A SRN/HMM System for Signer-independent Continuous Sign Language Recognition

Gaolin Fang[1], Wen Gao[1,2]
[1]*Department of Computer Science and Engineering,*
Harbin Institute of Technology, Harbin, 150001 China
[2]*Institute of Computing Technology,*
Chinese Academy of Sciences, Beijing, 100080, China
{fgl, wgao}@ict.ac.cn

Abstract

Sign language recognition is to provide an efficient and accurate mechanism to transcribe sign language into text or speech. State-of-the-art sign language recognition should be able to solve the signer-independent continuous problem for practical applications. A divide-and-conquer approach, which takes the problem of continuous Chinese Sign Language (CSL) recognition as subproblems of isolated CSL recognition, is presented for signer-independent continuous CSL recognition in this paper. In the proposed approach, the improved simple recurrent network (SRN) is used to segment the continuous CSL. The outputs of SRN are regarded as the states of hidden Markov models (HMM) in which the Lattice Viterbi algorithm is employed for searching the best word sequence. Experimental results show that SRN/HMM approach has better performance than the standard HMM.

1. Introduction

Sign language as a kind of gestures is one of the most natural means of exchanging information for most deaf people. The aim of sign language recognition is to provide an efficient and accurate mechanism to transcribe sign language into text or speech so that communication between deaf and hearing society can be more convenient. Sign language recognition has emerged as one of the most important research areas in the field of human-computer interaction. In addition, it has many other applications, such as controlling the motion of a human avatar in a virtual environment (VE) via hand gesture recognition, multimodal user interface in virtual reality (VR) system.

Attempts to automatically recognize sign language began to appear in the literature in the 90's. Previous work on sign language recognition focuses primarily on finger spelling recognition and isolated sign recognition. There has been very little work on continuous sign language recognition. Starner [1] used a view-based approach with a single camera to extract two-dimensional features as input to HMMs. The correct rate was 91% in recognizing the sentences comprised 40 signs. By imposing a strict grammar on this system, an accuracy of 97% was possible with real-time performance. Liang and Ouhyoung [2] used HMMs for continuous recognition of Taiwan Sign language with a vocabulary between 71 and 250 signs with Dataglove as input devices. However, their system required that gestures performed by the signer be slow to detect the word boundary. Vogler and Metaxas [3] used computer vision methods to extract the three-dimensional parameters of a signer's arm motions, coupled the computer vision methods and HMMs to recognize continuous American sign language sentences with a vocabulary of 53 signs. They modeled context-dependent HMMs to alleviate the effects of *movement epenthesis*. An accuracy of 89.9% was observed. In addition, they used phonemes instead of whole signs as the basic units [4], experimented with a 22 sign and achieved recognition rates similar to sign-based approaches. A system has been described in our previous work [5], which used Dataglove as input device and HMMs as recognition method. It can recognize 5177 isolated signs with 94.8% accuracy in real time and recognize 200 sentences with 91.4% word accuracy.

As the previous work showed, most researches on continuous sign language recognition were done within the signer-dependent domain. No previous work on signer-independent continuous sign language recognition was reported in the literature. For continuous sign language recognition, a key problem is the effect of *movement epenthesis*, that is, transient movements between signs, which vary with the context of signs. For signer-independent recognition, different signers vary their hand shape size, body size, operation habit, rhythm and so on, which bring about more difficulties in recognition. Therefore recognition in the signer-independent domain is more challenging than in the signer-dependent one. Signer-independent continuous sign language recognition is first investigated in this paper. Our experiments show that continuous sign language has the property of segments, so a divide-and-conquer approach is proposed for continuous CSL recognition. It divides the problem of continuous CSL recognition into the subproblems of isolated CSL recognition. The SRN modified to assimilate context information is regarded as the segment detector of continuous CSL. The Lattice Viterbi algorithm is employed to search the best word sequence path in the output segments of SRN.

The organization of this paper is as follows. In Section 2 we propose the improved SRN and SRN-based segmentation for continuous sign language. In Section 3 we introduce the Lattice Viterbi algorithm in the HMM framework and discuss the computation of segment emission probability. In Section 4 we show experimental results and comparisons. The conclusion is given in the last section.

2. SRN-based segmentation

2.1. Simple recurrent network and its improvement

Elman proposed a simple recurrent network[6] in 1990 on the basis of the recurrent network described by Jordan. Networks have the dynamic memory performance because of the introduction of recurrent links. They have been successfully applied to speech recognition [7], [8], handwriting recognition [9], isolated sign language recognition [10]. The SRN in Figure 1 has four units. The input units receive the input vector I_t at time t, and the corresponding outputs are hidden units H_t, feedback units C_t, output units O_t. Defining W_C^H, W_I^H, W_H^O as the weight matrices of feedback units to hidden units, input units to hidden units and hidden units to output units, respectively.

In the network the feedback units is connected one-to-one corresponding hidden units through a unit time-delay.

$$C_t = \begin{cases} H_{t-1} & t \geq 2 \\ \overrightarrow{0.5} & t = 1 \end{cases} \tag{1}$$

The input units and the feedback units make up of the combined input vector. Φ, Ψ is respectively the bias of hidden units, output units.

$$H_t = f\left(C_t \cdot W_C^H + I_t \cdot W_I^H \quad - \quad \Phi\right) \tag{2}$$

$$O_t = f\left(H_t \cdot W_H^O \quad - \quad \Psi\right) \tag{3}$$

The activation function $f(\cdot)$ is the standard sigmoid one, $f(x) = (1+e^{-x})^{-1}$. The errors between the network's outputs and the targets are propagated back using the generalized delta rule [11].

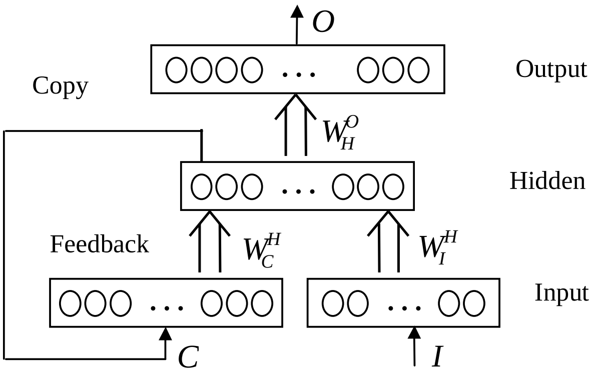

Figure 1. Simple recurrent network

Because of the introduction of feedback units, the outputs of network depends not only on the external input but also on the previous internal state which relies on the result of the preceding all external inputs. So the SRN can memorize and utilize a relatively larger preceding context [6].

The SRN can memorize the preceding context, but it cannot utilize following context information. Thus the SRN is modified to efficiently utilize the following context information in two ways. One is that the following context vector is regarded as one of the input. Thus the input vector is redefined as $I_t = [I_t \quad I_{t+1}]$, the rest of calculations are the same as the standard SRN. The other is that training samples are respectively in turn and in reverse turn fed into the SRN with the same architecture, and then one forward SRN and one backward SRN are trained. So the context information can be assimilated through two SRNs. The experiment of segmentation of continuous sign language is performed in two improved ways. But in the latter the outputs of the forward SRN conflict with those of the backward SRN so that experimental results are inferior to the former. Thus, we employ the former method as the improved SRN.

2.2. Segmentation of continuous sign language

Input preprocess: Two 18-sensor Datagloves and three position trackers are used as input devices. Two trackers are positioned on the wrist of hand and another is fixed at back (used as the reference tracker). The Datagloves collect the variation information of hand shapes with the 18-dimensional data at each hand, and the position trackers collect the variation information of orientation, position, movement trajectory. The data from position trackers can be converted as follows. The reference Cartesian coordinate system of the trackers at back is chosen, and then the position and orientation at each hand with respect to the reference Cartesian coordinate system are calculated as invariant features. Through this transformation, the data are composed of three-dimensional position vector and three-dimensional orientation vector for each hand. Furthermore, we calibrate the data of different signers by some fixed postures because everyone varies his hand shape size, body size, and operation habit. The data form a 48-dimensional vector in total for two hands. However, the dynamic range of each component is different. Each component value is normalized to ensure its dynamic range is 0-1.

Original experiment regards the 48-dimensional data as the input of SRN, chooses 30 hidden units and 3 output units. Training this network costs 28 hours in the PIII450(192M Memory) PC. But the result is not satisfied and the segment recall is only 87%. This approach is unsuccessful in the scalability and accuracy. Here, we only considers the case of detecting all actual segments and doesn't care the case of detecting several segments for an actual segment. The latter will be solved by the Lattice Viterbi algorithm searching in the HMM framework.

$$\text{Segment recall} = \frac{\text{Number of correct segments}}{\text{Number of all actual segments}} \times 100\% \tag{4}$$

Thus the above approach is modified. We use self-organizing feature maps(SOFM) as the feature extraction network. The outputs of SOFM have the strong segment properties (Figure 2 shows an example). The output of SOFM is regarded as the input of SRN by the encoding. We select 48 input units and 256 output units for the SOFM, and 16 input units and 15 hidden units and 3 output units for the SRN through trial and error. Training the SRN costs 45 minutes in the PIII450(192M Memory) PC. The segment recall is 98.8%. The input, output, training and recognition of SRN will be discussed in the following section.

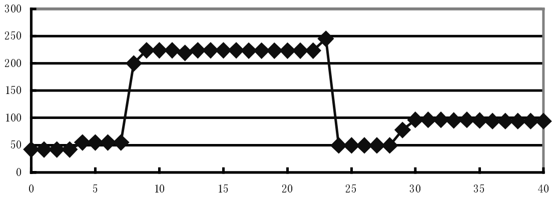

Figure 2. The segments property of sign language "我们什么时候走"(when will we leave)

Input: The encoding of 256 output units of SOFM requires 8 units, and the introduction of the following context also demands 8 units. In total there are 16 input units. The value of the input $I_t^i \in \{0,1\}$, $i = 1,2,\cdots,16$.

Output: We define 3 output units: the left boundary of segments 1, the right boundary of segments 2, the interior of segments 3, and the corresponding units o_t^1, o_t^2, o_t^3.

$$O_t = \begin{cases} [1 \quad 0 \quad 0] & \text{Output is 1} \\ [0 \quad 1 \quad 0] & \text{Output is 2} \\ [0 \quad 0 \quad 1] & \text{Output is 3} \end{cases} \tag{5}$$

Training: We cannot straightforward find the target segments because sign language is continuous. Thus automatic segmentation is employed to find the target segments. Let the sample sentence in the training $W = w_1 w_2 \cdots w_k$, the corresponding frame sequence $T = t_1 t_2 \cdots t_l$, we decide frame t_i belongs to word w_m or w_{m+1}, and if $t_i \in w_m$, $t_{i+1} \in w_{m+1}$, then frame t_i is the right boundary of segments and frame t_{i+1} is the left boundary of segments. Each state probability of frame t_i belonging to word w_m ($m = 1 \cdots k$) is calculated by the approach of isolated sign language recognition with the isolated sign language model parameters (see Section 4). Then the constrained Viterbi algorithm is used to search the best segment sequence. The constrained refers to the search path followed the sequence $w_1 w_2 \cdots w_k$. We regard the segment sequence result as the target output.

Back-propagation through time is introduced as the learning algorithm of SRN. The samples in the training set are transformed by the SOFM, and the SOFM outputs by the encoding together with the following ones are fed into the SRN, then the errors between the SRN outputs and the targets are propagated back using back-propagation and changes to the network weights are calculated. At the beginning of learning, the weight matrices, the bias of hidden units and output units are initialized the random value $(-1,+1)$, the feedback units are initialized to activations of 0.5.

Recognition: Continuous sign language in the test set is firstly fed into the SOFM. The quantized outputs are formed by the feature extraction of SOFM. The SOFM outputs by the encoding together with the following ones are fed into the SRN. The segmentation result of SRN is $i^* = \arg\max_i (o_t^i)$ at time t. The adjacency property of the left and the right boundary of segments is used as constraint in the segmentation.

3. HMM Framework

The segmentation result of SRN is fed into the HMM framework. The state of HMM for one word may cover one or several segments (2—4). Because different signs vary their lengths and may be composed of several segments, we should search the best path in those segments through the Lattice Viterbi algorithm. The best refers to two respects: the recombined segments sequence is the best, and the word sequence selected from the best segment sequence is the best. The standard Viterbi algorithm is modified in order to adapt itself to the search on lattices, and the modified algorithm is referred to as the Lattice Viterbi algorithm.

To illustrate more conveniently, we define two Viterbi algorithms: one is the isolated sign language Viterbi algorithm. For the data frame sequence of the input which is known as only one word component in advance, the algorithm will search all states of the word for each frame in recognition, and get the best state sequence.

The other is the continuous sign language Viterbi algorithm. For the data frame sequence of the input whose component cannot be known beforehand, the algorithm will search not only all states of this word but also the states of the rest words for each frame in recognition, and get the best state sequence. It requires more time and has less accuracy than the isolated sign language Viterbi algorithm in recognizing the same data frame sequence.

Both the isolated sign language Viterbi algorithm and the continuous sign language Viterbi algorithm in essence belong to the standard one, because they search the frame one by one. But the Lattice Viterbi algorithm that is different from the standard one can span one or more segments to search. It will be discussed in detail in the following section.

3.1. Lattice Viterbi algorithm

We define an edge as a triple $< t, t', q >$, starting at segment t, ending at segment t' and representing word q, where $0 \le t < T$, $t < t' \le T$. All triple $< t, t', q >$ form the set L. We introduce accumulator $\delta(t, t', q)$ that collects the maximum probability of covering the edge $< t, t', q >$. To keep track of the best path, we define the auxiliary argument $\psi(t, t', q)$ as the previous triple argument pointer of the local maximum $\delta(t, t', q)$. We denote $b(t, t', q)$ as the emission probability of word q covering segments from position t to t'. $P(q \mid q')$ is defined as the transition probability from word q' to q, which is estimated in some 3000 million Chinese words corpus from the Chinese newspapers in the 1994-1995 year. The Lattice Viterbi algorithm is as follows.

1) Intialization:

$$\delta(0, t, q) = b(0, t, q) \qquad (6)$$

$$\psi(0, t, q) = NULL \qquad (7)$$

2) Recursion:

$$\delta(t, t', q) = \max_{<t'', t, q'> \in L} \delta(t'', t, q') P(q \mid q') b(t, t', q) \qquad (8)$$

$$\psi(t, t', q) = \arg\max_{<t'', t, q'> \in L} \delta(t'', t, q') P(q \mid q') \qquad (9)$$

3) Termination:

$$P^* = \max_{<t, T, q> \in L} \delta(t, T, q) \qquad (10)$$

$$< t_1^*, T, q_1^* > = \arg\max_{<t, T, q> \in L} \delta(t, T, q) \qquad (11)$$

4) Path backtracking:

Let $T = t_0^*$, we iterate the function $< t_{i+1}^*, t_i^*, q_{i+1}^* > = \psi(t_i^*, t_{i-1}^*, q_i^*)$ until $< t_{k+1}^*, t_k^*, q_{k+1}^* > = NULL$, and the word sequence $q_k^* \text{ L } q_1^*$ is the best path.

3.2. Computation of segment emission probability

The computation of segment emission probability is similar to the one of isolated sign language recognition. But it stores the probabilities of all possible candidate words that are looked on as the emission probability $b(t, t', q)$ in the HMM framework. We get the best path through the Lattice Viterbi algorithm search at last.

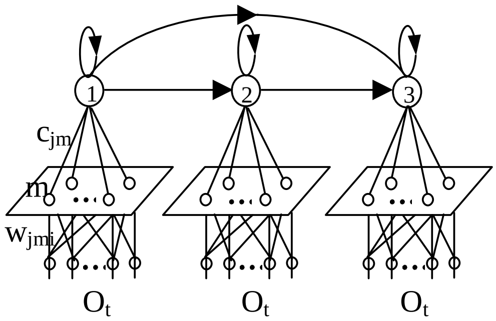

Figure 3. The architecture of SOFM/HMM

The SOFM/HMM method is employed for isolated sign language recognition [12]. This method uses an alternative probability density function (pdf) scheme that each SOFM eigenveter centroid is regarded as one of the components in the state of HMMs. And this component forms the state pdf in term of the weighted sum. Then the state pdf can be calculated by the Forward-Backward Procedure(or by the Viterbi algorithm). SOFM weights are iteratively updated in the supervision of computed state pdfs. The parameters of SOFM and HMM are re-estimated through the Expectation-Maximization algorithm. In this way we combine the powerful self-organizing capability of SOFM with excellent temporal processing properties of HMM so that we improve the performance of HMM-based sign language recognition systems. The architecture of SOFM/HMM sees Figure 3.

4. Experiments and Comparisons

The data are collected from 7 signers with each performing 208 isolated signs 3 times. The vocabulary is words from elementary textbooks of 1-2 grades for Chinese deaf pupil. We at random select 5 from 7 signers, and then select 2 group data from each signer. In total 10 group data are regarded as the isolated sign language training set. The isolated sign language model parameters are trained through the SOFM/HMM method with the isolated sign language training set. The rest one group data in 5 signers are referred to as the registered test set (Reg.). The data from 2 signers are referred to as the unregistered test set (Unreg.). The results of isolated sign language recognition see Table 1. We select 2 from 5 signers in the isolated sign language training set, respectively represented with A, B, and select 1 from the rest 2 signers represented with C. There are 3 signers in total. The 100 continuous sentences composed of a vocabulary of 208 signs are respectively performed twice by 3 signers with the natural speed. There are 6 group data marked with A_1, A_2, B_1, B_2, C_1, C_2 .We choose A_1, B_1 as the training set which is used in the SOFM, SRN and embedded training. One of A_2, B_2 is referred to as the registered test set (Reg.), and one of C_1, C_2 is referred to as the unregistered test set (Unreg.). We compare the performances of SRN/HMM with those of HMM in signer-independent continuous sign language recognition and the results see Table 2 and Table 3.

Table 1. The isolated sign language recognition results

Signer		SOFM/HMM
Reg.	S_1	98.6%
	S_2	95.2%
	S_3	96.7%
	S_4	91.3%
	S_5	94.7%
	Mean	95.3%
Unreg.	S_6	88.5%
	S_7	88.0%
	Mean	88.2%

Table 2. The continuous sign language recognition results in Unreg.

Method	Recognition accuracy (%)
HMM	81.2 (S=35, I=25, D=9)
SRN/HMM	85.0 (S=33, I=6, D=16)

Table 3. The continuous sign language recognition results in Reg.

Method	Recognition accuracy (%)
HMM	90.7 (S=13, I=18, D=3)
SRN/HMM	92.1 (S=12, I=5, D=12)

All experiments are performed with the Bigram language model in the PIII450(192M Memory) PC. S, I and D denote respectively the error number of substitution, insertion and deletion. The total number of signs in the test set is 367. Compared with the standard HMM, the SRN/HMM has higher recognition accuracy. The possible reasons for the results are as follows. Firstly, the HMM uses the continuous sign language Viterbi algorithm which is liable to be influenced by the *movement epenthesis*, but the SRN/HMM alleviates the effects of *movement epenthesis* by discarding the transient frames near the sentences segmentation judged according to the output of SOFM. Secondly, unlike the HMM which searches the best state sequence, the SRN/HMM gets the best word sequence. Thirdly, the SRN/HMM employs the isolated Viterbi algorithm that can get the higher accuracy than the continuous Viterbi algorithm. However, the divide-and-conquer approach in the SRN/HMM may lead to the accumulation of errors. Thus we introduce the soft-segmentation instead of the fixed-segmentation in the SRN segmentation and decide the boundary of words in the Lattice Viterbi algorithm so as to improve the performance of SRN/HMM.

5. Conclusion

While the HMM employs the implicit word segmentation procedure, we present a novel divide-and-conquer approach which uses the explicit segmentation procedure for signer-independent continuous CSL recognition, and which, unlike the HMM, isn’t liable to the effects of *movement epenthesis*. The divide-and-conquer approach is as follows. Firstly, the improved SRN is introduced to segment the continuous sentences, and the results of segmentation are regarded as the state inputs of HMM. Secondly, the best word sequence is gotten through the search of Lattice Viterbi algorithm in the sentence segments. This approach alleviates the effects of *movement epenthesis*, and experimental results show that it increases the recognition accuracy. In addition, the improved SRN, the Lattice Viterbi algorithm and the segment property of continuous sign language found in the experiment are not only used in this approach but also provide the foundation for further researches.

Acknowledgment

This work has been supported by National Science Foundation of China (contract number 69789301), National Hi-Tech Development Program of China (contract number 863-306-ZD03-01-2 and 2001AA114160), and 100 Outstanding Scientist foundation of Chinese Academy of Sciences.

References

[1] T. Starner, A. Pentland, “Visual Recognition of American Sign Language Using Hidden Markov Models”, International Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, 1995, pp. 189-194.

[2] R.H. Liang, M. Ouhyoung, “A Real-time Continuous Gesture Recognition System for Sign Language”, In Proceeding of the Third International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998, pp. 558-565.

[3] C. Vogler, D. Metaxas, “ASL Recognition Based on a Coupling between HMMs and 3D Motion Analysis”, In Proceedings of the IEEE International Conference on Computer Vision, 1998, pp. 363-369.

[4] C. Vogler, D. Metaxas, “Toward Scalability in ASL Recognition: Breaking Down Signs into Phonemes”, In Proceedings of Gesture Workshop, Gif-sur-Yvette, France, 1999, pp. 400-404.

[5] W. Gao, J.Y. Ma et al, “HandTalker: A Multimodal Dialog System Using Sign Language and 3-D Virtual Human.”, Advances in Multimodal Interfaces-ICMI 2000, pp. 564-571.

[6] J.L. Elman, “Finding Structure in Time”, Cognitive Science, 1990, 14 (2), pp. 179-211.

[7] T. Robinson, “An Application of Recurrent Nets to Phone Probability Estimation”, IEEE Transactions on Neural Networks, 1994, 5(2), pp. 298-305.

[8] D.J. Kershaw, M. Hochberg, A.J. Robinson, “Context-Dependent Classes in a Hybrid Recurrent Network-HMM Speech Recognition System”, Advances in Neural Information Processing Systems, 1996, Vol. 8, pp. 750-756.

[9] A. Senior, A.J. Robinson, “Forward-Backward Retraining of Recurrent Neural Networks”, Advances in Neural Information Processing Systems, 1996, Vol. 8, pp. 743-749.

[10] K. Murakami, H. Taguchi, “Gesture Recognition using Recurrent Neural Networks”, In CHI’91 Human Factors in Computing Systems, 1991, pp. 237-242.

[11] D.E. Rumelhart, G. E. Hinton, R. J. Williams, “Learning Internal Representations by Error Propagation”, In Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, 1986, pp. 318-362.

[12] G.L. Fang, W. Gao, “A SOFM/HMM System for Person-Independent Isolated Sign Language Recognition”, INTERACT2001 Eight IFIP TC.13 Conference on Human-Computer Interaction, Tokyo, Japan, 2001, pp. 731-732.

IEEE COMPUTER SOCIETY