Automated Vision System for Inspection of IC Pads and Bonds

Koduri K. Sreenivasan, *Student Member, IEEE,* Mandyam Srinath, *Senior Member, IEEE,* and Alireza Khotanzad, *Member, IEEE*

***Abstract*—One of the problems in increasing reliability in the manufacture of integrated circuit devices is inspection of the bond pads and the bonds connecting the bond pads to the lead fingers of the device. The continuing increase in packing density of VLSI circuits requires that the inspection process be completely automated. Here, we present methods for visual inspection of bond pads and bonds, which are intended to automatically extract parameters of significance in determining their quality, from two-dimensional images taken from the top of the IC wafer.**

I. Introduction

ONE of the critical procedures in semiconductor device manufacturing is making the electrical interconnections inside the device's protective enclosure. These fine leads connect active microelectronic circuit chips to sturdy electronics which ultimately mate with external electrical components [1]. A significant problem in increasing reliability in the manufacture of integrated circuit devices is that of inspection of these bonds. At present, such inspection and quality control is done off-line and frequently on a sample basis by human operators. In addition to being expensive and time consuming, the results of such testing are also somewhat subjective due to factors such as fatigue and limitations of human visual consistency. An automated system, on the other hand, leads to increased consistency and reliability of the inspection process. Further, the continuing increase in packing density of VLSI circuits requires that the inspection process be completely automated. For such a scheme to be feasible, the algorithms and techniques used in the inspection process must be implementable on workstations of reasonable cost. This paper utilizes image processing and pattern recognition techniques [2] to develop some new algorithms for use in an automated system for inspection of different shapes of pads and bonds.

The complete inspection of the bonding process consists of the following steps.

- Prebonding inspection to:
  - i) find the shape and size of the pad;
  - ii) locate the center of pad;
  - iii) analyze the probe and scratch marks on the bond pad.
- Post-bonding inspection to:
  - i) compute the shape, size, and location of the bond;
  - ii) analyze orientation of pig-tails in wedge bonds.
  - iii) Determine the 3-D profile of the wire connecting the pads and the pins.

In this paper we present algorithms to solve the 2-D problems. Procedures for determining the 3-D profile of the wires are significantly different in that a stereo pair of images of the device taken from two different positions of the camera will be typically required. As such this problem is not addressed in this paper.

Manuscript received September 21, 1992; revised January 5, 1993. This work was supported by a research grant from Texas Instruments.

The authors are with the Image Processing and Analysis Laboratory, Electrical Engineering Department, Southern Methodist University, Dallas, TX 75275.

IEEE Log Number 9207953.

II. Prebonding Inspection

One of the final steps in the manufacture of integrated circuits is the inspection of the probe marks on bonding pads caused during electrical testing of the integrated circuit wafers. Prebonding inspection is useful in isolating potential problem areas caused when the probe mark extends beyond the pad boundary, the mark is too large in size, or extraneous scratch marks have been introduced during electrical testing. Inspection of the bond pads is facilitated by evaluating parameters which can be used to quantify these defects. These include: 1) size of the pad, 2) location of the pad center, 3) size of marks on the pad, and 4) location and size of the largest mark, which is assumed to be the probe mark.

Here we present an algorithm to automatically evaluate all four parameters from images taken from the top of pads with probe marks. The algorithm is efficient in terms of storage requirements and speed; hence, seems suitable for real-time application.

A. Description of Algorithm

Given an image taken from the top of an integrated circuit wafer, the first step is to isolate the bond pad by segmenting a region around the pad and binarizing the resulting image. The pad is extracted from the binary image and all marks on the pad are filled. By doing a simple logical XOR operation on these two images, one with marks and the other with marks filled up, the marks on the pad can be determined.

The center of the pad is determined by computing the centroid of the pad image with all marks filled in. The probe mark is isolated by determining the largest object in the pad image.

The algorithm is based on using a runlength code [7] to represent the image in all the processing stages, thereby reducing the processing time. The algorithm has been tested on several images provided by Texas Instruments Inc., Dallas, TX. Details of the various steps in the algorithm are given in the following.

1) Binarization: Inspection of the histograms of several pad images shows that the images are basically bimodal, consisting of the pad (white), some small objects (white) of somewhat lower intensity around the pad, and the background (black). Ideally, the extraneous objects around the pad are not connected to the pad region. In such a case, the Isodata technique [3] would yield a good threshold for binarizing the image to segment out the pad region. However, in some of our images, possibly because of blur introduced by the camera system, the extraneous objects are not completely separated from the pad region. For such cases, it was found that increasing the threshold value improved the segmentation, by eliminating the extraneous objects surrounding the pad. As such, the threshold used for binarization is determined as

$$T = T_{\text{iso}} + (\text{GLEV} - T_{\text{iso}}) * 0.1 \tag{1}$$

where GLEV is the number of grey levels in the image (256 for a 8-b image).

2) Coding: The binary image is coded using a runlength coding scheme in which runs of ones in each row are identified separately. The data structure used to represent a run of ones consists of the x and y locations of the starting point of every run, the runlength (i.e., number of pixels in that run) and a color tag which is used later for connected component labeling. Initially the tag associated with any run is set to be the same as the order in which it appears as the image is scanned. That is, the tag number associated with the nth run obtained during the scanning process is initially set as n. Additionally, the number of runs in each row is also noted. Once this code representation of the image is determined, the actual image of the pad is no longer needed, as all further processing is carried out in the run-code domain.

3) Pad Segmentation Using Connected Component Labeling: Since the largest object within each window of interest is the pad, connected component labeling can be used to segment the pad. In the code domain, determination of connected components is equivalent to determining runlengths in successive rows which are "touching." We consider runs in adjacent rows to be touching if the x coordinate of either the start or the end point of a run in one row lies between the x coordinates of start and end points of a run in either of the two adjacent rows. Runs which touch belong to the same object (component).

Assuming that the image is scanned row wise from bottom to top, connected component labeling can be done row-wise, starting from the second row. Thus for every run $j$run, in row i, look for a touching run in the previous $(i-1)$st row, and change the color tag of both runs to the smaller color of the two. If there is a second run touching $j$run, proceed as above to assign the same tag to both runs as well as to all the runs preceding $j$run that have the same color.

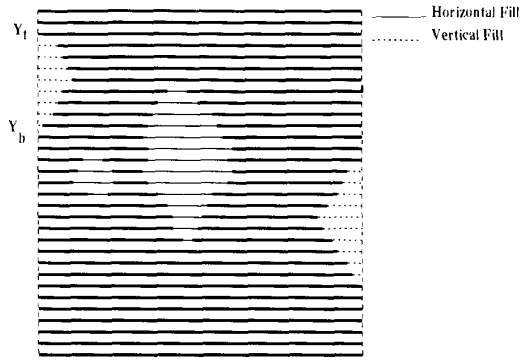


Fig. 1. Horizontal and vertical filling of marks on pad.

To find the largest connected object, the runlengths of like-colored runs are summed together. The component with the largest area is retained by neglecting all other runs. Here, the largest object is the pad containing markings due to probe marks, surface irregularities, dust, etc. We will refer to this coded and labeled form of the image as *raw_pad.*

4) Removal of Marks: Probe or other marks on the *raw_pad* image consist of black regions on a white background and can be removed by filling in the image. For most of the marks, filling in can be done in the code domain by scanning the image in a horizontal direction and merging together into a single run all the runs for any row in which there is more than one run. The parameters of the merged run are determined by the start point of the first run in the row and the end point of the last run in that row. The color tag is initialized in the same manner as before. The resulting coded image, referred to as *hfill_pad* represents the pad with the marks filled up in the horizontal scan. Note that now there would be a maximum of only one run for each row of the image.

The procedure for filling-in in the horizontal direction, described previously, can only take into account marks that cut across the rows of the pad image. Other marks, such as those occurring at the vertical edges of the pad, must be filled in using a scan procedure in the vertical direction on the horizontally filled *hfill_pad* image. Starting from the left, each column is simultaneously scanned from the top and bottom rows and the y coordinates y_t and y_b of the first pixel belonging to the pad region to be encountered in the two scan directions are noted. For all rows within these limits, the x location and/or runlength of the run in that row in the *hfill_pad* image is modified to extend the run up to the column being scanned. Fig. 1 shows an example of vertical filling. This procedure results in a pad image in which all the marks are filled in, giving the full shape and size of the pad, referred as *full_pad.*

5) Centroid Computation: The centroid and other information about the pad can be easily computed in the code domain using *full_pad* image. We note that all pixels in any run have the same y coordinates. Thus the y coordinate of the centroid

is easily determined as

$$\bar{y} = \frac{\sum_i y_i * \text{rlen}_i}{\sum_i \text{rlen}_i} \tag{2}$$

where y_i denotes the y coordinate of the ith run, and rlen_i its length.

To determine the x coordinate of the centroid, $\bar{x}$, we note that

$$\sum_{n=N_0}^{N_0+d-1} n = \sum_{n=0}^{N_0+d-1} n - \sum_{n=0}^{N_0-1} n = \frac{(2N_0+d-1)d}{2}.$$

It follows that $\bar{x}$ can be determined as

$$\bar{x} = \frac{\sum_i (2x_i + \text{rlen}_i - 1)\text{rlen}_i}{2\sum_i \text{rlen}_i} \tag{3}$$

where x_i denotes the x coordinate of the ith run.

6) Computation of Areas: The pad area can be easily determined from *full_pad* image. To find the area covered by the marks, we determine the area corresponding to runs in *raw_pad* image and subtract it from the pad area computed from *full_pad* image.

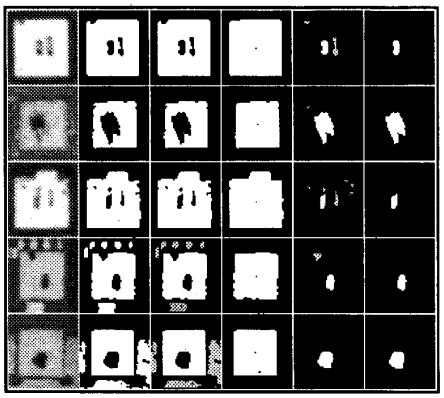
7) Probe Mark Identification: The location and size of the probe mark, which is assumed to be the largest mark on the pad, can be determined in run code by subtracting from *full_pad* image all the runs in *raw_pad* image. From *full_pad* image we know the first and last rows of the pad region. Within this region, consider for each row, the run from *full_pad* and the runs from *raw_pad.* Wherever there is a gap between two runs of *raw_pad,* there will be an element of mark. The region of search in each row is determined by the x location and runlength of the corresponding run from *full_pad.* Care should be taken in considering the end points.

By representing the marks also in runlength code, connected component labeling can be performed as explained earlier, to determine the biggest mark and the area of individual marks.

B. Experimental Results

Fig. 2 shows the different stages of the algorithm implemented on randomly selected samples. Fig. 2(a) is the 256 level image taken from the top of the integrated chip. Fig. 2(b) shows the binary images with the threshold selected as explained. Fig. 2(c) shows the image after connected component labeling. The largest object (pad) is shown as white and others are shaded to show the difference. Next the pad region is filled-in in both horizontal and vertical directions, and the centroid is located as shown in Fig. 2(d). Fig. 2(e) shows the marks on the pad, with the largest mark (probe mark) highlighted. This largest mark is isolated and is shown in Fig. 2(f).

It may be noted that for the pad images in Fig. 2(c), binarization yields edges along the boundary of the pad which are not straight lines but consist of deviations of one pixel. These regions are not to be considered as marks on the pad (Fig. 2(d)) during the filling procedure.



a b c d e f

Fig. 2. Stages of bond pad analysis.

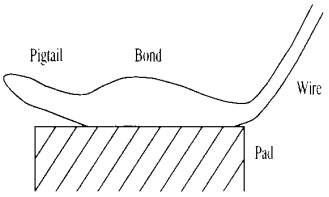


Fig. 3. Sideview of a typical wedge bond.

III. Post-Bonding Inspection

The pad parameters such as location, size, and pad center can be used during the bonding operation for placement of the bond. The bond is then inspected to determine its quality.

While both ball and wedge bonds have several features in common, wedge bonds, in many cases, have a fairly long pigtail as shown in Fig. 3. Further, with wedge bonds, the nature of the bonding process used may cause reflections from the bond surface, which may not be totally eliminated even by controlling the lighting conditions. Thus analysis of wedge bonds is somewhat more difficult than ball bonds. In the sequel, we, therefore, discuss a procedure for inspection of wedge bonds, which is also applicable to ball bonds.

A. Description of Algorithm

Bond quality is determined by the location of the bond on the pad and its shape. Ball bonds, ideally, must be centered on the pad and should be circular in shape. While the center of mass of the bond gives its location, measures of circularity can be used to determine how close its shape is being to circular. In addition, wedge bonds, in many cases, have a fairly long pig-tail. Thus parameters of interest include, in addition to bond size and location, the length and orientation of the pig-tail, etc. [6]. Bond quality is deemed to be acceptable if these parameters fall into ranges which define good bond quality. As this part of the algorithm involves morphological operations, it is more efficient to work in the pixel domain rather than the runlength code domain.

1) Extraction of Bond and Connecting Wire: After digitizing the bond image, the first step in the automatic inspection

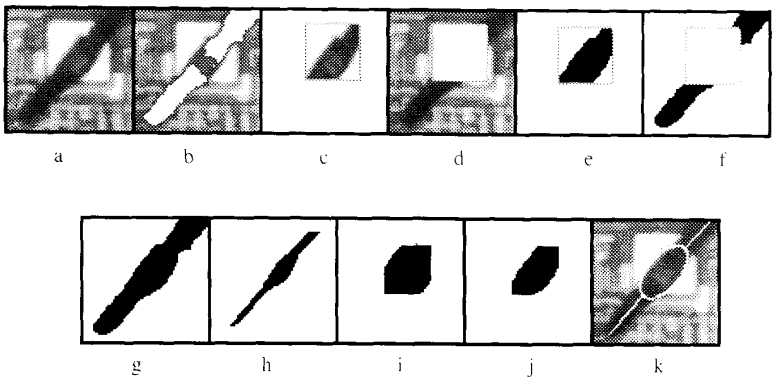


Fig. 4. Stages of wedge bond analysis.

process is to segment the bond, the connecting wire, and the pig-tail from the rest of the image. The images, because of the nature of the bonding process used, may contain reflections from the bond surface, which may not be totally eliminated by controlling the lighting conditions. Also, some of the images contain probe marks which may extend beyond the pad region, as well as scratch and other marks. Fig. 4(a) shows an image of a typical wedge bond. Commonly used binarization procedures based on choosing a suitable threshold from the histogram of the image cannot be used for segmentation in wedge bonds, since in many images the reflections have the same intensity levels as the background. As such, we propose a simple method of binarization by parts. Assuming that the pad size and shape are known from the prebonding inspection and that the reflections on the bond are confined to the pad region, we can use two different thresholds, one for the pad region and one for the rest of the image.

Inspection of the image shows that its histogram (see Fig. 5) can be modeled as having three regions consisting of i) the pad region (bright), ii) the wire and bond (dark), and iii) the background/reflections with an intensity level in between the other two. A simple clustering algorithm yielding three clusters can be used to find a threshold T_1 which is used to isolate the wire and bond from the rest of the image. First the image is semi-thresholded by setting all intensity levels below T_1 (that is, the intensities of those pixels corresponding to the wire and bond) to a level approximately equal to that of the pad and leaving the other pixels unaltered. Fig. 4(b) shows the semi-thresholded image corresponding to the image in Fig. 4(a). Under the assumption that the pad size and shape are known, we determine the location of the pad on this image by performing a correlation match with a template consisting of a block of white pixels equal to the size of the pad. The pad location is now transferred to the original image of Fig. 4(a). The resulting image is divided into two subimages, one consisting of the pad region (Fig. 4(c)) and the other (Fig. 4(d)), consisting of the rest of the image.

The subimage, Fig. 4(c), of the pad region contains the bond, reflections, and the pad. The histogram of this subimage can be modeled as having only two clusters and a binarization is performed using an Isodata threshold [3]. This yields a

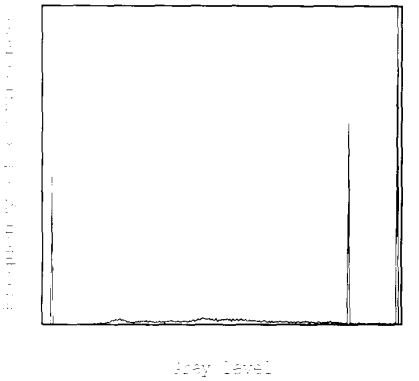


Fig. 5. Typical wedge bond image histogram.

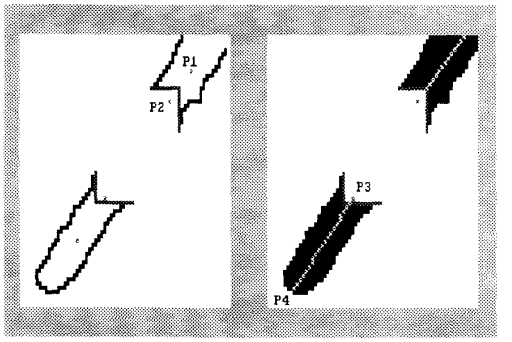


Fig. 6. Wire and pig-tail analysis.

binary image in which the white pixels correspond to the pad, and the reflections and other marks are merged with the bond to form the black pixels, see Fig. 4(e). The subimage Fig. 4(d), consisting of background, connecting wire, and pig-tail can be binarized using the threshold T_1 with black pixels corresponding to the connecting wire and the pig-tail and the white pixels corresponding to background as shown in Fig. 4(f). Combining the two subimages then yields a

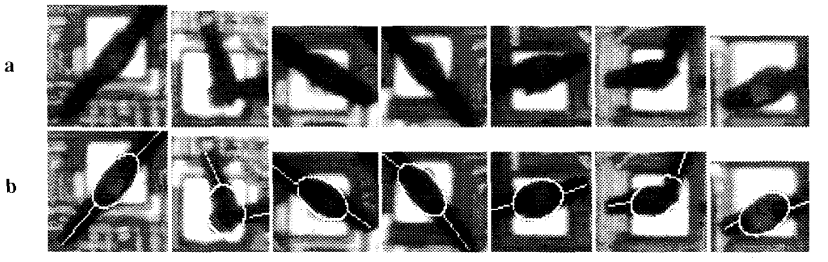


Fig. 7. (a) Images of typical wedge bonds. (b) Images after processing.

binarized image in which the bond, connecting wire, and pig-tail have been separated out. See Fig. 4(g). Any errors due to inaccuracies in the template matching procedure for locating the pad can be removed by eroding and dilating the black regions once.

2) Separation of Bond from Wire and Pig-Tail: Since the largest connected component in the segmented image will consist of the connecting wire, the bond, and the pig-tail, these can be isolated by performing connected component labeling on the segmented image. To separate the bond, connecting wire, and pig-tail from one another, we note that the bond is wider than the other two. Thus the area-to-perimeter ratio for this part of the image will be larger than for the other parts. This assumption makes it possible to use successive erosion and dilation operations to isolate the bond. The number of erosions, N_e, can typically be chosen to be between 3 and 5 depending on the wire size and magnification in the image, the assumption being that the wire and pig-tail get eroded much faster than the bond. The erosion procedure results in an image consisting of a cluster of black pixels from the bond with at most a line of pixels from the wire and pig-tail, as shown in Fig. 4(h). The cluster of pixels corresponding to the bond is isolated by matching with a template consisting of a rectangular block of black pixels of suitable size. Here, a block size of 15 by 15 pixels was used. Now dilating this cluster $N_e + 1$ times generates a mask which serves to isolate the bond in the segmented image. Fig. 4(h) shows the results of the erosion operation while Fig. 4(i) shows the mask used in isolating the bond. The isolated bond is shown in Fig. 4(j). Note that this step also yields an image consisting of only the wire and pig-tail as shown in Fig. 6(b).

3) Extraction of Bond Parameters: Bond quality is determined by a number of factors such as the bond size, location of the bond relative to the pad, length of pig-tail, etc. In order to determine these quantities, the best-fitting ellipse to the bond is first determined. The parameters of the ellipse such as the lengths of the major and minor axes, and orientation are related to the second-order moments of the isolated bond region as explained in [4].

Fig. 4(k) shows the best-fitting ellipse superimposed on the bonds, for the sample image.

4) Pig-Tail and Wire Analysis: Removing the bond from the segmented image yields an image consisting of only the pig-tail and wire. The first step is to identify which part corresponds to the pig-tail and which to the connecting wire.

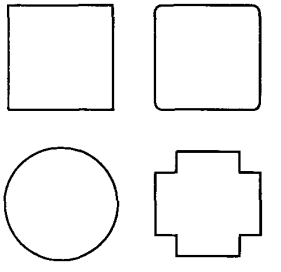


Fig. 8. Typical pad shapes that can be analyzed using this algorithm.

Clearly these two parts will lie on opposite sides of the minor axis of the best-fitting ellipse, and hence, can be labeled in different colors.

Assuming that the wire and the pig-tail are independently reasonably straight, we can determine their orientation by drawing a straight line between two suitably chosen points in each region. One of these points, P_1, can be the centroid of the region. The other point, P_2, is chosen to be the average of the pixels corresponding to the edge that results when the region is separated from the bond. See Fig. 6(a). An approximate length for the wire and pig-tail can then be obtained as twice the Euclidean distance between the two points P_1 and P_2. Alternatively, the two end pixels, P_3 and P_4, in the region along the line joining P_1 and P_2 can be identified and the length of the region determined as the distance between P_3 and P_4. See Fig. 6(b). Fig. 7(b) also shows the straight lines indicating the orientation and length of the connecting wire and pig-tail for the wedge bonds of Fig. 7(a).

IV. Conclusions

The algorithm for prebonding inspection has been implemented on images of pads with probe marks. Satisfactory results have been achieved with the algorithm as shown in the accompanying figures. Use of runlength encoding does make the program somewhat complex and lengthy. However, the execution time for the algorithm is small enough for its use to be feasible in real-time applications. The execution time compares very favorably with that reported in [5], where an algorithm for pad inspection which uses morphological operations directly on the pad image is described.

Fig. 8 shows some typical pad shapes that can be analyzed by this algorithm without any modifications. We note that if

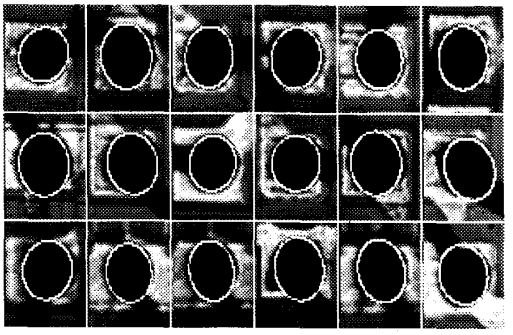


Fig. 9. Samples of ball bonds with the best fitting ellipse superimposed.

a polygon-like shape is assumed for the pad, certain stages of this algorithm can be simplified and the corner points can be estimated from the coded domain representation. It will also be easy to check if a part of the probe mark is off the pad.

This algorithm can also be implemented in the pixel domain, but then it will not be efficient in terms of time.

The algorithm to inspect the wedge bonds was implemented and tested on a set of 70 images from several integrated circuit devices. For each image, the location and dimensions of the best fitting ellipse along with the length and orientation of the pig-tail and the connecting wire were found. These are then compared with cut-off values obtained from design specifications to determine bond quality. Some typical images of wedge-bonds are shown in Fig. 7(a), while Fig. 7(b) shows these images with the locations of the bond and the orientation and length of the connecting wire and pig-tail superimposed.

The algorithm was also run on a set of images of ball bonds to determine their shape and location. In this case only the connecting wire needs to be determined. Fig. 9 shows some typical images of ball bonds with the best-fitting ellipses superimposed.

For prebonding inspection, on the average, processing of each (50 by 50) image took about 7 ms of time on DEC-5000/240 workstations, while the post-bonding inspection took on average 125 ms for all the stages of computation, which holds promise for implementation of the algorithms on a real-time inspection system.

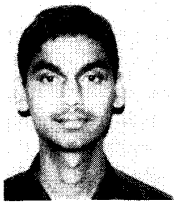
V. Acknowledgment

The authors gratefully acknowledge discussions with Dr. David Ho and Dr. Yee-Hsun U during various stages of this paper.

All the images used in the development and testing of these algorithms were supplied by the Process Automation Center of Texas Instruments Inc., Dallas, TX.

References

[1] B. L. Gehman, "Bonding wire microelectronic interconnections," *IEEE Trans. Comp., Hybrid, Manuf. Technol.*, pp. 375–383, Sept. 1980.
[2] R. C. Gonzalez and P. Wintz, *Digital Image Processing.* Reading, MA: Addison-Wesley, 1987.
[3] T. W. Ridler and E. S. Calvard, "Picture thresholding using an iterative selection method," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-8, pp. 630–632, Aug. 1978.
[4] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision.* New York: Wiley, 1992.
[5] M. Ahmed, C. E. Cole, R. C. Jain, and A. R. Rao, "INSPAD: A system for automatic bond pad inspection," *IEEE Trans. Semiconduct. Manuf.*, pp. 145–147, Aug. 1990.
[6] K. K. Sreenivasan, M. D. Srinath, and A. Khotanzad, "Automated vision system for inspection of wedge bonds," in *Proc. SPIE/IS&T Symp. on Electronic Imaging Science and Technology*, Feb. 1992, pp. 425–430.
[7] A. K. Jain, *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice Hall, 1989.



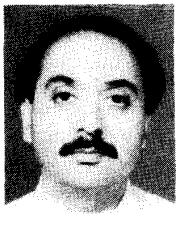
Koduri K. Sreenivasan (M'91) received the B.Eng. degree in electrical and electronics engineering with distinction, from Bharatiyar University, India in 1989, and the M.S.E.E. degree from Southern Methodist University in 1991, where he is currently working towards a Ph.D. degree.

His interests are in signal processing, computer vision, and neural networks.

Mr. Sreenivasan is a member of Tau Beta Pi and Who's Who Among Students in American Universities.

Mandyam D. Srinath (M'59–SM'74) received the B.Sc. degree from the University of Mysore, India, in 1954, the diploma in electrical technology from the Indian Institute of Science, Bangalore, India in 1957, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois, Urbana, in 1959 and 1962, respectively.

He has been on the Electrical Engineering faculties at the University of Kansas, Lawrence, and the Indian Institute of Science, Bangalore, India. He is currently a Professor of Electrical Engineering at Southern Methodist University, Dallas, TX, where he has been since 1967. He has published numerous papers in signal processing, control, and estimation theory and has co-authored two textbooks in signal processing.



Alireza Khotanzad (S'82–M'83) was born in Tehran, Iran, in 1956. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1978, 1980, and 1983, respectively.

In 1984 he joined the Faculty of the Department of Electrical Engineering, Southern Methodist University, Dallas, TX, where he is now an Associate Professor. His research interests are in the areas of computer vision, pattern recognition, and neural networks.