Transition Movement Models for Large Vocabulary Continuous Sign Language Recognition

Wen Gao[1, 2] , Gaolin Fang[1], Debin Zhao[1], Yiqiang Chen[2]
[1]*Department of Computer Science, Harbin Institute of Technology, Harbin, 150001, China*
[2]*Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China*
{wgao, glfang, dbzhao, yqchen}@jdl.ac.cn

Abstract

The major challenges that sign language recognition (SLR) now faces are developing methods that solve large vocabulary continuous sign problems. In this paper, large vocabulary continuous SLR based on transition movement models is proposed. The proposed method employs the temporal clustering algorithm to cluster a large amount of transition movements, and then the corresponding training algorithm is also presented for automatically segmenting and training these transition movement models. The clustered models can improve the generalization of transition movement models, and are very suitable for large vocabulary continuous SLR. At last, the estimated transition movement models, together with sign models, are viewed as candidate models of the Viterbi search algorithm for recognizing continuous sign language. Experiments show that continuous SLR based on transition movement models has good performance over a large vocabulary of 5113 signs.

1. Introduction

Sign language as a kind of gestures 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 so that communication between deaf and hearing society can be more convenient. Sign language recognition, as one of the important research areas of human-computer interaction (HCI), has spawned more and more interest in HCI society.

Chinese sign language (CSL) is the language of choice for most deaf people in China. CSL consists of about 5500 conventional vocabularies including postures and gestures. With the evolution of CSL, up-to-date CSL can express any meaning in natural spoken Chinese with the aid of finger spelling. Similar to Stokoe's analysis of American sign language [1], each Chinese sign can be broken down into four parameters: hand shape, position, orientation and movement. These parameters are performed simultaneously and form multiple data streams. Compared with traditional speech recognition that only deals with one stream of speech signal data, SLR has to handle multiple data streams. Moreover, unlike phoneme in speech, in sign language, no basic unit is defined in the signs' lexical forms.

The major challenges that face SLR now are developing methods that will solve large vocabulary continuous sign language problems. The research on large vocabulary continuous SLR 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 continuous SLR, the main issue is how to handle the movement epenthesis. The movement epentheses, i.e. transition movements between two signs, begin at end of the preceding sign and finish at start of the following sign, which vary with the sign contexts. The presence of movement epenthesis greatly complicates the recognition problem, since it inserts a great variety of extra movements that are not present in the signs' lexical forms, instead of merely affecting the performance of adjacent signs.

In continuous speech recognition, context-dependent models such as biphone or triphone are generally employed for modeling the co-articulation. However, in continuous SLR, the number of phoneme extracted manually or automatically is so large that the training data becomes very sparse. This leads to the impossibility to train the context-dependent models in large vocabulary SLR. Directly modeling the movement epenthesis between signs also exists the same problem as context-dependent models.

However, transition movements are only related with the end of the preceding sign and the start of the following sign, so transition movement models in terms of signs have many identical and very similar clusters. Thus, we can cluster the transition movements so as to reduce their number and avoid the sparseness of training data. This can also improve the generalization of transition movements, which is very suitable for large vocabulary continuous SLR only with only certain sentence training samples. Nevertheless, transition movement is the temporal sequence of the vector. The k-means clustering algorithm cannot handle the temporal data because its distance measure builds between the two spatial vectors. Volger[2] employed k-means clustering with a least-squares distance criterion on the start and endpoint of the signs' lexical forms to produce the less possible combining models. However, in the large vocabulary size, it is infeasible to manually segment the continuous signs because of the huge workload and the introduction of man-made errors. Furthermore, there are some deviations between the isolated sign performance and continuous signs, so it is very difficult to model the start and endpoint of movement epenthesis with isolated signs.

In this paper, the temporal clustering algorithm with dynamic time warping (DTW) as the distance measure is proposed for clustering transition movements between two signs, and then the corresponding algorithm is presented for training those transition movement models. The training algorithm can automatically segment the transition movements with a bootstrap iteration, where the temporal algorithm is used to cluster transition movements. The estimated transition movement models, together with sign models, are used for large vocabulary continuous SLR. Experiments show that continuous SLR based on transition movement models has good performance over a large vocabulary.

The remainder of this paper is organized as follows. Section 2 gives an overview of the related work. In Section 3 we present the transition movement models. In Section 4, the temporal clustering algorithm is proposed to dynamically cluster transition movements. Section 5 gives large vocabulary SLR based on transition movement models. Section 6 shows the experimental results. The conclusions are given in the last section.

2. Related work

Attempts to automatically recognize sign language began to appear in the literature in the 90's. Following the similar path to early speech recognition, many previous attempts at sign language recognition focused on isolated sign. The recognition methods usually include rule-based matching [3], artificial neural networks [4, 5], and hidden Markov models [6]. However, because there is no clear pause between the individual signs for continuous SLR, explicit segmentation of a continuous input stream into the individual signs becomes intractable. For this reason, together with the effect of movement epenthesis, work on isolated recognition often does not generalize easily to continuous sign language recognition.

Starner et al. [7] used a view-based approach with a single camera to extract two-dimensional features as the input of HMM for continuous American SLR. 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. HMM was also employed by Hienz and Bauer [8] to recognize continuous German sign language with a single color video camera as input. An accuracy of 91.7% can be achieved in recognition of sign language sentences with 97 signs. Furthermore, they developed the K-means clustering algorithm to get the subunits for continuous SLR [9]. The accuracy of 80.8% was achieved in the corpus of 12 different signs and 10 subunits.

Liang and Ouhyoung [10] 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 HMM was employed and a Dataglove was taken as input device. Sagawa and Takeuchi [11] used the changes of hand shape, orientation, and position to detect the borders of Japanese sign language words. They experimented 10 sentences and got 83.0% accuracy with top five choices.

Vogler and Metaxas [2] used computer vision methods to extract the three-dimensional 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. They built context-dependent HMM and modeled transient movement to alleviate the effects of movement epenthesis over 64 phonemes extracted from 53 signs. Experiments showed that modeling the movement epenthesis has better performance than context-dependent HMM. The reported best accuracy is 95.83%. In addition, they used phonemes instead of whole signs as the basic units and achieved similar recognition rates to sign-based approaches over a vocabulary of 22 signs [12].

Gao et al. [13] used a dynamic programming method to obtain the context-dependent models for recognizing continuous CSL. Datagloves were used as input devices and state-tying HMM as recognition method. Their system can recognize 5177 CSL isolated signs with 94.8% accuracy in real time and recognize 200 sentences with 91.4% word accuracy.

Previous research on sign language recognition focuses primarily on the small or medium vocabulary sign language recognition. There has been very little work reported on large vocabulary sign language recognition.

3. Transition movement models

For continuous sign language recognition, the main issue is how to handle the movement epenthesis (i.e. transition movements between two signs). In fact, modeling the movement epenthesis has been first proposed by Vogler to reduce its effect. However, all the possible combinations between two signs are so mass, especially in the large vocabulary size, that a large amount of continuous sentences are required for training those models. Furthermore, there are no lexical definitions in the sign dictionary for the movement epenthesis, so it is difficult to model those movement epentheses. However, movement epentheses are usually related with the end of the preceding sign and the start of the following sign. In Figure 1, T(V|U) represents the transition movement model from Sign U to Sign V. Different transition movements between two signs have the identical and very similar end-start sequences. Thus, we can reasonably cluster those transition movements into one class. This will reduce not only the transition movement number to avoid the sparseness of training data and also improve the generalization of transition movements. This is very suitable for large vocabulary SLR.

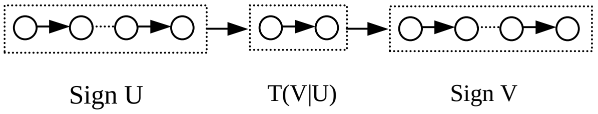

Figure 1. The transition movement model between two signs

For the continuous sign language, the start and end of the corresponding signs cannot be known, so it is infeasible to segment the transition movements with manual annotation. In this paper, the training algorithm of transition movement models is proposed to automatically extract the movement epentheses from continuous sentences and simultaneously estimate their model parameters. In the continuous sentences, sign model parameters are initialized by the isolated sign models, and transition movement parameters are trained by the iterative segmented transition data. During the iterative process of estimating those parameters, sign models and transition models are combined into the whole models for the description of the sentences.

The training algorithm of transition movement models is described as follows.

1) With the isolated HMM models, continuous sentences are segmented into the corresponding isolated sign sequence using automatic segmentation.
2) Set the transition movements from the last state of the preceding sign to the first state of the following sign as the initial values of transition movements.
3) Cluster the transition movements through the temporal clustering algorithm, and train transition movement models with the data after being clustered, and train sign models with the segmented data and isolated sign data.
4) Using the new models (transition movement models and sign models) to segment continuous sentences into signs and the corresponding movement epentheses, and judge whether the transition frame number has changed compared with the last segmentation, if it has changed and then return to Step 3, otherwise save the trained models and exit.

4. Temporal clustering

Since the transition movement is the time sequence of the vector, the clustering algorithm is required to handle not only the spatial vector but also the temporal sequence information. Furthermore, there is no criterion to describe how many clusters are very rational, so we must dynamically cluster the vector sequence according to the data distribution.

The k-means clustering algorithm cann’t handle the temporal data because its distance measure only builds between the two spatial vectors. Wilpon[14] proposed modified k-means algorithm(MKM) for producing the robust matching templates for speaker-independent speech recognition. However, MKM cannot dynamically cluster the data. In this paper, temporal clustering algorithm based on MKM is proposed to cluster the temporal sequence of the vectors. DTW is employed as the distance computation criterion because it can measure the distance between two temporal sequences by aligning different time signals and normalizing them to a warping function. In the algorithm, the corresponding skills are proposed to solve the issues of cluster splitting and combination. The proposed algorithm can automatically split and combine the centroids according to the data distribution to get the more reasonable cluster number and centers. The following subsection will discuss DTW-based distance computation and temporal clustering algorithm in detail.

4.1. DTW-based distances computation

Dynamic time warping (DTW) is to search the best warping function using the dynamic programming technique so as to minimize the distance between the two temporal sequences. Let two temporal sequences $X = (X_1, X_2, ..., X_{T_X})$, $Y = (Y_1, Y_2, ..., Y_{T_Y})$, where X_i and Y_i are the 48-dimensional vectors. Define the warping function $\phi = \{\phi(1), \phi(2), ..., \phi(N)\}$, where N is the “normal” duration of the two sequences on the normal time scale, and $\phi(n) = (\phi_X(n), \phi_Y(n))$, $\phi_X(n) \in \{1, \cdots, T_X\}$, $\phi_Y(n) \in \{1, \cdots, T_Y\}$. The n-th matching pair $\phi(n)$ consists of the $\phi_X(n)$ vector in X and the $\phi_Y(n)$ vector in Y .

The measure $d(\phi_X(n), \phi_Y(n))$ is defined as the Euclidian distance. The goal of DTW is to search the minimal accumulating distance and the associated warping path, that is:

$$D(X,Y) = \min_{\phi} \sum_{n=1}^{N} d(\phi_X(n), \phi_Y(n)) \qquad (1)$$

The warping functions used in our experiment satisfy endpoint constraints, monotony constraints and one-step local continuity constraints. Unlike in speech recognition, we do not put any region constraint to the DTW search so as to get the best path among the possible candidates.

The minimum partial accumulated distortion along a path from $(1,1)$ to (i_X, i_Y) is defined as:

$$D(i_X, i_Y) = \min_{\phi, T'} \sum_{n=1}^{T'} d(\phi_X(n), \phi_Y(n)) \ , \qquad (2)$$

where $\phi_X(T') = i_X$ and $\phi_Y(T') = i_Y$.

The auxiliary parameter $\psi(i_X, i_Y)$ is defined to record a point before the point (i_X, i_Y) in the local optimal path. The recursive relations according to the constraints are given as follows:

$$D(i_X, i_Y) = \min_{(i'_X, i'_Y)} [D(i'_X, i'_Y) + d(i_X, i_Y)] \qquad (3)$$

$$\psi(i_X, i_Y) = \arg\min_{(i'_X, i'_Y)} [D(i'_X, i'_Y) + d(i_X, i_Y)] \qquad (4)$$

where $(i'_X, i'_Y) \in \{(i_X - 1, i_Y), (i_X - 1, i_Y - 1), (i_X, i_Y - 1)\}$.

Through the dynamic programming search, the minimal distance $D(X,Y)$ between the two temporal sequences and the associated warping function pair ϕ are gotten at last.

4.2. Temporal clustering algorithm

Let $\Pi = \{O_1, O_2, ..., O_V\}$ be a data set for V temporal sequences to be clustered. Temporal clustering algorithm is to dynamically cluster the c centers, and get $\Pi = \bigcup_{j=1}^{c} \Gamma_j$.

The temporal clustering algorithm is described as follows:

1. Initialization:

Calculate all distances $d(O_i, O_j)$ using DTW. Set the initial parameters: c - the number of clusters, C - the expected number of clusters, θ_N - the minimum number of samples in each cluster, θ_C - the threshold of the intercluster distance that determines whether to combine or not, t - the number of iteration, and $t_{\max}$ - the maximum iterations.

2. Initialize the cluster centers:

The method described in [16] is employed to set the initial cluster centers. It splits the clusters from one to the expected number C step by step.

3. Classification:

According to the minimum DTW distance rule, each sample is classified to the corresponding center.

For each cluster, if its sample number is less than θ_N , then this cluster is discarded, and set $c = c - 1$, and re-classify the samples in this cluster.

4. Recalculate the cluster center:

The recalculation is described by the following two steps:

First, find the pseudo-average center O' . A particular element in the cluster has the largest population of elements (subset of the cluster) whose distance to the particular sample falls within a threshold. If several patterns have the same largest count of samples with distances below the threshold, then the element that has the smallest average distance to all samples in the subcluster is chosen as the pseudo-average center.

Second, all samples in Γ_j are warped to the pseudo-average center O' . We then group the samples according to their individual warping paths with respect to O' . The vectors that are aligned to the same index i are then averaged to produce an average vector for the new cluster. The resultant sequence with vectors indexed from 1 to $T_{O'}$ (duration for O') is the average cluster center $m(\Gamma_j)$.

5. If $\mathrm{t} \bmod 2 = 0$ or $c \ge 2C$, then goto step 7, else goto step 6.

6. Cluster splitting:

Calculate intracluster distance λ_j for each cluster j :

$$\lambda_j = \frac{1}{\|\Gamma_j\|} \sum_{O \in \Gamma_j} d(m(\Gamma_j), O) \ , \ \ j = 1,2,..,c \qquad (5)$$

Find the cluster $\Gamma_{j\max}$ with the maximum intracluster distance, if $\|\Gamma_{j\max}\| \ge 2\theta_N$ or $c \le C/2$, then split $\Gamma_{j\max}$ as follows. Find two temporal sequences O_{p1} and O_{p2} satisfying $d(O_{p1}, O_{p2}) \ge d(O_{p3}, O_{p4})$ for any other pair O_{p3} , O_{p4} in $\Gamma_{j\max}$. Two sequences O_{p1} and O_{p2} are used as the new cluster centers to replace original cluster, and set $c = c + 1$, then goto step 8.

7. Cluster combination:

For all the cluster centers, calculate the intercluster distances $d(m(\Gamma_i), m(\Gamma_j))$ between all the pairs. Find the pair with the minimum interclass distance $d(m(\Gamma_p), m(\Gamma_q))$, if $d(m(\Gamma_p), m(\Gamma_q)) < \theta_C$, then combine Γ_p and Γ_q . Using DTW the optimal path between the sequences Γ_p and Γ_q is gotten. Let T be the warping path length for ϕ , and the new cluster $\overline{m}$ is calculated as follows:

$$\overline{m}_k = \frac{1}{2}(m(\Gamma_p)_{\phi_X(k)} + m(\Gamma_q)_{\phi_Y(k)}) \ , \ k = 1,2,...,T \qquad (6)$$

Replace these two clusters with new cluster $\overline{m}$, set $c = c - 1$.

8. $t = t + 1$, if $t < t_{\max}$, then return to step 3, otherwise, save the clusters data and exit.

5. Large vocabulary SLR based on transition movement models

In this section, we will discuss how to use the trained transition movement models for large vocabulary sign language recognition. Transition movement models and sign models are combined into the candidate models of the Viterbi search algorithm for large vocabulary SLR. However, because the candidate models are so huge that the pruning operation has to be employed to make the system performance in real time.

Each sign has its own trajectory in sign space, if an observation vector is close to the trajectory, then the sign may be active at that time, else the sign will be inactive. For each observation vector, how to judge if a sign is active is very important to speed up the recognition process. If only a small fraction of signs is active at a frame, the most likely active signs are those which are active at previous frame due to the continuous property of gestures. Only these active signs need to be further searched at the next frame, thus a large mount of computation load can be saved.

According to the analyses above, the rules of adding new words and eliminating words are obtained for the candidate selection in the search process. The details are as follows:

Adding new words: Calculate the first state probability of all the words excluding the candidates at the last frame, if the word is greater than certain threshold, then enter the candidates of current frame, and the other state probabilities of this word needn't to be further calculated at this frame.

Eliminating words: For all the candidates of the last frame, if all the state path scores of a word are less than certain threshold, then this word is eliminated from the current candidates, and its path will not be further searched.

6. Experiments

In our 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 fixed 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. By this transformation, the data are composed of a relative three-dimensional position vector and a three-dimensional orientation vector for each hand. In the case of two hands, a 48-dimensional vector is formed, including the hand shape, position and orientation vector. As each component in the vector has different dynamic range, its value is normalized to [0,1].

Experimental data consist of 25565 sign samples over 5113 isolated signs with each sign having five samples. The vocabulary is taken from the Chinese sign language dictionary. Four samples are used as the training set and the rest one samples are used as the isolated sign test set. Continuous sign language database consist of the 1500 sentence samples with 750 different sentences over a vocabulary of 5113 signs. The sentences are extracted from the 200M corpus which is composed of China Daily and Family Collection Book.

The first experiment validates that the proposed temporal clustering algorithm can cluster similar sequences into one class. Database consists of 1268 samples from 317 signs which are random selected among 5113 signs, each having four samples. Because the corresponding classes are known beforehand, the clustering validation can be judged. The expected cluster center is set to 317. After the processing of temporal clustering algorithm, the 309 cluster centers can be obtained. The 301 centers are the same as the sign data, and each has four samples. The rest 8 centers are the sample combination of two signs.

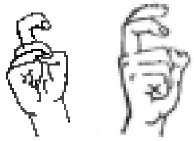

Figure 2. The description of the words "J" and "ninety", left for "J", and right for "ninety"

In the 8 centers, they can be classified into three categories. One is that the two signs have the same action, such as zhu-ren(director) and zhu-chi(preside). The second is that two signs have the same postures, but only small differences in position, such as zhong-zu(race or tribe) and zhong-lei(category). The third is that two signs have very similar postures, where one has slight movement and the other hasn't. For example J and jiu-shi(ninety) in which the sign J is static, and ninety has a slight movement of first finger. Figure 2 shows the description of J and ninety.

From the experiments above we can know the temporal clustering algorithm can effectively cluster the segments with high similarity into the same cluster.

The second experiment is to analyze the factors influencing the isolated sign accuracy. There are two factors that can directly influence the recognition accuracy of HMM. The first factor is the number of states (N) and the other is the number of mixture component (M). N depends on the number of potential phonemes of the sign, where phoneme is defined as a dynamic continuous sign data of the variability of hand shape, position and orientation being very stable. The value of M is determined by the distribution of sign data. To get the best parameters for HMM, experiments are performed, where N is set to 2, 3, 4, 5, and M is set to 1, 2,3, 4, 5, 6, respectively.

As shown in Figure 3, the best accuracy 95.4% for 5113 isolated signs can be gotten when M=3 and N=3. When M grows from 1 to 3, the recognition performance is also improved. However, if M increases from 3 to 6, the recognition rate stays similar or even slightly decreases. Thus, M=3 is regarded as the best number of mixture component. Though N=5 and N=3 have the comparative accuracy from the Figure 5, N=3 is chosen because of its less computational complexity.

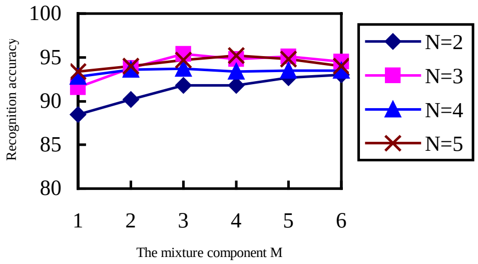

Figure 3. The relations between the model parameter M, N and the isolated sign recognition accuracy

The third experiment is to test the performance of transition movement models for continuous sign language recognition. Among 1500 sentence samples, 750 sentences are used as training and another 750 samples as the test set. Those sentences consist of the words from 3 to 15 with the average 6.6 words each sentence.

Table 1. The accuracy for continuous SLR

Method	Accuracy	Time
Transition movements models	90.8% (S=279, I=53, D=127)	1.29 second/word

All experiments are performed with the bigram language model on the PIV1600 (512M Memory) PC. S, I and D denote the error numbers of substitution, insertion and deletion, respectively. The whole number of signs in the test set is 4994 and the number of transition movements without clustering is 3945. The candidates for recognition consist of 546 clustered transition movements and 5113 signs, where their models have 3 states and 3 mixture components. Table 1 shows that the recognition rate of 90.8% for transition models is gotten on the test set. Experiments also show that transition models can also be performed in real time without clear delay.

7. Conclusions

In this paper, continuous sign language recognition over a large vocabulary with 5113 signs is first implemented based on transition movement models. Aiming at the a large amount of transition movements between two signs, we present the temporal clustering algorithm with dynamic time warping as the distance measure and the corresponding algorithm for automatically segmenting and training those transition movement models. The clustered models can improve the generalization of transition movement models, and very suitable for large vocabulary continuous SLR with certain training samples of typical sentence. Experimental results show that continuous SLR has a recognition rate of 90.8% on 1500 sentence samples over a large vocabulary of 5113 signs. Furthermore, the temporal clustering algorithm can be further extended to extract the basic units from Chinese sign language and automatically seek the anonymous gestures.

8. Acknowledgment

This work was supported in part by Natural Science Foundation of China (Grant No.60303018) and National High-Technology Development '863' Program of China (Grant No.2001AA114160).

9. References

[1] 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), Linstok Press, University of Buffalo, 1960.

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

[3] M.W. Kadous, Machine recognition of Auslan signs using PowerGloves: towards large-lexicon recognition of sign language, Proc. Workshop on the Integration of Gesture in Language and Speech, pp. 165-174, 1996.

[4] 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, no. 1, pp. 2-8, 1993.

[5] J.S. Kim, W. Jang, and Z. Bien, A dynamic gesture recognition system for the Korean sign language (KSL), IEEE Trans. Systems, Man, and Cybernetics, vol. 26, no. 2, pp. 354-359, 1996.

[6] K. Grobel and M. Assan, Isolated sign language recognition using hidden Markov models, Proc. Int'l Conf. System, Man and Cybernetics, pp. 162-167, 1997.

[7] T. Starner, J. Weaver, and A. Pentland, Real-time American sign language recognition using desk and wearable computer based video, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 12, pp. 1371-1375, 1998.

[8] B. Bauer and H. Hienz, Relevant features for video-based continuous sign language recognition, Proc. Fourth Int'l Conf. Automatic Face and Gesture Recognition, pp. 440-445, 2000.

[9] B. Bauer and K.F. Kraiss, Towards an automatic sign language recognition system using subunits, Proc. Int'l Gesture Workshop, pp. 64-75, 2001.

[10] R.H. Liang and M. Ouhyoung, A real-time continuous gesture recognition system for sign language, Proc. Third Int'l Conf. Automatic Face and Gesture Recognition, pp. 558-565, 1998.

[11] H. Sagawa and M. Takeuchi, A method for recognizing a sequence of sign language words represented in a Japanese sign language sentence," Proc. Fourth Int'l Conf. Automatic Face and Gesture Recognition, pp. 434-439, 2000.

[12] C. Vogler and D. Metaxas, A framework for recognizing the simultaneous aspects of American sign language, Computer Vision and Image Understanding, vol. 81, no. 3, pp. 358-384, 2001.

[13] W. Gao, J.Y. Ma, J.Q. Wu, and C.L. Wang, Sign language recognition based on HMM/ANN/DP, Int'l J. Pattern Recognition and Artificial Intelligence, vol. 14, no. 5, pp. 587-602, 2000.

[14] J.G. Wilpon, and L.R. Rabiner, A modified K-means clustering algorithm for use in isolated word recognition, IEEE Trans. ASSP, 33: 587-594, 1985.