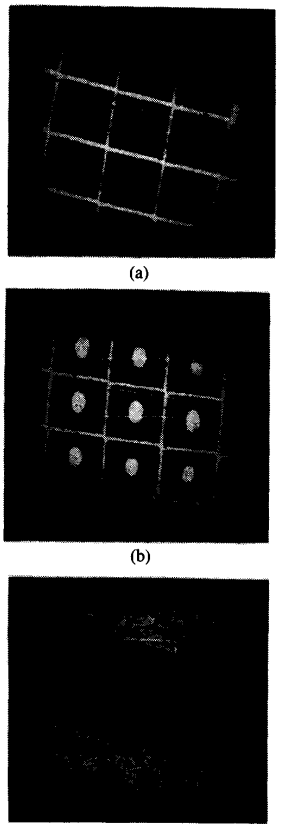
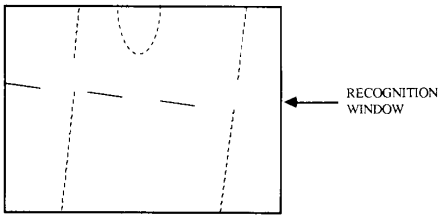


Fig. 7. Orientation recognition results: (a) Die with no inkdot; (b) die with inkdot; (c) die on noisy mylor.



----- Vertical Edges by Non-dominant Boundary

—— Upper Edges of a Die

Fig. 8. Detected edges in the image in Fig. 7(b).

In the experiment in which the orientation of components was detected, we considered three types of samples: a die with an inkdot, a die with no inkdot, and a die placed on a mylor that represents a noisy background, as is shown in Fig. 7. In the figure, the line depicted within each window represents the detected slope of a die by using the proposed windowed Hough transform method.

Examining the edges in the recognition window, one may find that there are upper edges of a die as well as nondominant edges formed by noise, inkdots, and other boundary segments, as is shown in Fig. 8. The edges caused by noise and inkdots can destroy the upper edges of the die and render cleft line segments useless. Therefore, in order to detect the orientation of the die correctly, it is important to extract only the upper edges from all the other nuisance edges in the window and perform compensation for the gaps in the cleft line segments simultaneously. We have confirmed that the windowed Hough transform, which has very effective clustering characteristics for dominant edges, is efficient in determining the line equation of the upper edges when the degree of inclination of the line is limited to, say, $\pm 30°$, as will be noted shortly.

To use the windowed Hough transform in the experiment, the slope a of a line is divided into 128 intervals so that for $|a| < 1$, the resolution is less than 1°, and the y intercept b is split into 64 parts. Thus, it follows from (1) and (2) that the N_a, N_b, Δa, Δb, and M_s become

$$N_a = 128, \quad N_b = 64$$

$$\Delta a = \frac{2}{128} \qquad (\Delta\theta = 0.89°)$$

$$\Delta b = \frac{|W_y|}{64}$$

$$M_s = N_a \cdot N_b = 8192 \text{ (words)}$$

where the size of a die S_x and S_y is given by

$$S_x = 64 \text{ (pixels)} \quad S_y = 48 \text{ (pixels)}$$

and the size of the recognition window W_x and W_y is determined by (3) as follows:

$$W_x = 128 \text{ (pixels)}, \quad W_y = 84 \text{ (pixels)}.$$

To show the efficiency of the proposed algorithms, the processing time versus recognition ranges of orientation in degrees was measured and is shown in Fig. 9(a). For comparison, several known pattern-recognition algorithms that use the global shape of a component or that emphasize noise immunity characteristics are also tested in the same system. In Fig. 9(a), the broken line denoted as the "Hough transform" shows the case in which the Hough transform is applied without detecting upper edges of a die, whereas the "proposed method" in the solid line implies the windowed Hough transform. The method of using the "second momentum" of the component as a representative of existing methods and the case of parallel processing of the proposed method in which the two central processors are used is also shown in Fig. 9(a). As is revealed in the figure, the processing time of the proposed method increases as the recognition range of deviated angle becomes wider, as expected; it is confirmed that the processing time for the enlarged parameter space in the windowed Hough transform would take more time. As a result, we have found that the proposed method shows high-speed recognition performance compared with the other methods if the orientation range, $|\theta|$ is less than 30°, and that parallel processing further improves the performance.

It is remarked that the image of a wafer used in the experiment includes three kinds of objects, that is, the rectangular

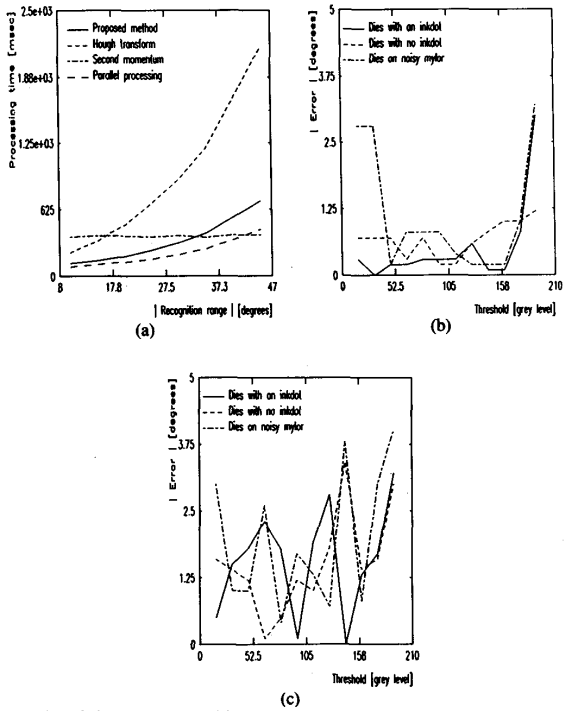


Fig. 9. Orientation recognition characteristics: (a) Orientation recognition time versus range; (b) orientation recognition errors and threshold; (c) orientation recognition errors and threshold (LSEM).

TABLE I

	Windowed Hough Method		LSE Method	
	Mean	Variance	Mean	Variance
An Inkdot	0.27	0.16	1.46	0.99
No Inkdot	0.54	0.29	1.34	0.99
Noisy Mylor	0.49	0.28	1.61	1.12

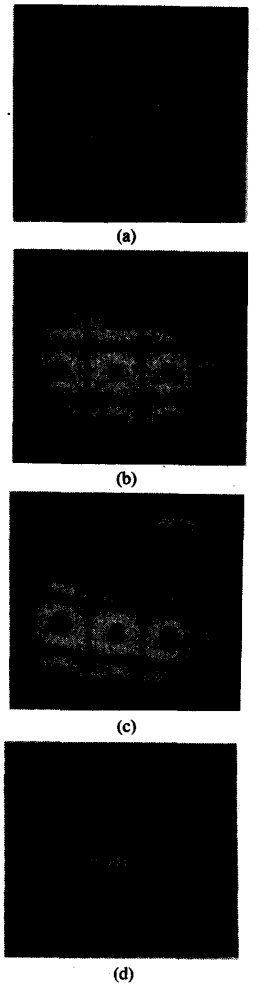


Fig. 10. Position recognition results: (a) Die with no inkdot; (b) die with inkdot; (c) die on tilted wafer; (d) die on noisy mylor.

shape of semiconductor dies, kerfs and metal base. The kerf is a channel between the semiconductor dies. The brightness of the semiconductor dies and that of kerfs are slightly different, whereas the brightness of the semiconductor dies and that of the metal base is quite different. Therefore, the slight brightness variation and the threshold level for image binarization yield variant dappled binary images and at the same time, affect the amount of noise in the images as well. Thus, in order to observe noise immunity, we plotted, in Fig. 9(b), measurement errors versus threshold levels for binarization of the proposed method. To compare, we also showed in Fig. 9(c) the case in which the least-square-error (LSE) method is used; it is most popular for a line equation estimation. The mean value and the variance of the measurement error between the threshold value 64 and 160 for each method is shown in Table I. In the case of the proposed method, the errors are less than 0.6°, whereas in the case of the LSE method, the error is greater than 1.3°. Thus, we may conclude that the noise immunity characteristic of the proposed method is better than the LSE method. It was confirmed that the errors of the LSE method were much more sensitive to both changes of the threshold and to noises in images.

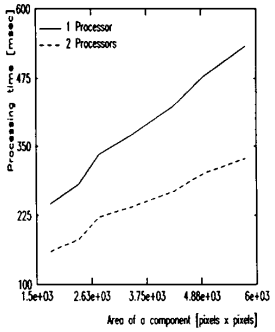
In Fig. 10, four types of samples are shown: a die with no inkdot, a die with an inkdot, a die for which the whole wafer is slightly tilted, and a die on a noisy mylor.

In the experiment, the size of the recognition window W_x and W_y is set up in reference to (4) as follows:

$$W_x = S_x + 0.1 \cdot S_x$$

$$W_y = S_y + 0.1 \cdot S_y.$$

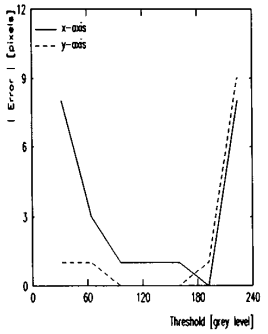
Experiments were conducted for position detection by the proposed method, and relations between the processing time and the size of the die are depicted in Fig. 11(a). Naturally, the processing time increases as the size of the die becomes larger because the recognition window should be extended. Nevertheless, if the size of the die is selected properly based on the required accuracy of the machine, the real-time appli-



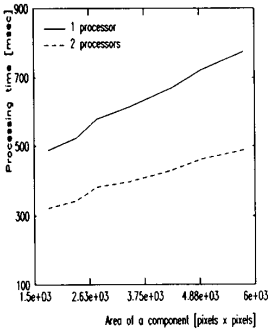
(a)

Size of template (pixels x pixels)	Processing time (msec)
2 x 2	310.9
3 x 3	507.3
4 x 4	701.4
5 x 5	982.5
6 x 6	1212.0
7 x 7	1562.8

(b)



(c)



(d)

Fig. 11. Position recognition characteristics: (a) Position recognition time and component size; (b) position recognition time by template matching; (c) position recognition errors and threshold; (d) parallel processing of position and orientation.

cation for the die-bonding machine is possible. For comparison, we conducted the same position detection by using the software template matching method, and as a result, the processing time versus the size of a template is measured, as is shown in Fig. 11(b). We observed that unless the size of the template is smaller than 2 (pixels) $\times$ 2 (pixels), which is too small for practical applications, real-time application is difficult. Further, if background images change frequently, as in the case of existence of other dies and inkdots, the reliability of the template matching method is not guaranteed.

Next, in order to examine noise immunity, position detection errors versus the threshold levels were obtained for the proposed method, as in Fig. 11(c), in which it is shown that the errors are smaller than 1 pixel if the threshold exists between grey level 96 and 192.

Finally, for the automatic assembly of dies in the die-bonding machine, the quality inspection is performed by checking whether an inkdot is marked on the central region of a die or not. The on-line processor in our vision system sets up a hardware window corresponding to the inkdot position of predetermined size and counts pixels for the inkdot in one frame time (1/30 s). Hence, the ratio

$$R_I = \frac{\text{Number of Pixels for the Inkdot}}{\text{Area of the Hardware Window (pixels)}}$$

determines the quality of the die as follows:

$$\text{The die is } \begin{cases} \text{good} & if\ R_I < 0.2 \\ \text{bad} & if\ R_I \geq 0.2. \end{cases}$$

The value 0.2 is chosen experimentally.

In light of all the observed results, we have concluded that the proposed windowed Hough method and the projection method successfully detect the orientation and the position of a component in the automatic assembly machines. The proposed methods are robust to imaging noise, fluctuations of edge points, and discontinuities of edges. In addition, the real-time application of the proposed algorithms is realistic for wide tilting angles of the component. In particular, for the die-bonding machine, where a typical requirement of allowed tilting angle of a die is less than or equal to 10° for the die placement, the windowed Hough method is a very efficient choice for orientation detection.

In the case in which the tilting angle of a component is rather wide, the parallel processing of the proposed algorithms is recommended by adopting two central processors. In the case of parallel processing of the algorithms, the recognition window is partitioned into two regions, whereas the parameter space for the windowed Hough transform and the projection array for the position detection are constructed in shared mem-

ories on the VME-BUS. The experiment for detecting the position and the orientation of a die simultaneously is executed, and the processing time is shown in Fig. 11(d) under the condition that the size of the die is 60×45 (pixels $\times$ pixels), and the orientation range is $|\theta| < 30°$. As a result, the processing time for determining the posture of a die is reduced from 600 ms for the single central processor to 350 ms.

IV. Concluding Remarks

We have developed a fast industrial vision system and pattern-recognition algorithms for automatically determining position, orientation, and quality of simple-shaped electronic components such as integrated-circuit chips. The vision system was successfully incorporated in the die-bonding machine [13] in which the projection method with dynamic thresholding finds the position of a die, and the windowed Hough transform determines the orientation of the die precisely.

The algorithms considered in the paper, however, are limited in applications; if the shape of the component to be handled is complicated or its placement is random, the algorithms may not be successful. Nevertheless, taking into account the practicalities of industrial manufacturing, we believe that this approach of designing a specific vision-based system with emphasis on real-time aspects and robustness can be a welcome alternative for the industrial sector if the productivity of manufacturing is enhanced, even if the algorithms employed are not general.

References

[1] R. T. Chin, "Survey: Automated visual inspection: 1981 to 1987," *Comput. Vision Graphics Image Proc.*, vol. CVGIP-41, pp. 346–381, Jan. 1988.

[2] R. T. Chin and C. A. Harlow, "Automated visual inspection: A survey," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. PAMI-4, no. 6, pp. 557–569, Nov. 1982.

[3] H. Yoda, Y. Ohuchi, Y. Taniguchi, and M. Ejiri, "An automatic wafer inspection system using pipelined image processing techniques," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. PAMI-10, no. 1, pp. 4–16, Jan. 1988.

[4] M. Yachida and S. Tsuji, "Industrial computer vision in Japan," *Tutorial Robotics*, pp. 325–337, 1984.

[5] M. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inform. Theory*, pp. 179–187, Feb. 1962.

[6] W. A. Perkins, "A model based vision system for industrial parts," *IEEE Trans. Comput.*, vol. C-27, pp. 126–143, 1978.

[7] M. Yachida and S. Tsuji, "A versatile machine vision system for complex industrial parts," *IEEE Trans. Comput.*, vol. C-26, pp. 882–894, 1977.

[8] J. D. Dessimoz, "Recognition and handling of overlapping industrial parts," in *Proc. 9th Int. Symp. Ind. Robots* (Washington, DC), 1979, pp. 357–366.

[9] N. Ayache and O. D. Faugeras, "HYPER: A new approach for the recognition and position of two-dimensional objects," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. PAMI-8, pp. 44–54, 1986.

[10] W. Brune and K. Bitter, "S.A.M. opto-electronic picture sensor in a flexible manufacturing assembly system," in *Robot Vision*. New York: Springer Verlag, IFS (Publications) Ltd., 1983, pp. 325–337.

[11] S. Inaba, "Applications of robots in assembly cells," in *Handbook of Industrial Robotics*. New York: Wiley, 1985, pp. 1117–129.

[12] R. Cunningham, "Segmenting binary images," in *Robotics Age in the Beginning*. New York: Hayden, 1983, pp. 121–135.

[13] Z. Bien, S.-R. Oh, I. H. Suh, J. O. Kim, and Y. S. Oh, "Automatic assembly for microelectronic components," *IEEE Cont. Syst. Mag.*, vol. 9, no. 4, pp. 15–19, June 1989.

[14] Video-Auto Die Bonder: Model 8030A, Production FOTON Automation, Chandler, AZ, 1982.

[15] J. Skalansky, "On the Hough technique for curve detection," *IEEE Trans. Comput.*, vol. C-27, pp. 923–926, 1976.

[16] M. Nagao and T. Matsuyama, "Edge preserving smoothing," *Comput. Graphics Image Proc.*, vol. CGIP-9, pp. 394–407, 1979.

[17] R. Pinkham, M. Novak, and K. Guttag, "Video RAM excels at fast graphics," *Electron. Design*, pp. 161–171, Aug. 1983.

[18] H. Ostu, "A threshold selection method from gray level histogram," *IEEE Trans. Syst. Man Cyber.*, vol. SMC-9, pp. 62–69, 1979.

[19] A. M. Mood, F. A. Graybill, and D. C. Boes, *Introduction to the Theory of Statistics*. New York: McGraw Hill, 1982, pp. 498–499.