Large Vocabulary Sign Language Recognition Based on Fuzzy Decision Trees

Gaolin Fang, Wen Gao, *Member, IEEE*, and Debin Zhao

***Abstract*—The major difficulty for large vocabulary sign recognition lies in the huge search space due to a variety of recognized classes. How to reduce the recognition time without loss of accuracy is a challenging issue. In this paper, a fuzzy decision tree with heterogeneous classifiers is proposed for large vocabulary sign language recognition. As each sign feature has the different discrimination to gestures, the corresponding classifiers are presented for the hierarchical decision to sign language attributes. A one- or two- handed classifier and a hand-shaped classifier with little computational cost are first used to progressively eliminate many impossible candidates, and then, a self-organizing feature maps/hidden Markov model (SOFM/HMM) classifier in which SOFM being as an implicit different signers' feature extractor for continuous HMM, is proposed as a special component of a fuzzy decision tree to get the final results at the last nonleaf nodes that only include a few candidates. Experimental results on a large vocabulary of 5113-signs show that the proposed method dramatically reduces the recognition time by 11 times and also improves the recognition rate about 0.95% over single SOFM/HMM.**

***Index Terms*—Finite state machine, fuzzy decision trees, hidden Markov models (HMM), self-organizing feature maps (SOFM), sign language recognition.**

I. Introduction

SIGN LANGUAGE as a kind of gesture is one of the most natural ways of exchanging information for most deaf people. The goal of sign language recognition (SLR) is to provide an efficient and accurate mechanism to transcribe sign language into text or speech so that communication between deaf and hearing society can be more convenient. SLR, as one of the important research areas of human-computer interaction (HCI), has spawned more and more interest in HCI society. From a user's point of view, the most natural way to interact with a computer would be through a speech and gesture interface. Thus, the research of sign language and gesture recognition is likely to provide a shift paradigm from point-and-click user interface to a natural language dialogue-and-spoken command-based interface. In addition, it has many other applications, such as controlling the motion of a human avatar in a virtual environment via hand gesture recognition, learning demonstration for the robot, and multimodal user interface in a virtual reality system.

Unlike general gestures, sign language is highly structured so that it provides an appealing test bed for new ideas and algorithms before they are applied to gesture recognition. Attempts to automatically recognize sign language began to appear in the literature in the 1990s. The recognition methods usually include rule-based matching, artificial neural networks (ANNs), and hidden Markov models (HMMs).

Kadous [1] demonstrated a system based on Powergloves to recognize a set of 95 isolated Australian sign language with 80% accuracy. Instance-based learning and decision-tree learning were adopted by the system to produce the rules of the pattern. Matsuo *et al.* [2] used a similar method to recognize 38 signs from Japanese sign language with a stereo camera for recording three-dimensional (3-D) movements. Morphological analysis was used in their method to get sign language patterns.

Fels and Hinton [3] developed a system using a Dataglove with a Polhemus tracker as input devices. In their system, five neural networks were employed for classifying 203 signs. Kim *et al.* [4] used fuzzy min-max neural network and fuzzy logic approach to recognize 31 manual alphabets and 131 Korean signs based on Datagloves. An accuracy of 96.7% for manual alphabets and 94.3% for the sign words were reported. Waldron and Kim [5] also presented an expandable SLR system using the self-organizing maps to recognize a small set of isolated signs. They used Stokoe's transcription system to separate hand shape, orientations and motion aspects of the signs.

Grobel and Assan [6] used HMM to recognize isolated signs with 91.3% accuracy out of a 262-sign vocabulary. They extracted two-dimensional (2-D) features from video recordings of signers wearing colored gloves. HMM was also employed by Hienz and Bauer [7] to recognize continuous German sign language with a single color video camera as input. Their research was an extension of the work by Grobel and Assan. An accuracy of 91.7% can be achieved in recognition of sign language sentences with 97 signs.

Liang and Ouhyoung [8] employed the time-varying parameter threshold of hand posture to determine end-points in a stream of gesture input for continuous Taiwan SLR with the average recognition rate of 80.4% for 250 signs. In their system, a Dataglove was used as an input device, and HMM was taken as recognition method.

Starner *et al.* [9] used a view-based approach for continuous American SLR. They used single camera to extract 2-D features and the extracted features were then taken as the input of HMM.

Manuscript received September 4, 2002; revised July 28, 2003 and December 1, 2003. This work was supported in part by the Natural Science Foundation of China under Grant 60303018 and by the National High-Technology Development "863" Program of China under Grant 2001AA114160. This paper was recommended by Associate Editor W. Pedrycz.

G. Fang was with the Department of Computer Science, Harbin Institute of Technology, Harbin 150001, China. He is now with the Joint Research and Development Lab, Chinese Academy of Sciences, Beijing 100080, China (e-mail: glfang@jdl.ac.cn).

W. Gao is with the Institute of Computing Technology, Beijing 100080, China (e-mail: wgao@ict.ac.cn).

D. Zhao is with the Department of Computer Science, Harbin Institute of Technology, Harbin 150001, China (e-mail: dbzhao@jdl.ac.cn).

Digital Object Identifier 10.1109/TSMCA.2004.824852

1083-4427/04$20.00 © 2004 IEEE

The word accuracy of 92% or 98% was gotten when the camera was mounted on the desk or in a user's cap in recognizing the sentences with 40 different signs.

Vogler and Metaxas [10] used computer vision methods to extract the 3-D parameters of a signer's arm motions as the input of HMM, and recognized continuous American sign language sentences with a vocabulary of 53 signs. The reported best accuracy is 95.83%. In addition, they used phonemes instead of whole signs as basic units and achieved similar recognition rates to sign-based approaches over a vocabulary of 22 signs [11], [12].

From the review above, we know that most researchers focus on small or medium vocabulary SLR in the signer-dependent field. For signer-independent SLR, only Vamplew and Adams [13] reported a signer-independent system based on a Cyberglove to recognize a set of 52 signs. Their system employed a modular architecture consisting of multiple feature-recognition neural networks and a nearest-neighbor classifier to recognize isolated signs. They got 94% recognition rate in the registered test set and 85% in the unregistered test set. To the best of our knowledge, no research report was found in the literature on large vocabulary signer-independent SLR.

The major challenges that SLR faces now are developing methods that solve signer-independent and large vocabulary problems. Signer independence is highly desirable since it allows a system to recognize a new signer's sign language without retraining the models. The ability to recognize large vocabulary signer-independent sign language has a profound influence on the naturalness of the human-computer interface and is clearly an essential requirement for the widespread use of SLR system.

For signer-independent SLR, there are two difficulties that need to be solved. The model convergence difficulty is caused by noticeable distinctions among different people signs. Since different signers vary their hand shape size, body size, operation habit, rhythm, and so on, the noticeable distinctions between the data of the same sign due to different signers are almost larger than sign variations due to the change in sign identity. The lack of effective features was extracted from different signers' data. Unlike the speech recognition in which every speech feature has been profoundly explored, the feature extraction of sign recognition is still in its infancy. How to effectively extract common features from different signers is even a more challenging problem.

For large vocabulary sign recognition, the major difficulty lies in the huge search space due to a variety of recognized classes. How to reduce the recognition time without loss of accuracy is a challenging issue. In speech recognition, a phoneme-based method was generally employed to tackle large vocabulary problems. However, there is no basic unit defined in the sign's lexical forms. The phonemes extracted manually or automatically were experimented on the small vocabulary, so it is very difficult to extend these phonemes to act as basic units of whole sign language. Our previous system [14], [15] used Datagloves as input devices and HMM as recognition method. The system can recognize 5177 isolated signs with 94.8% accuracy in real time and recognize 200 sentences with 91.4% word accuracy in the signer-dependent field. The state-tying HMM with one mixture component was employed to overcome the time-consuming problem that arises from the large vocabulary size. However, when this method was applied to signer-independent SLR, the recognition performance decreased.

To overcome these difficulties from both large vocabulary and signer-independent problems, a fuzzy decision tree with heterogeneous classifiers is presented for large vocabulary SLR in this paper based on the divide-and-conquer principle. As each sign feature has the different discrimination to gestures, the corresponding classifiers are proposed for the hierarchical decision to sign language attributes. A one- or two- handed classifier based on the Gaussian mixture model (GMM) and hand shape classifier based on finite state machine (FSM) with little computational cost are first used to progressively eliminate many impossible candidates, and then SOFM/HMM classifier is proposed as a special component of fuzzy decision tree to get the final results at the last nonleaf nodes that only include few candidates. To alleviate the effect of crisp classification errors, fuzzification is introduced in the decision tree, i.e., the classes that cannot be robustly classified will not be handled at this classifier, and they simultaneously enter next level for further decision. Experimental results show that the proposed method can dramatically reduce the recognition time and also improve the recognition performance over single SOFM/HMM.

The remainder of this paper is organized as follows. In Section II, we analyze sign language features. Section III proposes feature classifiers of the fuzzy decision tree. In Section IV, the fuzzy decision tree for SLR is presented. Section V shows experimental results. The conclusions are given in the last section.

II. Sign Language Features

According to Stokoe's definition [16], each sign can be broken into four parameters: hand shape, orientation, position and motion. These parameters as four important features play an important role in sign language recognition. Furthermore, according to the number of participating hands in the sign performance, sign language can be divided into two categories: one-handed signs or two-handed signs. Thus, one- or two-handed is also one of the important features of sign language. Five features are, respectively, detailed as follows.

Hand shapes are one of the primitives of sign language and reflect the information of hand configuration. They are very stable and can be used to distinguish most signs. In the Chinese sign language dictionary, there are 75 basic hand shapes extracted by the sign language experts.

The orientation of the hand can be described in terms of two orthogonal directions — the facing of the palm and the direction to which the hand is pointing. If we consider only six possible directions (up, down, left, right, toward the signer, away from the signer), then there are 15 different orientations used in Chinese sign language (CSL).

The position of the hand is usually partitioned into three parts: head, chest, and below chest in terms of the hand with respect to the body part. In each part, the position can be further subdivided into body's left, right, and middle. In total, there are 12 positions defined in CSL.

Motion differs from other sign language features in that it is inherently temporal in nature. It is difficult to enumerate the complete range of possible categories used within CSL, because many signs involve unique tracing motions that indicate the shape of an object. For this research only the 13 most commonly used motions were defined.

In the CSL dictionary, one-handed signs are always performed by right hand except for one sign "luo-ma-ni-ya" by left hand. The difference between one-handed sign and two-handed sign is whether signer's left hand participates the action. In the one-handed sign performance, signer's left hand usually puts on the left knee and remains motionless. However, in the two-handed sign, left hand may either stay a fixed posture or perform a movement trajectory. Thus, the position and orientation information of left hand plays a dominant part in determining one- or two- handed signs.

III. Feature Classifiers in the Fuzzy Decision Tree

In this section, on the basis of the analysis of all the sign language features, GMM is first employed as one- or two-handed classifier, and then FSM-based method is proposed as hand shape classifier. At last, the SOFM/HMM classifier as a special component of fuzzy decision tree is presented for tackling the signer-independent difficulties.

A. One- or Two- Handed Classifier

The GMM is in essence one of the multivariate probability density functions. It has been successfully used as a classifier in a variety of applications. According to the estimation, one-handed sign and two-handed sign probability distributions can be approximately described by GMM. The estimation process is designed as follows: in the training set, one- and two- handed signs are manually labeled, and then we calculate the frequency distributions of one- and two- handed signs at each point for every dimension of the vector. Fig. 1 shows the frequency distributions of one- and two- handed signs in the x-component value of the left hand position vector, where the value of x-axis is the result of the x-component value [0,1] multiplied by 25, and the value of y-axis is the result of the number of the estimated signs with this x-axis value divided by the total number of one- or two- handed signs. The similar distributions are observed in other components of the left hand position and orientation vectors. From the curves of statistical results, we can speculate that the one-handed sign and two-handed sign probability density can be approximated by respective GMM. Therefore, in this paper, GMM is employed to determine whether a gesture is represented by one hand or two hands.

GMM can be described by the mixture parameter, the mean vector and the covariance matrice, and formulated as $\lambda = \{\pi, \mu, \Sigma\}$. The probability density function of an observation x is represented by the linear combination of Gaussian density

$$p(x|\lambda) = \sum_{i=1}^{M} \pi_i p(x|i) \tag{1}$$

where x is the observed vector of d dimensions, M is the number of mixture term, and π_i is the mixture parameter and satisfies the constraint $\sum_{i=1}^{M} \pi_i = 1$.

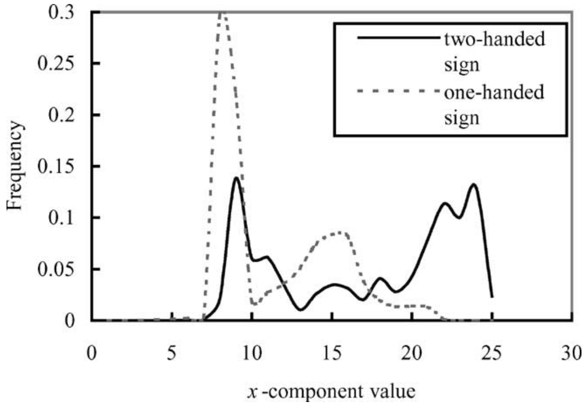


Fig. 1. Frequency distributions of one- and two-handed signs in the x-component value of the left hand position vector.

$p(x|i)$ is the Gaussian probability density function of d dimensions

$$p(x|i) = \frac{\exp\left\{-\frac{1}{2}(x-u_i)^T \Sigma_i^{-1}(x-u_i)\right\}}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \tag{2}$$

where u_i is the mean vector, and Σ_i is the covariance matrice.

The parameter λ can be trained using the expectation-maximization (EM) algorithm [17], [18]. There are two key issues in the model training: initialization and the selection of training samples.

Initialization: In the EM algorithm, the parameter λ needs to be initialized. However, no general theoretical framework but empirical or experimental appoach is employed to solve this problem. Here, k-means clustering is used to get the mean and convariance of centriods as the initialization values. π_i^0 is initialized to $1/M$. The mixture term M is determined according to the distribution of training data and the classification accuracy of one- or two- handed signs. In our classification experiments, M is set to the values from 5 to 30. From the experiments, the classification performance grows with the mixture term M. When the value is greater than 15, the classification performance doesn't improve or even slightly decline but the classification time increases. Thus, M in the one-handed GMM and two-handed GMM are both set to 15, where the mixture terms are set to the same for the comparability of their probabilities.

The selection of training samples. How to select typical samples for training one-handed and two-handed GMM is a difficult issue. Different training data will produce different results, that is, too many data will make the model training difficult to converge, and not enough data will train the model that cann't be well generalized. Through the experiments, the following strategy is taken. For a one-handed sign, left hand stays motionless and its data are very stable, so the stablest frame is extracted from all the frames of this word. For a two-handed sign, if left hand is in motion, then all the data frames are extracted as the training data. If left hand is motionless, the extraction method is the same as one-handed signs. In all those training data, only the position and orientation information of left hand is used for classification.

Classification: Given the frame sequence for one sign $O = O_1 O_2 \cdots O_T$, the probabilities of belonging to one-handed and two-handed signs for each frame are calculated with the trained GMM. The probabilities can be expressed as $P_i(O_j)$, $i = 1, 2,$

where 1 denotes the one-handed sign and 2 for the two-handed sign. The classes can be gotten through the following formula:

$$i^* = \arg \max_{i \in \{1,2\}} \left(\sum_{j=1}^{T} \delta_i(O_j) \right)$$
$$\delta_k(O_j) = \begin{cases} 1 & \text{if } k = \arg\max_{i \in \{1,2\}} P_i(O_j) \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

After all the training samples are classified using above method, the candidate words associated with one handed sign or two-handed sign are generated, which will be used by the following hand shape classifier.

B. Hand Shape Classifier

Lee *et al.* [19] used FSM to segment the motion of Korean sign language. Hong *et al.* [20] also used FSM to recognize gestures, where each state is modeled as a multivariate Gaussian function. A gesture can be described as an ordered sequence of hand shape states in spatial-temporal space and well modeled by a FSM, whose states consist of 75 basic hand shapes. The structure of an FSM is like that of an HMM, where each state can jump to either itself or its next state. FSM has the advantages of easy interpretation and faster classification rate, and can well solve the different frame alignment for the same sign. Furthermore, hand shape is very stable in all the features of sign language and it plays a very important role in distinguishing most signs due to its distinct feature discrimination, so it is feasible to use it as a classifier of the fuzzy decision tree. Thus, in this paper, FSM-based method is proposed for hand shape classifier, which is viewed as part of the fuzzy decision tree.

The training algorithm of FSM-based hand shape classifier is described as follows.

1) Clustering. Basic hand shapes extracted by the experts from the dictionary of sign language are regarded as initial centroids, and then the k-means clustering algorithm is employed to get new centroids in the training set.
2) Fuzzy vector quantization (FVQ). Fuzzy N-best results are outputted at each frame with new centroids. For utilizing the context information of sign frames to supervise the quantization, the Viterbi algorithm is employed to get the best vector quantization sequence on the fuzzy N-best results.
3) Pattern extraction and pruning. The word patterns are extracted from the quantization results. Pattern to word is a many-to-many map, where the patterns are regarded as the classification criterion. For the better generalization ability, simple pruning is operated on the extracted patterns.
4) Candidate word generation. Training samples are classified using FSM, and each branch in the classifier denotes one pattern. After all the samples in the training set are handled, the candidate word set associated with each pattern is generated, which will be used by SOFM/HMM classifier.

To eliminate the effect of noise and make the extracted patterns have better generalization ability, two key techniques are employed in this algorithm: fuzzy vector quantization, and pattern extraction and pruning.

1) Fuzzy Vector Quantization: In the vector quantization, every frame data is independent of each other, so the noise will have the direct influence on the quantization results. However, a gesture usually consists of several basic hand shapes, and the changes between hand shapes are very stable, that is, slow changes from one series of hand shapes to another series of hand shapes. This context information can be utilized to supervise the quantization through FVQ so that the algorithm can reduce the effect of noise and get the well-generalized patterns.

Given the frame sequence for one sign $O = O_1 O_2 \cdots O_T$, where T denotes the frame number. Define $V = V_1 V_2 \cdots V_T$, $V_t \in \{1, 2, \ldots, 75\}$ as one of the quantization sequence. In FVQ, the vector O_t is quantized as N top scoring outputs rather than only one top output, and the corresponding probabilities associated with N top outputs are also outputted, where $N = 3$. At last, the Viterbi algorithm is employed to get the best vector quantization sequence among all the results, and formulated as

$$V^* = \arg\max_{V} b_{V_1}(O_1) \prod_{t=2}^{T} a_{V_{t-1}V_t} b_{V_t}(O_t). \tag{4}$$

Transition probability between the quantization results V_{t-1}, V_t is defined as follows: $a_{V_{t-1}V_t} = \begin{cases} 1 & V_{t-1} = V_t \\ 0.1 & V_{t-1} \neq V_t \end{cases}$, where the values of 1 and 0.1 are manually set through the experiments.

The emission probability of the frame O_t being quantized as V_t is defined: $b_{V_t}(O_t) = \exp(-d_{V_t}(O_t)) / \sum_V \exp(-d_V(O_t))$, where $d_{V_t}(O_t)$ denotes the Euclidean distance between the vector O_t and the centroid V_t.

An example of fuzzy vector quantization is shown in Fig. 2, where V_t^1, V_t^2, and V_t^3 represent the three-top outputs for the frame O_t. Through the Viterbi algorithm search on all the likely paths, the final results of fuzzy vector quantization are V_1^1, V_2^2, V_3^1, V_4^1, and V_5^2.

Through this method, the context information of sign frames is fully utilized to supervise the vector quantization, so we can alleviate the effect of the noise data and get the consistent quantization results.

2) Pattern Extraction and Pruning: Pattern extraction is performed as follows. After the previous step — fuzzy vector quantization — processing, the quantized sequences are so regular that the classification patterns can be directly extracted from the quantization results according to the duration of hand shape. If the duration is greater than four frames, then this hand shape is regarded as one of the pattern states, otherwise regarded as the noise and discarded.

However, the extracted pattern number is very large so that the generalization of patterns becomes delicate and the corresponding classification deviation is getting large. To solve this problem, the pruning is performed on those patterns. From the data of FVQ, if all the patterns are kept, the total number of the

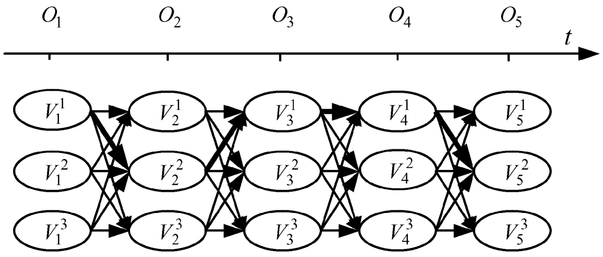


Fig. 2. Example of fuzzy vector quantization.

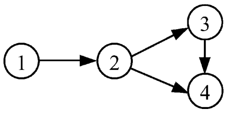


Fig. 3. Examples of hand shape pattern.

classification template for all the vocabulary is about 1860. If the first three hand-shape states are kept, the number is about 1340. If the first two hand-shape states are kept, the number is about 450. If the first hand-shape state is kept, the number is about 75. For making the extracted patterns have better generalization, the number of classification pattern cannot get too big. Compromising the generalization and the classification time, the first two hand shape states are kept, which is in accord with the fact that most of the signs consist of two hand shape states. For example, three quantized word sequences: 1 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 3 4 4 4 4, 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 and 1 1 1 1 1 1 1 2 2 2 2 2 4 4 4 4 4 4 are, respectively, extracted as the patterns 1 2 3 4, 1 2 3 and 1 2 4 (see in Fig. 3, where 1, 2, 3 and 4 denote basic hand shape state). They will be pruned as one pattern 1 2, which will be used as the classification template. For the words with the long frame data, the pruning can distinctly reduce the classification time, because only previous parts are used to distinguish rather than the whole sequence. After the pruning, the patterns are regarded as the classification templates of FSM.

Classification: Input data are processed by fuzzy vector quantization, pattern extraction and pruning, then classified into the corresponding pattern branch through FSM.

Similar methods are experimented with position, orientation and motion features. However, since the information of these features is not very stable, the extracted patterns for the same sign are not very consistent. Though the recognition time is reduced, the performance cannot be improved. Thus, one- or two-handed, left hand shape, and right hand shape are chosen as three attributes of the fuzzy decision tree.

C. SOFM/HMM Classifier

Aiming at two difficulties of signer-independent SLR—the model convergence difficulty caused by noticeable distinctions among different people signs, and the lack of effective features extracted from different signers' data, SOFM/HMM classifier is presented in this paper. The proposed method uses self-organizing feature maps (SOFM) as an implicit different signers' feature extractor for continuous HMM to allow signer independence and its parameters are trained simultaneously in a global optimization criterion. SOFM transforms input sign representations into significant and low-dimensional representations that can be well modeled by the emission probabilities of HMM.

SOFM/HMM is different from the ANN/HMM models [21] used in speech recognition in terms of the architecture and the corresponding training algorithm. The proposed method is also different from Corradini's approach [22] because the parameters in our method are trained simultaneously in a global optimization criterion. In SOFM/HMM, each eigenvector centroid of SOFM is regarded as one of the components in the state of HMM. The state probability density functions (pdf) of HMM are constructed in the form of the weighted sum of components. The state pdf can be computed through the forward-backward procedure or the Viterbi algorithm. The weights of SOFM are iteratively updated in the supervision of the computed state pdf.

1) SOFM/HMM Architecture: Let the vector of observation sequence $O_t = [o_{t1}, o_{t2}, \ldots, o_{tn}]$, $t = 1, \ldots, T$, where t is the time of observation sequence, and n is the dimension of the vector. The input vector O_t is linked with the SOFM/HMM neuron m by the weight vector W_{jm}, where j is the state variable and $W_{jm} = [w_{jm1}, w_{jm2}, \ldots, w_{jmn}]$. Denote M as the variable set of SOFM neurons and $|M|$ as the number of elements in the set.

Fig. 4 shows the architecture of a three-state left-right model with skip. The input vector O_t at time t for all states is first transformed into the corresponding neurons by the linking weight W_{jm}, and then the neurons make up of the state pdf $b_j(O_t)$ of HMM in the form of the weighed sum.

The contribution to the state pdf of the probability of being the mth neuron in state j can be constructed as follows.

$$b_{jm}(O_t) = k \exp[-D(W_{jm}, O_t)] \tag{5}$$

where k is a constant. $b_{jm}(O_t)$ is the mth neuron's contribution to the state pdf, and it gradually decreases as the observation vector deviates from the corresponding neuron. $D(W_{jm}, O_t)$ is the Euclidean distance between the observation O_t and the neuron m.

Since the contribution to the state pdf $b_j(O_t)$ varies from different neurons, the weights that reflect the importance of the contribution are associated with different $b_{jm}(O_t)$. So $b_j(O_t)$ is defined as follows.

$$b_j(O_t) = \sum_{m=1}^{|M|} c_{jm} b_{jm}(O_t) \tag{6}$$

where $\sum_{m=1}^{|M|} c_{jm} = 1$. W_{jm} in (5) and c_{jm} in (6) are computed through the following re-estimation formulas.

2) SOFM/HMM Parameter Estimation: Denote the set of K observation sequences for one sign as $O = [O^{(1)}, O^{(2)}, \ldots, O^{(K)}]$, where $O^{(k)} = [O_1^{(k)} O_2^{(k)} \cdots O_{T_k}^{(k)}]$ is the kth observation sequence, and T_k is the frame number of the kth observation sequence. The initial model is defined as λ, and the re-estimated model $\overline{\lambda}$. Q is a state sequence, and denoted as $Q = q_1, q_2, \ldots, q_{T_k}$, $q_i \in \{1, 2, \ldots, N\}$, where N is the number of states.

Assuming each observation sequence independent of every other observation sequence, a global optimization criterion is to adjust the parameters of the model λ to maximize K observation sequences, and formulated as $\lambda^* = \arg\max_\lambda P(O|\lambda)$, where $P(O|\lambda) = \prod_{k=1}^{K} P(O^{(k)}|\lambda)$. Let $P_k = P(O^{(k)}|\lambda)$ and $w_k = P(O|\lambda)/KP_k$, then $P(O|\lambda)$ can be expressed as: $P(O|\lambda) =$

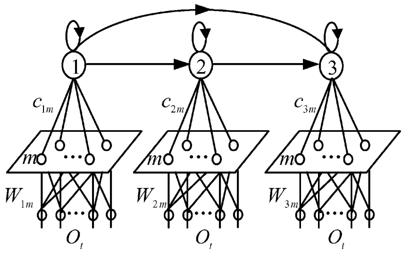


Fig. 4. Architecture of a three-state left-right SOFM/HMM model with the input vector O_t at time t for all states.

$\sum_{k=1}^{K} w_k P_k$. Since $P(O|\lambda)$ depends on the hidden state variable Q and the SOFM neurons variable M, it cannot be maximized directly. The maximization of $P(O|\lambda)$ is the problem of maximum likelihood estimation with missing values (i.e., hidden variables). The EM algorithm is a popular algorithm for maximum likelihood estimation given incomplete data samples [18]. It iterates two estimation steps, called expectation (E-step) and maximization (M-step). The E-step estimates the (log-likelihood) expectation of the hidden variable distributions using the knowledge of the current parameters and the observed incomplete data, and then the M-step calculates the maximum likelihood model for the distributions computed in the E-step .

In the E-step, the auxiliary function $Q(\lambda,\bar{\lambda})$ is introduced to facilitate the maximization of $P(O|\lambda)$, and constructed as

$$Q(\lambda,\bar{\lambda}) = \sum_{k=1}^{K} w_k Q_k(\lambda,\bar{\lambda}) \tag{7}$$

where $Q_k(\lambda,\bar{\lambda}) = \sum_Q \sum_M P(O^{(k)},Q,M|\lambda)\log P(O^{(k)},Q,M|\bar{\lambda})$.

It can be proved $Q(\lambda,\bar{\lambda}) \geq Q(\lambda,\lambda) \Rightarrow P(O|\bar{\lambda}) \geq P(O|\lambda)$ with the similar method in [18], [23]. Thus the maximization of $P(O|\lambda)$ is converted to get the critical point of $Q(\lambda,\bar{\lambda})$.

The following section will discuss how to get the critical point of $Q(\lambda,\bar{\lambda})$ (that is, calculate the reestimated formulas).

$\log P(O^{(k)},Q,M|\bar{\lambda})$ can be rewritten as

$$\log P(O^{(k)},Q,M|\bar{\lambda}) = \log\bar{\pi}_{q_1} + \sum_{t=1}^{T_k-1} \log\bar{a}_{q_t q_{t+1}} + \sum_{t=1}^{T_k} \log\bar{b}_{q_t m_t}(O_t^{(k)}) + \sum_{t=1}^{T_k} \log\bar{c}_{q_t m_t} \tag{8}$$

where $m_t \in M$ and $q_t \in \{1,2,\ldots,N\}$.

Inspired by the parameter derivations of HMM, we define $\alpha_t^{(k)}(i)$ and $\beta_t^{(k)}(i)$ as the forward variable and the backward variable given the model λ and the kth observation sequence. They can be computed through forward-backward procedure [24].

The probability of being in state j at time t with the mth neuron accounting for $O_t^{(k)}$ is defined as:

$$\begin{aligned}\Phi_t^{(k)}(j,m) &= P(q_t=j, m_t=m|O^{(k)},\lambda)\\ &= \frac{\alpha_t^{(k)}(j)\beta_t^{(k)}(j)}{\sum\limits_{j=1}^{N}\alpha_t^{(k)}(j)\beta_t^{(k)}(j)} \cdot \frac{c_{jm}b_{jm}(O_t^{(k)})}{b_j(O_t^{(k)})}\end{aligned} \tag{9}$$

where the former part is the proportional probability of state j to all the states, and the latter part $c_{jm}b_{jm}(O_t^{(k)})/b_j(O_t^{(k)})$ is the proportional probability of the mth neuron to all the neurons. The straightforward interpretation is the contribution probability of the mth neuron in state j to the state probability at time t.

Then we can get

$$\begin{aligned}&Q(\lambda,\bar{\lambda})\\ &= \sum_{k=1}^{K}\left(\sum_{i=1}^{N}\sum_{m=1}^{|M|}\Phi_1^{(k)}(i,m)\log\bar{\pi}_i\right.\\ &\quad + \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T_k-1}\sum_{M} P(q_t=i,q_{t+1}=j,M|O^{(k)},\lambda)\log\bar{a}_{ij}\\ &\quad + \sum_{j=1}^{N}\sum_{t=1}^{T_k}\sum_{m=1}^{|M|}\Phi_t^{(k)}(j,m)\log\bar{b}_{jm}(O_t^{(k)})\\ &\quad \left. + \sum_{j=1}^{N}\sum_{t=1}^{T_k}\sum_{m=1}^{|M|}\Phi_t^{(k)}(j,m)\log\bar{c}_{jm}\right)\frac{P(O|\lambda)}{K}.\end{aligned} \tag{10}$$

Through maximizing the individual terms in $Q(\lambda,\bar{\lambda})$, we can get the re-estimation formulas for W_{jm} and c_{jm} as follows:

$$\bar{W}_{jm} = \frac{\sum\limits_{k=1}^{K}\sum\limits_{t=1}^{T_k}\Phi_t^{(k)}(j,m)O_t^{(k)}}{\sum\limits_{k=1}^{K}\sum\limits_{t=1}^{T_k}\Phi_t^{(k)}(j,m)} \tag{11}$$

$$\bar{c}_{jm} = \frac{\sum\limits_{k=1}^{K}\sum\limits_{t=1}^{T_k}\Phi_t^{(k)}(j,m)}{\sum\limits_{k=1}^{K}\sum\limits_{t=1}^{T_k}\sum\limits_{m=1}^{|M|}\Phi_t^{(k)}(j,m).} \tag{12}$$

The formulas for π_i and a_{ij} are the same as the conventional HMM [24].

Training: Every word in the set of vocabulary has several samples collected from different signers. One SOFM/HMM model is built for each word in the vocabulary through the following training procedure:

```
1) Initialize the parameters of $c_{jm}$ and $W_{jm}$;
2) Re-estimate the parameters by (11) and (12);
3) If the convergence criterion is met, terminate the procedure; otherwise replace old parameters with new ones and return to 2).
```

Classification: Given the observation sequence $O = O_1 O_2 \cdots O_T$, the probability $P(O|\lambda_v)$ is computed for each codebook λ_v in the candidate word database $\lambda = \{\lambda_1,\lambda_2,\ldots,\lambda_V\}$. $P(O|\lambda_v)$ is approximated as $P(O|\lambda_v) = P(O,Q^*|\lambda_v)$, where Q^* is the best state sequence among the state spaces for the given O, which can be gotten through the Viterbi algorithm search [24]. The recognized class can be obtained through the following formula:

$$v^* = \arg\max_{1\leq v\leq V} P(O|\lambda_v). \tag{13}$$

IV. Fuzzy Decision Trees for Sign Language Recognition

A general decision tree [25]–[27] consists of root nodes, nonleaf nodes, and leaf nodes, where each leaf node denotes a class. The input data include the values of different attributes and these values are initially put in the root node. By asking questions about the attributes, the decision tree splits the values into different child nodes. At last, which class the input data belongs to is decided at the leaf node.

Decision trees are, by their nature, readily interpretable and well-suited to classification problems. They are also remarkable for their ability to combine diverse information sources. The disadvantage of decision trees lies in their crisp classification. The misclassification at some nodes will result in the nonrecovered classification errors in the subsequent classification.

A fuzzy decision tree [28], [29] is constructed by superimposing a fuzzy structure over the skeleton of a decision tree. Fuzzification is achieved by allowing the possibility of partial membership of a point in the tree nodes. This extension of expressive capabilities transforms the decision tree into a powerful functional approximant that incorporates features of connectionist methods, while remaining easily interpretable. In this paper, somewhat unlike the standard definition, fuzzification is to divide the problem into subproblems with common elements — a soft split of input space into overlapping clusters [30]. Those common elements are the categories that cannot be robustly classified. They will not be handled at this level classifier, and simultaneously enter next level for further decision.

The fuzzy decision tree for SLR is constructed as follows.

1) In the training set, all the training samples are classified using GMM, the candidate words associated with one-handed sign or two-handed sign are generated. Those words cannot be robustly classified will appear in both the candidate words of one-handed signs and those of two-handed signs.
2) For the candidate words of one-handed signs, their training samples are inputted into the right hand shape classifier. After all the training samples are classified using FSM, the candidate words associated with each pattern are generated. Those candidates with common elements will be used as the candidate words of SOFM/HMM classifier.
3) For the candidate words of two-handed signs, the processing is the same as Step 2), first into left hand shape classifier, and then into right hand shape classifier. However, the classification results of left shape are used as the candidate words of right hand shape classifier, and the classification results of right shape are regarded as the candidate words of SOFM/HMM classifier.

After the fuzzy decision tree for SLR is constructed, the architecture of the fuzzy decision tree and the candidates associated with every node are produced. Fig. 5 illustrates the diagram of the fuzzy decision tree for sign language recognition, where each nonleaf node denotes a classifier associated with the corresponding word candidates, and each branch at a node represents one class of this classifier. There are common elements among adjacent branches under one node, and their intersection is learned by the large amount of training samples. The input data are first fed into one- or two-handed classifier, then into left hand shape classifier and right hand shape classifier (no left hand shape classifier for the one-handed sign branch), and at last into the SOFM/HMM classifier only with few candidates in which the final recognition results are gotten. To illustrate the idea more concretely, consider the sign "ai-hu" with a vector sequence of 48-dimensional features. The sign is firstly fed into GMM with its six-dimensional feature sequence of left hand position and orientation information for judging one- or two- handed sign, and then into FSM with its 18-dimensional feature sequence of left hand shape information for classifying into one of the branches, along this branch into FSM with its 18-dimensional feature sequence of right hand shape information for classifying, along the assigned branch into SOFM/HMM with its 48-dimensional features for decision-making, and the final classes are gotten among the candidates of this SOFM/HMM node.

V. Experimental Results

In the experiments, two Cybergloves and three Pohelmus 3SPACE-position trackers are used as input devices. Two trackers are positioned on the wrist of each hand and another is mounted at signer's back (as the reference tracker). The Cybergloves collect the variation information of hand shape with the 18-dimensional data each hand, and the position trackers collect the variation information of orientation, position, and movement trajectory.

In order to extract the invariant features to signer's position, the tracker at signer's back is chosen as the reference Cartesian coordinate system, and the position and orientation at each hand with respect to the reference system are calculated and can be taken as invariant features. In the case of two hands, a 48-dimensional vector is formed, including the hand shape, position and orientation vector. The data from different signers are calibrated by some fixed postures performed by each signer. In our experiments the 14 postures that can represent the min-max value ranges of the corresponding sensor are defined. Furthermore, the transform for each signer is defined so as to normalize the different signer data, and the transform are the reverse model transform estimated by the constrained maximum likelihood linear transformations (MLLR) [31] with 75 basic hand shape data from six signers. As each component in the vector has different dynamic range, its value is normalized to [0-1].

All experiments were carried on a large vocabulary with 5113 signs. Experimental data consist of 61 356 samples over 5113 signs from six signers with each performing signs twice. The vocabulary is taken from the CSL dictionary excluding synonymous words with the same gestures. One group data from six signers represented by A-F are referred to as the registered test set (Reg) and the other 11 group data are used as the training samples. Using the approach of cross validation test, ten group data samples from five signers are used as the training samples and the other signer data represented by A-F are referred to as the unregistered test set (Unreg).

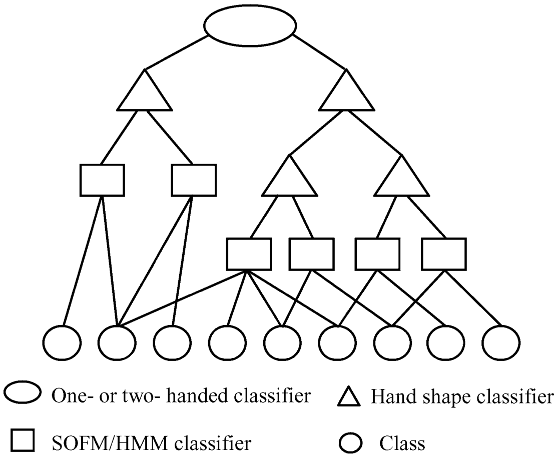


Fig. 5. Diagram of the fuzzy decision tree for sign language recognition.

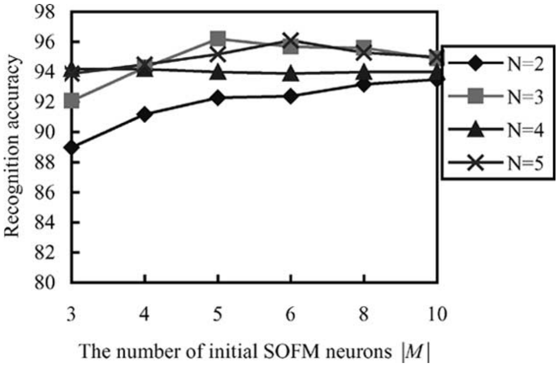


Fig. 6. Relations between the model parameter $|M|$, N and the recognition accuracy of fuzzy tree.

TABLE I
RECOGNITION RESULTS OF DIFFERENT METHODS ON A LARGE VOCABULARY

Signer		HMM	SOFM/ HMM	Fuzzy Tree
Reg	A	92.5%	95.0%	96.2%
	B	89.3%	91.7%	92.9%
	C	88.1%	90.0%	91.3%
	D	82.5%	87.9%	89.7%
	E	80.7%	85.8%	86.0%
	F	90.7%	92.3%	93.6%
	Average	87.3%	90.5%	91.6%
UnReg	A	86.9%	89.6%	91.2%
	B	80.2%	84.4%	85.3%
	C	79.0%	83.3%	83.3%
	D	76.6%	77.8%	78.5%
	E	74.8%	76.5%	78.0%
	F	82.5%	85.5%	85.9%
	Average	80.0%	82.9%	83.7%

The first experiment is to analyze the factors influencing the fuzzy decision tree accuracy. There are two factors that can directly influence the recognition accuracy of fuzzy tree. The first factor is the number of states (N) and the other is the number of initial SOFM neurons ($|M|$) in the SOFM/HMM classifier. N depends on the number of potential phonemes of the sign, where phoneme, as the basic unit of sign language, is defined as a dynamic continuous sign data of the variability of hand shape, position and orientation being very stable. $|M|$ can be considered as the center number of different features after the transform of SOFM. Its value is determined by the distribution of sign data. To get the best parameters for fuzzy tree, a series of experiments are performed, where N is set to 2–5, and $|M|$ is set to 3, 4, 5, 6, 8, 10, respectively.

As shown in Fig. 6, the best accuracy 96.2% can be obtained when $|M| = 5$ and $N = 3$. When $|M|$ grows from 3 to 5, the recognition performance is also improved. However, if $|M|$ increases from 5 to 10, the recognition rate stays similar or even slightly decreases. Thus, $|M| = 5$ is regarded as the best number of initial SOFM neuron. Though $N = 5$ and $N = 3$ have the comparative accuracy from the Fig. 6, $N = 3$ is chosen because of its less computational complexity.

The second experiment is to test the recognition performances on large vocabulary signer-independent SLR respectively with HMM, SOFM/HMM and fuzzy tree. SOFM/HMM is special case of fuzzy decision tree, that is, only SOFM/HMM classifier is used to recognize sign language. Table I shows the test results of HMM, SOFM/HMM and fuzzy tree, where HMM has three states and five mixture components and SOFM/HMM has three states and five initial SOFM neurons.

In Table I, the average recognition rates of 87.3% for HMM, 90.5% for SOFM/HMM and 91.6% for fuzzy tree are observed for the registered test set. For the unregistered test set, the average recognition rates of 80.0%, 82.9% and 83.7% are obtained, respectively. From the experiments above, we know that SOFM/HMM has better performance than HMM. The possible reasons are as follows. SOFM is trained as a feature extractor for continuous HMM in a global optimization criterion to transform signer-independent input sign representations into significant and low-dimensional representations that can be well modeled by the emission probabilities of HMM. In conventional HMM, more parameters need to be re-estimated than in SOFM/HMM, thus HMM is more inclined to converge to the local optimum in the model training. From the training procedure of HMM, we find that the variances for some sign models don't occur at the means in multimixture of HMM, i.e., the means remain unaltered, but occur at the corresponding covariances. During the recognition, the means play a more important role in scoring one sign than covariance. Clearly, this phenomenon is due to the model converging to the local optimum. Therefore, SOFM/HMM is more suitable for signer-independent SLR, and it has better performance than conventional HMM on large vocabulary signer-independent SLR.

On basis of SOFM/HMM, fuzzy tree increases the recognition accuracy by 1.1% on the registered test set and by 0.8% on the unregistered test set on the experiments. This may be due to the following reasons. First, in an integrated fuzzy decision tree framework, different features can be further researched individually, and their discriminations can be fully utilized through different feature classifiers. Second, the introduction of fuzzy classification idea through allowing the partitions with common elements can alleviate the loss of crisp classification of decision tree.

TABLE II
RECOGNITION TIME OF DIFFERENT METHODS ON A LARGE VOCABULARY

Signer		SOFM/HMM (s/w)	Fuzzy Tree (s/w)
Reg	A	2.946	0.271
	B	3.010	0.256
	C	2.926	0.260
	D	2.890	0.274
	E	2.855	0.268
	F	2.902	0.281
	Average	2.922	0.268
UnReg	A	3.017	0.266
	B	2.968	0.257
	C	2.788	0.255
	D	2.980	0.240
	E	2.741	0.269
	F	2.968	0.261
	Average	2.910	0.258

The third experiment is to test the recognition time on large vocabulary signer-independent SLR with SOFM/HMM and fuzzy tree. The approach of cross validation test is employed both in the registered test set and in the unregistered test set. All experiments are performed on the PIV1600 (512 M Memory) PC.

Table II shows the recognition time of SOFM/HMM and fuzzy tree on the vocabulary of 5113 signs, where s/w represents second per word. For the registered test set, the average recognition time of 2.922 (s/w) for SOFM/HMM and 0.268 (s/w) for fuzzy tree are observed. For the unregistered test set, the average recognition time of 2.910 (s/w), 0.258 (s/w) are obtained, respectively. The average recognition time of SOFM/HMM and fuzzy tree are respectively 2.916 second per word and 0.263 second per word in the registered and unregistered test sets. Experiments illustrate that fuzzy tree dramatically reduces the recognition time by 11 times over single SOFM/HMM. In the fuzzy tree, the feature classifiers of one- or two- handed and hand shape with little computational cost are first employed to progressively eliminate the impossible candidates, and then the complex classifier of SOFM/HMM is performed on the previous candidates. Thus, this coarse-to-fine hierarchical decision leads to the dramatic reduction of computational complexity and recognition time.

VI. CONCLUSIONS

In this paper, a fuzzy decision tree with heterogeneous classifiers is first presented for large vocabulary sign language recognition. As each sign feature has the different importance to gestures, the corresponding classifiers are proposed for the hierarchical decision to sign language attributes. GMM-based one- or two- handed classifier and FSM-based hand shape classifier are respectively proposed to progressively eliminate the impossible candidates. SOFM/HMM classifier as a special component of fuzzy decision tree is employed to get the final results at the last nonleaf nodes that only include few candidates. In the decision tree, fuzzification is introduced to alleviate the effect of crisp classification errors. Experimental results show the fuzzy decision tree has an average recognition rate of 91.6% in the registered test set and 83.7% in the unregistered test set over a 5113-sign vocabulary. The average recognition time is 0.263 second per word. Experiments also show the proposed method dramatically reduces recognition time by 11 times and also improves the recognition rate about 0.95% over single SOFM/HMM.

Though we have researched into large vocabulary signer-independent SLR, there are still some issues to be further investigated. 1) Effective feature extraction from different signers. In our method, feature extraction is implicitly incorporated into the SOFM/HMM model. If explicit effective features can be extracted through the transformation to frequency domain such as in speech recognition, the recognition may have a better performance. It is a challenging issue that deserves further study. 2) The utilization of nonmanual parameters in sign language. Non-manual parameters in sign language include gaze, facial expression, mouth movement, position, and motion of the trunk and head. Incorporating the understanding of nonmanual parameters into sign language recognition is a further direction.

ACKNOWLEDGMENT

The authors would like to thank the editors and the anonymous reviewers for their invaluable comments and suggestions.

REFERENCES

[1] M. W. Kadous, "Machine recognition of auslan signs using powergloves: Toward large-lexicon recognition of sign language," in *Proc. Workshop Integration Gesture Language Speech*, 1996, pp. 165–174.

[2] H. Matsuo, S. Igi, S. Lu, Y. Nagashima, Y. Takata, and T. Teshima, "The recognition algorithm with noncontact for Japanese sign language using morphological analysis," in *Proc. Int. Gesture Workshop*, 1997, pp. 273–284.

[3] S. S. Fels and G. E. Hinton, "Glove-talk: A neural network interface between a data-glove and a speech synthesizer," *IEEE Trans. Neural Networks*, vol. 4, pp. 2–8, Jan. 1993.

[4] J. S. Kim, W. Jang, and Z. Bien, "A dynamic gesture recognition system for the Korean sign language (KSL)," *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, pp. 354–359, Apr. 1996.

[5] M. B. Waldron and S. Kim, "Isolated ASL sign recognition system for deaf persons," *IEEE Trans. Rehab. Eng.*, vol. 3, pp. 261–271, June 1995.

[6] K. Grobel and M. Assan, "Isolated sign language recognition using hidden Markov models," in *Proc. Int. Conf. Systems, Man Cybernetics*, 1997, pp. 162–167.

[7] B. Bauer and H. Hienz, "Relevant features for video-based continuous sign language recognition," in *Proc. 4th Int. Conf. Automatic Face Gesture Recognition*, 2000, pp. 440–445.

[8] R. H. Liang and M. Ouhyoung, "A real-time continuous gesture recognition system for sign language," in *Proc. 3rd Int. Conf. Automatic Face Gesture Recognition*, 1998, pp. 558–565.

[9] T. Starner, J. Weaver, and A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 1371–1375, Dec. 1998.

[10] C. Vogler and D. Metaxas, "Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods," in *Proc. IEEE Int. Conf. Systems, Man Cybernetics*, 1997, pp. 156–161.

[11] ——, "Toward scalability in ASL recognition: Breaking down signs into phonemes," in *Proc. Int. Gesture Workshop*, 1999, pp. 400–404.

[12] ——, "A framework for recognizing the simultaneous aspects of American sign language," *Comput. Vis. Image Understanding*, vol. 81, no. 3, pp. 358–384, 2001.

[13] P. Vamplew and A. Adams, "Recognition of sign language gestures using neural networks," *Aust. J. Intell. Informat. Process. Syst.*, vol. 5, no. 2, pp. 94–102, 1998.

[14] W. Gao, J. Y. Ma, J. Q. Wu, and C. L. Wang, "Sign language recognition based on HMM/ANN/DP," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 14, no. 5, pp. 587–602, 2000.

[15] W. Gao *et al.*, "HandTalker: A multimodal dialog system using sign language and 3-D virtual human," in *Proc. 3rd Int. Conf. Multimodal Interface*, 2000, pp. 564–571.

[16] W. C. Stokoe, *Sign Language Structure: An Outline of the Visual Communication System of the American Deaf. Studies in Linguistics: Occasional Papers 8 (Revised 1978)*. Buffalo, NY: Linstok, 1960.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[18] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1997.

[19] C. S. Lee, G. Park, J. S. Kim, Z. Bien, W. Jang, and S. K. Kim, "Real-time recognition system of Korean sign language based on elementary components," in *Proc. 6th Int. Conf. Fuzzy Systems*, 1997, pp. 1463–1468.

[20] P. Hong, M. Turk, and T. S. Huang, "Gesture modeling and recognition using finite state machines," in *Proc. 4th Int. Conf. Automatic Face Gesture Recognition*, 2000, pp. 410–415.

[21] E. Trentin and M. Gori, "A survey of hybrid ANN/HMM models for automatic speech recognition," *Neurocomput.*, vol. 37, no. 1/4, pp. 91–126, 2001.

[22] A. Corradini, H. J. Böhme, and H. M. Gross, "A hybrid stochastic-connectionist approach to gesture recognition," *Int. J. Artif. Intell. Tools*, vol. 9, no. 2, pp. 177–204, 2000.

[23] X. Li, M. Parizeau, and R. Plamondon, "Hidden Markov model multiple observation training,", Tech. Rep. EPM/RT-99/16, 1999.

[24] R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–285, 1989.

[25] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1996.

[26] ——, *C4.5: Program for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.

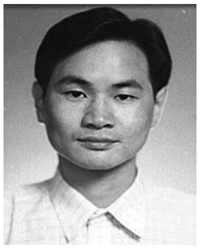
[27] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. New York: Chapman & Hall, 1984.

[28] A. Suárez and J. F. Lutsko, "Globally optimal fuzzy decision trees for classification and regression," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 1297–1311, Dec. 1999.

[29] C. Z. Janikow, "Fuzzy decision trees: Issues and methods," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 1–14, Feb. 1998.

[30] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181–214, 1994.

[31] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput. Speech Lang.*, vol. 12, no. 1, pp. 75–98, 1998.



Gaolin Fang received the M.S. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2000, where he is currently working toward the Ph.D degree.

He was a Visiting Research Assistant at Microsoft Research Asia, Beijing, in 2003. Since 2000, he has been with the Joint Research and Development Lab (JDL), Chinese Academy of Sciences, Beijing, China. His research interests include intelligent human–machine interaction, statistical language model, pattern recognition, and machine learning.

Wen Gao (M'99) received the M.S. and Ph.D degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1985 and 1988, respectively, and the Ph.D degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He was a Research Fellow at Institute of Medical Electronics Engineering, University of Tokyo, in 1992, and a Visiting Professor at Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, in 1993. From 1994 to 1995 he was a Visiting Professor at the Artificial Intelligence Lab, Massachussetts Institute of Technology (MIT), Cambridge. Currently, he is the Vice President of the University of Science and Technology of China, the Deputy President of Graduate School of Chinese Academy of Sciences, Professor of Computer Science at Harbin Institute of Technology, and the Honor Professor in computer science at the City University of Hong Kong. He has published seven books and over 200 scientific papers. He is the Editor-in-Chief of the *Chinese Journal of Computers*. His research interests are in the areas of signal processing, image and video communication, computer vision, and artificial intelligence.

Dr. Gao was the General Co-Chair of the IEEE International Conference on Multimodel Interface in 2002. He is the Head of Chinese National Delegation to the MPEG Working Group (ISO/SC29/WG11).

Debin Zhao received the B.S., M.S., and Ph.D. degrees in computer science, all from the Harbin Institute of Technology, Harbin, China, in 1985, 1988, and 1998, respectively.

He was an Associate Professor in the Department of Computer Science, Harbin Institute of Technology, and a Research Fellow in the Department of Computer Science, City University of Hong Kong, from 1989 to 1993. He is currently Professor with the Department of Computer Science, Harbin Institute of Technology. His research interests include data compression, image processing, and human–machine interface. He has authored or coauthored over 50 publications.