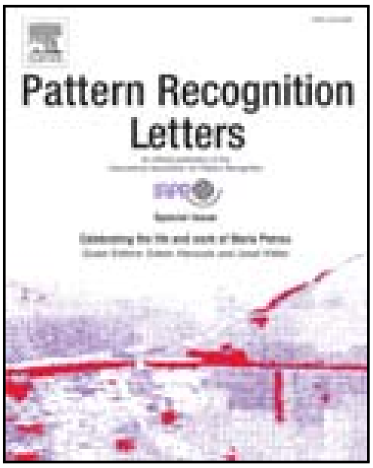


Continuous sign language recognition using level building based on fast hidden Markov model

Wenwen Yang , Jinxu Tao , Zhongfu Ye

PII: S0167-8655(16)30034-4
DOI: 10.1016/j.patrec.2016.03.030
Reference: PATREC 6496

To appear in: *Pattern Recognition Letters*

Received date: 13 August 2015
Accepted date: 31 March 2016

Please cite this article as: Wenwen Yang , Jinxu Tao , Zhongfu Ye , Continuous sign language recognition using level building based on fast hidden Markov model, *Pattern Recognition Letters* (2016), doi: 10.1016/j.patrec.2016.03.030

Highlight：

- HMM-based Level Building algorithm outperforms other methods.
- Performance rises when employing Grammar and sign length constraints.
- System runs faster by using a fast algorithm for HMM.

Continuous sign language recognition using level building based on fast hidden markov model

Wenwen Yang , Jinxu Tao ∗,Zhongfu Ye

National Engineering Laboratory for Speech and Language Information and Processing,
Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China, 230027

ARTICLE INFO

Keywords:
Continuous sign language recognition
Level Building
Fast hidden Markov model
Grammar constraint
Sign length constraint

ABSTRACT

Sign sequence segmentation and sign recognition are two main problems in continuous sign language recognition (CSLR) system. In recent years, dynamic time warping based Level Building (LB-DTW) algorithm has successfully dealt with both two challenges simultaneously. However, there still exists two crucial problems in LB-DTW: low recognition performance due to bad similarity function and offline due to high computation. In this paper, we use hidden Markov model (HMM) to calculate the similarity between the sign model and testing sequence, and a fast algorithm for computing the likelihood of HMM is proposed to reduce the computation complexity. Furthermore, grammar constraint and sign length constraint are employed to improve the recognition rate and a coarse segmentation method is proposed to provide the maximal level number. In experiments with a KINECT dataset of Chinese sign language containing 100 sentences composed of 5 signs each, the proposed method shows superior recognition performance and lower computation compared to other existing techniques.

∗ Corresponding author. Tel.: +86-551-63601329; fax: +0-000-000-0000; e-mail: tjingx@ustc.edu.cn

1. Introduction

In recent years, there has been increasing interest in developing automatic sign language recognition (SLR) systems to enhance communication between normal hearing and deaf people. Generally, this kind of systems most focus on the manual aspect of signs and recognize hand configurations including hand shape, position, orientation and movement. The systematic change of these hand configurations produces massive different signs, which are well-defined in Chinese sign language dictionaries. Usually, there are three levels of sign language recognition: finger spelling (alphabets), isolated words, and continuous sign language (sentences), while main researches focus on the latter two. In the isolate sign language recognition (ISLR), a hand gesture is a sequence with fixed starting/ending points as the sign boundary, while there's no such explicit sign boundary in CSLR. In the CSLR, a sentence sequence consists of several true-sign sequences and non-sign sequences, also called movement epenthesis (ME), which connect the end location of the previous sign to the start location of the next sign. The task of ISR is to label sign with right sign label, while the main tasks of the CSLR are: 1) splitting sentence sequence into true-sign sequences and movement epenthesis sequences; 2) labeling each true-sign sequence with right sign label. To handle these crucial problems of ISLR and CSLR, many statistical methods or machine learning methods are proposed and employed.

Towards the ISLR, HMM (Yang and Sarkar, 2006; Han et al., 2009; Pitsikalis et al., 2011), conditional random field (CRF) (Yang and Lee, 2010) and dynamic time warping (DTW) (Hernández-Vela, 2014) are mainstream methods. Otherwise, convolutional neural network (CNN) (Pigou et al., 2014) and deep neural network (DNN) (Wu and Shao, 2014) have been applied in SLR. Starner and Pentland (1998) utilize HMM training sign model with feature vector consisting of each hand's x and y position, eccentricity of the bounding ellipse and angle of axis of least inertia in American sign language (ASL) recognition system. For large-scale ASL applications, Vogler and Metaxas (1999) first break down sign sequences into phonemes with Movement-Hold model, then utilize parallel HMM to model movement and hand shape features for each phonemes respectively, and combine the probability of each channel as final output in recognition stage. Product-HMM, a variant of multistream HMM, is employed for fusing movement and shape information in the Greek Sign Language and achieves higher performance than parallel HMM (Theodorakis et al., 2009). Inspired by Movement-Hold model, Theodorakis et al., 2014 propose a phonetic modeling framework for sign language recognition based on dynamic-static (D-S) subunits. Firstly, signs are segmented into dynamic or static segments, which are clustered to construct D-S subunits. Then parallel HMM are utilized to model these D-S subunits. Further experiments on Boston University ASL, Greek SL Lemmas and ASL Large Vocabulary Dictionary show the effectiveness of their method. Sminchisescu et al., (2005) use CRF to recognize human motion, which outperforms HMM. For incorporating hidden structures of gesture sequences, Wang et al. (2006) propose hidden state conditional random field (HCRF), a discriminative hidden-state approach for the recognition of gestures, which outperforms CRF. Lichtenauer et al. (2008) utilize statistical DTW only for time warping, and a combined statistical classifier is employed to model signs. Pigou et al. (2014) utilize two CNNs to extract features, one for hand features and one for upper body features. And artificial neural network (ANN) is employed as a classifier of CNN. Wu and Shao (2014) utilize a deep dynamic neural networks (DDNN) for gesture segmentation and recognition in the ChaLearn Looking at People 2014 challenge.

Towards the CSLR, mainstream methods are based on HMM, CRF and DTW. Lee and Kim (1999) propose a threshold-model with HMM for ME which calculates the adaptive likelihood threshold of an input pattern. The result show that the proposed method can successfully extract trained gestures from continuous hand motion with a 93.14% reliability, when testing sentence consists of ten hand gestures. Fang and Gao (2002) propose a simple recurrent networks/HMMs (SRNs/HMMs) for signer-independent CSLR, where SRN is used as soft segmentation of continuous sign language. The system obtains a 85% accuracy in recognizing 100 sentences from seven signers on a vocabulary of 208 signs. Fang et al. (2007) utilize a transition movement model (TMM) to handle ME in large-vocabulary CSLR. Testing on 1500 sentences composed of 5113 Chinese signs yields an average accuracy of 91.9%. However, the data is acquired by the data Glove. Kelly et al. (2009) also propose a parallel HMM threshold model to handle ME based on the threshold HMM (T-HMM). Yang et al. (2010) adopt an enhanced Level Building algorithm to simultaneously segment and match signs to the testing continuous sentence. With the trigram grammar constraint, the system obtains 83% recognition rate in sentence level. Based on Microsoft depth camera KINECT, Zafrulla et al. (2011) construct an American sign language recognition system for deaf children education games, where HMM is employed to depict each sign. Kong and Ranganath (2014) propose a segment-based probabilistic approach to robustly recognize continuous sign language sentence. Firstly, the sentences are segmented into sign or ME sub-segments by utilizing Bayesian network fusing the outputs of CRF and support vector machine (SVM). Then a sign sub-segments are merged and recognized by a two-layer CRF classifier. Making tests on the data from 8 signers, the system obtains a recall rate of 95.7% and a precision of 96.6% for unseen samples from seen signers, and a recall rate of 86.6% and a precision of 89.9% for unseen signers. This approach achieves 0.8162 score in the gesture spotting challenge. Koller et al. (2015) aim at building a real-life continuous sign language recognition system. HMM-based visual models are employed to complete the recognition task together with the class language model and the constrained maximum likelihood linear regression (CMLLR) is utilized to deal with signer-dependency.

In this paper, we only consider the manual part of Chinese sign language signs in our work and work on the problems of the Yang's method: enhanced Level Building (Yang et al., 2010). In Yang's work, DTW is utilized to calculate the distance between the sign model and the candidate sign sequence at each level, and then search a global optimal matching distance, accordingly yielding the segments and recognition result of the sentence through the backtracking path. So, the distance function is crucial in the whole process. As known, DTW is not the better model in isolated words recognition compared to HMM, which results in low sentence recognition rate. What's worse, the system runs very slowly due to high computation caused by massive search times at each level and calculating DTW during each search. In order to overcome these two problems, HMM is employed to calculate the similarity between sign model and test sign sequence in our paper, since HMM characterizes the sign better than DTW. Then the sign length constraint and grammar constraint are embedded into Level Building recognition process to enhance the recognition performance. Furthermore, to reduce the computation of HMM-based Level Building (LB-HMM), we propose a fast algorithm (called Fast-HMM) to calculate the likelihood of the HMM approximately. In Fast-HMM, given a test sentence sequence with M frames, we firstly calculate the optimal decoding probabilities of M sequences via Viterbi algorithm (Rabiner, 1989) where the i th decoded sequence is from 1th frame to ith frame. Thus, we can get M decoded probabilities for each sign model and this process executes only one time before Level Building. In Level Building, through several basic mathematic operation using M probability values, we can obtain the similarity between the sign model and the arbitrary candidate sign sequence.

The main contributions of this paper are concluded as follows:

1. HMM is embedded into the Level Building algorithm, which will improve the recognition rate at sentence level.

2. Grammar and sign length constraints are employed to reduce substitution, insertion and deletion errors.

3. A fast calculation algorithm is proposed to approximately compute the likelihood of HMM, which reduces the computation of LB-HMM.

4. Coarse segmentation method is employed to obtain the number of levels adaptively for each sentence, not the fixed.

In the following parts of the paper, Section 2 describes the Level Building algorithm based on HMM. We will present fast algorithm for calculating the likelihood of HMM and the coarse segmentation method to decide the number of levels in Section 3. Section 4 presents the experiment results of our method compared with other methods and our conclusion is in Section 5.

2. The Level Building Algorithm Based on HMM

We declare our notations referring to Yang et al. (2010):

1) $\boldsymbol{\lambda}=\left(\lambda_1,\cdots\lambda_{N_\lambda-1},\lambda_{N_\lambda}\right)$: the sign model set where true sign models are from λ_1 to $\lambda_{N_\lambda-1}$ and λ_{N_λ} is a non-sign model, with N_λ as the number of sign model.

2) T : a M frame sequence composed of several signs.

3) $\boldsymbol{e}_L=\left(e_0,e_1,\cdots,e_l,\cdots,e_L\right)$: a sign boundaries sequence of a query sentence, where e_l is a frame number on which lth sign ends and e_0 represents 0 th frame.

4) $\boldsymbol{S}_L=\left(S_1,\cdots,S_l,\cdots,S_L\right)$: a sign label sequence where S_l represents one sign model in $\boldsymbol{\lambda}$.

5) $L_{\max}$: the maximal number of signs in a test sentence, also considered as the level number in Level Building algorithm.

6) $T\left(i:j\right)$: a subsequence of T from frame i to frame j , which is considered as a candidate sign segment in the searching process.

7) $prob\left(\lambda_i,T\left(j:m\right)\right)$: the probability or likelihood of the subsequence $T\left(j:m\right)$ generate by sign model λ_i .

8) $ll\left(\lambda_i,T\left(j:m\right)\right)$: log of $prob\left(\lambda_i,T\left(j:m\right)\right)$.

9) $P\left(\boldsymbol{S}_L,T\right)$, $\tilde{P}\left(\boldsymbol{S}_L,T\right)$: the probability and log probability of the sentence T labeled as $\boldsymbol{S}_L$.

2.1. The Level Building Algorithm based on HMM

In this paper, HMM is employed to train the sign model. $\lambda_i=\left(\boldsymbol{\pi},\mathbf{A},\mathbf{B}\right)$ is often employed to indicate the probabilistic parameter of HMM. Here, $\boldsymbol{\pi}$ denotes the vector of the initial probability π_i that hidden state i as starting state. $\mathbf{A}$ stands for the matrix of state transition probabilities a_{ij} that a transition from state i to j. And $\mathbf{B}$ represents the matrix of the observation probability $b_j\left(\mathbf{O}_t\right)$ that observation $\mathbf{O}_t$ emitted at time t in state j.

For a test sequence, we often use likelihood to measure the similarity between the test sequence and sign model, where the likelihood measures the probability that the test sequence is generated by a sign model. In the CSLR, given a test sentence sequence T , firstly the whole sentence is broken into several segments, where each segment represents a candidate sign in sentence. Then, we label T with label sequence $\boldsymbol{S}_L$ via HMM, with each label in $\boldsymbol{S}_L$ considered as a sign label of each segment accordingly. Then, the sentence likelihood can be obtained through multiplying the likelihood between each label and each segment, where the sentence likelihood measures the probability of the sentence T generated by the label sequence $\boldsymbol{S}_L$. However, there are many strategies to break down T into various segment sets, with one sentence likelihood between T and $\boldsymbol{S}_L$ for each split strategy. So the recognition task is to find the optimal label sequence $\boldsymbol{S}_L^*$, whose sentence likelihood is maximal among all cases. Referred to Yang et al. (2010), the objective function can be written as:

$$\begin{aligned} D^* &= \arg\max_{L,\boldsymbol{e}_L,\boldsymbol{S}_L} P\left(\boldsymbol{S}_L,T\right) \\ &= \arg\max_{L,\boldsymbol{e}_L}\max_{\boldsymbol{S}_L}\prod_{l=1}^{L} prob\left(S_l,T\left(e_{l-1}+1:e_l\right)\right) \end{aligned} \tag{1}$$

Considering the range of numeric in computer and, D^* can be rewritten as follows by employing log operation.

$$\begin{aligned} D^* &= \arg\max_{L,\boldsymbol{e}_L,\boldsymbol{S}_L} \tilde{P}\left(\boldsymbol{S}_L,T\right) \\ &= \arg\max_{L,\boldsymbol{e}_L}\max_{\boldsymbol{S}_L}\sum_{l=1}^{L} ll\left(S_l,T\left(e_{l-1}+1:e_l\right)\right) \end{aligned} \tag{2}$$

To find out the solution of formula (2), we need to search over all possible combination of sign sequence labels, with all possible sign segment sequence for each sign label. Obviously, it's a huge work to search an optimal sign sequence label without nice searching strategy. In this paper, Level Building algorithm, a dynamic programming method, is utilized to find out the optimal solution (Yang et al., 2010).

In the Level Building algorithm, the question is divided into two parts recursively, isolated sign matching in the current level and sentence matching before current level. For example, by structuring optimization from the last label, we have

$$\begin{aligned} \max_{L,\boldsymbol{e}_L,\boldsymbol{S}_L} \tilde{P}\left(\boldsymbol{S}_L,T\right) = \max_{L,S_L}\Big\{ & \max_{\boldsymbol{e}_{L-1},\boldsymbol{S}_{L-1}} \tilde{P}\left(\boldsymbol{S}_{L-1},T\left(e_0+1:e_{L-1}\right)\right) \\ & +ll\left(S_L,T\left(e_{L-1}+1:e_L\right)\right)\Big\} \end{aligned} \tag{3}$$

In this paper, we utilize a 3-dimensional logarithmic likelihood matrix LL of size $L_{\max}\times N_\lambda\times M$ to store the likelihood. $LL\left(l,i,m\right)$ measures the maximal cumulative logarithmic probability of matching l labels to the sentence up to mth frame, with ith sign as lth label, where $1\le l\le L_{\max}$, $1\le i\le N_\lambda$, $1\le m\le M$. According to formula (3), LL can be written as

$$LL\left(l,i,m\right)=\begin{cases} ll\left(\lambda_i,T\left(1:m\right)\right), & \text{if } l=1 \\ \max\limits_{k,j}\Big\{LL\left(l-1,k,j\right) & \\ \quad +ll\left(\lambda_i,T\left(j+1:m\right)\right)\Big\}, & \text{otherwise} \end{cases} \tag{4}$$

Since our goal is to find the optimal score to match the sentence, we need to find the local optimal score at the last frame of the sentence at each level and search the global optimal score over these local optimal scores. After the global optimal score is obtained, a backtracking method is employed to reconstruct the optimal sign labels sequence. The global optimal score can be obtained by

$$p^*=\max_{l,i}\left(LL\left(l,i,M\right)\right) \tag{5}$$

To backtrack for reconstructing the optimal sign label sequence, a predecessor array ψ is employed as Yang et al. (2010), whose indices are corresponded to LL. $\psi\left(l,i,m\right)$ recodes sign label and ending frame of $(l-1)$th sign at the level $l-1$.

$$\psi\left(l,i,m\right)=\begin{cases} -1, & \text{if } l=1 \\ \arg\max\limits_{k,j}\Big\{LL\left(l-1,k,j\right) & \\ \quad +ll\left(\lambda_i,T\left(j+1:m\right)\right)\Big\}, & \text{otherwise} \end{cases} \tag{6}$$

2.2. Handling ME with Threshold Model

Generally, sentences are composed of sign sequences and movement epenthesis. Obviously, embedding the ME label into the Level Building algorithm can achieve better performance.

In this paper, a threshold-HMM model is utilized for ME label, proposed by Lee and Kim (1999). In the threshold-HMM model, the likelihood between the sign model and test sequence is computed firstly. Then, we judge whether the test sequence is a true sign sequence by comparing the maximal likelihood among the true-sign models with the likelihood generated by the non-sign model. If the former is beyond the latter, the test sequence is labeled as corresponding sign label, otherwise ME label.

In Level Building algorithm, sign model λ_{N_λ} is considered as ME model. Note that $LL(l,N_\lambda,m)$ cannot be computed using formula (4) directly, since $LL(l,N_\lambda,m)$ is activated only when the sign sequence is labeled as ME at level *l*. Instead, we compute $LL(l,N_\lambda,m)$ as follows.

Firstly, $LL(l,i,m)$ is calculated for sign i label and assuming the optimal candidate sign sequence is from frame j^*+1 to m at level *l*. Then, if $ll\left(\lambda_i,T\left(j^*+1:m\right)\right)$ is lower than $ll\left(\lambda_{N_\lambda},T\left(j^*+1:m\right)\right)$, $LL(l,N_\lambda,m)$ is activated with formula (7), otherwise not activated.

$$LL(l,N_\lambda,m)=LL\left(l-1,k^*,j^*\right)+ll\left(\lambda_{N_\lambda},T\left(j^*+1:m\right)\right) \quad (7)$$

$$\psi(l,N_\lambda,m)=\left(k^*,j^*\right) \quad (8)$$

where k^* and j^* are the label and ending frame of sign at level $l-1$, along the optimal backtracking path via $\psi(l,i,m)$.

Note that there may exist several $LL(l,N_\lambda,m)$, since the process above executes for each sign label. As $LL(l,N_\lambda,m)$ measures the maximal cumulative logarithmic probability, the maximal one would be recoded as final $LL(l,N_\lambda,m)$ and $\psi(l,N_\lambda,m)$ is also updated accordingly.

2.3. Sign Length Constraint and Grammar Constraint

2.3.1. Sign Length Constraint

In the recognition task of the continuous sign language, three errors often occur: substitution error, insertion error and deletion error. In the substitution error, one sign is labeled as another sign. In the insertion error, one complete sign is split into two incomplete signs and two signs are merged into one sign in the deletion error. Actually, good sign models are beneficial to reducing these three errors. As HMM, a generative model, is used in this paper, discriminative information between different signs are not considered. Hence, embedding sign discriminative information in Level Building would be benefit to enhancing recognition performance.

To add sign discriminative information, sign length constraint is employed in this paper. In the sign length constraint, sign length is the number of the frame in sign and it is discriminative since a long sign owns more frames with less frames for a short sign. We use an array C to store the sign length range, where C_i^{low} and C_i^{high} stores the minimal and maximal frame length of ith sign, respectively. C can be obtained from sign training data.

2.3.2. Grammar Constraint

Unlike the isolated sign language recognition, grammar constraint can be employed to improve recognition performance at sentence level or phrase level in the continuous sign language recognition. In this paper, referred to Yang et al. (2010), a sample-based model of the bigram is employed and represented using a relationship matrix $R(i,j), 1\le i\le N_\lambda, 1\le j\le N_\lambda$, where

$$R(i,j)=\begin{cases}1, \text{if } \lambda_i \text{ can be the predecessor of } \lambda_j\\ 0, \text{if } \lambda_i \text{ cannot be the predecessor of } \lambda_j\end{cases} \quad (9)$$

Based on instances in training text corpus, $R(i,j)$ is set to be 1 or 0, if an instance is ether found or not found in the corpus. Considering that one sign sequence may be split into two sequences with same label, we set $R(i,i)=1$ for all i. Besides, we set $R(i,j)=1$ if $i=N_\lambda$ or $j=N_\lambda$ to allow ME label before or after each sign.

Sign length constraint and grammar constraint are incorporated into the Level Building algorithm by updating formula (4) and (6) as

$$LL(l,i,m)=\begin{cases}ll\left(\lambda_i,T(1:m)\right), & \text{if } l=1\,\&\, C_i^{low}\le m\le C_i^{high}\\ -\text{inf}, & \text{if } l=1\,\&\left(m\le C_i^{low} \text{ or } m\ge C_i^{high}\right)\\ \max\limits_{\substack{k,j\\ C_i^{low}\le m-j\le C_i^{high}\\ R(k,i)=1}} \left\{LL(l-1,k,j) + ll\left(\lambda_i,T(j+1:m)\right)\right\}, & \text{otherwise}\end{cases} \quad (10)$$

$$\psi(l,i,m)=\begin{cases}-1, & \text{if } l=1\\ \arg\max\limits_{\substack{k,j\\ C_i^{low}\le m-j\le C_i^{high}\\ R(k,i)=1}} \left\{LL(l-1,k,j) + ll\left(\lambda_i,T(j+1:m)\right)\right\}, & \text{otherwise}\end{cases} \quad (11)$$

3. Fast Algorithm for HMM and Coarse Segmentation for level number

3.1. Fast Algorithm for HMM

It is found that both Yang's method LB-DTW (Yang et al., 2010) and our method LB-HMM hold high computation and the derivations of time complexity are stated as follows.

Given one test sentence T with N_T frames, to find the global optimal score of 3-dimensional logarithmic likelihood matrix LL in Level Building algorithm, we need to calculate $L_{\max}\cdot N_\lambda\cdot N_T$ elements of LL and the average number of candidate signs is $N_\lambda\cdot N_T/2$ for computing one element. Hence, the total number of candidate signs is

$$N_{SIGN}=L_{\max}\cdot N_\lambda\cdot N_T\cdot N_\lambda\cdot N_T/2=L_{\max}\cdot\left(N_\lambda\right)^2\cdot\left(N_T\right)^2/2 \quad (12)$$

with $N_T/2$ as the average number of frames in these candidate signs.

In LB-DTW algorithm, the time complexity of DTW is $O\left(N_T\cdot N_P\right)$, with N_P as the average number of frames in sign patterns. Thus, the overall time complexity of LB-DTW is

$$O\left(N_{SIGN}\cdot N_T\cdot N_P\right)=O\left(\left(N_T\right)^3\right). \quad (13)$$

In LB-HMM algorithm, the time complexity of HMM is $O\left(\left(N_S\right)^2 N_T\right)$, with N_S as the number of hidden states in HMM. Note that we just consider the computation of Forward algorithm (Rabiner, 1989), since $b_j\left(o_t\right)$ need to be calculated only once using Gaussian mixture model (GMM) before Level Building algorithm, where $b_j\left(o_t\right)$ is the probability of the observation o_t generated by hidden state j (Rabiner, 1989). So the overall time complexity of LB-HMM is

$$O\left(N_{SIGN}\cdot\left(N_S\right)^2 N_T\right)=O\left(\left(N_T\right)^3\right). \quad (14)$$

Both our system and Yang's system (Yang et al., 2010) are too slow to run online, since they both run in hours when testing on Intel Core i5 3.3 GHz CPU with MALTAB and no parallel acceleration. Thus, it is necessary to reduce computation of LB-HMM, and there are two ways to implement it. One is to reduce N_{SIGN} by determining more explicit searching window at each level, while the other one is to reduce the time complexity of HMM. Working on the latter way, a fast method is proposed to calculate the approximate likelihood of HMM in this paper.

To claim our method, some notations are declared as:

1) $T=\left\{o_1,o_2,\cdots,o_{N_T}\right\}$: a N_T frames sequence, with o_1 as the first observation symbol and o_{N_T} as the last one.

2) $T_1=\{o_1,o_2,\cdots,o_M\}$: a M frames sequence, with o_1 as the first observation symbol and o_M as the last one.

3) $T_2=\{o_{M+1},o_{M+2},\cdots,o_{N_T}\}$: a N_T-M frames sequence, with o_{M+1} is the first observation symbol and o_{N_T} as the last one.

4) $ll(\lambda,T_2)$: the log likelihood of sequence T_2 generated by sign model λ .

5) $\mathrm{O}_{m:n}=\{o_m,o_{m+1},\cdots,o_n\}$: observation symbol sequence with o_m as start frame and o_n as end frame.

6) $\mathrm{Q}_{m:n}=\{q_m,q_{m+1},\cdots,q_n\}$: state sequence with q_m as start state and q_n as end state.

Mostly, Forward algorithm is employed to calculate $ll(\lambda,T_2)$. However, its repeated application without storing previous results in Level Building makes LB-HMM run slowly. In order to reduce the time complexity of LB-HMM, a fast algorithm is proposed to compute $ll(\lambda,T_2)$ approximately. The detail derivation of fast algorithm will be presented in **Appendix A** and the result formula is presented as follows:

$$\begin{aligned} ll(\lambda,T_2)&=\log\max_{\mathrm{Q}_{1:N_T}}P\left(\mathrm{O}_{1:N_T},\mathrm{Q}_{1:N_T}\mid\lambda\right)\\ &\quad-\log\max_{\mathrm{Q}_{1:M}}P\left(\mathrm{O}_{1:M},\mathrm{Q}_{1:M}\mid\lambda\right)+\mathrm{B}_{q_M^*} \end{aligned} \tag{15}$$

where,

$$\mathrm{B}_{q_M^*}=(N_T-M-1)\cdot\log N_s+\log\sum_{i}^{N_s}\frac{P(q_{M+1}=i\,|,\lambda)}{P\left(q_{M+1}=i\mid q_M=q_M^*,\lambda\right)} \tag{16}$$

q_M^* is the optimal decoded state at frame M of T_1 .

Using Viterbi algorithm, the former two items in formula (15) can be computed before Level Building algorithm, with $O\left((N_S)^2N_T\right)$ time complexity. And $\mathrm{B}_{q_M^*}$ can be stored offline, thus we can neglect it. It must be noted that these three items can be obtained before Level Building algorithm, so only two basic operations are needed to compute $ll(\lambda,T_2)$ and the time complexity of Fast-HMM based Level Building (LB-Fast-HMM) algorithm is

$$O\left(N_{SIGN}\cdot 2+(N_S)^2N_T\right)=O\left((N_T)^2\right). \tag{17}$$

As shown in Table 1, LB-Fast-HMM algorithm owns $O\left((N_T)^2\right)$ time complexity while $O\left((N_T)^3\right)$ for LB-DTW and LB-HMM. Obviously, our system with LB-Fast-HMM runs faster than the other two.

Table 1
Comparison of complexity

Method	Time Complexity
LB-DTW	$O\left((N_T)^3\right)$
LB-HMM	$O\left((N_T)^3\right)$
LB-Fast-HMM	$O\left((N_T)^2\right)$

Applying the fast algorithm to HMM, the main process of LB-Fast-HMM is described as follows.

Firstly, before using Level Building algorithm, we compute decoding probability $\max P\left(\mathrm{O}_{1:j},\mathrm{Q}_{1:j}\mid\lambda_i\right)$ and decoding path φ_i for ith sign model via Viterbi algorithm, and use decoding probability array P_i^j to store the probability and the state array Φ_i^j to store the last state along the backtracking path, respectively.

$$\mathrm{P}_i^j=\max P\left(\mathrm{O}_{1:j},\mathrm{Q}_{1:j}\mid\lambda_i\right),\quad 1\le j\le N_T,1\le i\le N \tag{18}$$

$$\Phi_i^j=\varphi_i^j,\quad 1\le j\le N_T,1\le i\le N \tag{19}$$

Then, at level 1 , P_i^m is utilized to represent the log likelihood $ll(\lambda_i,T(1:m))$ for ith sign model.

$$ll(\lambda_i,T(1:m))=\mathrm{P}_i^m \tag{20}$$

At other level, $ll(\lambda_i,T(j+1:m))$ can be computed using formula (21).

$$ll(\lambda_i,T(j+1:m))=\mathrm{P}_i^m-\mathrm{P}_i^j+\mathrm{B}_{\Phi_i^j} \tag{21}$$

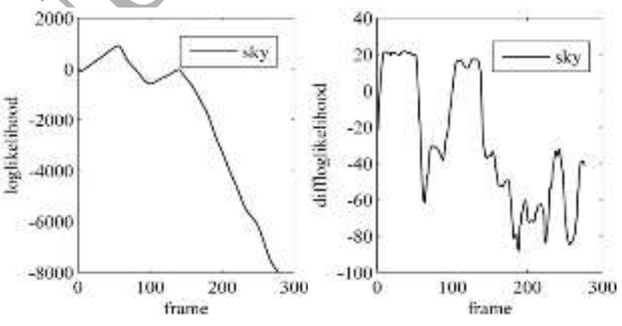
Finally, $LL(l,i,m)$ can be computed using formula (4), together with formula (20) or (21) at different level.

3.2. Coarse Segmentation for Level Number

In Level Building algorithm, the maximal number of sign in a test sentence is determined by the level number $L_{\max}$. If $L_{\max}$ is lower than the true sign number in test sentence, serious deletion error may occur in sentence interpreting, due to bad segmentation caused by lower $L_{\max}$. Hence, we must ensure $L_{\max}$ is higher than the true sign number. However, system will run slowly when using too high $L_{\max}$. Thus, it is necessary to determine a proper $L_{\max}$. In this paper, a coarse segmentation method is proposed to provide an adaptive $L_{\max}$, which is close to the true sign number in the test sentence.

Through experiments, it is found that the log likelihood is approximately proportional to frame *n*, when using true or similar sign label. In Fig.1, given sentence "sky, snow, power, request, welcome", the log likelihood of sign label “sky” is plot in (a) and the difference of log likelihood is shown in (b). For subsequence from frame 8 to 51, the corresponding log likelihood increases linearly, and the variance of difference is very small, so this subsequence is labeled as a candidate sequence for sign “sky”. So as subsequence from frame 105 to 135.

Hence, we can utilize this linear property to provide a coarse segmentation for generating $L_{\max}$ of sentence T and the details are discussed in **Algorithm 1**.



(a) (b)

Fig.1. (a) log likelihood of sentence "sky, snow, power, request, welcome", under sign label "sky". (b) difference of the log likelihood between the adjacent frame.

Algorithm 1:

Initial: $i=1,k=1$

Step1: calculate N_T log likelihoods for i th sign model using Viterbi algorithm and store these as P, where N_T is the frame length of T and P_j is the log likelihood of sequence from frame 1 to j .

Step2: calculate difference of log likelihoods P using $P_{j+1}-P_j$ and store these in an array DP .

Step3: search segmentation for ith sign model. Towards one subsequence from frame m to frame n , if all elements in $DP(m:n)$ are above 0 and variance of $DP(m:n)$ is lower than variance threshold VT (determined by experiments), then we label this subsequence as segmentation for sign i and store in $SEG\{k\}$, updating $k=k+1$ simultaneously.

Step4: update $i=i+1$, if $i<N_\lambda$ go to Step 1, otherwise, go to step 5.

Step5: merge segmentations in *SEG*. If absolute difference between start frames or end frames in two segmentations is below the threshold (determined by experiments), these two segmentations need be merged into a new segmentation with the lower starting frame as the starting frame and the higher ending frame as the ending frame.

Step6: the size of *SEG* is $L_{\max}$.

4. Experiment

4.1. Dataset

In this paper, we test on a Chinese sign language dataset recorded by Microsoft KINECT, which can provide RGB image with 640*480 resolution, depth image with 640*480 resolution and 20 skeleton points at the frame rate of 30 FPS. Several signs for daily communication are included in our dataset and there exists long signs, short signs and similar signs in these signs.

The dataset contains two parts: 1) 714 sign samples over 21 isolated signs from 8 signers, where there are 34 samples for each sign; 2) 100 sentence samples over 20 different sentences from 2 signers, where each sentence is consisted of 5 signs randomly selected from 21 isolated signs. Table 2 shows the list of 21 isolated signs and Fig 2 shows the screenshots of three signs: Welcome, Everyone and Happy.

4.2. Feature representation

Towards features, only motion trajectory is used in this paper. We can obtain 3D body skeleton information from KINECT SDK and motion trajectory descriptor consists of six skeleton points: Right Hand, Right Wrist, Right Elbow, Left Hand, Left Wrist and Left Elbow. To ensure robustness, the scale and position normalizations are employed using

$$\bar{\mathbf{Pt}}(i)=\frac{\mathbf{Pt}(i)-\mathbf{Pt}(\text{Spine})}{\left\|\mathbf{Pt}(\text{Head})-\mathbf{Pt}(\text{Spine})\right\|} \tag{22}$$

where $\mathbf{Pt}(i)$ represents one of the six skeleton points mentioned above and $\mathbf{Pt}(\text{Head})$ is the head skeleton point, while $\mathbf{Pt}(\text{Spine})$ as the spine point and $\bar{\mathbf{Pt}}(i)$ as the normalized result of $\mathbf{Pt}(i)$.

4.3. Training of HMM

In order to achieve optimal performance, we employ an ergodic structure with 4 hidden states and 2 Gaussians per state for each HMM, and train HMM for each sign using Baum Welch algorithm (Rabiner, 1989). In ISR, 3-fold cross validation is conducted on 714 sign samples dataset, where 2/3 of 34 sign samples are employed to train HMM for each sign and 1/3 of 34 sign samples are employed for testing in each fold. And the HMM sign models, achieving optimal performance in 3-fold cross validation, is utilized as sign models in CSLR.

4.4. Isolated sign recognition

To demonstrate HMM is better than DTW when depicting signs, experiments of ISR are conducted in this section. As mentioned in section 4.3, 3-fold cross validation is conducted on 714 sign samples dataset. In each fold, 2/3 of 714 sign samples are set as the training set and other 1/3 are set as the testing set, with 22 samples in the training set and 12 samples in the testing set for each sign. It must be noted that, for each sign, we use the total training samples to train HMM and use the optimal sign sample among the training samples as DTW pattern. The average recognition results of isolated sign recognition are illustrated in Table 3.

As shown in the Table 3, in our dataset, HMM achieves lower recognition error rate than DTW, which also implies HMM works better in continuous sign language recognition.

4.5. Continuous sign language recognition

In continuous sign language recognition, we conduct three studies. In the first study, we focus on parameters of threshold model for ME label and test using fast-HMM-based level building, with grammar and sign length constraints. In the second study, we compare effectiveness of different constraints in level building algorithm. At last, we compare the recognition performance and execution time with other methods in the last study. All experiments are tested on 100 sentences. To measure the recognition performance the sentences, the error rate ε can be defined referring to Fang et al. (2007).

$$\varepsilon=\frac{S+I+D}{N}\times 100\% \tag{23}$$

where S, I and D denote the error numbers of the substitution, insertion and deletion. N denotes the number of signs in all sentences.

4.5.1. Parameter for threshold model

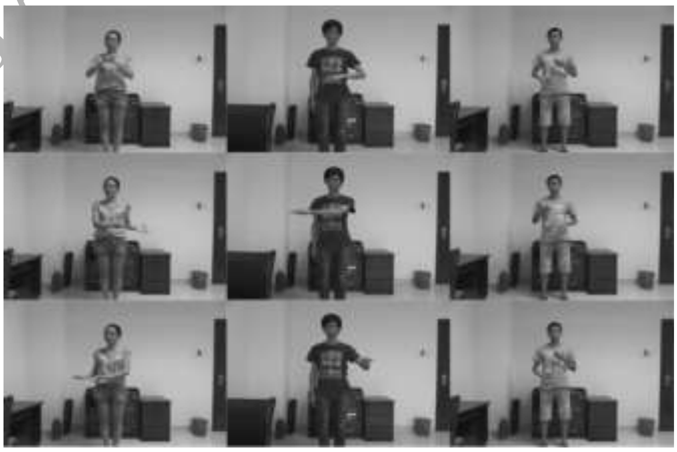
Threshold model is obtained by including all hidden states of each HMM and merging similar states (Lee and Kim, 1999). The state number in threshold model determines the ability to distinguish whether test sign sequence is a non-sign sequence. To achieve optimal performance in CSLR, we choose optimal state number via experiments. Fig. 3 shows the variation of the errors with different state number. As seen, when the state number is N_s, experiment achieves minimal error rate, where N_s equals 4 in the experiments. Thus, the state number of threshold model is 4.

Table 2
Signs in the database

Signs	Sky, Snow, Power, Entrust, Welcome, Need, Very, Help, Fly, Staunch, Warm, Happy, Yet, Airplane, Hope, Sorry, Know, Good Bye, Take Charge, Everyone, Surrender

Table 3
Isolated sign recognition performance

Method	DTW	HMM
Error Rate	4.91%	2.23%



Welcome Everyone Happy

Fig. 2. Screenshots of three sign samples

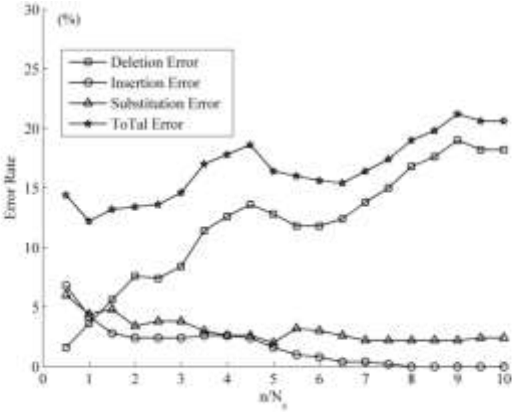


Fig. 3. The variation of the errors with different state number in threshold model.

4.5.2. Effectiveness of different constraints

The primary focus of these experiments in this study is to test the effectiveness of different constraints. Using optimal threshold model in section 4.5.1, we conduct the following experiments. In Fig. 4, we present comparison results of the error rates using LB-Fast-HMM with different constraints. The results show that both the insertion errors and substitution errors have decreased significantly by using the grammar constraint, and the insertion errors also have decreased by using the sign length constraint.

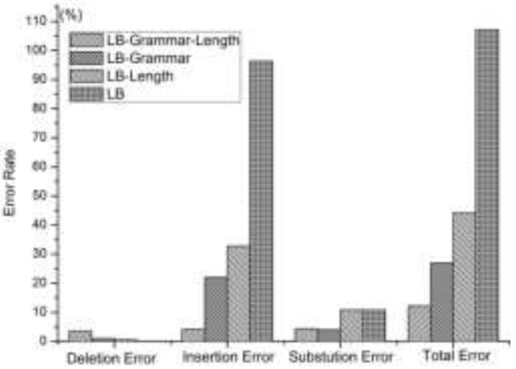


Fig. 4. The error rates for level building with different constraints.

4.5.3. Comparison with other approaches

We compare the performance of our approach with two other methods: T-HMM (Lee and Kim, 1999), LB-DTW (Yang, 2010). Table 4 shows comparison results of error rate and runtime using different approaches. All the methods are implemented in MATLAB on a Windows 7 computer with an Intel Core i5 3.3GHz CPU and no GPU parallel acceleration. The result demonstrates that our proposed method outperforms LB-DTW and T-HMM, with a low error rate of 12.20%. Besides, LB-DTW and LB-HMM run in hours for testing one sentence while LB-Fast-HMM run in several seconds.

Table 4
Comparison of accuracy and runtime

Method	ErrorRate	RunTimePerSentence
T-HMM (Lee and Kim,1999)	33.80%	0.3217s
LB-DTW (Yang, 2010)	25.60%	5101.53s
LB-HMM (Proposed method)	13.40%	3559.35s
LB-Fast-HMM (Proposed method)	12.20%	8.98s

5. Conclusion

In terms of accuracy and running time, the advantages of the LB-Fast-HMM in continuous sign recognition are addressed and demonstrated in this paper. To improve the recognition performance of Level Building, grammar and sign length constraints are employed to reduce the insertion and substitution errors. To handle the high computation in Level Building, we propose Fast-HMM. Experiments on a Chinese KINECT dataset demonstrate the effectiveness of our approach compared to several existing methods.

Future work includes employing hand shape information to improve recognition rate and designing explicit search window for level building to decrease running time.

Appendix A. Derivation of fast algorithm for HMM.

Before the derivation, the following notations are declared as in section 3.1.

In Rabiner (1989), given the observation sequence $\mathrm{O}=\{o_1,o_2,\cdots,o_n\}$ and the model λ, $P(\mathrm{O}\mid\lambda)$ is the likelihood of the observation sequence generated by model λ and Forward algorithm is often used to compute it. However, its repeated application without storing previous results in Level Building makes LB-HMM run slowly. In this paper, working on formula (A.1), a fast algorithm is proposed to approximately compute $P(\mathrm{O}\mid\lambda)$ to reduce the time complexity of LB-HMM.

$$P(\mathrm{O}\mid\lambda)=\sum_{\text{all Q}}P(\mathrm{O},\mathrm{Q}\mid\lambda) \qquad \text{(A.1)}$$

Obviously, the joint probability $P(\mathrm{O},\mathrm{Q}\mid\lambda)$ is simply the product of $P(\mathrm{O}\mid\mathrm{Q},\lambda)$ and $P(\mathrm{Q}\mid\lambda)$ (Rabiner, 1989).

$$P(\mathrm{O},\mathrm{Q}\mid\lambda)=P(\mathrm{O}\mid\mathrm{Q},\lambda)P(\mathrm{Q}\mid\lambda) \qquad \text{(A.2)}$$

where $P(\mathrm{O}\mid\mathrm{Q},\lambda)$ is the probability of the observation sequence O for the state sequence Q and $P(\mathrm{Q}\mid\lambda)$ is the probability of the state sequence Q.

Thus, for test sequences T, T_1 and T_2 declared in section 3.1, the corresponding joint probability items can be written as

$$P(\mathrm{O}_{1:N_T},\mathrm{Q}_{1:N_T}\mid\lambda)=P(\mathrm{O}_{1:N_T}\mid\mathrm{Q}_{1:N_T},\lambda)P(\mathrm{Q}_{1:N_T}\mid\lambda) \qquad \text{(A.3)}$$

$$P(\mathrm{O}_{1:M},\mathrm{Q}_{1:M}\mid\lambda)=P(\mathrm{O}_{1:M}\mid\mathrm{Q}_{1:M},\lambda)P(\mathrm{Q}_{1:M}\mid\lambda) \qquad \text{(A.4)}$$

$$\begin{aligned}&P(\mathrm{O}_{M+1:N_T},\mathrm{Q}_{M+1:N_T}\mid\lambda)\\&=P(\mathrm{O}_{M+1:N_T}\mid\mathrm{Q}_{M+1:N_T},\lambda)P(\mathrm{Q}_{M+1:N_T}\mid\lambda)\end{aligned} \qquad \text{(A.5)}$$

First, referred to Lee and Kim (1999), $P(\mathrm{O}_{1:N_T},\mathrm{Q}_{1:N_T}\mid\lambda)$ and $P(\mathrm{Q}_{1:N_T}\mid\lambda)$ can be written as follows, through some simple decompositions.

$$\begin{aligned}&P(\mathrm{O}_{1:N_T}\mid\mathrm{Q}_{1:N_T},\lambda)\\&=P(\mathrm{O}_{1:M}\mid\mathrm{Q}_{1:M},\lambda)P(\mathrm{O}_{M+1:N_T}\mid\mathrm{Q}_{M+1:N_T},\lambda)\end{aligned} \qquad \text{(A.6)}$$

$$\begin{aligned}&P(\mathrm{Q}_{1:N_T}\mid\lambda)\\&=P(\mathrm{Q}_{1:M}\mid\lambda)P(\mathrm{Q}_{M+1:N_T}\mid\lambda)\frac{P(q_{M+1}\mid q_M,\lambda)}{P(q_{M+1}\mid\lambda)}\end{aligned} \qquad \text{(A.7)}$$

Then, using formula (A.3)-(A.7), $P(\mathrm{O}_{1:N_T},\mathrm{Q}_{1:N_T}\mid\lambda)$ can be rewritten as

$$\begin{aligned}&P(\mathrm{O}_{1:N_T},\mathrm{Q}_{1:N_T}\mid\lambda)\\&=P(\mathrm{O}_{1:M},\mathrm{Q}_{1:M}\mid\lambda)P(\mathrm{O}_{M+1:N_T},\mathrm{Q}_{M+1:N_T}\mid\lambda)\frac{P(q_{M+1}\mid q_M,\lambda)}{P(q_{M+1}\mid\lambda)}\end{aligned} \qquad \text{(A.8)}$$

By derivation from formula (A.8), $P(\mathrm{O}_{M+1:N_T},\mathrm{Q}_{M+1:N_T}\mid\lambda)$ can be written as

$$P(\mathrm{O}_{M+1:N_T},\mathrm{Q}_{M+1:N_T}\mid\lambda)=\frac{P(\mathrm{O}_{1:N_T},\mathrm{Q}_{1:N_T}\mid\lambda)}{P(\mathrm{O}_{1:M},\mathrm{Q}_{1:M}\mid\lambda)}\cdot\frac{P(q_{M+1}\mid,\lambda)}{P(q_{M+1}\mid q_M,\lambda)} \qquad \text{(A.9)}$$

According to formula A.1, likelihood of T_2 can be obtained by summing joint probability $P(\mathrm{O}_{M+1:N_T},\mathrm{Q}_{M+1:N_T}\mid\lambda)$ over all possible state sequences.

$$P(\mathrm{O}_{M+1:N_T}\mid\lambda)=\sum_{\text{all }\mathrm{Q}_{M+1:N_T}}P(\mathrm{O}_{M+1:N_T},\mathrm{Q}_{M+1:N_T}\mid\lambda) \qquad \text{(A.10)}$$

It must be noted that $P(\mathrm{O}_{1:M},\mathrm{Q}_{1:M}\mid\lambda)$ is independent on $\mathrm{Q}_{M+1:N_T}$, hence, we can replace $P(\mathrm{O}_{1:M},\mathrm{Q}_{1:M}\mid\lambda)$ with $\max\limits_{\mathrm{Q}_{1:M}}P(\mathrm{O}_{1:M},\mathrm{Q}_{1:M}\mid\lambda)$ and obtain inequality below.

$$P(\mathrm{O}_{M+1:N_T}\mid\lambda)\geq\frac{\sum\limits_{\text{all }\mathrm{Q}_{M+1:N_T}}P(\mathrm{O}_{1:N_T},\mathrm{Q}_{1:N_T}\mid\lambda)\cdot\dfrac{P(q_{M+1}\mid,\lambda)}{P(q_{M+1}\mid q_M,\lambda)}}{\max\limits_{\mathrm{Q}_{1:M}}P(\mathrm{O}_{1:M},\mathrm{Q}_{1:M}\mid\lambda)} \qquad \text{(A.11)}$$

In addition, $P(\mathrm{O}_{M+1:N_T}\mid\lambda)$ can be further simplified by replacing $P(\mathrm{O}_{1:N_T},\mathrm{Q}_{1:N_T}\mid\lambda)$ with $\max\limits_{\mathrm{Q}_{1:N_T}}P(\mathrm{O}_{1:N_T},\mathrm{Q}_{1:N_T}\mid\lambda)$.

$$P\left(\mathrm{O}_{M+1:N_T}\,|\,\lambda\right)\geq\frac{\max\limits_{\mathrm{Q}_{1:N_T}}P\left(\mathrm{O}_{1:N_T},\mathrm{Q}_{1:N_T}\,|\,\lambda\right)}{\max\limits_{\mathrm{Q}_{1:M}}P\left(\mathrm{O}_{1:M},\mathrm{Q}_{1:M}\,|\,\lambda\right)}\cdot\sum_{\text{all }\mathrm{Q}_{M+1:N_T}}\frac{P\left(q_{M+1}\,|,\lambda\right)}{P\left(q_{M+1}\,|\,q_M,\lambda\right)}\tag{A.12}$$

Finally, the likelihood can be calculated approximately as

$$P\left(\mathrm{O}_{M+1:N_T}\,|\,\lambda\right)\approx\frac{\max\limits_{\mathrm{Q}_{1:N_T}}P\left(\mathrm{O}_{1:N_T},\mathrm{Q}_{1:N_T}\,|\,\lambda\right)}{\max\limits_{\mathrm{Q}_{1:M}}P\left(\mathrm{O}_{1:M},\mathrm{Q}_{1:M}\,|\,\lambda\right)}\cdot\sum_{\text{all }\mathrm{Q}_{M+1:N_T}}\frac{P\left(q_{M+1}\,|,\lambda\right)}{P\left(q_{M+1}\,|\,q_M,\lambda\right)}\tag{A.13}$$

where,

$$\sum_{\text{all }\mathrm{Q}_{M+1:N_T}}\frac{P\left(q_{M+1}\,|,\lambda\right)}{P\left(q_{M+1}\,|\,q_M,\lambda\right)}=\left(N_s\right)^{N_T-M-1}\sum_{i}^{N_s}\frac{P\left(q_{M+1}=i\,|,\lambda\right)}{P\left(q_{M+1}=i\,|\,q_M=q_M^*,\lambda\right)}\tag{A.14}$$

N_s is the number of hidden state in HMM, q_M^* is the optimal decoding state at frame M of T_1, via Viterbi algorithm.

Performing logarithm, formula (A.13) is written as

$$\begin{aligned}ll\left(\lambda,T_2\right)&=\log\max_{\mathrm{Q}_{1:N_T}}P\left(\mathrm{O}_{1:N_T},\mathrm{Q}_{1:N_T}\,|\,\lambda\right)\\&\quad-\log\max_{\mathrm{Q}_{1:M}}P\left(\mathrm{O}_{1:M},\mathrm{Q}_{1:M}\,|\,\lambda\right)+\mathrm{B}_{q_M^*}\end{aligned}\tag{A.15}$$

where

$$\begin{aligned}\mathrm{B}_{q_M^*}&=\left(N_T-M-1\right)\cdot\log N_s\\&\quad+\log\sum_{i}^{N_s}\frac{P\left(q_{M+1}=i\,|,\lambda\right)}{P\left(q_{M+1}=i\,|\,q_M=q_M^*,\lambda\right)}\end{aligned}\tag{A.16}$$

References

Rabiner, L., 1989.A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. IEEE 77(2), 257-289.

Starner, T., Pentland, A., 1998. Real-time American sign language recognition using desk and wearable computer based video. IEEE Trans. Pattern Anal. Mach. Intell. 20 (12), 1371-1375.

Vogler, C., Metaxas, D., 1999. Parallel Hidden Markov Models for American Sign Language Recognition. In: Proc. 7th IEEE Internat. Conf. on Computer vision, pp. 116-122.

Lee, H.-K., Kim, J., 1999. An HMM-Based Threshold Model Approach for Gesture Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 21 (10), 961-973.

Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D., 2005. Conditional models for contextual human motion recognition. In: Proc. 10th IEEE Internat. Conf. on Computer Vision, pp. 1808-1815.

Wang, S., Quattoni, A., Morency, L., Demirdjian, D., Darrell, T., 2006. Hidden Conditional Random Fields for Gesture Recognition. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1521-1527.

Theodorakis, S., Katsamanis, A., Maragos, P., 2009. Product-HMMs for automatic sign language recognition. In: Proc. IEEE Internat. Conf. on Acoustics, Speech and Signal Processing, pp. 1601-1604.

Theodorakis, S., Katsamanis, A., Maragos, P., 2014. Dynamic-static unsupervised sequentiality, statistical subunits and lexicon for sign language recognition. Image and Vision Computing. 32 (8), 533-549.

Pitsikalis V., Theodorakis S., Vogler C., Maragos, P., 2011. Advances in Phonetics-based Sub-Unit Modeling for Transcription Alignment and Sign Language Recognition. In: 2011 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops, pp. 1-6.

Yang, H.-D., Sclaroff, S., Lee, S.-W., 2009. Sign language spotting with a threshold model based on conditional random fields. IEEE Trans. Pattern Anal. Mach. Intell. 31 (7), 1264–1277.

Yang, H.-D., Lee, S.-W., 2010. Robust Sign Language Recognition with Hierarchical Conditional Random Fields. In: Proc. 20th Internat. Conf. on Pattern Recognition, pp. 2202-2205.

Fang, G., Gao, W., 2002. A SRN/HMM System for Signer-independent Continuous Sign Language Recognition. In: Proc. 5th IEEE Internat. Conf. on Automatic Face and Gesture Recognition, pp. 312-317.

Fang, G., Gao, W., Zhao, D., 2007. Large-Vocabulary Continuous Sign Language Recognition Based on Transition-Movement Models. IEEE Trans. Syst. Man Cybern. A Syst. Humans 37 (1), 1-9.

Kelly, D., McDonald, J., Markham, C., 2009. Recognizing spatiotemporal gestures and movement epenthesis in sign language. In: Proc.13th Internat. Conf. on Machine Vision and Image Processing, pp. 145–150.

Yang, R., Sarkar, S., 2006. Gesture Recognition using Hidden Markov Models from Fragmented Observations. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 766-773.

Yang, R., Sarkar, S., Loeding, B., 2007. Enhanced Level Building Algorithm for the Movement Epenthesis Problem in Sign Language Recognition. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1-8.

Yang, R., Sarkar, S., Loeding, B., 2010. Handling Movement Epenthesis and Hand Segmentation Ambiguities in Continuous Sign Language Recognition Using Nested Dynamic Programming. IEEE Trans. Pattern Anal. Mach. Intell. 32 (3), 462–477.

Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., Presti, P., 2011. American Sign Language Recognition with the Kinect. In Proc. 13th Internat. Conf. on Multimodal Interfaces, pp. 279-286.

Lichtenauer, J.F., Hendriks, E.A., Reinders, M.J.T., 2008. Sign Language Recognition by Combining Statistical DTW and Independent Classification. IEEE Trans. Pattern Anal. Mach. Intell. 30 (11), 2040–2046.

Hernández-Vela A., Bautista M.Á., Perez-Salab X.,Ponce-López V., Escalera E.,Baró X., Pujol O., Angulo C., 2014. Probability-based Dynamic Time Warping and Bag-of-Visual-and-Depth-Words for Human Gesture Recognition in RGB-D. Pattern Recogn. Lett. 50 (1), 112-121.

Kong, W.W., Ranganath, S., 2008. Automatic Hand Trajectory Segmentation and Phoneme Transcription for Sign Language. In: 8th IEEE Internat. Conf. on Automatic Face & Gesture Recognition, pp. 1-6.

Kong, W.W., Ranganath, S., 2014. Towards subject independent continuous sign language recognition: A segment and merge approach. Pattern Recogn. 47 (3), 1294–1308.

Zhang, J., Zhou, W., Li, H., 2014. A Threshold-based HMM-DTW Approach for Continuous Sign Language Recognition. In: Proc. Internat. Conf. on Internet Multimedia Computing and Service, pp. 237.

Wu, D., Shao, L., 2014. Deep Dynamic Neural Networks for Gesture Segmentation and Recognition. In: Proc. Eur. Conf. on Computer Vision, vol. 8925, pp. 552-571.

Koller, O.; Forster, J., Ney, H., 2015. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Comput. Vis. Image Understand. 141, 108-125.

Elmezain, M., Al-Hamadi, A., Appenrodt, J., Michaelis, B., 2008. A Hidden Markov Model based continuous gesture recognition system for hand motion trajectory. In: Proc. 19th Internat. Conf. on Pattern Recognition, pp. 1-4.

Alon, J., Athitsos, V., Yuan, Q., Sclaroff, S., 2009. A unified framework for gesture recognition and spatiotemporal gesture segmentation. IEEE Trans. Pattern Anal.Mach. Intell. 31, 1685–1699

Han, J., Awad, G., Sutherland, A., 2009. Modelling and segmenting subunits for sign language recognition based on hand motion analysis. Pattern Recogn. Lett. 30 (6), 623-633.

Cheng, J., Xie, C., Bian, W., Tao, D., 2012. Feature fusion for 3D hand gesture recognition by learning a shared hidden space. Pattern Recogn. Lett. 33 (4), 476-484.

Jacob, M.G., Wachs, J.P., 2014. Context based hand gesture recognition for the operating room. Pattern Recogn. Lett. 36 (45), 196-203.

Pigou L., Dieleman S., Kindermans P.J., Schrauwen B., 2014. Sign Language Recognition Using Convolutional Neural Networks. In: Proc. Eur. Conf. on Computer Vision, pp. 572-578.

Nayak S., Duncan K., Sarkar S., Loed B., 2012. Finding Recurrent Patterns from Continuous Sign Language Sentences for Automated Extraction of Signs. The Journal of Machine Learning Research 13 (1), 2589-2615.