Learning Appearance Manifolds from Video

Ali Rahimi
MIT CS and AI Lab,
Cambridge, MA 02139
ali@mit.edu

Ben Recht
MIT Media Lab,
Cambridge, MA 02139
brecht@media.mit.edu

Trevor Darrell
MIT CS and AI Lab,
Cambridge, MA 02139
trevor@mit.edu

Abstract

The appearance of dynamic scenes is often largely governed by a latent low-dimensional dynamic process. We show how to learn a mapping from video frames to this low-dimensional representation by exploiting the temporal coherence between frames and supervision from a user. This function maps the frames of the video to a low-dimensional sequence that evolves according to Markovian dynamics. This ensures that the recovered low-dimensional sequence represents a physically meaningful process. We relate our algorithm to manifold learning, semi-supervised learning, and system identification, and demonstrate it on the tasks of tracking 3D rigid objects, deformable bodies, and articulated bodies. We also show how to use the inverse of this mapping to manipulate video.

1. Introduction

The change in appearance of most scenes is governed by a low-dimensional time-varying physical process. The 3D motion of a camera through a scene in an egomotion problem, the contraction of the muscles in a face in expression analysis, or the motion of limbs in articulated-body tracking are examples of low-dimensional processes that almost completely determine the appearance of a scene. Recovering these processes is a fundamental problem in many areas of computer vision.

Recently, manifold learning algorithms have been used to automatically recover low-dimensional representations of collections of images [1, 3, 4, 11, 14]. But when these collections are video sequences, these algorithms ignore the temporal coherence between frames, even though this cue provides useful information about the neighborhood structure and the local geometry of the manifold. Semi-supervised regression that take advantage of the manifold structure of the data set provide another framework for addressing this problem [2, 17]. These algorithms learn a mapping between high-dimensional observations and low-dimensional representations given a few examples of the mapping. But they do not take advantage of the temporal coherence between video frames either. One could use nonlinear system identification [6, 15] to model the dynamics of low-dimensional states to simultaneously estimate these states while learning a lifting from them to the observed images. But current nonlinear system identification methods do not scale to image-sized observations and get stuck in local minima.

Our main contribution is a synthesis of a semi-supervised regression model with a model for nonlinear system identification. The result is a semi-supervised regression model that takes advantage of the dynamics model used in system identification to learn an appearance manifold. The algorithm finds a smooth mapping represented with radial basis functions that maps images to a low-dimensional process consistent with physical dynamics defined by a linear-Gaussian Markov chain. The algorithm allows a user to label a few data points to specify a coordinate system and to provide guidance to the algorithm when needed.

We demonstrate our algorithm with an interactive tracking system where the user specifies a desired output for a few key frames in a video sequence. These examples, together with the unlabeled portion of the video sequence, allow the system to compute a function that maps as-yet unseen images to the desired representation. This function is represented using radial basis kernels centered on the frames of the video sequence. We demonstrate our algorithm on three different examples: 1) a rigid pose estimation problem where the user specifies the pose of a synthetic object for a few key frames, 2) on a lip tracking example where the user specifies the shape of the subject's lips, and 3) an articulated body tracking experiment where the user specifies positions of the subject's limbs. The algorithm operates on the video frames directly and does not require any preprocessing. Semi-supervision allows the user to specify additional examples to improve the performance of the system where needed.

By inverting the learned mapping, we can also generate novel frames and video sequences. We demonstrate this

by manipulating low-dimensional representations to synthesize videos of lips and articulated limbs.

2. Related Work

Manifold learning techniques [1, 3, 4, 11, 14, 16] find a low-dimensional representation that preserves some local geometric attribute of the high-dimensional observations. This requires identifying data points that lie in a local neighborhood along the manifold around every high-dimensional data point. When the manifold is sparsely sampled, these neighboring points are difficult to identify, and the algorithms can fail to recover any meaningful structure. Our algorithm obviates the need to search for such neighbors by utilizing the time ordering of data points instead. Jenkins and Mataric [8] suggest artificially reducing the distance between temporally adjacent points to provide an additional hint to Isomap about the local neighborhoods of image windows. We also take advantage of dynamics in the low-dimensional space to allow our algorithm to better estimate the distance between pairs of temporally adjacent points along the manifold. This requires only fine enough sampling over time to retain the temporal coherence between video frames, which is much less onerous than the sampling rate required to correctly estimate neighborhood relationships in traditional manifold learning algorithms. While various semi-supervised extensions to manifold learning algorithms have been proposed [7, 10], these algorithms still do not take advantage of the temporal coherence between adjacent samples of the input time series.

The semi-supervised regression approaches of [17] and [2] take into account the manifold structure of the data. But they also rely on brittle estimates of the neighborhood structure, and do not take advantage of the time ordering of the data set. These semi-supervised regression methods are similar to our method in that they also impose a random field on the low-dimensional representation. The work presented here augments these techniques by introducing the temporal dependency between output samples in the random field. It can be viewed as a special case of estimating the parameters of a continuously-valued conditional random field [9] or a manifold learning algorithm based on function estimation [13].

Nonlinear system identification (see [6, 15] and references within) provides another framework for introducing dynamics into manifold learning. In this context, the frames in the video are modeled as observations generated by a Markov chain of low-dimensional states. Nonlinear system identification recovers the parameters of this model, including an observation function which maps low-dimensional states to images. This usually requires approximate coordinate ascent over a non-convex space, making the algorithms computationally intensive and susceptible to local minima.

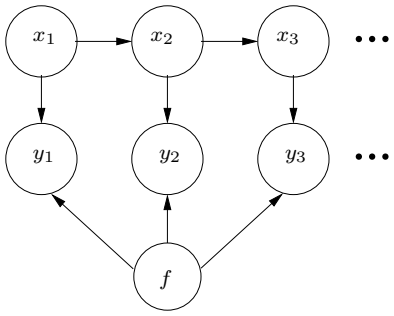

Figure 1. A generative model for video sequences. The states x_t are low-dimensional representations of the scene. The embedding f lifts these to high-dimensional images y_t.

Dynamic Textures [5] sidesteps these issues by performing linear system identification instead, which limits it to linear appearance manifolds. Instead of searching for a mapping from states to images, as would be done in nonlinear system identification, we search for a mapping from images to states. This results in an optimization problem that is quadratic in the latent states and the parameters of the projection function, making the problem computationally tractable and not subject to local minima.

3. Model for Semi-Supervised Nonlinear System ID

Figure 1 depicts a plausible generative model for video. The latent state of the scene evolves according to a Markov chain of states x_t, $t = 1 \dots T$. At each time step, a nonlinear function $f : \mathcal{R}^d \rightarrow \mathcal{R}^D$ maps a d-dimensional subset of the state x_t to an image with D pixels represented as a D-dimensional vector y_t. The Markov chain captures the notion that the underlying process that generates the video sequence is smooth. Effects not accounted for by f are modeled as iid noise modifying the output of f.

Learning the parameters of this generative model from a sequence of observations $y_1, \dots, y_T$ can be computationally expensive [6, 15]. Instead of solving for f in this generative model, we recover a projection function $g : \mathcal{R}^D \rightarrow \mathcal{R}^d$ that maps images to their low-dimensional representation in a random field. This random field consists of a function g that maps the sequence of observed images to a sequence in $\mathcal{R}^d$ that evolves in accordance with a Markov chain. The random field mirrors the generative model of Figure 1 by modeling the interactions between a Markov chain, the observations, and supervised points provided by the user. We address each interaction in turn, gradually building up a cost functional for g.

Consider each component g^i of $g = [g^1(y) \dots g^d(y)]$ separately. If the desired output of g^i at time steps $t \in \mathcal{S}$ were known to be z_t^i, we could use Tikhonov regularization

on a Reproducing Kernel Hilbert Space (RKHS) to solve for the best approximation of g^i:

$$\min_{g^i} \sum_{t \in \mathcal{S}} \|g^i(y_t) - z_t^i\|^2 + \lambda_k \|g^i\|_k^2. \quad (1)$$

The first term in this cost functional penalizes the deviation from the desired outputs, and the norm in the cost function governs the smoothness of g^i. In particular, according to the Representer Theorem [12], when the norm is an RKHS norm induced by a radial basis kernel, such as $k(y', y) = \exp(-\|y - y'\|^2/\sigma_k^2)$, any cost functional of the form $\sum_{t \in \mathcal{S}} V(g^i(y_t)) + \|g^i\|_k^2$ will be minimized by a weighted sum of kernels centered at each y_t:

$$g^i(y) = \sum_{t \in \mathcal{S}} c_t^i k(y, y_t)\,, \quad (2)$$

where the vector c^i contains the coefficients for the ith dimension of g.

But in practice, only a few z_t^is are provided by the user. Because we know the low-dimensional process is smooth, we assume the missing z_t^is evolve according to second-order Newtonian dynamics:

$$x_{t+1} = \mathbf{A}x_t + \omega_t, \quad (3)$$

$$\mathbf{A} = \begin{bmatrix} 1 & A_v & 0 \\ 0 & 1 & A_a \\ 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

$$z_t^i = h' x_t\,. \quad (5)$$

The Gaussian random variable ω_t has zero-mean and a diagonal covariance matrix Λ_ω. The matrices $\mathbf{A}$ and Λ_ω specify the desired dynamics, and are parameters of our algorithm. The components of x_t have intuitive physical analogs: the first component corresponds to a position, the second to velocity, and the third to acceleration. The vector $h = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}'$ extracts the position component of x_t.

We can compensate for the absence of z_t^i at every data point by forcing $g^i(y_t)$ to agree with the position component of the corresponding x_t using additional penalty terms:

$$\begin{aligned} \min_{g^i, x} \quad & \textstyle\sum_{t=1}^{T} \quad \|g^i(y_t) - h' x_t\|^2 \quad (6) \\ +\lambda_d \quad & \textstyle\sum_{t=2}^{T} \quad \|x_t - \mathbf{A}x_{t-1}\|_{\Lambda_w}^2 \\ +\lambda_s \quad & \textstyle\sum_{t \in \mathcal{S}} \quad \|g^i(y_t) - z_t^i\|^2 + \lambda_k \|g^i\|_k^2. \end{aligned}$$

The first term favors functions whose outputs evolve according to the trajectory x. The term weighted by λ_d favors trajectories that are compatible with the given dynamics model.

Figure 2 depicts a random field describing the factorization prescribed by (6). This random field mirrors the generative model of Figure 1. Note that according to the Representer Theorem, the the optimal g^i has the form $\sum_{t=1}^{T} c_t^i k(y, y_t)$, where the kernels are now placed on all data points, not just those supervised by the user.

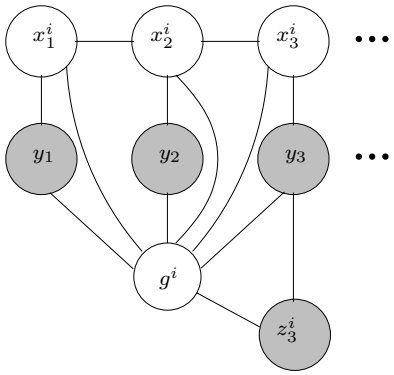

Figure 2. Forcing agreement between projections of images y_t and a Markov chain of states x_t. z_3 is a semi-supervised point provided by the user. The function g maps observations to states.

4. Learning the Projection Function

The optimization (6) is quadratic in the quantities of interest. Substituting the representer form results in a finite-dimensional quadratic optimization problem:

$$\begin{aligned} \arg\min_{c^i, x} \|\mathbf{K}c^i - \mathbf{H}x\|^2 + \lambda_d x' \Omega_x x \quad (7) \\ +\lambda_s \|\mathbf{G}\mathbf{K}c^i - z^i\|^2 + \lambda_k {c^i}' \mathbf{K} c^i. \end{aligned}$$

The matrix $\mathbf{K}$ has $k(y_t, y_\tau)$ in entry t, τ. The path x is a $3T$-dimensional vector of states stacked on top of each other, and $\mathbf{H} = \mathbf{I}_T \otimes h'$ extracts the position components of x. The matrix Ω_x is the inverse covariance of the Markov process and is block tri-diagonal. The matrix $\mathbf{G}$ extracts the rows of $\mathbf{K}$ corresponding to the supervised frames $t \in \mathcal{S}$, and z^i is a column vector consisting of the ith component of all the semi-supervised points.

The minimizer can be found by setting derivatives to zero and solving for c^i. After an application of the matrix inversion lemma, we find

$$c^{i*} = \lambda_s \mathbf{S}^{-1} \mathbf{G}' z^i \quad (8)$$

$$\mathbf{S} = \mathbf{K} + \lambda_s \mathbf{G}'\mathbf{G}\mathbf{K} - \mathbf{H}\mathbf{R}^{-1}\mathbf{H}'\mathbf{K} + \lambda_k \mathbf{I} \quad (9)$$

$$\mathbf{R} = \lambda_d \Omega_x + \mathbf{H}'\mathbf{H} \quad (10)$$

Having recovered the coefficients of the radial basis functions, g can be applied to an as-yet unseen image y_{new} by computing the vector K_{new} whose tth component is $k(y_{new}, y_t)$. Then, according to Equation (2), $g^i(y_{new}) = K_{new} c^i$. Unlabeled frames of the video can be labeled by using the tth row of $\mathbf{K}$, K_t, to get $g^i(y_t) = K_t c^i$.

5. Experiments

To compare with Isomap, LLE and Laplacian Eigenmaps, we relied on source code available from the respective authors' web sites. We also compare against Belkin and

Nyogi's graph Laplacian-based semi-supervised regression algorithm [2], which we refer to as BNR in this section. We used our own implementation of BNR.

5.1. Synthetic Results

We first demonstrate our algorithm on a synthetic 2D manifold embedded in $\mathcal{R}^3$. The neighborhood structure of this manifold is difficult to estimate from high-dimensional data, so traditional manifold learning techniques perform poorly on this data set. Taking into account the temporal coherence between data points and using user supervision alleviates these problems.

Figure 3(top-middle) shows an embedding of the 2D Markov process shown in Figure 3(top-left) into $\mathcal{R}^3$. The semi-supervised points are marked with a large triangle. Figure 3(top-right) shows our interpolated results for the unlabeled points. The interpolated values are close to the true values that generated the data set. Although the process is smooth, it clearly does not follow the dynamics assumed by Equation (3) because it bounces off the boundaries of the rectangle $[0, 5] \times [-3, 3]$. Nevertheless, the assumed dynamics of Equation (3) are sufficient for recovering the true location of unlabelled points.

To assess the quality of the learned function g on as-yet unseen points, we evenly sampled the 2D rectangle $[0, 5] \times [-3, 3]$ and lifted the samples to $\mathcal{R}^3$ using the same mapping used to generate the training sequence. See Figure 3(bottom-left and bottom-right). Each sample in $\mathcal{R}^3$ is passed through g to obtain the 2D representation shown in Figure 3(bottom-right). The projections fall close to the true 2D location of these samples.

We applied LLE, Laplacian Eigenmaps, and Isomap to the data set of Figure 3(top-middle). Isomap produced the result shown in Figure 4(left). It is difficult to estimate the neighborhood structure near the neck, where the manifold comes close to intersecting itself, so Isomap creates folds in the projection.

Figure 4(right) shows the result of BNR. Compared to our result in Figure 3(top-right), the interpolated results are incorrect for most points. Since BNR does not attempt to enforce any geometric invariance in the projection, it is fairly robust to the neighborhood estimation problem.

For this and subsequent data sets, neither LLE nor Laplacian Eigenmaps produced sensible results. This may be due to the low rate at which the manifold is sampled.

5.2. Synthetic Images

We quantitatively gauged the performance of our algorithm on images by running on a synthetic image sequence. Figure 5 shows frames in a synthetically generated sequence of 50×50 pixel images of a rigidly rotating object. Six images were chosen for semi-supervision by providing their true elevation and azimuth to the algorithm.

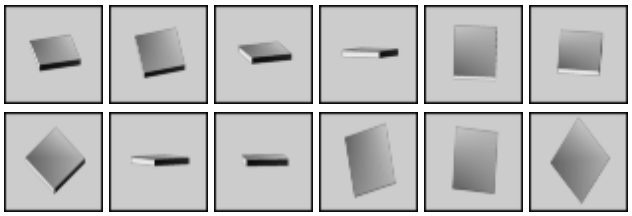

Figure 5. (top) A few frames of a synthetically generated 1500 frame sequence of a rotating cube. (bottom) The 6 semi-supervised frames. The rotation for each frame in the sequence was recovered with an average deviation of 4° from ground truth.

The azimuth and elevation of the filled-in frames deviated from ground truth by only an average of 3.5°. We evaluated BNR on the same data set, with the same semi-supervised points, and obtained average errors of 17° in elevation and 7° in azimuth. To test the learned function g, we generated a video sequence that swept through the range of azimuths and elevations in 4° increments. These images were passed through g to estimate their azimuth and elevation. The mean squared error was about 4° in each direction.

5.3. Interactive Tracking

Our algorithm is not limited to rigid body tracking. We applied it to a lip tracking experiment exhibiting deformable motion, and to an upper-body tracking experiment exhibiting articulated motion. In these experiments, we restricted ourselves to recovering the missing labels of the training data and labeling frames acquired under the same setting from which the training data was gathered. Our algorithm operates on the entire frames, as shown in the figures. Images were not in any way preprocessed before applying our algorithm, though to apply the learned mapping to different settings, more taylored representations or kernels could be employed. We tuned the parameters of our algorithm (A_v, A_a, the diagonal entries of Λ_ω, and the weights λ_d, λ_s, and λ_k) by minimizing the leave-one-out cross validation error on the semi-supervised points using the simplex method.

Figure 6 shows frames in a 2000 frame sequence of a subject articulating his lips. The top row shows the frames that were manually annotated with a bounding box around the lips. The bottom row shows the bounding boxes returned by g on some typical frames in the sequence. Only five labeled frames were necessary to obtain good lip tracking performance. The tracker is robust to natural changes in lighting, blinking, facial expressions, small movements of the head, and the appearance and disappearance of teeth.

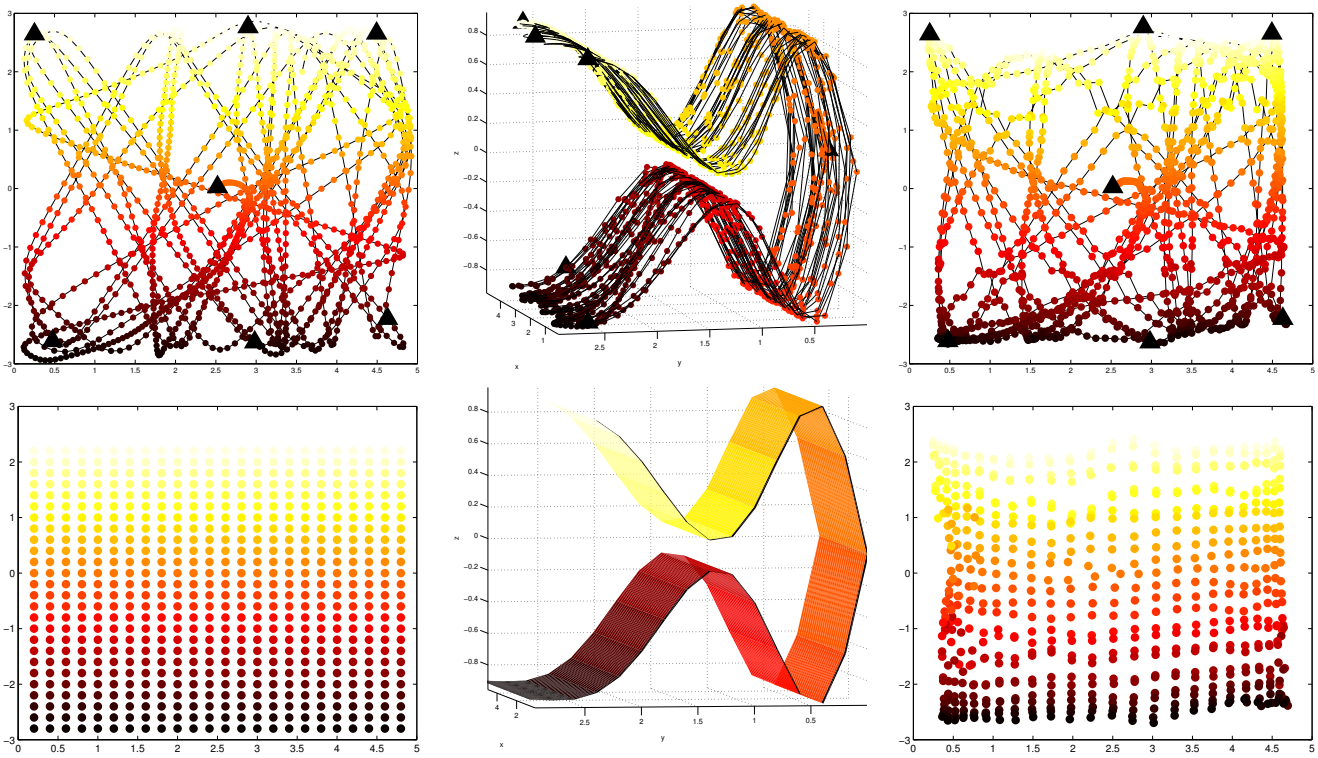

Figure 3. (top-left) The true 2D parameter trajectory. Semi-supervised points are marked with big black triangles. The trajectory is sampled at 1500 points (small markers). Points are colored according to their y-coordinate on the manifold. (top-middle) Embedding of a path via the lifting $F(x, y) = (x, |y|, \sin(\pi y)(y^2 + 1)^{-2} + 0.3y)$. (top-right) Recovered low-dimensional representation using our algorithm. The original data in (top-left) is correctly recovered. (bottom-left) Even sampling of the rectangle $[0, 5] \times [-3, 3]$. (bottom-middle) Lifting of this rectangle via F. (bottom-right) Projection of (bottom-middle) via the learned function g. g has correctly learned the mapping from 3D to 2D. These figures are best viewed in color.

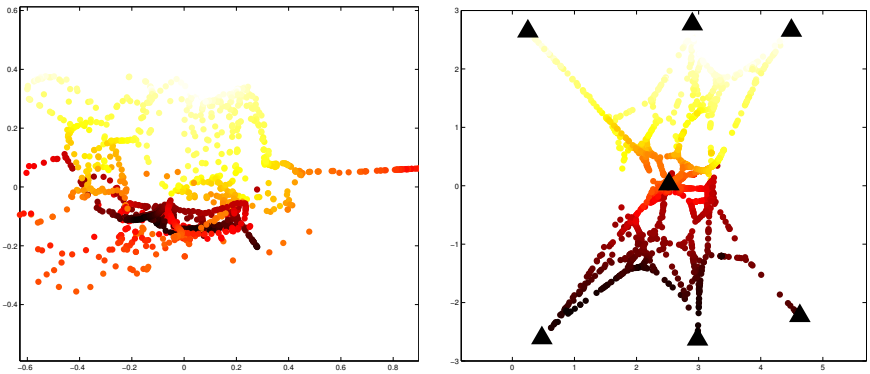

Figure 4. (left) Isomap's projection into $\mathcal{R}^2$ of the data set of Figure 3(top-middle). Errors in estimating the neighborhood relations at the neck of the manifold cause the projection to fold over itself. (right) Projection with BNR, a semi-supervised regression algorithm. There is no folding, but the projections are not close to the ground truth shown in Figure 3(top-left).

Figure 8 shows 12 labeled images in a 2300 frame sequence of a subject moving his arms. These frames were manually labeled with line segments denoting the upper and lower arms. Figure 9 shows the recovered limb positions for unlabeled samples, some of which were not in the training sequence. Because the raw pixel representation is used, there are very few visual ambiguities between appearance and pose, and occlusions due to crossing arms do not present a problem.

The utility of dynamics is most apparent in articulated tracking. Setting λ_d to zero makes our algorithm ignore dynamics, forcing it to regress on the semi-supervised examples only. The resulting function produced the limb locations shown in black in Figure 9. Using dynamics allows the system to take advantage of the unsupervised points, producing better estimates of limb position.

5.4. Resynthesizing Video

When g is one-to-one, it can be inverted. This inverse function maps the intrinsic representation to images, allowing us to easily create new video sequences by controlling the intrinsic representation. We have explored two different approaches for computing pseudo-inverses of g that do not require g to be exactly one-to-one.

In Figure 7, where we animate the mouth by manipulating its bounding box, the inverse simply returns the training image whose estimated parameter is closest to the desired intrinsic parameter. In Figure 10, where we manipulate limb locations to generate new images, we computed the inverse by fitting a function using Tikhonov regularization to a data set consisting of the training images and their estimated labels. This representation can automatically interpolate between images, allowing us to generate images that do not appear in the training sequence.

6. Conclusion

We have presented a semi-supervised regression algorithm for learning the appearance manifold of a scene from a video sequence. By taking advantage of the dynamics in video sequences, our algorithm learns a function that projects images to a low-dimensional space with semantically meaningful coordinate axes. The pseudo-inverse of this mapping can also be used to generate images from these low-dimensional representations.

We demonstrated our algorithm on lip tracking and articulated body tracking, two domains where the appearance manifold is nonlinear. With very few labeled frames and no preprocessing of the images, we were able to recover poses for the frames in the training sequences as well as outside the training sequences.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, 2002.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[3] M. Brand. Charting a manifold. In *Neural Information Processing Systems (NIPS)*, 2002.

[4] D.L. Donoho and C. Grimes. Hessian eigenmaps: new locally linear embedding techniques for highdimensional data. Technical report, TR2003-08, Dept. of Statistics, Stanford University, 2003.

[5] G. Doretto, A. Chiuso, and Y.N. Wu S. Soatto. Dynamic textures. *International Journal of Computer Vision (IJCV)*, 51(2):91–109, 2003.

[6] Z. Ghahramani and S. Roweis. Learning nonlinear dynamical systems using an em algorithm. In *Neural Information Processing Systems (NIPS)*, pages 431–437, 1998.

[7] J.H. Ham, D.D. Lee, and L.K. Saul. Learning high dimensional correspondences from low dimensional manifolds. In *ICML*, 2003.

[8] O. Jenkins and M. Mataric. A spatio-temporal extension to isomap nonlinear dimension reduction. In *International Conference on Machine Learning (ICML)*, 2004.

[9] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[10] R. Pless and I. Simon. Using thousands of images of an object. In *CVPRIP*, 2002.

[11] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[12] B. Schölkopf, R. Herbrich, A.J. Smola, and R.C. Williamson. A generalized representer theorem. Technical Report 81, NeuroCOLT, 2000.

[13] A. Smola, S. Mika, B. Schoelkopf, and R. C. Williamson. Regularized principal manifolds. *Journal of Machine Learning*, 1:179–209, 2001.

[14] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[15] H. Valpola and J. Karhunen. An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural Computation*, 14(11):2647–2692, 2002.

[16] K.Q. Weinberger and L.K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *Computer Vision and Pattern Recognition (CVPR)*, 2004.

[17] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

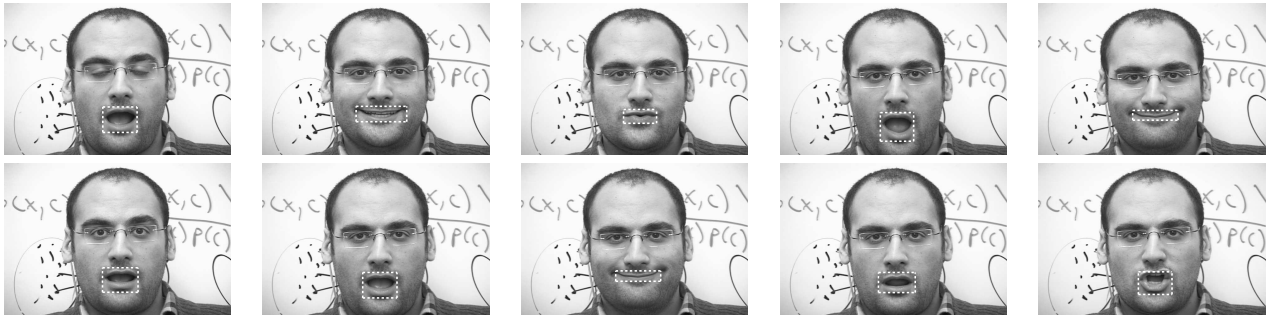

Figure 6. The bounding box of the mouth was annotated for 5 frames of a 2000 frame video. The labeled points (shown in the top row) and first 1500 frames were used to train our algorithm. The images were not altered in any way before computing the kernel. The parameters of the model were fit using leave-one-out cross validation on the labeled data points. Plotted in the second row are the recovered bounding boxes of the mouth for various frames. The first three examples correspond to unlabeled points in the training set. The tracker is robust to natural changes in lighting, blinking, facial expressions, small movements of the head, and the appearance and disappearance of teeth.

Figure 7. Resynthesized trajectories using radial basis interpolation. The two rows show a uniform walk along two of the coordinate axes of the low-dimensional space. The appearance and disappearance of the tongue is a nonlinearity that is well captured with our model.

Figure 8. The twelve supervised points in the training set for articulated hand tracking (see Figure 9).

Figure 9. The hand and elbow positions were annotated for 12 frames of a 2300 frame video. The labeled points (shown in Figure 8) and the first 1500 frames were used to train our algorithm. The images were not preprocessed in any way. Plotted in white are the recovered positions of the hands and elbows. Plotted in black are the recovered positions when the algorithm is trained without taking advantage of dynamics. Using dynamics improves tracking significantly. The first two rows correspond to unlabeled points in the training set. The last row correspond to frames in the last 800 frames of the video, which was held out during training.

Figure 10. Resynthesized trajectories using nearest neighbors. Top row: The left hand moving straight up while keeping the right hand fixed. Middle row: The same trajectory with the hands switched. Bottom row: Both arms moving in opposite directions at the same time.