Learning behavioral manifolds for recognition and synthesis

Matthew Brand

MERL — a Mitsubishi Electric Research Laboratory
201 Broadway, Cambridge, MA 02139

Abstract

We consider the problem of inferring the behavior of a target system given cues in the form of a time-series taken from another co-evolving system. We show how to approximate the target system's behavioral manifold with a probabilistic model, then compute trajectories on this manifold that are optimal w.r.t. the target system's dynamics and w.r.t. configural clues in the cue signal. The principal advances are an entropy-minimization framework that simultaneously estimates the structure and parameterization of probabilistic models, and a closed-form solution for calculating geodesics on the behavior manifold relative to constraints implicit in the cue signal. The resulting method integrates information over the entire time-series to disambiguate many-to-many mappings between cue configurations and target configurations, yet can run in linear time. We demonstrate with systems that infer 3D nonrigid motion from image cues and, in the case of facial motion, from audio.

1 Introduction

In Plato's allegory of the cave, cave-dwellers try to infer the world from the shadows it casts on the rocky wall, with no experience of its true 3D forms [Pla74, book 7]. This is a non-trivial inference problem: Many forms can cast the same shadow, and many shadows can fall from the same form. Therefore, inference must be a search for a most probable *sequence* of events that best explains a sequence of observables. Plato stipulated prior knowledge of what the true forms look like and how they cast shadows, but in real life, one has experience both of 3D forms and their shadows, and how they evolve in time.

In this paper we propose learning all the relevant knowledge from synchronized observations of a system's true state and some cue signal—perhaps its appearance, but perhaps just measurements of another system that approximately co-evolves with our system of interest. Our goal is to approximate the target system's dynamical manifold via learning, then learn a mapping from paths taken by the cue system to paths over the target system's manifold. Our framework will support traditional vision tasks such as recognition, classification, and pose estimation, plus a novel one: synthesis. We demonstrate with facial animation synthesized from voice data, and 3D body articulation synthesized from 2D "shadow" data.

Technically, we seek to infer a hidden sequence $\boldsymbol{X} \doteq \{\boldsymbol{x}_1, ...\boldsymbol{x}_T\}$ from an observable sequence $\boldsymbol{Y} \doteq \{\boldsymbol{y}_1, ...\boldsymbol{y}_T\}$. We call this inference a *path map*, as opposed to a point map, because the function $f : \boldsymbol{Y} \rightarrow \boldsymbol{X}$ operates on sequences of variable dimension. We assume that the process generating $\boldsymbol{X}$ is Markovian and stationary, so that we can model it with a finite number of time-invariant parameters. There are two special cases where the maximum *a posteriori* problem $p(\boldsymbol{X}|\boldsymbol{Y})$ is analytic: (1) when the process and observations evolve linearly ($p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ and $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ are both Gaussian with known covariances), in which case Kalman filtering and smoothing are optimal; and (2) when $\boldsymbol{x}_t$ is a discrete-valued variable, in which case hidden Markov models are optimal. In virtually all other cases the posteriors and marginals appear to be analytically intractable, despite a 30-year literature and hundreds of papers. The methods presented below combine the desirable properties of the two tractable models in a way that covers a very general nonlinear case (nonlinear system evolution, arbitrary mixture-of-Gaussian distributions $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$), but only require linear-time inference algorithms.

2 Behavior Manifolds

A behavior manifold is a locus of all possible system pose and velocity configurations embedded in a higher-dimensional measurement space, like a crumpled paper in 3-space. In figure 2 we illustrate a 4D manifold (u, v, and most probable du, dv at each point) and its embedding in 6-space. Note that the embedding is highly nonlinear; the manifold curls over and even self-intersects, thus any linear projection will conflate multiple manifold points. Actual system histories are trajectories on this manifold. Usually there are far too few samples to support an estimate of the manifold's geometry.

We model a behavior manifold by estimating a continuous probability distribution from observed

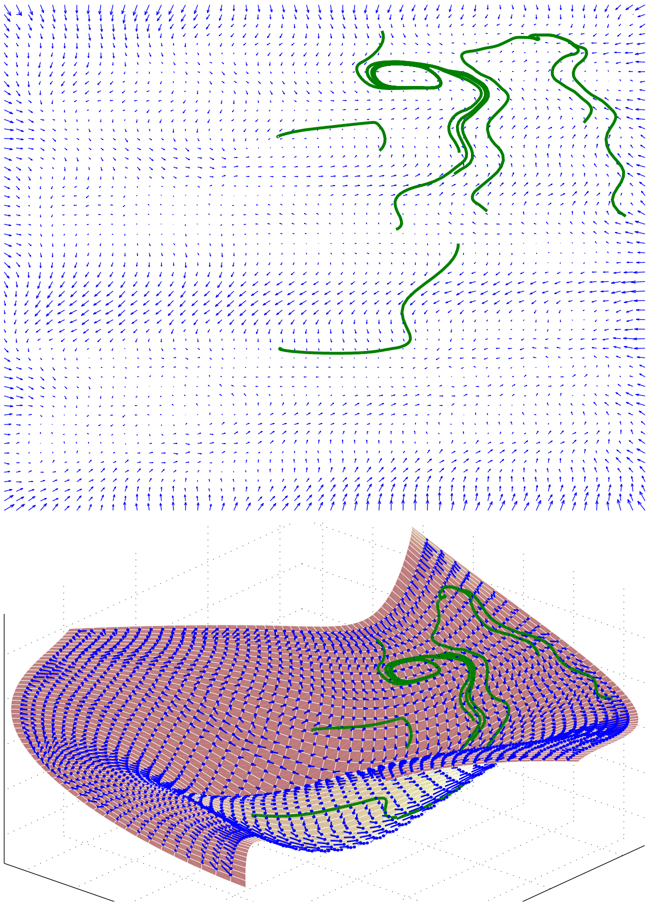

Figure 1: Schematic of a behavior manifold and its embedding in a measurement space. Arrows indicate most likely velocity at each point; lines signify actual observed system trajectories.

trajectories. This will lend itself to recognition and classification tasks, and more importantly, to a good degree of extrapolation to trajectories that do not coincide with the training data.

We can model this manifold by identifying neighborhoods in which the manifold is roughly flat, fit a linear subspace to each neighborhood, then probabilistically "glue" together these fittings into a piecewise linear approximation. It is convenient to describe each neighborhood with a multivariate Gaussian. The covariance matrix does a double service, specifing how pose and velocity linearly covary, and also encoding, implicitly, the extent of its neighborhood (figure 2). This piecewise linear approximation is simpler than previous schemes (e.g., [BO95]) and will lead to closed-form solutions for all quantities of interest.

Because the manifold contains dynamical information the "glue" between neighborhoods must describe how quickly the system passes from neighborhood to neighborhood; this is conveniently expressed as a matrix of transition probabilities.

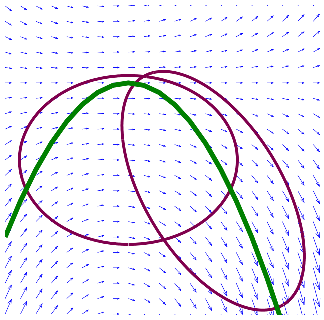

Figure 2: Two neighborhoods in which velocity covaries linearly with position. The ellipses represent isocontours of Gaussians whose covariances capture this relation. The parabola is one of many piecewise quadratic trajectories from which these neighborhoods might be estimated.

Not surprisingly, our manifold approximation is essentially a discrete-state hidden Markov model (HMM), with each Gaussian being the output of a hidden state. But HMMs are typically very poor models for our purposes because conventional training techniques [Bau72, Rab89] are extremely suboptimal with regard to placement of the Gaussian outputs. Moreover the estimated transition probabilities rarely improve on modeler's nearly-blind guess about connectivity, e.g., a full transition matrix will let every neighborhood pass directly to every other neighborhood, almost certainly violating the manifold's topology. Consequently, no one would expect a random walk on an HMM to produce a sequence that has the same long-term structure as the time-series in the training set, yet that is what we require of a good model. To overcome this problem, we introduce a fast, monotonic learning procedure based on entropy minimization which produces very high quality models and, incidentally, eliminates overfitting by unifying parameter and model structure estimation. Entropy minimization restructures the transition matrix to reflect the topology of the manifold and shifts Gaussians to locales where the covariance is narrowest, meaning that the linear relations between dimensions are strongest. This will be illustrated later in §3.0.1.

With some additional training, this model can be made to predict the *target* system's hidden state from clues contained in another co-evolving *cue* signal, e.g., a 3D figure's orientation from a sequence of shadows. This method optimally integrates information over time, using clues from one end of the sequence to disambiguate mappings at the other end, if necessary.

Perhaps the most interesting result is that we can predict actual target *signals* from cue signals, and the results do in fact look realistic. In particular, we give a linear-time closed-form solution for the most probable trajectory on the behavior manifold given a sequence of the co-evolving signal. We use this technique to synthesize video-realistic facial animation from voice data, and 3D body pose from 2D tracking data. All algorithms are linear in the length of the time series and the number of model parameters; all applications run slightly faster than real-time in interpreted Matlab code on current workstations.

3 Learning

We briefly review the entropic training framework here, and refer readers to [Bra99b, Bra99a] for details and derivations. We begin with a dataset $\boldsymbol{X}$ and a model whose parameters and structure are specified by the matrix $\boldsymbol{\theta}$. In conventional training, one guesses the sparsity structure of $\boldsymbol{\theta}$ in advance and merely re-estimates nonzero parameters to maximize the likelihood $P(\boldsymbol{X}|\boldsymbol{\theta})$. In entropic training, we learn the size of $\boldsymbol{\theta}$, its sparsity structure, and its parameter values simultaneously by minimizing the three entropies

$$H(\boldsymbol{\omega}) + D(\boldsymbol{\omega}\|\boldsymbol{\theta}) + H(\boldsymbol{\theta}), \tag{1}$$

where $H(\boldsymbol{\omega})$ is the entropy of the data's expected sufficient statistics and can be interpreted as the expected coding cost of the data relative to the model; $D(\boldsymbol{\omega}\|\boldsymbol{\theta})$ is the cross-entropy between the sufficient statistics and the model and measures the expected coding cost of aspects of the data not captured by the model; and $H(\boldsymbol{\theta})$ is an entropy measure on the model itself and, depending on its formulation, can be interpreted either as the entropy of the distribution or the expected coding costs of the model itself.

Minimizing eqn. 1 is equivalent to maximizing the posterior probability given by Bayes' rule,

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[P(\boldsymbol{\theta}|\boldsymbol{X}) \propto P(\boldsymbol{X}|\boldsymbol{\theta}) e^{-H(\boldsymbol{\theta})} \right]. \tag{2}$$

This prior can be shown to be a continuous approximation of the expected Kolmogorov complexity of the model [Bra98]. Entropic estimation also has interpretations as optimal compression and free energy minimization [Bra99a].

Given a factorizable model such as an HMM, the maximum *a posteriori* (MAP) problem decomposes into a separate equation for each independent parameter $\theta_i \in \boldsymbol{\theta}$, each having its own entropic prior. For spread parameters such as the Gaussian covariance $\boldsymbol{K}$ over N samples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, the estimator is

$$\hat{\boldsymbol{K}} = \frac{\sum_i^N \boldsymbol{x}_i \boldsymbol{x}_i^\top}{N + Z}. \tag{3}$$

For conditional probabilities and more generally for multinomial densities over N alternatives, the estimator

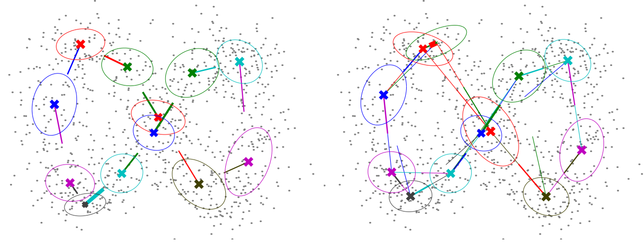

Figure 3: HMMs estimated entropically and conventionally from identical initial conditions given figure-8 data. $\times$s and ellipses indicate Gaussian means and covariances of the hidden states; half-arcs point to transitionable states; data points are plotted in gray.

is given by the fixpoint

$$\lambda = \frac{1}{N} \sum_i^N -\frac{\omega_i}{\theta_i} - Z \log \theta_i - Z \tag{4}$$

$$\hat{\theta}_i = \frac{-\omega_i / Z}{W(-\omega_i e^{1+\lambda/Z}/Z)}, \tag{5}$$

where $\boldsymbol{\theta}$ is a vector of multinomial parameters; $\boldsymbol{\omega}$ is a vector of their sufficient statistics (e.g., event counts); W is the Lambert inverse function satisfying $W(x)e^{W(x)} = x$; and Z is negative temperature term. Exponentiating the prior by Z and letting the temperature $T = 1 - Z$ decay to zero, we obtain deterministic annealing within expectation-maximization for free, thereby speeding learning and turning EM into a quasi-global optimizer. In [Bra99a] we show how estimators are applied to variety of probabilistic models, yielding very fast learning algorithms; the case of HMMs is extensively treated in [Bra99b]. MAP re-estimation extinguishes excess parameters and maximizes the information content of the surviving parameters. Consequently, if we begin with a large fully-connected HMM, the training procedure whittles away all parts of the model that are not in accord with the hidden structure of the signal. This allows us to learn the proper size[1] and sparsity structure of a model. Frequently, entropic estimation of HMMs recovers a finite-state machine that is very close to the mechanism that generated the data.

3.0.1 Figure-eight example

Figure 3.0.1 shows an HMM entropically estimated from very noisy samples of a system that orbits in a figure-eight. The true system is a 2D manifold (phase and its rate of change) embedded in a 4D measurement space (observed 2D position and velocity); the HMM

[1] Note that there is no "correct" number of states for a continuous signal, but if there is insufficient data to support many states, some will be automatically removed.

approximates this manifold with neighborhoods of locally consistent curvature in which velocity covaries linearly with position. Note that even though the data is noisy and has a continuation ambiguity where it crosses itself, the entropically estimated HMM recovers the deterministic structure of the system. The conventionally estimated HMM gets "lost" at the crossing, bunching states at the ambiguity and leaving many of them incorrectly over-connected, thus allowing multiple circuits on either loop as well as numerous small circuits on the crossing and on the lobes. In fact, is possible for a random walk to traverse the conventionally estimated HMM backwards! It is the additional precision of entropic models that makes them significantly outperform their conventionally estimated counterparts, making possible their application to manifold modeling.

3.1 Remapping

Using entropic estimation, we learn a dynamical model from the time-series of poses and velocities of the target system. The learning algorithm gives us an HMM plus a vector of hidden state probabilities for each frame in the training set. Using that, we estimate a second set of output probabilities, given each state, of the synchronized cue signal. This associates cue signal features to each target system state, resulting in a new HMM which has the dynamics of the target system, but is driven by the cue signal.

4 Analysis

Given a new cue signal, we apply the Viterbi dynamic programming algorithm [For73] to the cue HMM to find the most likely sequence of predicted target states. Although it is steered by information in the new cue signal, this state sequence is constrained to follow the natural dynamics of the target system. Readers familiar with HMMs will note that the number of possible state sequences is exponential in the number of time steps, and that the Viterbi sequence, while most likely, may only represent a small fraction of the total probability mass. E.g., there may be thousands of slightly different state sequences that are nearly as likely as the Viterbi sequence. If this were the case, the state sequence would be a very poor representation of the observed signal, and any calculations downstream would suffer. It is possible to integrate over all possible sequences via forward-backward analysis, obtaining an occupancy matrix $\gamma_{s,t} \doteq p(\text{state } s \text{ at time } t | \text{sequence } \boldsymbol{Y})$. This captures all the information the model can "extract" from the cue signal $\boldsymbol{Y}$, and, if it differs from the Viterbi state sequence, it may increase the complexity of the synthesis step. However, we found this to be unnecessary with entropically estimated models because entropy minimization concentrates the probability mass on the optimal Viterbi sequence.

5 Synthesis

For synthesis, we seek a short, smooth trajectory that passes through regions of high probability density in configuration/velocity space at the right time—a geodesic. Prior approaches to trajectory estimation typically involve maximizing an equation having a likelihood term plus penalty terms for excess length and/or kinkiness and/or point clumpiness. The user must choose a parameterization and weighting for each term. This leads to variational techniques that are often approximate, iterative, and computationally expensive (e.g., [SJ96]). Moreover, the equation may have many local maxima and one cannot tell whether the found maximum is near-optimal or mediocre.

Our current setting constrains the problem so significantly that a globally optimal closed-form solution is available. Because we model both pose and velocity, the target Gaussians alone contain enough information to completely specify the smooth trajectory that is most consistent with the target dynamics and a given state sequence.

The HMM's output Gaussians together with its hidden state probabilities $\boldsymbol{\gamma}^{\boldsymbol{Z}}_{s,t}$ define a mixture of Gaussians over trajectory-space; we seek the optimal trajectory $\boldsymbol{X}^* = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, ...\} = \text{argmax}_{\boldsymbol{X}}\, p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{\gamma}^{\boldsymbol{Z}}_{s,t})$. Let $[\boldsymbol{\mu}_i, \dot{\boldsymbol{\mu}}_i]$ be the mean position and velocity for state i, and $\boldsymbol{K}_i$ be a full-rank covariance matrix relating position and velocity across all dimensions. The mixture probability is

$$p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{\gamma}^{\boldsymbol{Z}}_{s,t}) \doteq \prod_t \left[\sum_c \boldsymbol{\gamma}^{\boldsymbol{Z}}_{s,t} \mathcal{N}(\tilde{\boldsymbol{x}}_{s,t}; \boldsymbol{K}_s)\right] \mathcal{N}_P(\dot{\boldsymbol{x}}_t; \boldsymbol{K}_P), \tag{6}$$

where the vector $\tilde{\boldsymbol{x}}_{s,t} \doteq [\boldsymbol{x}_t - \boldsymbol{\mu}_s; (\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) - \dot{\boldsymbol{\mu}}_s]$ is the mean-subtracted target position and velocity at time t; and $\mathcal{N}_P(\dot{\boldsymbol{x}}_t)$ is an optional Gaussian prior on velocities, usable for smoothing. We lower-bound with the log-sum inequality, obtaining

$$p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{\gamma}^{\boldsymbol{Z}}_{s,t}) \geq e^{-\frac{1}{2}\left(\sum_t \left[\sum_s \boldsymbol{\gamma}^{\boldsymbol{Z}}_{s,t} \tilde{\boldsymbol{x}}_{s,t} \boldsymbol{K}_s^{-1} \tilde{\boldsymbol{x}}_{s,t}^\top\right] + \dot{\boldsymbol{x}}_t \boldsymbol{K}_P^{-1} \dot{\boldsymbol{x}}_t^\top + c\right)}. \tag{7}$$

It is not unusual for eqn. 6 to be a mixture of $O(10^{5+})$ Gaussians; eqn. 7 essentially fits a single Gaussian around $p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{\gamma}^{\boldsymbol{Z}}_{s,t})$ and thereby gives the approximate posterior mean. How good is this strategy? Note that the bound in eqn. 7 tightens as the entropy $H(\boldsymbol{\gamma}^{\boldsymbol{Z}}_{s,t}) \doteq \langle -\log \boldsymbol{\gamma}^{\boldsymbol{Z}}_{s,t} \rangle_{\boldsymbol{\gamma}^{\boldsymbol{Z}}_{s,t}} \to 0$ because the mixture becomes sparse, with a single Gaussian dominating each time frame. At $H(\boldsymbol{\gamma}^{\boldsymbol{Z}}_{s,t}) = 0$, the joint $p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{\gamma}^{\boldsymbol{Z}}_{s,t})$ *is* a single Gaussian; so the bound and its solution are both exact. It is entropy minimization that makes eqn. 7 viable, by concentrating the probability mass on the Viterbi sequence, so that $H(\gamma^{\boldsymbol{Z}}_{s,t}) \approx 0$. This licenses an even faster way to compute a MAP trajectory directly from the Viterbi sequence $s(1), ..., s(T)$:

$$\begin{aligned} \boldsymbol{X}^* &\approx \underset{\boldsymbol{X}}{\operatorname{argmax}} \log \prod_t \mathcal{N}(\tilde{\boldsymbol{x}}_t; \boldsymbol{K}_{s(t)}) \mathcal{N}_P(\dot{\boldsymbol{x}}_t; \boldsymbol{K}_P) \\ &= \underset{\boldsymbol{X}}{\operatorname{argmin}} \frac{1}{2} \sum_t \tilde{\boldsymbol{x}}_t \boldsymbol{K}^{-1}_{s(t)} \tilde{\boldsymbol{x}}_t^\top + \dot{\boldsymbol{x}}_t \boldsymbol{K}_P^{-1} \dot{\boldsymbol{x}}_t^\top + c \end{aligned} \quad (8)$$

where $\tilde{\boldsymbol{x}}_t \doteq [\boldsymbol{x}_t - \boldsymbol{\mu}_{s(t)}; (\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) - \dot{\boldsymbol{\mu}}_{s(t)}]$. Eqn. 7 and eqn. 8 are optimized by solving the sparse system of linear equations:

$$\forall_{t=0}^T \forall_{j,i} \left\{ \left[\begin{array}{c} -B_j - D_j \\ A_j + B_j + C_j + D_j + D_i \\ -C_i - D_i \end{array} \right]^\top \left[\begin{array}{c} x_{t-1} \\ x_t \\ x_{t+1} \end{array} \right] = F_j - E_i \right\}, \quad (9)$$

where $\left[\begin{smallmatrix} A_j & B_j \\ C_j & D_j \end{smallmatrix}\right] \doteq \Sigma_j^{-1}, E_j \doteq [C_j \;\; D_j] [\mu_j \;\; \dot{\mu}_j]^\top, F_j \doteq [A_j \;\; B_j] [\mu_j \;\; \dot{\mu}_j]^\top + E_j$ and each constraint in eqn. 9 is weighted by $\gamma_{j,t}\gamma_{i,t+1}$ in the least-squares solver. The endpoints are obtained by dropping the appropriate terms from equation 9:

$$\forall_j \left\{ \left[\begin{array}{c} D_j \\ -C_j - D_j \end{array} \right]^\top \left[\begin{array}{c} x_0 \\ x_1 \end{array} \right] = -E_j \right\} \quad (10)$$

$$\forall_j \left\{ \left[\begin{array}{c} -B_j - D_j \\ A_j + B_j + C_j + D_j \end{array} \right]^\top \left[\begin{array}{c} x_{T-1} \\ x_T \end{array} \right] = F_j \right\} \quad (11)$$

which are weighted by $\gamma_{j,1}$ and $\gamma_{j,T}$, respectively. In the low-entropy or Viterbi case, the occupancy matrix $\gamma_{i,t}$ collapses to an indicator function and the system of equations can be set up in linear time. Otherwise, setup time is quadratic in the number of states at each time step that have non-vanishing probabilities.

Eqns. 9–11 give a block-tridiagonal system of linear equations for the mode of eqn. 7. For T frames and $D \doteq \dim(\boldsymbol{x}_t)$ dimensions, it can be LU-decomposed and solved in time $O(TD^3)$ [Gv96, §4.3.1].

This approach easily accommodates variations including: (1) replacing Gaussian neighborhood functions with mixtures-of-Gaussians; (2) adding priors on $\boldsymbol{x}_t$ or $\dot{\boldsymbol{x}}_t$ as in eqn. 7; (3) modeling accelerations as well as positions and velocities (making the left-hand side of eqn. 9 quintidiagonal); and (4) solving for $\boldsymbol{X}$ at different sampling rates than that of $\boldsymbol{Y}$ (where infinite sampling gives the mode associated with the path integral $\langle X \rangle_{p_\theta(X|\gamma)}$).

Figure 5 shows various ways of estimating trajectories from an HMM model of the manifold of motion in a figure-eight. Increasing the number of HMM states improves the quality of the synthesized trajectory, provided there is sufficient data to support estimates of the additional parameters. (Entropic estimation will automatically remove insufficiently supported parameters.)

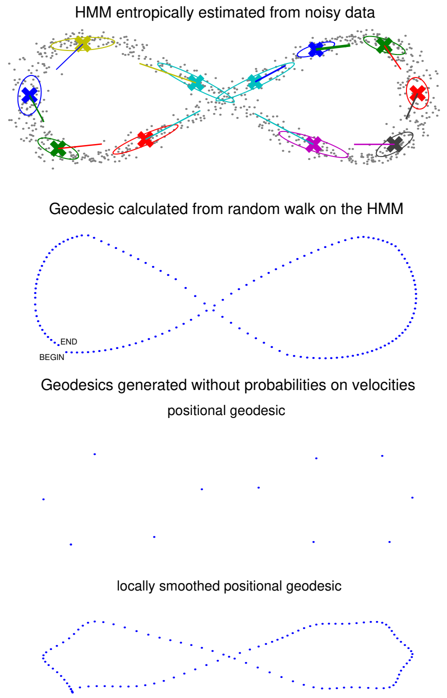

Figure 4: TOP: An entropically estimated HMM projected onto synthetic training data. An $\times$ indicates the mean output of a state; an ellipse indicates its covariance; and arcs indicate allowable transitions. See §3.0.1. SECOND: A trajectory generated using our method based on positional and velocity distributions. The state sequence is obtained from a random walk through the HMM. (Irregularities are due to variations between state dwells in the random walk.) See §5. THIRD: If one solves for a geodesic using just positional constraints, all control points clump on the means. BOTTOM: Traditionally, clumpiness is ameliorated by smoothing terms, but the trajectory is still unacceptable. (This could be improved if one is willing to hand-tune the objective function.)

6 Examples

6.1 Facial animation from voice

Our goal is to "face-sync" audio, generating full facial animation that is consistent with an audio track. Previously this has been attempted by learning an instantaneous or fixed-window mapping from vocal features to facial pose (e.g., [MRB97]), but the results are unsatisfactory, largely due to the many-to-many nature of the true mapping and the fact that co-articulation effects require propagation of context forwards *and* backwards over variable time-scales. Virtually all other methods use a speech recognition front-end to obtain phonemes (e.g., [Lew91, WL95, BCS97]), thereby losing much of the prosodic and phrasing information that predicts facial action. Some frame-reordering systems, notably [BCS97], use triphones as a first-order model of context, but since the video to be rewritten provides a small sampling of all possible triphones, it is necessary to take recourse to *ad hoc* smoothing methods. In synthesizing facial action from a learned manifold, we obtain optimal propagation of context, proper animation of non-phonemic vocalizations, and whole-face animation from neck to hairline.

To obtain training data we developed a visual facial tracker which tracks 25 separate landmarks on the face using per-landmark texture trackers organized into a virtual spring mesh. Texture tracking is based on minimizing a residual between an affine warped image window and a reference texture [HT97]. The spring mesh prevents slippage by preserving spatial relations between trackers. Spring lengths and tensions are estimated from the video. Spring energy and tracking residuals are combined into an objective function, which is relaxed repeatedly on each frame.

The resulting tracking data is synchonized with vectors of RASTA-PLP coefficients obtained from the audio track. Roughly three minutes of video were used as training data for the manifold. The resulting system can take arbitrary adult male speech and generate realistic-looking facial animation by deforming a texture-mapped head model. Although the effect is difficult to evoke on paper, we provide figure 6.1 as an illustration. Performance of vowels is excellent. Most consonants are correctly shaped by the mouth, but plosives, which happen 3 times faster than the training video's 33msec frame rate and therefore often have no evidence, have to be inferred from context and are sometimes missed. When used to resynthesize training data from audio alone, the predicted facial feature locations average less than two pixels dislocation from the ground-truth tracking (see figure 6.1). Surpisingly, the system predicts eyebrow and other upper-facial motion even better than lip and jaw motion, presumably using prosodic information in the audio signal. Perhaps this is because the upper face has a simple control program.

In addition analysis of the Viterbi paths showed that co-articulation information was being carried an average of 150 msec in either temporal direction, consistent with what is known about verbal co-articulation [(MI98] and the temporal offset between vocal and facial action [BK94]. In practice, we have seen this model carry context over 330 msec., indicating that the system has discovered facial co-articulation phenomena that last

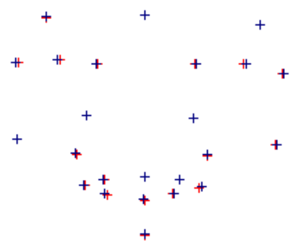

Figure 6: Voice-predicted facial feature locations superimposed over ground truth locations. See §5.

longer than vocal co-articulations, most probably slow relaxation of tissues in the cheeks.

The Viterbi sequence can be compressed, and is itself a highly compressed code for facial action. In the case of a "talking-head" video (e.g., a news anchor) we can compress facial action down to 4 bits per frame and reconstruct the video with less residual than afforded by 500 bits per frame encodings using the Facial Action Coding System [EF78], a psychologically motivated basis set for facial motion that has been promoted in the video compression literature [FK89] and is now standardized in MPEG-4 for animation.

6.2 3D from 2D

In this task we seek to infer 3D body pose and orientation from 2D projections. The input could be joint locations obtained from trackers, but we are interested in learning a more challenging mapping whose input might be reliably computed by current computer vision techniques: foreground motion "shadows" of the sort obtained from background subtraction.

Tracking-based pose recovery in 2D [BYJF97, LML98] and 3D [GBUP95, MR98, KM96, KM95, BM98, Roh97, GD96] are currently very active research topics in computer vision. Typically researchers assume a carefully constructed kinematic model, and must contend with problems of singularities, costly per-frame optimizations, or potentially troublesome approximations based on truncated Taylor series. A notably different approach (which inspired us to try this application) is the use of *observed* 3D motions as a linear basis set for interpreting 2D projections [LF98].

To demonstrate the viability of the manifold approximation approach we obtained five motion-tracking sequences from a public archive [Bio97], trained on four and used the model to reconstruct the fifth from random-viewpoint 2D renderings. Each sequence consists of 37 3D points, mainly joint positions, tracked over an average of 200 frames.

We took the four 3D training sequences, removed

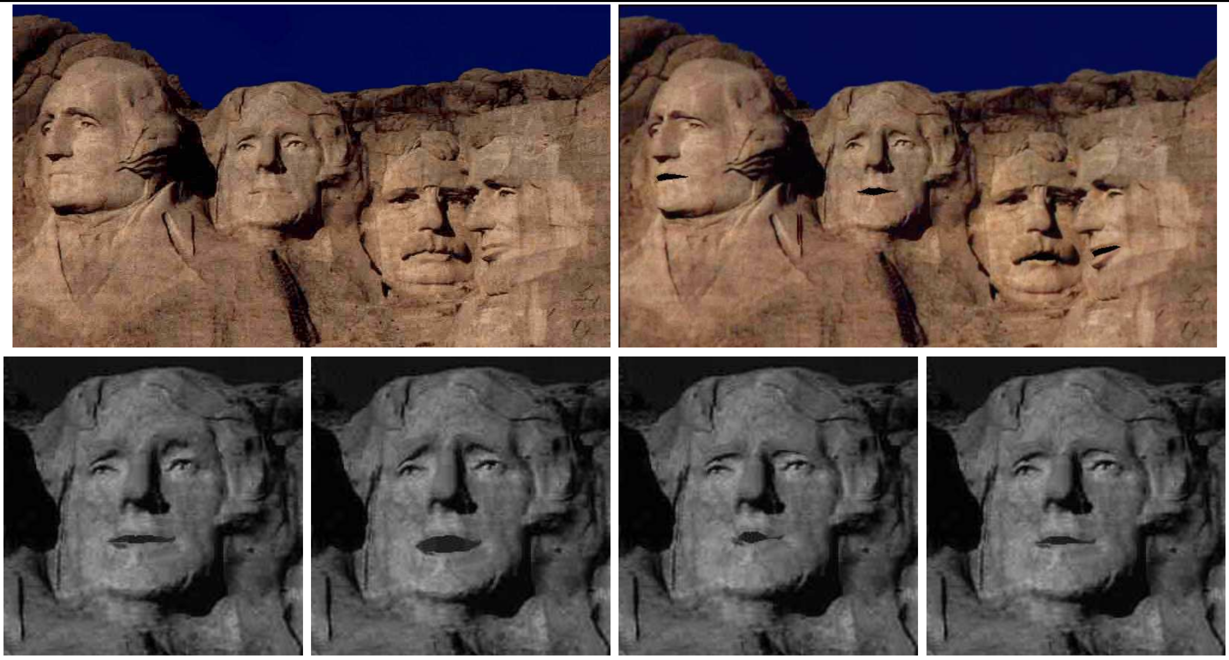

Figure 5: Mt. Rushmore at rest and reciting nursery rhymes.

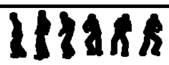

Figure 7: An example 2D projection.

variations due to whole-body translation, rotation, and scale changes, and estimated a manifold-approximating HMM. The HMM was initialized with states fully connected and random parameter values; after training roughly 90% of the transitions were removed and the internal Markov model had an entropy rate of 0.4941, yielding a perplexity (average branching factor) of 1.6391. In short, the HMM is a very unambiguous model of the dynamics.

We then replicated this HMM once for each view, re-estimating the output distributions of each view-specific HMM to cover an appropriately rotated version of the 3D data and its 2D "projection." In these experiments, the 2D "projection" is a descriptive vector calculated from the outline and moments of the silhouette of a rendered "tin-man" (figure 6.2). We then linked together all view-specific HMMs in the following manner: If $P_{i|j}$ is the probability of transitioning into state i from state j in the original HMM, and state i' is the i^{th} equivalent state in a duplicated HMM observing data rotated by ϕ, then $P_{i'|j} = P_{i|j}\mathcal{V}(\phi; 0, \sigma^2)$, where $\mathcal{V}$ is a circular von Mises density and the variance σ^2 is chosen small. The resulting HMM had 32 states $\times$ 32 views = 1024 states. Although this architecture works for monocular 3D from 2D, it trivially accommodates multiple cameras if their relative viewing angles are known: E.g., with cameras at 0, 15, and 30 degrees, each view-specific hidden state simply observes the product of its own Gaussian and those of equivalent states viewing 15 and 30 degrees beyond it.

Figure 6.2 shows the 3D content of the test sequence, in which a "thief" walks forward, stands tall, looks left and right, crouches down to grab something, then turns around and scurries back carrying the loot with both hands. This sequence highlights a constant source of ambiguity in 2D projections: One cannot tell whether the figure is facing toward or away from the camera. In order to correctly infer the direction of walking (remember, we remove all translation information), the system will have to propagate orientation information across the intervening 120 nonwalking frames.

To test the model, we rendered the test sequence from a several random viewpoints (in depicted examples, the camera was rotated 326 degrees about the vertical axis), computed descriptive image statistics, contaminated these statistics with 5% white noise, then performed a Viterbi analysis to obtain the most likely hidden state sequence. Figure 6.2 shows the resulting interpretation of the shadow sequence. The system correctly infers the camera rotation and changes in body orientation throughout the sequence. In addition, the graph shows how each activity (walking, carrying) is broken into rhythmic cycles of states.

Calculating the geodesic for this Viterbi sequence, we obtained a resynthesized 3D sequence that agrees well with the original (unseen) 3D data. In order to show that the very discrete state sequence yields

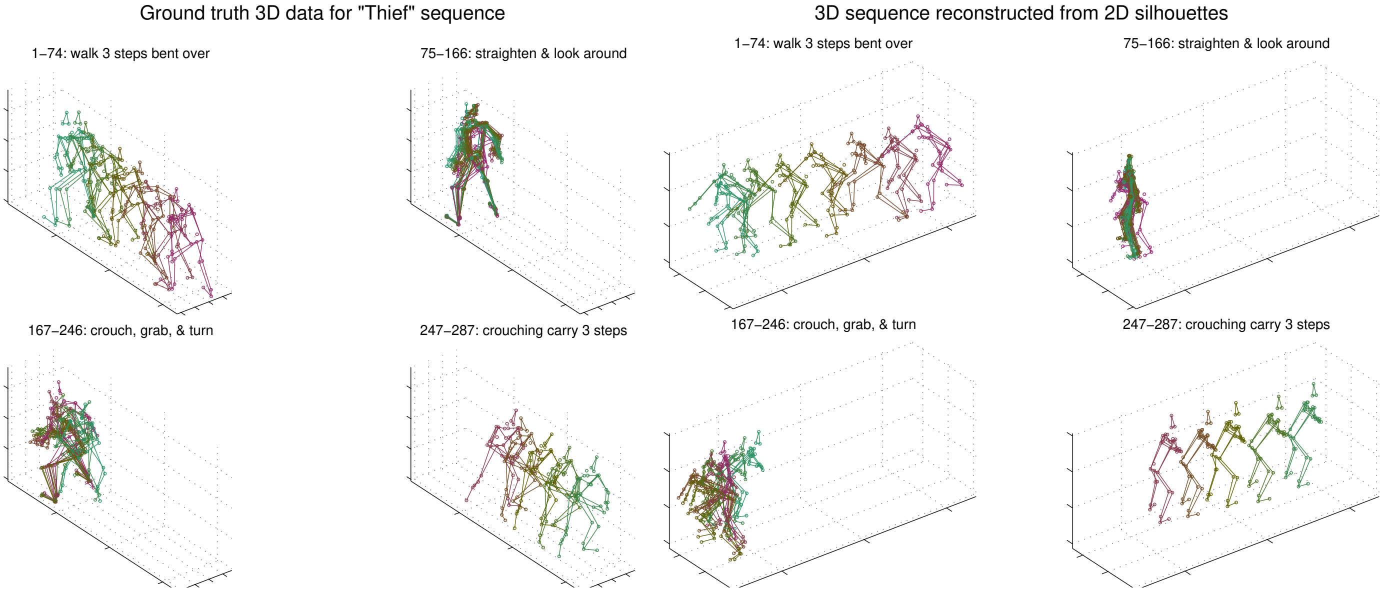

Figure 8: LEFT: A view of the test sequence. RIGHT: 3D poses reconstructed from a 2D sequence viewed at 326 degrees. The division into activities derives from the system's analysis in figure 6.2.

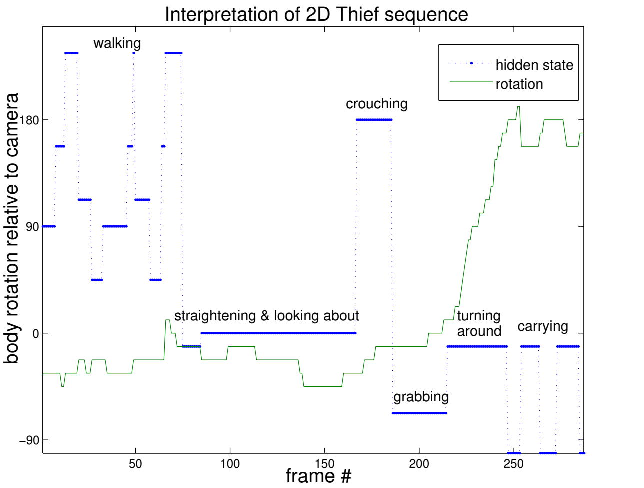

Figure 9: A hidden-state analysis of a 2D rendering of the test sequence. The hidden states have been factored into pose and orientation information.

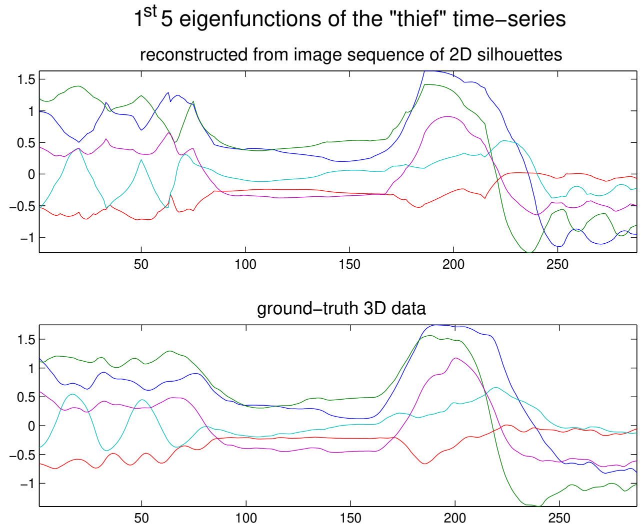

Figure 10: Comparative eigen-representations of the ground-truth 3D data and the reconstructed 3D data.

a smooth trajectory, we show the first 5 principal components of the synthesized time-series in figure 6.2. Figure 6.2RIGHT depicts the full 3D reconstruction. The principal source of error is that the shoulders now twist slightly with each step. This appears to be how walking is performed in the training set, which consists mainly of dance sequences. It may also be an artifact of the multiple-view HMM construction; small swings of the leg are compatible both a small change of pose and/or a small change of view angle. Increasing the number of states in the model improves the accuracy of the reconstruction, up to the limit of the maximum number of states supported by the training data (in this case, about 45).

Figure 6.2 shows the system's inference given image moments of low-resolution background-subtracted video.

There are several simple ways in which this example can be improved, in particular, modeling the manifold of joint angles rather than Cartesian joint positions. In addition, because the underlying model is an HMM, it can be used to make a running prediction of how the

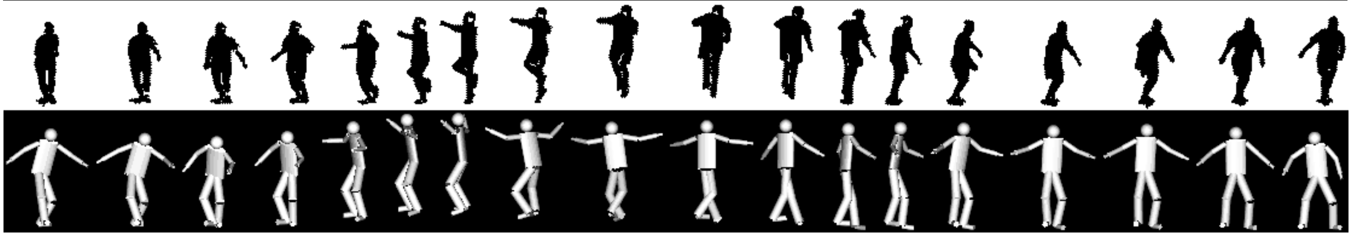

Figure 11: Frames from a jump-and-twirl-360° sequence and the corresponding inferred 3D structure. The inferred 3D is qualitatively correct but not pose-for-pose accurate: The system exploits learned dynamics to successfully resolve image ambiguities, but often substitutes nearby poses where the training set contains no plausible near-matches to the observed silhouette. At present we are not aware of any other method that can recover 3D from such low-resolution, high-occlusion, motion-blurred video.

body may move (and thus be tracked) 1-5 frames ahead of current visual processing.

Surprisingly, both of these applications are competitive with the state of the art even though the training is entirely unsupervised. We are now exploring supervised variants of the same framework, including explicit minimization of reconstruction error and automatic selection of maximally informative cue features.

7 Conclusion

We have shown how to reconstruct the behavior of a "target" system given a time-series of another co-evolving "cue" system, for example, inferring facial motion from vocal signals and 3D body pose and movement from 2D shadow sequences. These problems defy regression methods because the instantaneous mappings between the two systems are many-to-many. We learn a function between paths in the two systems, resolving ambiguities by integrating information over the whole length of a sequence. The key idea is that the target system's behavior can be viewed as a manifold containing configural and dynamical information, and this manifold can be approximated by a hidden Markov model, provided that the HMM has such low entropy that random walks on it produce time-series having the same long-term structure as the training time-series. This method stands on two mathematical results: (1) An entropy-minimization learning framework yields fast algorithms that produce extremely compact and accurate probabilistic models of the target system's behavior from examples. (2) A closed-form solution for geodesics yields optimal trajectories through the resulting density spaces. This allows us to do analysis, recognition, classification, and realistic synthesis of behaviors in linear time.

References

[Bau72] L. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of Markov processes. *Inequalities*, 3:1–8, 1972.

[BCS97] Christoph Bregler, Michele Covell, and Malcolm Slaney. Video Rewrite: Driving visual speech with audio. In *Proc. ACM SIGGRAPH 97*, 1997.

[Bio97] Biovision. 3D figural motion tracking sequences. http://www.biovision.com/samples/, 1997.

[BK94] Christoph Bregler and Yochai Konig. "eigenlips" for robust speech recognition. In *ICASSP94*, 1994.

[BM98] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *CVPR98*, pages 8–15, 1998.

[BO95] C. Bregler and S. Omohundro. Nonlinear manifold learning for visual speech recognition. In *ICCV95*, pages 494–499, 1995.

[Bra98] Matthew Brand. The entropic prior is the continuous and computable analogue of the universal prior. *(in reviews)*, 1998.

[Bra99a] Matthew Brand. Pattern discovery via entropy minimization. In David Heckerman and Christopher Whittaker, editors, *Artificial Intelligence and Statistics #7*. Morgan Kaufmann., January 1999.

[Bra99b] Matthew Brand. Structure discovery in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 11(5):1155–1182, July 1999.

[BYJF97] M.J. Black, Y. Yacoob, A.D. Jepson, and D.J. Fleet. Learning parameterized models of image motion. In *CVPR97*, pages 561–567, 1997.

[EF78] P. Ekman and W.V. Friesen. *Manual for the Facial Action Coding System.* Consulting Psychologists Press, Inc., Palo Alto, CA, 1978.

[FK89] R. Forchheimer and T. Kronander. Image coding - from waveforms to animation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(12):2008–2023, 1989.

[For73] G.D. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 6:268–278, 1973.

[GBUP95] L. Goncalves, E.D. Bernardo, E. Ursella, and P. Perona. Monocular tracking of the human arm in 3D. In *CVPR98*, pages 403–410, 1995.

[GD96] D.M. Gavrila and L.S. Davis. Towards 3-D model-based tracking of humans in action. In *AIU96*, pages 264–279, 1996.

[Gv96] Gene H. Golub and Charles F. van Loan. *Matrix Computations.* Johns Hopkins, 1996. 3rd edition.

[HT97] G. Hager and K. Toyama. The XVision system: A general-purpose substrate for portable real-time vision applications. *Computer Vision and Image Understanding*, 69(1), 1997.

[KM95] I. Kakadiaris and D. Metaxas. 3d human body model acquisition from multiple views. In *ICCV95*, pages 618–623, 1995.

[KM96] I.A. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. In *CVPR96*, pages 81–87, 1996.

[Lew91] J.P. Lewis. Automated lip-sync: Background and techniques. *The Journal of Visualization and Computer Animation*, 2:118–122, 1991. Available at www.idiom.com/˜zilla.

[LF98] Michael E. Leventon and William T. Freeman. Bayesian estimation of 3D human motion from an image sequence. Technical Report TR98-06, Mitsubishi Electric Research Laboratories, July 1998.

[LML98] Y. Li, S.D. Ma, and H. Lu. Human posture recognition using multi-scale morphological method and Kalman motion estimation. In *ICPR98*, pages PRP1:175–177, 1998.

[(MI98] Ken Stevens (MIT). Personal communication., 1998.

[MR98] D.D. Morris and J.M. Rehg. Singularity analysis for articulated object tracking. In *CVPR98*, pages 289–296, 1998.

[MRB97] D. F. McAllister, R. D. Rodman, and D. L. Bitzer. Speaker independence in lip synchronization. In *compugraphics '97*, December 1997.

[Pla74] Plato. *The Republic.* Penguin Classics, 1974.

[Rab89] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[Roh97] K. Rohr. Recognition of human movements based on explicit motion models. In *MBR97*, page Chapter 8, 1997.

[SJ96] L.K. Saul and M.I. Jordan. A variational principle for model-based interpolation. Technical report, MIT Center for Biological and Computational Learning, 1996.

[WL95] K. Waters and T. Levergood. DECface: A system for synthetic face applications. *Multimedia Tools and Applications*, 1:349–366, 1995.