

# Metal Surface Inspection Using Image Processing Techniques

HON-SON DON, KING-SUN FU, FELLOW, IEEE, C. R. LIU, AND WEI-CHUNG LIN, MEMBER, IEEE

**Abstract**—The feasibility of applying image processing techniques to metal surface inspection is demonstrated. Two methods for metal surface inspection are described. In the first method, the metal surface reflective power and the metal surface normal are related by a random surface scattering model. The metal surface profile can then be computed from metal surface normal. The second method applies pattern recognition techniques to classify metal surfaces into classes of different roughness. Methods of feature extraction and classification have been tested experimentally and the performances of different types of classifier have been compared. A two-level tree classifier using nonparametric linear classifiers at each node gives better than 90 percent correct classification on our testing set.

## I. INTRODUCTION

A N IMPORTANT problem confronting industry, whenever accurate inspection and rapid production are necessary, is the designation and inspection of surfaces. The performance of a surface from a finishing viewpoint depends upon the dimensional characteristics of the surface irregularities. These characteristics vary considerably and depend upon the surface material and the method of processing. The most important characteristics are the following:

1) surface roughness, which means the relatively finely spaced surface irregularities: on surfaces produced by machining and abrasive operations, roughness including irregularities produced by cutting action of tool edges and abrasive grains, and by the feed of machine tool;

2) surface waviness, which specifies the surface irregularities of greater spacing than the roughness: on machined surfaces with irregularities resulted from machine or work deflections, vibrations, etc.

Conventionally, surface roughness and waviness are measured by using stylus instruments [1], [2], laser scanners, etc., which can actually measure the surface profile of the samples. However, these methods are rather slow since they measure the surface profile at every point on the surface. In practice, many circumstances exist where we do not need such detailed information about the metal surface. For example, we may only want to know the relative roughness of a metal surface in order to determine whether

or not further surface processing is necessary. In this paper, we describe an application of image processing techniques to metal surface inspection. Since no mechanical part is used in the equipments of image processing operations, the speed of inspection is potentially faster than that of using conventional methods.

This paper contains two major parts. In the first part, we have applied an image processing technique to reconstruct the metal surface profile from its surface image. In general, it is not an easy task to reconstruct the surface profile from its surface image for materials of arbitrary textures and colors (like the wooden board). However, for materials such as metals, which have uniform color and surface reflectance, this becomes possible. In this case, the surface reflective power at each point can uniquely determine the local normal direction, and the overall surface profile can be determined from the local information. An experiment was conducted to reconstruct the surface profile of a metal sample from its surface image. Although the result is not yet good enough to compete with that using conventional methods, it can certainly be used for some inspection purpose with the advantage of higher speed.

In the second part of this paper, we propose a texture classification scheme for automatic metal surface inspection. The high speed of this approach makes on-line metal surface inspection possible. An experiment was conducted to classify six classes of metal surface with different degrees of roughness. The contrast measurements of the spatial gray level co-occurrence matrices from the reflective light intensities of metal surfaces were selected as the features. A nonparametric linear classifier constructed from the fixed increment error-correcting training procedure was used, but the result was not satisfactory. However, a tree classifier implemented by linear classifiers has given very successful results. It gives better than 90 percent correct classification which should be good enough for many inspection purposes.

## II. DETERMINATION OF METAL SURFACE PROFILE

### A. Metal Surface Scattering Model [3]

Several phenomena play a role in the distribution of light scattered from a metallic surface. The most important effect is the variation of surface normal. The variation of surface normal generally occurs on two scales, a fine scale variation representing basic surface roughness and a rough

Manuscript received February 22, 1983; revised June 26, 1983. This work was supported by NSF Grant ECS 81-19886 and ONR Contract N83K0385.

H. S. Don, K. S. Fu, and W. C. Lin are with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

C. R. Liu is with the Department of Industrial Engineering, Purdue University, West Lafayette, IN 47907.

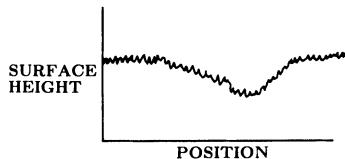


Fig. 1. Variation in surface height for rough surface.

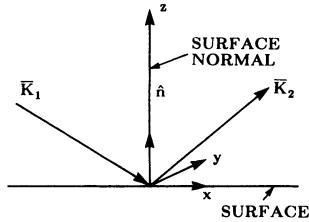


Fig. 2. Scattering geometry according to Beckmann.

surface waviness resulting from surface irregularities of greater spacing than the fine scale variation. This is shown in Fig. 1. Mundy and Porter used a surface scattering model for the inspection of metal surface defects [3]. We also use this model as a basis for the reconstruction of metal surface profile. In this paper, our purpose is to determine the profile of the rough surface waviness.

The scattering model is most suitable for the case of rough surface normal variations in the presence of random fine scale surface variations. Beckmann and Spizzichino [4] have explored this case extensively for various scales of surface roughness. In this paper, we only consider the case where the surface is rough compared to the wavelength of light. The reflection coefficient for scattered power in this case is given as

$$\langle \rho \rho^* \rangle = \frac{\pi R^2 F^2}{A V_z^2} \left( \frac{T}{2\sigma} \right)^2 \exp \left\{ - \left( \frac{V_{xy}}{V_z} \right)^2 \left( \frac{T}{2\sigma} \right)^2 \right\} \quad (1)$$

where

- $R$  reflection coefficient of an equivalent smooth surface,
- $k = 2\pi/\lambda$ ,
- $\lambda$  wavelength of illumination,
- $A$  area of illuminated surface,
- $\bar{V}$   $= \bar{k}_1 \bar{k}_2$ ,
- $V_z$  component of  $V$  along the surface normal,
- $V_{xy}$  component of  $V$  perpendicular to surface normal,
- $\bar{k}_1$  incident wave vector,
- $\bar{k}_2$  reflected wave vector,
- $T$  correlation distance of surface roughness,
- $\sigma$  surface height variance,
- $F$  angular dependency of the incident and reflected waves.

The coordinate system and scattering vectors are defined in Fig. 2.

It will prove useful to transform this notation to that introduced by Horn [5], which is defined in Fig. 3. It

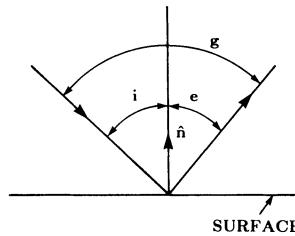


Fig. 3. Scattering geometry according to Horn.

follows that

$$\begin{aligned} -\bar{k}_1 * \bar{k}_2 &= k^2 \cos g \\ V^2 &= \bar{k}_1^2 + \bar{k}_2^2 - 2\bar{k}_1 * \bar{k}_2 \\ &= 2k^2(1 + \cos g) \\ V_z &= k(\cos i + \cos e). \end{aligned}$$

Horn defines  $I$ ,  $E$ ,  $G$  as the cosines of angles  $i$ ,  $e$ ,  $g$ , respectively. From the consideration above,

$$\begin{aligned} V^2 &= 2k^2(1 + G) \\ V_z^2 &= k^2(I + E)^2. \end{aligned}$$

Noting that

$$V_{xy}^2 = V^2 - V_z^2,$$

we finally obtain

$$\langle \rho \rho^* \rangle E_0^2 = \alpha \beta \Theta^2 e^{-\beta(2\Theta-1)} \quad (2)$$

where

$$\begin{aligned} \alpha &= \frac{R^2 A}{\pi r_0^2} \\ \Theta &= \frac{(1+G)}{(I+E)^2} \\ \beta &= \left( \frac{T}{2\sigma} \right)^2. \end{aligned}$$

Here  $E_0^2$  is proportional to the power scattered by a smooth plane of area  $A$ . This is given by

$$E_0^2 = \frac{k^2 A^2 I^2}{4n^2 r_0^2}$$

where  $r_0$  is the distance from the observer to the plane. Thus the result in (2) would be proportional to the scattered power from the rough surface.

If the illumination source is collimated (unidirectional) and located on the axis of the observation, then  $I = E$  independent of the surface normal and  $G = 1$ . In this case we find

$$P = \langle \rho \rho^* \rangle E_0^2 = \frac{\alpha \beta}{4I^4} e^{-\beta((1/I^2)-1)} \quad (3)$$

where  $\Theta = 1/2I^2$ . This is a very simple form which can be easily interpreted. The first observation is that  $\beta$  depends only on surface roughness. As the surface becomes rougher  $\beta$  decreases. The exponential term in (3) dominates the

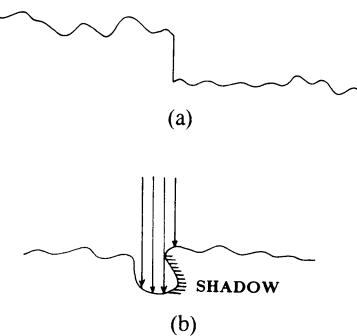


Fig. 4. (a) Nondifferentiable point on metal surface. (b) Shadow on metal surface.

behavior of  $P$  and, for large  $\beta$ , will lead to a rapid falloff in intensity for  $I \neq 1$ . Thus when  $\beta$  is small, (3) can be rewritten as

$$P = Ke^{-\beta((1/I^2)-1)}. \quad (4)$$

Equation (4) indicates that the reflective power  $P$  is a one-to-one function of the incident angle  $I$  if we limit the range of  $I$  to be within  $(0, 1)$ . Therefore, the local surface normal at each point can be determined from the reflective power uniquely.

When we use (4) as the basis of our experiments, we have to notice some special cases where (4) may fail. Fig. 4(a) and (b) show two of such cases. In Fig. 4(a) a point exists where it is nondifferentiable, and thus the so-called "incident angle" becomes meaningless. In Fig. 4(b) shadows exist, i.e., places which are not observable. This also makes (4) fail. Therefore, when we use (4) as the basis of our experiments, we need to assume that the surface is differentiable everywhere and observable everywhere. These assumptions also constitute some limitations to our approach of using image processing techniques to reconstruct the surface profile when compared with the conventional stylus instruments and laser scanners.

### B. Determination of Metal Surface Profile from Image Data

The data used in this study are the samples provided by General Electric. These samples were first taken as photographs using a microscope with magnification of 144. The photographs were then digitized into  $128 \times 128$  images using a video scanning system. The digitized images cover 128 gray levels.

When we took those metal surface photographs, we had to arrange the illumination source to be collimated and located on the axis of observation. This is the condition for the validity of (3). Besides, we were not able to make a direct measurement of the reflective power  $P$ . In our experiment, we measure the metal surface reflective power  $P$  in an indirect way, and the unit of the reflective power is "gray level" ( $0 \sim 127$ ) in the digitized images. To make things manageable, we assume that the brightest point (largest gray level) in the image has the surface normal located on the axis of the incident light ( $I = 1$ ). Based on this assumption and  $\beta$  for the given sample, the constant  $K$

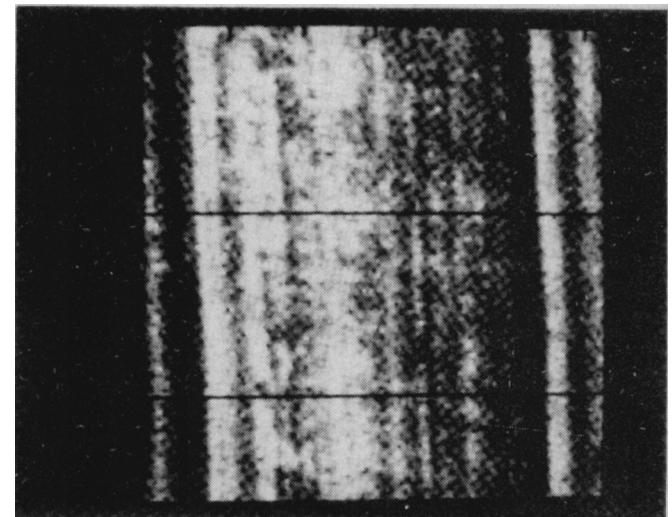


Fig. 5. Sample image used in metal surface profile experiments. Upper black line is  $x = 50$  and lower is  $x = 100$ .

in (4) can be obtained. A table showing the numerical relation between digitized gray level ( $0 \sim 127$ ) and incident angle  $I$  ( $0 \sim 1$ ,  $I = \cos$  of incident angle) can then be constructed. From this table, we can obtain the local surface normal at each point in the image.

After obtaining the local surface normal of the image, we can proceed to construct the surface profile of the image. We need only to construct the line profile for  $x = 0, 1, \dots, 128$  or  $y = 0, 1, \dots, 128$ . Here we choose the line profile for  $x = 0, 1, \dots, 128$ . Let  $I_{ij}$  be the cosine of the angle between the incident light and the local surface normal obtained at pixel  $(i, j)$ . Then the incident angle  $i_{ij}$  at pixel  $(i, j)$  is

$$i_{ij} = \cos^{-1} I_{ij}, \quad 0 \leq i_{ij} \leq \frac{\pi}{2}.$$

Thus the magnitude of the gradient at pixel  $(i, j)$  can be obtained by

$$\nabla_{xy}(i, j) = |\tan i_{ij}|.$$

The orientation of the gradient at pixel  $(i, j)$  can be obtained from gray level measurement as

$$\varphi_{ij} = \tan^{-1} \left( \frac{g_{ij} - g_{i,j-1}}{g_{ij} - g_{i-1,j}} \right)$$

where  $g_{ij}$  is the gray level at pixel  $(i, j)$ . The gradient at pixel  $(i, j)$  along the  $y$  direction is the projection of  $\nabla_{xy}(i, j)$  onto  $y$  direction, i.e.,

$$\nabla_y(i, j) = \nabla_{xy}(i, j) \cos \varphi_{ij}.$$

Finally, the surface profile along the line  $x = i$  is obtained by

$$\text{profile}(i, j) = \sum_{k=0}^j \nabla_y(i, k) \Delta y.$$

A low-pass filter was used to remove some noise in the images before we applied the foregoing procedures to reconstruct the metal surface profiles. The low-pass filter

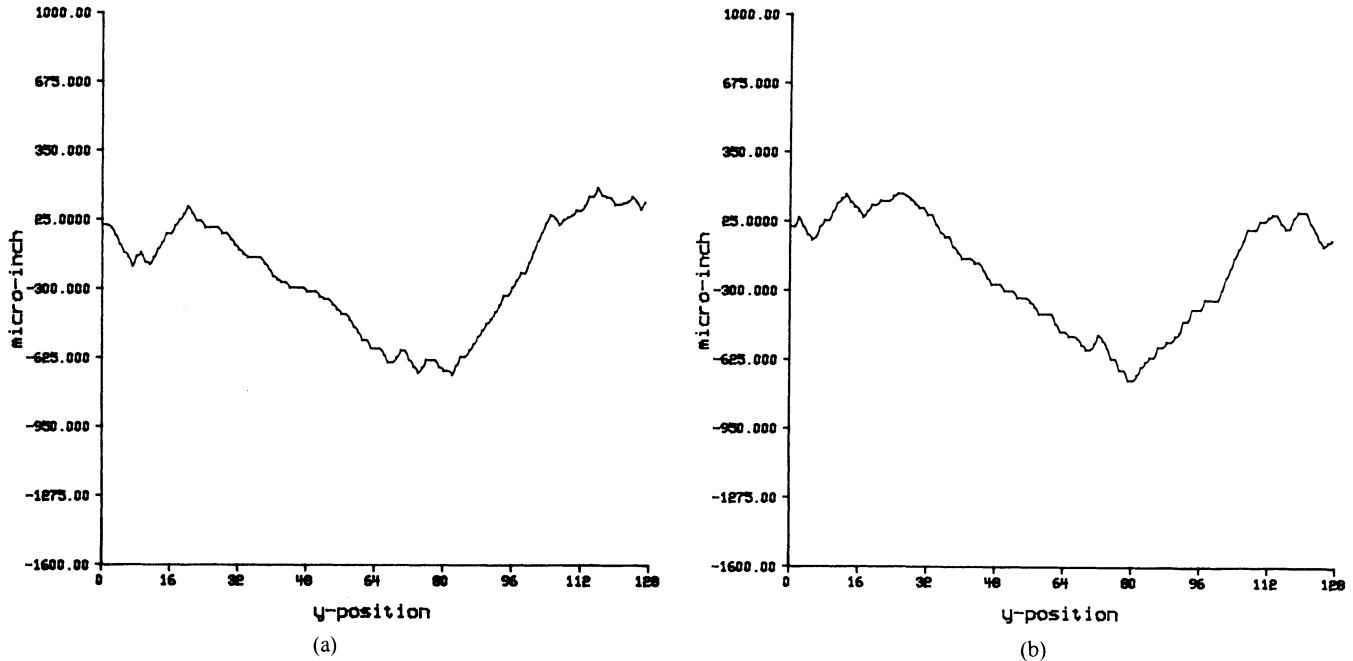


Fig. 6. (a) Reconstructed metal surface at  $x = 50$ . (b) Reconstructed metal surface at  $x = 100$ .

was implemented by taking a weighted average of the adjacent pixels for each pixel [6]. Fig. 5 shows one sample image used in our experiment. In this figure, two lines at  $x = 50$  and  $x = 100$  are marked by black lines. The reconstructed surface profiles along these two lines are shown in Fig. 6(a) and (b), respectively. As we can see, although these reconstructed surface profiles are not perfect, they can certainly be used for some inspection purposes if the quality requirement is not very high. In our experiment, we used a PDP11/45 computer. Each 128-point line profile took 55 ms of CPU time to digitize and 1.3 s to reconstruct the line profile.

### III. CLASSIFICATION OF SURFACE ROUGHNESS

In this section, we demonstrate the feasibility of applying pattern recognition techniques to automatic metal surface inspection. More specifically, we have applied a texture classification technique to classify metal samples into classes according to their surface roughness. In this approach, we do not directly measure metal surface profiles or reconstruct the metal surface profiles as we did in Section II. Instead, we extract some statistical properties (features) from metal surface images and apply pattern classification techniques to classify metal surfaces according to their statistical properties.

#### A. Preprocessing and Feature Selection

The data we used in this section are the same samples used in Section II. There are six classes of metal surfaces  $C_1, C_2, \dots, C_6$  with different roughness. The surface roughness of these six classes are 4, 8, 16, 32, 63, and 125 (average, in  $\mu$  in), respectively. These samples were first taken as photographs using a microscope with 25.6 magnification. Typical samples from each of the six classes are shown in Fig. 7. The photographs were then digitized and partitioned into 1864  $\times$  1024 images/ class. The image gray

scale was modified to cover just 64 gray levels and histogram equalization [6] was performed on each of the images. This was done in order to remove the effects of unequal overall brightness and contrast in the original images; these effects might otherwise dominate the measured feature vectors. The enhanced images were then used for the classification experiments.

To perform the classification experiments, we have selected the features obtained from the second-order gray level statistics [7]. Let  $\delta = (\Delta x, \Delta y)$  be a vector in the  $(x, y)$  plane. For any such vector and for any picture  $f(x, y)$ , we can compute the joint probability density of a pair of gray levels occurring at a pair of points separated by  $\delta$ . If there is only a finite number of gray levels (e.g.  $0, \dots, 63$ ), this joint density takes the form of an array  $h_\delta$ , where  $h_\delta(i, j)$  is the probability of a pair of gray levels  $(i, j)$  occurring at separation  $\delta$ . This array is  $m \times m$ , where  $m$  is the number of possible gray levels.

If picture  $f$  is discrete, it is easy to compute  $h_\delta$  array for  $f$ , where  $\Delta x$  and  $\Delta y$  are integers, by counting the number of times that each pair of gray levels occurs at separation  $\delta = (\Delta x, \Delta y)$  in the picture. As a very simple example, if the picture is

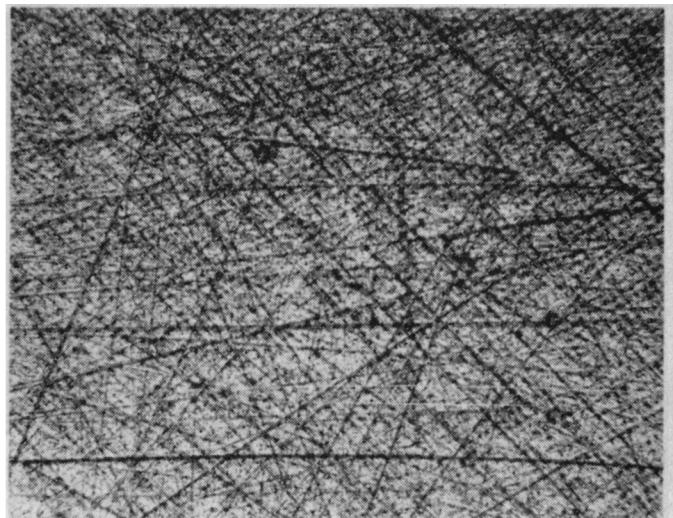
0	1	1	2	3
0	0	2	3	3
0	1	2	2	3
1	2	3	2	2
2	2	3	3	2

and  $(\Delta x, \Delta y) = (1, 0)$ , then  $h_\delta(i, j)$ ,  $i, j = 0, 1, 2, 3$ , are given in the following matrix

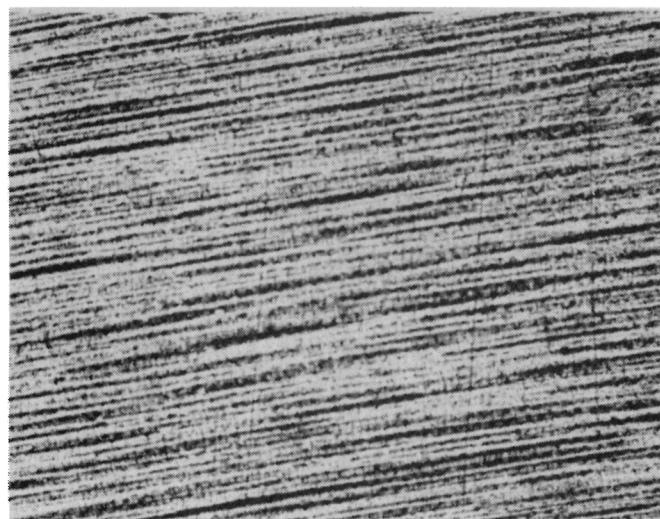
	0	1	2	3
0	1	2	1	0
1	0	1	3	0
2	0	0	3	5



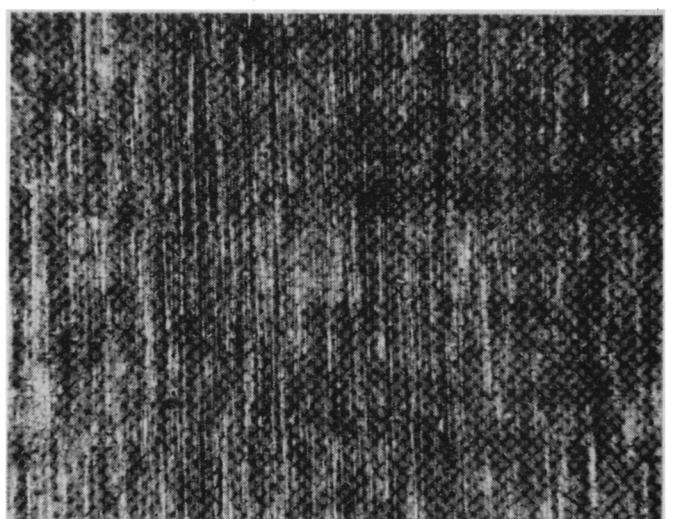
(a)



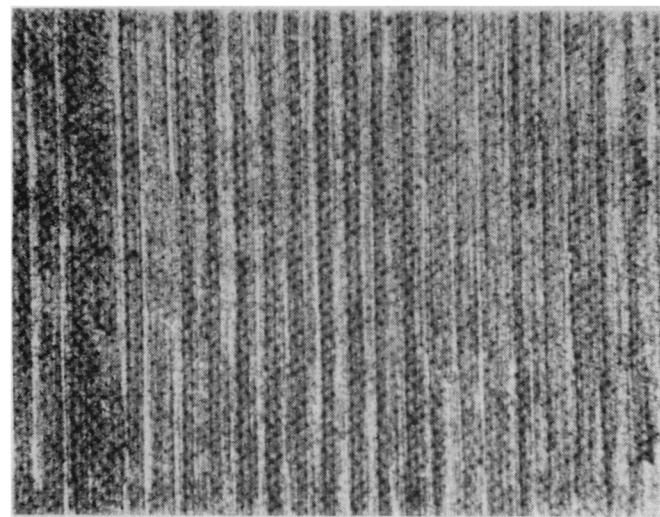
(b)



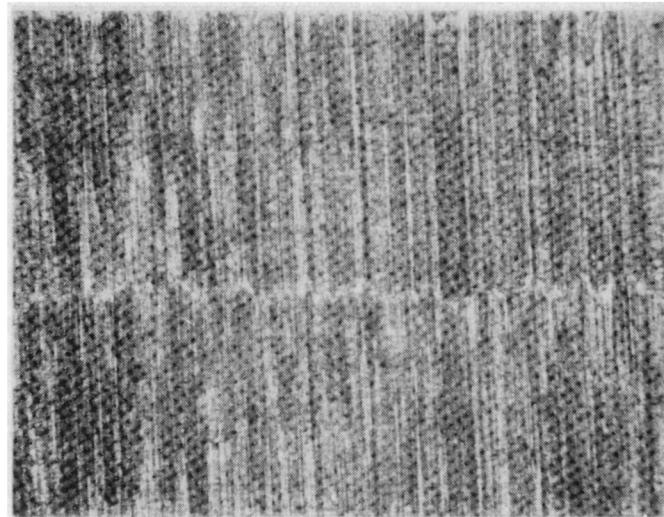
(c)



(d)



(e)



(f)

where the entry in row  $i$  and column  $j$  is the number of times that gray level  $i$  occurs immediately to the left of gray level  $j$ . A matrix of this form is sometimes called a gray-level co-occurrence matrix. It is sometimes convenient to use a symmetric matrix in which a pair of gray levels at separation either  $\delta$  or  $-\delta$  is counted. We shall denote this matrix by  $M_\delta$ .

Haralick [8] has proposed a variety of measures that can be employed to extract useful textural information from  $M_\delta$  matrices. The contrast measurement, defined by

$$\text{CON} = \sum (i - j)^2 p(i, j),$$

where  $p(i, j)$  is  $M_\delta(i, j)$  divided by the sum of all the matrix elements, is essentially the moment of inertia of the matrix around its main diagonal; it is a natural measure of the degree of spread of the matrix values. In our experiments, 16 contrast measures, which involve four spatial displacement (i.e., 1, 2, 4, 8) and four directions (degree = 0, 45, 90, 135) were used for classification.

### B. Design of Classifiers

1) *Linear Classifier:* A fixed increment error-correcting training procedure was used to construct a linear classifier [9]. Let  $y$ 's be the augmented feature vectors of the training samples and  $w$  be the weight vector of the linear classifier which is to be trained. The training procedure for a two-class problem is as follows (assuming  $w$  is initially a zero vector).

- 1) If a sample belongs to class 1 and  $y'w > 0$ , then do nothing.
- 2) If a sample belongs to class 1 and  $y'w \leq 0$ , then  $w = w + y$ .
- 3) If a sample belongs to class 2 and  $y'w \geq 0$ , then  $w = w - y$ .
- 4) If a sample belongs to class 2 and  $y'w < 0$ , then do nothing.

This procedure was applied to each of the samples in the two classes repeatedly until all of the samples can be correctly classified. If the number of iterations exceeds 1000 (or other suitable number), then the process would terminate when the percentage of the training samples that was correctly classified exceeds 95 percent. This was introduced because a few training samples might exist which were not linearly separable.

Since we have more than two classes, i.e., six classes  $C_1, C_2, \dots, C_6$ , we use a voting scheme to classify a given feature vector  $z$ . For each pair of classes  $C_i, C_j$ , we classify  $z$  as described earlier. If  $z$  is classified as a member of  $C_i$ , then the class  $C_i$  gets a vote. Finally, we assign  $z$  to the class that receives the largest number of votes.

2) *Tree Classifier:* In Section III-B1) we noticed that for an  $n$ -class problem, we need to compute  $C(n, 2)$  two-class classifiers for each test sample. For large  $n$ , this constitutes a heavy load. An alternative is to use a tree classifier [10]. The basic concept of a tree classifier is as follows. The sample space is first split into several groups, and each group is then split again into smaller groups. This process

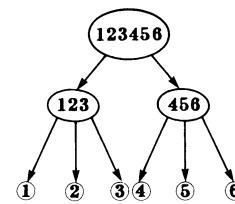


Fig. 8. Tree-classifier used in classification experiment.

continues until each group contains only one class. Several methods exist for the design of a tree classifier [11], [12]. The most straightforward one is to design the top (or "root") part of the tree classifier in an *ad hoc* fashion, then to use this partial tree to partition the set of training patterns into several groups, and to develop subtrees further until the task is completed. Another approach is to concentrate on the pathways which ought to be followed by each pattern class by blanking off most of the tree and noting how often a pattern attempts to follow a forbidden pathway. The performance of the tree classifier designed by these methods is subject to the experience of the designer, yet this approach provides a convenient method for designing a tree classifier. Other more complex tree classifier design methods are also discussed in [12].

It can be shown that for an  $n$ -class problem, at most  $n - 1$  classifiers need to be computed to classify a test sample. This is much more efficient than the voting scheme described in 1). Efficiency sometimes is not the only consideration for a tree classifier. In many situations, it may not be feasible to use a single set of features to classify all the sample classes, or it may be important to derive the evaluation of later features according to the results of earlier classifications. These types of situation are particularly relevant to the application of a tree classifiers.

In our classification experiment, we used the first method discussed earlier to design the tree classifier. Since our aim is to classify metal samples into classes according to their surface roughness, it is quite natural to split the sample space into two groups at the root node of the tree classifier so that one group contains the less rough samples, and the other group contains the rougher samples. The tree classifier we have designed is shown in Fig. 8. Two levels exist in this tree classifier and each node is implemented by a linear classifier as described in Section III-B1). The first level is a binary classifier which classified the test samples into two groups. The first group contains classes  $C_1, C_2$ , and  $C_3$ , and the second group contains classes  $C_4, C_5$ , and  $C_6$ . In the second level, these two groups are further classified into six classes.

### C. Experimental Results

Totally, 288  $64 \times 64$  sample images (48 samples for each class) were used. For each sample image, a 16-dimensional feature vector was extracted as described in Section III-A. We used the "sample-partitioning method" [9] in training and testing the classifiers. In each of the six classes, we used 24 sample images to train the classifier and the remaining 24 sample images to test the classifier. A PDP11/45 computer was used for image digitization and

preprocessing and a VAX11/78 computer for the classification. For each test sample, it took the CPU 3.6 s to digitize a  $64 \times 64$  sample image, 0.5 s to perform the histogram equalization, and 0.5 to extract feature vector.

Using a linear classifier, only 70 percent of the test samples could be correctly classified. This was due to the large deviation of roughness between the rougher classes and the less rough classes. When we used a microscope to take the pictures of those samples, we magnified them so that for those samples of larger roughness, the pictures might cover only a very small area on the sample. For example, it might only cover the bottom of a valley or the peak of a hill, but not the overall rough surface. Thus the images of the rougher samples might be similar to the images of the less rough samples. For this reason the designed classifier could not classify those samples correctly.

To overcome this difficulty, we have designed a tree classifier which is shown in Fig. 8. In the first level of this tree classifier, images with 25.6 magnification are used, and in the second level images with different magnifications are used. For the first group, we used the magnification of 25.6 and for the second group, we used the magnification of 11.2. Linear classifiers as described in Section III-B1) are implemented for each node of tree classifier. A correct classification rate of 97.5 percent was obtained at the first level of the tree classifier, and at the second level a correct classification rate of 95.8 percent was obtained within the first group and 96.3 percent within the second group. The overall correct classification rate is over 90 percent. This result is indeed very satisfactory and is suitable for many inspection purposes. Using a VAX11/780 computer to perform the tree classification task, the CPU took 34 ms to classify a test sample of which the feature vector had already been obtained from the preprocessing stage.

#### IV. DISCUSSION

In Section II we showed that theoretically a one-to-one correspondence exists between the local surface normal and the reflective surface power. Under the assumption that the illumination source is collimated and located on the axis of observation and  $\beta$  is small, we obtain the relation (4). Before we proceed to reconstruct the metal surface profiles, we have used a low-pass filter to remove noise in the surface images. Result shows that if the required quality of the surface profile is not critical to a specific application, this approach is indeed feasible. Since the table of incident angle versus surface normal can be constructed off-line, the total computation time for reconstructing a 128-point line profile can be reduced to less than 0.5 s.

From the theoretical and the experimental results, we realize that the key in this approach is an accurate function (or table) which can give us the direct correspondence between local surface normal and digitized image gray level. With this information, the remaining task will be straightforward. In practice, it is not feasible to measure parameters  $\alpha$  and  $\beta$  and use (4) to obtain such a correspondence,

since these parameters depend on lighting condition and metal surface properties which may change from case to case. Thus a more practical way is to combine a conventional method, i.e., the stylus instruments, laser scanner, etc., with the proposed image processing method. We can use a conventional method to scan one line profile and simultaneously use a vidicon system to digitize this line. From these measurements, we can obtain an accurate correspondence between local surface normal and image gray level. Then we can use this information to process the rest of the surface and reconstruct the surface profile by using the proposed image processing method. In this way, we can take the advantages of both methods, i.e., the accuracy of conventional method and efficiency of image processing method.

In Section III we have applied pattern recognition techniques to automatic metal surface inspection. Features are extracted from metal surface images by computing the contrast measurement from its co-occurrence matrix. The tree classifier implemented by nonparametric linear classifiers has correctly classified more than 90 percent of the samples in a testing set of 144 samples (24 samples/class). In practice, the classifiers used for the classification can be trained off-line. In Section III-C we noticed that most of the computation time was used for image digitization and preprocessing tasks, and very little computation time was spent on the classification task. This indicates that if we want to improve the overall speed of this method, we need to improve the speed of image digitization and preprocessing. The speed of image digitization is highly dependent on the input equipment, and it can be increased by using specially-designed hardware. The speed of image preprocessing can also be improved by using a faster computer or specially designed processor.

We also notice that metal surface roughness is a measure of surface height variations while metal surface image is a measure of the surface normal variations. In both approaches described in this paper, we used metal surface images to infer metal surface roughness. The successful results of our experiments show that surface height variations and surface normal variations are very closely related. These results can also be regarded as a verification of the theoretical result given by (3).

#### V. CONCLUSION

The feasibility of applying image processing and pattern recognition techniques to automatic metal surface inspection has been demonstrated. A metal surface profile can be reconstructed from its image by using a random surface scattering model. Metal surface roughness can also be classified without reconstructing its surface profile. This can be done by using a pattern recognition technique. Texture analysis techniques can be used to extract features from metal surface images. A tree-classifier implemented by linear classifiers is proposed for this purpose. The results of experiment on 288 pattern samples have shown very satisfactory performance.

## REFERENCES

- [1] E. C. Teague, "Evaluation, revision and application of the NBS stylus/computer system for the measurement of surface roughness," National Bureau of Standards, U.S. Department of Commerce, NBS Tech. Note 902, Apr. 1976.
- [2] J. Peters, P. Vanherck, and M. Sastrodinoto, "Assessment of surface typology analysis techniques," CIRP-AACIS Project, Kath Univ. Leuven, Dep. Werktuigkunde Afdeling Werkplaatsen en Industriële Beleid, Celestijnlaan 300B, B-3030 Heverlee (Belgium).
- [3] J. L. Mundy and G. B. Porter, III, "Visual inspection of metal surfaces," General Electric Corporate Res. Develop., Schenectady, NY.
- [4] P. Beckmann and A. Spizzichino, *The Scattering of Electromagnetic Waves from Rough Surfaces*. New York: Pergamon, 1968.
- [5] P. Winston, Ed., *The Psychology of Computer Vision*. New York: McGraw-Hill, 1975.
- [6] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic, 1976.
- [7] J. S. Weszka, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, Apr. 1976.
- [8] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Texture features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 610-621, Nov. 1973.
- [9] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1972, ch. 7.
- [10] W. S. Meisel, *Computer-Oriented Approaches to Pattern Recognition*. New York: Academic, 1972, ch. 1.
- [11] B. G. Batchelor, *Pattern Recognition, Ideas in Practice*. New York: Plenum, 1978, ch. 5.
- [12] P. H. Swain and H. Hauska, "The decision tree classifier: design and potential," *IEEE Trans. Geosci. Electron.*, vol. GE-15, July 1977.

# Correspondence

## Representations of Connectives in Fuzzy Reasoning: The View Through Normal Forms

I. B. TURKSEN AND DAVID D. W. YAO

**Abstract**—The representation of the fuzzy connectives AND, OR, NOT, and IF THEN is reinvestigated in the setting of fuzzy reasoning. This was first considered by Zadeh. Since then the appropriate representations of these connectives, particularly the implication statement, have been discussed in various papers. There are two main categories of approaches for the representation of the implication. In the first category the fuzzy implication is seen through the view of Lukasiewicz's multivalued logic. The second category, i.e., the view through interval-valued fuzzy sets of higher type is represented by Hisdal. Here, a new approach, i.e., the view through the fuzzy normal forms is presented that differs from both of the approaches stated above. The truth table approach and the degrees of truth associated with an implication are not our concern at the foundation. Our concern is based on the fact that the implication as well as other connectives are linguistic entities that are inescapably associated with fuzziness. We are interested in the manner in which they introduce, or rather, add fuzziness into the statement in which they are included. New insight is brought into the structure of the "generalized modus ponens" by our representation."

### I. INTRODUCTION

In this work we study the representation of the fuzzy connectives AND, OR, NOT, and IF THEN in the setting of fuzzy reasoning, where they play a critical role. This representation problem is first considered by Zadeh [10]–[12], where AND, OR, and NOT are defined, respectively, as the intersection, union, and complement

Manuscript received November 24, 1981; revised January 1983, and July 1983. This work was supported in part by NSERC under Grant A7698 and in part by the University of Toronto Fellowship program.

The authors are with the Department of Industrial Engineering, the University of Toronto, ON, Canada.

of fuzzy sets, while the implication statement IF  $x = A$  THEN  $y = B$  is represented by a certain relation between the fuzzy sets  $A$  and  $B$ . Zadeh also proposed a "compositional rule" of fuzzy inference. With this rule, the consequence of the "generalized modus ponens" (GMP) can be deduced as follows:

*Antecedent 1:* IF  $x = A$  THEN  $y = B$

*Antecedent 2:*  $x = P$

*Consequence:*  $y = P \circ (A \rightarrow B)$

where,  $(A \rightarrow B)$  denotes the relation induced by IF  $A$  THEN  $B$ . When  $P = A$ , GMP reduces to modus ponens (MP) in the usual sense, although the entries  $A$  and  $B$  are fuzzy sets. (For details of definitions here and below, see preliminaries presented in Section II.)

Since then a vast amount of papers (refer to [1], [2], [4], [5], [6], [8], [9] and references included there) have been published that discuss the appropriate representations of the connectives, particularly the implication statement. This is mainly due to the fact that neither of the two representations originally proposed by Zadeh, namely,

$$(A \rightarrow B) = (A \times B) \cup (\bar{A} \times V) \quad (1)$$

$$(A \rightarrow B) = (\bar{A} \times V) \oplus (U \times \bar{B}) \quad (2)$$

is deemed satisfactory, since they will not always give the consequence  $y = B$  in the case of MP, i.e.,  $A \circ (A \rightarrow B) \neq B$  in general. Approaches to the representation of the implication can be classified into two main categories. The first category covers numerous authors ([1], [2], [4], [5] are perhaps among the most representative ones) who see the fuzzy implication through the view of Lukasiewicz multivalued logic. They interpret the entries of IF  $A$  THEN  $B$  (expressed in the form of a fuzzy relation) as the truth values of the implication itself (i.e., the grade of the implication being "true"). Approaches through this view are essentially extensions of the "truth table" approach of the traditional "crisp" or Boolean logic. For instance, Baldwin [1] proposes the concept of "truth value restrictions," where the 0-1