Color and Illumination Independent Hand Tracking and Gesture Recognition

Tekin Meriçli
Department of Computer Sciences
University of Texas at Austin
tmericli@cs.utexas.edu

Abstract

Recognition of human motion provides hints to understand human activities and gives opportunities to the development of a new human-computer interaction (HCI) interface. Hidden Markov Models (HMMs) are used for visual recognition of complex, structured hand gestures such as the ones found in a sign language, since they have proved their success in recognizing speech and handwriting. In this paper, we introduce a hand gesture recognition system to recognize gestures in real-time. Hand tracking is performed in two different ways. The first method is based on color segmentation and blob generation over the hand region, and the second method uses block matching and particle filtering algorithms to detect the moving hand which makes the system totally color and illumination independent. In both methods, extracted information is used as the input to the HMM based gesture recognizer.

1. Introduction

Recognition and interpretation of human motion has become one of the most attractive topics in computer vision and pattern recognition because of its wide application possibilities. Being able to interpret the motion in a scene obtained from a camera makes vision-based human-computer interaction possible in a more natural way. In this paper, a system which uses a color camera for tracking hands in real time and interpreting American Sign Language (ASL) by using Hidden Markov Models (HMMs) is described.

Sign language recognition is a multidisciplinary research area involving pattern recognition, computer vision, natural language processing and psychology, and a comprehensive problem because of the complexity of the visual analysis of hand gestures. Although they are well-structured languages with a phonology, morphology, syntax and grammar, the linguistic characteristics of sign languages are different than that of spoken languages due to the existence of several components affecting the context such as the use of facial expressions and head movements in addition to the hand movements. However, the movements of the body parts other than the hands are not used to aid the recognition task addressed here. Also, studies have shown that recognition of a sign does not require a very complex model of the hand shape [4, 6]; therefore, our system produces only a coarse description of hand shape and trajectory during the tracking process and feeds the HMM with this information for recognition of the signed words.

The overall goal of this work is to make it possible for a computer to interact with a human via visual perception in real time by recognizing the observed motions with a low error rate. The rest of this paper is organized as follows. Section 2 gives background information about Hidden Markov Models, Block Matching algorithm, and particle filtering. Previous work on gesture recognition is provided in Section 3. General architecture of the overall system and details of each sub-system are explained Section 4. Section 5 provides information about experiments and obtained results. The last section summarizes the work and gives information about possible extensions and future work.

2. Background

This section provides some background information on Hidden Markov Models, Block Matching algorithm, and particle filtering.

2.1. Hidden Markov Models

If, given all present and past events, the conditional probability density of the current event in a time domain process depends only on the most recent k events, then this process is said to have Markov property. Here k is the constant that determines the order of the Markov process; that is, if $k = 1$, it means that the process is a first order process in which the current event depends only on the most recent past event.

The three key problems in HMM use can be listed as evaluation, estimation, and decoding. Given an observation sequence and a model, evaluation is calculating the prob-

ability that the observed sequence was generated by the model ($Pr(O|\lambda)$). Recognition is performed by choosing the model with the highest probability after evaluating the probability value for all competing models.

Although there are a number of different ways to calculate $Pr(O|\lambda)$, the nave way of doing that is to sum the probability over all possible state sequences in a model for the given observation sequence:

$$Pr(O|\lambda) = \sum_{allS} \prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(O_t) \quad (1)$$

Being an exponential computation, this method is inefficient although it is easy to implement. In order to increase the efficiency, the forward-backward algorithm can be used. Basically, the algorithm defines a forward variable α, and uses it to generate $Pr(O|\lambda)$, where π are the initial state probabilities, a are the state transition probabilities, and b are the output probabilities.

- $\alpha_1(i) = \pi_i b_i(O_1)$, for all states i (if $i \in S_I$, $\pi_i = \frac{1}{n_I}$; otherwise $\pi_i = 0$

- Calculating $\alpha()$ along the time axis, for $t = 2, ..., T$, and all states j, compute

$$\alpha_t(j) = [\sum_i \alpha_{t-1}(i) a_{ij}] b_j(O_t) \quad (2)$$

- Final probability is given by

$$Pr(O|\lambda) = \sum_{i \in S_F} \alpha_T(i) \quad (3)$$

In summary, the first step is for initializing the forward variable with the initial probability for all states, and the second step carries the forward variable forwards through time inductively. The final step gives the desired value of $Pr(O|\lambda)$.

The second problem in HMM use, the estimation problem, tries to adjust λ to maximize $Pr(O|\lambda)$ given an observation sequence O. Since the forward-backward algorithm already evaluates this probability, the only remaining task is to find a method to improve the initial model.

The evaluation and the estimation processes are sufficient for developing a HMM based system. However, Viterbi algorithm, which is a solution for the decoding problem, provides a quick means of evaluating a set of HMMs. The primary goal of decoding is to recover the state sequence given an observation sequence. It is very similar to forward-backward algorithm; in fact, it can be considered as a special form of the forward-backward algorithm where only the maximum path at each time step is taken instead of all possible paths. This, in turn, reduces computational load and allows the recovery of the most likely state sequence. Viterbi algorithm can be summarized as follows:

- Initialization. For all states i, $\delta_1(i) = \pi_i b_i(O_1)$; $\psi_i(i) = 0$

- Recursion. From $t = 2$ to T and for all states j, $\delta_t(j) = Max_i[\delta_{t-1}(i)a_{ij}]b_j(O_t)$; $\psi_t(j) = argmax_i[\delta_{t-1}(i)a_{ij}]$

- Termination. $P = Max_{s \in S_F}[\delta_T(s)]$; $s_T = argmax_{s \in S_F}[\delta_T(s)]$

- Recovering the state sequence. From $t = T - 1$ to 1, $s_t = \psi_{t+1}(s_{t+1})$

Viterbi algorithm is used for evaluation at recognition time in many HMM system implementations. Although the resultant scores are only an approximation since this algorithm only guarantees the maximum of $Pr(O, S|\lambda)$ over all state sequences S instead of the sum over all possible state sequences, [5] shows that it is sufficient most of the time.

Information on Hidden Markov Models have been compiled from the literature of much more detailed work on that technology [2, 3, 5, 8]. Much broader discussion on the subject can be found in [3, 7].

2.2. Block Matching Algorithm

The block matching algorithm is a standard technique for encoding motion in video sequences [14]. It aims at detecting the motion between two images in a block-wise sense. The blocks are usually defined by dividing the image frame into non-overlapping square parts. Each block from the current frame is matched into a block in the destination frame by shifting the current block over a predefined neighborhood of pixels in the destination frame. At each shift, the sum of the distances between the gray values of the two blocks is computed. The shift which gives the smallest total distance is considered the best match.

In the ideal case, two matching blocks have their corresponding pixels exactly equal. This is rarely true because moving objects change their shape in respect to the observer's point of view, the light reflected from objects' surface also changes, and finally in the real world there is always noise. Furthermore, from semantic point view, in scenes containing motion there are occlusions among the objects, as well as disappearing of objects and appearing of new ones. Despite the problems of pixel by pixel correspondence, it is fast to compute and is used extensively

for finding matching regions. Some of the most often used matching criteria based on pixel differencing are mean absolute distance (MAD), mean squared distance (MSD), and normalized cross-correlation (NCC) [15].

Choosing the right block size is not a trivial task. In general, bigger blocks are less sensitive to noise, while smaller blocks produce better contours. Certainly, the leading factor for choosing the block size is the size of the objects that need to be tracked. The size of the search region is important for finding the right match. Unfortunately the computational load grows fast (as a power of two) with the growth of the search area.

2.3. Particle Filtering

A particle filter is a sequential Monte Carlo algorithm, i.e. a sampling method for approximating a distribution that makes use of its temporal structure. The main objective of particle filtering is to "track" a variable of interest as it evolves over time, typically with non-Gaussian and potentially multi-model *pdf*. A series of actions are taken, each one modifying the state of the variable of interest according to some model. Moreover at certain times an observation arrives that constrains the state of the variable of interest at that time.

Multiple copies (particles) of the variable of interest are used, each one associated with a weight that signifies the quality of that specific particle. An estimate of the variable of interest is obtained by the weighted sum of all the particles. The particle filter algorithm is recursive in nature and operates in two phases: *prediction* and *update*. After each action, each particle is modified according to the existing model (*prediction* stage), including the addition of random noise in order to simulate the effect of noise on the variable of interest. Then, each particle's weight is re-evaluated based on the latest sensory information available (*update* stage). At times the particles with (infinitesimally) small weights are eliminated, a process called resampling [16].

3. Related Work

Sign language recognition requires both hand trajectory and hand posture (position, orientation, angles of the articulations) information. In order to solve the hand trajectory recognition problem, Hidden Markov Models have been used extensively for the last decade. Starner and Pentland implemented one of the earliest dynamic gesture recognition systems, where they used HMM to recognize American Sign Language using a single camera [1]. The vocabulary contains 40 signs and the sentence structure to be recognized was constrained to personal pronoun, verb, noun, and adjective. Lee and Kim [11] propose a method for online gesture spotting using HMMs. In 1997, Vogler and Metaxas [12] proposed a system for both isolated and continuous ASL recognition sentences with a 53-sign vocabulary. In a later study [13] the same authors attacked the scalability problem and proposed a method for the parallel modeling of the phonemes within an HMM framework.

Researches mentioned above are relatively new since the attempts at machine sign language recognition have begun to appear in the literature in the early 90's. Tamura and Kawasaki developed an image processing system which recognizes 20 Japanese signs based on matching *cheremes* (means "hand" in Greek) [9]. Takahashi and Kishino demonstrate a user dependent dataglove-based system that recognizes 34 of the 46 Japanese kana alphabet gestures using joint angle and hand orientation coding technique [10].

4. System Description

Detecting the hand in the image is the very first step in order to recognize even the simplest hand gesture. Detection of the hand in natural environments by using only the skin color information is a challenging and complex task. In the literature, two methods are widely used in order to make the detection problem easier; first one is to use special markers on the hand and fingers, and the second one is to restrict the environment to be able to detect the hand without the need for markers. Once the hand is detected, the system must be able to differentiate between gestures (which are defined by the gesture classes in the training set) and non gestures (either unintentional movements of the hand or gestures that are not in the training set) based on the information passed to the recognizer. This is especially essential for continuous recognition of hand gestures. A general framework of the system is given in Figure 1

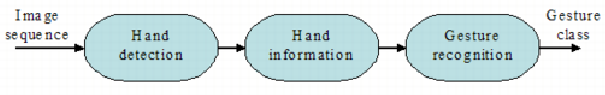

Figure 1. General architecture of the hand gesture recognition system

We used OpenCV and SharperCV, which is a Microsoft .NET wrapper for OpenCV, for image processing purposes, and Georgia Tech Gesture Toolkit (GT2k) for generation, training, and manipulation of HMMs. GT2k is a toolkit which leverages Cambridge University's speech recognition toolkit, HTK, to provide tools that support gesture recognition research. The entire system is developed by using Microsoft Visual Studio .NET 2003.

4.1. Hand Tracking

Although Hidden Markov Models are proven to be successful in gesture recognition applications, this success

highly relies on the quality of the input sequence. Measurement errors or noise in input sequences negatively affect both the training and recognition phases. In hand gesture recognition problem, the desired situation is recognition and tracking of the hand without using any extra object such as markers. There are some works in the literature on recognition of human skin and tracking it but nearly all of them relies on strict assumptions like fixed background with a color that in contrast to skin color. Even with these strict assumptions, proposed works suffer from hand-hand and hand-face ambiguities. If we relax the fixed background assumption, a third ambiguity source is mixing a hand with other people wandering around or objects with similar colors. Using sharply colored markers highly solves this problem and moreover it allows using two hands with different markers at the same time which is one of the base requirements of recognizing sign languages.

Two different methods are used for tracking the hands. First method is a colored-marker based approach. Second one is a powerful technique which uses hand movement as a cue for detection and applies particle filtering for stabilizing the result. This approach provides color and illumination independent recognition.

4.1.1 Colored Marker-Based Tracking

Using colored markers partially solves the skin ambiguity problem but the effect of environmental lighting conditions still causes trouble, especially in places receiving natural light. Even most robust segmentation algorithms fail when sun rises or falls. We will propose some methods that might overcome this reliability problem in discussions section.

For the sake of simplicity, we have used colored gloves as markers. In an offline process, pictures of markers are taken under a certain lighting condition and the mean red, green and blue values of pixels in the marker region are calculated for each marker color. During the execution, the following algorithm runs whenever a frame is grabbed:

- if a bounding box is calculated for a marker in previous frames,

  search a window having a size of the bounding box enlarged by a constant value to the four directions with the a priori known bounding box in center for each known bounding box.

- For each pixel being searched,

  calculate the sum of absolute differences among red, green and blue channels of the pixel and known mean values of red, green and blue of each marker.

  find the closest marker color

  if the difference is below some certain threshold, assign the pixel to that color. Otherwise, skip the pixel

  if assignment is done, use the pixel coordinates in calculation of centroid of the hand.

- if total number of pixels assigned to a marker color is over a threshold, conclude that the marker is detected and calculate centroid coordinates

If a marker is detected, coordinates of its centroid are stored in an array with temporal order to be used in recognizing the gesture.

4.1.2 Motion-Based Tracking

Motion is one of the building blocks of a gesture; therefore, our system is triggered by consistent (i.e. not caused by noise) motion, and tracks the moving object in the scene. In this specific domain, the main object in consistent motion is always the hand. Hence, after noise elimination phase, the system assumes that the motion is caused by the hand which becomes the main object of interest in the scene.

Block Matching algorithm is used for motion detection. First of all, the scene is divided into “blocks” which are square regions to be used for matching. Three important parameters of this algorithm are the block size, the amount of shift applied to the original image to explore the new image, and the maximum displacement amount that we allow a block to move between images. The result of this algorithm is a group of velocity vectors starting from the center of the original block and ending at the center of the matched block in the next image.

Motion-based tracking system uses an attention mechanism which is implemented by using a simplified version of particle filtering algorithm. The attention mechanism has two main purposes: eliminating the noise, and keeping track of the hand even when the hand motion slows down which makes the system much more stable. The particles are used for modeling possible locations of the center of the hand in the image; therefore, initially the particles are scattered over the whole image region. Gesture recognition phase starts when the confidence level on the hand position estimate exceeds a certain threshold. Therefore, in order to stimulate the system, the user shakes his hand for a couple of frames which causes the system to focus its attention to the centroid of the moving region. This focusing mechanism can best be explained as the attractive effect of particles having high confidence values on the particles with relatively low confidence values. Since the confidence value is directly proportional to the amount of motion, which is extracted from the magnitudes of the motion vectors provided by the Block Matching algorithm, the region on the image having a consistent movement attracts all the particles. Also, if the hand movement slows down, since it takes a couple of frames for the system to decrease the confidence value on the hand position and distribute the particles over the whole

image again, the system can still have a good-enough estimate of the hand location which is apparently much better than totally losing the track of the hand.

Noise elimination is a part of the motion-based detection method which is automatically provided by the core of this approach. Since the hand is the closest object to the camera while performing a gesture, it produces the biggest motion vectors among all the moving objects such as the head and the arm. However, those objects are not taken into consideration since the confidence values over those regions are relatively small compared to the confidence values over the hand region, which is the result of the magnitude of the corresponding motion vectors.

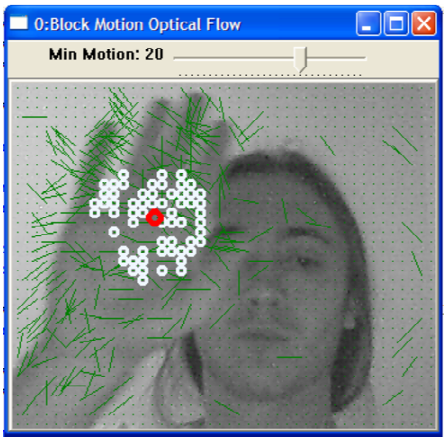

Figure 2. Example tracking #1

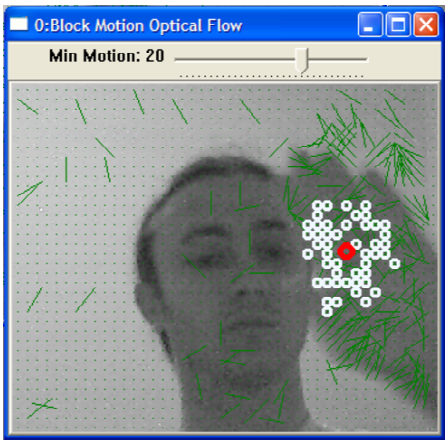

Figure 3. Example tracking #2

4.2. Hidden Markov Modeling

In order to determine the initial topology of the HMM, which can be fine-tuned emprically, the number of states used for specifying a sign should be estimated. One possibility is to use a fixed topology but train a different HMM with different parameters for each gesture to be recognized. Starner and Pentland [1] showed that a four state HMM with one skip transition is sufficient for this task. A shortcoming of this approach is the inability of the system to discriminate between a circle and a square, for example. Another possibility is to assign different topologies for each sign; however, as the number of signs to be recognized increases, this process becomes inefficient. In order to make our system more discriminative, we used a fixed topology with 8 states. Having more states in the topology made it possible to discriminate between gestures having very similar structure such as “square” and “circle”.

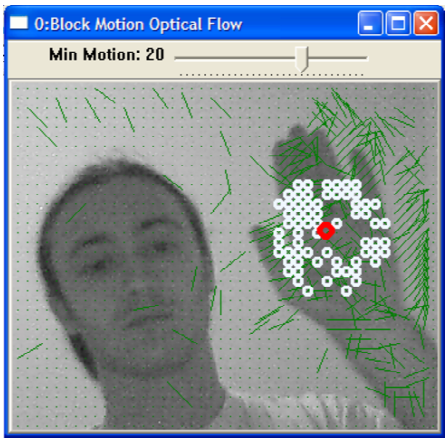

Figure 4. Example tracking #3

4.3. Gesture Recognition

Gesture recognition problem can be described as finding a HMM among the candidates giving the highest probability that the observed sequence is produced by that model. Whenever the hand tracking system reports that a gesture attempt is made, the sequence of centroid coordinates are treated as an observation of an unknown HMM. Although we have a single topology for all our HMMs, they differ in the velues of transition and observation probabilities and can be treated as separate HMMs. Therefore, we search among the trained HMMs for the one that most likely produced the observed sequence. Once all of the HMMs are evaluated, it is concluded that the gesture corresponding to the one with the highest probability of producing the observation sequence is performed. An example gesture attempt can be seen in Figure 5.

In Figure 5, pink pixels are the pixels classified as a part of the marker, pink rectangle is the bounding box, and red circles are the observed trajectory of the centroid.

Figure 5. An example gesture and its corresponding trajectory

5. Experiments and Results

Colored markers are used for the hand detection and tracking to simplify the process and get rid of color ambiguity caused by the face and some similarly-colored background objects. Since the process is color and illumination dependent, the system occasionally lost the track of the hands but the rate was around 1 frame over several hundreds of frames, which is quite reasonable. Some of the words (pants, bicycle, book, bowl, box) used in Starner and Pentlands paper were chosen as the words to be recognized in addition to some primitive geometric shapes such as a triangle, square, and circle.

Using more than 4 states in the HMM representation made it possible to discriminate between figures such as square and circle as well as bowl and box, which are quite similar. Examples of drawn gesture trajectories and trained HMM transition probabilities can be seen in Figures 6 and 7.

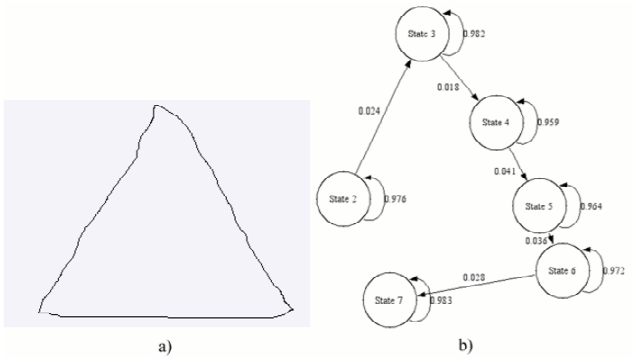

Figure 6. a) An example triangle gesture, b) Trained HMM for triangle

The success ratio of the system using colored markers is not quite satisfactory due to the lighting conditions and dynamic behavior of the environment in contrast to the most of the experiments carried out in the examined work in literature. Lighting related noise affects the stability of the trajectory which makes both the training and evaluation phases so challenging. As a future work, we are planning to use some filtering (for example, KALMAN filter) and combining color based segmentation with optical flow information. The measured success ratio is nearly $\%70$ which is a low ratio compared to the experiments carried out in isolated environments with strict assumptions.

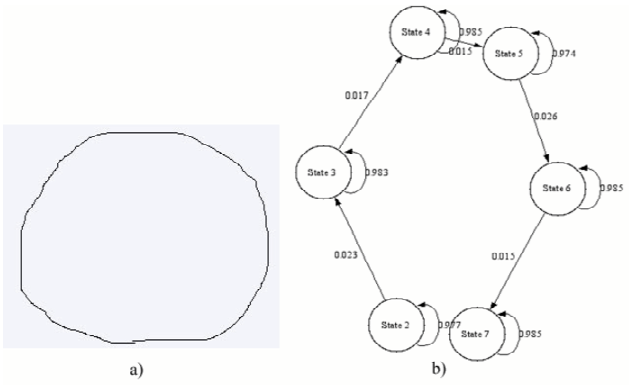

Figure 7. a) An example circle gesture, b) Trained HMM for circle

In order to simplify the experiments, we tracked only one hand with the motion-based tracking system, and used simple gestures which are primitive 2D geometric shapes such as triangle, square, and circle. Due to the high success rate in tracking and the simplicity of the gestures to be recognized, the system was able to recognize and classify the gestures correctly with a success rate of $> \%90$ As opposed to the colored marker-based tracking method, this method provides a color and illumination independent tracking which makes the system much more robust and successful.

6. Discussion and Conclusion

In this paper, a vision-based real-time American Sign Language (ASL) recognition system has been presented. Hidden Markov Models (HMMs), which are proven to provide a good performance in speech and handwriting recognition, are used as the primary tool during the recognition process. The recognition performance can be increased with the increasing number of training examples although the current system has a quite satisfactory recognition performance with a low error rate in case of using motion-based tracking and simple gestures.

Currently, the colored marker-based tracking and recognition system is designed for a small set of signs and the user has to wear colored gloves. Even with colored gloves,

if the illumination is not appropriate, it is not possible to detect the hand. Therefore, one of the possible extensions to the system is to improve it in such a way to perform a robust detection of the hand without any gloves and also without getting mixed with the face region.

For the motion-based tracking case, the system recognizes the gestures in a discrete manner; that is, it waits for the confidence level to exceed some specific threshold value to initiate the recognition process which may prevent it from recognizing continuous gestures. However, this problem can be solved by introducing a regular expression recognition mechanism which can extract a sequence of correctly performed gestures from among independent hand movement. This extension is left as a future work.

In order to increase the number of gestures that can be recognized, the system can be extended to incorporate positions of the hands relative to each respective shoulder or some other fixed point on the body, and finger and palm information - the number of visible fingers along the contour of the hand and whether the palm is facing up or down. Also, explicit face tracking and facial gesture information can be added to the feature set to enrich the capability of the system.

Hand-hand and hand-face ambiguity problems, which are caused by the color segmentation based hand detection approach, can be solved by tracking the hands in 3D instead of in 2D. Another solution would be using optical flow information for masking the image. In such approach, a binary threshold is applied to the optical flow field such that flows having a magnitude less than a certain threshold will be set to zero and flows over that threshold would be set to one. Then, the segmentation process executes in only regions with flow values one. We can expect from this approach to successfully separate a face with a small movement or a people passing through the background with a relatively small velotic from a foreground moving hand with a relatively high velocity.

References

[1] T. Starner and A. Pentland. “Realtime american sign language recognition from video using hidden markov models”. Technical report, MIT Media Laboratory, 1996. 3, 5

[2] 1 - L.Baum “An inequality and associated maximization technique in statistical estimation of probabilistic functions of Markov processes”. *Inequalities*, 3:1-8, 1972 2

[3] 8 - X. Huang, Y. Ariki, and M. Jack “Hidden Markov Models for Speech Recognition” Edinburgh Univ. Press, Edinburgh, 1990 2

[4] 12 - H. Poizner, U. Bellugi, and V. Lutes-Driscoll. “Perception of American Sign Language in dynamic pointlight displays” *J. Exp. Pyschol.: Human Perform.*, 7:430-440, 1981 1

[5] 13 - L. Rabiner and B. Juang. “An introduction to hidden Markov models”. *IEEE ASSP Magazine*, p. 4-16, Jan. 1996 2

[6] 16 - G. Sperling, M. Landy, Y. Cohen, and M. Pavel. “Intelligible encoding of ASL image sequences at extremely low information rates”. *Comp. Vision, Graphics and Image Proc.*, 31:335-391, 1985 1

[7] 17- T. Starner. “Visual Recognition of American Sign Language Using Hidden Markov Models”. Master’s Thesis, MIT Media Laboratory, Feb. 1995 2

[8] 23- S. Young. “HTK: Hidden Markov Model Toolkit V1.5” Cambridge Univ. Eng. Dept. Speech Group and Entropic Research Lab. Inc.,Washington DC, Dec. 1993 2

[9] S.Tamura and S. Kawasaki, “Recognition of sign language motion images”. *Pattern Recognition*, 21:343-353, 1988 3

[10] T. Takahashi and F. Kishino. “Hand gesture coding based on experiments using a hand gesture interface device”. *SIGCHI Bul.*, 23(2):67-73, 1991 3

[11] Hyeon-Kyu Lee and Jin-Hyung Kim. “Gesture spotting from continuous hand motion”. *Pattern Recognition Letters*, 19(5-6):513520, 1998. 3

[12] C. Vogler and D. Metaxas. “Adapting hidden markov models for asl recognition by using three-dimensional computer vision methods”. *In Conference on Systems, Man and Cybernetics (SMC97)*, Orlando, FL, pages 156161, 1997. 3

[13] C. Vogler and D. Metaxas. “Asl recognition based on a coupling between hmms and 3d motion analysis”. *In International Conference on Computer Vision (ICCV98)*, Mumbai, India, 1998. 3

[14] J. Watkinson. “MPEG Handbook”. *Focal Press*, 2001 2

[15] Y. Wang, J. Ostermann and Y. Zhang. “Video processing and communications”. *Prentice Hall, Signal Processing Series*, 2002 3

[16] Ioannis M. Rekleitis. “A particle filter tutorial for mobile robot localization” *International Conference on Robotics and Automation*, 2003 3