M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 485
Appears in Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000

Understanding Purposeful Human Motion

Christopher R. Wren, Brian P. Clarkson, Alex P. Pentland

MIT Media Laboratory; 20 Ames Street; Cambridge MA 02139 USA
{cwren, clarkson, sandy}@media.mit.edu http://www.media.mit.edu/vismod/

Abstract

Human motion can be understood on many levels. The most basic level is the notion that humans are collections of things that have predictable visual appearance. Next is the notion that humans exist in a physical universe, as a consequence of this, a large part of human motion can be modeled and predicted with the laws of physics. Finally there is the notion that humans utilize muscles to actively shape purposeful motion. We employ a recursive framework for real-time, 3-D tracking of human motion that enables pixel-level, probabilistic processes to take advantage of the contextual knowledge encoded in the higher-level models, including models of dynamic constraints on human motion. We will show that models of purposeful action arise naturally from this framework, and further, that those models can be used to improve the perception of human motion. Results are shown that demonstrate automatic discovery of features in this new feature space.

1 Introduction

This paper describes a motion understanding framework that relies on our real-time, fully-dynamic, 3-D person tracking system. The dynamic tracking is driven by 2-D *blob features* observed in two or more cameras [9], and is informed by behavior models that estimate control signals. These features and controls are then probabilistically integrated by a fully-dynamic 3-D skeletal model, which in turn drives the 2-D feature tracking process by setting appropriate prior probabilities. The intrinsic state of the skeletal model is also used by the behavior module to choose the appropriate control strategy.

The feedback between 3-D model and 2-D image features is a recursive filter, similar to an extended Kalman. One unusual aspect of our approach is that the filter directly couples raw pixel measurements with an articulated dynamic model of the human skeleton. Previous attempts at person tracking have utilized a generic set of image features (e.g., edges, optical flow) that were computed as a preprocessing step, without consideration of the task to be accomplished. In this aspect our system is similar to that of Dickmanns in automobile control[3], and our previous research shows that we obtain similar advantages in efficiency and stability though this direct coupling.

We will show how this framework can go beyond passive physics of the body by incorporating various patterns of control (which we call 'behaviors') that are *learned* from observing humans while they perform various tasks. Behaviors are defined as those aspects of the motion that cannot be explained by passive physics alone. In the untrained tracker these manifest as significant structure in the innovations process (the sequence of prediction errors). Sets of behaviors that explain some collection of motion will be called a *behavior alphabet*. These alphabets can be automatically discovered and can then be used to recognize and predict this purposeful aspect of human motion.

This paper will briefly discuss the formulation of our 3-D skeletal model in Section 2.1, followed by an explanation of how to drive that model from 2-D probabilistic measurements, and how 2-D observations and feedback relate to that model in Section 2.2. Section 3 explains the behavior system and its intimate relationship with the physical model. Finally, we will report on experiments showing the results of automatic alphabet discovery in different contexts and the affect of behavior models on tracking performance in Section 4.

1.1 Related Work

The work described in this paper attempts to combine the the dynamic modeling work with the advantages of a recursive approach, by use of a formulation related to the extended Kalman filter that couples a fully dynamic skeletal model with observations of raw pixel values, as modeled by probabilistic 'blob' models[5, 9].

This system also attempts to incorporate learned patterns of control into the body model. The approach we take is based on the behavior modeling framework introduced in Pentland and Liu 1995[6]; it is also related to the behavior modeling work of Blake[4] and Bregler[1]. However, the controller described here operates on a 3-D non-linear model of human motion that is closer to true body dynamics than 2-D linear models.

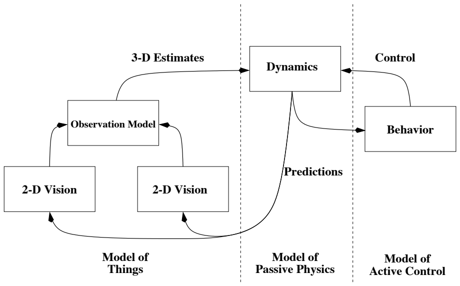

Figure 1: The flow of information though the system.

2 Recursive Tracking Framework

The human body is a complex dynamic system, whose visual features are time-varying, noisy signals. Accurately tracking the state of such a system requires use of a recursive estimation framework, as illustrated in figure 1. The tightly coupled elements of the framework are the observation model relating noisy pixel-level features to the higher-level skeletal model and *vice versa*, the dynamic skeletal model itself, and a collection of typical behaviors.

2.1 Dynamics

There are a wide variety of ways to model the human body. A *kinematic* model can only describe the static state of a system. A system in motion is better described when the *dynamics* of the system are modeled as well. In a dynamic model the state vector includes velocity as well as position: $(\mathbf{q}, \dot{\mathbf{q}})$. In real systems, such as the human body, the state evolves according to Newton's First Law:

$$\ddot{\mathbf{q}} = \mathbf{W} \cdot \mathbf{Q} \tag{1}$$

Where $\mathbf{Q}$ is the vector of external forces applied to the system, and $\mathbf{W}$ is the inverse system mass matrix.

2.1.1 Hard Constraints

Hard constraints represent additional absolute limitations imposed on a system. One example is the kinematic constraint of a skeletal joint. Our model follows the *virtual work* formulation[8]. In a virtual work formulation, all the links in a model have full range of unconstrained motion. Hard kinematic constraints on the system are enforced by a special set of forces $\mathbf{c}$:

$$\ddot{\mathbf{q}} = \mathbf{W} \cdot (\mathbf{Q} + \mathbf{c}(\mathbf{q}, t)) \tag{2}$$

The constraint forces do not add energy if they lie in the null space complement of the constraint Jacobian. Combining that requirement with the constraint definitions results in a linear system of equations with only the one unknown, $\boldsymbol{\lambda}$:

$$-\mathbf{J}^T \mathbf{W} \mathbf{J} \boldsymbol{\lambda} = \boldsymbol{\rho} \tag{3}$$

Where $\mathbf{J}$ is the constraint Jacobian, $\boldsymbol{\rho}$ is a known vector, and $\boldsymbol{\lambda}$ is the vector of unknown Lagrange multipliers. See [9] for more information this constraint system.

2.1.2 Soft Constraints

Some constraints are probabilistic in nature. Noisy image measurements are a constraint of this sort, they influence the dynamic model but do not impose hard constraints on its behavior. Soft constraints such as these can be expressed as a potential field acting on the dynamic system. The incorporation of a potential field function that models a probability density pushes the dynamic evolution of the model toward the most likely value, starting from the current model state.

The controllers in Section 3, will be represented as time-varying potential field and may depend on the system state itself:

$$\mathbf{Q}_f = f(\mathbf{X}, \mathbf{q}, \dot{\mathbf{q}}) \tag{4}$$

2.2 The Observation Model

Our vision system tracks regions that are visually similar, and spatially coherent: blobs. We can represent these 2-D regions by their low-order statistics. Clusters of 2-D points have 2-D spatial means and covariances, which we shall denote $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. For computational convenience we will interpret these blobs with a Gaussian model:

$$\Pr(\mathbf{O}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{\exp(-\frac{1}{2}(\mathbf{O} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{O} - \boldsymbol{\mu}_k))}{(2\pi)^{\frac{m}{2}} |\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \tag{5}$$

The Gaussian interpretation is not terribly significant, because we also keep a pixel-by-pixel *support map* showing the actual occupancy [9].

These observations supply constraints on the underlying 3-D human model. Due to their statistical nature, observations are easily modeled as soft constraints. Observations are integrated into the dynamic evolution of the system by modeling them as descriptions of potential fields, as discussed in Section 2.1.2.

2.2.1 The Inverse Observation Model

In the open-loop case, the vision system uses a Maximum Likelihood (ML) framework to label individual pixels in the scene:

$$\Gamma_{ij} = \arg\max_k [\Pr(\mathbf{O}_{ij}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] \tag{6}$$

where Γ_{ij} is the labeling of pixel (i, j), and $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ are the second-order statistics of model k.

The dynamic constraints expressed in the body model allow predictions to be made about likely future configurations. The predictions are projected from 3-D model space into 2-D feature space. These predicted observations become prior information, $\mathbf{v}$, for the vision system. This prior information directly affect the interpretation of individual pixels. Integrating this information into the 2-D statistical decision framework results in a Maximum *A Posteriori* decision rule:

$$\Gamma_{ij} = \arg\max_k \left[\Pr(\mathbf{O}_{ij}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \cdot \Pr(\mathbf{O}_{ij}|\mathbf{v}_k)\right] \quad (7)$$

3 Models of Purposeful Motion

Observations of the human body reveal an interplay between the passive evolution of a physical system (the human body) and the influences of a an active, complex controller (the nervous system). Section 2.1 explains how, with a bit of work, it is possible to model the physical aspects of the system. However, it is *very* difficult to explicitly model the human nervous system, so the approach of using observed data to estimate probability distributions over control space is very appealing.

3.1 A Model for Control

Kalman filtering includes the concept of an *innovations process*. This is the difference between the actual observation and the predicted observation transformed by the Kalman gain:

$$\boldsymbol{\nu}_t = \mathbf{K}_t(\mathbf{y}_t - \mathbf{H}_t\boldsymbol{\Phi}_t\hat{\mathbf{x}}_{t-1}) \quad (8)$$

The innovations process, $\boldsymbol{\nu}$, is the sequence of information in the observations that was not adequately predicted by the model. According to control theory, if we have a sufficient model of the dynamic process and the observation model, and white, zero-mean Gaussian noise is added to the system, both in the observation stream and into the real dynamic system itself, then the innovations process will be zero-mean and white. Inadequate models will manifest as correlations in the innovations process.

As described above, we have significant models of the observed human in terms of appearance, perspective, and passive physics. The most significant unmodeled aspect of human motion is active control. Purposeful human motion includes significant structure due to the active control of nerves and muscles that is not currently well modeled.

A simple example is helpful for illustrating this idea. Assume that we model hand motion with a linear, constant velocity Kalman filter. If we track the hand moving in a circular motion, and the model is sufficient, then the errors in the predictions should be solely due to the noise in the system. Figure 2 show that model is not sufficient. The innovations,$\boldsymbol{\nu}$, contain significant structure. Plotting the innovations along the path of observations make the relationship between the observations and the innovations clear: there is some unmodeled process acting to keep the hand moving in a circular motion (left of Figure 2). The most significant unmodeled process is the purposeful control signal that being expressed by the active muscles.

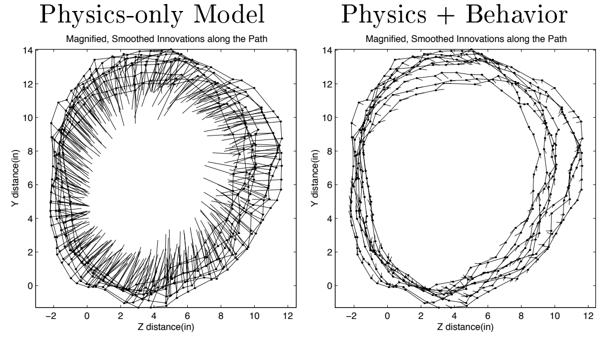

Figure 2: Modeling tracking data of circular hand motion. Errors from passive physics alone are reduced by a learned model of control.

In this example, there is one active, cyclo-stationary control behavior, and it's relationship to the state of the physical system is straightforward. If we use the smoothed innovations as our model and assume a linear control model of identity, then the linear prediction becomes:

$$\hat{\mathbf{x}}_t = \boldsymbol{\Phi}_t\hat{\mathbf{x}}_{t-1} + \mathbf{I}\mathbf{u}_{t-1} \quad (9)$$

where $\mathbf{u}_{t-1}$ is the control signal applied to the system. The right plot in Figure 2 shows the result. The innovations are now near zero and we have explained a significant source of unmodeled structure.

This same idea can be applied to the non-linear models described in Section 2.1. The next section examines a more powerful form of model for behavior.

3.2 Hidden Markov Models of Behavior

Since human motion evolves over time, in a complex way, it is advantageous to explicitly model temporal dependence and internal states in the control process. A Hidden Markov Model (HMM) is one way to do this, and has been shown to perform quite well recognizing human motion[7].

The probability that the model is in a certain state, S_j given a sequence of observations, $\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_N$, is defined recursively. For two observations, the density is:

$$\Pr(\mathbf{O}_1, \mathbf{O}_2, s_2 = S_j) = \left[\sum_{i=1}^{N} \pi_i b_i(\mathbf{O}_1)\mathbf{a}_{ij}\right] b_j(\mathbf{O}_2) \quad (10)$$

Where π_i is the prior probability of being in a state i, and $b_i(\mathbf{O})$ is the probability of making the observation $\mathbf{O}$ while in state i. This is the Forward algorithm for an HMM.

Estimation of the control signal proceeds by identifying the most likely state given the current observation and the last state, and then using the observation density of that state as described above. We restrict the observation densities to be either a Gaussian or a mixture of Gaussians.

3.3 Behavior Alphabet Auto-Selection

One of our goals is to observe a user who is interacting with a system and be able to automatically find patterns in their behavior. Interesting questions include:

- Is this (a)typical behavior for the user?
- Is this (a)typical behavior for anyone?
- When is the user transitioning from one behavior/strategy to another behavior/strategy?
- Can we do filtering or prediction using models of the user's behavior?

However, before we can build a model of behavior we must find the behavior alphabets that pick out the salient movements relevant to the questions above. There probably will not be one canonic alphabet for all tasks but rather many alphabets each suited to a group of tasks. Therefore we need an algorithm for automatically generating and selecting effective behavior alphabets. The goal of finding an alphabet that is suitable for a machine learning task can be mapped to the concept of feature selection.

In rough terms, our alphabet selection algorithm is a clustering algorithm that uses a task-related criterion. We chose to use HMMs to model each behavior in an alphabet. Candidate alphabets were generated by clustering the raw features with HMMs. Free parameters of the clustering were:

1. N, Number of HMMs (number of behaviors)
2. S, Number of States per HMM (complexity)
3. T, Time Scale (typical behavior length)

For each set of parameters, we clustered the raw features using an algorithm that can be interpreted as K-Means where the Gaussians are replaced with HMMs. A more complete description of the algorithm can be obtained in [2]. The centroids that result are HMMs that each encode a time sequence in raw feature space (i.e. a behavior). Each HMM is a behavior or symbol in an alphabet that was used to convert the raw features to soft symbols or likelihoods by means of the Forward Algorithm. So if the number of HMMs used is N, then the alphabet size is N and the raw features were mapped to a likelihood space of N dimensions. The next step was to use the likelihoods to build a classifier for a given task and evaluate the classifier's performance. The classifier's performance was then fed back to the cluster parameter search for model selection. This process is illustrated in Figure 3 and outlined below:

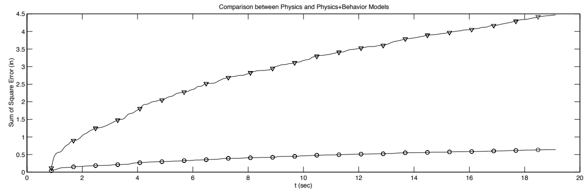

Figure 4: Sum Square Error of a Physics-only tracker (triangles) vs. error from a Physics+Behavior Tracker

1. Input Raw Features (Innovations), ν
2. For each (N, S, τ)
3. Cluster ν with N S-state HMMs at Time Scale τ HMMs H
4. Use HMMs obtained to generate likelihood traces $L(t) = P(\nu(t)|H)$
5. Use L to train and test a classifier for a given task,
6. Select H that maximizes step 5's performance.

We chose 2 tasks to explore this method of alphabet selection. For context, we employed a simple virtual reality game that requires a player to pop bubbles that fall from above and whack wuggles (small creatures) that sit on a table. The player's motions were tracked and labeled for three types of behavior:

1. Whacking a wuggle
2. Popping a bubble
3. Experiencing tracker failure

Task 1 was to recognize these 3 classes of behavior. Task 2 was to be able to distinguish the playing styles of different people.

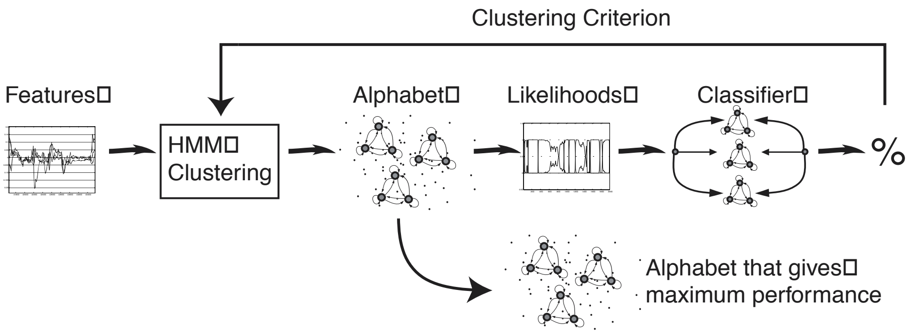

Figure 3: Auto-selection pipeline.

4 Results

The circular motion example from Section 3.1 was used for tracking performance evaluation. The dynamic model allows the system to recover from occlusions and reject inconsistent noise, but it cannot predict the purposeful aspect of the motion. The inclusion of a behavior model fixes this deficiency. Figure 4 compares noise in the two cases. It can be seen that there is a significant increase in performance.

The game described in Section 3.3 was used to evaluate automatic alphabet discovery. Volunteers played for ten minutes. The data was labelled for intentional state during game play and for player identity. This data was split into training and test sets.

Figure 5 shows the results of searching the model parameter space for the best alphabet in the player identification task. The plot show the performance surface parameterized by number of HMMs (N), and the fundamental timescale of the HMMs (T). The best alphabet parameters for distinguishing the 3 players was 3 elements, with 10 states each and a base time scale of 32 frames (1s). Figure 6 shows alphabet traces for the three players over approximately one minute of play. These traces are the features used to do player identification. Player identification performance was 75% for three players.

Figure 7 illustrates the actions of the best HMMs in the intention identification task for a specific player. For this player the best intention alphabet parameters were 3 elements, with 9 states each and a base time scale of 8 frames (250ms). The plots show the data in grey, and the mean position and iso-probability contours in black. The left and right HMMs seem to be explaining salient motions for recognizing player's intention, while the middle HMM is modeling the tracker failures

5 Conclusion

We have presented a framework for human motion understanding, defined as estimation of the physical state of the body combined with interpretation of that part of the motion that cannot be predicted by passive physics alone. The behavior system is capable of automatically discovering alphabets to describe salient motion paradigms that are tuned to specific interpretation tasks. The behavior system operates in conjunction with a real-time, fully-dynamic, 3-D person tracking system that provides a mathematically concise formulation for incorporating a wide variety of physical constraints and probabilistic influences. The framework takes the form of a non-linear recursive filter that enables even pixel-level processes to take advantage of the contextual knowledge encoded in the higher-level models. Some of the demonstrated benefits of this approach include: increase in 3-D tracking accuracy, and insensitivity to temporary occlusion.

References

[1] Christoph Bregler. Learning and recognizing human dynamics in video sequences. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 1997.

[2] Brian P. Clarkson and Alex Pentland. Unsupervised clustering of ambulatory audio and video. In *Proceedings of the International Conference of Acoustics Speech and Signal Processing*, Phoenix, Arizona, 1999.

[3] Ernst D. Dickmanns and Birger D. Mysliwetz. Recursive 3-d road and relative ego-state recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):199–213, February 1992.

[4] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density.

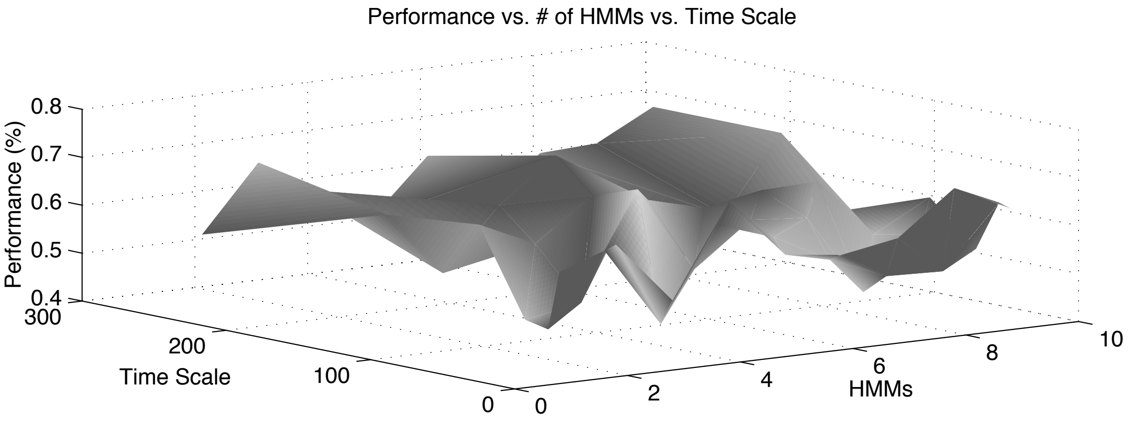

Figure 5: Performance for an identification task as a function of model parameters.

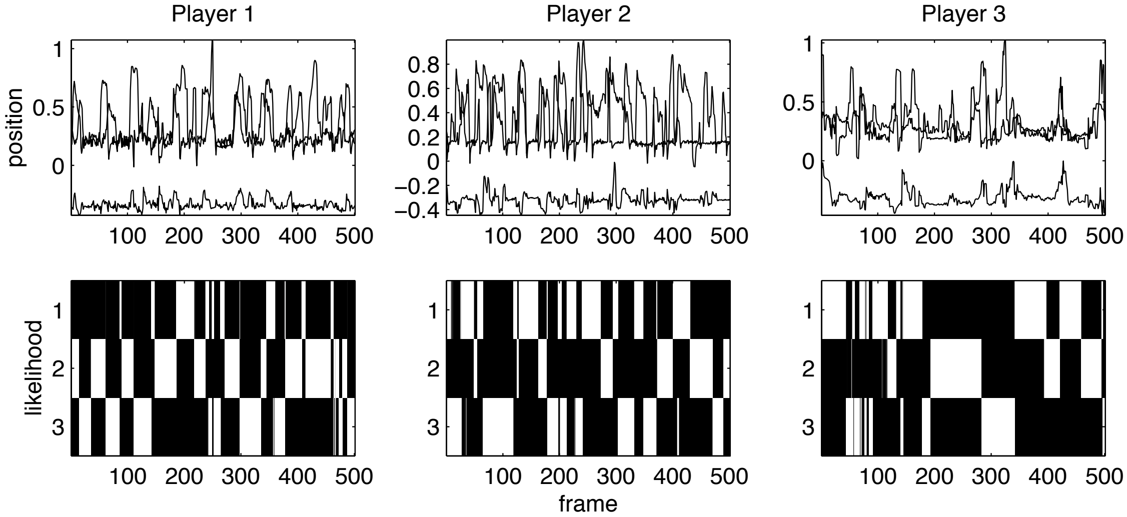

Figure 6: **Top:** Tracking data. **Bottom:** Corresponding likelihood trace of the identification alphabet.

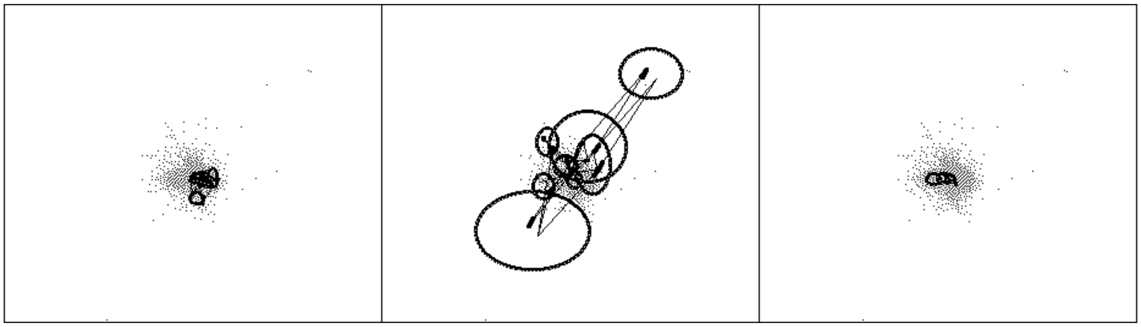

Figure 7: HMMs for the intentionality alphabet.

In *Proc. European Conference on Computer Vision*, pages 343–356, Cambridge, UK, 1996.

[5] I. Kakadiaris, D. Metaxas, and R. Bajcsy. Active part-decomposition, shape and motion estimation of articulated objects: A physics-based approach. In *CVPR94*, pages 980–984, 1994.

[6] Alex Pentland and Andrew Liu. Modeling and predicition of human behavior. In *IEEE Intelligent Vehicles 95*, September 1995.

[7] Thad Starner and Alex Pentland. Real-time american sign language recognition from video using hidden markov models. In *Proceedings of International Symposium on Computer Vision*, Coral Gables, FL, USA, 1995. IEEE Computer Society Press.

[8] Andrew Witkin, Michael Gleicher, and William Welch. Interactive dynamics. In *ACM SIGGraph, Computer Graphics*, volume 24:2, pages 11–21. ACM SIGgraph, March 1990.

[9] Christopher R. Wren and Alex P. Pentland. Dynamic models of human motion. In *Proceedings of FG'98*, Nara, Japan, April 1998. IEEE.