Recognizing Spatiotemporal Gestures and Movement Epenthesis in Sign Language

Daniel Kelly, John McDonald and Charles Markham
National University of Ireland, Maynooth
Ireland
dankelly@cs.nuim.ie

Abstract

A novel system for the recognition of spatiotemporal hand gestures used in sign language is presented. While recognition of valid sign sequences is an important task in the overall goal of machine recognition of sign language, recognition of movement epenthesis is an important step towards continuous recognition of natural sign language. We propose a framework for recognizing valid sign segments and identifying movement epenthesis. Experiments show our proposed system performs well when classifying eight different signs and identifying 100 different types of movement epenthesis. A ROC analysis of the systems classifications performance showed an area under the curve measurement of 0.949.

1 Introduction

Gestures are a form of non-verbal communication. Hand gestures can be classified into several categories such as conversational gestures, controlling gestures, manipulative gestures and communicative gestures [22]. In communicative hand gestures, sign language is often regarded as the most structured of the various gesture categories [6][12].

There have been many studies on human gestures, and on sign language in particular, in psycholinguistic research. Stokoe [17] identified the four building blocks of sign language; the hand shape, the position, the orientation and the movement. With these building blocks in mind hand gestures can be classified as either hand postures (hand shape and orientation) or spatiotemporal gestures (position and movement) [23]. The position of a hand refers to where the hand is placed relative to the body and hand movement traces out a trajectory in space.

One of the main difficulties with recognizing a gesture within a continuous sequence of gestures is that the hand(s) must move from the end point of the previous gesture to the start point of the next gesture. These inter gesture transition periods are called movement epenthesis [10] and are not part of either of the signs. As such, an accurate recognition system must be able to distinguish between valid sign segments and movement epenthesis. This work describes a framework for the recognition of spatiotemporal gestures and identification movement epenthesis.

1.1 Related Work

Extending isolated sign recognition to continuous signing requires automatic detection of movement epenthesis segments so that the recognition algorithm can be applied on the segmented signs.

One proposed solution to movement epenthesis detection is an explicit segmentation model were subsets, of features from gesture data, are used as cues for valid gesture start and end point detection [15, 9]. The limitation of this explicit segmentation model arises from the difficulty in creating general rules for sign boundary detection that could be applied to all types of gestures [13].

An approach to dealing with continuous recognition without explicit segmentation is to use Hidden Markov Models (HMM) for implicit sentence segmentation. Starner et al. [16] and Bauer and Kraiss [2] model each word or subunit with a HMM and then train the HMMs with data collected from full sentences. A downside to this is that training on full sentence data may result in a loss in valid sign recognition accuracy due to the large variations in the appearance of all the possible movement epenthesis that could occur between two signs.

Wang et al. [21] also use HMMs to recognize continuous signs sequences with 92.8% accuracy, although signs were assumed to end when no hand motion occurred. Assan et al. [1] model the HMMs such that all transitions go through a single state, while Gao et al. [4] create separate HMMs that model the transitions between each unique pair of signs that occur in sequence. Vogler at al. [20] also use an explicit

978-0-7695-3796-2/09 $26.00 © 2009 IEEE
DOI 10.1109/IMVIP.2009.33

epenthesis modeling system where one HMM is trained for every two valid combinations of signs.

While these works have had promising results in gesture recognition and movement epenthesis detection, the training of such systems involves a large amount of extra data collection, model training and recognition computation due to the extra number of HMMs required to detect movement epenthesis. In this work we propose a HMM based gesture recognition framework which accurately classifies a given gesture sequence as one of a number of pre trained gestures as well as calculating the probability that the given gesture sequence is or is not a movement epenthesis. The novelty of our work is the that the movement epenthesis detection is carried out by a single HMM and requires no extra data collection or training.

2 Feature Extraction

For completeness, we briefly describe the feature tracking techniques used, though we do not consider it to be the novel part of our work.

The sign recognition system described in this work is a computer vision based system. From the definition of a spatiotemporal gesture [17], we must track the position and movement of the hands in order to described a gesture sequence. We expand on the work of a hand posture recognition system proposed Kelly et al [5] to build a feature extraction system for spatiotemporal gesture recognition. Tracking of the hands is performed by tracking colored gloves using the Mean Shift algorithm [3].

Face and eye positions are also used as gestures cues. Face and eye detection is carried out using a cascade of boosted classifiers working with haar-like features proposed by Viola and Jones [18]. A set of public domain classifiers [11],for the face, left eye and right eye, are used in conjunction with the OpenCV implementation of the haar cascade object detection algorithm.

We define the raw features extracted from each image as follows; right hand position (RH_x, RH_y), left hand position (LH_x, LH_y), face position (FC_x, FC_y), face width (FW), left eye position (LE_x, LE_y) and right eye position (RE_x, RE_y).

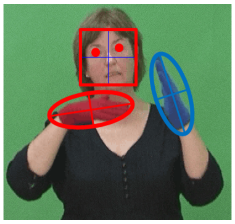


Figure 1. Extracted Features from Image

3 Hidden Markov Models

Hidden Markov Models (HMMs) are a type of statistical model and can model spatiotemporal information in a natural way. HMMs have efficient algorithms for learning and recognition, such as the Baum-Welch algorithm and Viterbi search algorithm [14].

A HMM is a collection of states connected by transitions. Each transition (or time step) has a pair of probabilities: a transition probability (the probability of taking a particular transition to a particular state) and an output probability (the probability of emitting a particular output symbol from a given state).

We use the compact notation $\lambda = \{A, B, \pi\}$ to indicate the complete parameter set of the model where A is a matrix storing transitions probabilities a_{ij} between states s_i and s_j, B is a matrix storing output probabilities for each state and π is a vector storing initial state probabilities.

HMMs can use either a set of discrete observation symbols or they can be extended for continuous observations signals. In this work we use continuous multidimensional observation probabilities calculated from a multivariate probability density function.

3.1 HMM Threshold Model

Lee and Kim [8] proposed a HMM threshold model to handle non-gesture patterns. The threshold model was implemented to calculate the likelihood threshold of an input pattern and provide a confirmation mechanism for provisionally matched gesture patterns. We build on this work carried out by Lee and Kim to create a framework for calculating a probability distribution of a two hand input sign using continuous multidimensional observations. The computed probability distribution will include probability estimates for each pre-trained sign as well as a probability estimate that the input sign is a movement epenthesis.

In general, a HMM recognition system will choose a model with the best likelihood as the recognized gesture if the likelihood is higher than a predefined threshold. However, this simple likelihood threshold often does not work, thus, Lee and Kim proposed a dynamic threshold model to define the threshold of a given gesture sequence.

A property of the left-right HMM model implies that a self transition of a state represents a particular segment of a target gesture and the outgoing state transition represents a sequential progression of the segments within a gesture sequence. With this property in mind, an ergodic model, with the states copied from all gesture models in the system, can be constructed as shown in Figure 2 and 3, where dotted lines in Figure 3 denote null transitions (i.e. no observations occur between transitions).

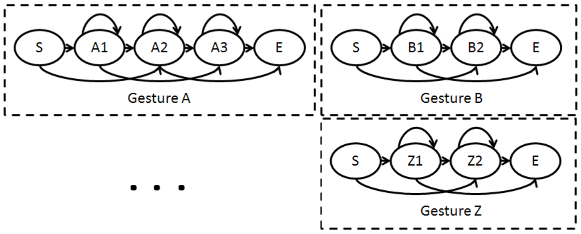


Figure 2. Dedicated Gesture Models

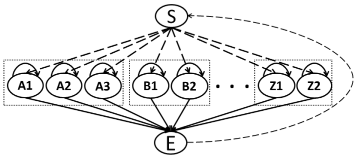


Figure 3. Threshold Model

States are copied such that output observation probabilities and self transition probabilities are kept the same, but all outgoing transition probabilities are equally assigned as defined in Equation 1 where N is the number of states excluding the start and end states (The start and end states produce no observations).

$$a_{ij} = \frac{1 - a_{ij}}{N - 1}, \quad \forall j, i \neq j, \tag{1}$$

As each state represents a subpattern of a pre-trained gesture, constructing the threshold model as an ergodic structure makes it match well with all patterns generated by combining any of the gesture sub-patterns in any order. The likelihood of the threshold model, given a valid gesture pattern, would be smaller than that of the dedicated gesture model because of the reduced outgoing transition probabilities. However, the likelihood of the threshold model, given an arbitrary combination of gesture sub-patterns, would be higher than that of any of the gesture models, thus the threshold model, denoted as $\overline{\lambda}$, can be used as a movement epenthesis likelihood measure.

4 System Overview

Our system initializes and trains a dedicated parallel HMM [19] for each gesture to be recognized. Each parallel HMM consists of two separate HMMs that model the right and left hand gesture respectively. A description of the models observations, training and recognition process will now be carried out.

4.1 Feature Processing

A spatiotemporal gesture is defined by the hands' position and movement, where the position refers to the hands' location relative to the body and movement traces out a trajectory in space. With this definition in mind, the position of the hands must be represented in a feature vector that describes the position relative to the body. Using the raw features, extracted from the image using the method described in Section 2, the observation vector we use to model a gesture is comprised of a combination of features calculated from the raw features. We carry out performance evaluations on a number different feature combinations in order to find features which best classify spatiotemporal features and movement epenthesis. These evaluations will be discussed in Section 5.

To represent a gesture sequence such that it can be modeled by a HMM, the gesture sequence must be defined as a set of observations. An observation O_t, is defined as an observation vector made at time t, where $O_t = \{o_1, o_2, ..., o_M\}$ and M is the dimension of the observation vector. A particular gesture sequence is then defined as $\Theta = \{O_1, O_2, ..., O_T\}$.

To calculate the probability of a specific observation O_t, a probability density function of an M-dimensional multivariate gaussian is implemented (see Equation 2). Where μ is the mean vector and Σ is the covariance matrix.

$$\aleph(O_t; \mu, \Sigma) = (2\pi)^{-\frac{N}{2}} |\Sigma|^{-\frac{1}{2}} exp\big(-\tfrac{1}{2}(O_t - \mu)^T \Sigma^{-1} (O_t - \mu)\big) \tag{2}$$

4.2 Model Training

Each dedicated gesture model is trained on isolated signs performed by a fluent signer. Before training a HMM using the Baum-Welch algorithm, the model must first be initialized. Initialization includes the computation of an initial state transition matrix and calculation of each states' emission variables μ and Σ. In order to initialize these components of the HMM, an understanding of the gesture segmentation, or state transitions, must be built. One approach to achieving this would be to explicitly hand label different subunits or gesture phonemes [21]. Part of the goal of this work is to create a general data collection, training and recognition system. Data collection consists of a recording step and a labeling step. Labeling is an integral step in creating valid sign data, thus we envisage that all data will be labeled by fluent signers.

Since movement and position of the hands are two of the four building blocks of sign language which Stokoe [17] identified, manually breaking these building blocks into smaller subunits would be an un-intuitive and time consuming step for fluent signers to segment in a consistent manner. With this in mind, a training system was developed to initialize and train data with minimum human intervention where signs are labeled at a sign level and not at a phoneme level.

We implement an automated HMM initialization and training model in our system. We extend an iterative HMM training model proposed by Kim at al [7] to develop a HMM initialization and training model which includes an extra parameter selection layer. The parameter selection layer finds the best combination of (S, R), where S is the total number of states in the HMM and R is the reach of a state (i.e. in a left-right model, the reach is the number of states that it is possible to transition to from the current state).

For a particular sign, we collect data from a number of video sequences of a fluent signer performing that sign. This produces a set of observation sequences $\Delta_c = \{\Theta_c^1, \Theta_c^2, ..., \Theta_c^K\}$ where c is the index of the sign being modeled and K is the total number of training examples.

To initialize λ_c, the HMM which will model the sign indexed by c, we first choose a random gesture sequence Θ_c^r from Δ_c and calculate $S-1$ indices of Θ_c^r which best segment the gesture into S sub-gestures. The $S-1$ indices are calculated by performing principal component analysis on the gesture sequence, performing a k-means clustering technique on the principal components and finally finding the $S-1$ indices which best divide the data into their corresponding k-means clusters.

The gesture data is then broken into the S subsets and the mean vector μ and the covariance matrix Σ is calculated for each state. The Baum-Welch algorithm[14] is then applied to λ_c using all training data Δ_c. After training, the Viterbi algorithm[14] is run on Θ_c^r to produce most probable state sequence. The initial S sub-gestures are then realigned to match the Viterbi path. This re-estimation and realignment process is continued until the likelihood, produced by the Baum-Welch algorithm, converges. The overall process is repeated for different combinations of (S, R) to find the combination which produces the highest likelihood from the Baum-Welch re-estimation. Figure 4.2 gives an overview of the iterative training and parameter selection procedure.

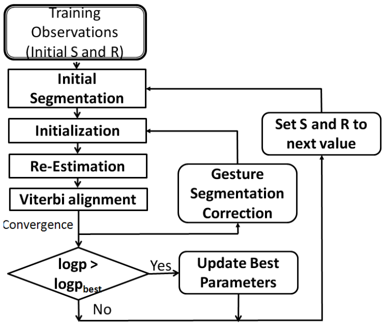


Figure 4. HMM Initialization and Training Procedure

It is desirable to weight λ_{Lc} and λ_{Rc}, the left hand HMM and right hand HMM respectively, due to variations in information held in each of the hands for a particular sign. The weighting applied in our system is based on a variance measure of the observation sequences. Using data from all observation sequences Θ_{Lc}^k and Θ_{Rc}^k, where $1 \leq k \leq K$, K is the total number of training examples and Θ_{Lc} and Θ_{Rc} are the left and right hand observations respectively. The variance of the left and right hand observations are calculated by calculating the variance of each observation dimension $\sigma_{Lc}^2[i]$ and $\sigma_{Rc}^2[i]$, where $0 \leq i \leq D$ and D is the dimension of the observation vectors. The left HMM weight, ω_{Lc}, and right HMM weight, ω_{Rc}, are then calculated as using Equation 3.

$$\omega_{Lc} = \sum_{i=0}^{D} \frac{\sigma_{Lc}^2[i]}{(\sigma_{Lc}^2[i] + \sigma_{Rc}^2[i]) \times D} \quad \omega_{Rc} = \sum_{i=0}^{D} \frac{\sigma_{Rc}^2[i]}{(\sigma_{Lc}^2[i] + \sigma_{Rc}^2[i]) \times D} \tag{3}$$

4.3 Sign Recognition

The set of parallel HMMs, to recognize the C pre-trained signs, is denoted as $\Lambda_L = \{\lambda_{L1}, \lambda_{L2}, ..., \lambda_{LC}, \overline{\lambda_L}\}$ and $\Lambda_R = \{\lambda_{R1}, \lambda_{R2}, ..., \lambda_{RC}, \overline{\lambda_R}\}$.

Given an unknown sequence of sign observations Θ_L and Θ_R, the goal is to accurately classify the sign as either a epenthesis sign or as one of the C trained signs. To classify the observations, the Viterbi algorithm is run on each model given the unknown observation sequence, calculating the most likely state path through that model and the likelihood of that state path.

We calculate the overall likelihoods of a dedicated gesture and a movement epenthesis with the equations defined in Equations 4 and 5 respectively.

$$P(\Theta|\lambda_c) = P(\Theta_L|\lambda_{Lc})\omega_{Lc} + P(\Theta_R|\lambda_{Rc})\omega_{Rc} \tag{4}$$

$$\Psi_c = \frac{P(\Theta_L|\overline{\lambda_L})\Gamma_{Lc} + P(\Theta_R|\overline{\lambda_R})\Gamma_{Rc}}{2} \tag{5}$$

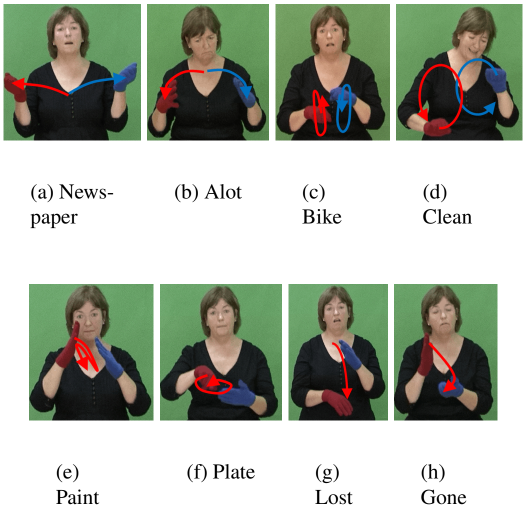
Where Γ_{Lc} and Γ_{Rc} are constant scalar values used to tune the sensitivity of the system to movement epenthesis.

The sequence of observations can then be classified as c if $P(\Theta|\lambda_c) \geq \Psi_c$ evaluates to be true.

5 Experiments

In this paper we describe a system for the recognition of spatiotemporal signs and the identification of movement epenthesis. Since the success of recognizing continuous gestures greatly depends on the discrimination power of the gesture models and the threshold model, we carry out an isolated gesture recognition experiment

To evaluate the performance of our recognition framework, a set of eight different signs, as performed by a fluent signer, were recorded and manually labeled. A visual example of a signer performing each of the eight signs is shown in Figure 5. All signs were performed naturally within full sign language sentences and labeled by a certified sign language interpreter. The set of eight test signs were not selected to be visually distinct but to represent a suitable cross section of the spatiotemporal signs that can occur in sign language.



(a) Newspaper (b) Alot (c) Bike (d) Clean

(e) Paint (f) Plate (g) Lost (h) Gone

Figure 5. Example of the eight different signs the system was tested on

Observation sequences Δ_c were extracted from the video sequence (where $1 \leq c \leq C$) and divided into a training set, Δ_c^τ, and a test set, Δ_c^ζ. For the experiments we report in this paper, a set of 5 training signs and a set of 5 test signs were recorded for each sign. Each dedicated gesture model λ_c was then trained on Δ_c^τ using our training procedure described in Section 4.2. The threshold models were then created using the trained gesture models.

An additional set of observations Δ_E, which represent a collection movement epenthesis, were also extracted from the video sequences to test the performance of the threshold model. To sufficiently test the performance of our system when discriminating between valid signs and movement epenthesis, the number of movement epenthesis, to evaluate the system on, should be sufficiently larger than the set of valid signs due to the numerous possible movement epenthesis that can occur between two valid signs. For each valid sign, we recorded 10 movement epenthesis that occurred before and after the valid sign in different sign language sentences. An additional set of 20 random movement epenthesis were also recorded, resulting is a test set of 100 samples to evaluate the system on.

The classification of a gesture is based on a comparison of a weighted threshold model likelihood with the weight denoted as Γ_c. In our ROC analysis of the system, we vary the weight, Γ_c, over the range $0 \leq \Gamma_c \leq 1$ and then create a confusion matrix for each of the weights. This procedure is caried out for both the left hand weights, Γ_{Lc}, and the right hand weights, Γ_{Rc}.

To evaluate the performance of different features, we performed a ROC analysis on the models generated from the different feature combinations and calculated the area under the curve (AUC) for each feature vector model as shown in Table 1.

Table 1. AUC Measurements for Different Feature Combinations

Features	ROC AUC
F_1 - Hand Direction (V_x, V_y)	0.8614
F_2 - Hand Direction $(V_x, V_y)+$ Distance Between Hands (D_H)	0.698
F_3 - Hand Direction $(V_x, V_y)+$ Distance Between Eyes and Hand (D_E)	0.7391
F_4 - Hand Positions Relative to Eyes (RP_x, RP_y)	0.789
F_5 - Hand Positions Relative to Eyes (RP_x, RP_y) + Distance Between hands (D_H)	0.936
F_6 - Hand Positions Relative to Eyes (RP_x, RP_y) + Hand Direction (V_x, V_y)	0.807
F_7 **- Hand Positions Relative** **to Eyes** (RP_x, RP_y) **+** **Hand Direction** (V_x, V_y) **+** **Distance Between hands** (D_H)	**0.949**

It can be seen from the AUC measurements shown in Table 1 that the best performing feature, with an AUC of 0.949, was the feature, $F_7 = \{RP_x, RP_y, V_x, V_y, D_H\}$, which describes the position of the hands relative to the eyes, the direction of the movement of the hand and the distance between the two hands.

To evaluate the performance of the threshold model, when applied to continuous multi dimensional sign language observations, we compare the performance of our system to a modified version of our system with no threshold model. The modified version of the system uses the same dedicated HMMs but the sequence of observations is classified as c only if the gesture likelihood is greater than a predefined static threshold. A ROC analysis of the modified systems classifications showed that the best performing feature was also the feature F_7. The AUC of the ROC graph produced by this feature was 0.897. From the experiments we have carried out, the performance of the system with the threshold model was 5.2% better than that of the system without the threshold model.

6 Conclusion

This work described a novel framework for classifying spatiotemporal signs and identifying movement epenthesis. We have shown the need for a system that can identify movement epenthesis. Previous attempts to model movement epenthesis required explicit modeling of every possible sequence. The novelty of this system is that we have expanded on the work of Lee and Kim [8] to develop a

HMM threshold model system which models continuous multidimensional sign language observations within a parallel HMM network to recognize two hand signs and identify movement epenthesis. The dedicated gesture models are the only models requiring explicit training and optimal dedicated gesture models are trained through our iterative training procedure were the only human intervention required is a labeling process where fluent signers label valid sign segments.

A threshold model, that can discriminate between valid signs and movement epenthesis, can be created using state and observation information taken directly from the dedicated gesture models. This is a desirable feature because as the sign vocabulary grows, the number of possible movement epenthesis that could occur between valid signs would grow to a number that would make explicitly modeling these sequences unfeasible.

The system was evaluated on how well it classified valid signs correctly, but also how well it could discriminate between valid sign sequences and 100 different types of movement epenthesis. A ROC analysis of the classification performance showed that the three dimensional feature vector F_7, defined in Table 1, was the best performing feature with an AUC measurement of 0.949.

6.1 Future Work

Future work will include implementing sign end point flagging and evaluating the performance of our system using online data. We will also investigate methods of integrating hand posture and non manual information into the recognition process.

7 Acknowledgements

The Authors would like to acknowledge the financial support of the Irish Research Council for Science, Engineering and Technology (IRCSET).

References

[1] M. Assan and K. Grobel. Video-based sign language recognition using hidden markov models. In *Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, pages 97–109, London, UK, 1998. Springer-Verlag.

[2] B. Bauer and K.-F. Kraiss. Towards an automatic sign language recognition system using subunits. In *GW '01: Revised Papers from the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction*, pages 64–75, London, UK, 2002. Springer-Verlag.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 2:142–149 vol.2, 2000.

[4] W. Gao, G. Fang, D. Zhao, and Y. Chen. Transition movement models for large vocabulary continuous sign language recognition. *IEEE FG 2004*, pages 553–558, May 2004.

[5] D. Kelly, J. McDonald, T. Lysaght, and C. Markham. Analysis of sign language gestures using size functions and principal component analysis. In *IMVIP 2008*, 2008.

[6] A. Kendon. How gestures can become like words. *Cross Perspectives in Nonverbal Comm.*

[7] Y.-J. Kim and A. Conkie. Automatic segmentation combining an hmm-based approach and spectral boundary correction. In *In ICSLP-2002*, pages 145–148, 2002.

[8] H. K. Lee and J. H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE PAMI*, 21(10):961–973, 1999.

[9] R. H. Liang and M. Ouhyoung. A real-time continuous gesture recognition system for sign language. In *IEEE FG 1998*, page 558, Washington, DC, USA, 1998. IEEE Computer Society.

[10] J. R. Liddell, S.K. American sign language: The phonological base. *Sign Langauge Studies*, 64.

[11] L. A.-C. M. Castrillon-Santana, O. Deniz-Suarez and J. Lorenzo-Navarro. Performance evaluation of public domain haar detectors for face and facial feature detection. *VISAPP 2008*, 2008.

[12] D. McNeill. *Hand and Mind: What Gestures Reveal about Thought*. Univ. of Chicago Press, 1992.

[13] S. C. W. Ong and S. Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(6):873–891, 2005.

[14] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.

[15] H. Sagawa and M. Takeuchi. A method for recognizing a sequence of sign language words represented in a japanese sign language sentence. In *IEEE FG 2000*, page 434, Washington, DC, USA, 2000. IEEE Computer Society.

[16] T. Starner, A. Pentland, and J. Weaver. Real-time american sign language recognition using desk and wearable computer based video. *IEEE PAMI*, 20(12):1371–1375, 1998.

[17] J. Stokoe, William C. Sign language structure: An outline of the visual communication systems of the american deaf. *Journal of Deaf Studies and Deaf Education, v10 n1 p3-37 Win 2005*, 2005.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *CVPR, IEEE*, 1:511, 2001.

[19] C. Vogler and D. Metaxas. Parallel hidden markov models for american sign language recognition. In *In ICCV*, pages 116–122, 1999.

[20] C. Vogler and D. Metaxas. A framework for recognizing the simultaneous aspects of american sign language. *Computer Vision and Image Understanding*, 81:358–384, 2001.

[21] C. Wang, S. Shan, and W. Gao. An approach based on phonemes to large vocabulary chinese sign language recognition. In *IEEE FG 2002*, page 411, Washington, DC, USA, 2002. IEEE Computer Society.

[22] Y. Wu and T. Huang. Human hand modeling, analysis and animation in the context of hci, 1999.

[23] Y. Wu, T. S. Huang, and N. Mathews. Vision-based gesture recognition: A review. In *Lecture Notes in Computer Science*, pages 103–115. Springer, 1999.