Pattern Recognition Vol. 17, No. 5, pp. 485–491, 1984.
Printed in Great Britain.

0031-3203/84 $3.00 + .00
Pergamon Press Ltd.

VISUAL INSPECTION USING LINEAR FEATURES

PETER KAUFMANN,* GERARD MEDIONI and RAMAKANT NEVATIA†
Intelligent Systems Group, University of Southern California,
Los Angeles, CA 90089-0272, U.S.A.

(*Received* 4 *August* 1983; *received for publication* 19 *December* 1983)

Abstract—This paper presents a method to inspect a printed circuit board. We use a feature-based system which matches symbolic descriptions of an object and a model. These symbolic descriptions are derived from line segments detected in an image. A model is built from a perfect board together with the expected defects. The matching technique is a variation of a line matching method ("Kernel" method) successfully applied at USC on aerial images. The results are then interpreted to generate a description of the match.

Automated visual inspection   Defects detection   Industrial automation   Pattern recognition
Graph matching

1. INTRODUCTION

Automatic inspection of printed circuit board assemblies is of obvious importance in electronics manufacturing. Several approaches to such inspection have been described in the past; a recent survey on automated visual inspection[1] contains over 200 references, including a large number for inspection of printed circuit boards and integrated circuits. We will not attempt a complete survey; basic techniques are described in textbooks on machine vision.[2,3]

The various approaches to inspection can be characterized as belonging to the following classes. In one class, the image is compared to an ideal or model image directly, on a pixel-by-pixel basis or by some form of area template matching. These methods are, of course, sensitive to variations in the lighting, reflectivity of the material and the size of the image. An alternative is to use a feature based description of the image and the model and to match the descriptions for inspection. Level and complexity of the description may vary; typical examples are the systems described by Baird, Perkins and Agin.[4–6] Lastly, some methods attempt to find defects described generically, for example connecting wires to be of a certain minimum width or the corners to have certain characteristics.[7–8] These methods have the advantage of being applicable to a new part without changes, however, in many cases the defects to be inspected may be product dependent.

Our system is a feature based matching system. In our system, the symbolic descriptions to be matched are derived from line segments detected in an image. In addition, we also have a model of the expected defects. The part we have used in our tests is a printed circuit board used in digital watches in Switzerland. Its main components are:

- a printed circuit board, approximately 2 × 2 cm;
- a square, black plastic integrated chip with 8 soldering points;
- an elongated, brass/ceramic capacitor with 2 soldering points;
- a cylindrical metallic quartz connected to the board by 2 soldering points;
- an elongated, brass battery contact riveted on the conductor.

The intensity of the parts of interest covers most of the black to white range and no effective threshold can be applied to the whole image to extract the desired parts. Figure 1 shows a complete printed circuit board with all elements in their proper locations. (All images shown in this paper were obtained by positioning the

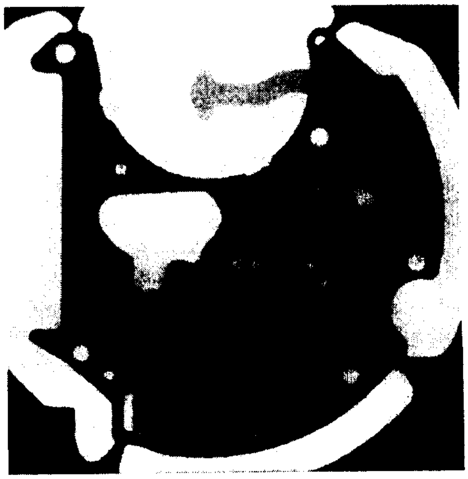


Fig. 1. Complete printed circuit board.

*Present address: E.T.A., Schild Strasse 17, Grenchen, Switzerland.
†To whom correspondence should be addressed.

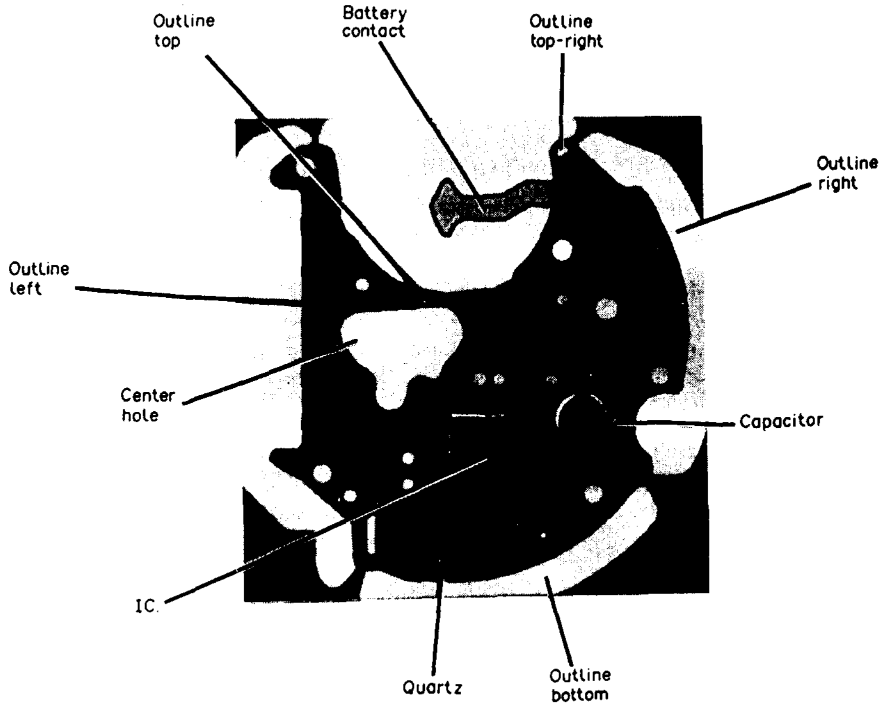


Fig. 2. Outline of the model.

camera above the object, with illumination provided by two sources, a light table under the object and a light source normal to the table. The resolution is 512 × 512 pixels.)

The defects we wish to detect are:

— broken printed circuit board;
— missing IC;
— missing capacitor;
— missing quartz;
— battery contact out of place or missing.

These defects are the ones that seemed to be of the major concern in the manufacturing process where this board is used. Here, we have not attempted inspection of the conductor paths; apparently, for simple boards, visual inspection of these paths is not of major concern.

2. DESCRIPTION OF THE MODEL

The model description consists of a set of line segments which are logically divided into submodels. Each submodel contains one or several segment chains which represent contours or outlines of parts or assemblies. Each of these submodels also has a descriptor which tells the comparison program how to interpret matches with the segment chains contained in the submodels. It contains, for example, a value for each segment chain which indicates the amount of correspondence with the part required to consider the chain as matched. The exact format of each submodel descriptor is defined as:

1. lexical description of submodel (e.g. name of corresponding part in assembly);
2. minimum required length of correspondence for each segment chain;
3. maximum acceptable length of non-correspondence for each segment chain (note that the absence of some parts, such as the capacitor or integrated circuit, is indicated by too many line segments being visible, rather than not enough);
4. message to be printed out in each case of success or failure.

Figure 2 shows the outlines of parts of interest that constitute the segment component of the model.

3. SEGMENT ENCODING OF THE PART

To find the segment representation of an image, several processing steps have to be executed in sequence:

— edge detection (followed by edge thinning and thresholding);
— approximation of edges by line segments.

These steps are only sketched here; a more complete description of the algorithms can be found in Nevatia and Babu.[9] In this method, edge templates in six directions are convolved with the image and the direction with the highest output determines the magnitude and direction of the edge associated with a pixel. Based on direction and magnitude information, an edge thresholding and thinning operation follows. The threshold value is kept rather low, so that no

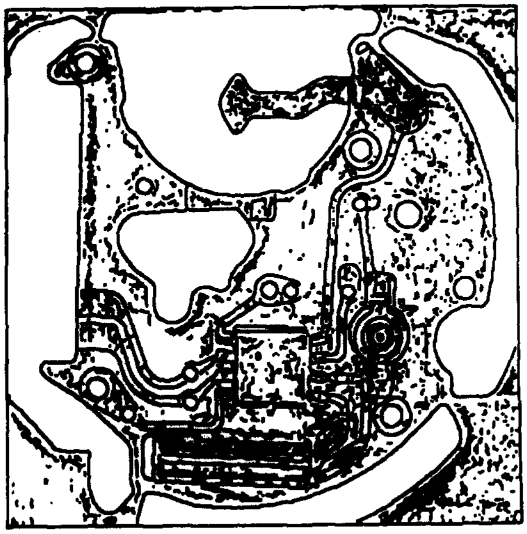


Fig. 3. Edges using 5 × 5 masks.

"good" edges are discarded at this early stage. Figure 3 shows the output of the edge detector using six 5 × 5 masks.

Next, the edges are linked to form continuous curves and these curves are approximated by piece-wise linear segments (using the method described in Nevatia and Babu[9]). Each line segment is described as follows:

- segment number (each segment has a unique identification);
- family number (unique for each linked segment chain);
- predecessor number (if the segment belongs to a chain);
- successor number (if the segment belongs to a chain);
- begin and end point coordinates;
- strength (average contrast along the segment);
- length;
- orientation angle.

Although length and orientation are redundant, they are stored so that they are calculated only once. The algorithm to fit the line segments to the edge points is defined so that all segment end points are actual edge points and the normal distance between the line segment and the edge is never greater than a fixed number of pixels, say d. While generating these line segments, several filtering operations can be performed. Weak edges can be suppressed using length and strength properties of the edges. Figure 4 shows the line segments computed from the edges in Fig. 3. A line fitting tolerance, d, of 2 pixels was used and isolated segments shorter than 10 pixels were discarded.

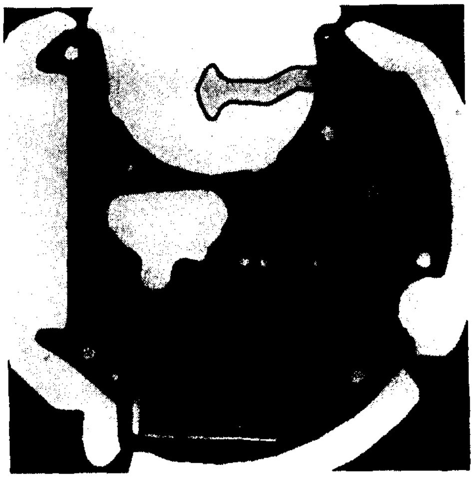


Fig. 4. Line segments.

4. COMPARING THE PART WITH THE MODEL

Now that we have the same representation for the model and a part under inspection, the next step is to find out how the part compares with the model. In order to compare them, we will assume identical position, orientation and size of the model and the part. If these constraints are not given by the layout of the inspection (fixtures, positioning device), then these parameters have to be roughly estimated in a preprocessing step, which is discussed in Section 4.1. Once we have the model and the part in an identical coordinate frame, corresponding segment chains have to be identified, as described in Section 4.2.

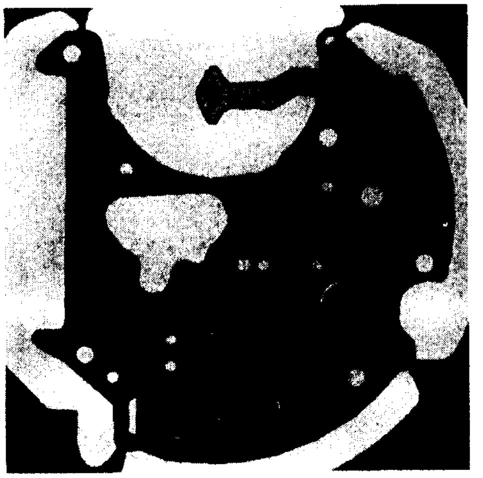
4.1. *Alignment of model and part*

Given our data, which is a printed circuit board to be used in digital watches, we extract the chain of segments containing the longest vertical segment and match it with the outline of the left side of the model using a brute force technique, i.e. evaluating the quality of every possible set of matches, to estimate the parameters of the rotation and translation. This technique is effective because of the small number of elements involved in the matching, typically less than 5.

4.2. *Matching the part with the model*

The method used to match each part with the model is a variation of the Kernel method[10,11] successfully applied at USC on aerial images. The main differences stem from the fact that the structures to be matched were already aligned in a previous step. We will sketch here the main components of the process.

Definitions. We will denote the segments of the part as a_i, $1 \leq i \leq n$ and call them *objects*. We will denote the segments of the model as l_j, $1 \leq j \leq m$ and call them *labels*. The set $A = \{a_i\}$ is the *scene*. The set $L = \{l_j\}$ is the *model*. We want to compute the *possibility* $p(i,j)$ for object a_i to have label l_j, $p(i,j)$ can be either 0 or 1. The method presented here relies mostly on geometrical constraints, meaning that when we assign label l_j to object a_i, we expect to find an object a_h with the label l_k in a certain area dependent on (i, j, k). This area is

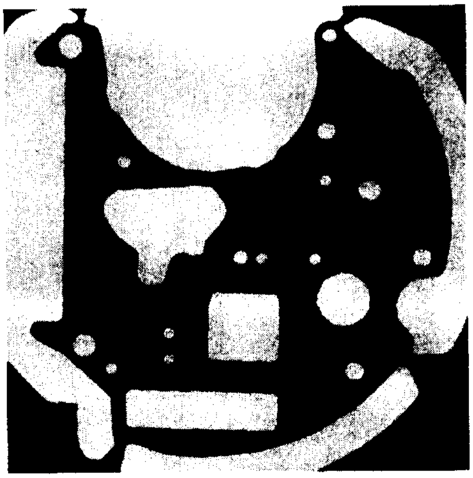


(a)

```
the part:  outline/top_right_corner is found
   not matched: .0000000           limit: 9.016455
   matched:      60.10970          limit: 51.09325
   model length: 60.10970
the part:  outline/left_side is found
   not matched: .0000000           limit: 86.02527
   matched:      573.5018          limit: 487.4765
   model length: 573.5018
the part:  outline/top is found
   not matched: .0000000           limit: 78.13128
   matched:      390.6564          limit: 312.5251
   model length: 390.6564
the part:  outline/right_side is found
   not matched: .0000000           limit: 65.95770
   matched:      439.7180          limit: 373.7603
   model length: 439.7180
the part:  outline/bottom is found
   not matched: .0000000           limit: 50.38727
   matched:      335.9151          limit: 285.5279
   model length: 335.9151
the part:  center_hole is found
   not matched: .0000000           limit: 62.00780
   matched:      413.3853          limit: 351.3775
   model length: 413.3853
the part:  capacitor is found
   not matched: 14.00000           limit: 9.855159
   matched:      162.4427          limit: 187.2480
   model length: 197.1032
the part:  Integrated Circuit is in place
   not matched: 81.78885           limit: 15.12139
   matched:      163.3025          limit: 287.3064
   model length: 302.4278
the part:  Quartz is in place
   not matched: 148.2147           limit: 20.80467
   matched:      167.5798          limit: 395.2887
   model length: 416.0933
the part:  Battery Contact is found
   not matched: .0000000           limit: 48.45157
   matched:      359.0243          limit: 274.5589
   model length: 323.0105
```

(b)

Fig. 5. Matching results.



(a)

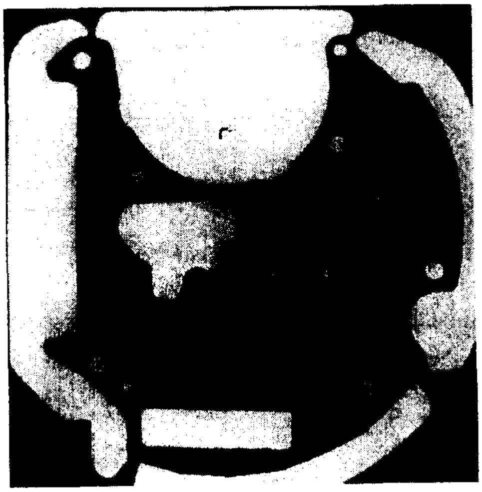
```
the part:  outline/top_right_corner is found
   not matched: .0000000           limit: 9.016455
   matched:      75.83074          limit: 51.09325
   model length: 60.10970
the part:  outline/left_side is found
   not matched: .0000000           limit: 86.02527
   matched:      574.5556          limit: 487.4765
   model length: 573.5018
the part:  outline/top is found
   not matched: 67.23146           limit: 78.13128
   matched:      326.8160          limit: 312.5251
   model length: 390.6564
the part:  outline/right_side is found
   not matched: .0000000           limit: 65.95770
   matched:      440.0282          limit: 373.7603
   model length: 439.7180
the part:  outline/bottom is found
   not matched: .0000000           limit: 50.38727
   matched:      337.7853          limit: 285.5279
   model length: 335.9151
the part:  center_hole is found
   not matched: .0000000           limit: 62.00780
   matched:      414.1680          limit: 351.3775
   model length: 413.3853
the part:  capacitor is missing or not in place.
   not matched: .0000000           limit: 9.855159
   matched:      197.1032          limit: 187.2480
   model length: 197.1032
the part:  Integrated Circuit is missing or not
in place
   not matched: .0000000           limit: 15.12139
   matched:      302.4278          limit: 287.3064
   model length: 302.4278
the part:  Quartz is missing or not in place
   not matched: .0000000           limit: 20.80467
   matched:      416.0933          limit: 395.2087
   model length: 416.0933
the part:  Battery Contact is not found
   not matched: 323.0105           limit: 48.45157
   matched:      .0000000          limit: 274.5589
   model length: 323.0105
```

(b)

Fig. 6. Matching results.

called the *window* $w(i, j, k)$. Finally, we define the relation C, "is compatible with", between (i,j) and (h,k) as: (i,j) IS COMPATIBLE WITH $(h,k) \Leftrightarrow (i,j)\ C(h,k) \Leftrightarrow a_h$ in $w(i,j,k)$ AND a_i in $w(h,k,j)$. We need to check both predicates because the relation "is in w" is not symmetric. The method then proceeds as follows. Find a few pairs of very likely matches that are also mutually compatible, call these pairs the *kernel* and check each possible assignment for compatibility with this kernel.

Assignment of possibilities. Since scene and model are approximately aligned, this step is very simple. We will assign label l_j to object a_i if the corresponding segments have approximately the same position and orientation. The tolerance on orientation is large for small segments and very small for long segments. As we assign these labels, we mark every pair that is very closely matched as a good candidate for kernel element. We then extract the largest set of mutually compatible assignments from these candidates and call it the kernel. Then each other assignment is verified

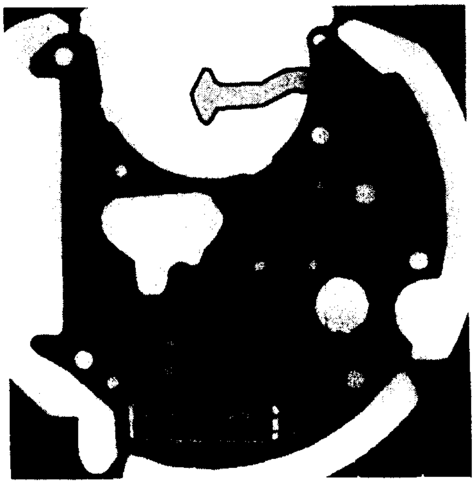


(a)

```
the part:  outline/top_right_corner is found
   not matched: .0000000         limit: 9.016455
   matched:       69.96133       limit: 51.09325
   model length: 60.10970
the part:  outline/left_side is found
   not matched: .0000000         limit: 86.02527
   matched:       604.6619       limit: 487.4765
   model length: 573.5018
the part:  outline/top is found
   not matched: .0000000         limit: 78.13128
   matched:       402.0784       limit: 312.5251
   model length: 390.6564
the part:  outline/right_side is found
   not matched: .0000000         limit: 65.95770
   matched:       444.0939       limit: 373.7603
   model length: 439.7180
the part:  outline/bottom is found
   not matched: .0000000         limit: 50.38727
   matched:       333.7796       limit: 285.5279
   model length: 335.9151
the part:  center_hole is found
   not matched: .0000000         limit: 62.00780
   matched:       413.3684       limit: 351.3775
   model length: 413.3853
the part:  capacitor is found
   not matched: 14.03567         limit: 9.855159
   matched:       149.6836       limit: 187.2480
   model length: 197.1032
the part:  Integrated Circuit is in place
   not matched: 10.29563         limit: 15.12139
   matched:       194.5388       limit: 287.3064
   model length: 302.4278
the part:  Quartz is missing or not in place
   not matched: .0000000         limit: 20.80467
   matched:       414.1096       limit: 395.2887
   model length: 416.0933
the part:  Battery Contact is not foud
   not matched: 323.0105         limit: 48.45157
   matched:       .0000000       limit: 274.5589
   model length: 323.0105
```

(b)

Fig. 7. Matching results.



(a)

```
the part:  outline/top_right_corner is found
   not matched: .0000000         limit: 9.016455
   matched:       59.29365       limit: 51.09325
   model length: 60.10970
the part:  outline/left_side is found
   not matched: .0000000         limit: 86.02527
   matched:       581.3698       limit: 487.4765
   model length: 573.5018
the part:  outline/top is found
   not matched: .0000000         limit: 78.13128
   matched:       414.3125       limit: 312.5251
   model length: 390.6564
the part:  outline/right_side is found
   not matched: .0000000         limit: 65.95770
   matched:       450.9297       limit: 373.7603
   model length: 439.7180
the part:  outline/bottom is found
   not matched: .0000000         limit: 50.38727
   matched:       330.0233       limit: 285.5279
   model length: 335.9151
the part:  center_hole is found
   not matched: .0000000         limit: 62.00780
   matched:       415.7958       limit: 351.3775
   model length: 413.3853
the part:  capacitor is missing or not in place
   not matched: .0000000         limit: 9.855159
   matched:       199.0475       limit: 187.2480
   model length: 197.1032
the part:  Integrated Circuit is in place
   not matched: .0000000         limit: 15.12139
   matched:       220.7356       limit: 287.3064
   model length: 302.4278
the part:  Quartz is in place
   not matched: .0000000         limit: 20.80467
   matched:       365.5972       limit: 395.2887
   model length: 416.0933
the part:  Battery Contact is found
   not matched: .0000000         limit: 48.45157
   matched:       339.1811       limit: 274.5589
   model length: 323.0105
```

(b)

Fig. 8. Matching results.

against this kernel for compatibility. Finally, we bridge accidental gaps in segment chains by looking at the labels assigned to the predecessor and successor of a given segment s; if both have a label, then we globally match chains and force a label on s.

Interpretation. Once scene and model have been matched, we generate a description of this match with each submodel part based on the length of matched chains. Each submodel contains the minimum and maximum bounds required for acceptance.

Results of this matching process are shown and discussed in the next section.

5. RESULTS AND CONCLUSIONS

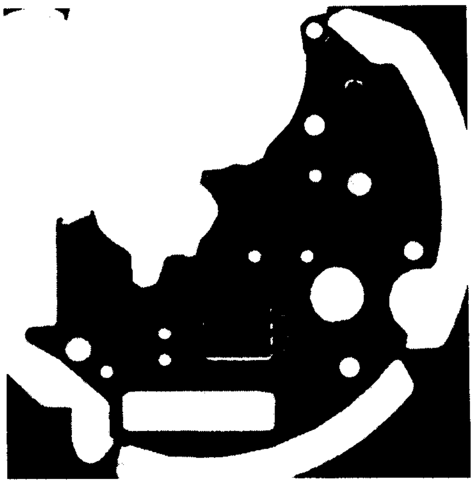
Figures 5–10 show the results obtained with 6 different instances of the given printed circuit board representing different possible configurations. For each example, Fig. (a) shows the segments matched

(a)

```
the part:  outline/top_right_corner is found
    not matched: .0000000          limit: 9.016455
    matched:      58.25959         limit: 51.09325
    model length: 60.10970
the part:  outline/left_side is found
    not matched: .0000000          limit: 86.02527
    matched:      571.9929         limit: 487.4765
    model length: 573.5018
the part:  outline/top is found
    not matched: 46.71318          limit: 78.13128
    matched:      356.0725         limit: 312.5251
    model length: 390.6564
the part:  outline/right_side is found
    not matched: .0000000          limit: 65.95770
    matched:      436.3224         limit: 373.7603
    model length: 439.7180
the part:  outline/bottom is found
    not matched: .0000000          limit: 50.38727
    matched:      368.0060         limit: 285.5279
    model length: 335.9151
the part:  center_hole is found
    not matched: .0000000          limit: 62.00780
    matched:      412.7569         limit: 351.3775
    model length: 413.3853
the part:  capacitor is found
    not matched: .0000000          limit: 9.855159
    matched:      164.6381         limit: 187.2480
    model length: 197.1032
the part:  Integrated Circuit is in place
    not matched: 73.30357          limit: 15.12139
    matched:      199.4696         limit: 287.3064
    model length: 302.4278
the part:  Quartz is in place
    not matched: 45.26370          limit: 20.80467
    matched:      333.6798         limit: 395.2887
    model length: 416.0933
the part:  Battery Contact is not found
    not matched: 323.0105          limit: 48.45157
    matched:      .0000000         limit: 274.5589
    model length: 323.0105
```

(b)

Fig. 9. Matching results.



(a)

```
the part:  outline/top_right_corner is found
    not matched: .0000000          limit: 9.016455
    matched:      55.81174         limit: 51.09325
    model length: 60.10970
the part:  outline/left_side is not found
    not matched: 102.8470          limit: 86.02527
    matched:      319.4996         limit: 487.4765
    model length: 573.5018
the part:  outline/top is not found
    not matched: 249.8139          limit: 78.13128
    matched:      125.5273         limit: 312.5251
    model length: 390.6564
the part:  outline/right_side is found
    not matched: .0000000          limit: 65.95770
    matched:      435.3755         limit: 373.7603
    model length: 439.7180
the part:  outline/bottom is found
    not matched: .0000000          limit: 50.38727
    matched:      366.6647         limit: 285.5279
    model length: 335.9151
the part:  center_hole is not found
    not matched: 114.4814          limit: 62.00780
    matched:      268.0096         limit: 351.3775
    model length: 413.3853
the part:  capacitor is missing or not in place
    not matched: .0000000          limit: 9.855159
    matched:      197.5216         limit: 187.2480
    model length: 197.1032
the part:  Integrated Circuit is in place
    not matched: .0000000          limit: 15.12139
    matched:      161.5724         limit: 287.3064
    model length: 302.4278
the part:  Quartz is missing or not in place
    not matched: .0000000          limit: 20.80467
    matched:      414.2710         limit: 395.2887
    model length: 416.0933
the part:  Battery Contact is not found
    not matched: 323.0105          limit: 48.45157
    matched:      .0000000         limit: 274.5589
    model length: 323.0105
```

(b)

Fig. 10. Matching results.

with the model of Fig. 2 superimposed on the image of the part and Fig. (b) gives a detailed explanation of the matching evaluation. For each sub-part of interest, the program outputs whether that sub-part is found and if so how much of the total length of model segments has been matched—this can be a measure of the quality of the match. A defect can be indicated by not enough of the model segments being found or by too much of the segment length being visible. The acceptable limits are also given in the result figures.

It is clear that our program is able to detect the desired defects for the examples shown. However, the matching system would only be a component of an inspection system. There is much additional useful information in the images that we have not utilized. For example, we indicated the presence of the quartz crystal merely by the boundaries of the hole being invisible. Many lines, parallel to the axis of the quartz, are also typically detected by the line finder and these along with some use of intensity variations could be used to more accurately verify that the correct part was in place (e.g. as in Horn[12]). Thus, our system may be regarded as a top-level cueing mechanism that detects major defects to be verified by a more specific, goal-oriented inspection module.

REFERENCES

1. R. T. Chin and C. A. Harlow, Automated visual inspection: a survey, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAM-11**, 557–573 (1982).
2. R. Nevatia, *Machine Perception*. Prentice Hall, Englewood Cliffs, New Jersey (1982).
3. D. Ballard and C. Brown, *Computer Vision*. Prentice Hall, Englewood Cliffs, New Jersey (1982).
4. M. L. Baird, SIGHT-1: a computer vision system for automated IC chip manufacture, *IEEE Trans. Syst. Man. Cybernet.* **SMC-8**, 133–139 (1978).
5. W. A. Perkins, A model based vision system for industrial parts, *IEEE Trans. Comput.* **C-27**, 126–143 (1978).
6. G. L. Agin, Computer vision systems for industrial inspection and assembly, *Computer* **13**, 5, 11–20 (1980).
7. M. Ejiri, T. Uno, M. Mese and S. Ikeda, A process for detecting defects in complicated patterns, *Comput. Graphics Image Process.* **2**, 326–339 (1973).
8. J. R. Jarvis, Visual inspection automation, *Computer* **13**, 5, 32–38 (1980).
9. R. Nevatia and K. R. Babu, Linear feature extraction and description, *Comput. Graphics Image Process.* **13**, 257–269 (1980).
10. G. G. Medioni, Matching linear features of images and maps, *Proceedings of the ARPA Image Understanding Workshop*, Palo Alto, CA, September 1982, pp. 103–111 (1982).
11. G. G. Medioni, Matching high level features of an aerial image with a map or another image, *Proceedings of Workshop on Computer Vision*, Rindge, NH, August 1982, pp. 113–115 (1982).
12. B. K. P. Horn, A problem in computer vision: orienting silicon integrated circuits chips for lead bonding, *Comput. Graphics Image Process.* **4**, 294–303 (1975).

About the Author—PETER KAUFMANN is a development engineer in a watch manufacturing company in Switzerland. He presently works on a vision system for industrial inspection and assembly monitoring from the Swiss Federal Institute of Technology in Zurich, Switzerland, in 1981. He then joined the company ETA in Switzerland, from where he was sent to the Institute of Physics and Imaging Science of USC to gain experience in image processing. At USC-IPIS, he spent one year working on several projects, including the one presented here, where Dr. Werner Frei of IPIS generously allowed him to do research with Dr. Nevatia and his group.

About the Author—GERARD G. MEDIONI received the diplôme d'Ingénieur Civil from the Ecole Nationale Supérieure des Télécommunications, Paris, France, in 1977, and the M.S. and Ph.D. degrees in Computer Science from the University of Southern California in 1980 and 1983, respectively.

He is currently a Research Assistant Professor in the Department of Electrical Engineering, University of Southern California, Los Angeles.

His research interests include artificial intelligence and computer vision.

Dr. Medioni is a member of the Institute of Electrical and Electronics Engineers and the Association for Computing Machinery.

About the Author—RAMAKANT NEVATIA was born in Fatehpur, India. He received the B.S. degree from the University of Bombay, India, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, all in Electrical Engineering.

He has been with the University of Southern California, Los Angeles, since 1975, where he is currently an Associate Professor of Electrical Engineering and Computer Science. He spent the academic year 1981–1982 at Stanford University as a Visiting Professor. He has authored *Machine Perception* and *Computer Analysis of 3-D Curved Objects*. He is an associate editor of *Pattern Recognition* and *Computer Vision, Graphics and Image Processing*. His research interests are artificial intelligence and computer vision.

Dr. Nevatia is a member of the Institute of Electrical and Electronics Engineers, the Association for Computing Machinery and the American Association of Artificial Intelligence.