SIGN LANGUAGE RECOGNITION BASED ON HMM/ANN/DP

WEN GAO
Institute of Computing Technology
Chinese Academy of Sciences, 100080, China
Department of Computer Science
Harbin Institute of Technology, China

JIYONG MA
Institute of Computing Technology
Chinese Academy of Sciences, 100080, China

JIANGQIN WU
Department of Computer Science
Harbin Institute of Technology, China

CHUNLI WANG
Department of Computer Science
Dalian University of Technology, China
E-mail: {*wgao,jyma,jqwu,clwang*}*@ict.ac.cn*

In this paper, a system designed for helping the deaf to communicate with others is presented. Some useful new ideas are proposed in design and implementation. An algorithm based on geometrical analysis for the purpose of extracting invariant feature to signer position is presented. An ANN–DP combined approach is employed for segmenting subwords automatically from the data stream of sign signals. To tackle the epenthesis movement problem, a DP-based method has been used to obtain the context-dependent models. Some techniques for system implementation are also given, including fast matching, frame prediction and search algorithms. The implemented system is able to recognize continuous large vocabulary Chinese Sign Language. Experiments show that proposed techniques in this paper are efficient on either recognition speed or recognition performance.

Keywords: Hand gesture recognition; sign language recognition; artificial neural network; dynamic programming; multimodal user interface; virtual reality.

1. INTRODUCTION

Sign language, a kind of structured gesture, is one of the most natural ways of exchanging information for most of the hearing impaired. This has spawned interest in developing systems that can accept sign language as one of the input modalities for human–computer interaction, as well as supporting the communication between the deaf and hearing society. In fact, a new field of sign language engineering is emerging that attempts to make use of advanced computer technology in order to enhance the system capability to serve all people in our society through powerful and friendly human–computer interface. Our aim of recognizing sign language is

International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14, No. 5 (2000) 587–602

to provide an efficient and accurate mechanism to transcribe human sign language into text or speech.

In a system, capturing gestures is the first step of sign language recognition. There are two approaches to get the data of gestures. The first one is the vision-based approach, which utilizes cameras to capture the images of hand gestures. Hand gesture features are subsequently extracted from the images. In order to obtain the features of hand gestures robustly, a special glove with areas painted on it to indicate the positions of the fingers is often used. A glove, for example, bright points on edges of fingers or color block on regions of fingers could be used. The vision-based approach has the advantage that the signer does not have to wear any complex powered input devices. The disadvantages of this approach are its instability and impreciseness due to poor illuminant conditions and limited computing power in popular computers. Furthermore, the vision-based approach has a difficult time performing the task of large vocabulary sign language recognition, because many technical issues on image understanding are still open or need to improve. The second class of approaches is wear-device based. On the contrary, this class measures hand gestures using direct devices such as datagloves and position-trackers. The advantage of this approach is that it captures gesture data robustly and extracts features for further recognition in real-time using less computing power. The disadvantage is that the signer has to wear the device, and the device might be expensive. For our purpose of fostering the sign language recognizer, not on data capturing technology, we decided to use the dataglove to capture gesture data in our first stage of system implementation, and to consider the camera-based input in our future system implementation.

Attempts to automatically recognize sign language began to appear in literature in the 1990s. Charaphayan and Marble[3] investigated a method using image processing to understand American Sign Language (ASL). This system can correctly recognize 27 of the 31 ASL symbols. Starner[14] used a color camera and had the users wear a yellow glove on their right hand and orange one on their left. 91.3% of the results were correct. By imposing a strict grammar on this system, the accuracy rates for 40 signs in excess of 99% were possible with real-time performance. Fels and Hinton[4,5] developed a system using a VPL DataGlove Mark II with a Polhemus tracker as input devices. In their system, the neural network was employed for classifying hand gestures. Wexelblbat[21] developed a gesture recognition system using three Ascension Flock-of-Bird position trackers together with a CyberGlove on each hand as input devices. Takahashi and Kishino[15] investigated a system for understanding the Japanese Kana manual alphabets corresponding to 46 signs using a VPL DataGlove. Their system could correctly recognize 30 of the 46 signs, while the remaining 16 could not be reliably identified. Murakami and Taguchi[12] made use of recurrent neural nets for sign Language recognition, they trained the system on 42 handshapes in the Japanese finger alphabet using a VPL Data Glove. The recognition rate of their system was about 98%. Kramer and Leifer[10] worked on a method for communication between deaf individuals. Kadous[7] employed a system based on Power Gloves to recognize a set of 95 isolated Auslan signs with 80% accuracy,

with an emphasis on fast match methods. Liang and Ouhyoung[11] used HMM for continuous recognition of Taiwan Sign language with a vocabulary between 71 and 250 signs based Dataglove as input devices. However, their system required that gestures performed by the signer be slow to detect the word boundary. Tung and Kak[16] created a system of automatic learning of robot tasks through a DataGlove interface. Kang and Ikeuchi[8] designed a system for simple task learning by human demonstration. Grobel and Assan[16] used HMMs to recognize isolated signs. 91.3% accuracy was achieved using a 262-sign vocabulary. They extracted the features from video recordings of signers wearing colored gloves. Vogler and Metaxas[17] used HMMs for continuous ASL recognition with a vocabulary of 53 signs and a completely unconstrained sentence structure. Vogler and Metaxas[18,19] developed an approach to continuous, whole-sentence ASL recognition that used phonemes instead of whole signs as the basic units. They experimented with 22 words and achieved similar recognition rates with phoneme and word-based approaches.

Chinese Sign Language (CSL) consists of about 5500 conventional vocabularies including postures and gestures. With the evolution of CSL, the up-to-date CSL can express practically any meaning that can be expressed as natural spoken Chinese. Therefore, the task of CSL recognition becomes very challenging.

Compared with traditional speech recognition that only deals with one stream of speech signal data, sign language recognition deals with multiple data streams including that of two hands and ones body. Therefore, unlike the words in speech that can be represented by one-dimensional string of alphabetic elements and indexed in a lexical dictionary, signs in sign language have to be described by at least two coordinated strings of alphabetic elements. Each string can have a dimension. A multidimensional alphabetic string can be established under the lexical rules of sign language for each sign word, and the collection of strings makes the sign language lexical dictionary. However, making this type of lexical dictionary by manual coding is tedious and time consuming. Hence, automatic coding method using machine learning from training samples is recommended. In order to develop this new kind of recognizer, we need to solve the issues of (a) How to extract the invariant features to singer position. It is important in practical applications because it cannot be assumed that a singer is to be restricted at a certain position, that does not change any time when he/she is gesturing; (b) How to extract the units of subword automatically from a sign data stream. This is a challenging task particularly for the large vocabulary of sign language recognition; (c) How to tackle the phenomenon of movement epenthesis in a continuous gesturing flow. This is one of the key problems for improving the accuracy of recognition for continuous sign language; (d) How to prune the candidate data tree efficiently during the tree search of Viterbi decoding. This is a key factor for speeding up the recognition procedure and reducing the memory resources; (e) How to use language model in the decoding process. Another key problem is how to improve the recognition accuracy by pruning unlikely hypothesis as soon as possible. The above five issues will be mainly addressed in this paper.

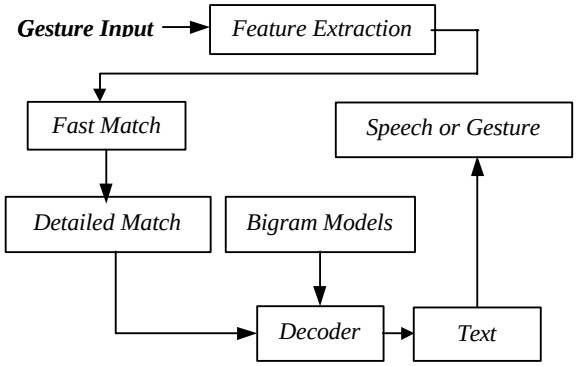

Fig. 1. Sign language recognition.

The organization of this paper is as follows: in Sec. 2 we describe the outline of the designed system. In Sec. 3 we give the extraction method of gesture features. Recognition approaches of sign language recognition are discussed in Sec. 4. We describe the performance evaluation of the proposed approaches in Sec. 5. The summary and discussion are given in the last section.

2. SYSTEM STRUCTURE

The baseline of designed system is shown in Fig. 1. The sign data collected by the gesture-input devices is fed into the feature extraction module, the output of feature vectors from the module is then input into the fast matching module. The fast matching module makes a list of word candidates. Each of the candidates is then subject to a detailed match. The language model that is currently used in our system is Bigram model which uses the transition probabilities of two neighbor words to assign, *a priori*, a probability to each pair of words based on a training context database. The decoder controls the search for the most likely priority of word appearance in a word sequence. The search algorithm we used will be described in Sec. 4. When the word sequence is output from the decoder, the sequence drives the speech synthesis module and the 3D-virtual human animation module to produce the voice of speech and the graphics of signs synchronously.

3. FEATURE EXTRACTION

For calculating all gesture data from the left hand, right hand and body parts in a well-defined space, we need to consider the modeling of the relative 3D motion of multiple receivers working with a transmitter. The timely motion of the transmitter will be also considered. 3D motion of receivers can be viewed as the rigid motion. It is well known that 3D displacement of a rigid object in the Cartesian coordinates can be modeled by an affine transformation as the following,

$$\mathbf{X}' = \mathbf{R}(\mathbf{X} - \mathbf{S}) \tag{1}$$

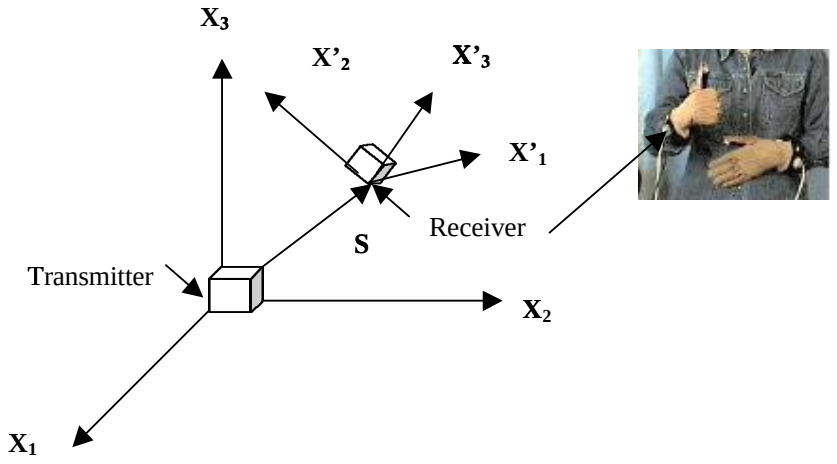

Fig. 2. The geometrical relationships between the transmitter and receivers.

where $\mathbf{R}$ is a 3×3 rotation matrix,

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\boldsymbol{\alpha} & -\sin\boldsymbol{\alpha} \\ 0 & \sin\boldsymbol{\alpha} & \cos\boldsymbol{\alpha}_2 \end{pmatrix} \begin{pmatrix} \cos\boldsymbol{\beta} & 0 & \sin\boldsymbol{\beta} \\ 0 & 1 & 0 \\ -\sin\boldsymbol{\beta} & 0 & \cos\boldsymbol{\beta} \end{pmatrix} \begin{pmatrix} \cos\boldsymbol{\gamma} & -\sin\boldsymbol{\gamma} & 0 \\ \sin\boldsymbol{\gamma} & \cos\boldsymbol{\gamma} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos\boldsymbol{\beta}\cos\boldsymbol{\gamma} & \cos\boldsymbol{\beta}\sin\boldsymbol{\gamma} & -\sin\boldsymbol{\beta} \\ \sin\boldsymbol{\alpha}\sin\boldsymbol{\beta}\cos\boldsymbol{\gamma} - \cos\boldsymbol{\alpha}\sin\boldsymbol{\gamma} & \sin\boldsymbol{\alpha}\sin\boldsymbol{\beta}\sin\boldsymbol{\gamma} + \cos\boldsymbol{\alpha}\cos\boldsymbol{\gamma} & \sin\boldsymbol{\alpha}\cos\boldsymbol{\beta} \\ \cos\boldsymbol{\alpha}\sin\boldsymbol{\beta}\cos\boldsymbol{\gamma} + \sin\boldsymbol{\alpha}\sin\boldsymbol{\gamma} & \cos\boldsymbol{\alpha}\sin\boldsymbol{\beta}\sin\boldsymbol{\gamma} - \sin\boldsymbol{\alpha}\sin\boldsymbol{\gamma} & \cos\boldsymbol{\alpha}\cos\boldsymbol{\beta} \end{pmatrix} \tag{2}$$

$\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3})^t$ and $\mathbf{X}' = \mathbf{x'_1}, \mathbf{x'_2}, \mathbf{x'_3})^t$ denote the coordinates of the transmitter and receiver respectively, $\mathbf{S}$ is the position vector of the receiver with respect to Cartesian coordinate systems of the transmitter.

The receiver outputs the data of Eulerian angles, namely, α, β, γ, the angles of rotation about X_1, X_2 and X_3 axes. Normally these data cannot be used directly as the features because inconsistent reference might exist since the position of the transmitter might be changed between the processing of training and that of testing. This inconsistent situation often occurs when the system is moved to different places. Therefore, it is necessary to define a reference point so that the features are invariant wherever the positions of transmitter and receiver are changed. The idea we propose to fix this problem is as follows. In the case of a system with two receivers, first, the reference Cartesian coordinate system of the receiver is chosen. Second, the position and Cartesian coordinate system of the receiver at right hand with respect to reference Cartesian coordinate system of the receiver at left hand are calculated as invariant features to the positions of the transmitter and the signer. The algorithm is described as follows: suppose that S_r, S_l are the position vectors of the receivers at both hands which are measured by the position tracking system. R_l is the rotation matrix of the receiver at the left hand respect to Cartesian

coordinate systems of the transmitter. R_r^t is the transpose matrix of R_r that is the rotation matrix of the receiver at the right hand respect to Cartesian coordinate systems of the transmitter. They can be calculated according to the Eulerian angles measured by the orientation tracking system. It is clear that the product $R_l R_r^t$ is invariant to the positions of the transmitter and the signer. The reason is that each element of the matrix is a dot product between a unit directional vector of axis of the receiver at the left hand and a unit directional vector of axis of the receiver at the right hand. The relative angle is invariant to the position of the transmitter. On the other hand, the relative position vector $R_l(S_r - S_l)$ is also invariant. For the case of three receivers working with a transmitter, in which the third receiver is mounted at a fixed position on the human body, such as at the top of head. The invariant features are RR_r^t, RR_l^t, $R(S_r - S)$ and $R(S_l - S)$, where R, S are the rotation matrix and the position vectors of the receiver attached to a fixed position on the human body. This approach can be generalized for the case of the number of receivers over three.

The raw gesture data, which in our system are obtained from 36 sensors on two datagloves, and two tracking receivers mounted on the datagloves, are formed as a 48-dimensional vector. The range of data value in our system is within 0–255, so that one byte is allocated to each element of the vector. A dynamic range concept is employed in our system for satisfying the requirement of using a tiny scale of data. The dynamic range of each element is different, and each element value is normalized to ensure its dynamic range 0–1.

4. SIGN LANGUAGE RECOGNITION

The most popular framework for sign language recognition is the statistical formulation in which the most probable word sequences are chosen from the sign language database to train the recognizer. Let $W = w_1, w_2, \ldots w_N$ be a sequence of words. Suppose F is the feature extracted from input gestures. The recognizer must choose a word string $\hat{W}$ that maximizes the probability given that the feature evidence of F was observed. This problem can be significantly simplified by applying the Bayesian method to find $\hat{W}$:

$$\hat{W} = \arg_W \max P(F|W)P(W)\,. \tag{3}$$

The probability, $P(F|W)$, that the feature F was observed if a word sequence W was gestured, is typically provided by the data model of hand gestures. A language model determines the likelihood $P(W)$ that denotes *a priori* chances of the word sequence W being gestured.

4.1. Spatial Data Models of Hand Gestures

Hidden Markov Models (HMMs)[13] have been used successfully in continuous speech recognition, handwriting recognition, etc. A HMM is a doubly stochastic state machine that has a Markov distribution associated with the transitions across various states, and a probability density function that models the output for every state.

A key assumption in stochastic gesture processing is that the gesture signal is stationary over a short time interval.

4.1.1. Sign coding based on ANN/DP

For each training data set of an isolated sign or a word, the training set is divided into a number of segments depending on *a prior* threshold. This threshold should be set to a value to ensure that the divided number of segments is close to the number of the phonemes that existed in the training set. Because, unlike words in speech which have a lexical dictionary to describe their pronunciations with a one-dimensional alphabetic string, signs in sign language have to be described by multiple dimensional strings. The multidimensional alphabetic strings, where each dimension describes position, orientation or hand shapes, can be established to describe the lexical rules for signs. In addition, unlike in speech, phonemes in sign language appear both sequentially and synchronously. The possible combinations are in order of 10^8 at the word level.[20] However, training such an order of HMMs is not feasible. Hence, subwords can be used as the smallest recognition units instead of words. The algorithm for finding subwords proposed in this paper is as follows. First, typical positions, orientations and postures for each hand are selected. We call them *subwords*. The training data for each subword can be collected by the input devices, because there are supervisor signals available for each training data, the Artificial Neural Networks (ANNs) can be used to classify them. In this way, each sign is composed of subwords that correspond to states in HMMs. Bourlard[2] proposed a HMM/ANN framework in speech recognition, we employed this framework in our system too.

Suppose that we have a sequence of training data for positions or orientations or hand shapes in a sign $x(t)$, $t = 1, 2, \ldots, T$, T is the number of total frames of training data. The T frames are divided into N segments with boundaries $t_0 = 0$, $t_N = T$. The average segmentation probability of each segment is defined as

$$P(t_{n-1}+1, t_n,\ \text{node}^*) = \frac{1}{t_n - t_{n-1}} \max_{\text{node}} \times \sum_{t=t_{n-1}+1}^{t_n} \text{Postprobability}\,(\text{node}|x(t)) \tag{4}$$

where Postprobability $(\text{node}|x(t))$ denotes the post probability of output **node** in BP network which can be calculated by the output value of the **node** in BP network divided by the sum of the values of all output nodes in BP network. The task is to find the segment boundaries $t_1, \ldots, t_{N-1}$ so that

$$\sum_{n=1}^{N} P(t_{n-1}+1, t_n,\ \text{node}^*) \tag{5}$$

is maximized. Dynamic programming can offer an efficient solution for solving this problem.[1] The optimal solution of the optimal subword sequence will be used to code the sign. We introduce an auxiliary function $\mathbf{F}(\mathbf{n}, \mathbf{t})$ which denotes the best

segmentation probability of the frame interval $[\mathbf{1}, \mathbf{t}]$ into n segments. By decomposing the frame interval $[\mathbf{1}, \mathbf{t}]$ into two frame intervals $[\mathbf{1}, \mathbf{j}]$ and $[\mathbf{j}+\mathbf{1}, \mathbf{t}]$ and using the optimality in the definition of $\mathbf{F}(\mathbf{n}, \mathbf{t})$, we can obtain the recurrence equation of dynamic programming:

$$F(n,t) = \max_j \{F(n-1,j) + P(j+1,t,\text{node}^*)\,. \tag{6}$$

As in the equation shown above, the best segmentation of the frame $[\mathbf{1}, \mathbf{j}]$ into $n-1$ segments is used to determine the partition of the frame interval $[\mathbf{1}, \mathbf{t}]$ into n segments. The optimal segment boundaries are calculated along with the maximum log likelihood $\mathbf{F}(\mathbf{N}, \mathbf{T})$ by using Eq. (6) recursively.

For the case of multiple training sign sequences, each training sign sequence is divided into N segments using the algorithm subject to the condition that the consistent subword sequence is generated. This can be achieved by summing up all optimal objective functions of the multiple training sign sequences and the consistent **node** is used in the same segment. In summary, the advantages of combining ANN and Dynamic Programming are,

(1) ANN/DP can provide discriminant learning among sign subwords or states that are represented by ANN output classes. This is in contrast with the minimum distance distortion criterion in VQ.
(2) Component features in a sign do not need to be assumed independent, so the correlation property among the component features is used in ANN.
(3) Because of highly parallel and regular structures in ANNs, high-performance architecture and hardware are easily implemented.
(4) ANN/DP provides a global optimal approach to code a sign.

4.1.2. Representation of state probability

For the case of two hands, all the six data streams are represented as,

$$\{\text{Position}_{\text{left}},\ \text{Orientation}_{\text{left}}\ \text{HandShape}_{\text{left}},\ \text{Position}_{\text{right}},$$
$$\text{Orientation}_{\text{right}},\ \text{HandShape}_{\text{right}}\}\,.$$

They are time synchronous. The probability density function $b_s(F)$ of state s can be defined as

$$b_s(F) = \Pi_{b=1}^{6} P_b(f_b) \tag{7}$$

where $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$, $P_b(f_i)$ denotes the post probability of the data stream i. The state S is denoted as $S = \{S_1, S_2, S_3, S_4, S_5, S_6\}$, where each s_i represents a subword in stream i which is coded by the method mentioned in the previous section.

4.1.3. Context-dependent training

In the case of continuous sign language recognition, another difficult problem is the coarticulation. Coarticulation means that both the sign in front and behind can

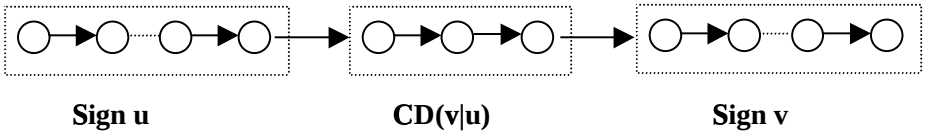

Fig. 3. Context-dependent modeling.

affect a sign. If two signs are performed in succession, an extra movement from the end position of the first sign to the start position of the second sign appears sometimes. This phenomenon is called movement epenthesis. This problem is handled in speech recognition by adding extra context-dependent HMMs to describe the effect of coarticlulation. This idea can be used for sign language recognition too.

In Fig. 3 CD($v|u$) denotes a context-dependent model to model the effect of movement epenthesis when sign u is preceeding to v.

To take into account the effect of epenthesis, sentence level training signs are necessary. Firstly, the subwords are concatenated to form a large lexical HMM for each training sentence, then the dynamic programming algorithm similar to the discussion in the previous section is used to segment the training sentence into basic subword units. For the case of multiple training sentences, each training sentence is divided into N segments by using the algorithm discussed above. Samples in the same segment are used for re-coding the sign.

4.2. Bigram Language Models

The language model provides constraints on the sequences of words to be recognized. In bigram models, we make the assumption that the probability of a word only depends on the identity of the immediately preceeding word. To make $P(w_1|w_0)$ meaningful, we asumme that the beginning of the sentence with a distinguished token$\langle\mathbf{bos}\rangle$, that is $w_0 = \langle\mathbf{bos}\rangle$. For a word string W, the probability over W is as

$$P(W) = P(w_1, w_2 \dots w_n) = \Pi_{i=1}^{n} P(w_i|w_{i-1})\,. \tag{8}$$

To estimate $P(w_i|w_{i-1})$, the frequency with which the word w_i occurs given that the last word is w_{i-1}, we can simply count how often the bigram occurs in some training corpus. If the training corpus is not large enough, many actually existing word successions will not be well enough observed which leads to many zero probabilities. So smoothing is critical to make the estimated probability robust for unseen data. In this paper, we use the Katz smoothing,[9]

$$P_{\text{Katz}}(w_i|w_{i-1}) = \begin{cases} C(w_{i-1}, w_i) & \text{if } r > k \\ d_r C(w_{i-1}, w_i)/C(w_{i-1}) & \text{if } k \geq r > 0 \\ \alpha(w_{i-1}) P(w_i) & \text{if } r = 0 \end{cases} \tag{9}$$

where $\alpha(w_{i-1})$ is the backoff weight for word w_{i-1}. The bigrams indirectly encode syntax, semantics and pragmatics by concentrating on the local dependencies between two words. The net result of the techniques is to limit the number of alternatives that must be searched for finding the most probable sequence of words.

Hence the bigram language model reduces the search space. The corpus used in our case to estimate the bigram probabilities consists of about 30 million Chinese words in the Chinese newspapers from the year 1993 to 1995.

4.3. Search Algorithms

Viterbi search and its invariant forms belong to a class of breadth-first search techniques. All hypotheses are pursued in parallel and gradually pruned away as the correct hypothesis emerges with the maximum score. In this case, the recognition system can be treated as a recursive transition network composed of the states of HMMs in which any state can be reached from any other state. In Viterbi beam searches only the hypothesis whose likelihood falls within a beam of the most likely hypothesis are considered for further growth. The best beam size is determined empirically. By expanding the network to include an explicit arc from the end of each word to the start of the next, the bigram language model has been incorporated to improve recognition accuracy. To speed up the decoding process, the following fast matching strategy is proposed in time-synchronous search.

4.3.1. Word candidates

The sign can be classified into one-handed class and two-handed class. In the case of one-handed, the left hand is motionless, the right hand conveys the information of a sign. To reduce the computation load, the right hand shape and position information is first used to prune unlikely words. For each frame t, each stream data is input to its BP network, a threshold is set to select the active subwords corresponding to the output nodes of the BP network. The word candidates are selected using a fast matching algorithm. The idea of fast matching algorithm is as follows: if the output value from the BP network of each stream exceeds a threshold, then the stream subword is active, otherwise inactive. If there are active stream subwords existing in data streams of right hand shape and position of a sign, then the sign is set active, otherwise inactive. The detailed matching algorithm is as follows. Let State(w) be the state set of sign w calculated,

$$\text{Score}(w) = \max_{S \in \text{state}(w)} b_s(F) \tag{10}$$

F is the feature vector at time $\mathbf{t}$, $b_s(F)$ is the output probability of the state s of sign w at t. The score Score(w) ranking is used as a measure of the relative activity of the active signs selected by Eq. (10). By setting a pruning threshold or beam width relative to the best scoring sign, the search at time $\mathbf{t}$ is limited to just those falling within the beam. Like other fast techniques, the proposed heuristic algorithm may cause pruning errors during search, the errors can be controlled by properly selecting the beam.

4.3.2. Pruning

In order to conserve the computing and memory resources, it is imperative to prune the low-scoring partial paths. In this paper, two different beams are used, one is

at subword level, and the other is at sign level. The beam width at each level is determined empirically, and the beam threshold with respect to the best path scoring at that level is computed.

(1) Pruning of subwords
If the cumulative score for the partial hypothesis in a subword exceeds the beam, the subword is set to inactive.
(2) Sign to sign transitions
If the cumulative score for the partial hypothesis in the last subword of a sign exceeds the beam, no transitions are computed from this subword.

4.3.3. Frame predicting

Since some components of feature vectors of gesture signals change slowly, such as the case of one-hand gesture, each component of feature vectors of the left hand is almost unchanged. Therefore, the component value of gesture features of the preceding frame can be used for predicting the gesture feature of the current frame. If the distance between the two frame component value of features is below a threshold, the observation probabilities of the current frame can be obtained from those of the preceding frame by only using a few modifications. This technique reduces the computation effort without loss of noticeable accuracy if suitable threshold is chosen.

5. EXPERIMENT

The hardware environment is Pentium III 450Hz, with two CyberGloves and two receivers of 3D tracker; each Cyberglove is with 18 sensors. The baud rate for both CyberGlove and 3D tracker is set to 38 400. For feature set, 82 handshapes, 50 major body locations and 50 hand orientations are selected for each hand. The BP algorithm is used to label each stream subword in handshapes, locations and orientations.

5.1. Isolated Sign Recognition Evaluation

For the case of isolated sign recognition, 1065 basic signs in Chinese sign language are used as evaluation vocabularies, each sign was performed eight times by a sign language teacher, seven times are used for training and one for testing. Using the approach of cross validation test, the tests time for each word is eight. The recognition accuracy is shown in Fig. 4.

The number of states in HMM of each sign is between 1–7. The number of states in HMMs context-dependent models is 3. The average is 2.76. The best recognition accuracy is about 93.2%. For different numbers of handshapes, positions and orientations of right hand, the recognition rates are given in Figs. 5–7, respectively.

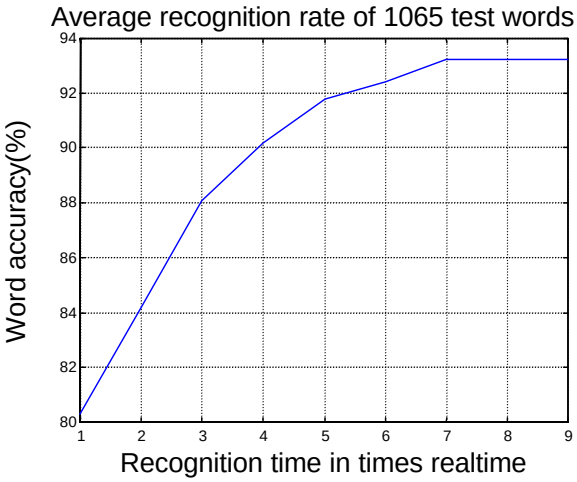

Fig. 4. Recognition rates for 1065 isolated words.

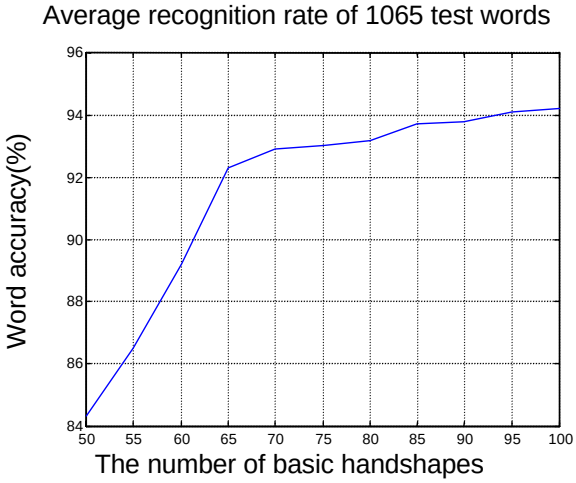

Fig. 5. Recognition rates for different number of basic handshapes.

5.2. Continuous Sign Recognition Evaluation

For the case of continuous recognition, the database of gestures consists of 220 words and 80 sentences. A sign language teacher performed gesture samples, four for training and one for test. In general, each sentence consists of 2 to 15 words. No intentional pauses were placed between signs within a sentence. For the isolated word, the recognition result is listed in Table 1. The recognition rate with feature normalization is 98.2% and the recognition rate without feature normalization is 96.3%. This result shows that feature normalization is necessary to increase system performance.

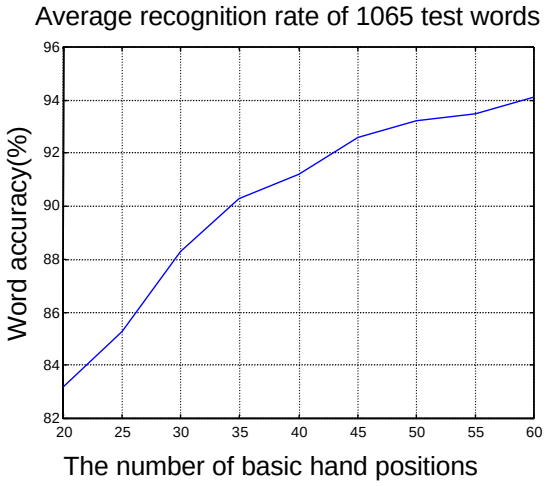

Fig. 6. Recognition rates for different number of hand positions.

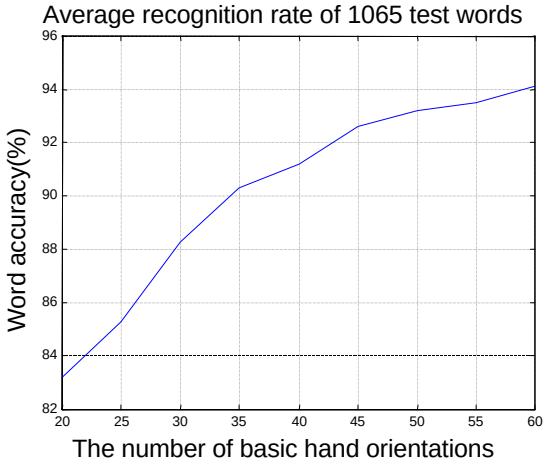

Fig. 7. Recognition rates for different number of orientations.

To get the result of the recognition performance at sentence level, one test was carried out as described in following. When 80 sentences were not used for any portion of the training, the 220 words are used as basic units. Within 80 sentences, 43 sentences can be correctly recognized, the remaining 37 sentences have deletion (D), insertion (I), and substitution (S) errors, $D = 23$, $I = 28$, $S = 55$. This shows

Table 1. The recognition rates of isolated word test.

Without Feature Normalization	73.1%
With Feature Normalization	98.2%

Table 2. The recognition rates of sentence level test.

Without Context Dependent Models	73.1%
With Context Dependent Models	95.2%

Table 3. The recognition rates of sentence level test.

With Context Dependent Models (Without Bigram Models)	90.4%
With Bigram Model (Without Context Dependent Models)	77.2%

that the movement epenthesis has greatly affected the recognition performance at sentence level. To take into account this effect, the context-dependent HMMs are trained by the sentences. For the remaining 37 sentences, the context-dependent training procedure was used for each sentence. In the collected sentence samples, four in five are used for training and one for test. The word recognition 95.2%, where $D = 5$, $S = 8$, $I = 6$, $N = 394$, N denotes the total number of signs in the test set. The accuracy measure is calculated by subtracting the number of deletion, insertion, substitution and errors from the total number of signs and divided by the total number of signs. The result shows that context-dependent models are necessary for sentence level recognition. To reach such a high recognition rate, bigram model is also necessary, otherwise the recognition performance becomes worse. Table 3 shows this comparison. The recognition rate with context-dependent models without bigram language model is 94.7%, where $D = 12$, $S = 16$, $I = 10$. The recognition rate with bigram without context-dependent models is 77.2%, $D = 19$, $I = 20$, $S = 51$. This indicates that the effect of context-dependent model is higher than that of bigram language model.

6. CONCLUSION

A CSL recognizer has been developed by using HMM/ANN/DP based technology, our contributions within this system include a new gesture feature extraction method, a fast matching algorithm, and so on. Experimental results show that the proposed techniques are capable of improving both the recognition performance and speed. The idea for extracting signer position independent feature is quite efficient and the method based on ANN combining DP for automatically extracting subwords in a sign is powerful for large vocabulary sign language recognition considering the problem of scalability.

REFERENCES

1. R. Bellman and S. Dreyfus, *Applied Dynamic Programming*, Princeton University, Princeton, NJ, 1962.
2. H. Bourlard, "State-of-the-art and recent progress in hybrid HMM/ANN speech recognition," *Proc. Int. Conf. Artificial Neural Networks*, 1997, pp. 201–208.

3. C. Charayaphan and A. Marble, "Image processing system for interpreting motion in American sign language," *J. Biomed. Engin.* **14** (1992) 419–425.
4. S. S. Fels, "Glove-TalkII: mapping hand gestures to speech using neural networks — An approach to building adaptive interfaces," Ph.D. thesis, Computer Science Department, University of Torono, 1994.
5. S. S. Fels and G. Hinton, "GloveTalk: a neural network interface between a DataDlove and a speech synthesizer," *IEEE Trans. Neural Networks*, **4** (1993) 2–8.
6. K. Grobel and M. Assan, "Isolated sign language recognition using hidden Markov models," *Proc. Int. Conf. System, Man and Cybernetics*, 1996, pp. 162–167.
7. M. W. Kadous, "Machine recognition of Auslan signed using PowerGlove: towards large-lexicon recognition of sign language," *Proc. WIGLS. The Workshop on the Integration of Gesture in Language and Speech*, 1996, pp. 165–174.
8. S. B. Kang and K. Ikeuchi, "Robust task programing by human demonstration," *Proc. Image Understanding Workshop*, 1994, pp. 303–308.
9. S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acous. Speech Sign. Proc.* **35**, 3 (1987) 400–401.
10. J. Kramer and L. J. Leifer, "A "talking glove" for nonverbal deaf individual," Technical Report CDR TR 1990 0312, Center For Design Research, Standford University, 1990.
11. R.-H. Liang and M. Ouhyoung, "A real-time continuous gesture recognition system for sign language," *Proc. Third Int. Conf. Automatic Face and Gesture Recognition*, Nara, Japan, 1998, pp. 558–565.
12. K. Murakami and H. Taguchi, "Gesture recognition using recurrent neural networks," *In CHI'91 Conference Proc.*, 1991, pp. 237–242.
13. L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, Jan. (1996) 4–16.
14. T. Starner, "Visual recognition of American sign language using hidden Markov models," Master's thesis, MIT Media Laboratory, July, 1995.
15. T. Takahashi and F. Kishino, "Gesture coding based in experiments with a hand gesture interface device," *SIGCHI Bull.* **23**, 2 (1991) 67–73.
16. C. P. Tung and A. C. Kak, "Automatic learning of assembly tasks using a dataglove system," *Proc. IEEE/RSJ Conf. Intelligent Robots and Systems*, 1995, pp. 1–8.
17. C. Vogler and D. Metaxas, "Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods," *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, Orlando, FL, 1997, pp. 156–161.
18. C. Vogler and D. Metaxas, "ASL recognition based on a coupling between HMMs and 3D motion analysis," *Proc. IEEE Int. Conf. Computer Vision*, Mumbai, India, 1998, pp. 363–369.
19. C. Vogler and D. Metaxas, "Toward scalability in ASL recognition: breaking down signs into phonemes," *Proc. Gesture Workshop*, Gif-sur-Yvette, France, 1999, pp. 400–404.
20. C. Vogler and D. Metaxa, "Parallel hidden Markov models for American sign language recognition," *Proc. Int. Conf. Computer Vision*, Kerkyra, Greece, September 1999, pp. 224–228.
21. A. D. Wexelblat, "A feature-based approach to continuous gesture analysis," Master thesis, MIT, 1993.

Wen Gao received his first Ph.D. in computer science from Harbin Institute of Technology in 1988, and received his second Ph.D. in electronics engineering from the University of Tokyo, Japan in 1991. Since December 1991, he has been a professor at the Department of Computer Science in Harbin Institute of Technology. He was a visiting research fellow in the Institute of Medical Electronic Engineering, the University of Tokyo, a visiting professor at the Institute of Robotics, Carnegie Mellon University, and also at the Artificial Intelligence Laboratory, MIT. Currently, he is a professor at the Institute of Computing Technology, Chinese Academy of Sciences, the chairman of steering committee for China national Hi-Tech programme intelligent computing system, chief editor of the Chinese *Journal of Computers*, and an honorary professor of computer science, City University of Hong Kong, guest professor at Tsinghua University, Xian Jiaotong University, Chinese University of Science and Technology, etc.

His research interests include multimodal interface, multimedia data compression, computer vision and artificial intelligence. He has published more than 150 papers and books.

Jiyong Ma received the B.S. degree in computational mathematics from Heilongjiang University, Harbin, China, in 1984 and Master degree in thermal physics in 1988. He received his Ph.D. in the Department of Computer Science, Harbin Institute of Technology, in 1999. Dr. Ma is now a postdoctoral researcher in the Institute of Computing Technology, Chinese Academy of Sciences.

His current research interests include sign language recognition, speaker recognition and multimodal perception.

Jiangqin Wu is an associate professor and is now a Ph.D. student in the Department of Computer Science and Engineering, Harbin Institute of Technology.

Her current research areas include artificial intelligence, pattern recognition, and optimization.

Chunli Wang received her B.A. in 1993 and her M.S. in 1996 both in computer science from Dalian University of Technology. She is currently a Ph.D. candidate in systems engineering at Dalian University of Technology.

Her research activities include work on pattern recognition and the technology of intelligent human–machine interface. The current emphasis of her work is on Chinese Sign Language recognition system.