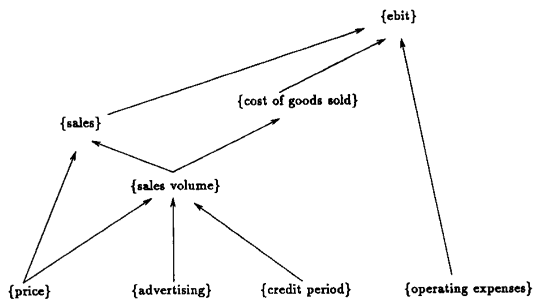


Fig. 3. Causal dependency graph for example 1.

REFERENCES

[1] E. I. Altman, R. G. Haldeman and P. Narayanan, "ZETA analysis, A new model to identify bankruptcy risk of corporations," *J. Banking and Finance,* vol. 1, 1977.
[2] C. Apté and S. J. Hong, "Using qualitative reasoning to understand financial arithmetic," in *Proc. AAAI-86.*
[3] N. H. Ata Mohammed, J. F. Courtney, and D. B. Paradice, "A prototype DSS for structuring and diagnosing managerial problems," *IEEE Trans. Syst., Man, Cybern.,* vol. 18, 1988.
[4] R. J. Berndsen and H. A. M. Daniels, "Qualitative economics: An implementation in PROLOG," *Computer Science in Economics and Management,* vol. 4, 1991.
[5] D. Bobrow *et al.,* Special Issue on Qualitative Reasoning, *Artificial Intell.,* vol. 24, 1984.
[6] J. W. Bonge, "Problem recognition and diagnosis: Basic inputs to business policy," *J. Bus. Policy,* 1972.
[7] P. Bourgine and O. Raiman, "Economics as reasoning on a qualitative model," in *IFAC Economics and Artificial Intelligence,* J-L. Roos, Ed, Aix-en-Provence, France, 1986.
[8] M. J. Bouwman, "Human diagnostic reasoning by computer: An illustration from financial analysis," *Management Sci.,* vol. 29, 1983.
[9] J. F. Courtney, D. B. Paradice, and N. H. Ata Mohammed, "A knowledge-based DSS for managerial problem diagnosis," *Decision Sci.,* vol. 18, 1987.
[10] H. A. M. Daniels and A. J. Feelders, "Model-based diagnosis of business performance," in *Proc. Tenth Int. Workshop on Expert Syst. Their Appl.,* (General conference on second generation expert systems), Avignon, France, 1990.
[11] W. Emory and P. Niland, *Making Management Decisions.* Boston: Houghton Mifflin, 1968.
[12] P. E. Hart, A. Barzilay and R. O. Duda, "Qualitative reasoning for financial assessments: A prospectus," *The AI Mag.,* vol. 7, 1986.
[13] Y. Iwasaki and H. A. Simon, "Causality in device behavior," *Artificial Intell.,* vol. 29, 1986.
[14] P. Judd, C. Paddock and J. Wetherbe, "Decision impelling differences: An investigation of management by exception reporting," *Inform. Management,* vol. 4, 1981.
[15] D. W. Kosy, "Applications of explanation in financial modeling," in *Artificial Intelligence, Simulation and Modeling,* L. E. Widman *et al.,* Eds. New York: Wiley, 1989.
[16] D. W. Kosy and B. P. Wise, "Self-explanatory financial planning models," *Proc. Nat. Conf. Artificial Intell., AAAI-84.*
[17] H. Mintzberg, D. Raisinghani, and A. Théorêt, "The structure of 'unstructured' decision processes," *Administrative Sci. Quart.,* vol. 21, 1976.
[18] R. S. Patil, P. Szolovits and W. B. Schwartz, "Causal understanding of patient illness in medical diagnosis," *Proc. Seventh IJCAI,* 1981.
[19] W. F. Pounds, "The process of problem finding," *Industrial Management Review,* vol. 11, 1969.
[20] N. Rauh, *"Wissensbasierte Systeme zur Unternehmensdiagnose auf der Grundlage von Jahresabschlussdaten und Branchenvergleichswerten in der Steuerkanzlei,"* dissertation, Friedrich-Alexander Universität Erlangen-Nürnberg, 1988.
[21] R. Reiter, "A theory of diagnosis from first principles," *Artificial Intell.,* vol. 32, 1987.
[22] H. A. Simon, *The New Science of Management Decision, rev. ed.* Englewood Cliffs, NJ: Prentice-Hall, 1977.
[23] H. A. Simon, "Causal ordering and Identifiability," in *Studies in Econometric Method,* W. C. Hood and T. C. Koopmans, Eds. New York: Wiley, 1953.
[24] J. F. Weston and T. E. Copeland, *Managerial Finance, 8th ed.* Chicago: The Dryden Press 1986.

Correction to "Fuzzy-Attribute Graph with Application to Chinese Character Recognition"[1]

K. P. Chan and Y. S. Cheung

Abstract—**In everyday life, many properties or concepts we encounter are fuzzy in nature. To include those fuzzy properties in solving some types of problems, we have extended the attributed graph to fuzzy-attribute graph (FAG). With such extension, equality of attributes can no longer be used when matching of FAG's is considered, as equality of two fuzzy sets is too stringent a condition. In the paper, fuzzy-attribute graph is formally defined and a new measure for matching two FAG's is suggested. The new measure has its interpretation under fuzzy logic. The model is applied to the recognition of handprinted Chinese characters and the result is presented.**

I. Introduction

Machine recognition of handprinted Chinese characters was considered to be a very hard problem by many researchers. This is caused by 1) the large number of Chinese characters (there are more than 40 000 characters of which more than 4 000 are in common use) and 2) the large variability in handwriting. One common approach, known as the decision-theoretic approach, is based on some kind of numerical feature extraction. The characters are represented as points in a multidimensional feature space and a discriminant function (or the equivalence) is used to perform the classification. Despite its simplicity, many characteristics of Chinese characters cannot be expressed in numerical form. For example, it is very difficult to represent the structural relations between the strokes of a character as numerical features.

Recently, structural approach begins to gain popularity. Each Chinese character is composed of strokes that form naturally the pattern primitives used in structural pattern recognition. Plex grammar [1] and attributed tree grammar [2] have been proposed. However, only preliminary results are available. One problem in using a grammatical approach is the need for the correct ordering of the strokes, which is nontrivial to extract in an off-line recognition

Manuscript received July 29, 1988; revised September 24, 1989 and June 8, 1991.

K. P. Chan was with the Department of Electronic Engineering, City Polytechnic of Hong Kong, Tat Chee Avenue, Yau Yat Chuen, Hong Kong and is now with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam Road, Hong Kong.

Y. S. Cheung is with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam Road, Hong Kong.

IEEE Log Number 9104109.

[1] *Originally published in IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, vol. 22, no. 1, pp. 153-160, January/February 1992. Due to a printer's error, the Chinese characters appearing in this version were omitted in the original printing.*

0018-9472/92$03.00 © 1992 IEEE

system. In addition, the use of higher dimensional grammar is very complex. Another approach is by pattern matching. Relaxation matching using polygonal approximation [3] and strokes extraction [4] have been attempted with considerable success. However, the relaxation technique cannot represent the hierarchical characteristics that exists in Chinese characters. By hierarchical characteristics, we mean the fact that each character is composed of radicals (which may themselves be complete characters) and the radicals in turn are composed of simpler radicals or primitive strokes. It is believed that such hierarchical characteristics are used by human in the recognition process. Due to this hierarchical structure, we propose to use radical matching rather than character matching and determine the character from the radicals found.

Fuzzy set theory was first introduced by L. A. Zadeh [5] in 1965. It is an attempt to use formal mathematical tools to investigate problems pertaining to uncertainty, ambiguity and vagueness. The concepts modeled by fuzzy set theory have no exact boundary between membership and nonmembership and the change is gradual rather than abrupt. Such concepts map well to the characteristics of handwritten Chinese characters. The four primitive strokes in Chinese characters, 一, 丨, 丿, and ㇏, have no exact boundary and the changes between different stroke types are gradual. Hence fuzzy set theory provides a possible vehicle in describing the uncertainty in stroke types. Similarly, many other properties of strokes and their relations are fuzzy in nature. Moreover, fuzzy set can be considered as a generalization. Ordinary set can be represented as a special case of fuzzy set, where $\tilde{A}$ is the fuzzy set that represents the ordinary set A, and $\mu_{\tilde{A}}(x)$ denotes the membership of x in the fuzzy set $\tilde{A}$:

$$\mu_{\tilde{A}}(x) = \begin{cases} 1 & \forall x \in A \\ 0 & \text{otherwise.} \end{cases}$$

We have chosen a graph structure to represent Chinese characters. The attributed graph proposed by Tsai and Fu [6] gives a straightforward representation of structural patterns and we have modified it to include fuzzy attributes. We call the resultant graph a fuzzy-attribute graph (FAG). In the next section, we will define FAG formally and discuss its properties.

II. Attributed Graph and FAG

Attributed graph was introduced by Tsai and Fu [6] for pattern analysis. It gives a more straightforward representation of structural patterns. The vertices of the graph represent pattern primitives describing the pattern while the arcs are the relations between these primitives. The following definition of attributed graph follows [7].

Each vertex may take attributes from the set $Z = \{z_i | i = 1, \cdots, I\}$. For each attributes z_i, it will have possible values taken from the set $S_i = \{s_{ij} | j = 1, \cdots, J_i\}$. $L_v = \{(z_i, s_{ij}) | j = 1, \cdots, J_i; i = 1, \cdots, I\}$ denotes the set of possible attribute-value pairs of the vertices. A valid pattern primitive is just a subset of L_v in which each attribute appears only once, and Π denotes the set of all those valid pattern primitives. Thus each vertex will be represented by an element of Π.

Similarly, for the arcs, we have the attribute set $F = \{f_i | i = 1, \cdots, I'\}$ in which each attribute f_i may have values taken from the set $T_i = \{t_{ij} | j = 1, \cdots, J_i'\}$. $L_a = \{(f_i, t_{ij}) | j = 1, \cdots, J_i'; i = 1, \cdots, I'\}$ denotes the set of possible relational attribute-value pairs. A valid relation is just a subset of L_a in which each attribute appears once. The set of all those valid relation is Θ.

The attributed graph can be defined as follows. Let N be a finite nonempty set of vertices and $E \subseteq N \times N$ a set of distinct ordered pairs of distinct elements in N.

Definition 1: An attributed graph G over $L = (L_v, L_a)$ with an underlying graph structure $H = (N, E)$ is defined to be the ordered pair (V, A) where $V = (N, \sigma)$ is called an ***attributed vertex set*** and $A = (E, \delta)$ is called an ***attributed arc set.***

$$\sigma : N \rightarrow \Pi \text{ is called vertex interpreter}$$
$$\delta : E \rightarrow \Theta \text{ is called arc interperter}$$

The vertex and arc interpreter is just a mapping that maps the vertices or arcs to their corresponding attribute sets.

When using attributed graph to represent Chinese characters, a natural way is to represent each stroke as a vertex and the relation between strokes as arcs. For illustration, we have attributes STROKE_TYPE and LENGTH for each stroke and for relation, we take JOINT_TYPE, VERT_REL and HORI_REL for example:

$$\begin{aligned} \text{STROKE\_TYPE} &= \{\text{Vertical, Horizontal, Slant 45, Slant 135}\} \\ \text{LENGTH} &= \{\text{Long, Short}\} \\ \text{JOINT\_TYPE} &= \{T\text{-from}, T\text{-into}, Ht, \text{Cross, Parallel}\} \\ \text{VERT\_REL} &= \{\text{On-top-of, Below-of, No-vert-relate}\} \\ \text{HORI\_REL} &= \{\text{Left-of, Right-of, No-hori-relate.}\} \end{aligned}$$

With the definition of attributed graph given previously:

$$\begin{aligned} Z &= \{z_1 = STROKE\_TYPE, z_2 = LENGTH\} \\ S_1 &= \{s_{11} = Vertical, s_{12} = Horizontal, s_{13} = Slant45, \\ &\quad s_{14} = Slant135\} \\ S_2 &= \{s_{21} = Long, s_{22} = Short\} \\ L_v &= \{(STROKE\_TYPE, Vertical), \\ &\quad (STROKE\_TYPE, Horizontal), \\ &\quad (STROKE\_TYPE, Slant45), \\ &\quad (STROKE\_TYPE, Slant135), \\ &\quad (LENGTH, Long), (LENGTH, Short)\}. \end{aligned}$$

Similarly:

$$\begin{aligned} F &= \{f_1 = JOINT\_TYPE, f_2 = VERT\_REL, \\ &\quad f_3 = HORI\_REL\} \\ T_1 &= \{t_{11} = T - from, t_{12} = T - into, t_{13} = Ht, \\ &\quad t_{14} = Cross, t_{15} = Parallel\} \\ T_2 &= \{t_{21} = On - top - of, t_{22} = Below - of, \\ &\quad t_{23} = No - vert - relate\} \\ T_3 &= \{t_{31} = Left - of, t_{32} = Right - of, \\ &\quad t_{33} = No - hori - relate\} \\ L_a &= \{(JOINT\_TYPE, T - from), \\ &\quad (JOINT\_TYPE, T - into), (JOINT\_TYPE, Ht), \\ &\quad (JOINT\_TYPE, Cross), \\ &\quad (JOINT\_TYPE, Parallel), \\ &\quad (VERT\_REL, On - top - of), \\ &\quad (VERT\_REL, Below - of), \\ &\quad (VERT\_REL, No - vert - relate), \\ &\quad (HORI\_REL, Left - of), \\ &\quad (HORI\_REL, Right - of), \\ &\quad (HORI\_REL, No - hori - relate)\}. \end{aligned}$$

The following equations are valid pattern primitives and relations:

$$\pi_1 = \{(STROKE\_TYPE, Vertical), (LENGTH, Long)\}, \text{and}$$

$$\theta_1 = \{(JOINT\_TYPE, Cross), (VERT\_REL, On-top-of), (HORI\_REL, No-hori-relate)\}.$$

The matching of vertices and arcs can be determined by equality of attributes.

However, it can be noted that many of the aforementioned attributes are in fact fuzzy in nature, e.g., STROKE_TYPE, and even VERT_REL and HORI_REL are rather fuzzy. It is thus a natural step to extend the definition to include fuzzy attributes.

Since every crisp set can be represented as a particular case of a fuzzy set by setting the membership to 0 and 1 for all elements, 1 if the element belongs to the set and 0 otherwise, we can generalize our definition as follows.

Each vertex may have attributes from the set $Z = \{z_i | i = 1, \cdots, I\}$. For each attribute z_i, it may take values from $S_i = \{s_{ij} | j = 1, \cdots, J_i\}$. The set of all possible attribute-value pair is $\tilde{L}_v = \{(z_i, \tilde{A}_{S_i}) | i = 1, \cdots, I\}$ where $\tilde{A}_{S_i}$ is a fuzzy set on the attribute-value set S_i. A valid pattern primitive is just a subset of $\tilde{L}_v$ in which each attribute appears only once, and $\tilde{\Pi}$ represent the set of all those valid pattern primitives.

Similarly, each arc may have attributes from the set $F = \{f_i | i = 1, \cdots, I'\}$ in which each f_i may take values from $T_i = \{t_{ij} | j = 1, \cdots, J_i'\}$. $\tilde{L}_a = (f_i, \tilde{B}_{T_i}) | i = 1, \cdots, I'$ denotes the set of all possible relational-attribute value pair, where $\tilde{B}_{T_i}$ is a fuzzy set on the relational attribute-value set T_i. A valid relation is just a subset of $\tilde{L}_a$ in which each attribute appears only once. The set of all valid relation is denoted $\tilde{\Theta}$.

Definition 2: An FAG $\tilde{G}$ over $\tilde{L} = (\tilde{L}_v, \tilde{L}_a)$ with an underlying graph structure $H = (N, E)$ is defined to be an ordered pair $(\tilde{V}, \tilde{A})$, where $\tilde{V} = (N, \tilde{\sigma})$ is called a fuzzy vertex set and $\tilde{A} = (E, \tilde{\delta})$ is called a fuzzy arc set and

$$\tilde{\sigma} : N \rightarrow \tilde{\Pi} \text{ is called a fuzzy vertex interpreter}$$

$$\tilde{\delta} : E \rightarrow \tilde{\Theta} \text{ is called a fuzzy arc interpreter.}$$

The definition also applies when there are nonfuzzy attributes.

With the aforementioned definition, we may have strokes represented as

$$\tilde{\pi}_1 = \{(STROKE_TYPE, \{0.7/Vertical, 0.85/Slant45, 0.01/Horizontal, 0/Slant135\}), (LENGTH, \{0.6/Long, 0/Short\})\}$$

$$\tilde{\theta}_1 = \{(JOINT_TYPE, \{0.7/T\text{-}from, 0.65/Cross, 0/T\text{-}into, 0/Ht, 0/Parallel\}), (VERT_REL, \{0.9/On\text{-}top\text{-}of, 0/Below\text{-}of, 0.25/No\text{-}vert\text{-}relate\}), (HORI_REL, \{0.2/Left\text{-}of, 0.4/Right\text{-}of, 0.77/No\text{-}hori\text{-}relate\})\}.$$

When we use the aforementioned extension, problems will arise when considering matching between two different vertices or arcs. For example, consider another vertex,

$$\tilde{\pi}_2 = \{(STROKE_TYPE, \{0.7/Vertical, 0.5/Slant45, 0/Horizontal, 0/Slant135\}), (LENGTH, \{0.4/Long, 0/Short\})\}.$$

When we match $\tilde{\pi}_1$ against $\tilde{\pi}_2$, obviously, we can no longer use equality as criterion because equality of two fuzzy set is too stringent a condition. We need other criteria. In the next section, we will define three measures feasibility, compatibility and λ-monomorphic to handle this problem.

III. Monomorphism Between FAGs

Consider two FAG's $\tilde{G}_1$ and $\tilde{G}_2$. Monomorphism problem is to find a one to one mapping from $\tilde{G}_1$ to $\tilde{G}_2$ that preserves incidence relations. As described in the previous section, we cannot use equality as a criterion. The proposed criteria for matching is defined as follows.

Definition 3: Feasibility α is a measure of similarity between two primitives v_1 and v_2 of $\tilde{G}_1$ and $\tilde{G}_2$ respectively, and

$$0 \leq \alpha(v_1, v_2) \leq 1$$

With the definition of fuzzy vertex set in Section II, let $\tilde{A}_{1S_i}$ be the fuzzy set that gives the attribute value for z_i of v_1 and $\tilde{A}_{2S_i}$ be that of z_i of v_2. The value of feasibility can be defined as

$$\alpha(v_1, v_2) = \bigwedge_{i=1}^{I} \bigvee_{j=1}^{J_I} \{(\mu_{\tilde{A}_{1S_i}}(s_{ij}) \wedge \mu_{\tilde{A}_{2S_i}}(s_{ij}))\}$$

where $\mu_{\tilde{A}_{KS_i}}(s_{ij})$ is the membership value of s_{ij} in the fuzzy set $\tilde{A}_{KS_i}$, K = 1,2, and, $\wedge$ and $\vee$ are the min and max operators respectively. The formula comes from the following analysis.

The membership of a fuzzy set can be interpreted as the compatibility of a member and its properties, or in other words, the truth value of a member possessing that properties, for example:

$$\tilde{A}_{1S_i} = \left\{\mu_{\tilde{A}_{1S_i}}(s_{ij})/s_{ij} | s_{ij} \in S_i\right\}$$

and

$$\tilde{A}_{2S_i} = \left\{\mu_{\tilde{A}_{2S_i}}(s_{ij})/s_{ij} | s_{ij} \in S_i\right\}$$

can be interpreted as

v_1 is s_{ij} with truth value $\mu_{\tilde{A}_{1S_i}}(s_{ij})$

and

v_2 is s_{ij} with truth value $\mu_{\tilde{A}_{2S_i}}(s_{ij})$.

Taking disjunction over all the possible attribute values, for a particular attribute z_i, the feasibility with respect to this attribute can be defined by the fuzzy logical expression,

$$(v_1 \text{is } s_{i1} \text{and } v_2 \text{is } s_{i1}) \; or \; (v_1 \text{is } s_{i2} \text{and } v_2 is s_{i2}) \; or \cdots \text{or } (v_1 \text{is } s_{iJ_i} \text{ and } v_2 \text{is } s_{iJ_i})$$

and the truth value of this fuzzy logical expression is,

$$(\mu_{\tilde{A}_{1S_i}}(s_{i1}) \wedge \mu_{\tilde{A}_{2S_i}}(s_{i1})) \vee (\mu_{\tilde{A}_{1S_i}}(s_{i2}) \wedge \mu_{\tilde{A}_{2S_i}}(s_{i2})) \vee \cdots \vee (\mu_{\tilde{A}_{1S_i}}(s_{iJ_i}) \wedge \mu_{\tilde{A}_{2S_i}}(s_{iJ_i})).$$

Taking conjunction over all attributes, we define feasibility of two vertices v_1 from $\tilde{G}_1$ and v_2 from $\tilde{G}_2$ as

$$((v_1 \text{ is } s_{11} \text{ and } v_2 \text{ is } s_{11}) \text{ or } \cdots \text{ or } (v_1 \text{ is } s_{1J_1} \text{ and } v_2 \text{ is } s_{1J_1})) \text{ and}$$

$$((v_1 \text{ is } s_{21} \text{ and } v_2 \text{ is } s_{21}) \text{ or } \cdots \text{ or } (v_1 \text{ is } s_{2J_2} \text{ and } v_2 \text{ is } s_{2J_2})) \text{ and}$$

$$\vdots$$

$$((v_1 \text{ is } s_{I1} \text{ and } v_2 \text{ is } s_{I1}) \text{ or } \cdots \text{ or } (v_1 \text{ is } s_{IJ_I} \text{ and } v_2 \text{is } s_{IJ_I})).$$

Definition 4: Compatibility β is a measure of similarity between two arcs, e_1 and e_2 of $\tilde{G}_1$ and $\tilde{G}_2$ respectively, and

$$0 \leq \beta(e_1, e_2) \leq 1.$$

Similarly, with the definition of fuzzy arc set in Section II, let $\tilde{B}_{1T_i}$ be the attribute value of the attribute f_i of e_1 and $\tilde{B}_{2T_i}$ be the attribute value of f_i of e_2. The value of compatibility can be obtained by

$$\beta(e_1, e_2) = \bigwedge_{i=1}^{I'} \bigvee_{j=1}^{J_i'} \{(\mu_{\tilde{B}_{1T_i}}(t_{ij}) \wedge \mu_{\tilde{B}_{2T_i}}(t_{ij}))\}$$

where $\mu_{\tilde{B}_{KT_i}}(t_{ij})$ is the membership of t_{ij} in the fuzzy set $\tilde{B}_{KT_i}$, K=1,2.

Definition 5: Let $\tilde{G}_1$ and $\tilde{G}_2$ be two FAG's with the underlying graph $H_1 = (N_1, E_1)$ of $\tilde{G}_1$ is monomorphic to the underlying graph H_2 of $\tilde{G}_2$. The degree of matching γ is defined as

$$\gamma(\tilde{G}_1, \tilde{G}_2) = \bigwedge_{i \in N_1} \alpha(i, h(i)) \bigwedge_{(i,j) \in E_1} \beta(e_1(i,j), e_2(h(i), h(j)))$$

where $h(i)$ is the vertex in $\tilde{G}_2$ that matches vertex i in $\tilde{G}_1$, and $e_k(i,j)$ is the arc joining vertices i and j in $\tilde{G}_k$.

Definition 6: Let $\tilde{G}_1$ and $\tilde{G}_2$ be two FAG's. $\tilde{G}_1$ is λ-monomorphic to $\tilde{G}_2$ if the underlying graph H_1 of $\tilde{G}_1$ is monomorphic to the underlying graph H_2 of $\tilde{G}_2$, and the degree of matching $\gamma \geq \lambda$.

The $\tilde{G}_1$ is λ-monomorphic to $\tilde{G}_2$ is equivalent to say that the underlying graphs are monomorphic and also for every matched pairs from $N_1 \times N_2$, $v_1(i)$ is feasible with $v_2(h(i))$ with truth value of at least λ, and for every matched pair from $E_1 \times E_2$, $e_1(i,j)$ is compatible with $e_2(h(i), h(j))$ with truth value of at least λ.

The matching between two FAG's, as defined previously, is a kind of inexact matching. Most inexact matching in pattern recognition is based on probability [8], [9]. However, as discussed earlier, the inexactness of Chinese character is fuzzy and can be modeled by fuzzy set theory. There have been works on applying fuzzy set theory to inexact matching—fuzzy matching [10]–[12]. This is another attempt in this direction, using fuzzy logic for inexact matching of fuzzy data.

IV. λ-Cuts of FAGs

In this section, we will discuss some properties of FAG's and explain its advantages over attributed graph in pattern recognition.

Definition 7: Let G be an attributed graph, with attribute set Z and F, and attribute values S_i and T_i. Then an extended attributed graph G' is an attributed graph with the same attribute set Z, but for each attribute, it may take values from

$$S_i' = S_i \cup \{\perp\}$$
$$T_i' = T_i \cup \{\perp\}$$

where $\perp$ has the meaning of *empty*.

Definition 8: Let $\tilde{G}$ be a FAG and G be an extended attributed graph with the same sets of attributes Z and F, the same set of attribute values S_i and T_i, $\forall$ i, and the same underlying graph $H = (N, E)$. Let $\theta_i = \sigma(n_i)$ and $\tilde{\theta}_i = \tilde{\sigma}(n_i)$, $\forall$ $n_i \in N$, and $\pi_i = \delta(e_i)$ and $\tilde{\pi}_i = \tilde{\delta}(e_i)$, $\forall e_i \in E$. G is said to be a λ-cut of $\tilde{G}$ if $\forall n_i \in N, \forall z_i \in Z$, if $(z_i, s_{ij}) \in \theta_i$, and $(z_i, \tilde{A}_{S_i}) \in \tilde{\theta}_i$, then $s_{ij} \in$ λ-cut of $\tilde{A}_{S_i}$, if the λ-cut is nonempty, otherwise, $s_{ij} = \perp$, and $\forall e_i \in E, \forall f_i \in F$, if $(f_i, t_{ij}) \in \pi_i$ and $(f_i, \tilde{B}_{T_i}) \in \tilde{\pi}_i$, then $t_{ij} \in$ λ-cut of $\tilde{B}_{T_i}$, if the λ-cut is nonempty, otherwise, $t_{ij} = \perp$.

We can note that the λ-cut of a FAG is in fact a nonfuzzy version of the FAG and is not unique. That is, each FAG represents a number of possible (nonfuzzy) attributed graphs, depending on the value λ. The introduction of extended attributed graph is to ensure that the λ-cut always exists, even when the λ-cuts of some fuzzy sets are empty.

The following two properties describe the relation between the fuzzy version and the nonfuzzy ones.

Property 1: If $\tilde{G}_1$ is λ-monomorphic to $\tilde{G}_2$, then one of the λ-cuts of $\tilde{G}_1$ is monomorphic (in the usual sense) to one of the λ-cuts of $\tilde{G}_2$.

Property 2: If $\tilde{G}_1$ is λ_1-monomorphic to $\tilde{G}_2$, then $\tilde{G}_1$ is λ_2-monomorphic to $\tilde{G}_2$ $\forall \lambda_2 < \lambda_1$.

The proof of Properties 1 and 2 can be found in [13].

In most cases, we must finally make a (nonfuzzy) decision. Making such decision will, in the authors' opinion, bind some fuzzy variables to some nonfuzzy values. For example, we may say that the radical 亻 is a 丿 stroke *on-top-of* a 丨 stroke. Here 丿, 丨 and *on-top-of* are fuzzy. Given a stroke, we cannot decide exactly what the stroke type is. (As discussed earlier, fuzzy concepts has no exact boundary on membership and nonmembership). However, by making decision that the radical is a 亻, we have bind the fuzzy variables into nonfuzzy values, i.e., the stroke type is 丿 and 丨 and the relation type is *on-top-of*. This binding process comes from the decision process. The advantage of using the current approach is to allow the fuzziness to be retained during the decision process until final decision is made. In fact, complex systems, where we make decision based on continuous, exact variables such as temperature and length, will be very complicated or even unsolvable. By introducing fuzzy concepts based on these variables, such as hot, cold, long, short (a fuzzy concept may be based on more than one variable), the system is much simplified.

V. Radical Combination

When we apply FAG to Chinese character recognition, the radical templates we use contain no fuzziness. This is because the radical is an ideal character or a definition. Hence no fuzziness should exist. For example, when we define the radical 土, we have the following definitions,

> a horizontal stroke crosses with a vertical stroke that forms a T-joint with another horizontal stroke. The latter is below the first horizontal stroke and with a longer length.

There should be no fuzziness involved in the definition. However, any instance of the character will contain fuzziness. Hence the pattern graph will be represented as a FAG while the radical template as a hard FAG (HFAG), which is a special case of a FAG with membership either 0 or 1 (which is slightly different from the attributed graph defined). The matching is, therefore, between a fuzzy pattern and a nonfuzzy description. Nevertheless, the discussion presented so far still applies, since a HFAG is still a FAG.

The set of HFAG with the same underlying graph forms a complete lattice with the order we define. We will try to represent two templates with the same underlying graph (two different definitions of the same radical with minor variations) by the least upper bound of the two templates, and use the new HFAG for matching.

Definition 9: Given two extended attributed graphs, G_1 and G_2 with the same underlying graph, G_1 is said to be contained in G_2 if $\forall n \in N$, and $\forall z \in Z$, either

$$(z, \perp) \in \sigma_1(n) \text{ or}$$
$$\text{if } (z,s) \in \sigma_1(n) \text{ then } (z,s) \in \sigma_2(n)$$

where σ_1 and σ_2 are the vertex interpreter of G_1 and G_2 respectively, and $\forall e \in E$, and $\forall f \in F$, either

$$(f, \perp) \in \delta_1(e) \text{ or}$$
$$\text{if } (f,t) \in \delta_1(e) \text{ then } (f,t) \in \delta_2(e)$$

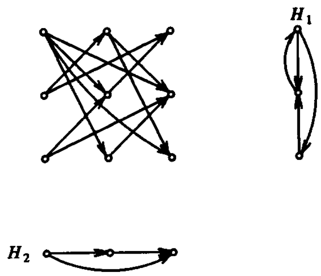


Fig. 1. The net formed from graphs H_1 and H_2.

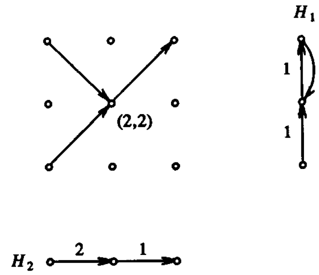


Fig. 2. The weighted edge subgraph of node (2,2).

where δ_1 and δ_2 are the arc interpreter of G_1 and G_2 respectively.

Definition 10: Let $\tilde{G}_1$ and $\tilde{G}_2$ be two HFAG's having the same underlying graph. We define $\tilde{G}_1 \geq \tilde{G}_2$ *iff* every 1-cut of $\tilde{G}_2$ is contained in a 1-cut of $\tilde{G}_1$.

Theorem 1: Let $\mathfrak{G}$ be the set of all HFAG's defined over a given underlying graph G. Then $(B, \geq)$ is a complete lattice.

The proof of theorem 1 can be found in [13].

Every 1-cut of a $HFAG$ is a λ-cut of it $\forall\lambda \in [0,1]$. Property 1 of the previous section ensures that if the pattern graph $\tilde{G}$ is λ-monomorphic to the radical graph $\tilde{H}$, then there exists a λ-cut of $\tilde{G}$ which is monomorphic to a λ-cut of $\tilde{H}$, one of the attributed graph defined by the radical definition. For example, 土 and 扌 can be combined to one HFAG as well as 口 and [illegible] instead of introducing new radicals. During the graph matching procedure, only one template will be required, as compared to using two different templates.

VI. Finding Monomorphism

After the discussion of some properties of FAG's, we proceed to describe the algorithm for matching of FAG's. Graph monomorphism is known to be NP-complete [14]. The following algorithm is based on a tree-search algorithm proposed by Akinniyi, Wong, and Stacey [15].

Let H_1 be the underlying graph of $\tilde{G}_1$ and H_2 be the underlying graph of $\tilde{G}_2$. We first construct a net from the graphs H_1 and H_2. A net is just a product graph of H_1 and H_2, in which a node (i,j) has an edge incident into (i',j') if there is an arc incident from i to i' in H_1 and j to j' in H_2 (Fig. 1).

Each node (i,j) in the net represents a matching of vertex i in H_1 and vertex j in H_2. Before going further, let us look at the strong necessary condition for graph monomorphism:

> The number of neighbors of node (i,j) in the net that are in distinct rows and columns of the net must equal the number of neighbors of vertex i in H_1.

(a) (b)

Fig. 3. Nullable strokes in radicals.

TABLE I
Attribute Values of the Nodes of Character 去

Stroke Number	Length		Stroke Type			
	Long	Short	一	丨	丿	㇏
1	1.00	0.00	0.00	0.01	0.00	0.89
2	1.00	0.00	0.00	1.00	0.00	0.00
3	1.00	0.00	1.00	0.00	0.97	0.00
4	1.00	0.00	0.37	0.00	0.00	0.00
5	1.00	0.00	1.00	0.00	0.00	0.00
6	1.00	0.00	1.00	0.00	0.00	0.00

The concept of weighted edge subgraph (WES) was introduced to check the strong necessary condition, which is just a projection of the net on H_1 and H_2, with the edge weighted by the number of arcs in the net that project onto that edge. For example, consider the graph in Fig. 1. The WES of node $(2,2)$ is shown in Fig. 2.

If the number of neighbors with nonzero weighted edges connected with vertex i in graph H_1 is *gcount* and that with vertex j in graph H_2 is *hcount* for the WES of node (i,j), then the number of neighbors of node (i,j) in the net in distinct rows and columns is equal to the minimum of *gcount* and *hcount*.

A node (i,j) in the net is λ-feasible if 1) it satisfies the strong necessary condition, and 2) the feasibility $\alpha(i,j) \geq \lambda$. A node in the net is λ-infeasible if it is not λ-feasible. From now on, infeasible in the text will mean λ-infeasible and feasible will mean λ-feasible.

When an infeasible node is found, all edges coming out from that node is deleted. We can apply the strong necessary condition again to find further infeasible nodes. When no more such node is found, we can start searching the net with depth-first search strategy.

During the search, we first select a feasible node in the first row of the net. Then we mark all the nodes in that row and column as infeasible and iterate to find more infeasible nodes until no more can be found. If only one feasible node exists on each row in the net, then a monomorphism is found, and we backtrack one level and continue the search. Otherwise, we apply the same strategy to the next row until all monomorphisms are found.

VII. Application to Chinese Character Recognition

As discussed earlier, every Chinese character is represented as a FAG. Every radical can be represented as a HFAG. By graph monomorphism (subgraph matching), all the radicals existing in the character can be found.

Before going further, we first discuss several points that must be noted.

1) As discussed in Section V, the radical templates are not fuzzy. This is because the radical is in fact an ideal character or a definition. Therefore no fuzziness should exist. Also, two HFAG's representing the same character can be combined together by finding their least upper bound in the lattice, if they have the same underlying graph.

TABLE II
ATTRIBUTE VALUES OF THE EDGES OF CHARACTER 去

Relation Number	Vertical Relation: Top	Below	Norel	Horizontal Relation: Left	Right	Norel	Joint Type: T-f	T-into	H-T	X	//
(1,2)	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
(1,3)	0.38	0.00	0.62	0.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00
(1,4)	0.00	0.73	0.00	0.00	1.00	0.00	0.00	0.45	0.00	0.00	0.00
(1,5)	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00
(1,6)	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00
(2,3)	1.00	0.00	0.00	0.00	0.35	0.65	0.00	1.00	0.00	0.00	0.00
(2,4)	1.00	0.00	0.00	0.00	0.72	0.28	0.00	1.00	1.00	0.00	0.00
(2,5)	1.00	0.00	0.00	0.00	0.12	0.88	0.00	1.00	0.00	0.00	0.00
(2,6)	0.00	0.03	0.97	0.00	0.15	0.85	0.01	0.19	0.00	1.00	0.00
(3,4)	0.00	0.95	0.05	0.00	0.03	0.97	0.58	0.00	1.00	0.00	0.00
(3,5)	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
(4,5)	0.00	0.95	0.05	0.06	0.00	0.94	0.08	1.00	0.00	0.00	0.00
(4,6)	0.00	1.00	0.00	0.20	0.00	0.80	0.00	0.00	0.00	0.00	0.00
(5,6)	0.00	1.00	0.00	0.00	0.00	1.00	0.85	0.00	0.00	0.00	1.00

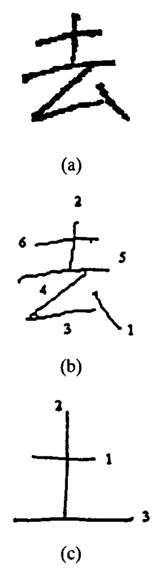


Fig. 4. The character 去 (a) Before thinning. (b) After thinning. (c) The template 土.

TABLE III
ATTRIBUTE VALUES OF THE NODES OF TEMPLATE 土

Stroke No.	Length: Long	Short	Check Flag	Stroke Type: —	|	/	\	Check Flag
1	0.00	0.00	F	1.00	0.00	0.00	0.00	T
2	0.00	0.00	F	0.00	1.00	0.00	0.00	T
3	0.00	0.00	F	1.00	0.00	1.00	0.00	T

2) *Selective attribute matching:* When checking radicals, not all attributes are useful. For example, when checking the radical 土, for the relation between two horizontal strokes, the attribute *HORI_REL* is not relevant at all. Hence for each attribute, we have a flag to indicate whether that particular attribute will be checked for that stroke or relation. In the previous example, the flag for the attribute *HORI_REL* will be set to false.
3) *Handling of null strokes:* In many cases, the existence or nonexistence of some strokes will not affect the radical. For example, in Fig. 3, the strokes labeled n is nullable, where its existence or nonexistence is not important. To handle such nullable strokes, we can add a flag to the radical graph to indicate whether the stroke can be a null stroke. In order to simplify computation, we also add null strokes to the pattern graph. (A null stroke is just a stroke with a null flag set). The number of null strokes in the pattern graph is the same as the number of nullable strokes in the template graph. For example, if there are two nullable strokes in the template graph, say S_{r1}, S_{r2}, we must add two null strokes to the pattern graph, S_{n1}, S_{n2}. Each null stroke in the pattern can only match exactly one nullable stroke in the template and nothing else. In mathematical terms,

$$\text{feasibility } \alpha(S_{r1}, S_{n1}) = 1.0;$$

and

$$\alpha(S_r, S_{n1}) = 0.0 \; \forall S_r \neq S_{r1}$$

and also

$$\alpha(S_{r2}, S_{n2}) = 1.0,$$

and

$$\alpha(S_r, S_{n2}) = 0.0, \forall S_r \neq S_{r2}.$$

For compatibility,

$$\beta((S_{r1}, S_r), (S_{n1}, S_p)) = 1.0 \; \forall \; S_r \text{ in the template graph, } S_p \text{ in the pattern graph.}$$

This is because the null stroke does not actually exist, so is the arc joining a null stroke with other strokes. Hence the matching always succeeds. For radicals with more than one nullable stroke, a flag BOTHNULL is kept for the relation to indicate whether both nullable strokes can match with null strokes. If BOTHNULL is set, then both nullable stroke can match with null strokes. Otherwise, if both are null strokes, $\beta = 0.0$.
4) *Incremental matching in Chinese characters*: Many radicals are part of another radical, for example, 土, 王 and 日, 目. After matching a particular character with the radical 土, the information can be used to perform matching for the radical 王 since if the character does not match 土, it will not match 王 either. If a matching {(1,s_1), (2,s_2), (3,s_3)} exists for a particular character with radical 土, then when matching with radical 王, we can delete all feasible nodes (i, s) $\forall s \neq s_i$ in the net.

TABLE IV
ATTRIBUTE VALUES OF THE EDGES OF TEMPLATE 土

Relation	Vertical Relation			Check	Horizontal Relation			Check	Joint Type					Check
	Top	Below	Norel	Flag	Left	Right	Norel	Flag	T-f	T-into	H-T	X	//	Flag
(1,2)	0.00	0.00	1.00	F	0.00	0.00	1.00	F	0.00	0.00	0.00	1.00	0.00	T
(1,3)	1.00	0.00	0.00	T	0.00	0.00	0.00	F	0.00	0.00	0.00	0.00	0.00	F
(2,3)	1.00	0.00	0.00	T	0.00	0.00	1.00	F	0.00	1.00	0.00	0.00	0.00	T

VIII. AN ILLUSTRATIVE EXAMPLE

Fig. 4(a) shows the original character 去 and Fig. 4(b) is the thinned version, with strokes found by our stroke segmentation algorithm. The underlying graph of the character is then formed and shown in Fig. 5(a). The attributes are determined by using the standard function S, π [16] and P [13] for membership evaluation. For example, the attribute value of $vertical$ in $STROKE\_TYPE$ is calculated as

$$\mu_{STROKE\_TYPE}(vertical) = P(\delta, 1.5, 0.4, 2)$$

where $\delta \in [0, 8)$ is the slope of the stroke, in chain code convention. Some attributes are combined from more primitive fuzzy sets. For example, the membership of T-joint in attribute *JOINT_TYPE* is given by the logical expression,

The intersection point is in the $middle$ of stroke 1 and *at one end* of stroke 2 where $middle$ and *at one end* are more primitive fuzzy sets where the membership is determined by the standard S, π, and P functions. Further details on attribute evaluation can be found in [13].

Fig. 6 shows the block diagram of our system. The characters were represented as 64 × 64 dot matrices. They were first thinned and undergone stroke segmentation. Then the attributes for strokes and relations were extracted. FAG matchings were performed to find the radicals existing in the character. Finally, a decision logic is used to perform the final classification. Undirected graph is used in the implementation.

Table I shows the attribute values of the vertices (strokes) of the character 去 while Table II shows the relational attribute values. Fig. 4(c) is the radical 土 and Fig. 5(b) is the underlying graph. Table III and IV are the corresponding attribute values of the vertices and relations. As discussed earlier, the template is not fuzzy and so the membership value is either 1.0 or 0.0. This also illustrates how the nonfuzzy version can be represented as a FAG.

When the character 去 is matched against the template 土, we obtain the monomorphism (1,6), (2,2), and (3,4) with

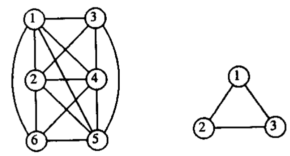
$$\alpha_{(1,6)} = 1.00, \qquad \beta_{(1,2)(6,2)} = 1.00,$$
$$\alpha_{(2,2)} = 1.00, \qquad \beta_{(1,3)(6,4)} = 1.00,$$
$$\alpha_{(3,4)} = 0.97, \qquad \beta_{(2,3)(2,4)} = 1.00.$$

The degree of matching, γ = 0.97 and the monomorphism (1,6), (2,2), (3,5) with

$$\alpha_{(1,6)} = 1.00, \qquad \beta_{(1,2)(6,2)} = 1.00,$$
$$\alpha_{(2,2)} = 1.00, \qquad \beta_{(1,3)(6,5)} = 1.00,$$
$$\alpha_{(3,5)} = 1.00, \qquad \beta_{(2,3)(2,5)} = 1.00,$$

The degree of matching, γ = 1.00.

After FAG matching, we then find a set of disjoint radicals that maximally covers the character. We first construct an undirected graph G, each node representing a radical. A link exists between two nodes if the two radicals concerned are disjoint with each other. Then a disjoint radical set is just a subgraph of G that is complete. Hence a maximal disjoint radical set is just a maximal complete subgraph of G that is not contained in any other complete subgraph ($clique$). The algorithm by Bron and Kerbosch [17] is used for this purpose. A list of possible candidates is determined from this result.



(a) (b)

Fig. 5. Underlying graph of (a) character 去. (b) Template 土.

Since one radical may be a subset of another radical, e.g., 土 is a subset of 王, not all cliques found should be considered. However, we do not want to use subset relation of strokes as our mere criterion to determine unsuitable cliques, since the stroke set of one clique may be a subset of another, yet they may have totally different sets of radicals. We would like to cater for this possibility. Hence we introduce another criterion.

Definition 11: Consider two radical sets (Cliques) C_1 and C_2. We say that the radical set C_1 $covers$ the radical set C_2 if for every radical r_2 in C_2, we can find distinct radicals r_1 in C_1 such that the stroke set of $r_2 \subseteq$ stroke set of r_1, with at least one radical being a proper subset.

However, performing the aforementioned checking for each pair of cliques found will be rather inefficient as the number of cliques found may be rather large. In order to reduce computation, we would like to have an equivalent checking that is independent of number of cliques found.

Let R be the set of all radicals found.

Theorem 2: A radical set C is covered by some other radical set iff $\exists$ radical $r_n \in R$ and $r \in C, r_n \supset r$ such that $\forall r' \in C$ and $r' \neq r$,

$$r_n \cap r' = \varnothing$$

The proof can be found in [13] and is omitted here.

We can now only check the *parents* of each radical in the clique (those radicals that are supersets of the radical) and see whether they are disjoint with other members of the clique. The checking no longer depends on the number of cliques found.

Further improvement can be made by incorporating this checking into the clique search algorithm.

Corollary: Let C be a radical set not covered by any other radical set. Then $\forall r \in C$ and $\forall s \in R$, if $s \supset r$ then $\exists r' \in C$ and $r' \neq r$ s.t.

$$s \cap r' \neq \varnothing$$

This is just the contrapositive form of Theorem 2.

From the corollary, we can see that there must exist a radical that is not disjoint with the parent of some other radicals. Hence, during the search, we can maintain a list of difference-set that is the stroke set of the difference between the radicals selected and their *parents*, which are not yet covered by any radical selected. (By cover we mean nonempty intersection). Further selection of element will require that it covers one of the difference-set in the list.

From the cliques found, a list of possible candidates can be determined. The pattern then go through a final matching (using the radical found as primtives) to determine which candidate is

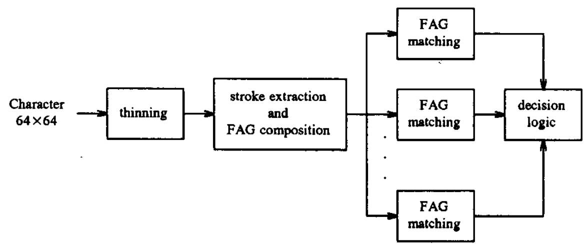


Fig. 6. Block diagram of the system.

the correct recognition. Information from preclassification can be employed during this stage. A preclassification stage using modified fuzzy isodata [18] was used. When the distance (fuzzy Mahalanobis distance) between the sample and the class template is smaller than three times the average of the distances of the training samples, that template will be included as a candidate for the final matching.

The described method was applied to recognize the most frequently used 240 Chinese characters. A total of 8980 samples were used and 8086 samples were correctly recognized, which corresponds to a recognition rate of 90.06%; 140 samples were wrongly recognized corresponding to an error rate of 1.6%. The rest were considered as rejection. When information from preclassification was used (as described previously), the recognition rate was improved to 91.8%. This suggests that the error pattern for preclassification and the final recognition are quite different and a combination of the two approach can give improvement in recognition rate.

IX. Conclusion

In this paper, we introduce the concept of fuzzy-attribute graph by extending attributed graph to handle fuzzy attributes. We also propose a measure on the degree of matching on fuzzy-attribute graph monomorphism. The algorithm for subgraph matching is adapted from [15] and modification have been made to handle the fuzzy attributes.

We also present a system for Chinese character recognition using approach different from the traditional ones. The authors believe that human perception is based on the seeking of identifiable sub-patterns in a complex scene and giving an interpretation of the scene from those identified parts and their relations. The current approach reflects this belief.

Subgraph matching is well known to be NP-complete. In the worst case, exponential time will be required. However, the current method matches a given character against template radicals that are usually much smaller in size (i.e., number of vertices). The method is readily applicable in parallel processing environment as all subgraph matchings can be done in parallel. Another advantage of the current approach is the use of incremental matching. For Chinese characters, some radicals are in fact a part of other radicals, e.g., the radical pairs 土 and 王. After the former radical is found, we can use it as a basis to perform subgraph matching on the latter. Templates of similar shape having common basis can be assigned to the same processor. Information from preclassification as well as simple parameters like position and number of strokes can be used to eliminate obvious infeasible radicals.

Finally, the concept of null stroke is introduced to handle missing strokes during subgraph matching. Strokes in the radicals templates can be classified into either nullable strokes or not. For every nullable stroke in a radical, there is exactly one null stroke in the pattern graph that can match this nullable stroke only and nothing else. From our experience, we know that some null strokes are tolerable while others are not. With this modification we can handle missing stroke with minimal effort.

There are some limitations in our current system. Since we are using radical matching rather than character matching, automatic learning of radicals will be very difficult. During the learning phase, the correct radicals must be extracted from different characters and the process should be under the supervision of human being. No attempt have been directed to automate this learning process in our current study. Instead, the radicals are defined by the user based on his knowledge in Chinese characters.

Improvement of the current system can be achieved by 1) improving the thinning algorithm, 2) improving the stroke segmentation process or using a stroke segmentation without thinning and 3) designing an optimal set of attributes. Most of the rejection or misrecognition, as observed, are caused by error in stroke extraction. This is either due to the thinning process, which inevitably introduces noise or due to the stroke extraction process itself. By improving the stroke extraction stage, we can improve both the recognition rate and the efficiency.

References

[1] W. K. Leow, "Syntactic approach to Chinese character recognition," in *Proc. 1986 Int. Conf. Chinese Computing*, Singapore, 1986, pp. 124-130.
[2] J. W. Tai, "A syntactic-semantic approach for Chinese character recognition," in *Proc. 7th Int. Conf. Pattern Recognition*, Montreal, PQ, Canada, 1984, pp. 374-376.
[3] K. Yamamoto and A. Rosenfeld, "Recognition of handprinted Kanji characters by a relaxation method," in *Proc. 6th Int. Conf. Pattern Recognition*, Munich, Germany, 1982, pp. 395-398.
[4] C. H. Leung, Y. S. Cheung, and Y. L. Wong, "A knowledge-based stroke-matching method for Chinese character recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-17, Nov./Dec., pp. 993-1003, 1987.
[5] L. A. Zadeh, "Fuzzy sets," *Inform. Contr.*, vol. 8, pp. 338-353, 1965.
[6] W. H. Tsai and K. S. Fu, "Error correcting isomorphism of attributed relational graphs for pattern analysis," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 12, Dec. 1979, pp. 757-768.
[7] A. K. C. Wong and M. You, "Entropy and distance of random graphs with application to structural pattern recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 5, Sept., pp. 599-609, 1985.
[8] L. G. Shapiro and R. M. Haralick, "Structural descriptions and inexact matching," *IEEE Trans. Pattern Anal. Machine Inntell.*, vol. PAMI-3, no. 5, pp. 504-519, Sept. 1981.
[9] K. L. Boyer and A. C. Kak, "Structural Stereopsis for 3-D Vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 2, pp. 144-166, Mar. 1988.
[10] O. D. Faugeras, "Image understanding and graph matching," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Paris, France, 1982, pp. 1162-1165.
[11] F. H. Cheng and W. H. Hsu, "Fuzzy recognition of handwritten Chinese characters," in *Proc. 1988 Int. Conf. Computer Processing of Chinese and Oriental Languages*, Toronto, ON, Canada, 1988, pp. 28-34.
[12] C. Granger, "Application of possibility theory to object recognition," *Fuzzy Sets Syst.*, vol. 28, no. 3, Dec. 1988, pp. 351-362.
[13] K. P. Chan, "Fuzzy set theoretic approach to handwritten Chinese character recognition," Ph. D. Thesis, Univ. Hong Kong, 1989.
[14] A. V. Aho, J. E. Hopcroft, and J. D. Ullmann, *The Design and Analysis of Computer Algorithm*. Reading, MA: Addison-Wesley, 1974.
[15] F. A. Akinniyi, A. K. C. Wong, and D. A. Stacey, "A new algorithm for graph monomorphism based on the projections of the product graph," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, pp. 740-751, Sept/Oct. 1986.
[16] A. Kandel, *Fuzzy Mathematical Techniques with Applications*. Reading, MA: Addison Wesley, 1986.
[17] C. Bron and J. Kerbosch, "Algorithm 457: Finding all cliques of an undirected graph [H]," *Commun. ACM*, vol. 16, no. 9, pp. 575-577, Sept. 1973.
[18] Y. S. Cheung and K. P. Chan, "Modified fuzzy isodata for the classification of handwritten chinese characters," in *Proc. 1986 Int. Conf. on Chinese Computing*, Singapore, 1986, pp. 361-364.

[19] S. Mori, K. Yamamoto, and M. Yasuda, "Research on machine recognition of handprinted characters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol PAMI-6, pp. 386-404, July 1984.

[20] H. Yamada, "Contour DP matching method and its application to handprinted chinese character recognition," *Proc. 7th Int. Conf. Pattern Recog.*, Montreal, PQ, Canada, 1984, pp. 374-376.

[21] P. S. P. Wang, "On-line Chinese character recognition," *Proc. IGC Electron. Imaging Conf.*, vol. 1, Boston, Oct. 1988, pp. 209-214.

[22] A. Kaufmann, *Theory of Fuzzy Subsets,* vol. I. New York: Academic, 1975.

[23] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications.* New York: Academic, 1980.

[24] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes" *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 28-44, Jan. 1973.

[25] R. E. Bellman, and M. Giertz, "On the analytic formalism of the theory of fuzzy sets" *Inf. Sci.*, vol. 5, pp 149-156, 1973.

[26] A. Rosenfeld, "Fuzzy graph," in *Fuzzy Sets and Their Application to Cognitive and Decision Processes*, L. A. Zadeh *et. al.*, Ed. New York: Academic, 1975, pp. 77-95.