Proc. IEEE Conf. Comp. Vision and Pattern Recognition, San Juan, Puerto Rico, June 1997

Learning and Recognizing Human Dynamics in Video Sequences

Christoph Bregler
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
bregler@cs.berkeley.edu

Abstract

This paper describes a probabilistic decomposition of human dynamics at multiple abstractions, and shows how to propagate hypotheses across space, time, and abstraction levels. Recognition in this framework is the succession of very general low level grouping mechanisms to increased specific and learned model based grouping techniques at higher levels. Hard decision thresholds are delayed and resolved by higher level statistical models and temporal context. Low-level primitives are areas of coherent motion found by EM clustering, mid-level categories are simple movements represented by dynamical systems, and high-level complex gestures are represented by Hidden Markov Models as successive phases of simple movements. We show how such a representation can be learned from training data, and apply it to the example of human gait recognition.

1 Introduction

This paper addresses the problem of learning and recognizing human and other biological movements in video sequences of an unconstrained environment. We attack this problem with a compositional framework consisting of statistical models at various levels. Starting at raw pixel values of an input video sequence, we show how hypotheses at various abstraction levels can be propagated probabilistically through space and time, and can be used to recognize complex movements. We demonstrate how to learn such multi-level decompositions from training data and use it for recognition of human gait categories in unconstrained cluttered environments.

Segmentation and recognition is treated as the same problem: Recognition is a succession of very general low level grouping mechanisms to increased specific and learned model based grouping techniques at higher levels. Hard decision thresholds are delayed and resolved by higher level statistical models and temporal context. Speech recognition is a prime example where multiple levels of abstraction, like speech features, phoneme categories, word models, and language models are integrated in a probabilistic way and estimated from large training corpuses. Although the domain of speech recognition is much more structured and simplified, recent trends in the field of statistical learning, and its application to low and mid-level computer vision provide a basic substrate for our ambitious goal to treat the visual domain with similar principles.

Section 2 describes the framework, which includes a low-level layered representation, mid-level temporal grouping using simple dynamical categories, and high-level recognition of complex movements. Subsection 2.3 describes how such an architecture can be estimated from training data. In Section 3 we describe learning, segmentation, and recognition experiments on human gait data, and in Section 4 we relate our approach to previous work.

2 Probabilistic Compositional Framework

While performing an action or gesture most of the human body segments are in motion most of the time. This is a very strong cue that we wish to exploit. The image region that belongs to a rigid body segment contains one coherent motion field. Two body segments can be disambiguated by detecting two different coherent motion areas in the image.

Over multiple frames, characteristic motion sequences can be detected. Simple movements (for example arm or leg swings) can be modeled with linear dynamical systems, whereas more complex movements like a walk cycle can be represented as a sequence of simple movements. While a leg has ground support, a certain linear dynamical system has validity, and while the leg is swinging above the ground, another linear dynamical system can describe the motion history.

Noisy input images, spatial and temporal ambiguities, occlusion, cluttered environments, and large variability call for a probabilistic framework. Guiding principles are, (a) no early commitment to specific hypotheses, (b) besides bottom up flow, higher level hypothesis should be able to disambiguate lower level estimates, (c) low computation and representation costs, (d) mid and higher level models should be learnable.

Figure 1 shows the 4 level decomposition. Each level represents a set of random variables and probability distributions over hypotheses. The lowest level is a sequence of input images. For each pixel we represent the spatio-temporal image gradient and optionally the color value as a random variable. At the next level are blob hypotheses. Each blob is represented with a probability distribution over coherent motion (rotation and translation or full affine motion), color (HSV values), and spatial "support-regions". In the third level, temporal sequences of blob tracks are grouped to linear stochastic dynamical models. At the highest level, each dynamical model corresponds to the emission probability of the state of a Hidden Markov Model (HMM).

For example, the movement of one leg during a walk cycle can be decomposed into one coherent motion blob for the upper leg, and one coherent motion blob for the lower leg, one dynamical system for all the time frames while the leg has ground support, and one dynamical system for the

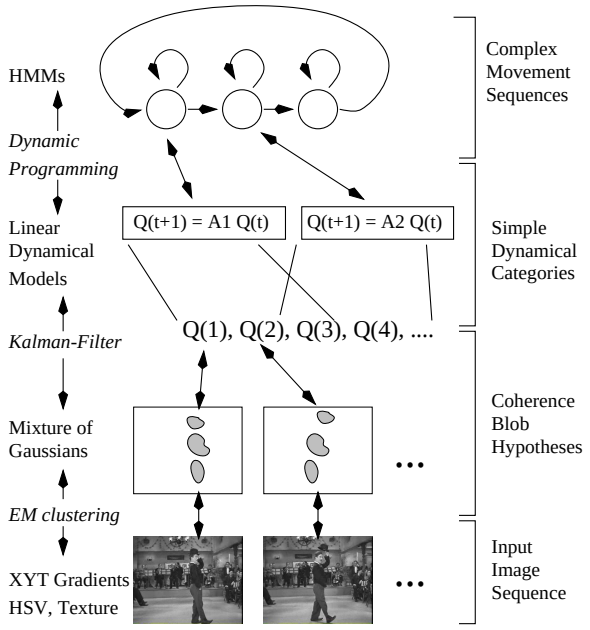

Figure 1: 4 level decomposition of human dynamics.

case the leg is swinging above ground, and a "cyclic" HMM with 2 states. The state space of the dynamical systems are the translation and angular velocities of the blob hypothesis. The HMM stays in the first state for as many frames as the first dynamical system is valid, and transits to the second state once the second dynamical system is valid, and then cycles back to the first state for the next walk cycle.

Given a sequence of images $I_1, I_2, ...I_t$, corresponding blob estimates, linear dynamical systems, and HMMs for a set of different gaits or gestures, we can perform a high-level classification in computing the following posterior:

$$P(\text{HMM}_i|I_1, I_2, ...I_t) \tag{1}$$

The Hidden Markov Model (HMM_i) with the highest score is the most likely complex gesture performed in the image sequence. If more than one gesture is performed, several HMMs corresponding to several blob tracks should have high score.

In the following subsections we describe the various levels and the multilevel estimation process.

2.1 Mixtures of Coherence Blobs

The likelihood that a group of pixel belong to the same blob is based on motion (and color) similarity, spatial proximity, and groupings in earlier time frames. For each pixel location (x, y) we need to estimate a "hidden variable" $S(x, y)$ that tells us to which blob it belongs, and we need to estimate for each blob the motion, color, and spatial distribution. If the number of blobs is K, the domain of $S(x, y) \in \{1, 2, ..., K\}$. This leads to a representation that is already used in so called "layered motion" approaches [29, 17, 1]. Simultaneously the labels $S(x, y)$ and the motion, color, and spatial parameters can be estimated using the *Expectation Maximization* (EM) maximum likelihood [9]. There are various ways to determine the number of layers K as well [1, 29]. Our approach differs in the way it also incorporates past histories of groupings in earlier frames, and how it encodes spatial proximity.

The set of blob hypotheses for a given image frame $I(t)$ are represented as a mixture of multivariate Gaussians $\theta(t)$. Each single Gaussian $\theta_k(t)$ encodes the coherent motion of that blob (either 2D translation and rotation, or affine motion), optionally the coherent HSV color values, and the center of mass and second moments of the (x, y) pixel coordinates in each blob. An additional outlier or background layer $\theta_0(t)$ is defined, that has uniform distribution.

The likelihood of an image frame $I(t)$ given a "mixture of blobs" hypothesis $\theta(t)$ is defined as:

$$P(I(t)|\theta) = \prod_{x,y} P(I(t, x, y), x, y|\theta(t)) \tag{2}$$

$$\begin{aligned} P(I(t, x, y), x, y|\theta(t)) &= \sum_{k=0}^{K} (\omega_k(t) \cdot P(x, y|\theta_k(t)) \\ &\cdot P(I(t, x, y)|x, y, \theta_k(t)) \end{aligned} \tag{3}$$

$\omega_k(t)$ are the mixing coefficients of the mixture model which should add up to 1. As we will see later, they are useful for "track-elimination" as well.

$P(x, y|\theta_k(t))$ is the spatial proximity prior for blob k (Gaussian distribution using the mean and second moments, or uniform for background).

2.1.1 Motion Model

$P(I(t, x, y)|x, y, \theta_k(t))$ is defined using the spatio-temporal image gradient and optionally the color values $\text{hsv}(t, x, y)$. The standard gradient formulation for optical flow is:

$$\nabla I(t, x, y) \cdot v(x, y) + I_t(t, x, y) = 0 \tag{4}$$

where ∇ is the spatial gradient operator in x and y direction, and I_t is the temporal derivative. $v(x, y)$ is the motion at point (x, y).

In case of an affine motion model:

$$v(x, y) = \begin{pmatrix} x \cdot s_{1,1} & +y \cdot s_{1,2} & +d_x \\ x \cdot s_{2,1} & +y \cdot s_{2,2} & +d_y \end{pmatrix} \tag{5}$$

where $S = \begin{bmatrix} 1 + s_{1,1} & s_{1,2} \\ s_{2,1} & 1 + s_{2,2} \end{bmatrix}$, $\begin{bmatrix} d_x \\ d_y \end{bmatrix}$ is the affine warp. In case of rotation and translation only, S is constrained to be orthonormal.

The term in (4) can be modeled with a zero-mean Gaussian distribution ([27]). This defines $P(\nabla I(t, x, y)|s_{1,1}, s_{1,2}, s_{2,1}, s_{2,2}, d_x, d_y)$ which we use for $P(I(t, x, y)|x, y, \theta_k(t))$.

2.1.2 EM algorithm

Maximizing (2) is done using the EM algorithm. The E-step is the estimation of the support layers for each blob.

The support layer for blob k at pixel (x, y) is the posteriori probability:

$$S_k(t,x,y) := P(S(t,x,y) = k | I(t,x,y,), x, y, \theta(t)) \quad (6)$$

$$\propto \omega_k P(x,y|\theta_k(t)) P(I(t,x,y)|x,y,\theta_k(t)) \quad (7)$$

The Gaussian distribution $P(x, y|\theta_k(t))$ dies off quickly beyond a certain region-of-interest, which allows pruning. We only maintain the support map of this region-of-interest.

M-Step is maximizing the expected log-likelihood which can be decomposed into minimizing the following three independent terms:

$$-\sum_k (\sum_{x,y} S_k(t,x,y) \log \omega_k \quad (8)$$

$$\sum_k \sum_{x,y} S_k(t,x,y) \log P(x,y|\theta_k(t)) \quad (9)$$

$$\sum_k \sum_{x,y} S_{k,x,y} \log P(I(t,x,y)|x,y,\theta_k(t) \quad (10)$$

Minimizing (8) with to the constraint $\sum_k \omega_k = 1$ is equivalent to assigning $\omega_k := \frac{\sum_{x,y} S_k(t,x,y)}{\sum_k \sum_{x,y} S_k(t,x,y)}$. Minimizing (9) is equivalent to computing the weighted means and covariances for support layer. (10) can be minimized using an extension of the Lucas-Kanade motion estimation described by Shi and Tomasi [26]. Our experiments have shown that with just a few E and M steps, we already converge to a stable estimate.

The initialization of the EM algorithm is done by splitting up the image into equal tiles. These are the initial support maps. The M-step then computes so-to-speak optical flow on these tile "super-pixels". The next E-step refines the support maps, and the next M-step refines the motion parameters for the new support maps. It has been proved that the likelihood (2) is non-decreasing at each iteration [9]. If it does not increase, it is at a convergence point.

Figure 2 shows two example support maps of two blob models covering the lower and upper leg of a runner. The support map has high probability for the motion model (black ink) at regions with high gradients, because these areas can be uniquely matched to the specific motion models. At non-textured regions, more than one motion model matches the data, and during the E-step, equal probability to several motion models is assigned. In some sense this approach is implicitly a edge based tracker at regions with high edge gradients, and a region based tracker at regions with high texture.

2.1.3 Incorporating past estimates

Taking into account that in a physical world objects can't change their motion in an arbitrary manner, we can compute prior distributions for the blob parameters using previous time frames. Kalman filters [2] are the obvious choice for computing such priors in a recursive way. The state space of the filter are the blob parameters $\theta(t)$. Based on the Gaussian distribution of $\theta(t-1)$, the Kalman update computes the predicted mean and covariance of $\theta(t)$, which

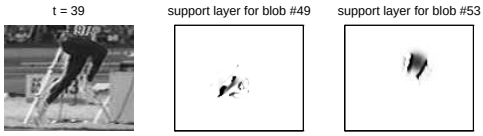

Figure 2: Two support maps for two coherent motion regions. The posteriori probabilities that a pixel location belongs to a coherent motion area is higher at areas with large image gradients (e.g. boundary of the legs)

we use as priors $P(\theta(t)|\theta(t-1), ...\theta(1))$ for the new EM iterations[1].

EM is shown to converge to a local maxima only. Therefore it is sensitive to the starting point. The Kalman filter provides an elegant solution in allowing for an "innovation" measurement relative to the predicted state ([2]). In this case the EM starting point is the predicted Kalman state.

Another interesting feature of the Kalman filter is the use of a measurement noise covariance. This is very important in case of motion aperture. For example along the boundary of a non-textured leg, the motion can only be estimated realiably perpendicular to the boundary line. Perpendicular to the boundary the covariance should peak sharply, and along the boundary it should have a large extent. The measurement noise covariance is computed using results from ([27]).

Besides viewing this method of blob segmentation as a MAP-EM estimate, where the Kalman filter provides the priors, we also can present this method as propagating a multinomial distribution (mixtures of Gaussians) of the "system state" $\theta(t)$ through time. This has relationships with the recently proposed "condensation tracker" [16].

Making only generic assumtions about the domain, the dynamical system equation of the Kalman filter is chosen to be a constant velocity update of the motion and spatial parameters. If it is known *a-priori* what specific motion is been tracked, better choices would be domain specific dynamical models learned from training data. The good performance of such models has been shown on spline based tracking of edge segments [4]. In our case, we don't know yet at this abstraction level which blob performs what specific motion. For example the lower leg segment during a certain phase of a running cycle complies to a different linear dynamical model then the upper arm segment during a different gesture. It is too early to commit to a certain motion model and this leads to the next higher abstraction level.

2.2 Mixtures of Dynamical Systems and Motor-program HMMs

Following the same principle of "soft-commitment" as we did in the blob segmentation with the hidden variables $S_k(t,x,y)$, we introduce another set of higher level hidden random variables $D_m(t,k)$, that group a sequence of blobs $\theta_k(t)$, $\theta_k(t+1)$, ... $\theta_k(t+d)$ to simple dynamical categories. The mid-level grouping into dynamical categories is done

[1] [14] have shown a straightforward method, how to extent EM for maximum a-posteriori (originally EM is only defined for maximum-likelihood without model priors).

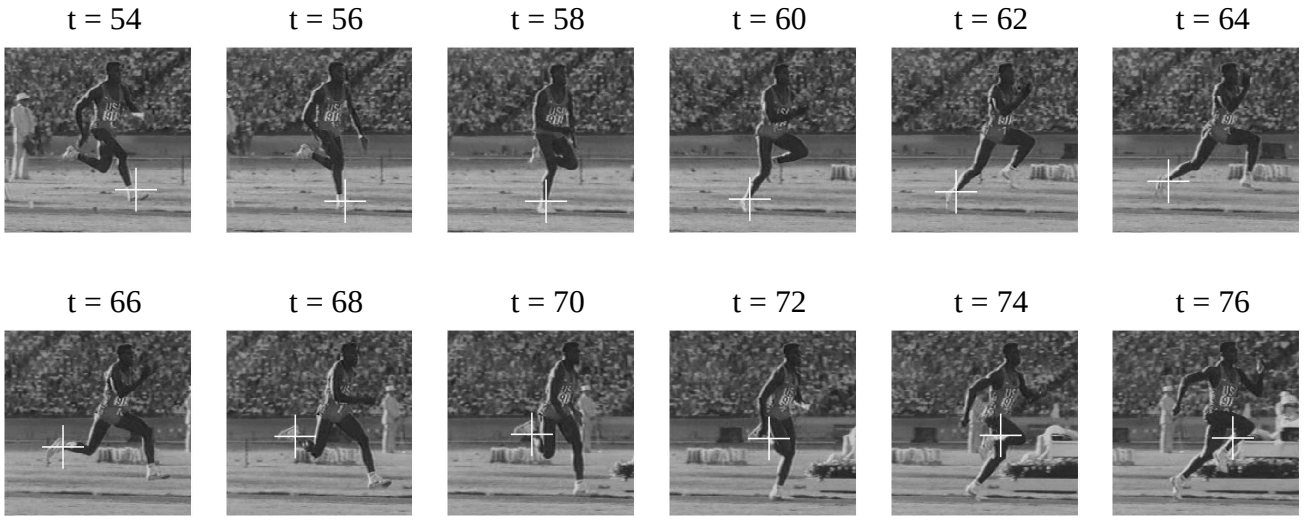

Figure 3: Footage of a running subject. The white cross close to the front foot shows the center of mass of one blob track. As you see the Kalman filter can cope with the short occlusion at time $t = 72$ by the runners hand.

across time t for each blob track separately. (Potentially we also can merge different blob tracks if they comply to the same dynamical category, and are in spatial proximity to each other.) The dynamical categories are represented with a set of M second order linear dynamical systems. Example categories are hand-waving or certain phases during a gait that can be approximated with a linear system. We call these categories "movemes" in relationship to phonemes. A complex gesture "word" should be composed out of simple movemes.

To compute the probability $D_m(t,k)$ that a certain blob $\theta_k(t)$ belongs to one of the dynamical categories m, the following notation of a discrete 2nd order stochastic dynamical system is used [4]:

$$Q(t) = A1_m \cdot Q(t-2) + A0_m \cdot Q(t-1) + B_m \cdot w \quad (11)$$

The state variable $Q(t)$ is the motion estimate of the specific blob $\theta_k(t)$. w is the system noise, and $C_m = B_m \cdot B_m^T$ is the system covariance.

Our final goal is to classify complex gestures that are composed of simple dynamical categories. If these simple phases follow in sequential order, like for example during a gait cycle, we can use Hidden Markov Models (HMM). Each HMM state corresponds to one phase, and the emission probabilities are represented by the corresponding stochastic dynamical system. The probability that blob k is in HMM state q_m at time t corresponds to the hidden variable $D_m(t,k)$. The top of figure 1 shows an example HMM topology of a 3 phase cyclic model. The transition arcs that loop back to the same state encode the prior probability of staying in the same phase across time, and the transition arcs to the next state encode the prior probability that a "phase-transition" occurs.

Unlike in the lower level groupings across space (x,y) in which we only compute a local maximum, we can compute across time t the global best segmentation using dynamic programming. The forward-backward [23] procedure provides a recursive (linear complexity over time) estimate that a complex motion model HMM_i fits a track:

$$P(\text{HMM}_i|\forall m D_m(t,k), ...D_m(1,k)) = \sum_m \alpha(m,t) \quad (12)$$

where

$$\alpha(m,t) := P(D_m(t,k)|\forall n D_n(t-1,k), ...D_n(1,k), \text{HMM}) \quad (13)$$

$$\alpha(m,t) = P(D_m(t,k) \cdot \sum_n \text{tr}_{n,m}\alpha(n,t-1) \quad (14)$$

$P(D_m(t,k)$ is the probability that dynamical system m fits blob k at time t as defined in (11), and $\text{tr}_{n,m}$ is the HMM transition probability between state n and m (transition matrix TR_{HMM}).

We do this for each complex category HMM_i and an outlier model HMM_0 (HMM_0 could be a single state HMM with a constant velocity dynamical system). Comparing the likelihoods across the different HMMs allows us to do the final high-level gesture classification.

2.3 Inducing Hybrid Dynamical Models

Although there exists a huge body of literature about models of human and biological motion dynamics including data from physiological studies, we believe that the parameters of the dynamical representations should be estimated from example data. Hand-coded domain knowledge is useful for model selection and initial values, but should be fine-tuned by statistical estimation techniques.

Given the number of linear dynamical systems M and the HMM topology, we present an iterative nonlinear training technique that is able to estimate the system parameters of each dynamical model $\phi_m := [A0_m, A1_m, B_m]$, and the entries $\text{tr}_{m,n}$ of the HMM transition probability matrix TR_{HMM}.

Given example motion trajectories $Q(1), Q(2), ... Q(T)$ and a partition into subsequences $Q(t)$,$Q(t+1)$,... $Q(t+d)$, where each subsequence belongs to exactly one dynamical

model m, it is straightforward to estimate the parameters ϕ_m using a linear maximum-likelihood system identification procedure. For example a training sequence of a walking subject could be partitioned into the time intervals while the leg has ground support (dynamical model $m = 1$), and into time intervals while the leg is swinging in the air (dynamical mode $m = 2$). In this case, following log likelihood function is maximized for each interval (i.e. dynamical model) separately (using the notation in [4]):

$$L_{linear}(Q(t), Q(t+1), ..., Q(t+d)|\phi_m) = -\frac{1}{2}\sum_{n=t+2}^{d} |B^{-1}(Q_n - A0_m Q_{n-2} - A1_m Q_{n-1})|^2 - (d-2)\log|B| \quad (15)$$

Not knowing such a partition a-priori, we apply an iterative system identification technique, that is an extension of the Baum Welch HMM estimation algorithm [23] (another incarnation of EM). It will maximize the total log likelihood of a set of M dynamical systems and the corresponding HMM:

$$L_{hybrid}(Q(1), ..., Q(T)|\phi_1, ..., \phi_M, \mathrm{TR}_{\mathrm{HMM}}) = \log(\sum_{\text{partition}} P(Q(1)..., Q(T)|\text{partition}, \phi_1, ..., \phi_M) \cdot P(\text{partition}|\mathrm{TR}_{\mathrm{HMM}})) \quad (16)$$

As shown in [23], we don't need to sum over all possible partitions in (16) to converge to a local maximum of the log likelihood. Instead, at each iteration a "soft" partition D is computed using the current guess of the model parameters (E-step), and the partition is used to compute a new model parameter estimate (M-step).

A soft partition D of the training set is equal to the previously described hidden variables $D_m(t, k)$. (For convenience we drop the k parameter, which was used in recognition mode as an index to the blob hypothesis k). $D_m(t)$ is the probability that training example $Q(t)$ was "generated" by dynamical system ϕ_m.

In the E-step the posterior $P(D_m(t)|Q(1)..., Q(T), \phi_1..., \phi_M, \mathrm{TR}_{\mathrm{HMM}})$ are computed with the forward-backward recursion (dynamic programming with linear complexity of the length of the training set). Given these probabilities, it turns out that the M-step is equal to maximizing following expected log likelihood for each model m:

$$E\{L_{weighted}(Q(1), Q(2), ..., Q(T)|\phi_m)|D\} = -\frac{1}{2}\sum_{n=3}^{T} D_m(t)|B^{-1}(Q_n - A0_m Q_{n-2} - A1_m Q_{n-1})|^2 - (\sum_{n=3}^{T} D_m(t))\log|B| \quad (17)$$

This term can be maximized by solving a linear equation.

The new estimate of the HMM transition probability matrix $\mathrm{TR}_{\mathrm{HMM}}$ is computed with the conventional Baum-Welch update.

This is an iterative procedure, that has to start with an initial guess of the model parameters, or an initial soft partition D. In case we have an intuition about the present dynamics, we could provide this knowledge with initial model parameters, and let our procedure "fine-tune" these parameters. As we will show in the next section, it is also possible to converge to good model parameters with a random initialization of a partition D.

3 Experiments

Although the described techniques are very general, we demonstrate its feasibility in this paper on the domain of human gait categories.

Our training and validation data are 33 sequences of 5 different subjects performing 3 different gait categories: **running, walking, and skipping**. An independent test set of two additional subjects was set aside for recognition experiments. The training sequences contained tracked MLD markers or were hand-labeled at the limb joints, so we can use detailed "ground-truth" data for the training process. The independent test set did not contain markers. Figure 3 and 7 shows three test sequences. As you can see, the test sequences are recorded in cluttered environment. The running sequence (sub-sampled from a slow motion recordings of the Olympic Games) contains many other moving objects, including the background, which makes this data set a hard segmentation and tracking problem.

3.1 Training and validation of gait models

The training process was done with the "ground-truth" data set (translation and angular velocities of the limb segments computed at the center of the lower or upper leg), but no labels were given that indicate at which phase the trajectory was during a gait cycle (partition). Different sequences started at different parts of the walking, running, or skipping cycle.

We experimented with different number of dynamical models, and concluded that 4 dynamical models (and HMM states) per gait is a reasonable choice (e.g. a skipping cycle has 2 ground support phases and 2 non-support phases).

Half of the ground-truth data was used for training, and the other half was used for validation (the set of "training people" and "validation people" was disjunct).

The training procedure was started with a partition, where each $D_m(t)$ was assigned to $\frac{1}{4}$ (each 4 models are equal likely for each training sample). We added a very small random value ($+0.001\cdot$ white noise). The random component was necessary; otherwise the iterative learning procedure would induce the same parameters for all 4 dynamical models.

The state space of the dynamical models was the translation and angular velocity of the lower leg limb. The first plot in figure 4 shows one example sequence of 2 running cycles. The next 4 plots show the first 4 iterations of the training procedure. Each plot illustrates the current partition D. Each row corresponds to a specific HMM state m, and each column to a training sample at time t. Black ink means $D_m(t) = 1$ and white means $D_m(t) = 0$. In the first 2 iterations, the training procedure is very "uncommitted" to a partition, and in the next two iterations, the 4 models tune into certain phases, and therefore the partitions peak sharper. The HMM topology constrained the partition to stay for a certain time with one HMM state, and

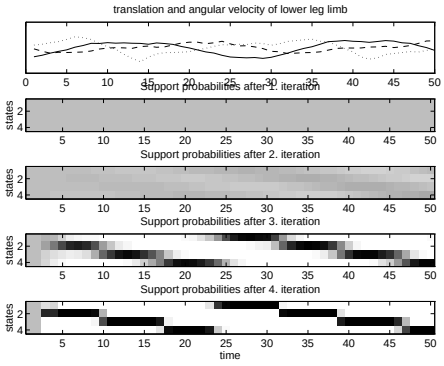

Figure 4: Top row: The solid line is the x-translation, the dashed line is the y-translation, and the dotted line the angular velocity over time. The next 4 plots are the support probabilities of the HMM states (4 rows in each plot for 4 states each, black ink means high probability).

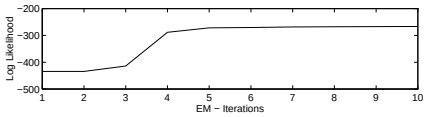

Figure 5: Increasing log-likelihood during 10 EM-iterations of the training process for the running model.

then transit to the next state. Repeating this experiment with different random initializations resulted in different time shifted partitions. But at each experiment, one cycle through the HMM clearly corresponded with one gait cycle. As you can see in figure 4, HMM state 1 corresponds to the phase where the x-translation (full line in top plot) has a negative minimum, and state 3 corresponds to the phase where the x-translation has a positive maximum. Therefore the linear dynamical system $m = 1$ covers the dynamics of the translation and angular velocity at that phase of a running cycle. As we will see later, it actually transcribed that phase on the independent test footage.

Figure 5 plots the log likelihood of the total hybrid model (16). From EM iteration 3 to 4 we see a sharp rise of the log likelihood value, which indicates that it was "falling" at this point into the sharply peaked partition of the training set, and converged after that to a local maximum. The plots for the 2 other gait models look similar.

The increasing log-likelihood is just one indication that our learning technique estimated a good fitting model. We also need to make sure that the models don't overfit the data, so that they are able to generalize. This is done by testing the models on the independent validation set collected from different subjects. The number of correct classified gait cycles in the validation set varied between 86% correct to 93% correct, depending on the random initialization. The performance was measured in computing the likelihood of each of the 3 gait HMMs after one walk cycle (walking-HMM, running-HMM, skipping-HMM), and choosing the highest likelihood as the classification category.

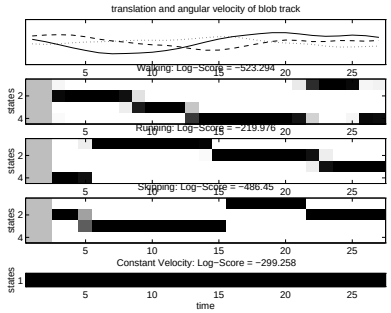

Figure 6: Top plot shows the velocities estimated by one blob track in the winning sequence (track of frontal foot). The second plot show the HMM transcription of this track by the Walking-HMM, the third blob by the Running-HMM, and the fourth blob by the Skipping-HMM. The last plot show the outlier model, which is only one state, and therefore does not partition the input sequence. The running HMM has the highest log-likelihood.

3.2 Tracking and recognizing gaits in video sequences

The final experiment was to apply the learned dynamical models and HMMs for the segmentation and classification task on another set of unseen input image sequences. For this task we introduced an additional outlier model HMM_0 that had only 1 state and a constant velocity dynamical model. This was used to discard blob tracks that are very unlikely to belong to leg segments.

Given a test image sequence containing an unsegmented and unspecified number of gait cycles ($30 - 60$ frames), we propagate 96 blob hypotheses (resulting from the tile grid initialization) and one background layer through all image frames. If the score of a blob (mixing coefficient) reached a lower limit (spurious blob), the blob hypothesis was discarded. The number of blob hypotheses might seem very large, assuming the human body could be described with only a few coherent motion regions. As you can see, complex scenes like Figure 3 contains many other moving objects that potentially could belong to something recognizable. We found it is better to track too many hypotheses than too few. With the help of the outlier model (HMM_0) a large number of blob tracks could be discarded as "non-gait" tracks (usually in the order of $75\% - 90\%$ of all tracks). The remaining blob tracks were classified by one of the three gait-models. The winning model (highest likelihood) was the final gait classification. In figure 3 the track following the shoe of the runner had the highest likelihood. The velocities and HMM transcription of this track by all three gait models is shown in figure 6. The running-HMM had the highest log-likelihood and therefore the sequence was classified correctly with this category.

Figure 8 shows the same models transcribing the track with the highest likelihood in the walking sequence of figure 7a. In this case the walk model on a track at the middle of the lower leg had the highest likelihood. The partition of the gait cycle also fits the partition found in the training example.

In some test sequences the blob track with the highest likelihood does not cover a lower leg track. Figure 7b shows

Figure 7: Example images of the second subject.

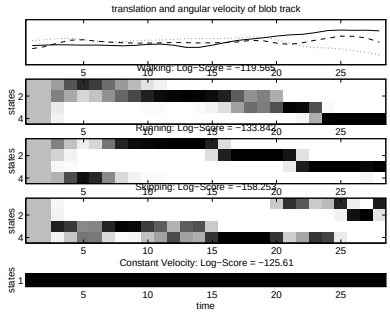

Figure 8: HMM transcriptions of one blob track in the walking sequence. As you see the walking model (second plot) has the highest log likelihood.

such a case where the "winning" blob tracked the upper leg segment. It was classified correctly (skipping-HMM). The dynamics of a skipping lower leg was the closest model, even if the dynamics were measured at the wrong body part in the test sequence. The second highest likelihood was a blob that tracked one leg for a while and switched to the other leg (caused by occlusion). It was classified incorrectly as running.

We are currently performing more experiments on larger datasets to further verify this very encouraging classification performance.

4 Related Work

One influential paper to many other motion-based approaches is the classic Moving Light Display experiments by Johansson [18]. Seeing lights attached to the joints of an actor, human subjects were able to distinguish human gaits, dance styles, stair climbing, or even can identify gender or idendity.

The earliest computer vision attempt to recognize human movements was reported in O'Rouke and Badler [21] working on synthetic images using constraint satisfaction techniques. Systems that deal with real input data and edge fitting to explicit structural models are reported by [15, 25, 12, 24, 19]. Techniques that don't use such explicit model knowledge are usually estimated from example image sequences. Common representations are space-time curves [20], and appearance based representations [7, 5, 28, 31]. Some of these techniques use HMMs to cover the temporal structure. [22] propose a technique that looks for appearance based periodicity, and [11] measure spatio-temporal angle histograms to recognize hand gestures. Motion based recognition techniques were presented in [8, 10, 3], and a system based on color blobs is described in [30].

Some systems ignore the low-level feature extraction and only focus on higher level representations and recognition strategies [6, 13].

Most explicit model approaches assume certain domain constraints, like calibrated cameras, known background, initial pose, and uncluttered environments that make edge matching feasible. In contrast, most appearance based techniques do not impose such constraints but are very specialized to the given training data. With the goal in mind to cover a large set of human actions in unconstrained environment, purely appearance based techniques might require an immense amount of training data. Combining layered image representations with dynamical models and Hidden Markov Models in a coherent probabilistic framework, our approach is an attempt to find the right balance of supplied structure and learned parameters.

5 Further work and conclusion

We introduced a new method for probabilistic segmenting, tracking, and classifying complex dynamics in video sequences. Our approach is unique in the way it decomposes the domain, and incorporates the different levels of abstraction using mixtures models, EM, recursive Kalman, and Markov estimation. A feasible computation can be done by exploiting various conditional independence assumption across the abstraction levels and time, and multi modal approximations within the levels. We demonstrated the technique on the domain of classifying human gait categories in cluttered video sequences.

Many domain constraints are not exploited yet. Besides additional features like texture coherence and more complex shape representations, experiments are in progress to group blob pair hypotheses together based on kinematic and further dynamical constraints. Ultimately the system should also estimate 3D pose and additional dynamical state variables, like speed. To apply this technique to larger corpuses with more categories, a comprehensive "moveme" decomposition is needed. Speech recognition is currently applied to vocabularies of more then $20,000$ words and 40 to 70 atomic phoneme categories. In the visual domain we are still far away from this goal due to the much larger complexity of the problem. But we have shown early steps that follow such principles of coherent probabilistic reasoning in a multi-level framework.

6 Acknowledgements

I would like to thank Jitendra Malik, Jerry Feldman, David Forsyth, Jianbo Shi, Phil McLauchlan and other members of the U.C. Berkeley Computer Vision Group for helpful discussions, and Nigel Goddard for providing the MLD data. This research was fundend by Interval Research Corp., California MICRO program, and Office of Naval Research (ONR N00014-92-J-1617).

References

[1] Serge Ayer and Harpreet S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *Int. Conf. Computer Vision*, pages 777–784, Cambridge, MA., 1995.

[2] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1987.

[3] M.J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *ICCV*, 1995.

[4] A. Blake, M. Isard, and D. Reynard. Learning to track the visual motion of contours. In *J. Artificial Intelligence*, 1995.

[5] A.F. Bobick and A.D. Wilson. A state-based technique for the summarization and recognition of gesture. In *Proc. Int. Conf. Computer Vision*, 1995.

[6] L.W. Campbell and A.F. Bobick. Recognition of human body motion using phase space constraints. In *ICCV*, 1995.

[7] T.J. Darrell and A.P. Pentland. Classifying hand gestures with a view-based distributed representation. In *NIPS*, 1994.

[8] J.W. Davis and A.F. Bobick. Real-time recognition of activity using temporal templates. In *to appear in Workshop on Applications of Computer Vision*, 1996.

[9] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(B), 1977.

[10] I.A. Essa and A.P. Pentland. Facial expression recognition using a dynamic model and motion energy. In *ICCV*, 1995.

[11] W. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face- and Gesture-Recognition*, 1995.

[12] D.M. Gavrila and L.S. Davis. Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *Proc. of the Int. Workshop on Automatic Face- and Gesture-Recognition, Zurich, 1995*, 1995.

[13] N. H. Goddard. *The Perception of Articulated Motion: Recgonizing Moving Light Displays*. PhD thesis, Dept. of Comp.Sci., Univ. Rochester, 1992.

[14] Peter J. Green. On use of the EM algorithm for penalized likelihood estimation. *Journal of the Royal Statistical Society, B*, 52(3):443–452, 1990.

[15] D. Hogg. A program to see a walking person. *Image Vision Computing*, 5(20), 1983.

[16] M. Isard and A. Blake. Contour tracking by stochasitc propagation of conditional density. In *Proc. ECCV*, 1996.

[17] A. Jepson and M.J. Black. Mixture models for optical flow computation. In *Proc. IEEE Conf. Computer VIsion Pattern Recognition*, pages 760–761, New York, 1993.

[18] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.

[19] M.K. Leung and Y.H. Yang. First sight: A human body outline labeling system. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(4):359–377, April 1995.

[20] S. A. Niyogi and E.H. Adelson. Analyzing and recognizing walking figures in xyt. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 469–474, Seattle, June, 1994.

[21] J. O'Rourke and N. I. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2(6):522–536, November 1980.

[22] R. Polana and R. Nelson. Detecting activities. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, 1993.

[23] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recogntion. In *Readings in Speech Recogntion*, 1989.

[24] J.M. Regh and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proc. Int. Conf. Computer Vision*, 1995.

[25] K. Rohr. Incremental recognition of pedestrians from image sequences. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 8–13, New York City, June, 1993.

[26] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

[27] E.P. Simoncelli, E.H. Adelson, and D.J. Heeger. Probability distributions of optical flow. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, 1991.

[28] T. Starner and A. Pentland. Visual recognition of american sign language using hidden markov models. In *Proc. of the Int. Workshop on Automatic Face- and Gesture-Recognition, Zurich, 1995*, 1995.

[29] Y. Weiss and E.H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *CVPR*, 1996.

[30] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *SPIE Conference on Integration Issues in Large Commercial Media Delivery Systems*, volume 2615, 1995.

[31] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time sequential images using hidden markov models. In *CVPR*, 1993.