# Gesture Recognition using Hidden Markov Models from Fragmented Observations

Ruiduo Yang and Sudeep Sarkar
Computer Science and Engineering Department
University of South Florida
4202 E. Fowler Ave. Tampa, FL 33620
{ryang,sarkar}@csee.usf.edu

## Abstract

*We consider the problem of computing the likelihood of a gesture from regular, unaided video sequences, without relying on perfect segmentation of the scene. Instead of requiring that low-and mid-level processes produce near-perfect segmentation of relevant body parts such as hands, we take into account that such processes can only produce uncertain information. The hands can only be detected as fragmented regions along with clutter. To address this problem, we propose an extension of the HMM formalism, which we call the frag-HMM, to allow for reasoning based on fragmented observations, via the use of an intermediate grouping process. In this formulation, we do not match the frag-HMM to one observation sequence, but rather to a sequence of observation sets, where each observation set is a collection of groups of fragmented observations. Based on the developed model, we show how to perform three kinds of computations. The first one is to decide on the best observation group for each frame, given a sequence of observation groups for the past frames. This allows us to incrementally compute the best segmentation of the hand for each frame, given the model. The second one involves the computation of likelihood of a sequence, averaged over all possible states sequences and possible groupings. The third is the computation of the likelihood of a sequence, maximized over all possible state sequences and group sequences. This can give us the best possible groupings for each frame, as well. We demonstrate our ideas using a publicly available hand gesture dataset that spans different subjects, is against complex background, and involves hand occlusions. The recognition performance is within 2% of that obtained with manually segmented hands and about 10% better than that obtained with segmentations that use the prior knowledge of the hand color.*

## 1. Introduction

Gesture recognition is a rich area of research (see [11, 12] for reviews) with many different applications and approaches. Vision-based approaches share the problem related to the vagaries of low-level segmentation. The states in a state space based gesture representations, such as the Hidden Markov Model [16] or Dynamic Time Warping [5, 3] or, Finite State Machine (FSM) [7] approaches are based on the low-level features detected in the image. Motion tracks in trajectory based gesture recognition approaches [20, 13] are dependent on the robustness of the tracking process, which in turn, is dependent on the stability of the low-level segmentation. Particle filtering[15] and shape model[17] can be used for hand tracking with noisy environment, however these methods requires initialization and hand can still be lost or occluded. This problem of low-level segmentation is sometimes addressed by engineering the imaging setup so as to ease the segmentation of hands by using controlled lighting, colored gloves or even non-vision based aids such as magnetic or optical markers. Pure vision-based solutions usually rely on skin color and/or motion information to detect hands. However, approaches based on predefined skin color models suffer from sensitivity with respect to changing illumination conditions. Motion-based hand segmentation approaches rely on the assumption that the features important for gesture will be associated with motion. Fusion [2] or multi-modal [6] approaches can be used to arrive at better segmentations. However, segmentation will never be perfect; not only will there be missed detections, but there will also be false alarms. There is danger that these errors are propagated to the recognition stage. In this work, we advocate using an intermediate grouping module, coupled with the recognition module, to handle low-level segmentation errors. Such grouping processes have been found to be useful for object recognition tasks, but have not been used for gesture recognition.

The model for each gesture is in terms of an HMM,

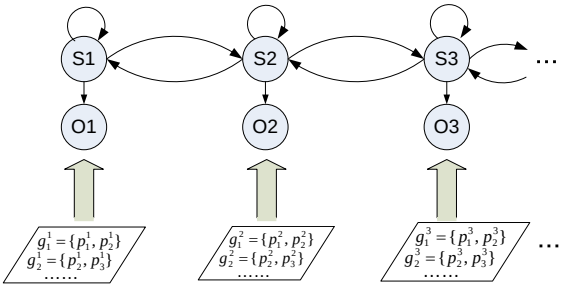


Figure 1. In frag-HMM, we do not have an unique observation sequence to match, rather we have a collection of possible observation sequences, implied by the sequence of multiple observations at each frame.

which is trained based on semi-manually specified, near perfect, segmentations of the hand. While the structure and the training of the HMM is a fairly standard one, the decoding process, i.e. computing the likelihood of an image sequence to the HMM, is novel. During matching we do not insist on excellent segmentation to produce an unique observation sequence of the hand. Rather, we start with segmentation of the image into a collection of fragmented regions. These region fragments are our primitives that are grouped into possible hand hypotheses using a greedy search technique, starting from multiple seed primitive patches that are selected based on size. Unlike for segmentation, these groups are not constrained to be disjoint. The generated groups are then associated across adjacent frames, based on shape, size, and location similarity, to result in sequence of linked group sets. Finally we match each gesture HMM to this linked group structure to simultaneously compute the matching score and the best possible grouping for each frame. *We refer to this extension of the HMM approach to handle fragmented observations as the frag-HMM.* (Fig. 1).

Previous approaches to gesture recognition using HMM also consider the problem of noisy observations. For example, Wilson *et al*. [18] use parametric hidden Markov Model to model the variation across the gesture family. Lan *et al*. [19] propose a two-dimensional spatio-temporal modeling approach that handles both self-occlusions and changes in viewpoint. Kettebekov *et al*. [10] used speech cue to overcome the errors of image signal. Our methods differ with these ones in that we allow for multiple, overlapping, observation hypotheses for each frame. Drastically noisy observations, i.e. significant deletions and additions of patches, are allowed at segmentation level to reduce the probability that the true observation has been lost.

The combination of top-down and bottom-up approach in gesture sequence recognition can be found in [14] and [1]. Although these approaches can handle multiple candidate observations, there are no grouping process incorporated. For example, a sliding window is used along with skin color model in both [1] and [14] to obtain the position of the moving hands. However, in real world application, bad lighting conditions may cause problems for skin color approaches, and a sliding window cannot be sufficient in some applications where exact hand shape are needed.

We demonstrate our approach using a publicly available hand gesture dataset collected by another research group. It is a two-view hand gesture dataset that was recently collected by Just and Marcel [9]. This dataset has images of hand gestures against a complex background, which makes hand segmentation hard with the use of the knowledge of the hand color. Although the dataset has been collected with colored gloves, we do *not* use the color information for each hand to construct a hand color model. We obtain encouraging results without relying on near-perfect hand segmentation or tracking of hand.

The rest of the paper is organized as follows. Section 2 describes the grouping process. The recognition algorithm is described in Section 3. Experimental results are shown in Section 4. We conclude with Section 5.

## 2. Grouping of Low-level Primitives

Low level processes are never perfect. Skin color is the most commonly used cue for segmenting image parts from the hand or face in gesture analysis. However, this does not always produce perfect segmentation, with over-segmentation being a particularly hard problem to handle. To help overcome this problem of over segmentation, we use an intermediate grouping process. The goal of this process is to construct groups of low-level image primitives that most likely are from the hands. So as not to short-change the subsequent recognition process by insisting on disjoint groups, as is usually the practice in grouping, we allow for overlapping groups, resulting in redundant sets of groups. Redundancy should help us counter grouping errors. Some region patches are selected as seeds based on its size. We then grow these seeds with adjacent regions to generate larger groups. As the seeds are grown, groups are checked for being possible hands based on size and shape. Grouping can be conducted based on color, position, boundary smoothness or boundary gradient. These basic similarity cues resemble those adopted by Hoogs and Mundy to group region patches [8] for object recognition, where they used spatial intensity, parallelism and perimeter to form a object hypotheses. However, unlike them we perform the grouping based on each criteria independently of other. Each criteria results in a set of groups, which we refer to as a grouping layer. Thus we have grouping layers color grouping layer, proximity grouping layer and boundary smoothness grouping layer.

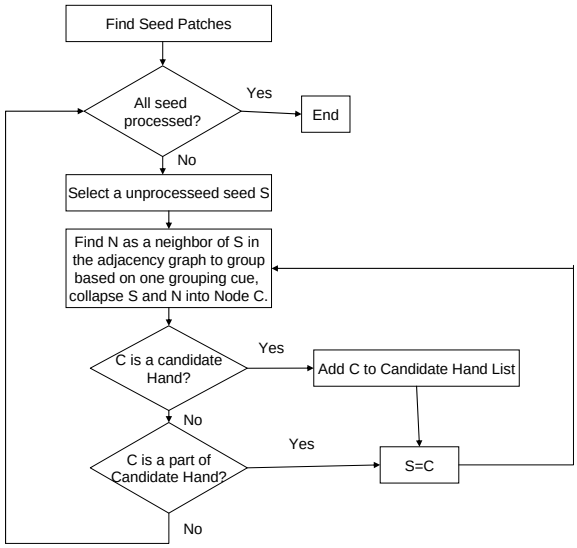


Figure 2. Flow chart of the grouping process used to generate each grouping layer.

### 2.1. Grouping Process

The low-level primitives of the grouping process are constant color (or intensity for gray level images) region patches. The nature of the algorithm used to detected these patches are not particularly important. We used the mean shift segmentation algorithm [4], which is fast and effective, to generate these patches based on color or intensity. Let the set of low-level primitives detected in the $k$-th image frame be denoted by $\{p_1^k, \cdots, p_{N_k}^k\}$. A grouping, of these region primitives will represent a subset of these primitives, $\{p_{i_1}^k, \cdots, p_{i_n}^k\}$.

We adopt a greedy approach to form the groups, outlined in the flowchart in Fig. 2. Let us denote the low-level region patches in the $k$-th image frame by $S_k = \{p_1^k, \cdots, p_{N_k}^k\}$. From this initial set of primitives, we select a subset of primitives that are likely to come from hand, based on the size of the patch. These are our seed patches. Given some knowledge of approximate size of hands in the sequence, we can eliminate large, non-homogeneous region patches from further consideration. We use a list $L$ to store the possible groups. This list is initialized by choosing each selected primitive to be a singleton group. These groups would be merged to form larger conglomerate.

$$L = \{\{p_x^k\} | a_s(p_x^k) \leq t_{size}, x = 1, \cdots, N_k\} \quad (1)$$

Here $a_s$ is the operator that returns the size of $p_x^k$. For the entries in $L$, we maintain an adjacency graph, whose nodes are the groups in $L$, and links exist between groups that share a boundary. This graph is incrementally updated at each iteration.

The grouping process starts by picking the first group in $L$, denoted here by $p$, and searches its neighbors $\{N_p^i\}$. Each neighbor $N_p^i$ is considered for grouping with $p$ to generate a tentative larger grouping. We select the best local grouping, and denote it as $g$.

The group $g$ is further tested to see if can possibly represent a hand. This test is based on three attributes: $[a_n, a_s, a_{cur}]$, where $a_n$ is the number of primitives in the group, $a_{cur}$ is the boundary curvature of the group, $a_s$ is the size of the bounding box.

$$(a_s \leq t_{size}) \wedge (a_{cur} \leq t_{curvature}) \wedge (a_n \leq t_{num}) \quad (2)$$

The test is conducted based on the result of Eq. 2, where $t_{size}, t_{curvature}, t_{num}$ are the corresponding thresholds. Here the boundary curvature is approximated as the integral of the squared root of second order derivative along the curve. If the group $g$ passes this test, it is inserted in into the final candidate group list, $C$, else if $a_s \leq t_{size}$ it is inserted at the end of the list $L$, to be considered for further grouping.

Note that the low-level primitives and the groups are formed on a frame by frame basis; There is no tracking or frame-to-frame correspondence. Also note that we do *not* restrict ourselves to disjoint groups; This is different from the usually employed disjoint groups constraint employed in segmentation and grouping. Allowing for overlapping groups allows us to avoid making hard decisions about group boundaries.

### 2.2. Associating Groups Across Frames

We denote the $j$th group detected in $t$th frame as $G_t^j$. The groups detected in each frame are associated with those detected in previous frames to result in a linked sequence of groups spanning all the frames. This structure will help us propagate constraints during the matching process and restrict considering exponentially large number of possible observation sequences. We define the predecessors set of each element in each groups set as

$$Pre(G_t^j) = [G_{t-1}^{j_1}, \cdots, G_{t-1}^{j_n}], t \geq 2, 1 \leq j_k \leq c_{t-1}, \quad (3)$$

where $G_{t-1}^{j_k}$ is one possible predecessor of $G_t^j$. The predecessor relationship between the groups from different time instants is based on feature similarity. It captures how likely the groups are from the same underlying cause in the image. Specifically, we test the difference in feature size and location between the two groups, with a liberally chosen threshold value.

## 3. Recognition Algorithm

While the structure and the training of the frag-HMM is a fairly standard one, the decoding process, i.e. computing

the likelihood of an image sequence to the frag-HMM, is significantly different and new. Each gesture $g_i$ is modeled using an frag-HMM $\lambda_i$ over $N$ states. The state at time $t$ is denoted as $q_t$, where $q_t \in 1, \cdots, N$, $a_{ij} = P[q_{t+1} = j | q_t = i]$ is the state-transition matrix. The initial state distribution is denoted as $\pi = \pi_i$, where $\pi_i = P[q_1 = i]$ is the probability that state is $i$ at $time = 1$. The observation probability is modeled as a mixture of Gaussian, the observation vector is denoted as $O = [O_1, \cdots, O_T]$ with $T$ to be the length of $O$, its probability at state $j$ is computed as $b_j(O) = \sum_{k=1}^{M} c_{jk} \Omega(O, \mu_{jk}, \sigma_{jk})$, where $\Omega$ is a Gaussian with $\mu_{jk}$ as the mean vector and $\sigma_{jk}$ as the covariance matrix, $c_{jk}$ is the mixture factor and $M$ is the number of mixture components. At training, we have observation sequences $O = O_j, j = 1, \cdots, K$, the above parameters $[a_{ij}, \pi_i, c_{jk}, \mu_{jk}, \sigma_{jk}]$ is found to maximize the likelihood $P(O|\lambda)$. We use the Baum-Welch estimation process to train the frag-HMM.

The decoding or matching process is radically different from conventional HMMs. In conventional HMM, the actual state sequence is unknown, but the observation sequence is unique. However, in vision gesture application, we consider the observation sequence to be non unique. In conventional HMM, the input observation feature vector $O = [O_1, \cdots, O_T]$ is known for each frame and the likelihood $P(O|\lambda)$ can be computed using an iterative forward pass process. In our framework , however, we do not assume that we know the exact observation vector $O_t$ at each time $t$, instead, we allow for multiple hypotheses about the observation. At time $t$ we have the group sets $G_t = [G_t^1, \cdots, G_t^{c_t}]$, where each element in $G_t$ is one possible observation and $c_t$ denotes the total number of groups in time $t$. We assume only one element in the observation set is the true observation. We do not decide upon the best group for each frame independently of the others. The entire sequence of group sets is used as the input. We will discuss the problem related to the optimal observation sequence and proposed 3 approaches to compute the matching score with such an input.

### 3.1. Maximal Observation, Summed State

We are given a sequence of group sets $G =< G_1, \cdots, G_T >$, where $G_t = [G_t^1, \cdots, G_t^{c_t}], 1 \le t \le T$ is the group set at time $t$. The optimal observation sequence problem is to find one groups sequence $\psi$ that maximize the likelihood, summed over possible HMM state transitions, $P_{sum}(\psi|\lambda)$, where $\lambda$ is the frag-HMM and

$$\psi =< \psi_1, \cdots, \psi_T >, \psi_i \in G_i, 1 \le i \le T, \psi_{t-1} \in Pre(\psi_t) \quad (4)$$

We denote the maximum value of likelihood probability by

$$P_{\max,sum}(G|\lambda) = \max_{k=1,\cdots,K} P_{sum}(\psi^k|\lambda) \quad (5)$$

where the possible sequence of groups are $\psi^1, \cdots, \psi^K$. The probability $P_{sum}(\psi|\lambda)$ represents the likelihood of the group sequence, *summed* over all possible frag-HMM state sequence. For each sequence of groups the computation of $P_{sum}(\psi^i|\lambda)$ can be done using the standard forward-backward algorithm used for HMMs.

A brute force solution for Eq. 5 will be to enumerate across the sets $G_1, \cdots, G_T$ to get all possible observation sequences $\psi^1, \cdots, \psi^K]$, compute likelihood for each of the observation sequence, and select the maximum value. Obviously, exhaustive enumeration is computationally expensive, hence we resort to approximation based on incremental construction of the optimal sequence.

To find the best group at time $t$, suppose the observation sequence at time $1, \cdots, t-1$ has been recovered as $\psi_1, \cdots, \psi_{t-1}$. We define the indexed forward variable $\alpha_t^j(i)$ as:

$$\alpha_t^j(i) = P(\psi_1, \cdots, \psi_t, q_t = i, \psi_t = O_t^j | \lambda) \quad (6)$$

that is, the probability of the partial observation sequence $< \psi_1, \cdots, \psi_t >$, at time $t$ the state is $i$ and the observation vector is $O_t^j$, and $< \psi_1, \cdots, \psi_{t-1} >$ is the observation vectors we have found at time $1, \cdots, t-1$

The initialization of the variable is:

$$\alpha_1^j(i) = \pi_i b_i(G_1^j); \quad (7)$$

and we have

$$\psi_1 = G_1^p, p = \arg\max_j \sum_{i=1}^{N} \alpha_1^j(i) \quad (8)$$

The induction solution is

$$\alpha_{t+1}^j(i) = [\sum_{k=1}^{N} \alpha_t^p(k) a_k^i] b_i(G_{t+1}^j), \psi_t = G_t^p \quad (9)$$

and then $\psi_{t+1}$ is selected as:

$$\psi_{t+1} = G_{t+1}^p, p = \arg\max_j \sum_{i=1}^{N} \alpha_{t+1}^j(i) \quad (10)$$

At time $T$, the observation vector sequence is computed as $< \psi_1, \cdots, \psi_T >$. At the same time, the probability of this observation sequence given the frag-HMM, can be computed as

$$P(< \psi_1, \cdots, \psi_T > | \lambda) = \max_j \sum_{i=1}^{N} \alpha_T^j(i) \quad (11)$$

Fig. 3 illustrates us the indexed forward process. The summation of the product of the forward variables and the observation probabilities remain the same as in the conventional HMM. The difference is that we take the observation

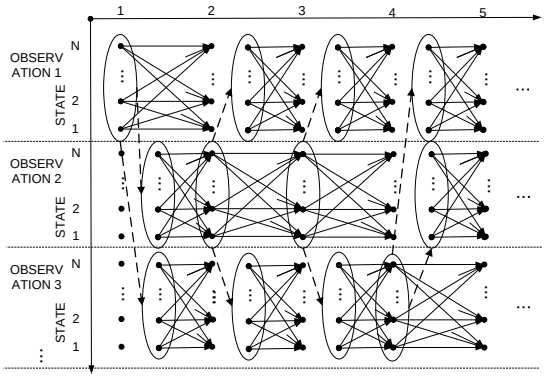


Figure 3. An illustration of the indexed forward process, the horizontal line represent the time, the vertical line correspond to the candidate observations and the sub-vertical line denotes the $N$ states. Note at each time step, only one best observation is selected based on the previous selected observations and the forwarding results. In this example, the optimally selected observations (circled ones) are $< 1, 2, 2, 3 >$.

vector dynamically depend on the previously decided observations. Note the result of Eq. 11 is not an exact solution for Eq. 5, instead it is the solution to select the best current observation based on a certain selected partial observation sequence.

### 3.2. Summed Observation, Summed State

Instead of considering the maximum probability over all possible group sequences, we could consider the summation over all possible group sequences. Thus, the probability of interest is.

$$P_{sum,sum}(G|\lambda) = \sum_{k=1,\cdots,K} P_{sum}(\psi^k|\lambda) \qquad (12)$$

where the possible sequence of groups are $\psi^1, \cdots, \psi^K$. The probability $P_{sum}(\psi|\lambda)$ represents the likelihood of the group sequence, *summed* over all possible frag-HMM state sequence. As before, for each sequence of groups the computation of $P_{sum}(\psi^i|\lambda)$ can be done using the standard forward-backward algorithm used for HMMs. However, we found the process of summing over all group sequence and over all state sequence can be effectively merged in the dynamic programming process. To do this, we defined the grouping forward variable $\kappa_t^j(i)$ as:

$$\kappa_t^j(i) = \sum_{\psi_1,\cdots,\psi_{t-1}} P(\psi_1, \cdots, \psi_t, q_t = i, \psi_t = G_t^j|\lambda) \qquad (13)$$

that is, the summation of the partial probability of all the group sequences that have $\psi_t = G_t^j$ and $q_t = i$. The initialization is

$$\kappa_1^j(i) = \pi_i b_i(G_1^j); \qquad (14)$$

The induction is

$$\kappa_{t+1}^j(i) = [\sum_{p \in Pre(G_t^j)} \sum_{k=1}^{N} \kappa_t^p(k) a_k^i] b_i(G_{t+1}^j) \qquad (15)$$

And, the result of Eq.12 is obtained at the end of the process:

$$P_{sum,sum}(G|\lambda) = \sum_{p \in Pre(O_t^j)} \sum_{k=1}^{N} \kappa_T^p(k) \qquad (16)$$

### 3.3. Maximal Observation, Maximal State

The third quantity of interest is maximum probability over all possible group sequences *and* frag-HMM state sequences. Thus, the probability of interest is.

$$P_{\max,\max}(G|\lambda) = \max_{\psi_1,\cdots,\psi_T} \max_{q_1,\cdots,q_T} P(\psi_1, \cdots, \psi_T; q_1, \cdots, q_T|\lambda) \qquad (17)$$

where the possible sequence of groups are $\psi^1, \cdots, \psi^K$ and $q_1, \cdots, q_T$ is a frag-HMM state sequence. This quantity can again be computed using dynamic programming. We define the max-forward variable $\zeta^j(i)$ as:

$$\zeta_t^j(i) = \max_{\psi_1,\cdots,\psi_{t-1}} P(\psi_1, \cdots, \psi_t, q_t = i, \psi_t = G_t^j|\lambda) \qquad (18)$$

This is the maximum partial probability among all the group sequences that have $\psi_t = G_t^j$ and $q_t = i$. The variable $\xi_t$ represents the backtrack index of the observations for the corresponding max-backward process. The initialization is:

$$\kappa_1^j(i) = \pi_i b_i(G_1^j) \qquad (19a)$$

$$\xi_1 = 0 \qquad (19b)$$

The induction is given by

$$\kappa_{t+1}^j(i) = [\max_{p \in Pre(G_t^j)} \max_{k=1}^{N} \kappa_t^p(k) a_k^i] b_i(G_{t+1}^j) \qquad (20a)$$

$$\xi_t = \arg \max_{p \in Pre(G_t^j)} \max_{k=1}^{N} \kappa_t^p(k) a_k^i \qquad (20b)$$

$\xi_1, \xi_2, ..., \xi_T$ is obtained the best group sequence (over the best state sequence) and this group sequence can be used to get the matching score.

### 3.4. Occluded Observations

Occlusion is a hard problem in traditional tracking-based and hand color-based approaches. One of the advantages of the proposed approach is that we can handle the occluded observations in a natural way. For each frame we have a dummy group representing the potential occluded group. This group is linked to all the groups in the previous and

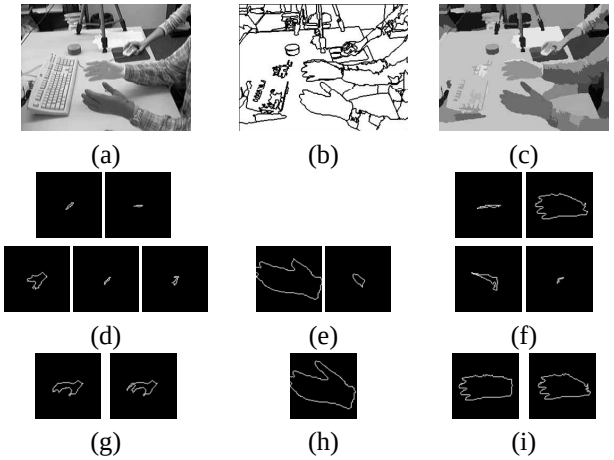

Figure 4. HCI dataset results. Candidate groups of regions generated for some frames. Notice there are 3 hands in the frame. (a) Original frame; (b) segmented image(boundary); (c) segmented image; (d) primitives around the third hand; (e) primitives around left hand; (f) primitives around the right hand; (g) the candidate groups for the third hand; (h) the candidate groups for the left hand; (i) the candidate groups for the right hand.

following frame, which means at any time the true observation can be occluded and then appear in the scene. During training, the observation probability of the occluded group is selected to be between that for a perfectly segmented hand and very noisy hand patch.

## 4. Experiments with Hand Gesture Sequences

We present results on the publicly available Human Computer Interaction (HCI) dataset that has been recently collected by another research group, i.e. Just and Marcel [9]. The dataset is for recognizing 7 hand actions: push, rotate front, rotate back, rotate left, rotate right, rotate up, and rotate down. The authors of the data has explicitly separated the training and test data, where the training data consist of 4 subjects, each of whom performed the 7 actions 10 times, with 5 of them at one session time and 5 of them at the other, and the test data has the same shots but with 3 different subjects. The total number of test sequences is 210. The dataset has shots from 2 fixed camera, one shot from the left side and the other shot from the right side. We used the joined results of the two views in this paper.

Since this dataset was collected with yellow and blue colored gloves, it allows us to make comparisons with color-based hand segmentation schemes. As baseline performance comparison, we consider (i) manually segmented hands, and (ii) hands segmented using the information about the color of the gloves. For color based hand segmentation, each glove color is modeled as mixture of 3 Gaussians in the color space. For the proposed approach, we consider just region segmentation patches, detected as outlined earlier. Note that although we use color for segmentation and grouping, we do not used the knowledge that a specific color corresponds to the hand. Fig. 4 shows examples of region segmentation and groups forms. Note that some hypotheses corresponds to non-hand parts of the image or for other hands that might be present. At grouping, the threshold values are manually assigned a large value so that the chances of losing the real hand group is reduced. We set the threshold as: $t_{size} = 200$(in pixels), $t_{num} = 20$, then we sort the hand groups by their curvature score and select the top 500 groups. For each frames we generated around 100 groups that are candidate hands. We selected the position, orientation of major axis, and aspect of the major and minor axes of the group as the feature vector representing the observations in the frag-HMM. We consider recognition with each of the three probabilistic measures outlined earlier. The correct recognition rates are shown in Fig 5. The 5 approaches – the two baseline and the three frag-HMM ones, give us the recognition rates: 79%, 94%, 91%, 92%, and 91%. From this result we can see:

1. For each frame, above 95% of the groups generated were noisy, with some being just random patches. However, their contribution to the final overall sequence is quite small, since they were not well linked across frames. Our approach allows us to recover from such errors. However, for the commonly used color-based hand segmentation approach, if any one frame has noisy hands, the recognition might fail. This is reason why the recognition with hands segmented using just color information results in low performance.

2. Our approach that accommodates imperfect segmentation is within 2% of recognition possible with manual segmentation.

Fig. 6 and Fig. 7 show the recognition rate on a per-gesture and per-subjects basis. We can see the majority of errors comes from one subject and the three gestures that can be easily mixed up. Subject 1 performed each gesture with larger motion than the other subjects in the training data. Such a case is hard to improve by using only the position features, hence subject 1 produced majorities of the errors. Among the gestures, Rotate Front, Push and Rotate Right all have motions moving forward and backward; there are only subtle orientation change in the palm. Hence these actions produced majorities of the errors. However, the performance measure of interest for this work is how well the recognition rate with fragmented observation match that with perfect segmentation. On this account, the performance is quite strong.

Fig. 8 shows visual example of the optimal groups selected for the best match corresponding to the Rotate Back

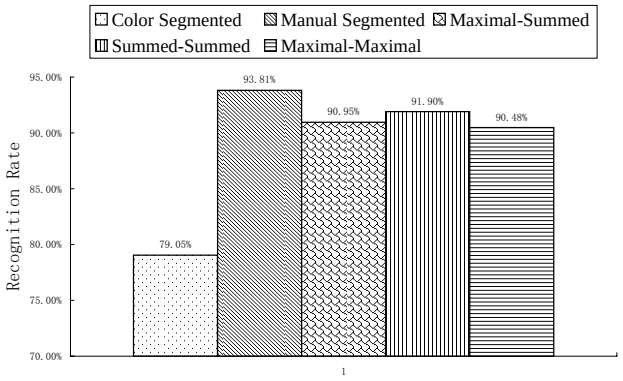


Figure 5. Recognition performance of 210 instances of 7 hand gestures for five different approaches. The first two are based on manual and color based segmentation of the hands. The next three does not use the knowledge of the hand color and take into account fragmented observations. The three corresponds to the three different kinds of probabilities that can be computed, $P_{\max,sum}$, $P_{sum,sum}$, and $P_{\max,\max}$ using the frag-HMM proposed in this paper.

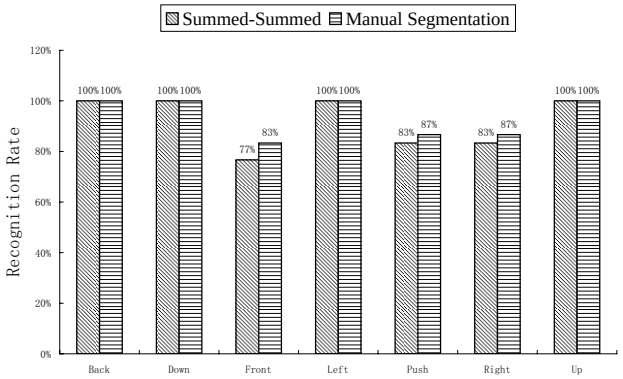


Figure 6. Recognition performance of each hand gestures, using Summed-Summed approach and Manual Segmentation.

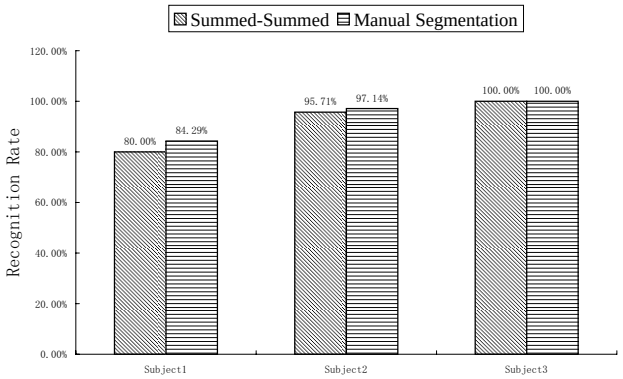


Figure 7. Recognition performance of each subjects separately, using Summed-Summed approach and Manual Segmentation.

action. There are two parts to the movement, backwards and forwards. Fig. 8(a) and (b) shows the selected groups

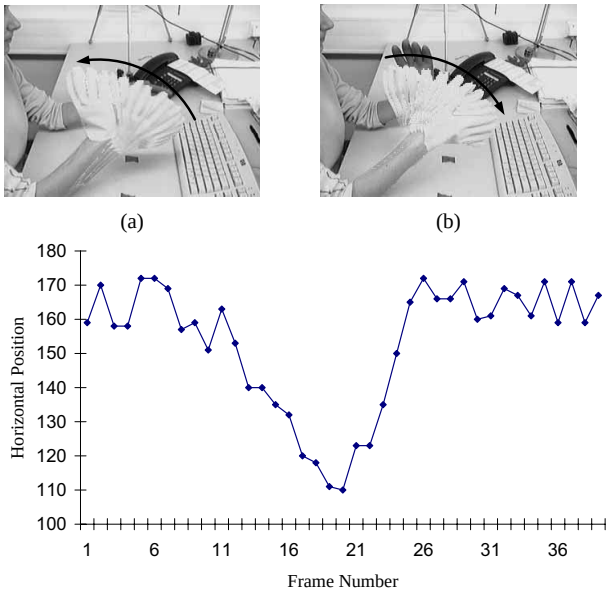


Figure 8. The optimal groups corresponding to one of the hands, as discovered, using the frag-HMM, for the (a) first part and (b) second part of the “Rotate Back” gesture. (c) The computed horizontal position of the hand.

for these two parts overlaid on each other. Fig. 8(c) shows the X (horizontal) coordinates of the revealed hand by using the optimal state and sequence pair approach, we can see the nature of the change of X coordinates match the hand positions. The indexed forward approach produce similar result.

Fig 9 shows us some results for the hand occlusions that exist for the ”Rotate Down” sequence, where the right hand moves down first, gets occluded by the left hand, and then moves up. The chosen hand group using the optimal state and group sequence method is shown for frames, before, during, and after the occluding event. We see that during occlusion no groups are selected for the right hand as it is not visible.

## 5. Conclusion

We proposed a new framework for gesture recognition from video that does not rely on a predefined color models and can work with imperfect segmentation of scenes. We addressed the hard problem of hand segmentation by coupling it with recognition, via an intermediate grouping process. The grouping process generated layers of overlapping groups that are linked across time in a graph structure. The recognition is based on an extension of the HMM, we call as the frag-HMM, which accounts for fragmented observations. We shows how three different kinds of probabilities can be computed using the frag-HMM, based on

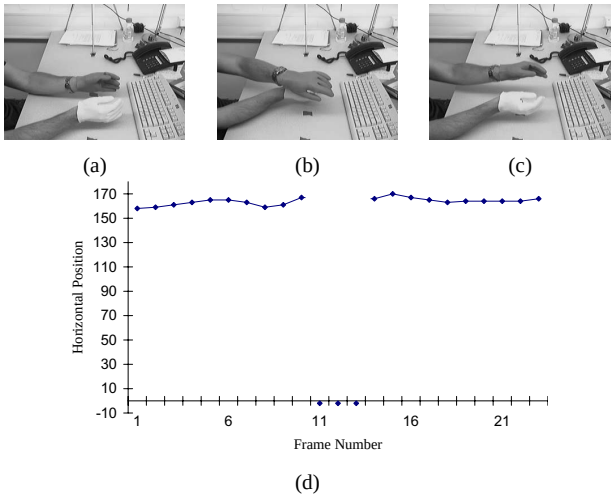


Figure 9. Ability to handle occlusions. Three frames from the "Rotate Down" gesture is shown in (a), (b), and (c). The estimated horizontal position of the hand is shown in (d). Note that there are no estimates for the frames for which there was occlusion, which shows that the best group was chosen correctly even across occlusions.

maximization and averaging over the underlying states and groups. We demonstrated its efficiency for HCI hand action recognition tasks using a publicly available dataset spanning multiple subjects and actions, against complex backgrounds. The recognition rates were very close (within 2%) of those achieved by manual segmentation and much better (by about 10%) than that achieved by color based hand segmentation. We hope that this work pushes gesture recognition into the real world domain, where segmentation is never perfect.

## 6. Acknowledgment

This work was supported in part by the National Science Foundation under grant IIS 0312993. Any Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

## References

[1] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. Simultaneous localization and recognition of dynamic hand gestures. In *WACV/MOTION*, pages 254–260, 2005.

[2] Y. Azoz, L. Devi, M. Yeasin, and R. Sharma. Tracking the human arm using constraint fusion and multiple-cue localization. *Machine Vision and Applications*, 13(5-6):286–302, 2003.

[3] A. Bobick and A. Wilson. A state based approach to the representation and recognition of gesture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1325–1337, December 1997.

[4] D. Comanicu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, May 2002.

[5] T. Darrell and A. Pentland. Space-time gestures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 335–340, 1993.

[6] H. Graf, E. Cosatto, D. Gibbon, M. Kocheisen, and E. Petajan. Multimodal system for locating heads and faces. In *In Proceedings of the Second IEEE International Conference on Automatic Face and Gesture Recognition*, pages 88–93, 1996.

[7] P. Hong, M. Turk, and T. Huang. Gesture modeling and recognition using finite state machines. In *Proc. Fourth IEEE International Conference and Gesture Recognition*, pages 410–415, 2000.

[8] A. Hoogs and J. Mundy. An integrated boundary and region approach to perceptual grouping. In *International Conference on Pattern Recognition*, pages Vol I: 284–290, 2000.

[9] A. Just and S. Marcel. Two-handed gesture recognition. Technical report, IDIAP Research Institute CH-1920, Martigny, Switzerland, 2005.

[10] M. S. R. Kettebekov, S.; Yeasin. Improving continuous gesture recognition with spoken prosody. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE*, volume 1 of *18-20*, pages I–565 – I–570, 2003.

[11] G. Konstantinos. A review of vision-based hand gestures. http://www.cvr.yorku.ca visisted at Jun 2005, 2004.

[12] V. Pavlovic, R. Sharma, and T. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, July 1997.

[13] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2), 2002.

[14] Y. Sato and T. Kobayashi. Extension of hidden markov models to deal with multiple candidates of observations and its application to mobile-robot-oriented gesture recognition. In *ICPR (2)*, pages 515–, 2002.

[15] C. Shan, Y. Wei, T. Tan, and F. Ojardias. Real time hand tracking by combining particle filtering and mean shift. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 669–674, 2004.

[16] T. Starner and A. Pentland. Real-time American Sign Language recognition from video using Hidden Markov Models. In *Symposium on Computer Vision*, pages 265–270, 1995.

[17] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *Proc. 9th International Conference on Computer Vision*.

[18] A. Wilson and A. Bobick. Hidden Markov Models for modeling and recognizing gesture under variation. *Intl J. of Pattern Recognition and Artificial Intelligence*, 15(1):123–160. 2001.

[19] L. X. and H. D. A unified spatio-temporal articulated model for tracking. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE*, volume 1, pages I–722 – I–729, 2004.

[20] M. Yang, N. Ahuja, and M. Tabb. Extraction of 2D motion trajectories and its application to hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1061–1074, 2002.