NORTH-HOLLAND

A Model-Based Machine Vision System Using Fuzzy Logic

Kil-jae Lee and Zeungnam Bien

Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Taejeon, Korea

ABSTRACT

An effective model-based machine vision system is designed for use in practical production lines. In the proposed system, the gray level corner is selected as a local feature, and a gray level corner detector is developed. The gray level corner detection problem is formulated as a pattern classification problem to determine whether a pixel belongs to the class of corners or not. The probability density function is estimated by means of fuzzy logic. A corner matching method is developed to minimize the amount of calculation. All available information obtained from the gray level corner detector is used to make the model. From a fuzzy inference procedure, a matched segment list is extracted, and the resulted segment list is used to calculate the transformations between the model object and each object in the scene. In order to reduce the fuzzy rule set, a notion of overlapping cost is introduced. To show the effectiveness of the developed algorithm, simulations are conducted for synthetic images, and an experiment is conducted on an image of a real industrial component.

KEYWORDS: ***gray level corner, fuzzy logic pattern classification, corner matching, hypothesis and verify***

1. INTRODUCTION

In robotic applications, a key role of the industrial vision system is to identify and locate relevant objects for the robots involved. In a typical industrial scene, the objects may be intermixed and/or partially occluded. To handle such a complicated scene, a segmentation procedure is often required to divide an image into the region of objects and the region of

Address correspondence to Zeungnam Bien, Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, 373-1, Kusong-dong, Yusong-Ku, Taejeon, 305-701, Korea.

Received July 1, 1996; accepted October 1, 1996.

International Journal of Approximate Reasoning 1997; 16:119–135

0888-613X/97/$17.00
655 Avenue of the Americas, New York, NY 10010
PII S0888-613X(96)00117-8

background, but this procedure is often error-prone and time-consuming. In most machine vision applications, this segmentation procedure is facilitated by arranging the environment in such a way that the background is uniform and is contrasted with respect to the objects. Then a simple thresholding technique is introduced to complete the segmentation. But in typical production lines there are uncertainties due to changing illumination, nonuniform background, and shadow, and also it is sometimes costly to arrange the environment. Thus, a robust machine vision system is needed for locating objects in an uncertain environment.

There have been many investigations into solving the *location problem* of occluded objects. The location problem can be handled in two stages: the first is local feature extraction, and the second is local feature matching. Local features such as corners, protrusions, holes, lines, and curves can be used in recognizing and identifying objects even in the presence of occlusion. The methods of extracting the local features are divided into two categories: binary level feature extraction and gray level feature extraction. The binary level feature extraction methods, such as polygonal approximation [1], require a segmentation procedure. On the other hand, the gray level feature extraction methods, such as gray level corner detection techniques [2–4], can extract features directly from a gray image with no segmentation procedure.

After extracting local features, matching between local features in a scene and local features in a model is necessary. Many matching algorithms have been developed to reduce the amount of calculation, which increases explosively as the number of local features increases. Neural networks [5], relaxation algorithms [6], the Hough transform [7], the maximal-clique-finding method [16], and hypothesis-and-verify methods [8, 9] are used to solve the matching problem. However, it is difficult to apply these methods in industrial applications, since they still require large computations.

In this paper, an effective method to solve the location problem for occluded objects will be developed. A gray level corner detection method is proposed which can utilize knowledge about corners to detect them. And a matching method is developed to minimize the amount of calculation. To handle the uncertainties of feature values, fuzzy logic is adopted to construct the fuzzy rule bases which contain the knowledge about model object. The overall design is focused on the development of a model-based machine vision system which can be used effectively in various industrial applications such as locating components to be assembled in a production line.

The paper is organized as follows. An algorithm for gray level corner detection and one for corner matching are described in Section 2. Simulation and experimental results are described in Section 3. The conclusion follows in Section 4.

2. DESIGN OF A MODEL-BASED MACHINE VISION SYSTEM

2.1. Gray Level Corner Detection

A corner is associated with two conditions: the occurrence of two consecutive edges, and some significant change in the edge directions. Thus, the measure of cornerity can be expressed as a function of the edge strength and the gradient of edge direction. Therefore, the input feature vector $X = (x_1, x_2)$ is constructed with the component x_1 being the edge strength and x_2 being the gradient of edge direction. To reject the spurious edge direction information due to noise, the gradient of edge direction is cut off when the edge strength is very weak. The patterns of feature vectors X are divided into a corner class denoted by c_0 and a noncorner class denoted by c_1. Thus, if certain information about the corners to detect is available, the corner detection problem can be cast into a pattern classification problem.

Let us consider the Bayes classifier [10] that minimizes the total average loss as follows:

If $L_{01}P(X \mid c_0)P(c_0) \geq L_{10}P(X \mid c_1)P(c_1)$,
then X *belongs to the class* c_0.

Here L_{ij} is the loss incurred from the decision that X belongs to c_j when it actually belongs to c_i, and $P(X \mid c_i)$ is the conditional probability density function. Because there is no relationship between the selected features and noncorner class, $P(X \mid c_1)$ is assumed to be some constant. Also, L_{01}, L_{10}, $P(c_1)$, and $P(c_0)$ are constants. Then the above statement can be rewritten as follows:

If $P(X \mid c_0) \geq \{L_{10}/L_{01}\}\{P(c_1)/P(c_0)\}P(X \mid c_1) = T$,
then X *belongs to the class* c_0.

Here T may be interpreted as a threshold level.

Thus, the corner detection problem can be formulated as an estimation problem for $P(X \mid c_0)$. In estimating the probability density function, fuzzy logic will be used in the paper. Some justifications are as follows: First, the fuzzy inference procedure possesses a good interpolation capability as well as a good approximation capability [11]. Second, fuzzy logic can handle uncertainties of the feature vector due to discretization and noise [12]. And third, it is easy to implement in a parallel hardware.

Suppose that m patterns $X_p = (x_{1p}, x_{2p})$, $p = 1, 2, \ldots, m$, are given as the training data set and these patterns belong to c_0. Our problem is to estimate the conditional probability density $P(X \mid c_0)$ from these data by using the fuzzy logic. As input membership functions, the second order B-spline functions form [13] is used because of its simplicity. Let the universe of discourse be $[0, L]$, and further let each axis of the feature vector space be partitioned into K fuzzy sets $\{A_1, A_2, \ldots, A_K\}$, where A_i is defined by the following membership function $\mu_i(x)$:

$$\lambda_i = \begin{cases} 0 & \text{if} \quad i = 0, \\ \dfrac{L}{K-1}(i-1) & \text{if} \quad 1 \le i \le K, \\ L & \text{if} \quad i = K+1, \end{cases} \tag{1}$$

$$N_i^1(x) = \begin{cases} 1 & \text{if} \quad \lambda_i \le x \le \lambda_{i+1}, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

$$N_i^2(x) = \frac{x - \lambda_{i-1}}{\lambda_i - \lambda_{i-1}} N_{i-1}^1(x) + \frac{\lambda_{i+1} - x}{\lambda_{i+1} - \lambda_i} N_i^1(x), \tag{3}$$

$$\mu_i(x) = N_i^2(x). \tag{4}$$

Figure 1 shows the input membership functions.

To get the output of a multivariate membership function, the output of each univariate membership function is calculated, and then combined by using the product operator:

$$\mu_{ij}(X) = \mu_i(x_1)\mu_j(x_2). \tag{5}$$

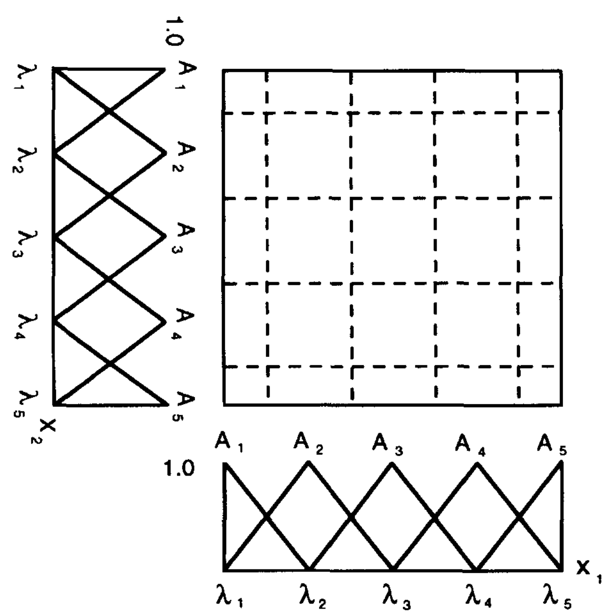


Figure 1. Input membership function.

As a fuzzy rule for estimating the probability density function, the following rule is used:

If x_1 is A_i and x_2 is A_j,
then X belongs to the corner class c_0 with weight w_{ij}.

As a method to get the weight w_{ij} from the given patterns $X_p = (x_{1p}, x_{2p})$, $p = 1, 2, \ldots, m$, a basic nonparametric estimation method [10] is used. Let the area enclosed by a region R be denoted as V. Then, consider first the probability P of vector X falling in the region R. When the region R is so small that P does not vary much within it, then

$$P = \int_R p(X')\, dX' \approx p(X)V. \tag{6}$$

The probability that k out of m samples drawn independently fall into R can be estimated by k/m. Thus,

$$P \approx \frac{k}{m}. \tag{7}$$

Combining (6) and (7), $p(X)$ can be estimated by the following equation [10]:

$$p(X) \approx \frac{k/m}{V}. \tag{8}$$

From the above argument, w_{ij} can be determined by the following equation:

$$w_{ij} = \frac{k_{ij}}{mV_{ij}}, \tag{9}$$

where m is the number of patterns, and k_{ij} is the number of patterns in which x_1 is the support of A_i, and x_2 is in the support of A_j. V_{ij} is the area of the support of A_{ij}.

The fuzzy inference procedure in [13] is employed to compute the estimated output y as follows:

$$y = \frac{\sum_{i=1}^{K} \sum_{j=1}^{K} \mu_i(x_1)\mu_j(x_2) w_{ij}}{\sum_{i=1}^{K} \sum_{j=1}^{K} \mu_i(x_1)\mu_j(x_2)}. \tag{10}$$

Since the B-spline functions satisfy the relationship

$$\sum_{i=1}^{K} \sum_{j=1}^{K} \mu_i(x_1)\mu_j(x_2) = 1, \tag{11}$$

Equation (10) can be rewritten as follows:

$$y = \sum_{i=1}^{K} \sum_{j=1}^{K} \mu_i(x_1)\mu_j(x_2)w_{ij}. \tag{12}$$

It is noted also that the number of fired rules is always 4, and hence the fuzzy inference procedure can be implemented in parallel hardware at low cost. Figure 2 shows the structure of the developed probability density function estimator.

2.2. Corner Matching Algorithm

To recognize and locate objects, a matching procedure should be followed after a local feature extraction. In the previous section, it was shown that the gray level corners can be selected as local features, and these corners can be localized to corner points. These corners usually contain enough information to recognize and local objects. The model objects also have corner points, and by matching these points in the model with corner points in a scene, it is possible to recognize and local objects.

Matching a template of points in a model with the points in a real image is a difficult task, since there can be many exceptional cases to consider [16]. The first is the case when there are too many feature points in the scene because of multiple instances of the chosen types of the objects in the image. Secondly, additional points may be present because of noise or clutter from irrelevant objects and structure in the background. The third case is when certain points that should be present are missing because of

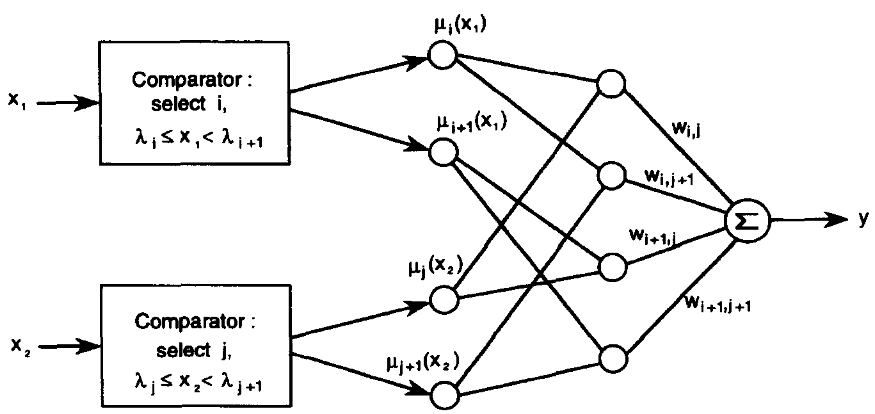


Figure 2. Structure of probability density function estimator.

noise or occlusion, or because of defects in the objects being sought. And finally, feature values from corner detectors may have uncertainties associated with their numerical properties. In designing the matching procedure, the above cases should be taken into account.

Many matching algorithms have been developed to reduce the amount of calculation because the amount of calculation increases explosively with an increase of local features. The maximal-clique-finding algorithm [16] is a mathematical method to solve the point matching problem. An association graph is introduced to solve this problem systematically. The association graph is made of the nodes which represent feature assignments, and arcs joining nodes which represent pairwise compatibilities between assignments. By finding a maximal clique of the association graph, the matching problem can be solved. But the problem of finding a maximal clique is not simple; indeed, it is NP-complete [16].

The graph matching process can be formulated as an optimization problem [17]. An energy function may be devised so that this function represents the constraints that the nodes in the two graphs have to satisfy in order to find the best match. A 2-D binary Hopfield neural network [5] or a relaxation algorithm [6] can be used to minimize the energy function. But because these methods need iterative operations, there must be considerations about convergence, and they may take a long time.

Another approach is the hypothesis-and-verify method [8, 9]. By using privileged features and a similarity measure, a hypothesis can be generated. By calculating match confidence, the hypothesis can be verified. The hypothesis-and-verify method can reduce the amount of calculation remarkably. But the hypothesis generation scheme is complex, and false hypothesis may cause the system to spend a long time in verification.

In this paper, an effective corner-matching algorithm is proposed by using all the information available from gray level corner detectors and fuzzy logic to accommodate the uncertainties of features. Improbable situations can be eliminated in an early stage if all available information such as position of corner, angle of corner, and angle change of corner is utilized.

A segment consists of two corners. Let the corner points of the model object be given by a set $\{\mathrm{MC}_1, \mathrm{MC}_2, \ldots, \mathrm{MC}_p\}$, where p is the number of corners in the model object. The information on a corner MC_i is given by a vector $(x_i, y_i, \alpha_i, d\alpha_i)$, where x_i and y_i are the coordinates of the position of a corner, α_i is the angle at the corner, and $d\alpha_i$ is the angle change at the corner. From two corners, MC_i and MC_j, a feature vector of a segment MS_k connecting these corners can be calculated. The feature vector of a segment MS_k is represented as (l_k, da_k, db_k, dc_k), where l_k is the length of the segment, da_k and db_k are the two angle changes at the two corners, and dc_k is angle difference between the two corners as shown in Figure 3. The features of the segment can be calculated as follows:

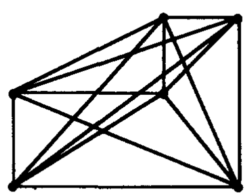
$$l_k = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{13}$$

$$da_k = d\alpha_i, \tag{14}$$

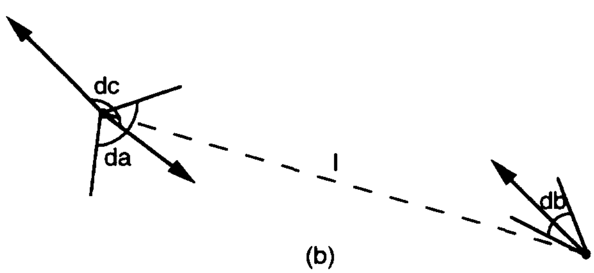
$$db_k = d\alpha_j, \tag{15}$$

$$dc_k = |\alpha_i - \alpha_j|. \tag{16}$$

By connecting all the corner points of the model object, a model segment set $\{MS_1, MS_2, \ldots, MS_P\}$ is obtained as shown in Figure 3(a), where $P = p(p-1)/2$. To represent the model object, the feature vectors of model segments are used. But these feature vectors have uncertainties due to discretization noise, geometric distortion, and feature calculation error. To accommodate these uncertainties, the fuzzy logic is adopted to represent these feature vectors. The feature vector of segment MS_i can be represented as (L_i, DA_i, DB_i, DC_i), where L_i, DA_i, DB_i and DC_i are fuzzy numbers, which are defined by the triangular membership functions $\mu_{L_i}(l)$, $\mu_{DA_i}(da)$, $\mu_{DB_i}(db)$ and $\mu_{DC_i}(dc)$ as shown in Figure 4. By using



(a)



(b)

Figure 3. The features of segment: (a) all possible segments of the model, (b) the features of a segment.

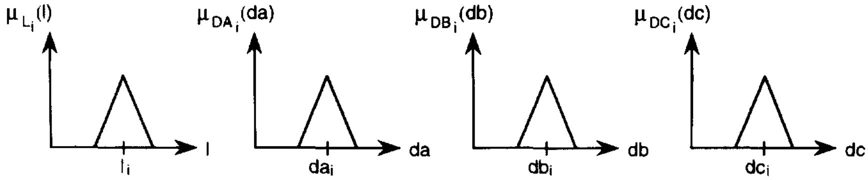


Figure 4. The membership functions to represent the features.

these feature vectors, the model object can be represented as P fuzzy rules as follows:

If the length is L_i, and the two angle changes are either DA_i or DB_i, and the angle difference is DC_i,
then the segment is MS_i, $i = 1, 2, \ldots, P$.

So the model object can be represented by assigning these fuzzy rules to all model segments.

By using the model represented by the fuzzy rules, matching segments can be extracted. A matched segment pair is made up of a model segment and a scene matching with the model segment. After gray level corner detection, corners of a scene are obtained as a set $\{\mathrm{SC}_1, \mathrm{SC}_2, \ldots, \mathrm{SC}_q\}$, where q is the number of corner points in the scene. From this scene corner set, a scene segment list $\{\mathrm{SS}_1, \mathrm{SS}_2, \ldots, \mathrm{SS}_Q\}$ can be obtained, where $Q = q(q-1)/2$. From a scene segment SS_j, a segment feature vector, represented as (l_j, da_j, db_j, dc_j), can be calculated with the same calculating procedure as for the model segment feature vector. After some fuzzy inference routine is executed for each segment feature vector, a matching degree between a model segment and a scene segment can be calculated. To calculate the matching degree between the model segment MS_i and the scene segment SS_j, the following equation is used [15]:

$$\mu_{\mathrm{MS}_i}(\mathrm{SS}_j) = \min\Big\{ \mu_{L_i}(l_j), \max\big(\mu_{\mathrm{DA}_i}(da_j), \mu_{\mathrm{DB}_i}(db_j)\big), \mu_{\mathrm{DC}_i}(dc_j)\Big\}. \quad (17)$$

Here, $\mu_{\mathrm{MS}_i}(\mathrm{SS}_j)$ represents the matching degree between the model segment MS_i and the scene segment SS_j. This fuzzy inferencing procedure is summarized in Figure 5. After calculating all the matching degrees between the model segments and the scene segments, a matched segment list (MSL) is obtained by thresholding with some threshold T:

$$\mathrm{MSL} = \Big\{(\mathrm{SS}_j, \mathrm{MS}_i)\Big| 1 \le j \le Q, 1 \le i \le P, \mu_{\mathrm{MS}_i}(\mathrm{SS}_j) \ge T\Big\}. \quad (18)$$

In an ideal case of one object, the number of matched segments is P, but this case is rare because a scene segment can be interpreted as multiple segments of the model. Also, there may be multiple instances of the model

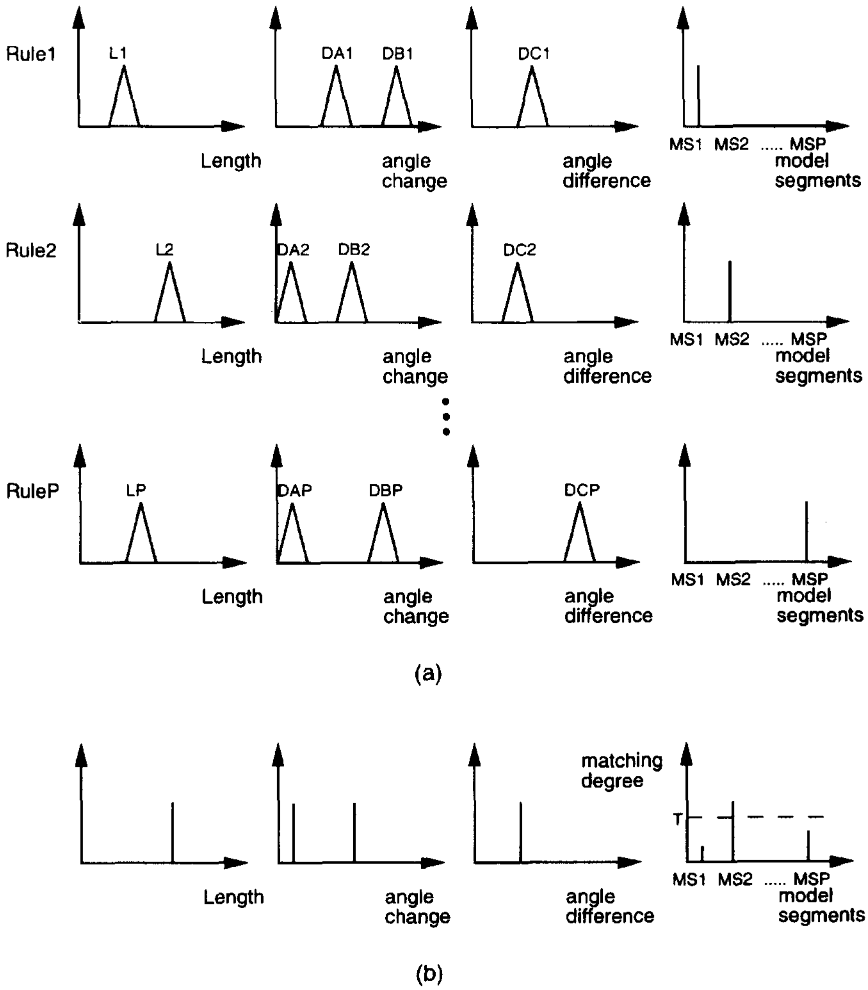


Figure 5. Fuzzy inference procedure to extract matched segment list: (a) fuzzy rules representing model segments, (b) feature values of a scene segment and the resulting matching degree.

object in a scene. To divide the matched segment list into the matched segments of each object, some grouping technique is necessary. From extracted matched segment list, transformation parameters made up of rotation and translation between a scene segment and the matched segment can be calculated [5]. Because the segments of an object have similar parameter values, a single pass clustering technique [14] in the transformed parameter space can group the segments of each object. A clustering procedure then renders the position and orientation of each object [5].

2.3. Selection of Privileged Segments

In general, all the model segments are used to represent a model. Because this set of model segments can be too big to extract a matched segment list in a short time, a reduction of the set of model segments is often needed. Moreover, when the number of rules is large, the case that a scene segment is interpreted as multiple segments of the model often occurs. These multiple matched segments may confuse the grouping procedure. So reducing the number of rules is important in shortening the time for matching.

The privileged segments are the segments selected so as to represent the model [8], and coded as the rules. But there is no structured method to select the privileged segments. In this paper, a cost is introduced to select the privileged segments. In this paper, a cost is introduced to reduce the likelihood that a scene segment is interpreted as multiple segments of the model. If there is no overlapping between a model segment and the other model segments, then a scene segment corresponding to the model segment cannot be interpreted as multiple segments of the model. So, if model segments that do not overlap with other model segments are selected as the privileged segments, it is possible to reduce the length of the matched segment list.

The overlapping cost is the extent to which a model segment and the other model segments overlap. To calculate the overlapping cost, a possibility measure [15] is used. The possibility of A with respect to B, $\Pi(A \mid B)$, is defined [15] as

$$\Pi(A \mid B) = \max_{x}\{\min(A(x), B(x))\}. \tag{19}$$

The possibility measure of A with respect to B reflects the extent to which A and B coincide or overlap. The overlapping cost of model segment MS_j, $OVC(MS_j)$, can be calculated by the following equation:

$$\begin{aligned} OVC(MS_j) = \sum_{1 \le i \le P} \Big[&\Pi(L_i \mid L_j)\Pi(DC_i \mid DC_j) \\ &\times \max\{\Pi(DA_i \mid DA_j), \Pi(DB_i \mid DB_j)\}\Big]. \end{aligned} \tag{20}$$

By selecting the privileged segments as model segments having low values of the overlapping cost, it is possible to reduce the time for matching.

3. SIMULATION AND EXPERIMENTAL RESULT

To test the effectiveness of the developed gray level corner-detecting algorithm, a simulation is first conducted. Figure 6(a) shows a synthetic image which contains a nonuniform background and two objects. The

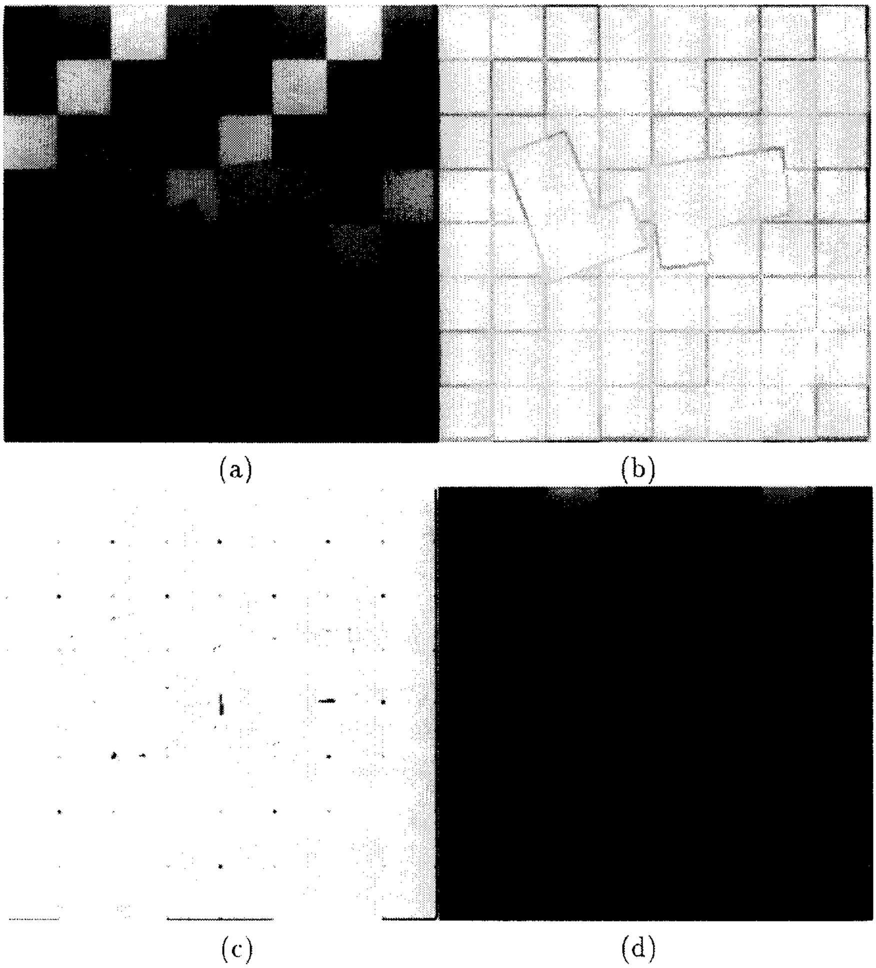


Figure 6. The corner detection example with synthetic image: (a) original image, (b) image of edge strength, (c) image of angle change, (d) corner image using the proposed corner detector.

contrast of the synthetic image is reduced progressively from top to bottom. The image is then smoothed by convolution with a gaussian noise ($\sigma = 0.5$) to simulate the blurring introduced by the imaging system. Figure 6(d) shows that only the corners of objects can be detected.

Secondly, to test the effectiveness of the developed corner-matching algorithm, two simulations have been conducted. First, a simulation was conducted to test the possibility of reducing the matched segment list by using information on the segment feature vector. In the case of using only the information on the segment length, the number of matched segments is 69. In case of using all available information such as length, change of angle, and difference of angle, the number of matched segments is 27. This shows that more information can result in a smaller matched segment list. Second, a simulation was conducted to test the effectiveness of the overlapping cost. Figure 7(a) shows the model segment list made up of boundary segments, and Figure 7(c) shows the resulted matched segment list. The number of matched segments is 33. Figure 7(b) shows the model segment list made by using the overlapping cost, and Figure 7(d) shows the resulting matched segment list. The number of matched segments is 27. This shows that the model segment list made by using the overlapping cost can reduce the matched segment list less than the one made up of boundary segments.

To test the effectiveness of the developed corner-matching algorithm, an experiment was conducted. Figure 8 shows the result of the experiment. Figure 8(a) shows eight model segments which are extracted from the model object by using the overlapping cost. Figure 8(b) shows information on corners extracted by using the gray level corner detector and a localization procedure. As shown there, corners of scene objects have a lot of uncertainties due to discretization noise and shadow. The recognized results are represented by arrows on the scene image. Figure 8(c) and (d) show that the developed corner matching algorithm works well in the presence of partial occlusion and shadow. Also Figure 8(e) and (f) show that the algorithm works well even in the presence of nonuniform background.

4. CONCLUDING REMARKS

It is shown in this paper that, in certain cases, the corner detection problem can be reformulated as a pattern classification problem, and a simple pattern classifier is developed for real-time gray level corner detection. By training the pattern classifier with the data on the corner class, a fuzzy rule base for gray level corner detection is obtained. The weights of the rules can be modified by new training data in a simple one-pass operation.

To reduce the matched segment list effectively, it is shown that all the available information on a segment can be used to represent its feature vector. Also, the concept of an overlapping cost is introduced. Some simulation results show that the resulting matched segment list is small when all the available information on segments and the notion of overlapping cost are used. Using the resulting matched segment list, the position and orientation of each object can be calculated effectively. By adjusting

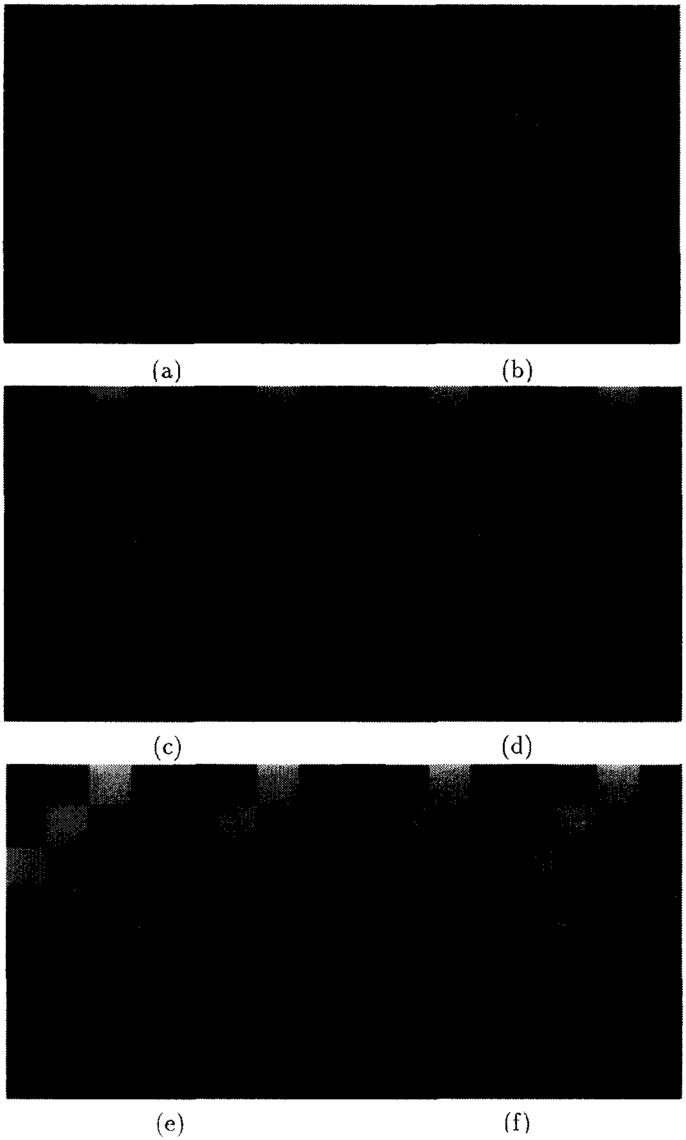
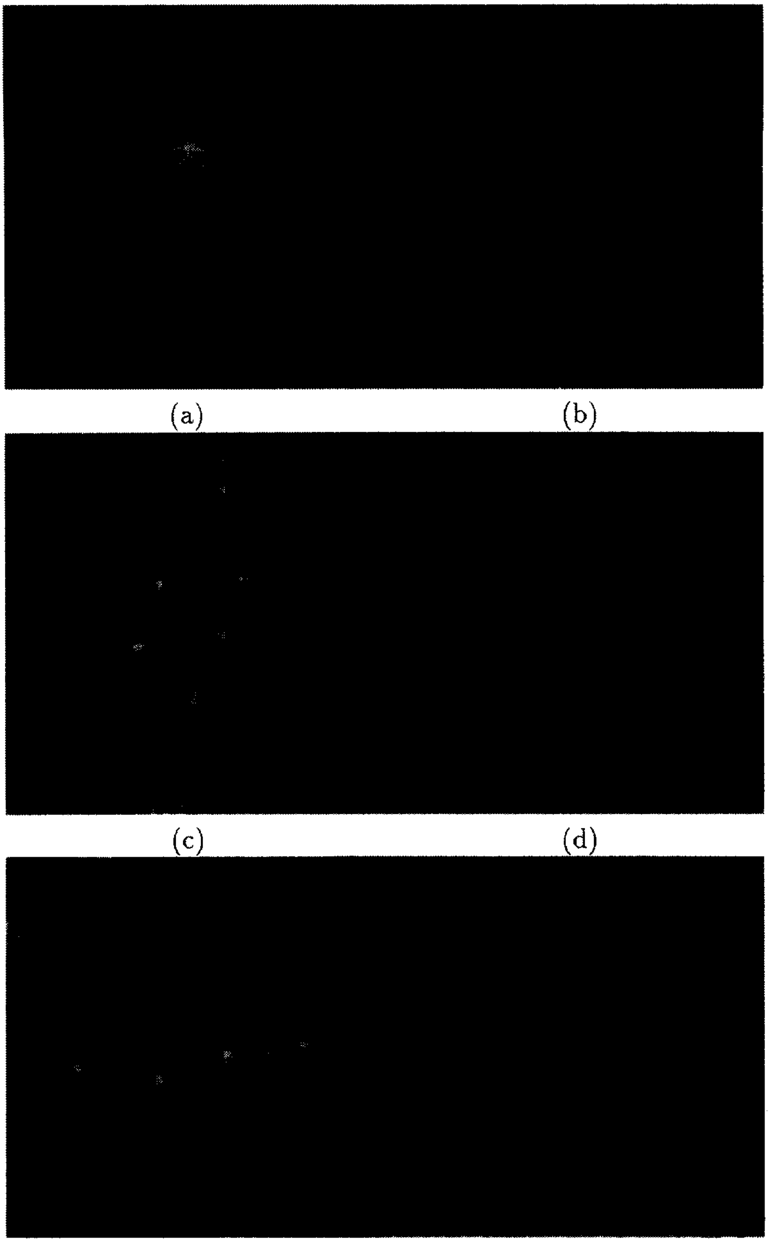


Figure 7. The result of corner matching with synthetic image: (a) model segment list made up of boundary segments; (b) model segment list using overlapping cost; (c) the resulting matched segment list using (a); (d) the resulting matched segment list using (b); (e) matched segments after transformation and clustering; (f) calculated position and orientation.



(a) (b)
(c) (d)
(e) (f)

Figure 8. Recognition results using the developed algorithm: (a) model segments; (b) detected corners; (c)–(f) recognition results.

the input membership functions of the fuzzy rule base, the amounts of uncertainties in feature values can be incorporated into the developed vision system.

This corner matching procedure can be easily extended to matching of other local features such as holes and curves. It is remarked that, because it is difficult to know the amounts of uncertainties, an automatic tuning algorithm of the fuzzy rule base needs to be studied. The developed algorithm is expected to be useful for real applications such as locating components to be assembled in production lines.

References

1. Koch, M. W., et al., Using polygons to recognize and locate partially occluded objects, *IEEE Trans. Pattern Anal. Machine Intell.* 9, 483–494, 1987.
2. Kitchen, L., and Rosenfeld, A., Gray-level corner detection, *Pattern Recognition Lett.* 1, 95–102, 1982.
3. Singh, A., and Shneier, M., Grey level corner detection: A generalization and a robust real time implementation, *Comput. Vision Graphics Image Process.* 51, 54–59, 1990.
4. Cooper, J., Venkatesh, S., and Kitchen, L., Early jump-out corner detectors, *IEEE Trans. Pattern Anal. Machine Intell.* 15, 823–828, 1993.
5. Nasrabadi, N. M., and Li, W., Object recognition by a Hopfield neural network, *IEEE Trans. Systems, Man Cybernet.* 21, 1523–1535, 1991.
6. Rutkowski, W. S., Recognition of occluded shapes using relaxation, *Comput. Vision Graphics Image Process.* 19, 111–128, 1982.
7. Li, Z. N., et al., Linear generalized Hough transform and its parallelization, *Comput. Vision Graphics Image Process.* 11, 11–24, 1993.
8. Ayache, N., and Faugeras, O. D., HYPER: A new approach for the recognition and position of two-dimensional objects, *IEEE Trans. Pattern Anal. Machine Intell.* 8, 44–54, 1986.
9. Bouyakhf, E. H., An approach for recognizing and locating overlapping industrial parts, *Internat. J. Pattern Recognition Artificial Intell.* 2, 673–689, 1988.
10. Duda, R. O., and Hart, P. E., *Pattern Classification and Scene Analysis,* Wiley, 1973, pp. 85–195.
11. Sudkamp, T., et al., Interpolation, completion, and learning fuzzy rules, *IEEE Trans. Systems Man Cybernet.* 24, 332–342, 1994.
12. Bezdek, J. C., *Pattern Recognition with Fuzzy Objective Function Algorithms,* Plenum, 1981, pp. 1–13.

13. Harris, C. J., et al., *Intelligent Control: Aspects of Fuzzy Logic and Neural Nets*, World Scientific, 1993, pp. 314–357.

14. Richards, J. A., *Remote Sensing Digital Image Analysis*, Springer-Verlag, 1994, pp. 236–239.

15. Pedrycz, W., *Fuzzy Control and Fuzzy Systems*, Wiley, 1989, pp. 1–59.

16. Davies, E. R., *Machine Vision: Theory, Algorithms, Practicalities*, Academic, 1990, pp. 345–368.

17. Sonka, M. et al., *Image Processing, Analysis and Machine Vision*, Chapman and Hall Computing, 1993, pp. 299–307.