# Exemplar-based Tracking and Recognition of Arm Gestures

Ahmed Elgammal    Vinay Shet    Yaser Yacoob    Larry S. Davis

Computer Vision Laboratory
University of Maryland
College Park, MD, 20742 USA

## Abstract

*This paper presents a probabilistic exemplar-based framework for recognizing gestures. The approach is based on representing each gesture as a sequence of learned body poses. The gestures are recognized through a probabilistic framework for matching these body poses and for imposing temporal constrains between different poses. Matching individual poses to image data is performed using a probabilistic formulation for edge matching to obtain a likelihood measurement for each individual pose. The paper introduces a correspondence-free weighted matching scheme for edge templates that emphasize discriminating features in the matching. The weighting does not require establishing correspondences between the different pose models. The probabilistic framework also imposes temporal constrains between different pose through a learned Hidden Markov Model (HMM) of each gesture.*

## 1 Introduction

The recognition of arm and hand gestures has many applications in human computer interaction, virtual reality and in robotics. Our objective is to recognize arm gestures performed by a human standing at a distance from the camera. This might be to operate a robot or a vehicle driven by a robot or generally to control an environment using gestures. This paper presents a prototype system for view-based recognition of gestures. In particular, the approach is designed to operate a robot driven vehicle by recognizing human arm signaling. The paper presents an exemplar-based approach where each gesture is represented as a sequence of learned body poses through a probabilistic framework for matching these body poses to the the image data. The probabilistic framework also imposes temporal constrains between different pose through a learned Hidden Markov Model (HMM) of each gesture. Matching individual poses is performed using a probabilistic formulation for Chamfer matching to obtain a likelihood measurement for each individual pose. The paper introduces a weighted matching scheme for edge templates that emphasize discriminating features in the matching. The weighting does not require establishing correspondences between the different pose models.

Different approaches have been proposed for gesture recognition. These approaches can be classified into three major categories: model-based, appearance-based, and motion-based. Model-based approaches focus on recovering three-dimensional model parameters of articulated body parts [13, 6]. Appearance-based approaches uses two dimensional information such as gray scale images or body silhouettes and edges. In contrast, motion based approaches attempt to recognize the gesture directly from the motion without any structural information about the physical body, for example [1]. In all these approaches, the temporal properties of the gesture are typically handled using Dynamic Time Warping (DTW) or statistically using Hidden Markov Models (HMM).

The paper is organized as follow. Section 2 gives an overview of the proposed system. Section 3 presents the gesture tracking framework. Section 4 presents the proposed pose classification approach. Section 5 describes the hidden Markov model used for gesture recognition. Section 6 illustrates some experimental results.

## 2 System Overview

Figure 1 describes an overview of the proposed system. The system consists of three modules: Training, Segmentation and Tracking, and Gesture Recognition. The training module learns models for body poses. It also learns temporal models of each individual gesture as a sequence of the learned body poses through a Hidden Markov Model (HMM). Finally the training module learns models of the activities, i.e, the different gesture that can be recognized and their temporal relations. The segmentation and tracking module continuously segment and track the person performing the gesture from the rest of the background. The segmentation from the background is performed using coarse range data through a series of plane fitting to the range data and a rule based system to determine which plane in the range corresponds to the person. The range data is registered to the video data, therefore locating the person in the range image locates the person in the video data. Since the segmentation is performed using a very coarse range data, the output of this module is just the location of the person in the image as a rectangle. The quality of the range data, in terms of accuracy, is *not* enough to provide fine silhouette segmentation or to do gesture recognition. For an example

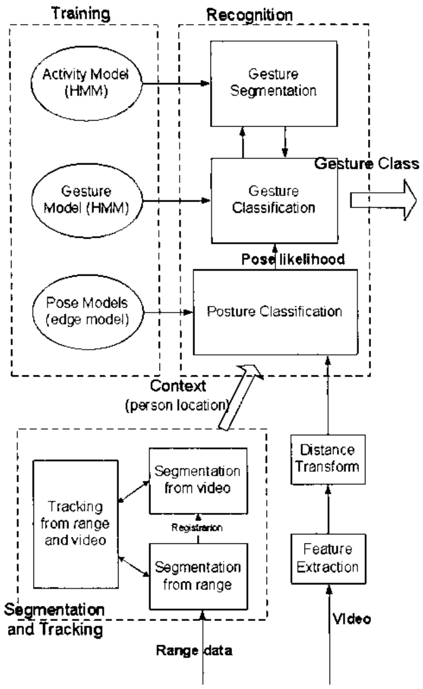

**Figure 1. System Overview**

of the quality of the result of the segmentation, see figure 8. The system is supposed to be mounted on a moving vehicle; therefore the person is continuously tracked in both the range and the video to provide the context information necessary for the recognition module.

The gesture recognition module uses only the video data. It matches learned silhouette models through coarse to fine search around the person location, provided by the segmentation and tracking module, to register the learned poses to the video data. The recognition module, then, matches all learned pose models to each new image to obtain pose probability likelihoods. The gesture classification part uses the learned HMM of each gesture to impose temporal constrains on the body poses and therefore determine the gesture class. The gesture recognition module also uses an HMM activity model to determine the beginning and the end of each gesture (gesture segmentation).

## 3 Exemplar-based model

We use the definition of [3, 2]: An exemplar space is specified by a set of "exemplars", $\mathbf{X} = \{x^k, k = 1 \cdots K\}$, containing representatives of the training data, and a distance function, $\rho$, that measures the distortion between any two points in the space. The work of [2] was a major step towards learning probabilistic models based on exemplars as centers of a probabilistic mixture. The work of [16] was another major step that introduces the use of exemplars in a metric space within the same framework.

Figure 2 shows the probabilistic graphical model for our exemplar-based tracking of arm gesture. The models used in this paper differs from that used in [2] and [16] in a principle way: We use a model where there is no coupling between the states and the exemplars. At each discrete time, $t$, the system state is denoted by the pair $(q_t, \alpha_t)$ and $x_t$ denotes the exemplar at time $t$. The hidden variable $q_t$, representing a Markov stochastic process, can take any value from a set of $M$ distinct abstract states, $S = \{s_1, s_2, \cdots, s_M\}$, . The R.V. $x_t$ can be any exemplar from the set of exemplars $\mathbf{X} = \{x^k, k = 1, \cdots K\}$. The observation $z_t$ at time $t$ is considered to be drawn from a probabilistic mixture, i.e., $z \approx T_\alpha \tilde{x}_k$ where $\{\tilde{x}_k, k = 1 \cdots K\}$ are the exemplars and $T_\alpha$ is a geometric transformation with parameter $\alpha$. The system dynamics is defined by the transitions $P(q_t|q_{t-1})$ and $P(\alpha_t|\alpha_{t-1})$.

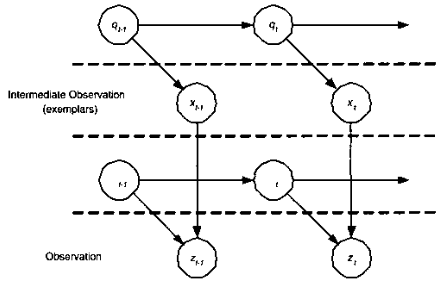

**Figure 2. Graphical model for gesture tracking**

The observation $z_t$ at time $t$ is a probabilistic mixture from all the exemplars and can be calculated using

$$P(z_t|q_t, \alpha_t) = \sum_{k=1}^{K} P(z_t|x_t = x^k, \alpha_t) P(x_t = x^k|q_t, \alpha_t) \tag{1}$$

Since the exemplar, $x_t$, does not depend on the transformation, $\alpha_t$, we can drop the transformation parameter from the second term and therefore the observation at time $z_t$ is

$$P(z_t|q_t, \alpha_t) = \sum_{k=1}^{K} P(z_t|x_t = x^k, \alpha_t) P(x_t = x^k|q_t) \tag{2}$$

The term $P(z_t|x_t = x^k, \alpha_t)$ represents the observation model and will be discussed in the section 4. We call the term $P(x_t = x^k|q_t)$ the intermediate observation probability and this can be learned offline from the exemplar training set as part of the HMM learning procedure. The dynamics of the system, represented in the transitions $P(q_t|q_{t-1})$ and $P(\alpha_t|\alpha_{t-1})$, are also learned from the training data. For details about the learning procedure refer to [?].

## 4 Pose Classification

### 4.1 Pose Likelihood

We represent each gesture as a sequence of body poses. This section focuses on matching individual body poses to image data. The objective is to evaluate all different body pose models with respect to each new frame, $z_t$, in order to obtain estimate of observation probability given each pose, $P(z_t|X^k)$, where $S = \{X^k, k = 1 \cdots M\}$ is the set of

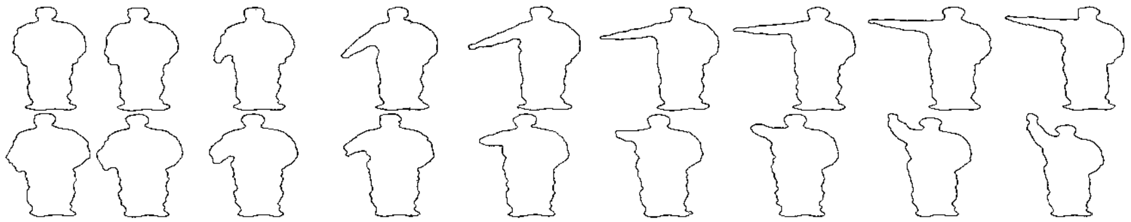

**Figure 3. Example body poses from two different Gesture**

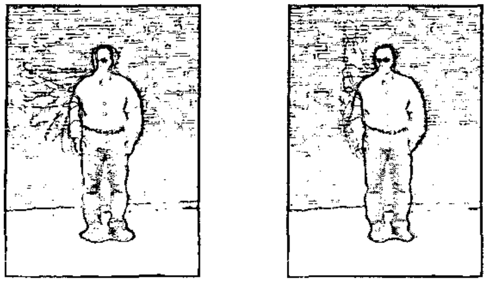

**Figure 4. Pose template registered to an image**

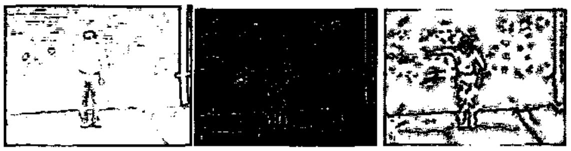

**Figure 5. Distance Transform**

all learned body poses for all the gestures to be recognized. Each pose $X^k$ is represented as an edge template, i.e., each pose is represented as a finite set of edge feature locations

$$X^k = \{x_1^k, x_2^k, \cdots, x_{m^k}^k\},$$

where $m^k$ is the number of edge features for pose exemplar $k$. Figure 3 shows example pose templates for two different gestures. All the poses are registered to each other during the learning so registering one pose to any new image will therefore register the rest of the poses. Registering these poses to the images is done while the person is not performing any gesture (idle). In this case, the matching is performed using an idle pose (shown in figure 3, first pose on top) through a coarse to fine search. Figure 4 shows the registered poses to a new frame.

The problem of matching a feature template corresponding to an object to an image is a classical problem in computer vision with many applications for object detection, recognition and tracking. The objective is to match a feature template, $X$, which is a finite set of feature points $x = \{x_1, x_2, \cdots x_n\}$ to an image.

The distance transform, DT, has been used in matching edge feature (and other feature) templates. Given a set of features, $F$, detected in an image $I$, the distance transform, $d_F(x)$, at pixel $x$, is defined to be the distance to the nearest feature point in the image, i.e.,

$$d_F(x) = \min_{f \in F} \rho(x, f)$$

where $\rho$ is a metric. Typically the Euclidean distance is used for the metric $\rho$ and in this case the function $d(x)$ defines the Voronoi surface of $F$ [7]. The distance transform is defined with respect to a set of binary feature of the same type, e.g., edges or corners.

Given a template $X = \{x_1, x_2, \cdots x_n\}$, where $x_i$'s are the locations of the template features transformed into the image space through translation, rotation, scaling or other geometric transformations, the matching can be achieved by averaging the distance transform values at each transformed template feature location $x_i$, i.e., the matching D(X,I) score is

$$D(X, F) = \frac{1}{n} \sum_i d_F(x_i) \qquad (3)$$

This form of matching is called Chamfer matching and the distance $D(X, F)$ is called the Chamfer distance. The smaller this matching score, the better the match and an ideal match will have the value 0 where the template exactly lie over its corresponding image location. Chamfer distance has been used extensively in object detection, for example in [5]. Note that Chamfer matching is asymmetric (model to image) so additional features in the image will not contribute to the matching. Figure 5 shows an example of an image and the detected edge features and the distance transformed image according to these features.

In [4] the matching was generalized to include multiple feature types, (for example, oriented edges) by matching each individual feature template with its corresponding distance transformed image and combining the results. Also the matching was generalized in [4, 5] to match multiple templates through a hierarchical template structure.

At each new image, $z_t$, it is desired to find a probabilistic matching score for each pose. Let $d_F(x)$ be the distance transformed image given the set of edge features,$F$, detected at this image. For each edge feature $x_i^k$ in pose model $X^k$, the measurement $D_i^k = d_F(x_i^k)$ is the distance to the nearest edge feature in the image. Consider the random variable associated with this distance measurement, and let the associated probability density function (PDF) be $p_i^k$. We assume that these random variable are independent. This assumption was used in [9, 11] based on the results obtained in [10]. Therefore, the likelihood function (the probability of the observation given the model $X^k$) can be defined as the product of these PDFs as

$$L(X^k) = Pr(z_t|X^k) = \prod_{i=1}^{m^k} p_i^k(D_i^k)$$

Since different templates have different numbers of features, this likelihood equation needs to be normalized using the number of features in each model,$m^k$. Taking the logarithm of this equation we obtain the log-likelihood function

$$\log L(X^k) = \frac{1}{m^k}\sum_{i=1}^{m^k} \log p_i^k(D_i^k) \qquad (4)$$

If all the poses are assumed to be equiprobable, then the model probability given the observation is proportion to the likelihood, i.e., $P(X^k|I) \propto P(I|X^k)$. Therefore we can use this likelihood function to evaluate different models.

The PDF $p_i^k$ for the distance between model features and nearest image feature location is defined for each feature $i$ in each pose model $k$. We use a PDF of the form

$$p_i^k(D) = c_1 + \frac{1}{\sigma_i^k\sqrt{2\pi}} e^{-D^2/2(\sigma_i^k)^2}$$

The scale parameter $\sigma_i^k$ is defined for each pose $k$ and each feature $i$. The motivation behind this is that different variations (or uncertainty) are expected at different model features; for example, the edges corresponding to the hand are expected to have more variations in location than the upper arm or the shoulder location. These variations are learned during the learning of the pose models. Since the distance $D$ can become arbitrary large, the probability can become very small and therefore the constant $c_1$ is used as a lower bound on the probability. This makes the likelihood function robust to outliers. A similar PDF was used in [9] but with the same scale variable $\sigma$ for all the features.

This probabilistic formulation was first introduced in [9] and was used in a Hausdorff matching context to find the best transformation of an edge template using maximum likelihood estimation. Equation 4 represents a probabilistic formulation of Chamfer matching. We use this probabilistic formulation to evaluate the observation likelihood given each gesture state as will be described in section 5

### 4.2 Correspondence-free Weighted Matching

Our objective is to match multiple pose templates to the same image location in order to evaluate the likelihood of the observation given each of these poses. Typically, the different pose templates are similar in some parts and different in another parts in the templates. For example, the head, torso and bottom parts of the body are likely to be similar in different pose templates, while articulated body parts that are involved in the gesture, such as the arm, will be at different positions at different pose templates. For example, see figure 4. Since the articulated part, such as the arm, is represented by a small number of features with respect to the whole pose templates, the matching is likely to be biased by the major body parts. Instead, it is desired to make the matching biased more by articulated parts involved in performing the gesture since these parts will be more discriminating between different poses templates.

To achieve this goal, different weights are assigned to different feature points in each pose template. Therefore each pose template, $X^k$, is represented as a set of feature locations as well as a set of weights, $\{w_1^k, w_2^k, \cdots, w_{m^k}^k\}$, corresponding to each feature where $\sum_{i=1}^{m^k} w_i^k = 1$. The likelihood equation 4 is then modified to be a weighted one

$$\log L(X^k) = \sum_{i=1}^{m^k} w_i^k \log p_i^k(D_i^k) \qquad (5)$$

In our case, the set of all recognized poses does not have a common correspondence frame. For example, some features in one pose might not have corresponding features in another poses. Also we do not restrict the pose templates to have the same number of features. Therefore we drive the weights with respect to the image locations.

Let $X$ be the set of all features in all registered poses in the training data, i.e.,

$$X = \bigcup_k X^k = \{x_1, x_2, \cdots x_m\}$$

where each $x_i$ is the image location of an edge feature. Given this sample of edge feature locations, the edge probability distribution $f(y)$ (the probability to see an edge at certain image location, $y$) can be estimated using kernel density estimation [14] as

$$\hat{f}(y) = \frac{1}{m}\sum_{i=1}^{m} K_h(y - x_i)$$

Where $K_h$ is a kernel function with a scale variable $h$. We used a Gaussian kernel $K_h(t) = \frac{1}{\sqrt{2\pi}h}e^{-1/2(\frac{t}{h})^2}$ for this probability estimation.

The weight assigned to each feature point is based on the information this feature provides. Given the estimated edge probability distribution, $\hat{f}(y)$, at any image pixel, $y$, the weight for a certain feature $i$ at a certain pose $k$ is the ratio of the information given by this feature to the total information by that pose, i.e.,

$$w_i^k = \frac{\log \hat{f}(x_i^k)}{\sum_{j=1}^{m^k} \log \hat{f}(x_j^k)}$$

## 5 Gesture classification

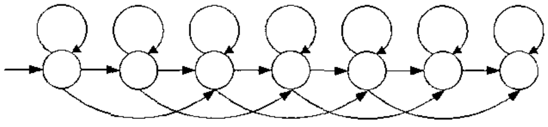

Figure 6. Left-Right HMM

The classification of gesture is achieved through evaluating the image sequences likelihood given a set of Hidden Markov Models (HMMs) each representing a gesture. HMM's have been used in gesture recognition context. They were used in [15] for American Sign Language recognition (ASL) by tracking the hands based on color. In [18, 17] HMM's were also used for ASL based on shape and motion parameters. In [8] HMM's were used to track head gestures. In [19] a parameterized HMM was introduced to model parametric gestures.

Generally, an HMM is defined as the states, $S$, the transition probabilities between the states, $A = \{a_{ij}\}$ where $a_{ij} = P[q_{t+1} = s_j | q_t = s_i]$ where $q_t$ is the state at time $t$, and the initial state distribution $\pi$ where $\pi_i = P[q_1 = s_i]$. Finally, the observation probability given the states $b_j(O) = P(O|s_j)$.

We represent each gesture $g$ by a set of poses $P_g = \{X^k, k = 1, \cdots, K\}$ and an HMM , $\lambda^g$, where the hidden states correspond to the progress of the gesture with time. The HMM elements are as follows:

1. A set of $N$ states $S = \{s_1, s_2, \cdots, s_N\}$. We use $q_t$ to denote the state at time $t$. Note that the number of states is not necessarily the same as the number of pose models, $M$, i.e., each state does not necessarily represent one pose. Instead, one state can represent a mixture of pose models.

2. The state transition probabilities $A = \{a_{ij}\}$ where $a_{ij} = P[q_{t+1} = s_j | q_t = s_i] \quad \forall i, j = 1 \cdots N$. We use a left-right model or a Bakis model [] as in figure 6 since the progress of the gesture is always forward in time. This imposes a constraint on the dynamics which leads to better generalization since there are less transitions to adjust.

3. The initial state distribution $\pi$ where $\pi_j = P[q_1 = s_j] \quad \forall j = 1 \cdots N$.

4. The probability of each pose $X^k$ given the states, $C = \{c_{jk} = P(X^k|s_j) \quad \forall j = 1 \cdots N, \forall k = 1 \cdots M\}$

Training sequences of poses are used to learn the model parameters

The actual observation $O_t$ is the detected edge features at each new frame, which is a probabilistic function of the current state of the gesture. This probabilistic function is defined using the set of recognized poses $P_g$. That is, the observation probability given the state can be written as

$$b_j(O_t) = P(O_t|s_j) = \sum_{k=1}^{M} P(O_t|X^k) P(X^k|s_j)$$

Given the definition of the variables $C$ above, this can be rewritten as

$$b_j(O_t) = P(O_t|s_j) = \sum_{k=1}^{M} c_{jk} P(O_t|X^k)$$

We can think of the set of poses $P_g$ as a set of discrete symbols or alphabet that is being emitted by the different states, but the actual observation is a probabilistic function of these symbols based on the mixture defined by the variables $C$. The observation probabilities given the poses, $P(O_t|X^k)$, are obtained using the likelihood equations 4 and 5 as was described in section 4

Given a set of observations $O = O_1 O_2, \cdots, O_T$ and given a set of HMM models $\lambda^g$ corresponding to different gesture, the objective is to determine the probability of that observation sequence given each of the models, i.e., $P(O|\lambda^g) \forall g$. This is a traditional problem for HMM and can be solved efficiently through a procedure called the Forward-Backward procedure [12]. This procedure defines a set of forward variables $\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i | \lambda)$ which can be updated recursively at each time step by:

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), 1 \le t \le T - 1$$

where $\alpha_1(i) = \pi_i b_i(O1)$. Given these forward variables, the observation likelihood can be calculated as

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

## 6 Experimental Results

The proposed approach was used to classify eight arm gestures. Basically, the eight recognized gestures are similar to the ones shown in figure 4, performed with both arms, in upward motion and downward motion. Figure 8 shows the segmentation results obtained from the range data. The image is color coded so that each fitted plane has a different graylevel and the segmented person is labeled white. Figure 9 shows some pose classification results for different people. The figures shows the pose with the highest likelihood score overlaid over the original image.

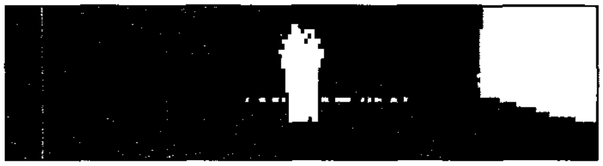

**Figure 8. Segmentation result**

Figure 7 shows the gesture likelihood probabilities for the eight gesture classes. As can be noticed from the graphs, All the gestures were close in likelihood at the beginning of the action but as the gesture progresses with time, the likelihood of the right gesture increases, and the the likelihood of the other gesture decreases as a result of the temporal constrains imposed by the HMM for each gesture.

## References

[1] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE*

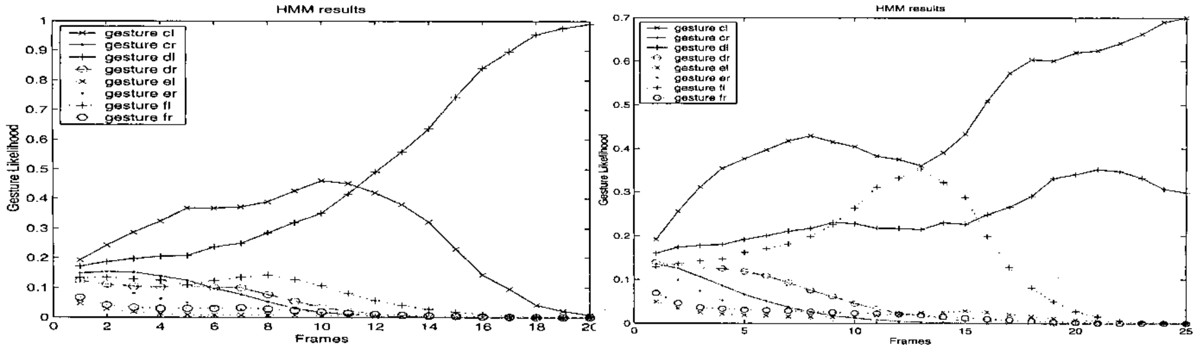

**Figure 7. Gesture classification results**

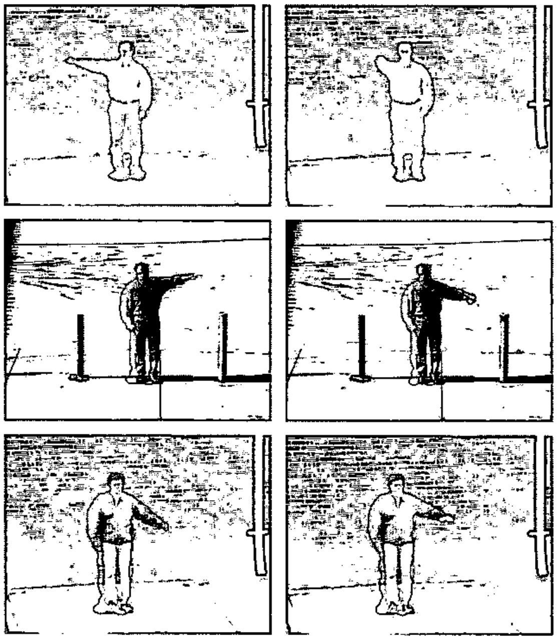

**Figure 9. Pose matching results**

*Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[2] Brendan J. Frey and Nebojsa Jojic. Learning graphical models of images, videos and their spatial transformation. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence - San Francisco, CA*, 2000.

[3] Brendan J. Frey and Nebojsa Jojic. Flexible models: A powerful alternative to exemplars and explicit models. In *IEEE Computer Society Workshop on Models vs. Exemplars in Computer Vision*, pages 34–41, 2001.

[4] D. Gavrila. Multi-feature hierarchical template matching using distance transforms. In *In Proc. of the International Conference on Pattern Recognition, Brisbane, Australia,1998.*, pages 439–444.

[5] D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *ICCV99*, pages 87–93.

[6] D. Hogg. Model based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[7] D. Huttenlocher, D. Klanderman, and A. Rucklige. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.

[8] C. Morimoto, Y. Yacoob, and L. Davis. Recognition of head gestures using hidden markov models international conference on pattern recognition. In *International Conference on Pattern Recognition, Vienna, Austria, August 1996*, pages 461–465, 1996.

[9] C. Olson. A probabilistic formulation for hausdorff matching. In *CVPR98*, pages 150–156.

[10] C. Olson and D. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, (1):103–113, 1997.

[11] Clark F. Olson. Maximum-likelihood template matching. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 52–57, 2000.

[12] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, February 1989.

[13] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *ICCV95*, pages 612–617, 1995.

[14] David W. Scott. *Mulivariate Density Estimation*. Wiley-Interscience, 1992.

[15] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. In *SCV95*, page 5B Systems and Applications, 1995.

[16] Kentaro Toyama and Andrew Blake. Probabilistic tracking in a metric space. In *ICCV*, pages 50–59, 2001.

[17] C. Vogler and D. Metaxas. Adapting hidden markov models for asl recognition by using three-dimensional computer vision methods. In *SMC'97.*, pages 156–161.

[18] Christian Vogler and Dimitris N. Metaxas. Parallel hidden markov models for american sign language recognition. In *ICCV (1)*, pages 116–122, 1999.

[19] Andrew D. Wilson and Aaron F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):884–900, 1999.