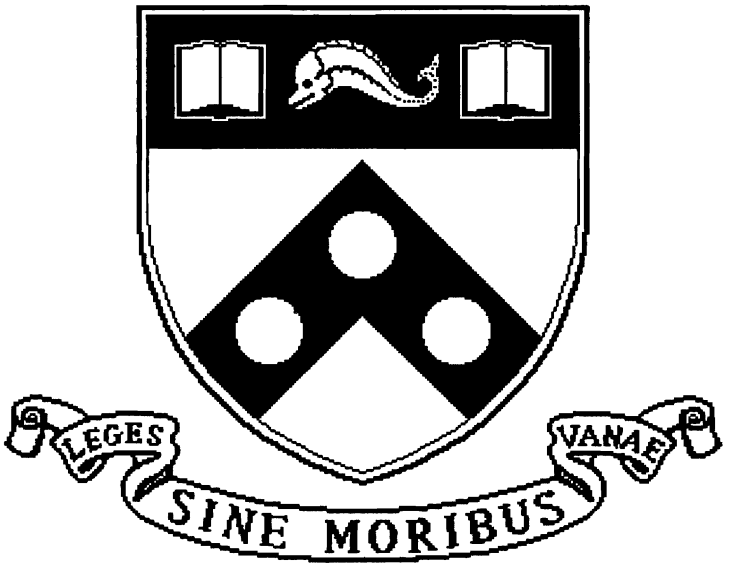
ASL Recognition Based on a Coupling Between HMMs and 3D Motion Analysis

MS-CIS-98-21

Christian Vogler and Dimitris Metaxas



University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

1998

An earlier version of this paper appeared in the proceedings of the International Conference on Computer Vision, pp. 363–369, Mumbai, India, January 4–7, 1998

ASL Recognition Based on a Coupling Between HMMs and 3D Motion Analysis

Christian Vogler and Dimitris Metaxas
Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104-6389
cvogler@gradient.cis.upenn.edu, dnm@central.cis.upenn.edu

Abstract

We present a framework for recognizing isolated and continuous American Sign Language (ASL) sentences from three-dimensional data. The data are obtained by using physics-based three-dimensional tracking methods and then presented as input to Hidden Markov Models (HMMs) for recognition. To improve recognition performance, we model context-dependent HMMs and present a novel method of coupling three-dimensional computer vision methods and HMMs by temporally segmenting the data stream with vision methods. We then use the geometric properties of the segments to constrain the HMM framework for recognition. We show in experiments with a 53 sign vocabulary that three-dimensional features outperform two-dimensional features in recognition performance. Furthermore, we demonstrate that context-dependent modeling and the coupling of vision methods and HMMs improve the accuracy of continuous ASL recognition.

1 Introduction

American Sign Language (ASL) is the primary mode of communication for many deaf people in the USA. It is a highly inflected language with sophisticated grammatical properties, which constrain strongly the order and appearance of signs. Because of the constraints, it provides an appealing test bed for understanding more general principles governing human motion and gesturing, including human-computer gesture interfaces. Such interfaces are essential in virtual reality applications, where the user must be able to manipulate virtual objects by gesturing. A working ASL recognition system could also facilitate interaction of deaf people with their surroundings.

To date, most attempts at ASL recognition have either used only two-dimensional computer vision methods, or they have used other input devices, such as datagloves, instead of computer vision, to collect input from the signer [18, 3, 23]. In this paper we present a new approach to ASL recognition. First, we use computer vision methods to extract the three-dimensional parameters of a signer's arm motions. We then use Hidden Markov Models (HMMs) to recognize isolated and continuous ASL utterances from the three-dimensional input. We develop context-dependent modeling of HMMs and methods for coupling the application of HMMs and the application of three-dimensional computer vision methods to improve continuous recognition performance. Our approach attempts to overcome some of the limitations of the previous approaches that use two-dimensional visual input, do not use context-dependent modeling, or do not couple computer vision methods with HMMs [18, 3, 17, 12].

Three-dimensional image-based shape and motion tracking of a human's arm and hand is difficult because of the complexity of the motions and occlusion effects. Recently, a methodology has been developed [8, 10] that allows three-dimensional tracking of human motion from multiple images. In this paper we augment this methodology to track the three-dimensional motion of a subject's arms and hands from multiple images. This method is based on the use of deformable models, whose shape and motion fits the given image sequences based on occluding contour information and theorems from projective geometry. The output of this method consists of the three-dimensional motion parameters of the subject's arms. For efficiency reasons, and because arm movements already carry much of the information needed for recognizing ASL signs, we do not use the hand information in this paper.

Apart from obtaining accurate data, ASL recognition is difficult, because there are always statistical variations in the way humans perform motions, even with identical meaning. In addition, in continuous utterances, there are no clear boundaries between individual signs. HMMs provide a framework for capturing statistical variations in both position and duration of the movement, as well as implicit segmentation of the input stream. Furthermore, continuous recognition is complicated by coarticulation effects, that is, the pronunciation[1] of a sign is influenced by the preceding and following signs. Coarticulation effects can be partly alleviated by training context-dependent HMMs.

The theory behind HMMs makes several assumptions that are often not valid in practice. For this reason, we develop a new approach that couples computer vision methods with HMM modeling. It is based on a temporal segmentation process that operates by extracting geometric properties of the three-dimensional computer vision pa-

[1] By "pronunciation" we mean motion. We follow the terminology of spoken language linguistics where applicable.

rameters. These properties are obtained independently from the HMM algorithms and are used to impose additional constraints on HMM-based recognition.

To test our algorithms and assumptions, we performed a series of experiments based on a vocabulary consisting of 53 different signs that make extensive use of space. We experimented with both isolated and continuous ASL recognition for both three-dimensional and two-dimensional data. As HMMs require large amounts of training data and the computer vision process is computationally expensive, we used data from an Ascension Technologies Flock of Birds and computer vision processes interchangeably.

Our goal is to discover and analyze a usable framework for both isolated and particularly continuous ASL recognition. We do not address more general gesture recognition topics and signer independence in this paper. Neither do we address the involved aspects of ASL linguistics [19] at this point, but obviously, a viable future ASL recognition system should be able to handle them.

In the following sections, we discuss related work and give an overview on the theory behind the vision methods and HMMs. Afterward, we address the use of HMMs for isolated and continuous ASL recognition, and coupling computer vision processes with the HMM algorithms. Finally, we outline data collection and provide experimentation results for isolated and continuous recognition and the coupling of computer vision and HMMs.

2 Previous Work

Previous work on sign language recognition focuses primarily on fingerspelling recognition and isolated sign recognition. Some work uses neural networks [3, 22]. For this work to apply to continuous ASL recognition, the problem of explicit temporal segmentation must be solved, which is a limitation that HMM-based recognition does not have. Mohammed Waleed Kadous [23] uses Power Gloves to recognize a set of 95 isolated Auslan signs with 80% accuracy, with an emphasis on computationally inexpensive methods. Kirsti Grobel and Marcell Assam [4] use HMMs to recognize isolated signs with 91.3% accuracy out of a 262 sign vocabulary. They extract the features from video recordings of signers wearing colored gloves.

There is very little previous work on continuous ASL recognition. Thad Starner and Alex Pentland [18] use a view-based approach to extract two-dimensional features as input to HMMs with a 40 word vocabulary. Yanghee Nam and Kwang Yoen Wohn [12] use three-dimensional data as input to HMMs for continuous recognition of a very small set of gestures.

3 Model-based Tracking of a Human's Arms

In this section we give a brief overview of our formulation that allows the three-dimensional arm shape and motion estimation from multiple images [6, 7, 8, 10].

Our approach consists of two parts. The first part [6, 7] consists of an active, integrated approach that identifies reliably the parts of a moving articulated object and estimates their shape and motion from a *controlled set* of motions that reveal the object's structure. We use the algorithm developed in [6, 7], which segments the apparent body contour of a moving human into the constituent parts. Initially, a single deformable model is used in order to fit the image data. As the model deforms to fit the deformed (due to the motion of the human) subsequent image contours, a novel **Human Body Part Identification Algorithm** (HBPIA) is developed to identify all the body parts. By applying the HBPIA iteratively over the subsequent frames, all the moving parts are identified. In addition, we have extended this algorithm to allow the estimation of the three-dimensional shape of a subject's body parts, based on the integration of images taken from three orthogonally placed cameras. We used this methodology to estimate the three-dimensional shape of the subject's arms shown in the examples in Section 7. It is worth noting that we have recovered the lower arm and the hand as one part, since in our ASL recognition experiments we did not use the motion of the lower arm and the hand relative to each other.

The second part of the algorithm consists of using the extracted three-dimensional shape of the arm to track the three-dimensional position and orientation of a subject's body parts [8]. To alleviate difficulties arising from occlusion and degenerate views during the unconstrained movement of the arm, we use three calibrated cameras placed in a mutually orthogonal configuration. At every image frame and for each body part, we derive a subset of the cameras that provide the most informative views for tracking. This *active* and time varying selection is based on the visibility of a part and the observability of its predicted motion from a certain camera. Once a set of cameras has been selected to track each part, we use concepts from projective geometry to relate points on the occluding contour to points on the three-dimensional shape model. Using a physics-based modeling approach, we transform this correspondence, in addition to two-dimensional forces arising from the discrepancy between the model's occluding contour and the image data, into generalized forces that are applied to the model to estimate the model's translational and rotational degrees of freedom. To improve the tracking results further, the dynamic system is embedded within an extended Kalman filter framework, and we use the *predicted* motion of the model at each frame to establish point correspondences between occluding contours and the three-dimensional model.

We used this two-step approach to track the motion of the subject's arms performing the ASL gestures, as shown

in Section 7. The output of the system is a set of rotation, $\mathbf{q}_\theta$, and translation, $\mathbf{q}_c$, parameters that we use as input to the HMMs and the vision-based segmentation algorithm presented in the following sections.

4 Hidden Markov Models

Hidden Markov Models (HMMs) are a type of statistical model. They have been used successfully in speech recognition, and recently in handwriting, gesture, and sign language recognition. We now give a summary of the basic theory behind HMMs, which is covered in detail in [15].

4.1 Definition of HMMs

An HMM consists of a number N of states $S_1, S_2, \ldots, S_N$, together with transitions between states. The system is in one of the HMM's states at any given time. At regularly spaced discrete time intervals, the system takes an outgoing transition from its current state to a new state.

Each transition from S_i to S_j has an associated probability a_{ij} of being taken. Hence, $\sum_i a_{ij} = 1$. Each state S_i also has an initial probability π_i of the system starting in S_i. In addition, each state S_i generates output $k \in \Omega$, which is distributed according to a probability distribution function $b_i(k) = P\{\text{Output is } k | \text{System is in } S_i\}$. An example is given in Figure 1. The model depicted there is also an example of a **left-right** model; that is, $a_{ij} > 0$ implies $j \geq i$. In other words, transitions only flow forward from lower states to the same state or higher states, but never backward. This topology is the most commonly used one for modeling processes over time.

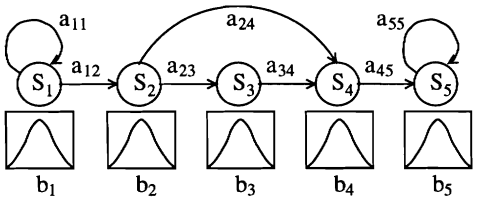


Figure 1: Example left-right HMM with its transition and output probabilities. "Left-right" means that transitions occur only from left to right, and never backward.

4.2 The Three Fundamental HMM Problems

There are three fundamental problems in HMM theory:

(1) For a sequence of observations $O = O_1, \ldots, O_T$, $O_i \in \Omega$, compute the probability $P(O|\lambda)$ that an HMM λ generated O.

(2) For some O and an HMM λ, recover the most likely state sequence $S_1, \ldots, S_T$ that generated O.

(3) Adjust the parameters of an HMM λ such that they maximize $P(O|\lambda)$ for some O.

The first problem corresponds to maximum likelihood recognition of an unknown data sequence with a set of HMMs, each of which corresponds to a sign. For each HMM, the probability $P(O|\lambda)$ is computed that it generated the unknown sequence, and then the HMM with the highest probability is selected as the recognized sign. For computing $P(O|\lambda)$, let $Q = Q_1, Q_2, \ldots, Q_T$ be a state sequence in λ:

$$\alpha_t(i) = P(O_1, O_2, \ldots, O_t, Q_t = S_i|\lambda) \quad 1 \leq i \leq N, \quad (1)$$

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i), \quad (2)$$

$$\alpha_1(i) = \pi_i b_i(O_1), \quad (3)$$

$$\alpha_{t+1}(i) = b_i(O_{t+1}) \sum_{j=1}^{N} \alpha_t(j) a_{ji} \quad 1 \leq t \leq T-1 \quad (4)$$

These equations assume that the O_i are independent, and they make the Markov assumption that a transition depends only on the current state, a fundamental limitation of HMMs. This method is called the forward-backward algorithm and computes $P(O|\lambda)$ in $O(N^2T)$ time.

The second problem corresponds to finding the most likely path Q through an HMM λ, given an observation sequence O, and is equivalent to maximizing $P(Q, O|\lambda)$. Let

$$\delta_t(i) = \max_{Q_1, \ldots, Q_{t-1}} P(Q_1 Q_2 \ldots Q_t = S_i, O|\lambda), \quad (5)$$

$$\delta_{t+1}(i) = b_i(O_{t+1}) \cdot \max_{1 \leq j \leq N} \{\delta_t(j) a_{ji}\}, \quad (6)$$

$$\max_Q P(Q, O|\lambda) = \max_{1 \leq i \leq N} \{\delta_T(i)\}. \quad (7)$$

$\delta_t(i)$ corresponds to the maximum probability of all state sequences that end up in S_i at time t. Equations 6 and 7 follow from Equation 5 by induction on t. The Viterbi algorithm is a dynamic programming algorithm that, using Equation 7, computes both the maximum probability $P(Q, O|\lambda)$ and the state sequence Q in $O(N^2T)$ time.

The recovery of the state sequence makes the Viterbi algorithm invaluable for continuous recognition, since it bypasses the difficult problem of segmenting the utterances into its individual parts. Instead, a sequence of HMMs corresponding to individual signs is concatenated into a network, as schematically depicted in Figure 2. Thus, the most likely state sequence recovers the sequence of signs.

The Viterbi algorithm also has the property that it can be optimized with the beam-searching algorithm. While updating $\delta_{t+1}(i)$, this optimization considers only those states S_j in the HMM network for which $\delta_t(j)$ is above a threshold value. The assumption is that if the probability

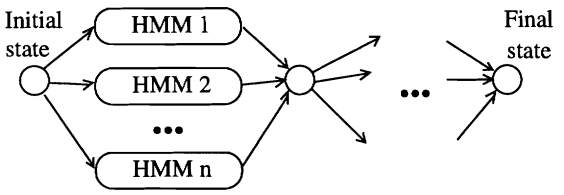


Figure 2: Concatenation of HMMs into a network

of a partial path through the network becomes too low, it cannot contribute to the most likely path. Beam-searching is essential for making large-scale applications tractable.

The third problem corresponds to training the HMMs with data, such that they are able to recognize previously unseen data correctly after the training phase. There exists no analytical solution for maximizing $P(O|\lambda)$ for given observation sequences, but an iterative procedure, called the Baum-Welch procedure, maximizes $P(O|\lambda)$ locally. In the case of continuous density output probabilities, the reestimation process works as follows.

Define $b_j(O)$ as $b_j(O) = \sum_{m=1}^{M} c_{jm} G(O, \mu_{jm}, U_{jm})$, where M describes the number of mixtures, j is the state number, c describes the weight of mixture m in state j, and G is a Gaussian density with mean μ, and covariance matrix U. Define the backward variable β as

$$\beta_t(i) = P(O_{t+1}O_{t+2}, \ldots, O_T | Q_t = S_i, \lambda), \quad (8)$$

$$\beta_T(i) = 1, \quad (9)$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad (10)$$

$$1 \leq i \leq N,\ 1 \leq t \leq T-1. \quad (11)$$

Furthermore, define ξ and γ as

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}, \quad (12)$$

$$\gamma_t(i) = \sum_{i=1}^{N} \xi_t(i,j). \quad (13)$$

$\sum_t \xi_t(i,j)$ can be interpreted as the expected number of transitions from S_i to S_j; likewise $\sum_t \gamma_t(i)$ can be interpreted as the expected number of transitions taken from S_i. With these interpretations, the reestimation formulae for the transitions and output probabilities are

$$\bar{\pi}_i = \gamma_1(i), \quad (14)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad (15)$$

$$\bar{c}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)}, \quad (16)$$

$$\bar{\mu}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m) O_t}{\sum_{t=1}^{T} \gamma_t(j,m)}, \quad (17)$$

$$\bar{U}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m)(O_t - \mu_{jm})(O_t - \mu_{jm})^T}{\sum_{t=1}^{T} \gamma_t(j,m)}. \quad (18)$$

Repeated use of this procedure converges to a maximum probability [15], typically after 5–10 iterations.

5 Use of HMMs for ASL Recognition

In the previous section we reviewed the extraction of three-dimensional features from computer vision and the HMM theory. We now discuss how they fit in the framework of ASL recognition.

HMMs are an attractive choice for processing three-dimensional sign data, because their state-based nature enables them to describe how a sign changes over time and to capture variations in the duration of signs, by remaining in a state for several time frames.

There are two ways to approach the recognition problem that pose very different research problems. Isolated recognition attempts to recognize one single sign at a time. Hence, it is based on the assumption that each sign can be individually extracted and then individually recognized.

Continuous recognition, on the other hand, attempts to recognize an entire stream of signs, without any artificial pauses or any other form of marked boundaries between the individual signs. Clearly, continuous recognition is desirable for the most natural interaction possible between humans and machines, but it is also much more difficult to tackle than isolated recognition. The next two subsections discuss each of the two approaches in detail.

5.1 Isolated Recognition

Isolated sign recognition assumes that each sign can be extracted individually. This requires clearly marked boundaries between signs. Such a boundary could simply be silence, that is, a brief resting phase after each sign, during which the signer performs no movements. Silence is easily detected through an analysis of the global variance over the hand movements.

Once there are clearly marked boundaries between signs, HMM recognition is comparatively straightforward. The recognition process extracts the signal corresponding to each sign individually. It then picks the HMM that yields the maximum likelihood for that signal as the recognized sign.

Training the HMMs to maximize recognition performance is also comparatively straightforward. Initially, all signs in the training set are labeled. For each sign in the dictionary, the training procedure then computes the

mean and covariance matrix over the data available for that sign and assigns them uniformly as the initial output probabilities to all states in the corresponding HMM. It also assigns initial transition probabilities uniformly to the HMM's states. Unlike the initial output probabilities, initial transition probabilities do not influence the performance of the fully trained HMMs greatly.

The training procedure then runs the Viterbi algorithm repeatedly on the training samples, so as to align the training data along the HMM's states. The aligned data are then used to estimate better output probabilities for each state individually. This realignment yields major improvements in recognition performance, because it increases the chances of the Baum-Welch reestimation algorithm converging to an optimal or a near-optimal maximum. After constructing these bootstrapped HMMs, the training procedure finishes by reestimating each HMM in turn with the Baum-Welch reestimation algorithm outlined in Section 4.2.

The by far most challenging problem in isolated recognition is extracting a feature vector that optimizes recognition performance. Even after obtaining accurate three-dimensional data from our computer vision method described in Section 3, we found that the features used for recognition — and the way that they are represented — greatly influence recognition performance. The experimental results given in Section 8.1 demonstrate how the feature vector affects performance.

There are several reasons why performance is so sensitive to choosing the type of feature vector: First, some features carry more information than others; for example, three-dimensional features are more reliable than two-dimensional ones. Second, some features are more invariant to changes in orientation and position than others; for example, polar coordinates are more invariant to rotations than Cartesian coordinates [1]. Third, the statistical properties of some features change, depending on the duration of a sign. For this reason, the positions of the hands in three-dimensional space perform better than the velocities of the hands (see also Section 8.2). Fourth, the statistical distribution of the features during the course of a sign seems to play a role. For some features, their distribution fits Gaussian densities naturally, whereas for others it does not.

If the latter explanation holds true, we should see a major improvement in recognition performance from using multiple Gaussian mixtures as the output probabilities for HMMs, instead of using just one single Gaussian density. However, we did not experiment with multiple mixtures because of the lack of sufficient training data.

The number of states and the topology used for the HMMs is also important. Sign language as a time-varying process lends itself naturally to a left-right model topology. Finding the optimum number of states, which depends on the frame rate and on the complexity of the signs involved, is an empirical process. We used the same model topology for all signs, and determined experimentally that for our task a model with 9 states was sufficient, which is depicted in Figure 3. The output probabilities were single Gaussian densities with diagonal covariance matrices, because we had insufficient training data for multiple mixtures.

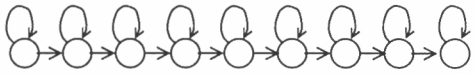


Figure 3: Left-right HMM topology for isolated ASL recognition.

5.2 Continuous Recognition

Continuous sign recognition, on the other hand, is much harder than isolated sign recognition. There is no silence between the signs, so the straightforward method of using silence to distinguish boundaries fails. Here HMMs offer the compelling advantage of being able to segment the streams of signs automatically with the Viterbi algorithm. Coarticulation effects further complicate continuous recognition. We now discuss them in detail, before we describe the techniques needed to train HMMs for continuous recognition.

5.2.1 The Coarticulation Problem

Coarticulation means that the pronunciation of a sign is influenced by the preceding and following signs. One of the most visible effects of coarticulation in ASL is that a wide range of movements are inserted between signs.

For example, the sign for "FATHER" is performed by repeatedly tapping the forehead, and the sign for "READ" is performed in neutral space in front of the chest. If these two signs are performed in succession, an extra movement from the forehead to neutral space appears (Figure 4). This phenomenon is called **movement epenthesis** [5]. We discuss its implications for ASL recognition more thoroughly in [20].

Figure 4: Movement epenthesis. The arrow in the middle picture indicates an extra movement between the signs for "FATHER" and "READ" that is not present in their lexical forms.

Speech recognizers handle coarticulation by training phoneme context-dependent HMMs. They train a separate model for each possible combination of three phonemes in sequence that could occur during natural speech. In principle, the same idea applies to sign language recognition, and we performed some experiments to verify the applicability, see Section 8.3.

A possible way to train context-dependent models for ASL recognition is to use whole signs as the phonological unit in ASL.[2] Thus, triphone context-dependent models from speech recognition correspond to tri-sign context-dependent models in ASL recognition. In other words, a separate model is trained for each combination of three signs in sequence. The first and the third sign in the sequence form the context for the middle sign, with which the model is associated.

Tri-sign context-dependent modeling, however, is prohibitively expensive, because it requires $O(W^3)$ models overall, where W is the vocabulary size. Collecting such a large amount of training data necessary to obtain reliable estimates for the models is intractable even for small vocabulary sizes. This intractability is a negative consequence of using whole signs as the phonological unit. Unlike for speech recognition, which has to handle only approximately 40 classes of allophones, there is no upper bound on the number of models required for ASL recognition with whole signs as the smallest unit.

Therefore, we used only bi-sign context-dependent models, which require a model for every possible combination of two signs. The model is associated with the second sign, and the first sign forms its preceding context.

Bi-sign context-dependent modeling requires $O(W^2)$ models. Although this complexity is an improvement over $O(W^3)$, it is still too large for anything but a small vocabulary. Speech recognizers reduce the number of models required by using the observation that many contexts are very similar. Therefore, they tie the parameters of the models corresponding to similar contexts, such that the transition and output probabilities are shared between these models. This technique significantly reduces the number of distinct models.

Parameter tying is also applicable to ASL recognition, but it is not as effective as for speech recognition. The main reason for the reduced effectiveness is that movement epenthesis inserts many movements unrelated to the signs' lexical forms. The implication is that context-dependent models will work well only with prohibitively large amounts of training data.

In fact, it is questionable whether context-dependent modeling is a good solution to the coarticulation problem in ASL recognition at all. Movement epenthesis is a phonological process in ASL and should be treated as such; that is, the movements induced by epenthesis are separate phonemes. Using context-dependent models to capture them is implausible from a phonological point of view. It seems to make more sense to model the movements explicitly. We follow up on this idea in [20] and show that it leads to better recognition performance.

[2]This assumption is not correct: Whole signs are not the smallest unit in ASL phonology, but this topic is beyond the scope of this paper.

5.2.2 The Training Procedure

A sign in our data collected at natural signing speeds was between 10 and 45 frames long, not counting the frames needed for the transition between signs. Because of the movements between signs, the HMM topology must be more flexible than the one described for isolated recognition in Section 5.1. These considerations led us to using the left-right model shown in Figure 5.

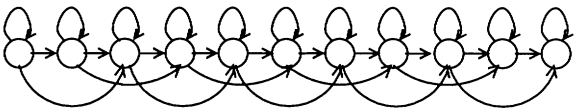


Figure 5: Topology of the context-dependent model. The arcs that skip states allow the modeling of variabilities in the duration of different signs.

Like for isolated recognition, we determined the optimal number of states experimentally. For the output probabilities, we chose a single Gaussian density with diagonal covariance, as we had insufficient training data for estimating full-rank covariance matrices.

Training continuous recognition models is much harder than training isolated recognition models, because it is difficult to obtain good initial estimates of the HMM parameters. Viterbi realignment (see Section 5.1) works only if the training data is accurately labeled, including the boundaries between the individual signs. Obtaining these boundaries is very difficult and time-consuming; even humans have trouble determining where a sign ends and the next one starts.

The alternative to using Viterbi realignment is using a flat-start scheme. It consists of computing the global mean and covariance matrix over the entire training data set and assigning these as the initial output probabilities to the HMMs. We used this scheme to initialize the HMMs.

We then used **embedded training** [24] to reestimate the HMMs. Each iteration of this procedure concatenates the HMMs corresponding to the individual signs in a training sentence into a single large HMM. It then reestimates the parameters of the large HMM with a single iteration of the Baum-Welch algorithm described in Section 4.2, as usual. The reestimated parameters, however, are not immediately applied to the individual HMMs. Instead, they are pooled

in accumulators, and applied to the individual HMMs only after the training procedure has iterated over all sentences in the training set.

Hence, embedded training effectively trains all models in parallel with the *entire* training set. It yields better parameter estimates than training the HMMs independently [24].

In the case of context-independent models, using the flat start scheme followed by several embedded training runs is all that is necessary to train HMMs for recognition. Context-dependent models are more difficult to train than context-independent models, because the training involves two extra steps. These consist of generating the context-dependent models, and tying the parameters of HMMs with similar contexts (see also Section 5.2.1).

The first extra step, which consists of generating the context-dependent models, requires care, because for context-dependent models there exist far fewer training examples per model than for context-independent models. In this case, embedded training is likely to yield the best parameter estimates for context-dependent models if they have already been initialized with better values than the global mean and covariance matrix from the flat-start scheme.

Therefore, we ran several embedded training runs on the context-independent models and then generated context-dependent models with the same parameters as the context-independent models. It is vital to avoid overtraining the context-independent models by keeping the number of initial training passes low. The probabilities should not have fully converged yet. Otherwise, using context-dependent models actually decreases recognition performance.

The second extra step, which consists of tying the parameters, is also vital to the context-dependent models' performance, especially because of our relative lack of training data. Tying parameters reduces the number of models, as signs with similar contexts then share a common model. As a result, more training data per model becomes available.

Unfortunately, parameter tying is a highly empirical process. Our experiments indicated that tying the transition probabilities properly had the greatest influence on recognition results. We used the ending locations of the signs in the preceding context to decide on the tying. For example, the signs for "BROTHER" and "SISTER" end in the same location. As a result, the two models for a sign occurring after the signs for "BROTHER" or "SISTER," such as "LIKE," can share the same transition probabilities. We also used the ending locations to decide on tying the output probabilities. For our data set, the tying process reduced the number of models to less than one sixth of their original number.

6 Coupling of Vision and HMMs

In the preceding section we reviewed how HMMs can be used for ASL recognition. The use of HMMs alone, however, imposes some limitations, one of which is insufficiency of training data, especially while training context-dependent models. Furthermore, the probability theory assumptions underlying the HMM theory, as described in Section 4.2, are often not valid: Successive observations are often not independent, the transition from one state to the next often depends not only on the current state, but also on the state history, and the distribution of observations does not necessarily resemble a normal density.

Another problem is that the HMM theory does not provide for any dynamic weighting of features depending on a sign's context. For example, the invariant features for some signs, such as "I," are the endpoints of their movements with respect to a body part, and the movements are unimportant. For other signs, only the movements are invariant. The parts of the feature set that should be examined and ignored for each class of signs are mutually exclusive.

To alleviate these limitations, we investigated the coupling of the HMM recognition process with an independent computer vision-based motion analysis that temporally segments the signal and extracts its geometric properties. The idea is that a sign can be described in terms of one or more geometric primitives, such as hand movements along a line, in a plane, or a circle. This idea is supported by the existence of transcription systems, such as the HamNoSys [14], that base the description of the movements on geometric primitives.

The presence of three-dimensional information is crucial for the coupling to work. In the past, geometric fitting of planes has already been used for rough segmentation [12], but not for providing additional information about the nature of the fits to the HMM recognition process.

6.1 Segmentation of the Signal

To extract the geometric properties of the continuous signal estimated with our computer vision methods, it must first be segmented temporally into its parts. Any change of the type of arm movement is likely to be accompanied by a dip in the velocity. Thus, minima in the absolute values of the velocity vector provide strong hints at segmentation boundaries. However, there are typically many more velocity minima than segmentation boundaries. Thus, the segmentation process must provide facilities to merge adjacent segments.

After performing initial segmentation based on velocities, our algorithm attempts to fit geometric primitives to the individual segments. These currently consist of lines, planes, and holds[3] at a position in space.

[3] A hold is a short period of time, during which no hand movements

The fit of a hold is determined by computing the covariance matrix over the segment's position data. If there is little movement, the eigenvalues of the matrix in every direction are small, and consequently its trace is small.

The least-squares fit of a line is governed by

$$\sum_i e_i = \sum_i ||\mathbf{p}_i - (\mathbf{d} \cdot \mathbf{p}_i)\,\mathbf{d}||^2, \qquad (19)$$

where e_i is the distance of $\mathbf{p}_i$ to the line, and $\mathbf{d}$ is the line's unit direction vector. Let $\mathbf{P}$ be a matrix containing the points $\mathbf{p}_i$ in the segments as its row vectors. Minimizing Equation 19 with respect to $\mathbf{d}$ corresponds to maximizing $\mathbf{d}^T\mathbf{P}^T\mathbf{P}\mathbf{d}$. By Rayleigh's principle, the maximal-eigenvalue eigenvector of $\mathbf{P}^T\mathbf{P}$ maximizes this equation, which is equivalent to the maximal-eigenvalue eigenvector of the points' covariance matrix. This eigenvector is the line's direction vector. The other two eigenvalues indicate the goodness of fit — the smaller they are with respect to the largest eigenvalue, the better the fit.

The least-squares fit of a plane is governed by

$$\sum_i e_i = \sum_i ||\mathbf{p}_i \cdot \mathbf{n}||^2, \qquad (20)$$

where e_i is the distance of $\mathbf{p}_i$ to the plane, and $\mathbf{n}$ is the plane's unit normal vector. If $\mathbf{P}$ is a matrix containing the points $\mathbf{p}_i$ as its row vectors, the minimal-eigenvalue eigenvector of $\mathbf{P}^T\mathbf{P}$ minimizes Equation 20 with respect to $\mathbf{n}$. Hence, minimizing this equation is equivalent to finding the minimal-eigenvalue eigenvector of the points' covariance matrix. The other two eigenvalues indicate the goodness of fit — the larger they are with respect to the smallest eigenvalue, the better the fit.

Using least-squares fitting is based on the assumption that the signal noise term is captured by a normal distribution. If this assumption is not valid, the least-squares estimator is likely to yield poor results, because of its sensitivity to outliers. On the other hand, in three-dimensional space, the least-squares estimator is much easier to compute than more robust estimators. It would be interesting to compare its performance on temporal segmentation to the performance of robust regression estimators [13], such as the least median of squares estimator [2, 11], or the repeated median estimator [16, 9].

After the initial fit, the algorithm pools the primitives into a directed acyclic graph (DAG), schematically depicted in Figure 6. Note that the individual segments are not mutually exclusive; for example, data can fit both a line and a plane.

If the algorithm fails to fit any geometric primitives to some segment, it inserts the segment into the DAG as a "wild card," which is defined conservatively to match any kind of geometric primitive. It then attempts to merge adjacent segments if they are compatible, in an attempt to eliminate spurious segmentation boundaries.

We defined adjacent segments to be compatible for a merge if they shared the same type of geometric primitive in similar orientations, and if the merged segment still fit the same type of geometric primitive as its constituting segments. In addition, we considered a wild card to be compatible with another geometric primitive if this primitive also fit the merged segment.

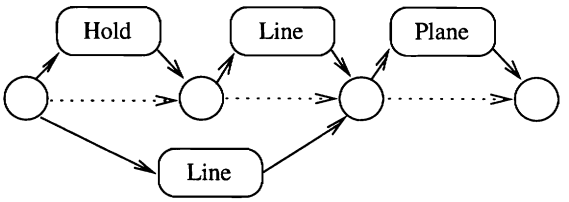


Figure 6: Geometric primitives pooled into a DAG. Circles denote segmentation boundaries. Dotted arcs denote possible null transitions; they are necessary to compensate for spurious segments. Sometimes data can fit multiple geometric primitives; in this DAG the data of the first two segments fit both a hold followed by a line, and a simple line.

The DAG now gives all possible segment sequences that are a valid representation of the signal. If a sequence is to be valid, it must be obtainable by tracing a path through the DAG from the leftmost segmentation boundary to the rightmost segmentation boundary. In the example given in Figure 6 the sequences "Hold, Line, Plane," and "Line, Plane" would both be valid sequences, but "Plane, Plane" would not, because the latter does not lie on any path through that DAG.

This discussion has so far ignored the possibility of spurious segments arising from the vision analysis. That is, the analysis might recognize a segment that should be part of another, but the merge process fails to merge it into another segment. The main reason for the existence of spurious segments is undersampling. If a segment consists of very few samples, it is often impossible to extract reliable information from it. Our algorithm attempts to solve this problem by adding arcs to the DAG from each segmentation boundary to the next (represented by the dotted arcs in Figure 6). Thus, a path through the DAG can optionally skip these spurious segments.

6.2 Using the Motion Analysis with HMMs

Each sign in the vocabulary has associated one or more templates that comprise the sign's geometric primitives with weights of each feature's relative importance. These

take place.

primitives are matched against those in the DAG. Assuming that the segmentation process yields correct results, the following must be true: If a sequence of signs is represented by the input signal, the sequence of geometric primitives corresponding to the signs must form a path through the DAG. We call such a sequence of signs **valid** with respect to the computer vision DAG.

This observation suggests an application of the motion as a backup check for the HMM framework. First recognize a candidate sentence from the input signal via the Viterbi algorithm. Then generate all possible sequences of geometric primitives corresponding to the recognized signs and construct another DAG from them. Using dynamic programming, match the two DAGs against each other. If the two DAGs share a common path, accept the candidate sentence as correct. Otherwise, reject the candidate sentence as incorrect.

The justification for this algorithm comes from the following properties of the DAGs: If the two DAGs share a common path, there is a sequence of geometric primitives that forms a path through the computer vision DAG. Furthermore, this sequence of geometric primitives is one of the possible sequences generated from the candidate sentence. Thus, the candidate sentence is valid with respect to the computer vision DAG. Conversely, if no such common path exists, none of the sequences of geometric primitives generated from the candidate sentence forms a path through the computer vision DAG. Thus, the candidate sentence is not valid with respect to the computer vision DAG and should be rejected.

6.3 Discussion of the Coupling

The HMM recognition algorithm and the vision matching algorithm complement each other. The advantages of the HMM recognition method are automatic segmentation during both training and recognition, and a fully formalized training procedure. The disadvantages are poor performance in the presence of insufficient training data, no formal way to weight features dynamically, and possible violations of the stochastic independence assumptions.

The advantages of the vision matching method are the possibility of weighting the relative importance of features dynamically, and independence from insufficient training data. A significant disadvantage is that estimating the geometric properties of the signs in the vocabulary requires manual labeling and analysis of the data. Furthermore, segmentation must be done explicitly, which raises the possibility of spurious segments, as described in Section 6.1, or the possibility of missing segments. Coarticulation sometimes also changes the geometric properties of the signal, such that the templates for the correct sequence of sign no longer match the actual signal. Coping with the changes in the geometric properties is an important task for future research.

7 Data Collection

For our experiments we collected data, using both our computer vision system, and an Ascension Technologies Flock of Birds. The reason for using the latter was that it is faster at this point than the computer vision system, and hence more suitable for prototyping.

The computer vision system yields rotation, $\mathbf{q}_\theta$, and translation, $\mathbf{q}_c$, of each segment of the arm, as described in Section 3. Figure 7 gives an example of the computer vision tracking process. The images show the high accuracy of the computer vision system; in fact, it is comparable to the accuracy achieved by the Flock of Birds system.

The Flock of Birds system consists of a magnet and six sensors that detect their rotation, $\hat{\mathbf{q}}_\theta$, and translation, $\hat{\mathbf{q}}_\mathbf{c}$, with respect to the magnet at 25 frames per second. We used the data from both systems interchangeably with a simple alignment of coordinate systems. The coordinate system was right-handed, with the origin at the base of the signer's spine and the x axis facing up.

Figure 7: Fitting the three-dimensional models to the signer's arms. From top to bottom, the signs for "FATHER," "I," and "MAIL" are displayed. From left to right, the front, side, and top views are displayed.

We used the 53-sign vocabulary listed in Table 1. Their pronunciations followed the ASL dialect used in the Philadelphia, PA, area. The goals in choosing the vocabulary were to be able to express sentences that could have occurred in a natural conversation, and to make intensive use of the signing space, so as to demonstrate the advantages of three-dimensional data over two-dimensional data. We collected 486 continuous ASL sentences, each between

Category	Signs used
Nouns	America, Christian, Christmas, book, brother, chair, college, family, father, friend, interpreter, language, mail, mother, name, paper, president, school, sign, sister, teacher
Pronouns	I, my, you, your, how, what, where, why
Verbs	act, can, give, have, interpret, like, make, read, sit, teach, try, visit, want, will, win
Adjectives	deaf, good, happy, relieved, sad
Other	if, from, for, hi

Table 1: The complete 53 sign vocabulary

2 and 12 signs long, with a total of 2345 signs. The only constraints on the order and occurrence of signs were those dictated by the grammar of ASL [19].

Furthermore, we collected examples of each sign for isolated recognition. Because part of the data were corrupted during the collection process, we discarded all signs for which we did not have enough intact training examples. This left 656 examples over a range of 40 signs. Each sign had at least 6 examples available for the training set, and 2 examples available for the test set.

8 Experiments

We performed isolated, continuous, and vision-HMM coupled ASL recognition experiments. We used Entropic's Hidden Markov Model Toolkit (HTK) Version 2.02 for training and testing in all of our experiments.

8.1 Isolated Recognition Experiments

The goal of the isolated recognition experiments was to discover a set of features that maximizes HMM recognition performance. We used different features in our experiments, including wrist position coordinates of both hands (denoted by x, y, z), wrist position expressed in polar coordinates in the x-y plane (denoted by r_{xy}, θ_{xy}), polar coordinates in the x-z plane (denoted by r_{xz}, θ_{xz}), wrist position expressed in spherical coordinates (denoted by r, θ, ϕ), and wrist orientation angle (denoted by δ), as well as derivatives of these (denoted by a dot). We also combined several features in some experiments.

We ran repeated experiments, more than 10,000 total, with different features and randomly selected training and test sets on a per-experiment basis. Three quarters of the examples for each sign were in the training set and the rest were in the test set. Each selection yielded 178 test examples per experiment. Some typical results are given in Table 2. In addition, we performed experiments to compare the merits of using three-dimensional coordinates versus two-dimensional coordinates by projecting the coordinates on planes. The results are shown in Table 3.

Features	μ	σ	B	W	N
x, y, z	98.42%	0.99%	100.0%	93.8%	463
r_{xy}, θ_{xy}, z	98.72%	0.79%	100.0%	95.5%	494
r_{xy}, r_{xz}, θ_{xy}, θ_{xz}, x, y, z	98.78%	0.78%	100.0%	94.9%	882
r, θ, ϕ	96.48%	1.31%	100.0%	93.3%	210
$\dot{x}, \dot{y}, \dot{z}$	96.87%	1.21%	100.0%	93.3%	167
x, y, z, δ	98.25%	0.92%	100.0%	95.5%	167
$\dot{r}_{xy}, \dot{\theta}_{xy}$, $\dot{z}$	96.28%	1.04%	98.9%	93.8%	120
$\dot{r}, \dot{\theta}, \dot{\phi}$	95.89%	1.29%	98.9%	92.1%	150

Table 2: Results of isolated sign recognition with three-dimensional features. μ, σ, B, W, and N correspond to the average percentage of correctly recognized signs, standard deviation, best case, worst case, and number of experiments, respectively. All experiments used a test set of 178 signs.

Features	μ	σ	B	W	N
r_{xy}, θ_{xy}	98.06%	1.26%	100.0%	94.9%	118
x, y	97.75%	1.20%	100.0%	94.9%	118

Table 3: Results of isolated sign recognition with two-dimensional features. The meaning of the columns is the same as in Table 2.

8.2 Analysis of Isolated Recognition

The low error rates of the best feature sets show that with a good selection of features, the hand movements alone, without hand configuration information, carry sufficient information to discriminate among many different signs. Polar coordinates slightly outperformed Cartesian coordinates. A combination of both yielded the best results, although the difference is not significant. However, the standard deviation of the combined feature set was lowest, indicating that a complex feature vector is more robust than a simple feature vector.

Position coordinates significantly outperformed velocities. The reason for the poor performance of velocity features is that the statistical properties of the velocities change with variations in the sign's duration. In contrast, the statistical properties of position coordinates are largely unaffected by the duration of signs, because HMMs absorb variations in duration through transitions looping back to the same state. Yet, position coordinates have the significant disadvantage that they are not invariant with respect to location. The lack of invariance will cause problems for future applications that attempt to capture commonalities between movements at different locations in space.

Three-dimensional features performed better than two-dimensional features, although the difference is not large. The difference would probably become more significant with a larger vocabulary. The differences in standard deviation, however, indicate that three-dimensional features are more robust than two-dimensional features.

It is an important consequence of the experiments' results that the performance of the feature vectors depends on the actual examples in the training set, all other factors being equal. Thus, only performing a large number of experiments yields reliable estimates of the relative merits of different features.

8.3 Continuous Recognition Experiments

We split the 486 sentences randomly into a training set with 389 examples and a test set with 97 examples (containing 456 signs). Each sign in the vocabulary occurred at least once in the test set. The training and test sets were the same throughout all experiments, and no portion of the test set was used for training in any way. We ran three-dimensional experiments with and without context-dependent HMMs, and two-dimensional experiments (by projecting the data on planes; the results given are the best that we found).

In accordance with the results from isolated experiments that position coordinates perform better than velocities, and that a complex feature vector is more robust than a sparse one, we chose our feature vector to be $(x, y, z, \theta_{xy}, \theta_{xz}, \dot{x}, \dot{y}, \dot{z}, \delta)$ for both hands. That is, it consisted of Cartesian and polar position coordinates, velocities, and wrist orientation angles. The task grammar was a simple word loop, so every sign was equally likely at any time in the HMM network.

Table 4 shows the experimental results. We use word accuracy as our evaluation criterion. It is computed by subtracting the number of insertion errors from the number of correctly spotted signs. The number of words in the result for two-dimensional data is lower than in the other results, because for one sentence the Viterbi beam-searching optimization pruned all paths through the HMM network (see also Section 4.2).

8.4 Analysis of Continuous Recognition

The results are clearly in favor of using three-dimensional data over two-dimensional for continuous recognition. The 6.3 percent difference is large, although, according to our experiences with isolated recognition, one experiment is not enough to estimate the real difference reliably.

Context-dependent models outperformed context-independent models, but the increase in performance was small, probably to a large extent because of insufficient training data — context-dependent modeling requires huge amounts of data to become effective. Also, cross-sign context-dependent modeling for ASL is implausible from a phonological point of view (see Section 5.2.1). The alternative is modeling movement epenthesis directly, and it appears to perform better [20].

Type of experiment	Word accuracy	Details
3D context independent	87.71%	H=416, D=8, S=32 I=16, N=456
3D context dependent	89.91%	H=424, D=6, S=26 I=14, N=456
2D context dependent	83.63%	H=394, D=14, S=44 I=16, N=452

Table 4: Results of continuous recognition experiments. H denotes the number of correct signs, D the number of deletion errors, S the number of substitution errors, I the number of insertion errors, and N the total number of signs in the test set.

More than half of the substitution errors in each experiment were confusions between "I" and "MY," and "YOU" and "YOUR," which differ only in hand configuration. We expect that adding features describing the hand configuration will improve recognition performance significantly.

Repeating the context-dependent experiment with five-best recognition showed that the absence of a strong grammar for constraining the HMM network degrades recognition performance significantly. In many cases, the correct sentence was the only grammatical sentence among the five best candidates. In other cases, all five candidates were ungrammatical.

Unfortunately, using a strong grammar for a test set as diverse as ours is not practical, because the size of an HMM network grows exponentially with the number of rules present in the grammar. Statistical language models, such as bigram models, have proved to be an effective solution to this problem in speech recognition. We show in [20] that bigram language models are promising for ASL recognition as well. However, they require a large corpus of labeled real-world data to become truly effective. Presently, no such corpus exists for ASL.

8.5 Coupling Experiments

To investigate the effects of coupling the three-dimensional motion analysis with the HMM framework, we performed two experiments. In the first experiment, we analyzed all sentences in the test set with our motion analysis, so as to provide an upper bound on its performance. If the motion analysis had worked perfectly, it should have accepted all of these 97 test sentences. In reality, however, it rejected 10 out of these 97 sentences.

A closer look at the 10 rejected sentences revealed that five of these were not recognized correctly by the context-dependent HMMs either. Thus, it is likely that these five

sentences were not signed precisely enough during the data collection process. The other five rejected sentences indicate that the motion analysis still needs improvement.

In the second experiment, we ran the coupling algorithm on the actual recognition hypotheses from the context-dependent HMMs in the experiments in Section 8.3. This time, the algorithm also eliminated 10 sentences out of 97. Five of these were correctly rejected; that is, the HMM framework had provided incorrect results for them. Thus, at the current moment, coupling HMMs with motion analysis breaks even with using the HMM framework by itself. The word accuracy achieved by the coupling was 90.10%, which is slightly better than the 89.91% word accuracy achieved by the context-dependent models alone.

As we have used only a small part of the full power of computer vision motion analysis so far, we see these results as evidence that coupling will eventually be able to outperform either method independently.

9 Summary

We have developed a framework for recognizing American Sign Language from three-dimensional data obtained with computer vision techniques. We showed how to collect three-dimensional data from computer vision and use them as input to Hidden Markov Models. We also determined that three-dimensional features are superior over two-dimensional ones.

By using context-dependent modeling, we improved recognition performance. Through coupling vision processes with Hidden Markov Models, we took a first step toward overcoming the limitations of either method by itself.

10 Future work

The collection of a standardized corpus of real-world ASL conversations and story telling should be a high priority for future work. The current lack of such a corpus makes it impossible to compare results from different researchers. Furthermore, it makes the development of statistical language models for ASL difficult. Such language models are necessary for large-scale applications.

Testing the algorithms described in this paper and in [20, 21] with a larger vocabulary is also important. Only then it will be possible to judge how well these algorithms scale.

On the linguistic side of ASL recognition, future work should incorporate facial expressions and other phonological processes in ASL into the recognition framework. It also needs to address how to make use of hand configuration information; using this information effectively seems to be nontrivial. Furthermore, future work has to find ways to use statistical language models, so as to counterbalance the impracticability of using strongly constrained task grammars.

On the computer vision side of ASL recognition, future work should elaborate on the coupling of computer vision and HMMs and make the computer vision analysis more robust. This work should consist of recognizing more different geometric properties, fine-tuning the sign templates, and fine-tuning the dynamic weighting of features based on the properties of each sign that is matched to the signal. It also needs to address coarticulation effects, which it has ignored so far.

It is also necessary to to develop an anthropometrically correct model of the human hand, so that the computer vision tracking process can make hand configuration information available to the recognition framework.

Acknowledgments

This work was supported in part by a NSF Career Award NSF-9624604, ONR-DURIP'97 N00014-97-1-0385 and N00014-97-1-0396, ONR Young Investigator Proposal, and NSF IRI-97-01803. Ioannis Kakadiaris helped in obtaining the computer vision samples.

References

[1] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick, and A. Pentland. Invariant features for 3-D gesture recognition. 2nd International Workshop on Face and Gesture Recognition, Killington, VT, 1996.

[2] H. Edelsbrunner and D. L. Souvaine. Computing least median of squares regression lines and guided topological sweep. *Journal of the American Statistical Asscoiation*, 85:115–119, 1990.

[3] R. Erenshteyn and P. Laskov. A multi-stage approach to fingerspelling and gesture recognition. Proceedings of the Workshop on the Integration of Gesture in Language and Speech, pp. 185–194, Wilmington, DE, 1996.

[4] K. Grobel and M. Assam. Isolated sign language recognition using hidden Markov models. SMC'97, pp. 162–167.

[5] S. K. Liddell and R. E. Johnson. American Sign Language: The phonological base. *Sign Language Studies*, 64:195–277, 1989.

[6] I. A. Kakadiaris, D. Metaxas, and R. Bajcsy. Active part-decomposition, shape and motion estimation of articulated objects: A physics-based approach. CVPR'94, pp. 980–984.

[7] I. A. Kakadiaris and D. Metaxas. 3D human body model acquisition from multiple views. ICCV'95, pp. 618–623.

[8] I. A. Kakadiaris and D. Metaxas. Model based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. CVPR'96, pp. 81–87.

[9] J. Matoušek, D. Mount, and N. S. Netanyahu. Efficent randomized algorithms for the repeated median line estimator. To appear in Algorithmica. Available online at `http://www.cs.umd.edu/~mount/pubs.html`.

[10] D. Metaxas. *Physics-based Deformable Models: Applications to Computer Vision, Graphics and Medical Imaging.* Kluwer Academic Publishers, November 1996.

[11] D. Mount, N. Netanyahu, K. Romanik, R. Silverman, and A. Y. Wu. A practical approximation algorithm for the LMS line estimator. 8th ACM-SIAM Symposium on Discrete Algorithms, pp. 473–482, 1997.

[12] Y. Nam and K. Y. Wohn. Recognition of space-time hand-gestures using Hidden Markov model. ACM Symposium on Virtual Reality Software and Technology, pp. 51–58, Hong Kong, 1996.

[13] N. S. Netanyahu, V. Philomin, A. Rosenfeld, and A. J. Stromberg. Robust detection of straight and circular road segments in noisy aerial images. *Pattern Recognition*, 30(10):1673–1686, 1997.

[14] S. Prillwitz et al. *HamNoSys. Version 2.0; Hamburg Notation System for Sign Languages. An introductory guide*, volume 5 of *International Studies on Sign Language and Communication of the Deaf.* Signum Verlag, Hamburg, 1989.

[15] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[16] A. F. Siegel. Robust regression using repeated medians. *Biometrika*, 69:242–244, 1982.

[17] J. M. Siskind and Q. Morris. A maximum-likelihood approach to visual event classification. ECCV'96.

[18] T. Starner and A. Pentland. Visual recognition of American Sign Language using Hidden Markov models. International Workshop on Automatic Face and Gesture Recognition, pp. 189–194, Zürich, Switzerland, 1995

[19] C. Valli and C. Lucas. *Linguistics of American Sign Language: An Introduction.* Gallaudet University Press, Washington DC, 1995.

[20] C. Vogler and D. Metaxas. Adapting Hidden Markov models for ASL recognition by using three-dimensional computer vision methods. SMC'97, pp. 156–161.

[21] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. ICCV'98, pp. 363–369.

[22] M. B. Waldron and S. Kim. Isolated ASL sign recognition system for deaf persons. *IEEE Transactions on Rehabilitation Engineering*, 3(3):261–71, September 1995.

[23] M. Waleed Kadous. Machine recognition of Auslan signs using PowerGloves: Towards large-lexicon recognition of sign language. Proceedings of the Workshop on the Integration of Gesture in Language and Speech, pp. 165–174, Wilmington, DE, 1996.

[24] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland. *The HTK Book (for HTK 2.0).* Cambridge University, 1995.