Proceedings of the 1998 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Victoria, B.C., Canada • October 1998

A Learning Fuzzy Decision Tree and its Application to Tactile Image

Han-Pang Huang and Chao-Chiun Liang
Robotics Laboratory
Department of Mechanical Engineering
National Taiwan University
Taipei, Taiwan 10674, R.O.C.

ABSTRACT

Decision trees play important roles in many fields such as pattern recognition and classification. It is because they have simple, apparent and fast reasoning process. This paper develops an algorithm to generate a learning fuzzy decision tree. This algorithm firstly collects enough training data for generating a practical decision tree. It then uses fuzzy statistics to calculate fuzzy sets for representing the training data in order to save computing memory and increase generation speed. Finally, this algorithm uses a sub-optimal criterion to learn a decision tree from the resultant fuzzy sets. The algorithm is applied to a general-purpose tactile force sensing system. This system uses fuzzy logic to interpolate the force data. Then, the proposed algorithm is used to generate the desired decision tree from the tactile data. Based on the decision tree, the objects can be on-line recognized precisely.

1. Introduction

The decision tree is a well-known decision structure [7]. A binary tree is one class of decision trees [5]. Decision trees can be applied to many fields [1,5] such as pattern recognition, classification, decision support system, expert system, etc. A decision tree has several nodes arranging in hierarchical structure. Each of its nodes has different function and meaning. Fig.1 is an example of a decision tree applying to a classification problem in a two-dimensional feature space, where each object to be classified is represented by a two-dimensional feature. In Fig.1, the condition $g_i(X)\geq 0$, i=1~4, is the decision function or splitting rule to split the corresponding region into two parts and to decide which part the object should belong to. The label C_j, j=I~V, is one of the names of the predefined classes; the ellipse containing $g_0(X)\geq 0$ is the root node where the classification begins; the ellipse containing $g_i(X)\geq 0$, i=1~3 is the internal node where the classification continues; the square containing Ci, i=I~V, is the terminal node where the classification stops.

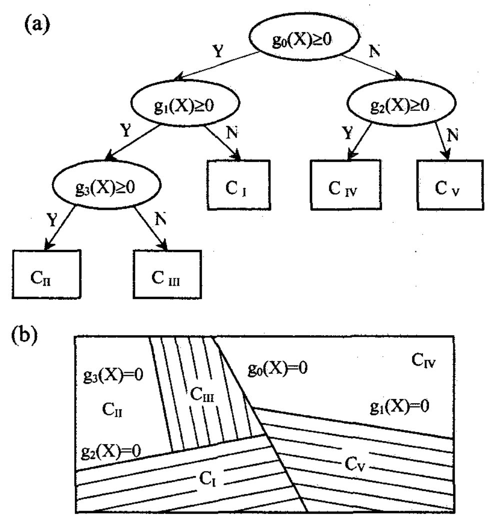


Fig. 1 An example of a decision tree. (a) Its structure. (b) The hierarchical partition of feature space.

From the example in Fig.1, we can easily see that the decision tree implements decision or classification in a simple, apparent, and multistage manner. Furthermore, since each node of a decision tree uses only a simple splitting rule and a small subset of all features, the whole decision process is fast and efficient [5]. It is interesting to note that decision trees can be mapped into a class of artificial neural network or entropy nets with far few connections [7]. In some applications of prediction and recognition, decision trees have comparable performance with artificial neural networks [1].

To generate a practical decision tree, numerous training data are necessary. If the number of training data is enormous, it requires long processing time and large memory space to generate a useful decision tree. This paper uses fuzzy statistics to generate several fuzzy sets from training data. It intends to meaningfully represent the statistic distribution of training data, increase computation speed, and save storage memory.

Given a set of training data, an optimal decision tree is inherently difficult to obtain. This is because (a) optimizing a node's performance does not necessarily optimize the performance of the whole tree, and (b) the possible number of tree topologies is extremely large even for moderate number of classes and a moderate number of features [5]. This paper adopts a sub-optimal splitting rule at each node to generate a decision tree. This is a cost-effective method of quickly obtaining a decision tree with satisfactory performance.

Experiments suggest that human tactile perception system can fast and accurately recognize real world objects [8]. With rapid advances in tactile sensor technology, there are more and more tactile sensors applied to numerous regions, such as biomedicine, computer peripherals, industry [12], etc. This paper also constructs a fuzzy logic based tactile sensing system. As tactile perception is the human inherent capacity and fuzzy is the nature of human reasoning, it is straightforward to apply fuzzy logic [2, 6, 10, 11] to increase the resolution of a tactile sensor. This is a cost-effective method and can integrate human expertise as well as fulfill nonlinear interpolation.

To examine the performance of the proposed learning fuzzy decision tree, the proposed algorithm is applied to the tactile sensing system to classify several geometric objects and human hands.

The organization of the paper is as follows. A learning fuzzy decision tree is developed in section 2. The tactile sensing system and the fuzzy interpolation algorithm are described in section 3. In section 4, the proposed learning fuzzy decision tree is applied to the tactile sensing system to classify several geometric objects and human hands. Finally, several concluding remarks are drawn.

2. A Learning Fuzzy Decision Tree

This paper uses the following steps to obtain a practical decision tree for classifying objects: (1) selecting a set of features that can well describe the objects and then massively measuring every feature of each object to get sufficient training data, (2) applying fuzzy statistics to find fuzzy sets for compressing and representing the training data, (3) learning a decision tree from the fuzzy sets. The above mentioned three steps are described in the subsequent context.

2.1 Features Selection

The application in this paper is to classify objects from their tactile images. By mathematical transforms, a lot of useful features can be extracted from the tactile image [3]. This paper selects complex moment invariants as features. The advantages of complex moment invariants are (1) invariant to translation and rotation, (2) simpler and more powerful than other moment features in object recognition [4]. The complex moment invariants are derived below.

The definition of continuous complex moment is:

$$C_{pq} = \iint (x+iy)^p (x-iy)^q g(x,y)\,dxdy$$

where p and q are nonnegative integers; g(x, y) is the gray level function of the tactile image. Because the resolution of the tactile sensor is limited, the above formula is modified into discrete complex moments as:

$$C_{pq}^{d} = \sum_{j=1}^{n} \sum_{k=1}^{m} (j\Delta x + ik\Delta y)^p (j\Delta x - ik\Delta y)^q g(j\Delta x, k\Delta y)\Delta x\Delta y$$

The equation can be simplified by setting

$$\Delta x = \Delta y = 1$$

The simplified equation is:

$$C_{pq}^{ds} = \sum_{j=1}^{n} \sum_{k=1}^{m} (j+ik)^p (j-ik)^q g(j,k)$$

where (j,k) represents the coordinates, relative to the corresponding center of pressure for each grid of tactile image. A set of complex moment invariants then can be obtained as [4]:

$$S_1 = C_{11}^{dsc} \qquad S_7 = \left|C_{40}^{dsc}\right|^2$$

$$S_2 = \left|C_{20}^{dsc}\right|^2 \qquad S_8 = \left|C_{31}^{dsc}\right|^2$$

$$S_3 = \left|C_{30}^{dsc}\right|^2 \qquad S_9 = C_{22}^{dsc}$$

$$S_4 = \left|C_{21}^{dsc}\right|^2 \qquad S_{10} = \left(C_{40}^{dsc}\right)^* \left(C_{31}^{dsc}\right)^2 + C.C.$$

$$S_5 = \left(C_{30}^{dsc}\right)^2 \left(C_{21}^{dsc}\right)^* + C.C. \qquad S_{11} = C_{31}^{dsc} \left(C_{20}^{dsc}\right)^* + C.C.$$

$$S_6 = \left(C_{21}^{dsc}\right)^2 \left(C_{20}^{dsc}\right)^* + C.C.$$

where the * represents complex conjugate, and $C.C.$ denotes the complex conjugate of its previous term.

2.2 Fuzzy Sets for Compressing Training Data

This paper adopts LR type fuzzy sets [2] and uses fuzzy statistics [9] to determine several points on their membership functions. The curve fitting method and the pre-determined points are used to identify the LR type fuzzy sets.

2.2.1 Determining Several Points on a Fuzzy Set

The training data is denoted as $T=\{T_1, T_2, \ldots, T_n\}$, where $T_i=\{T_{i1}, T_{i2}, \ldots, T_{im}\}$ is the training data for the object i, and T_{ij} is the training data for the feature j of the object i. The goal is to find fuzzy sets F_{ij} to represent T_{ij}, i=1, 2, …, n, j=1, 2, …, m. The algorithm for estimating points on F_{ij} is outlined as follows:

(1) The standard deviation and the mean of T_{ij} are calculated as SD_{ij} and M_{ij}, respectively.

(2) The number of points, (2n+1), to be identified is determined.

(3) For M_{ij}, $M_{ij}\pm(1/n)SD_{ij}$, $M_{ij}\pm(2/n)SD_{ij}$,…, $M_{ij}\pm SD_{ij}$, determine their membership grades, $\mu_{ij}(k)$, with respect to F_{ij} by using

$$\mu_{ij}(k)=S_{ij}(k)/S_{ij}(0)$$

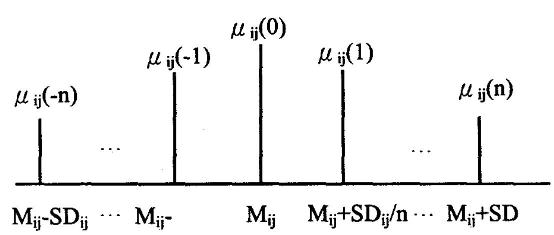


Fig. 2 2n+1 points on a fuzzy set F_{ij}

where $S_{ij}(k)$ is the number of data located in $[k- SD_{ij} /n, k+ SD_{ij} /n]$, and $k=0, \pm 1, \pm 2, \ldots, \pm n$. From Fig. 2 we can easily understand that $\mu_{ij}(k)$'s are the desired points.

This algorithm is valid under the assumptions that F_{ij} is a symmetric, normal fuzzy set, and the probability distribution of training data has the property $S_{ij}(0) > S_{ij}(k)$, $\pm 1, \pm 2, \ldots, \pm n$. Because the fuzzy set is symmetric, only n+1 points need to be estimated.

2.2.2 Identifying a LR Type Fuzzy Set

A fuzzy set M is of LR-type if there exist reference functions L (for left) and R (for right), and scalars a>0, b>0 with

$$\mu_M = \begin{cases} L(\frac{m-x}{a}) & \text{for } x \le m \\ R(\frac{x-m}{b}) & \text{for } x \ge m \end{cases}$$

where m is the mean value of M [2]. Define L and R as

$$L = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$
$$R = b_0 + b_1 x + b_2 x^2 + \cdots + b_n x^n$$

Since we have determined in the previous sub-section n points for L and R respectively, the coefficients $a_0, a_1, \ldots, a_n$ and $b_0, b_1, \ldots, b_n$ can be computed by curve fitting method.

2.3 Learning a Decision Tree

The steps of learning a decision tree are stated in the following:

Step1. Select a set of features suitable for classification as described in section 2.1 and massively measure each feature to get sufficient training data.

Step2. Find the fuzzy sets F_{ij}, $i=1\sim n$, $j=1\sim m$, to represent training data, as depicted in section 2.2. Use the fuzzy sets to form fuzzy pattern vectors $F_{pvi} = (F_{i1}, F_{i2}, \ldots, F_{im},)$ for representing object i, $i=1\sim n$.

Step3. Build the root node and start the learning process. The root node contains all objects represented by F_{pvi}, $i=1\sim n$.

Step4. Find the most powerful feature for the present node. A feature is the most powerful feature if it can split the node into the most sub-nodes. If more than one feature have the same power, the feature requiring the least computation time is chosen.

Step5. Split the present node into several sub-nodes by a splitting rule and the feature selected in step 4.

According to a heuristic threshold T and the distances between the objects, the splitting rule clusters the objects in the present node into several groups. These groups are the content of the desired sub-nodes. The heuristic threshold T is specific to the selected feature and the current level (root node is level 0). It is defined by the following equation:

$$T = M_{SD} \times \frac{L_{MAX} - L_{current}}{2}$$

where M_{SD} is the mean of the standard deviation of the selected feature; L_{MAX} and $L_{current}$ are the maximum allowable level and current level of the decision tree under design respectively. It is easy to see that the above equation is reasonable by considering the following facts:

- The splitting process with a larger threshold is more rigorous, accurate, and slow than that with a smaller threshold.
- When the standard deviation of the selected feature of any object increases, M_{SD} will also increase and the threshold T should increase in order to guarantee a satisfactory and correct classification rate.
- When the splitting rule applies to the deeper node, $L_{current}$ will increase and the threshold T should become smaller in order to reach terminal nodes more easily with insignificant decrease of correct classification rate. This insignificant decrease is due to fewer objects in the deeper node.

The distance between objects i and j can be viewed from each feature and is defined by the following equation:

$$d_k(i,j) = \alpha\left|M_{ik} - M_{jk}\right| + \beta\left|SD_{ik} - SD_{jk}\right|$$
$$\alpha + \beta = 1 \qquad \alpha \text{ and } \beta \text{ are real}$$

where $d_k(i,j)$ is the distance between objects i and j, viewing from feature k; M_{ij} and SD_{ij} are defined in section 2.2; α and β are heuristically selected weighting factors for indicating the relative importance between M_{ij} and SD_{ij}. The above definition has the advantage of considering both M_{ij} and SD_{ij}. The experiment presented in section 4 sets α and β as 0.9 and 0.1 respectively. It is reasonable that α should be larger than β. Several values of α and β may be tried in order to get a good result.

A node is a terminal-node if it satisfies stopping criterion, otherwise it is a non-terminal node. A node satisfies stopping criterion if it

- contains only one fuzzy pattern vector, or
- contains more than one fuzzy pattern vector that are too similar to split.

Step6. For each non-terminal node, repeat steps 4 and 5 until there are no non-terminal nodes.

Step7. For every terminal node containing more than one fuzzy pattern vector, the next powerful feature is linearly combined to split it. This step repeats until all terminal nodes contain only one fuzzy pattern vector.

Based on the above steps, a practical decision tree can be obtained. The decision tree will generate a piecewise smooth decision surface. Please note that a decision tree is learned in batch manner from the training data in this paper. Namely, the proposed algorithm processes all the training data and then generates a decision tree by determining its structure and parameters. The parameters of a decision tree include the objects and the threshold in each node. When there is any change in training data, even a small change, the proposed algorithm must re-process all the training data to obtain a new decision tree. This is equivalent to forgetting everything and then re-learning. Although the learning manner is not smart, the proposed algorithm is so efficient that it can overcome this drawback.

In a decision tree, a path from the root node to one of the terminal nodes will contain several nodes. The splitting rules in the nodes must simultaneously be satisfied so that the terminal node can be reached from the root node. Using 'and' operator, those splitting rules can be combined to form a new rule for representing the path. Therefore, a decision tree can be mapped into a rule-based system with each path representing by a rule. Especially, the fuzzy decision tree in this paper can be mapped into a fuzzy rule-based system. As every rule will be checked in a fuzzy rule-based system, a fuzzy decision tree reaches a result about n-times faster than its corresponding fuzzy rule-based system, where 'n' is the number of rules (or paths). For the same reason, however, a fuzzy decision tree is less reliable than its corresponding fuzzy rule-based system. Another reason for this less reliability is that there is fuzzyness in each non-terminal node and the fuzzyness will be accumulated until reaching a terminal node. By selecting suitable threshold values, limiting the deep, and collecting good training data for a decision tree, the problem of fuzzyness can be diminished and reliable results can be obtained with speed still much faster than its corresponding fuzzy rule-based system.

3. A Tactile Sensing System

The tactile sensing system constructed in this paper is shown in Fig. 3.

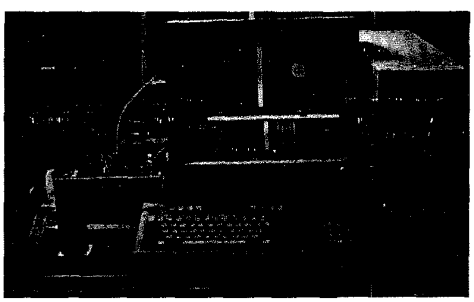


Fig. 3 A tactile sensing system

3.1 Hardware Devices

The important hardware devices in Fig. 3 are described briefly below:

- Personal Computer: The personal computer in Fig. 3 is 486 DX4-100 with 16 MB RAM. It can receive tactile data through RS 232 serial port, display their relative magnitude in terms of different colors on the screen, interpolate those tactile data, and perform intelligent tasks such as classification and recognition.
- Tactile Sensing Device: The black pad in the left-bottom of Fig. 3 is a 16×16 tactile sensing matrix manufactured by Interlink Electornics [12]. It consists of 16×16 = 256 FSRs whose resistance will change with pressure on them. The changes in resistance are the raw tactile data. The size of each FSR is 0.2"$\times$0.2".
- Interface Device: The device in the left-middle part is an interface device made by our laboratory. It reads tactile data from tactile sensing matrix and on-line sends them to personal computer through RS 232 port.

3.2 Fuzzy Interpolation

This paper uses a two-dimensional fuzzy number [2] to depict the pressure information carried by the tactile data at each grid of a tactile sensor, as shown in Fig. 4.

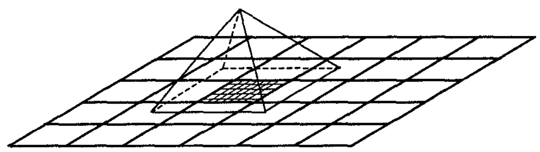


Fig. 4 A fuzzy number depicting the pressure information

The fuzzy number in Fig. 4 is a regular pyramid with its top vertex located at the center of the gird, and its four bottom vertices at the centers of diagonally adjacent grids. It reflects the fact that the closer a point to the center of the grid, the closer their pressure value. The fuzzy number also makes an assumption about the pressure distribution in terms of its geometric shape. Different geometric shapes may be suitable to different applications and should be designed according to experience and expertise.

Using the fuzzy number in Fig. 4, every point in a tactile sensor will have four fuzzy numbers to cover it. To interpolate the pressure value of a point by the fuzzy interpolation, we should calculate its membership grades with respect to its four covering fuzzy numbers and treat the resultant membership values as weightings. The pressure value of this point can be computed as the weighting sum of the pressure values of its four adjacent grids. The computation steps are formulated as follows:

(1) For a given point, read the pressure values of its four adjacent girds and assign them as

$$m_i, i = 1, 2, \ldots, 4$$

(2) Calculate the membership values of the point with respect to the fuzzy numbers of its four adjacent grids and assign them as

$$v_i, i = 1, 2, \ldots, 4$$

(3) The pressure value of the point v_s is calculated as

$$v_s = \frac{\sum_{i=1}^{4} m_i v_i}{\sum_{i=1}^{4} m_i}$$

4. Experiments

This section applies the proposed decision tree to a tactile image classification problem for examining its performance.

4.1 Experiment Procedure

The procedures of experiment are described below:

(1) Select the objects to be classified. The real objects selected in this experiment are circle, square, rectangle, and hand. For testing the capacity of orientation recognition, the same hand and rectangle in different orientation are regarded as different objects. Therefore, the logical objects used in this experiment are circle, square, rectangle in 0°, rectangle in 27°, rectangle in 45°, hand in 0°, and hand in 45°. Table 1 lists the object names and their corresponding class names.

Table 1 Object names and their corresponding class names. Row 1 contains class names, and row 2 contains object names.

C_1	C_2	C_3	C_4	C_5	C_6	C_7
Rectangle in 0°	Rectangle in 27°	Rectangle in 45°	Circle	Square	Hand in 0°	Hand in 45°

(2) Choose the complex moment invariants described in section 2.1 as the features in this experiment.

(3) Massively measure the features of the objects in (1) to get sufficient training data. In this experiment, every feature of each object is measured 50 times at different positions. The mean values of the measured training data are summarized in Table 2.

(4) Learn a decision tree from the training data by using the algorithm proposed in section 2.3. The resultant decision tree is shown in Fig. 5.

(5) Install the decision tree in (4) into the tactile sensing system depicted in section 3.

(6) Place an object on the tactile sensing device, measure its tactile image, and on-line interpolate the tactile image.

(7) On-line classify the object on the tactile sensing device.

Table 2 Mean values of the training data

Features\Objects	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Feature 1: S_1	27.55	22.50	32.18	6.72	23.49	972.12	804.93
Feature 2: S_2	512.47	237.06	625.92	1.94	1.05	1.92e5	1.59e5
Feature 3: S_3	2.49	21.05	42.49	1.90	3.11	1.19e6	1.71e6
Feature 4: S_4	1.87	2.37	9.96	0.12	1.40	5.81e6	5.19e6
Feature 5: S_5	2.49	41.21	212.25	0.37	4.68	4.62e9	6.06e9
Feature 6: S_6	58.61	55.49	412.59	0.08	2.37	5.03e9	4.04e9
Feature 7: S_7	7088.0	1566.4	9512.8	2.56	358.88	7.70e7	1.07e8
Feature 8: S_8	1.35e4	4497.7	1.91e4	10.81	18.83	4.12e8	4.01e8
Feature 9: S_9	129.68	82.85	157.84	10.78	63.37	3.83e4	3.29e4
Feature: S_{10}	2.28e6	3.65e5	3.78e6	32.46	577.80	7.06e12	8.03e12
Feature 11:	5260.1	2060.9	6918.1	9.10	8.83	1.77e7	1.58e7

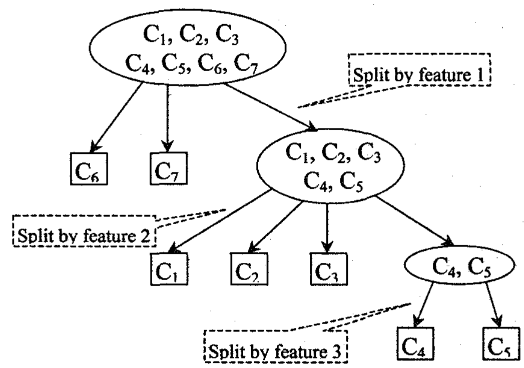
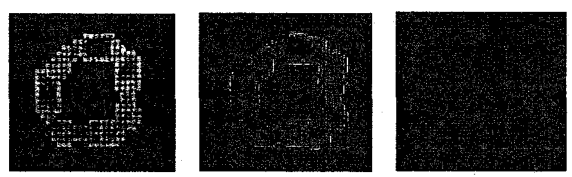


Fig. 5 The resultant decision tree

4.2 Experiment Results

The tactile images of a test object after fuzzy interpolation of several resolutions are shown in Fig. 6.



(a) (b) (c)

Fig. 6 Tactile images after fuzzy interpolation. (a) 3-time interpolation, (b) 5-time interpolation, (c) 7-time interpolation.

The correct classification rates of the objects are listed in Table 3.

Table 3 Correct Classification Rate

Objects Name	Correct Number	Error Number	Correct Classification Rate
Circle	47	3	94 %

Square	40	10	80 %
Rectangle in 0°	46	4	92 %
Rectangle in 27°	48	2	96 %
Rectangle in 45°	44	6	88 %
Hand in 0°	45	5	90 %
Hand in 45°	44	6	88 %

4.3 Discussions

Observing the boundaries and pressure distribution of tactile images in Fig. 6, it is easy to see that fuzzy interpolation can give smooth and reasonable results. On the other hand, the accuracy of fuzzy interpolation depends on the accuracy of the fuzzy numbers assigned to sensor grids. Theoretically, if the assigned fuzzy numbers are exact, the tactile image can be interpolated to infinite resolution with exact accuracy. Physically, the resolution of the fuzzy interpolation, however, depends on available computer memory.

The correct classification rates in Table 3 are satisfactory but not perfect. The reasons for imperfect are the limited resolution of the tactile sensor and the variable contact condition between objects and the tactile sensor. The limited resolution causes the selected features of an object can not remain the same even for slight change in position and orientation. The variable contact condition makes the selected features of an object can not remain unchanged even for the same position and orientation.

5. Conclusions

This paper uses fuzzy sets to represent and compress training data in order to save computation time and storage space. By learning the fuzzy sets, the proposed tree generation algorithm can rapidly generate a decision tree. To examine the performance of the algorithm, the algorithm is installed in a tactile sensing system and conducts an experiment of tactile image classification. The generated decision tree can perform classification with satisfactory correct rate.

As the resolution of a tactile sensor is limited, the complex moment invariants can not be exact invariant under translation and rotation. This is a drawback that may make tactile image more difficult to classify. This paper, however, makes use of this drawback to recognize the orientation of an object.

The heuristic threshold is used in the proposed tree generation algorithm. It reflects the probability distribution of the training data and hence increases the correct classification rate. As a tree grows deeper and deeper, the heuristic threshold will become smaller and smaller and hence reduce the tree size because terminal nodes are easier to obtain with smaller thresholds. Therefore, the heuristic threshold makes the generated tree close to optimal in the sense of correct classification rate and tree size.

6. References

[1] L. Atlas, R. Cole, Y. Muthusamy, A. Lippman, J. Connor, D. Park, M.E. Sharkawi, and R.J. Marks II," A Performance Comparison of Trained Multilayer Perceptrons and Trained Classification Trees," *Proceedings of the IEEE*, Vol. 78, No. 10, pp. 1614-1619, October 1990.

[2] D. Duboid and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press Inc., 1980.

[3] J. Jurczyk and K.A. Loparo," Mathematical Transforms and Correlation Techniques for Object Recognition Using Tactile Data," *IEEE Trans. on Robotics and Automation*, Vol. 5, No. 3, pp. 359-362, June 1989.

[4] R.C. Luo and H.H. Loh," Tactile Array Sensor for Object Identification Using Complex Moments," *Inter. Conf. on Robotics and Automation*, pp. 1935-1940, 1987.

[5] Y. Park and J. Sklansky," Automated Design of Linear Tree Classifiers," *Pattern Recognition*, Vol. 23, No. 12, pp. 1393-1412, 1990.

[6] W. Pedrycz," Fuzzy Sets in Pattern Recognition: Methodology and Methods," *Pattern Recognition*, Vol. 23, No. 1-2, pp. 121-146, 1990.

[7] I.K. Sethi," Entropy Nets: From Decision Trees to Neural Networks," *Proceedings of the IEEE*, Vol. 78, No. 10, pp. 1605-1613, October 1990.

[8] S.A. Stansfield, "Primitives, Features, and Exploratory Procedures: Building a Robot Tactile Perception System," *Inter. Conf. on Robotics and Automation*, pp. 1274-1279, 1986.

[9] P.Z. Wang, "From the Fuzzy Statistics to the Falling Random Subsets," *Advances on Fuzzy Sets Theory and Applications*, P.P. Wang (ed.) Pergamon Press, pp. 81-95, 1983.

[10] L.A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Trans. Syst., Man, Cyber.*, SMC-1, pp. 28-44, 1973.

[11] H.J. Zimmerman, *Fuzzy Set Theory and its Application*, Kluwer Publishing, 1985.

[12] "*What is a Force Sensing Register™*", Interlink Electronics