International Journal of Fuzzy Logic and Intelligent Systems, vol. 3, no. 1, June 2003 pp. 52-57

Fuzzy Classifier System for Edge Detection

Kwee-Bo Sim

School of Electrical and Electronic Engineering, Chung-Ang University
221, Huksuk-Dong, Dongjak-Ku, Seoul, 156-756, Korea
E-mail : kbsim@cau.ac.kr, URL: http://alife.cau.ac.kr

Abstract

In this paper, we propose a Fuzzy Classifier System(FCS) to find a set of fuzzy rules which can carry out the edge detection. The classifier system of Holland can evaluate the usefulness of rules represented by classifiers with repeated learning. FCS makes the classifier system be able to carry out the mapping from continuous inputs to outputs. It is the FCS that applies the method of machine learning to the concept of fuzzy logic. It is that the antecedent and consequent of classifier is same as a fuzzy rule . In this paper, the FCS is the Michigan style. A single fuzzy if-then rule is coded as an individual. The average gray levels which each group of neighbor pixels has are represented into fuzzy set. Then a pixel is decided whether it is edge pixel or not using fuzzy if-then rules. Depending on the average of gray levels, a number of fuzzy rules can be activated, and each rules makes the output. These outputs are aggregated and defuzzified to take new gray value of the pixel. To evaluate this edge detection, we will compare the new gray level of a pixel with gray level obtained by the other edge detection method such as Sobel edge detection. This comparison provides a reinforcement signal for FCS which is reinforcement learning. Also the FCS employs the Genetic Algorithms to make new rules and modify rules when performance of the system needs to be improved.

Key words : Fuzzy Classifier System, Edge Detection, Bucket Brigade Algorithm

1. Introduction

A Classifier System (CS) is an adaptive system that learns to achieve a task through interacting with environment. CS is also a sort of the reinforcement learning because the learning of it is affected by reinforcing value receiving from environment. A classifier system is a machine learning system that learns syntactically simple string rules to guide its performance in an arbitrary environment[1]. Holland suggested the Bucket Brigade algorithm to learn the effectiveness of classifiers. With Genetic Algorithms , it is possible to make new rules and delete useless rules[2]. But Holland's CS processes the discrete coded information from the environment. When the system codes the information to discontinuous data, it loses excessively the information of the environment. It is called perceptual aliasing[3]. The Fuzzy Classifier System uses the mechanism of fuzzy controllers for mapping continuous inputs to outputs. It is that the antecedent and consequent of classifier is same as a fuzzy rule of the rule base. Valenzuela-Rendon[4] gives the first description of a fuzzy classifier system based on the Michigan approach, with credit assignment to individual rules. He applied it to the single input single output function approximation problem. Unlike the priori study using the fixed membership functions, Parodi and Bonelli[5] describe a fuzzy classifier system using a real-numbered representation which simultaneously learns membership functions and fuzzy relations. And they don't use the message list and the bucket brigade algorithm and simplify the procedure of the credit assignment. Generally the output of the fuzzy system depends on plural rules rather than one rule. Furuhashi et al.[6] employ multiple stimulus-response Michigan-style fuzzy classifier systems for learning to steer a simulated ship into a goal. Multiple classifier systems are used to suppress excessive fuzziness. The system also employs fixed fuzzy set membership function.

This paper proposes the fuzzy classifier system that has the message list and uses the implicit Bucket Brigade Algorithm [1]. The detector make the fuzzified messages that represent the membership function defined by real input variables and the degree of belonging of the input variable to the fuzzy membership function. An input variable can be fuzzified into plural messages according to the degree of overlapping of its membership functions. Fuzzified messages are stored in the message list. Then, FCS finds the rules matching the stored messages in the fuzzy classifier list. The matched classifiers bid the proportion of the product of their strength and firing strength in order to have the right to participate the rule base. The matched classifiers suggesting larger bid value organize the rule base. And each classifier of the rule base proposes an action with weight that depends on the degree of matching rule. The output comes from the aggregation of all the proposed, weighted outputs. Then, the global output is defuzzyfied, thus becoming a real-value. The bids of classifiers in the rule base are distributed to the classifiers of the rule base organized at the priori control step. After the system processes a control action, the system receives the reinforcement signals from the environment and distributes the reinforcement signal to the rules contributing to the action.

Manuscript received March 1, 2003; revised May 1, 2003.

And the FCS employs the GAs to make new rules and modify rules when performance of the system needs to be improved.

Edge detection is an area that embraces both image processing and computer vision. It is used as a pre-process in many computer vision tasks, such as shape recognition or segmentation. The classic problem that edge detection is plagued with is noise. When the input edge has small perturbations, it cause large perturbations in the output image. Because of that noise pixel the output image would consist of two line segments instead of one[7]. Bezdek had developed a fuzzy edge detector FRED which is based on the fuzzy control paradigm[8].

In this paper, we apply the FCS to finding a set of fuzzy rules which can carry out the edge detection. The average gray levels which each group of neighbor pixels has are represented into fuzzy set. Then a pixel is decided whether it is edge pixel or not using fuzzy if-then rules. Depending on the average of gray levels, a number of fuzzy rules can be activated, and each rules makes the output. These outputs are aggregated and defuzzified to take a new gray value of the pixel. To evaluate this edge detection, we compare the new gray level of a pixel with gray level obtained by the other edge detection method such as Sobel edge detection. This comparison provides a reinforcement signal for FCS which is reinforcement learning.

2. Fuzzy Classifier System

The Fuzzy Classifier System makes the classifier system be able to carry out the mapping from continuous inputs to outputs. The classifier system can evaluate the usefulness of rules represented by classifiers with repeated learning. It is the FCS that applies this ability of the machine learning to the concept of fuzzy controller. In this paper, the FCS is the Michigan style and fuzzifies the input values to create the messages. The system stores those messages in the message list and uses the implicit Bucket Brigade Algorithms. And the FCS uses the fixed membership function. Figure 1 shows the structure of the FCS proposed in this paper.

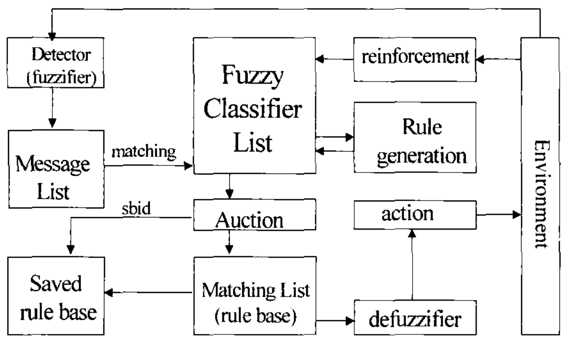


Fig 1. Structure of Fuzzy Classifier System.

2.1 Messages and Message List

The detector carries out fuzzification for the values of input variables, which receive from the environment. The detector is the same as the fuzzifier of the fuzzy logic controller. When the detector fuzzifies the input values, it makes the fuzzified messages stored in the message list. Figure 2 represents the fuzzified message format.

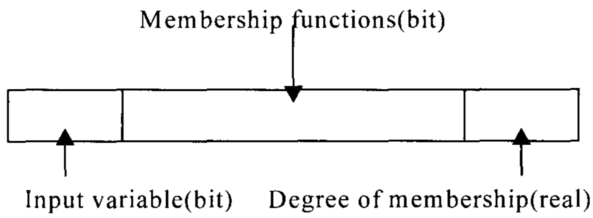


Fig 2. Message Format

A message is composed of three parts. The First represents the input variable by a bit string and the second indicates the membership function to which the input value has the degree of belonging by a bit string. The last part contains the real value representing the degree of belonging of the input variable to membership function. For example, there are input variables X and Y which have separately four membership functions: A, B, C, and D. The input value of X belongs to the membership function A with a degree 0.6 and the membership function B with a degree 0.4 and the input value of Y belongs to the membership function C with a degree 0.8. Then, the detector makes three messages such as 000:0.6, 001:0.4, 110:0.8 and stores them in the message list.

2.1 Classifier List

The classifier in the classifier list has the antecedent representing the membership function of each input variable and the consequent representing the membership function of each output variable. That is the same as a fuzzy rule of the fuzzy controller. Each classifier has the strength to measure its usefulness and modifies its strength according to the results of the action it contributed to. Also the antecedent of the classifier can include the "don't care" symbols in place of the membership function of the variables. This classifier implements the general rule.

2.2 Apportionment of Credit Algorithm

When the detector converts the value of the input variables to messages and stores them in the message list, the system finds the classifier that is satisfied with the messages. To get the right of participating in the rule base, the satisfied classifiers bid in proportion to the product of its strength and firing strength similar to the classifier system. The bid value can be expressed in terms of bid coefficient C_{bid}, strength S_r, and firing strength F_r for a rule r :

$$Bid_r = C_{bid} \cdot S_r \cdot F_r \qquad (1)$$

The firing strength of a rule is the minimum value among the values of which the messages satisfying the rule have the value of the degree of belonging to the membership function. With these bid values, the system selects N rules among the matched rules with a probability proportional to their bid values and organizes the rule base whose cardinality is N. In the standard bucket brigade a classifier activated at time $t+1$ is activated by messages sent by a precise set of classifiers at time t. Therefore the matched classifiers make their payments to the previously active classifiers that sent the messages which matched the currently active classifiers. But, in the FCS, there is no direct connection between the classifiers of a rule base organized at t step and ones of a rule base organized at $t+1$ step. Figure 3 shows that the currently organized rule base makes payment to the previously organized rule base even though threre is no direct link through a message list. Thus the rule base at $t+1$ step is implicitly activated by the previously organized rule base. In this way there is linkage between time-adjacent classifiers, an assumption warranted by the temporal order by the environment. This is the Implicit Bucket Brigade Algorithm[9].

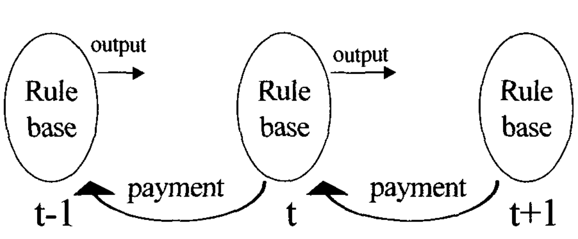


Fig 3. Implicit Bucket Brigade Algorithm

The Implicit Bucket Brigade Algorithm for the FCS can be formulated as follows. First, $S_{bis}(t)$ is the summation of all bid values from the classifiers participating in the rule base at t step. That is

$$S_{bid}(t) = \sum_{k \in M(t)} Bid_k \tag{2}$$

where $M(t)$ is a set of indices of classifiers participating in the rule base at t step and k is classifier index. Then the system distributes the bid summation to the classifiers that participated in the previous rule base. During this bid process, a rule r of the rule base at t step changes its strength at $t+1$ step. At $t+1$ step, the rule r reduces its strength for the bid value obtained by the equation (1). And the rule r receives the $S_{bid}(t+1)$ obtained by the rule base at $t+1$ step proportional to t step firing strength F_r of rule r as follows

$$S_r(t+1) = S_r(t) - Bid_r + \frac{F_r}{\sum_{i \in M(t)}} S_{bid}(t+1) \tag{3}$$

2.4 Reinforcement Learning.

Each rule of the rule base organized at t step proposes the weighted output with its firing strength. And the rule base composes the strength values of the proposed outputs with an aggregation operator and then defuzzyfies the composed strength value. This output of the FCS is also a real value. According to the output of the FCS, the system receives the reinforcement value $R(t)$ from the environment and distributes this value to the classifiers of the currently organized rule base, proportional to the firing strength as follows.

$$S_r(t+1) = S_r(t) + \frac{F_r}{\sum_{i \in M(t)} F_i} R(t) \tag{4}$$

In the CS, the system taxes the classifier in order to find the unused classifier. FCS taxes the classifiers, too. The system taxes all classifiers of the classifier list when the system receives the reinforcement. That is

$$S_r(t+1) = S_r(t) - C_{tax}\, \dot{S}_r(t) \tag{5}$$

where C_{tax} is a tax constant which have $C_{tax} \ll C_{bid}$.

2.5 Rule Discovery

When the system performance doesn't improve during some step, the system tries to find the new rules by GAs. The strength of a classifier in the fuzzy classifier list is regarded as the fitness of GAs. Since the FCS is searching, not for the best single rule, but for a well adapted set of rules, we use the modified crowding replacement to choose the classifiers that die to make room for new offspring[1]. In this way crowding replaces low-performance individuals who are similar to the children being inserted into the population. The new classifier has the initial strength. When there isn't a rule in the classifier list that satisfies the input state, a cover detector[10] generates a given number of new rules.

3. Edge Detection with FCS

3.1 Edge Detection

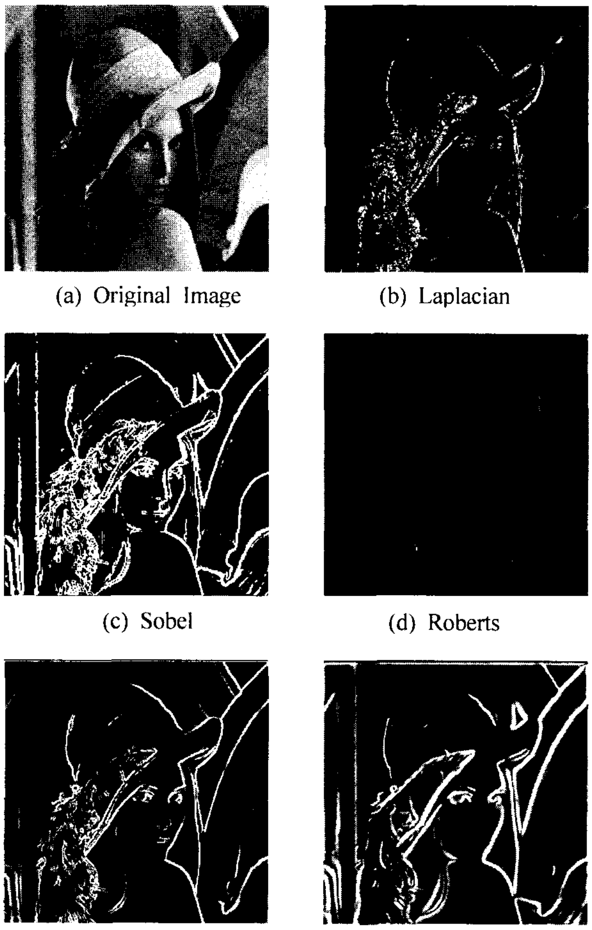
Edge detection is a general name for a class of routines and techniques that operate on the image and result in a line drawing of the image. The lines represent changes in values such as cross sections of planes, intersections of planes, textures, lines, and colors, as well as differences in shading and textures. Some techniques are mathematically oriented, some are heuristic, and some are descriptive. All generally operate on the differences between the gray levels of pixels or groups through masks or thresholds. The final result is a line drawing or similar representation that requires much less memory to be stored costs. Edge dection is also necessary, and subsequent processes, such as segmentation and object recognition. Without edge may be impossible to find overlapping parts, to calculate features such as a diameter and an area, or to determine parts by region growing.

Different techniques of edge detection yield slightly different results. Thus they should be chosen carefully and used wisely.

In the most techniques, the horizontal and vertical gradients between neighboring pixels are calculated and squared, and the square root of the sum is found. Mathematically,

$$\nabla f = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \tag{6}$$

Equation (6) is equivalent to calculating the absolute value of the differences between pixel intensities. The five masks, that is, Laplacian, Sobel, Roberts, Prewitt, and canny edge detectors, effectively do the same gradient differentiation with somewhat different results and very common. When they are applied to an image, the two pairs of masks calculate the gradients in the x and y directions, are added, and then are compared with a threshold. Figure 4 is an original image (a) with its edges detected by Laplacian (b), Sobel (c), Roberts (d), Prewitt (e), Canny (f) edge detectors. the result for other images may be different because the histogram of the image and the chosen thresholds have great effects on the final outcome. Some routines allow the user to change the thresholding values, and some do not. In each case, the user must decide which routine performs the best.



(a) Original Image (b) Laplacian

(c) Sobel (d) Roberts

(e) Prewitt (f) Canny

Fig. 4. (a) An original image (b) and its edge from Laplacian, (c) Sobep, (d) Roberts, (e) Prewitt, and (f) Canny edge dectors.

3.2 Edge Detection with FCS

In order to carry out the edge detection with FCS, we use 3 by 3 window like other edge detection methods. In Figure 4, nine pixels have a gray value separately. The center point p5 is determined whether edge is or not through the eight neighborhood pixels. The four average gray values of horizontal and vertical neighborhood pixels are used as inputs for FCS as follows.

$$X_1 = \frac{(p1 + p2 + p3)}{3}, \quad X_2 = \frac{(p7 + p8 + p9)}{3} \qquad (6)$$

$$X_3 = \frac{(p1 + p4 + p7)}{3}, \quad X_4 = \frac{(p3 + p6 + p9)}{3} \qquad (7)$$

The average gray values of horizontal neighborhood pixels are represented as equation (6) and vertical ones as equation (7). It is similar to the average filter which has the noise-filtering properties.

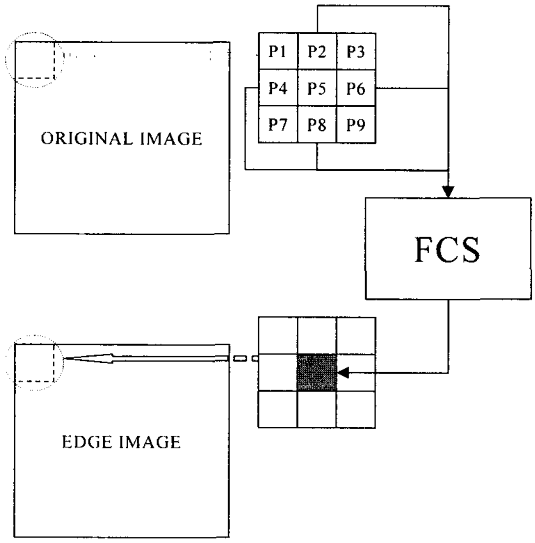
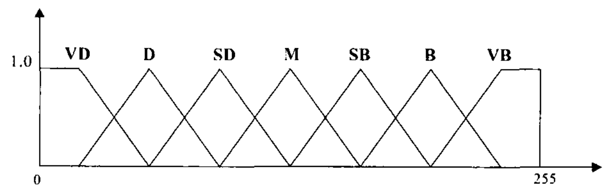
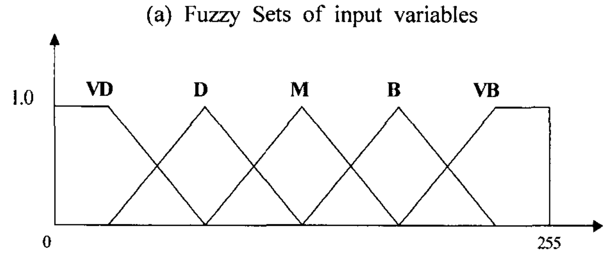


Fig 4. Edge detection with FCS

Fuzzy sets and membership functions are defined for each input variables X_1-X_4 in Fig 5(a). The output variable is defined in Fig 5(b).



(a) Fuzzy Sets of input variables



(b) Fuzzy Sets of output variable

VD : Very Dark, D : Dark
SD : Slightly Dark, M : Middle
SB : Slightly Bright, B : Bright
VB : Very Bright

Fig 5. Fuzzy Sets and Membership Functions

For $n \times m$ image, FCS makes output G_{fcs} with sliding window columnwise. Then FCS thresholds each G_{fos} which is a gray value. That is,

$$P_{ij} = \begin{cases} 0 & \text{if } G_{fod} \leq Th \\ 255 & \text{if } G_{fod} > Th \end{cases}$$
$$i = 1,\ 2,\ \cdots\ n-1 \quad j = 1,\ 2,\ \cdots\ m-1 \tag{8}$$

where Th is a threshold.

After that , FCS compares the P_{ij} to the pixel of Sobel edge image and generate a reinforcement signal which is a teaching signal in this case.

$$R_{ij} = \begin{cases} R & \text{if } P_{ij_{sobel}} = P_{ij} \\ -R & \text{otherwise} \end{cases} \tag{9}$$

where R is reward value of reinforcement signal.

FCS learns each pixel of the entire image with reinforcement and bucket brigade algorithm and applies the rule discovery method as mentioned in section 2.

3.3 Simulation Result

We applied the proposed method to the test image in Figure 4(a). In this simulation, we use seven fuzzy sets for input variable and five fuzzy sets for output variable. The reference image for learning is the Sobel edge image of the test image shown in Figure 4(c). After edge detection through the entire size 256 × 256 image, rule discovery method is executed according to the strength of classifiers which is changed by the reinforcement from the reference image. Then, FCS continues learning edge. After 190 rule discovery, we got a edge detected image with FCS in Figure 7.

Fig. 7. Edge detected image with FCS

In Figure 7, FCS detects new edge which isn't detected on the mirror in Figure 4(c). This shows that FCS can find a set of rules which detects edge through learning.

4. Conclusions

In this paper, we propose a Fuzzy Classifier System to find a set of fuzzy rules which can carry out the edge detection. The fuzzy classifier system can evaluate the usefulness of rules represented by classifiers with repeated learning. FCS fuzzifies the information about neighborhood pixels into messages. Then, FCS executes the repeated learning with reference image and runs the rule discovery to learn the edge detection. FCS may be the one of the most interesting and promising approaches to find useful rules for the environment.

References

[1] David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[2] Holland J. H., "Properties of the bucket brigade algorithm," *Proceedings of the First International Conference on Genetic Algorithms*, pp 1-7. 1985.

[3] Whitehead, S. D. & D. H. Ballard, "Learning to perceive and act by trial and error," *Machine Learning*. 7, 45-83

[4] Valenzuela-Rendon M, "The Fuzzy classifier system: Motivations and first results," *Parallel Problem Solving from Nature - PPSNⅡ*, Springer-Verlag, pp. 330-334, 1991.

[5] Parodi A., Bonelli P., "A new approach to fuzzy classifier systems," *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp223-230, 1993

[6] Furuhashi T., Nakaoka K., Morikawa K., Uchikawa Y., "Controlling excessive fuzziness in a fuzzy classifier system," *Proceeding of the Fifth International Conference on Genetic Algorithms*, pp635, 1993

[7] Alan W., Fabio P., *The Computer Image*, Addison-Wesley, 1998

[8] Bezdek J. C., M. Shirvaikar, "Edge detection using the fuzzy control paradigm," *in Proc. 2nd Eur. Congress Intell. Tech. Soft Computing*, Aachen, Germany, 1994, vol. 1, pp. 1-12.

[9] Dorigo, M. and Bersini, H., "A Comparison of Q-Learning and Classifier Systems," *Proc. of From Animals to Animats, Third International Conference on Simulation of Adaptive Behavior*, 1994

[10] Wilson, S. W., "Knowledge growth in an artificial animal," *Proceedings of the first International Conference on Genetic Algorithms and their Applications*, 1985

Kwee-Bo Sim

Kwee-Bo Sim was born September 20. 1956. He received the B.S. and M.S. degrees in Department of Electronic Engineering from Chung-Ang University, Seoul Korea, in 1984 and 1986 respectively, and Ph. D. degree in Department of Electronic Engineering from the University of Tokyo, japan, in 1990. Since 1991, he has been a faculty member of the School of Electrical and Electronic Engineering at the Chung-Ang University, where he is currently a Professor. His research interests are Artificial Life, Neuro-Fuzzy and Soft Computing, Evolutionary Computation,

Learning and Adaptation Algorithm, Autonomous Decentralized System, Intelligent Control and Robot System, Artificial Immune System, Evolvable Hardware, and Artificail Brain etc. He is a member of IEEE, SICE, RSJ, KITE, KIEE, ICASE, and KFIS.

Phone : +82-2-820-5319
Fax : +82-2-817-0553
E-mail : kbsim@cau.ac.kr