Modeling Correspondences for Multi-Camera Tracking Using Nonlinear Manifold Learning and Target Dynamics [*]

Vlad I. Morariu
Computer Vision Laboratory
University of Maryland
College Park, MD 20742
morariu@umd.edu

Octavia I. Camps
Department of Electrical Engineering
Pennsylvania State University
University Park, PA 16802
camps@whale.ee.psu.edu

Abstract

Multi-camera tracking systems often must maintain consistent identity labels of the targets across views to recover 3D trajectories and fully take advantage of the additional information available from the multiple sensors. Previous approaches to the "correspondence across views" problem include matching features, using camera calibration information, and computing homographies between views under the assumption that the world is planar. However, it can be difficult to match features across significantly different views. Furthermore, calibration information is not always available and planar world hypothesis can be too restrictive. In this paper, a new approach is presented for matching correspondences based on the use of nonlinear manifold learning and system dynamics identification. The proposed approach does not require similar views, calibration nor geometric assumptions of the 3D environment, and is robust to noise and occlusion. Experimental results demonstrate the use of this approach to generate and predict views in cases where identity labels become ambiguous.

1. Introduction

Multi-camera tracking systems often must maintain consistent identity labels of the targets across views to recover 3D trajectories and fully take advantage of the additional information available from the multiple sensors. Previous approaches to the "correspondence across views" problem include matching features such as color and apparent height [10, 13, 36, 15], using 3D information from camera calibration [13, 4, 1, 14, 16] or computing homographies between views [32, 33, 12]. More recently, Khan and Shah [27] presented an approach based on finding the limits of the field of view of each camera as visible by the other cameras under the assumption that the world is planar. However, it can be difficult to find matching features across significantly different views, camera calibration information is not always available and planar world hypothesis can be too restrictive.

In this paper, we propose a new approach to the problem of finding correspondences across frames that does not require feature matching, camera calibration or planar assumptions. Instead, the proposed approach exploits the high spatial and temporal correlations between frames and across sequences to find a set of intrinsic coordinates on which finding correspondences becomes an easy problem. In particular, we propose to use nonlinear dimensionality reduction methods to map the high dimensional images into low dimensional manifolds that preserve neighborhood properties of the original data. Additional robustness to noise and occlusion is incorporated by capturing the temporal evolution of the manifolds with system dynamics identification techniques. Two alternative methods to find correspondences between sequences using these manifolds are presented. In the first method, manifolds from different sequences are aligned so corresponding views have the same intrinsic coordinates in the low dimensional space. In the second method, the points on the manifold of one view are modeled as the output of a linear time invariant (LTI) system excited with the manifold corresponding to the other view as an input. The main contribution of this paper is that it provides a method for (1) learning appearance correspondences between different frames without requiring one-to-one correspondence, (2) learning temporal as well as spatial dynamics between views, and (3) generating new unseen views either by using another existing view as input or by predicting all views simultaneously based on temporal dynamics.

The paper is organized as follows. In section 2 previous related work is discussed. In section 3 we briefly discuss the preprocessing steps needed before learning correspon-

[*] This work was supported in part by AFOSR under grant FA9550-05-1-0437 and NSF under grants IIS-0117387, ECS-0221562 and ITR-0312558.

dences. In sections 4 and 5 the algorithms for nonlinear manifold learning and system dynamics identification used to capture spatial and temporal correlations are respectively summarized. The proposed approach to learn view correspondences and generate views is described in section 6 and 7, respectively, and is illustrated with experiments in section 8. Finally, conclusions and future work are discussed in section 9.

2. Relation to Previous Work

Correlation of image sets has been extensively used in image compression, object recognition and tracking [40, 34, 5, 30, 31]. In these applications, images are viewed as high dimensional vectors that can be represented as points in lower dimensional subspaces without much loss of information. Principal component analysis (PCA) is the tool most often used to extract the linear subspaces in which the data has the highest variance. More recently, low-dimensional linear subspace models have been proposed to predict an image sequence from a related image sequence [6, 29] and to model dynamic texture [18].

However, image data does not usually lie in a linear subspace but instead on a low dimensional nonlinear manifold within the higher dimensional space [7, 8, 9, 19, 20, 21, 23, 22, 37, 41, 43, 42]. As a result, images that are far apart can have similar representations when they are projected onto a linear subspace using a PCA decomposition.

Thus, in this paper we propose to use a nonlinear dimensionality reduction technique to obtain low dimensional mappings that preserve the spatial and temporal neighborhoods of the data. There are various techniques that can be used for this purpose. Methods such as LLE [37], Isomap [39], Laplacian Eigenmaps [3], Hessian LLE [17], and Semidefinite Embedding [43, 42] seek to find an embedding of the data which preserves some relationship between the datasets, without providing an explicit mapping function. Other methods have been proposed that do provide a mapping for the embeddings, such as nonlinear Canonical Correlation Analysis (CCA) [41], Charting [7], Local Tangent Space Analysis [44] and Geodesic Nullspace Analysis (GNA) [9]. LLE has been used for gait and activity recognition [19, 20, 21] and ways have been proposed to use prior information to align two manifolds [23, 22]. Also, new samples can be approximately mapped into the embedding space using the training dataset despite the lack of a mapping function [37].

3. Preprocessing

To model correspondences between person appearance in multiple views, the objects first need to be extracted and normalized so that they can be compared in a meaningful way. First, we use foreground segmentation methods such as background subtraction and morphological operations to smooth the resulting binary images. After thresholding for size, only the blobs corresponding to persons remain in the image. These are then resized to a standard size for each frame. Figure 1 illustrates one example of preprocessing multiple views of a scene containing two persons. The appearance templates are then transformed into column vectors that are then used for manifold learning and system identification steps.

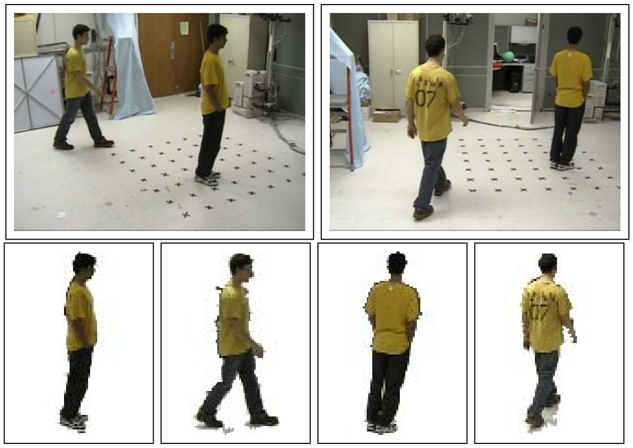

Figure 1. Example of tracking in two views. Row 1: The input images. Row 2: Normalized person appearance.

4. Nonlinear Manifold Learning

Ideally, we would like to use a nonlinear manifold learning technique that gives both the mapping and the embedding of our training set. However, such luxury comes at extra computational cost and algorithm complexity. Thus, we use the locally linear embedding (LLE) algorithm to find the embedding of the data [37]. Though LLE does not directly provide a mapping from the high dimensional image space to the embedding space, methods similar to those described in [37] can approximate the mapping.

Given a set of images $X = [x_1 \dots x_n] \in \Re^{D \times n}$, where x_i is the view of an object at time i, we want to find an embedding $Y = [y_1 \dots y_n] \in \Re^{d \times n}$ such that $d \ll D$. The LLE algorithm finds an embedding where data point relationships in the high dimensional space are preserved in the embedding.

To learn a locally linear embedding of X, we seek to represent each sample x_i as a linear combination of k neighbors. We define $i \sim j$ to be true if i is a neighbor of j. Thus, we want to find the weights W_{ij} so that for each sample x_i

$$W = argmin_W \sum_i |x_i - \sum_j W_{ij} x_j|^2 \tag{1}$$

so that $\sum_j W_{ij} = 1$ and $W_{ij} = 0$ if x_i and x_j are not neighbors. Using these weights we then find the embedding Y so that

$$Y = argmin_Y \sum_i |y_i - \sum_j W_{ij} y_j|^2 \tag{2}$$

Letting

$$L = (I - W)^T (I - W), \tag{3}$$

the solution is found by calculating the eigenvalues and eigenvectors of L. Because it can be shown that the smallest eigenvalue is zero, the embedding coordinates are given by $Y = [v_2 \dots v_{d+1}]^T$, where v_i is the eigenvector corresponding to the i^{th} smallest eigenvalue of L.

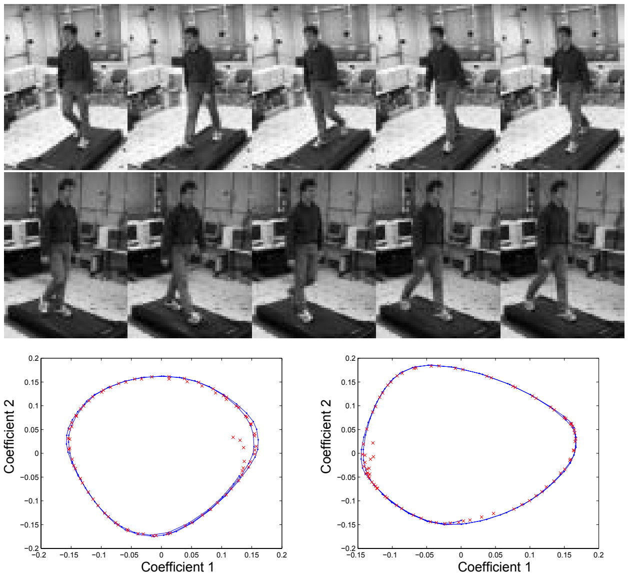

Figure 2. Top: Sample images. Bottom: Embeddings of two sequences found by LLE. Blue and red points are training and test image embedding coordinates, respectively.

To map a new vector x_{new} into the embedding, we use the method described in [37]. We find the k nearest neighbors of x_{new} in the training set X, and compute the weights corresponding to the neighbors which best approximate x_{new}. Using these weights we combine the values in Y corresponding to the neighbors to get an approximation of the new coordinates in the embedding, y_{new}. A similar approach can be used to map from the embedding coordinates to the initial high dimensional space.

The constraints we place on the weights also have an effect on the embeddings. For example, we can allow the weights to be negative values to give us an affine reconstruction, or we can force the weights to be positive to give a convex reconstruction. Affine weights can be found in closed form and they do not cause the embedding corners to be rounded. Convex weights provide more robustness to noise, but are found by solving a convex quadratic programming problem [37]. In our experiments, we found that convex weights result in a lower normalized error. Affine reconstruction weights resulted in very high normalized error in cases where the weights were of very high magnitude (such as 17.26 and -16.26 for two neighbors). Figure 2 shows the embeddings found using the LLE technique on sequences of a person walking on a treadmill obtained from the CMU MoBo database. The values needed for k and d depend on the intrinsic dimensionality of the input dataset, so there is no preset value. The problem of finding acceptable values for k and d is explored in more depth by Saul and Roweis [37].

5. System Dynamics Identification

In principle, the location of a target in a video sequence can be predicted using a combination of its (assumed) dynamics, empirically learned noise distributions and past position observations [24, 25, 26, 35]. While successful in many scenarios, these approaches remain vulnerable to model uncertainty and occlusion. Camps et al [11] addressed these difficulties by modeling the dynamical appearance and motion of the target as the output of a linear operator driven by a stochastic signal. In turn, they identified this operator using an extended Caratheodory-Fejer (CF) interpolation theory [38] that allows for dealing with operators that are not necessarily stable[1].

In this paper, we propose to use CF interpolation theory to identify the dynamic evolution of the data on the reduced manifolds. Let f_k denote the coordinates of a data point on a LLE manifold. Assume that the position of the data point f_k is related to the location of the previous N data points by

$$f_k \quad = \quad \mathcal{F}\mathbf{f} + \mathbf{e}, \qquad y_k = f_k + \eta_k \tag{4}$$

where $\mathbf{f} = \begin{pmatrix} f_{k-1} & \dots & f_{k-N} \end{pmatrix}^T$ contains the previous locations of the data, $\mathbf{e} = \begin{pmatrix} e_k & e_{k-1} & \dots & e_{k-m} \end{pmatrix}^T$ represents an input, y_k denotes the available measurement of the data, corrupted by noise η_k, and where $\mathcal{F}$ is a LTI suitable operator. A simple example is the case when the data points progress on the manifold with random acceleration:

$$f_{k+1} = 2f_k - f_{k-1} + e_{k-1}$$

[1] A simple example is the case of a person moving with random acceleration: a double integrator.

Moreover, assume that $\mathcal{F}$ admits a finite expansion of the form $\mathcal{F} = \mathcal{F}_p + \mathcal{F}_{np}$. Here, $\mathcal{F}_p = \sum_{j=1}^{n} p_j \mathcal{F}^j$ where $\mathcal{F}^j$ are known, given, not necessarily ℓ_2 stable operators that contain all the information available about possible modes of motion of the data on the manifold [2].

In this context, the next data point on the manifold f_k can be predicted by first identifying the relevant dynamics $\mathcal{F}$ and then using it to propagate its past N values. In turn, identifying the dynamics entails finding an operator $\mathcal{F} \in \mathcal{S} \doteq \{\mathcal{F}\colon \mathcal{F} = \mathcal{F}_p + \mathcal{F}_{np}\}$ such that $y - \eta = \mathcal{F}\mathbf{f} + \mathbf{e}$. In [38] Sznaier et al showed that establishing existence of this operator is equivalent to establishing feasibility of a set of Linear Matrix Inequalities (LMIs): a finite dimensional convex optimization problem.

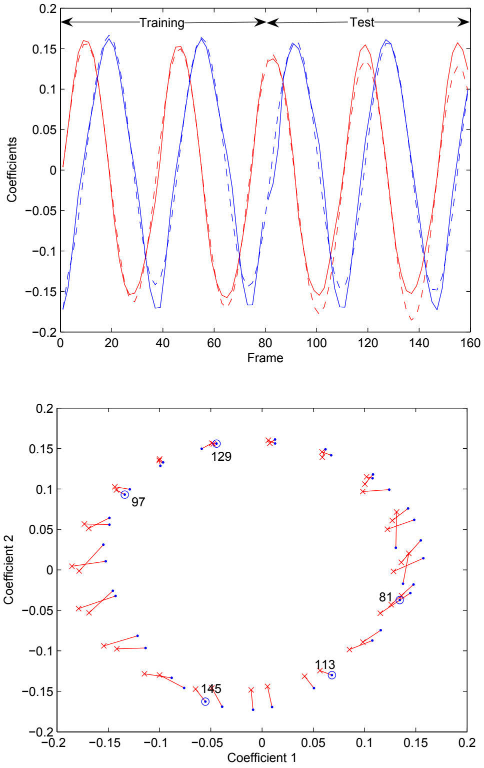

Figure 3. Learning temporal dynamics. Top: First two coefficients of sequence 2 as time progresses. Solid and dotted lines show actual and interpolated coefficients, respectively. Bottom: The predicted(red) and actual(blue) points on the embedding.

Figure 3 illustrates the use of CF interpolation to learn the temporal evolution of the points on an embedding. In this example, CF interpolation was applied to one of the embeddings shown in Figure 2 corresponding to a sequence of 160 frames. The dynamics of the points on this embedding was learned from its first 80 points, assuming an impulse signal as the input. Figure 3 (top) shows the close agreement between the temporal evolution of the coordinates of the points on the embedding and the positions predicted by the CF identified dynamics. An alternative view of these results is given in Figure 3 (bottom) where the predicted and actual points on the embedding are shown.

6. Learning View Correspondences

After obtaining low dimensional representations of a set of video sequences, we want to learn correspondences between views across sequences. One way to learn this correspondence is to align the embeddings so that corresponding views map to the same low dimensional coordinates. Another option is to model correspondence as an input-output LTI system, where the embedding coordinates of one view are the input to the system and the corresponding image embedding coordinates are the output. These approaches are described in more detail next.

6.1. Correspondences By Embedding Alignment

Finding correspondences between views of two video sequences X^1 and X^2 becomes trivial if their corresponding manifolds are aligned – i.e. if corresponding views $x_i^1 \in X^1$ and $x_j^2 \in X^2$ have *identical* low dimensional embedding representations $y_i^1 = y_j^2$. In general one-to-one correspondences between all training views are not available, since the cameras may not be synchronized or one camera may be occluded at times. However, it is not unreasonable to assume that *some* correspondences might be available. In this case, the method proposed in [23, 22] can be used to align the manifolds.

First we divide the data sets into subsets for which we know correspondences and for which we do not. Let X_c^1 and X_c^2 contain the same number of samples each, where x_i^1 corresponds to x_i^2. Similarly X_u^1 and X_u^2 contain the samples from each sequence for which we do not know correspondences (X_u^1 and X_u^2 can be empty and do not necessarily have the same number of samples).

To align two data sets where we know the correspondence of some or all of the samples, we first compute L^1 and L^2 as shown in Equation 3, where $X^1 = \left[\begin{array}{cc} X_c^1 & X_u^1 \end{array}\right]$ and $X^2 = \left[\begin{array}{cc} X_c^2 & X_u^2 \end{array}\right]$. We can then split each L^k into

[2] If this information is not available the problem reduces to purely non–parametric identification by setting $\mathcal{F}^j \equiv 0$. In this case the proposed approach still works, but obtaining comparable error bounds requires using a larger number of samples.

corresponding and non-corresponding parts:

$$L^k = \begin{bmatrix} L^k_{cc} & L^k_{cu} \\ L^k_{uc} & L^k_{uu} \end{bmatrix}$$

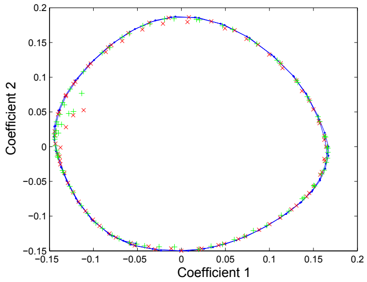

Figure 4. Embeddings aligned using LLE. Blue dots: training embeddings. Red X: test sequence 2 embeddings. Green +: test sequence 5 embeddings.

To find the embedding where $Y^1_c = Y^2_c$ is a hard constraint, we let

$$L = \begin{bmatrix} L^1_{cc} + L^2_{cc} & L^1_{cu} & L^2_{cu} \\ L^1_{uc} & L^1_{uu} & 0 \\ L^2_{uc} & 0 & L^2_{uu} \end{bmatrix}$$

and we then find the eigenvalues and eigenvectors for the solution. Once the embedding is computed, we can then map a new sample x^1_{new} into the embedding using the method described above to get y^1_{new}, which we assume is equal to y^2_{new} since the embeddings are aligned for the two sequences. We can then generate the second image by mapping from y^2_{new} to x^2_{new}. The results of this approach are illustrated in Figure 4 where the embeddings from Figure 2 are now aligned using LLE.

6.2. Correspondences By System Identification

An alternative approach to finding view correspondences is to capture the temporal correlations between sequences with a LTI operator that generates as output the points on the manifold from one camera when it is excited with a sequence of points from the manifold of the other camera as an input. This operator can be easily identified with the CF interpolation technique described in section 5, by setting in equation (4) f and e to the coordinates of sets of points in the first and second manifold, respectively[3].

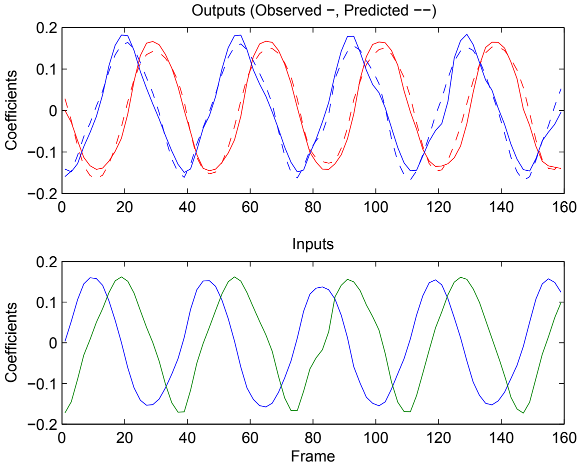

Figure 5. View correspondences using system dynamics. Top: First two output coefficients as time progresses. Solid and dotted lines show actual and interpolated coefficients, respectively. Bottom: First two coefficients of sequence 2 are the inputs.

Figure 5 shows plots of the temporal evolution of the coordinates of the points on two embeddings, and the predictions obtained by learning the dynamic relation between them. In this case, f was set to the coordinates of the first 80 points of one embedding and e was set to the coordinates of the corresponding points on the second embedding. The plot on the top of the figure shows the accuracy of the predictions for the next 80 points, obtained using the learned dynamics excited with the coordinates from the second embedding.

7. Generating Views

If the correspondences between views and their dynamics are learned using the methods described above, they can be used to generate new views in two situations: (1) when at time t, we have the image of an object in one view but not in the other, and (2) when we do not have the image of an object in any of the views at time t but we had it in the previous views.

In the first case, we can generate a new image in one of two ways, depending on how the correspondences were learned. If the embeddings were aligned during training by the dimensionality reduction method, then we can simply map the input view x_{in} onto the embedding to get a corresponding y_{in}. Since the embeddings of both views are aligned, $y_{in} = y_{out}$, so we simply map y_{out} into the output space using the neighbors of y_{out} from the output sequence. If the embeddings were aligned using system identification, then y_{in} and y_{out} are not equal, but are related by a dy-

[3] Note that the number of points in f and e do not have to be the same.

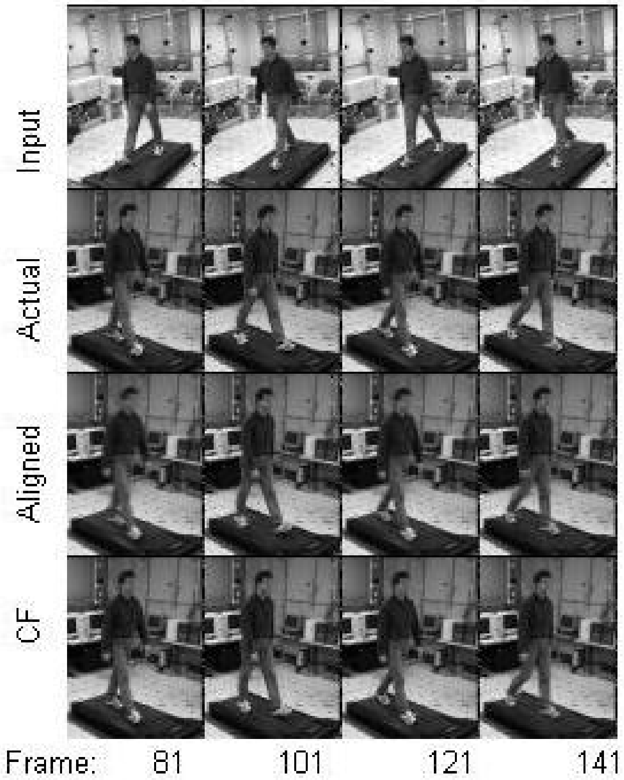

Figure 6. Generating one sequence from another. Row 1: input. Row 2: actual images. Rows 3 and 4: generated by aligned LLE and CF interpolation, respectively.

namic system that we learned. Thus, we can obtain y_{out} from a sequence of inputs from the other manifold using the identified dynamics, and then map it into the high dimensional output space to get a new view. We note that each mapping(to and from) will use different neighboring points in the embedding since the training sequences can be of different sizes and not all images in the sequences are in one-to-one correspondence. Figure 6 illustrates the results of using both methods to generate missing views on the treadmill sequences. We conducted our experiments on the first 160 frames of the *slowWalk* image sequence from the CMU MoBo database. The first 80 images were used to train our embeddings and the last 80 were used for testing the reconstruction of the views. One sequence (top row) is used as input to generate the other (row 2). Both methods are very effective at reconstructing the actual views.

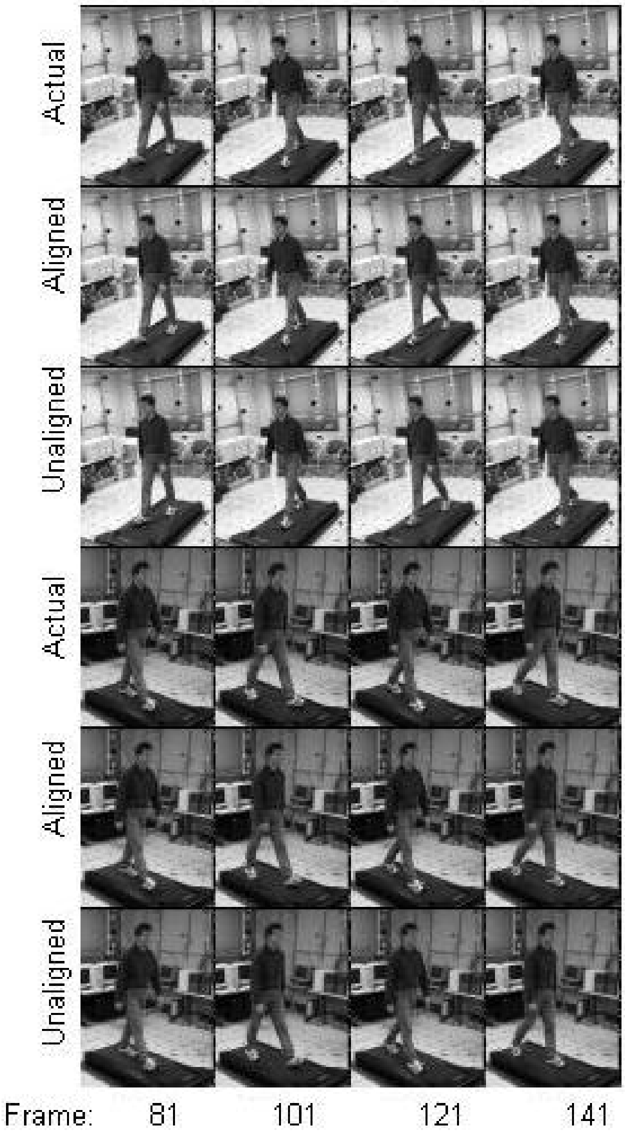

Figure 7. Generated and actual images generated by predicting position on embedding.

In the second case, we can predict new views in one of two ways, again depending on how the correspondences were learned. If correspondences were learned as part of the dimensionality reduction step, there is only one embedding for all images. The temporal dynamics of the low dimensional coordinates along the embedding can then be learned and used to predict where on the low dimensional embedding a view will be in the future, y_{future}. From that point, we can generate the high-dimensional views by mapping into the spaces of each of the input sequences. Similarly, if system identification was used to learn correspondences, the embeddings will be separate for each view, so the dynamics will be learned for each embedding separately and used to generate a new position on each embedding from which a new view can be constructed. Figure 7 illustrates the result of predicting views using both methods. We used the first 80 frames to learn the low dimensional embeddings and then learned the temporal dynamics of the coefficients of the low dimensional embeddings to predict the next 80 views.

8. Experimental Results

For our experiments, we implemented a tracker that extracts persons from multi-camera views and, given an initial manual labeling, tracks the persons and their appearance throughout the sequence, while maintaining their correct identities. For the foreground segmentation, we used the Codebook Background Subtraction algorithm [28]. During the training period, we tracked each person using the blob tracker described by Argyros and Lourakis [2] and extracted the appearance template for each person. During the occlusion periods, the appearance templates could no longer be extracted in one of the videos. However, we used one of our proposed proposed methods, alignment of embeddings through LLE, to create the views of each person despite the occlusion. When the occlusion period ends, we compare the two extracted templates with our generated templates to make sure that the identities are correct, and relabel if necessary. We note that the persons had very similar appearance – both persons were wearing yellow shirts and jeans and both persons were of approximately the same build. Thus, methods that normally depend on such appearance characteristics as color would not be able to maintain correct identities. Figure 8 shows selected frames before, during, and after the occlusion period. In the corner of each view are the templates maintained by the tracker. The templates for person 2, which are generated during the occlusion are provided at the bottom of the figure. Additional results are available at www.umiacs.umd.edu/~morariu/demo.html.

9. Conclusions

Previous approaches to establishing object correspondence in multi-camera systems have required matching features across views, using camera calibration information, or making planar world assumptions. Because of the difficulties that arise in these approaches, we sought to extract the spatial and temporal correlations present in multiple camera views using nonlinear manifold learning and target dynamics. Nonlinear manifold learning techniques allow us to extract intrinsic coordinates of the observed objects, ameliorating the problem of high dimensionality when working with images. To provide robustness to noise and occlusion, we incorporate the dynamics of the calculated intrinsic coordinates.

To improve the method in the future, we can study additional manifold learning methods, such as Hessian LLE, Semidefinite Embedding and Geodesic Nullspace Analysis which are more robust when extracting non-convex manifolds. In addition, improved mapping functions from embedding coordinates could improve generated appearance templates. In future work, we will study the use of this method in more complicated scenarios (e.g. outdoor scenes) with more complex motion.

10. Acknowledgments

We thank the University of Maryland for allowing us to use the Keck Laboratory and the Codebook Background Subtraction code. Also, the blob tracking code was written by the first author at the Navy Center for Applied Research in Artificial Intelligence.

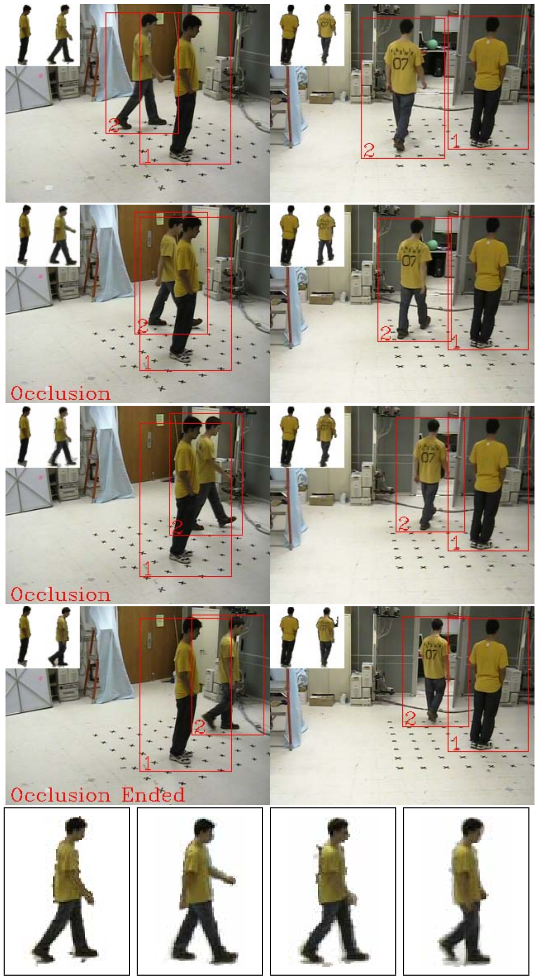

Figure 8. Learned correspondence is used to generate appearance of occluded person and to maintain identity. Top: tracker views. Bottom: templates of occluded person.

References

[1] A.Mittal and L. S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *IJCV*, 51(3), 2003.

[2] A. Argyros and M. I. A. Lourakis. Real time tracking of multiple skin-colored objects with a possibly moving camera. In *ECCV*, volume 3, pages 368–379, 2004.

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

[4] M. Black and T. Ellis. Multiple camera image tracking. In *PETS*, 2001.

[5] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *IJCV*, 26(1):63–84, 1998.

[6] M. Brand. Subspace mappings for image sequences. In *Workshop Statistical Methods in Video Processing*, 2002.

[7] M. Brand. Charting a manifold. In *NIPS*. MIT Press, 2003.

[8] M. Brand. Continuous nonlinear dimensionality reduction by kernel eigenmaps. In *IJCAI*, pages 547–554, 2003.

[9] M. Brand. From subspaces to submanifolds. In *BMVC*, 2004.

[10] Q. Cai and J. K. Aggarwal. Tracking human motion in structured environments using a distributed camera system. *PAMI*, 22(8):1241–1247, 2000.

[11] O. I. Camps, H. Lim, C. Mazzaro, and M. Sznaier. A caratheodory-fejer approach to robust multiframe tracking. In *ICCV*, pages 1048–1055, 2003.

[12] Y. Caspi and M. Irani. A step towards sequence-to-sequence alignment. In *cvpr*, 2000.

[13] T. H. Chang and S. Gong. Tracking multiple people with a multi-camera system. In *ICCV*, 2001.

[14] R. Collins, O. Amidi, and T. Kanade. An active camera system for acquiring multi-view video. In *ICIP*, volume I, pages 517–520, 2002.

[15] D. Comaniciu, F. Berton, and V. Ramesh. Adaptive resolution system for distributed surveillance. *Real Time Imaging*, 8:427–437, 2002.

[16] S. L. Dockstader and A. M. Tekalp. Multiple camera tracking of interacting and occluded human motion. In *Proceedings of the IEEE*, volume 89, pages 1441–1455, October 2001.

[17] D. L. Donoho and C. E. Grimes. Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. In *Proceedings of the National Academy of Arts and Sciences*, volume 100, pages 5591–5596, 2003.

[18] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *IJCV*, 51(2):91–109, 2003.

[19] A. Elgammal. Nonlinear generative models for dynamic shape and dynamic appearance. *2nd International Workshop on Generative Model-Based Vision*, 2004.

[20] A. Elgammal and C.-S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *CVPR*, pages 681–688, 2004.

[21] A. Elgammal and C.-S. Lee. Separating style and content on a nonlinear manifold. In *CVPR*, pages 478–485, 2004.

[22] J. Ham, D. D. Lee, and L. K. Saul. Learning high dimensional correspondences from low dimensional manifolds. In *Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining at ICML*, pages 34–39, 2003.

[23] J. Ham, D. D. Lee, and L. K. Saul. Semisupervised alignment of manifolds. In *Artificial Intelligence and Statistics*, 2005.

[24] M. Isard and A. Blake. CONDENSATION – conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

[25] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the 1995 American Control Conference*, pages 1628–1632, 1995.

[26] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME Ser. D: J. Basic Eng.*, 83:95–108, March 1961.

[27] S. Khan and M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *PAMI*, 25(10):1355–1360, October 2003.

[28] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. S. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005.

[29] F. D. la Torre and M. J. Black. Dynamic coupled component analysis. In *CVPR*, volume 2, pages 643–650, 2001.

[30] F. D. la Torre and M. J. Black. Robust principal component analysis for computer vision. In *ICCV*, pages 362–369, 2001.

[31] F. D. la Torre and M. J. Black. Robust parameterized component analysis: theory and applications to 2d facial appearance models. *CVIU*, 91(1-2):53–71, 2003.

[32] L. Lee, R. Romano, and G.Stein. Monitoring activities from multiple video streams: Establishing a common frame. *PAMI*, 22(8):758–767, 2000.

[33] L. Lee and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *PAMI*, 22(8):758–767, August 2000.

[34] H. Murase and S. K. Nayar. Visual Learning and Recognition of 3-D Objects from Appearance. *IJCV*, 14:5–24, January 1995.

[35] B. North, A. Blake, M. Isard, and J. Rittscher. Learning and classification of complex dynamics. *PAMI*, 22(9):1016–1034, September 2000.

[36] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L. V. Gool. Color-based object tracking in multi-camera environments. In *DAGM*, Springer LNCS 2781, pages 591–599, 2003.

[37] L. K. Saul and S. T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal on Machine Learning Research*, 4:119–155, 2003.

[38] M. Sznaier, C. Mazzaro, and O. I. Camps. Open-loop worst-case identification of nonschur plants. *Automatica*, 39(6):1019–1025, June 2003.

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

[40] M. Turk and A. Pentland. Face Recognition Using Eigenfaces. In *CVPR*, pages 586–591, June 1991.

[41] J. J. Verbeek, S. T. Roweis, and N. A. Vlassis. Non-linear cca and pca by alignment of local models. In *NIPS*, 2003.

[42] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *CVPR*, pages 988–995, 2004.

[43] K. Q. Weinberger, F. Sha, and L. K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *ICML*. ACM Press, 2004.

[44] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimension reduction via local tangent space alignment. In *SIAM Journal of Scientific Computing*, volume 26, pages 313–338, 2004.