Robust Cooperative Visual Tracking: A Combined NonLinear Dimensionality Reduction/Robust Identification Approach*

Vlad I. Morariu[1] Octavia I. Camps[2] Mario Sznaier[2] Hwasup Lim[3]

[1] Computer Vision Laboratory, University of Maryland, College Park, MD 20742
morariu@umd.edu
[2] Robust Systems Lab, ECE Department, Northeastern University, Boston, MA 02115
{camps,msznaier}@ece.neu.edu
[3] Dept. of Elect. Eng., Penn State University, University Park, PA 16802, hxl211@psu.edu

Abstract. In this paper we consider the problem of robust visual tracking of multiple targets using several, not necessarily registered, cameras. The key idea is to exploit the high spatial and temporal correlations between frames and across views by (i) associating to each viewpoint a set of intrinsic coordinates on a low dimensional manifold, and (ii) finding an operator that maps the dynamic evolution of points over manifolds corresponding to different viewpoints. Once this operator has been identified, correspondences are found by simply running a sequence of frames observed from one view through the operator to *predict* the corresponding current frame in the other view. As we show in the paper, this approach substantially increases robustness not only against occlusion and clutter, but also against appearance changes. In addition, it provides a scalable mechanism for sensors to share information under bandwidth constraints. These results are illustrated with several examples.

1 Introduction

In this paper we consider the problem of robustly tracking multiple targets using several, not necessarily registered, cameras. In principle, tracking targets using multiple cameras should increase robustness against occlusion and clutter since, even if the targets appear largely occluded to some sensors, the system can recover by using the others. Furthermore, examining data from spatially distributed cameras can reveal activity patterns not apparent to single or closely clustered sensors. However, although intuitively appealing, multicamera tracking *does not necessarily improve robustness*. This is illustrated in Figure 1, showing the results of an experiment where a Kalman filter based tracker is implemented using data from two (registered) cameras. Even though the target is always visible in at least one of the cameras, the tracker still loses it, due to occlusion resulting in incorrect data from the other camera.

Avoiding situations like the one illustrated above requires an efficient coordination mechanism to (i) reject incorrect measurements, and (ii) maintain consistent identity

* 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.

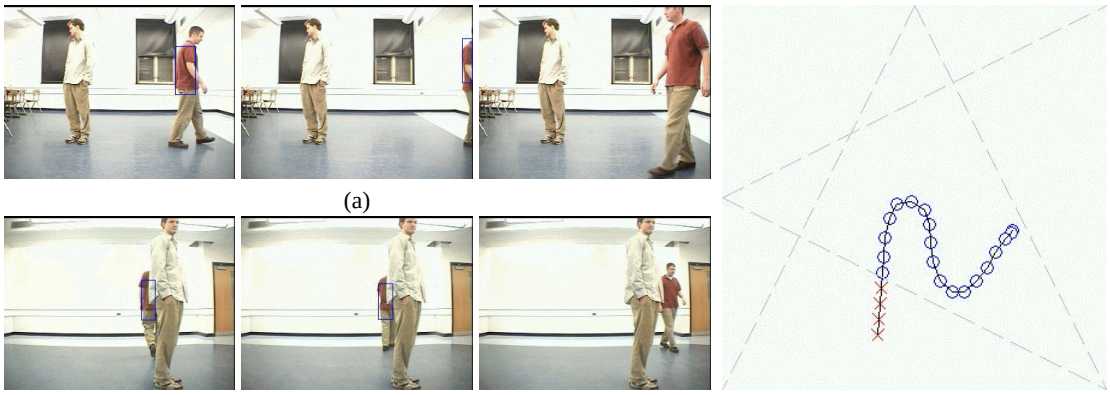


Fig. 1. Multicamera tracking: (a) West view, (b) North view, (c) The trajectory of the target is estimated incorrectly (red crosses) after the target leaves and re-enters the field of view of one of the cameras.

labels of the targets across views. Previous approaches to the "correspondence across views" problem include matching features such as color and apparent height [1,2,3,4], using 3D information from camera calibration [2,5,6,7,8] or computing homographies between views [9,10,11]. More recently, Khan and Shah [12] 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.

To avoid these difficulties, in this paper, we propose a new approach to the problem of cooperative multicamera tracking that does not require feature matching, camera calibration or planar assumptions. The key idea is to exploit the high spatial and temporal correlations between frames and across views by (i) associating to each viewpoint a set of intrinsic coordinates on a low dimensional manifold and (ii) finding an operator that maps the dynamic evolution of points over manifolds corresponding to different viewpoints. Once this operator has been identified, correspondences are found by simply running a sequence of frames observed from one view through the operator to *predict* the corresponding current frame in the other view. It is worth emphasizing that this approach substantially increases robustness not only against occlusion and clutter, but also against appearance changes. In addition, it provides a scalable mechanism for sensors to share information under bandwidth constraints. These results are illustrated with several examples

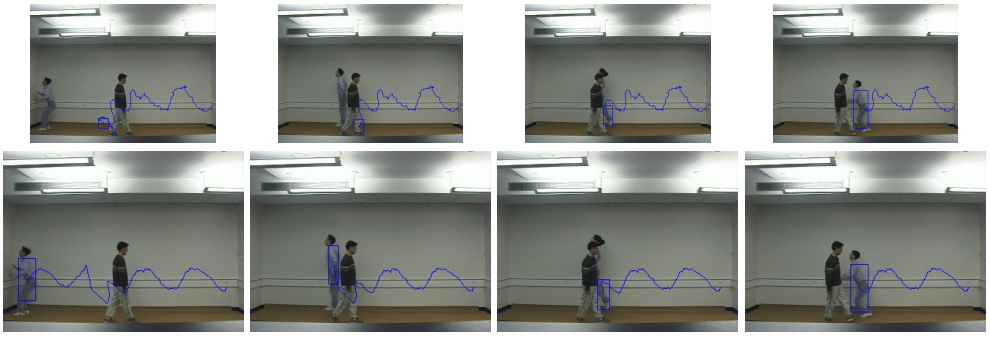
2 Notation

$\mathcal{H}_\infty$ denotes the space of functions with bounded analytic continuation inside the unit disk, equipped with the norm: $\|G\|_\infty \doteq ess\sup_{|z|<1} \overline{\sigma}\{G(z)\}$, where $\overline{\sigma}(.)$ is the max-

imum singular value. ℓ_∞ denotes the space of vector valued sequences $\{\mathbf{x}_i\}$ equipped with the norm: $\|\mathbf{x}\|_\infty \doteq \sup_i \|\mathbf{x}_i\|_\infty$. Similarly, ℓ_2 denotes the space of vector valued sequences equipped with the norm: $\|\mathbf{x}\|_2^2 = \sum_{i=0}^{\infty} \|\mathbf{x}_i\|^2$, where $\|.\|$ is the usual euclidian norm in R^n. Given a sequence $\{\mathbf{x}_k\}$, $\mathbf{x}(z) \doteq \sum_{i=0}^{\infty} \mathbf{x}_k z^k$ denotes its z–transform. Finally, given a finite sequence$\{x_k\}_{k=0}^{n-1}$, T_x^n denotes its corresponding (lower triangular) Toeplitz matrix:

$$\mathsf{T}_x^n \doteq \begin{bmatrix} x_o & 0 & \dots & 0 \\ x_1 & x_o & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n-1} & x_{n-2} & \dots & x_1 & x_o \end{bmatrix}$$

3 Dynamic Identification Based Robust Tracking



Frame 150 Frame 105 Frame 95 Frame 85

Fig. 2. Tracking in the presence of occlusion. Top: Unscented Particle Filter based tracker loses the target due to occlusion. Bottom: Combination Identified Dynamics/Kalman Filter tracks through the occlusion.

In this section we show that robust multicamera tracking can be reduced to a convex optimization problem. For simplicity, in the sequel we first present the main ideas using the simpler single camera case and then extend these ideas to multicamera scenarios. 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 [13,14,15,16]. While successful in many scenarios, these approaches remain vulnerable to model uncertainty and occlusion, as illustrated in the top portion of Figure 2. Following the approach introduced in Camps et al [17] for the single camera case, in this paper we will address these difficulties by modeling the motion of the target as the output of an operator driven by a stochastic signal. Specifically, consider first the simpler case where the dynamics of the target are approximately linear and start by

modelling the evolution of y, the position of a given target feature as:

$$y(z) = \mathcal{H}(z)e(z) + \eta(z) \tag{1}$$

where e_k and η_k represent a suitable input and measurement noise, respectively, $y(z),\ e(z)$ and $\eta(z)$ denote the corresponding z-transforms, and where the operator $\mathcal{H}$ is not necessarily ℓ_2 stable. For example, in the case of a feature moving with random acceleration, $H(z) = \frac{z^2}{(z-1)^2}$. Further, we will assume that the following *a priori* information is available:

(a) Set membership descriptions $\eta_k \in \mathcal{N}$ and $e_k \in \mathcal{E}$. These can be used to provide deterministic models of the stochastic signals e, η.

(b) $\mathcal{H}$ admits an expansion of the form $\mathcal{H} = \overbrace{\sum_{j=1}^{N_p} p_j \mathcal{H}^j}^{\mathcal{H}_p} + \mathcal{H}_{np}$. Here $\mathcal{H}^j$ are known, given, not necessarily ℓ_2 stable operators that contain all the information available about possible modes of the target[1].

In this context, the next location of the target feature y_k can be predicted by first identifying the relevant dynamics $\mathcal{H}$ and then using it to propagate the effect of the input e. In turn, identifying the dynamics entails finding an operator $\mathcal{H}(z) \in \mathcal{S} \doteq \{\mathcal{H}(z) \colon \mathcal{H} = \mathcal{H}_p + \mathcal{H}_{np}\}$ such that $y - \eta = \mathcal{H}e$, precisely the class of interpolation problem addressed in [18]. As shown there, such an operator exists if and only if the following set of equations in the variables $\mathbf{p}, \mathbf{h}$ and K is feasible:

$$\mathsf{M}(\mathbf{h}) = \begin{bmatrix} \mathsf{I} & \mathsf{T}_h^T \\ \mathsf{T}_h & K^2\mathsf{I} \end{bmatrix} \geq 0 \tag{2}$$

$$\mathbf{y} - \mathsf{P}\mathbf{p} - \mathbf{h} \in \mathcal{N} \tag{3}$$

where T_h denotes the Toeplitz matrix associated with the sequence $\mathbf{h} = [h_1, \ldots, h_n]$, the first n Markov parameters of $\mathcal{H}_{np}(z)$, and $\mathsf{P} \doteq [f^1\ f^2\ \cdots\ f^{N_p}]$, where f^i is a column vector containing the first n Markov parameters of the i-th transfer function $\mathcal{H}^i(z)$[2].

The effectiveness of this approach is illustrated in the bottom portion of Figure 2, showing that a Kalman filter based tracker using the identified dynamics for prediction, instead of a purely assumed simple model such as constant acceleration, is now able to track the target past the occlusion.

Consider now the situation where several (roughly) registered cameras are available. In this case the resulting geometric constraints translate into additional convex constraints that can be added to the identification above. This allows for individual cameras to accurately "guess" the location of a momentarily occluded target by simply translating to the local coordinate system measurements provided by other (non–occluded)

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

[2] Here, we have assumed without loss of generality (by absorbing the spectral properties of e into $\mathcal{H}$, if necessary), that $e_k = \delta(0)$, a unit impulse applied at $k = 0$.

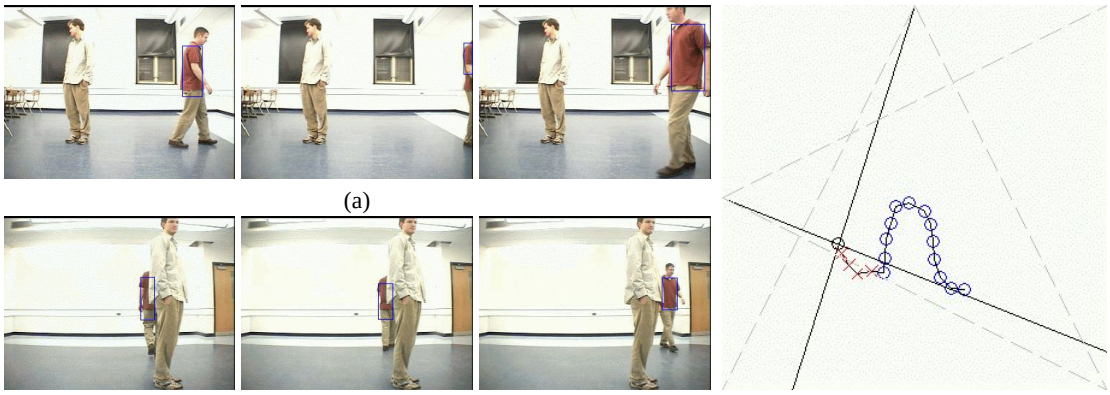


Fig. 3. Dynamics based multicamera tracking: (a) West view, (b) North view, (c) The trajectory of the target is correctly estimated (red crosses) even after the target leaves and re-enters the field of view of one of the cameras.

cameras and then propagating these measurements through the local model. Figure 3 shows the result of applying the approach outlined above to the same two–camera example used in the introduction. As shown there, the resulting tracker is now capable of continuous tracking, even when the target is momentarily occluded to one of the cameras. In this example, the experimental information used for identifying the dynamics consisted of centroid position measurements from the first 12 frames, where the target is not occluded. The *a priori* information, estimated from the non–occluded portion of the trajectory is:

1. measurement noise level 5 pixels
2. $\mathcal{H}_p \in \text{span}[1, \frac{1}{z-1}, \frac{z}{z-1}, \frac{z}{(z-1)^2}, \frac{z^2}{(z-1)^2}]$

Using this information, the minimum value of K yielding feasibility of the LMI (2) was found to be $K = 5 \cdot 10^{-4}$, indicating that indeed the relevant dynamics are captured by the parametric portion $\mathcal{H}_p$. During operation of the tracker, the target in each camera was segmented by the backprojection method using the hue histogram and occlusion was detected by changes in its size. In this event, the camera used information from the second sensor, when available, together with the local dynamics, to predict the position of the target.

4 Handling Nonlinear Dynamics and Computational Complexity

As illustrated with the simple example above, the approach outlined in the previous section has the potential to exploit multicamera information to accomplish robust tracking in the presence of severe occlusion. However, extending this approach to realistic, more complex scenarios requires addressing the issues of (i) nonlinear target dynamics and

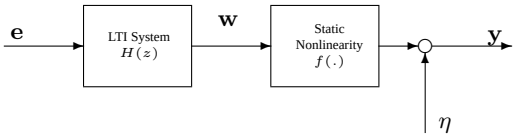


Fig. 4. Wiener System Structure

(ii) the computational complexity entailed in combining data from multiple sensors, due to the poor scaling properties of LMI based identification algorithms[3]. As we show in this section, both issues can be addressed by using nonlinear dimensionality reduction methods to project features to points on low dimensional manifolds where the dynamics are linear. Computationally efficient camera coordination can be achieved by having the cameras share projections onto these manifolds (and associated dynamical models), rather than high dimensional raw video streams. Since the projection onto the lower dimensional manifold can be modelled as a static nonlinearity, this approach leads naturally to a Wiener system structure of the form illustrated in Figure 4, consisting of the interconnection of a LTI system $H(z)$ and a memoryless nonlinearity $f(.)$. Identification of the linear dynamics on the manifold can be accomplished using essentially the same methods described in Section 3; identification of the nonlinearity $f(.)$ is addressed next.

4.1 Nonlinear Manifold Learning

Correlation of image sets has been extensively used in image compression, object recognition and tracking [20,21,22,23,24]. 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 [25,26] and to model dynamic texture [27].

However, image data does not usually lie in a linear subspace but instead on a low dimensional nonlinear manifold within the higher dimensional space [28,29,30,31,32,33,34,35,36,37,38,39]. 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 [36,40,41,42,38,39] seek to find an embedding of the data which preserves some relationship between the datasets, without providing an explicit mapping function.

[3] Recall that the computational complexity of conventional LMI solvers scales as (number of decision variables)^{10}[19].

Ideally, we would like to use a nonlinear manifold learning technique such as [37,28,43,30] that provides both the mapping and the embedding of our training set. However, such luxury comes at extra computational cost and algorithm complexity. Thus, in order to obtain algorithms compatible with real time operation, in this paper we use the locally linear embedding (LLE) algorithm to find the embedding of the data [36]. Though LLE does not directly provide a mapping from the high dimensional image space to the embedding space, methods similar to those described in [36] can approximate the mapping.

Given a set of images $X = [x_1 \dots x_n] \in \mathbb{R}^{D \times n}$, where x_i is the view of an object at time i, we want to find an embedding $Y = [y_1 \dots y_n] \in \mathbb{R}^{d \times n}$ 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

$$W = \underset{W}{\operatorname{argmin}} \sum_i |x_i - \sum_j W_{ij} x_j|^2 \tag{4}$$

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

$$Y = \underset{Y}{\operatorname{argmin}} \sum_i |y_i - \sum_j W_{ij} y_j|^2 \ . \tag{5}$$

Letting

$$L = (I - W)^T (I - W), \tag{6}$$

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.

Fig. 5. Representative frames from a walking sequence.

To map a new vector x_{new} into the embedding, we use the method described in [36]. 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

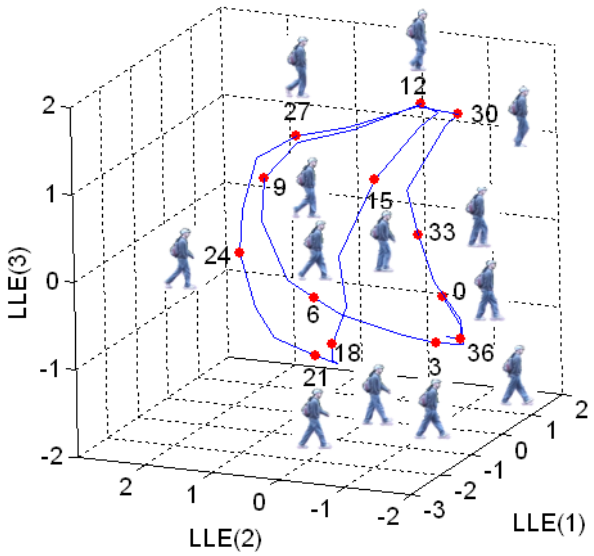


Fig. 6. Low dimensional representation of the walking sequence using Locally Linear Embeddings(LLE).

the embedding coordinates to the initial high dimensional space. 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 [36]. 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 [36]. 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).

Figures 5 and 6 illustrate the projection of a walking sequence onto a low dimensional manifold using the LLE technique. Figure 7 shows the embeddings of sequences of a person walking on a treadmill obtained from the CMU MoBo database.

4.2 System Dynamics Identification in Manifold Space

Once the low dimensional manifold has been found, the dynamics governing the motion there can be found using the identification approach outlined in Section 3, by simply using as data the projection w_k on the manifold, rather than the actual high dimensional feature y_k (see Figure 4).

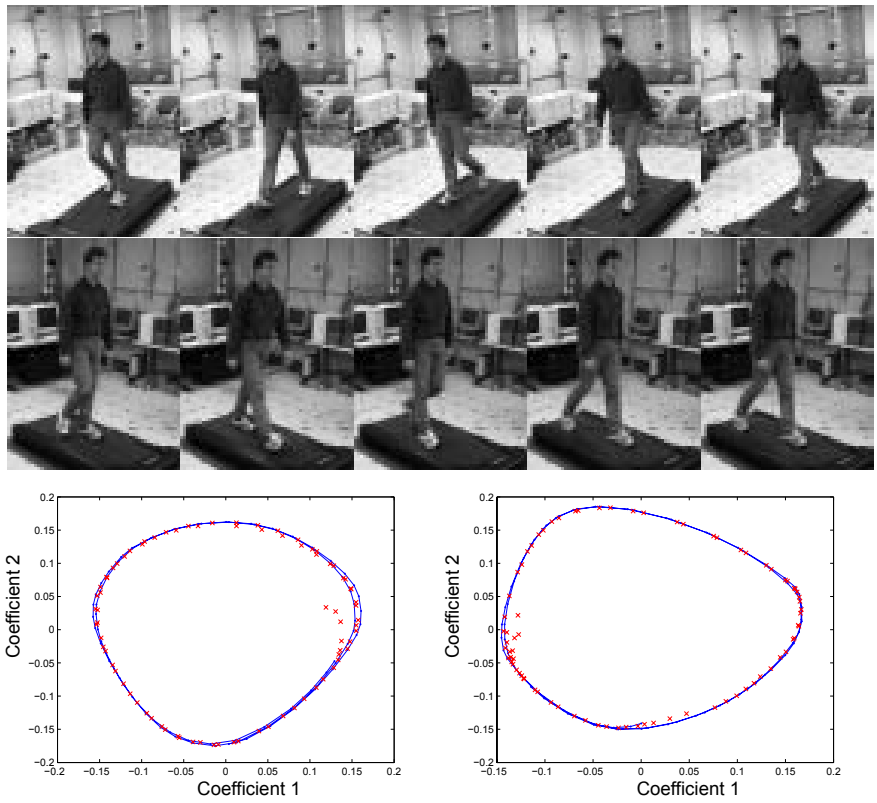


Fig. 7. Top: Sample images. Bottom: Embeddings of two sequences found by LLE. Blue and red points are training and test image embedding coordinates, respectively.

Figure 8 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 Fig. 7 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 8 (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 Fig. 8 (bottom) where the predicted and actual points on the embedding are shown.

4.3 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

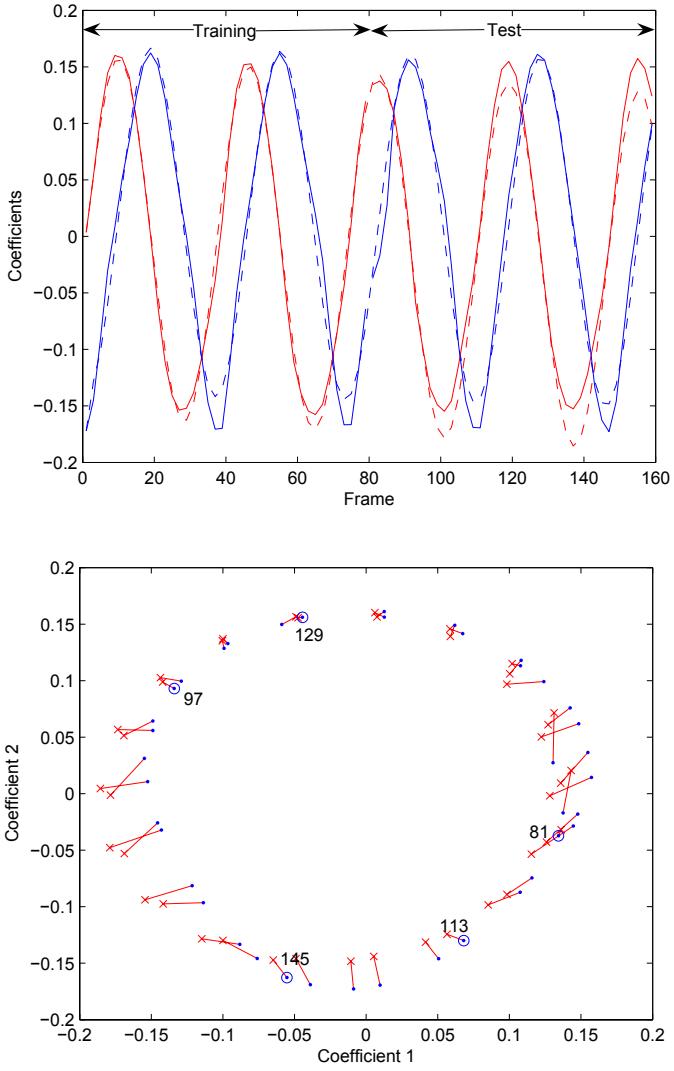


Fig. 8. Learning temporal dynamics. Top: First two coefficients of sequence 2 as time progresses. Solid and dotted lines show actual and interpolated coefficients, respectively. Bottom: The predicted(red) and actual(blue) points on the embedding.

and the corresponding image embedding coordinates are the output. These approaches are described in more detail next.

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 [34,35] 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 6, where $X^1 = \left[X_c^1 \; X_u^1 \right]$ and $X^2 = \left[X_c^2 \; X_u^2 \right]$. We can then split each L^k into corresponding and non-corresponding parts:

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

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

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

and we then find the eigenvalues and eigenvectors for the solution. Once the embedding is computed, we can then map a new sample x_{new}^1 into the embedding using the method described above to get y_{new}^1, which we assume is equal to y_{new}^2 since the embeddings are aligned for the two sequences. We can then generate the second image by mapping from y_{new}^2 to x_{new}^2. The results of this approach are illustrated in Fig. 9 where the embeddings from Fig. 7 are now aligned using LLE.

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 Sect. 3, by setting in equation (1) f and e to the coordinates of sets of points in the first and second manifold, respectively[4]. This approach is illustrated in Figure 10. Figure 11 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.

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

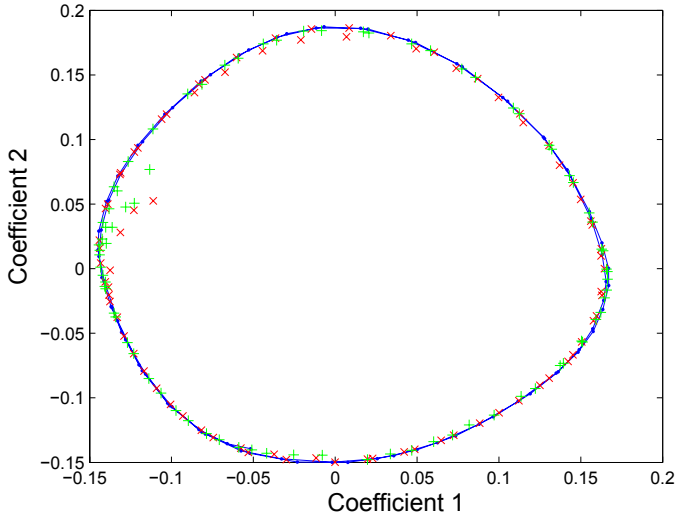


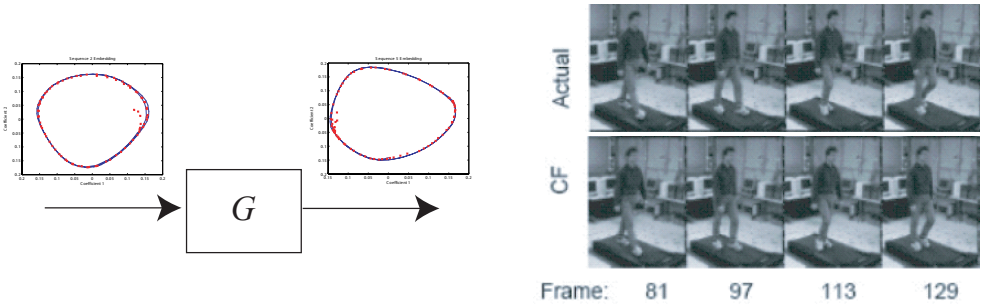
Fig. 9. Embeddings aligned using LLE. Blue dots: training embeddings. Red X: test sequence 2 embeddings. Green +: test sequence 5 embeddings.

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.

4.4 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, $y_{\text{in}} = y_{\text{out}}$, 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 dynamic 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 12 illustrates the



(a) (b)

Fig. 10. (a) Operator mapping manifolds (b)Actual (top) and Predicted (bottom) correspondences.

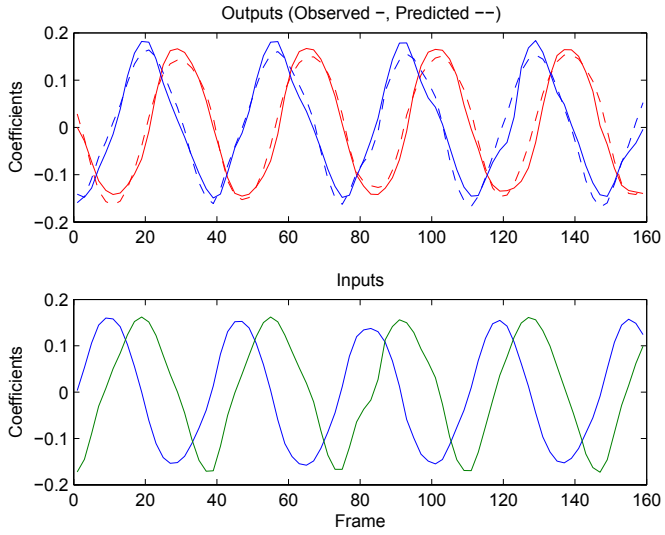


Fig. 11. View correspondences using system dynamics. Top: First two output coefficients as time progresses. Solid and dotted lines show actual and interpolated coefficients, respectively. Bottom: First two coefficients of sequence 2 are the inputs.

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.

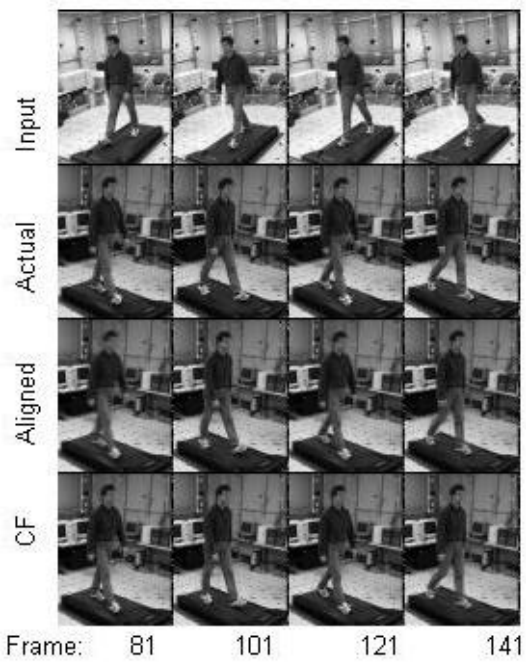


Fig. 12. Generating one sequence from another. Row 1: input. Row 2: actual images. Rows 3 and 4: generated by aligned LLE and CF interpolation, respectively.

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 13 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.

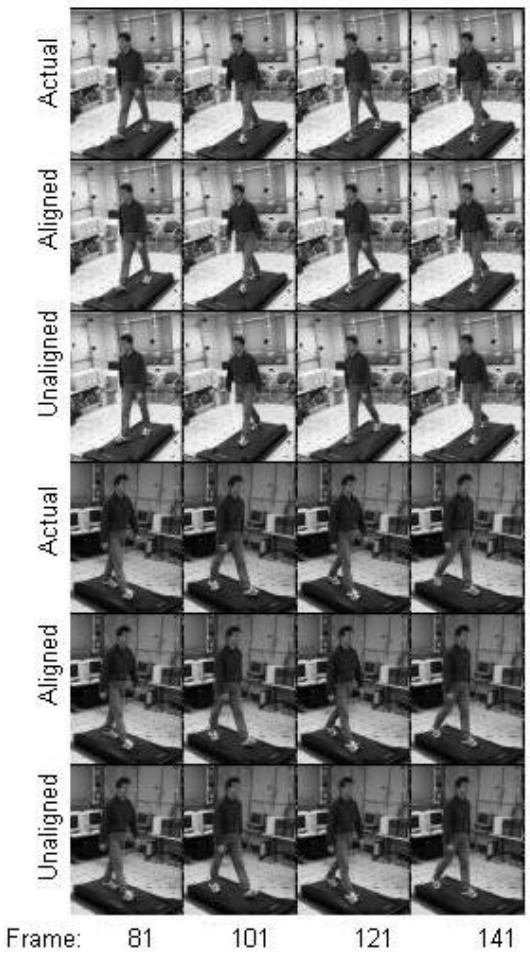


Fig. 13. Generated and actual images generated by predicting position on embedding.

5 Experimental Validation

5.1 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 14 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.

Fig. 14. Example of tracking in two views. Row 1: The input images. Row 2: Normalized person appearance.

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 [44]. During the training period, we tracked each person using the blob tracker described by Argyros and Lourakis [45] 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 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 15 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 and the corresponding

videoclips are available at www.umiacs.umd.edu/~morariu/demo.html and http://robustsystems.ee.psu.edu.

6 Conclusions

Dynamic vision – the confluence of control and computer vision – is uniquely positioned to enhance the quality of life for large segments of the general public. Aware sensors endowed with tracking and scene analysis capabilities can prevent crime, reduce time response to emergency scenes and allow elderly people to continue living independently. Moreover, the investment required to accomplish these goals is relatively modest, since a large number of imaging sensors are already deployed and networked. For instance, the number of outdoor surveillance cameras in public spaces is already large (10,000 in Manhattan alone), and will increase exponentially with the introduction of camera cell phones capable of broadcasting and sharing live video feeds in real time. The challenge now is to develop a theoretical framework that allows for *robustly* processing this vast amount of information, within the constraints imposed by the need for real time operation in dynamic, partially stochastic scenarios. In this paper we showed that efficient camera coordination leading to robust tracking in the presence of occlusion and clutter can be accomplished by exploiting a combination of identification and manifold discovery tools. The main idea is to exploit the high degree of spatio–temporal correlation of the data to project it, via nonlinear dimensionality tools, to a low order manifold where the underlying dynamics are approximately linear. Once in this manifold, tracking is accomplished by using robust identification tools to extract a compact model of the dynamics that can be used to predict the next position of the target, thus assisting in overcoming occlusion and disambiguating targets with similar appearance. Efficient camera coordination is accomplished by having the sensors share the low order data and associated models in these manifolds, rather than raw video streams. These results were illustrated with several examples. Research is currently under way seeking to reduce even further the amount of data to be shared among sensors by exploiting concepts from Information Based Complexity to eliminate redundancies.

7 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.

References

1. Cai, Q., Aggarwal, J.K.: Tracking human motion in structured environments using a distributed camera system. PAMI **22** (2000) 1241–1247
2. Chang, T.H., Gong, S.: Tracking multiple people with a multi-camera system. In: ICCV. (2001)
3. Nummiaro, K., Koller-Meier, E., Svoboda, T., Roth, D., Gool, L.V.: Color-based object tracking in multi-camera environments. In: DAGM. Springer LNCS 2781 (2003) 591–599

4. Comaniciu, D., Berton, F., Ramesh, V.: Adaptive resolution system for distributed surveillance. Real Time Imaging **8** (2002) 427–437
5. Black, M., Ellis, T.: Multiple camera image tracking. In: PETS. (2001)
6. A.Mittal, Davis, L.S.: M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. IJCV **51** (2003)
7. Collins, R., Amidi, O., Kanade, T.: An active camera system for acquiring multi-view video. In: ICIP. Volume I. (2002) 517–520
8. Dockstader, S.L., Tekalp, A.M.: Multiple camera tracking of interacting and occluded human motion. In: Proceedings of the IEEE. Volume 89. (2001) 1441–1455
9. Lee, L., Romano, R., G.Stein: Monitoring activities from multiple video streams: Establishing a common frame. PAMI **22** (2000) 758–767
10. Lee, L., Stein, G.: Monitoring activities from multiple video streams: Establishing a common coordinate frame. PAMI **22** (2000) 758–767
11. Caspi, Y., Irani, M.: A step towards sequence-to-sequence alignment. In: cvpr. (2000)
12. Khan, S., Shah, M.: Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. PAMI **25** (2003) 1355–1360
13. Isard, M., Blake, A.: CONDENSATION – conditional density propagation for visual tracking. IJCV **29** (1998) 5–28
14. Julier, S., Uhlmann, J., Durrant-Whyte, H.F.: A new approach for filtering nonlinear systems. In: Proceedings of the 1995 American Control Conference. (1995) 1628–1632
15. Kalman, R.E., Bucy, R.S.: New results in linear filtering and prediction theory. Trans. ASME Ser. D: J. Basic Eng. **83** (1961) 95–108
16. North, B., Blake, A., Isard, M., Rittscher, J.: Learning and classification of complex dynamics. PAMI **22** (2000) 1016–1034
17. Camps, O.I., Lim, H., Mazzaro, C., Sznaier, M.: A caratheodory-fejer approach to robust multiframe tracking. In: ICCV. (2003) 1048–1055
18. Parrilo, P.A., Pena, R.S.S., Sznaier, M.: A parametric extension of mixed time/frequency domain based robust identification. IEEE Trans. Autom. Contr. **44** (1999) 364–369
19. Paganini, F., Feron, E.: LMI methods for robust $\mathcal{H}_2$ analysis: A survey with comparisons. In Ghaoui, L.E., Niculescu, S., eds.: Recent Advances on LMI methods in Control. SIAM press (1999)
20. Turk, M., Pentland, A.: Face Recognition Using Eigenfaces. In: CVPR. (1991) 586–591
21. Murase, H., Nayar, S.K.: Visual Learning and Recognition of 3-D Objects from Appearance. IJCV **14** (1995) 5–24
22. Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. IJCV **26** (1998) 63–84
23. la Torre, F.D., Black, M.J.: Robust principal component analysis for computer vision. In: ICCV. (2001) 362–369
24. la Torre, F.D., Black, M.J.: Robust parameterized component analysis: theory and applications to 2d facial appearance models. CVIU **91** (2003) 53–71
25. Brand, M.: Subspace mappings for image sequences. In: Workshop Statistical Methods in Video Processing. (2002)
26. la Torre, F.D., Black, M.J.: Dynamic coupled component analysis. In: CVPR. Volume 2. (2001) 643–650
27. Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S.: Dynamic textures. IJCV **51** (2003) 91–109
28. Brand, M.: Charting a manifold. In: NIPS, MIT Press (2003)
29. Brand, M.: Continuous nonlinear dimensionality reduction by kernel eigenmaps. In: IJCAI. (2003) 547–554
30. Brand, M.: From subspaces to submanifolds. In: BMVC. (2004)
31. Elgammal, A.: Nonlinear generative models for dynamic shape and dynamic appearance. 2nd International Workshop on Generative Model-Based Vision (2004)

32. Elgammal, A., Lee, C.S.: Inferring 3d body pose from silhouettes using activity manifold learning. In: CVPR. (2004) 681–688
33. Elgammal, A., Lee, C.S.: Separating style and content on a nonlinear manifold. In: CVPR. (2004) 478–485
34. Ham, J., Lee, D.D., Saul, L.K.: Semisupervised alignment of manifolds. In: Artificial Intelligence and Statistics. (2005)
35. Ham, J., Lee, D.D., Saul, L.K.: 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. (2003) 34–39
36. Saul, L.K., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. Journal on Machine Learning Research **4** (2003) 119–155
37. Verbeek, J.J., Roweis, S.T., Vlassis, N.A.: Non-linear cca and pca by alignment of local models. In: NIPS. (2003)
38. Weinberger, K.Q., Sha, F., Saul, L.K.: Learning a kernel matrix for nonlinear dimensionality reduction. In: ICML, ACM Press (2004)
39. Weinberger, K.Q., Saul, L.K.: Unsupervised learning of image manifolds by semidefinite programming. In: CVPR. (2004) 988–995
40. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290** (2000) 2319–2323
41. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation **15** (2003) 1373–1396
42. Donoho, D.L., Grimes, C.E.: Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. In: Proceedings of the National Academy of Arts and Sciences. Volume 100. (2003) 5591–5596
43. Zhang, Z., Zha, H.: Principal manifolds and nonlinear dimension reduction via local tangent space alignment. In: SIAM Journal of Scientific Computing. Volume 26. (2004) 313–338
44. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.S.: Real-time foreground-background segmentation using codebook model. Real-Time Imaging **11** (2005) 172–185
45. Argyros, A., Lourakis, M.I.A.: Real time tracking of multiple skin-colored objects with a possibly moving camera. In: ECCV. Volume 3. (2004) 368–379

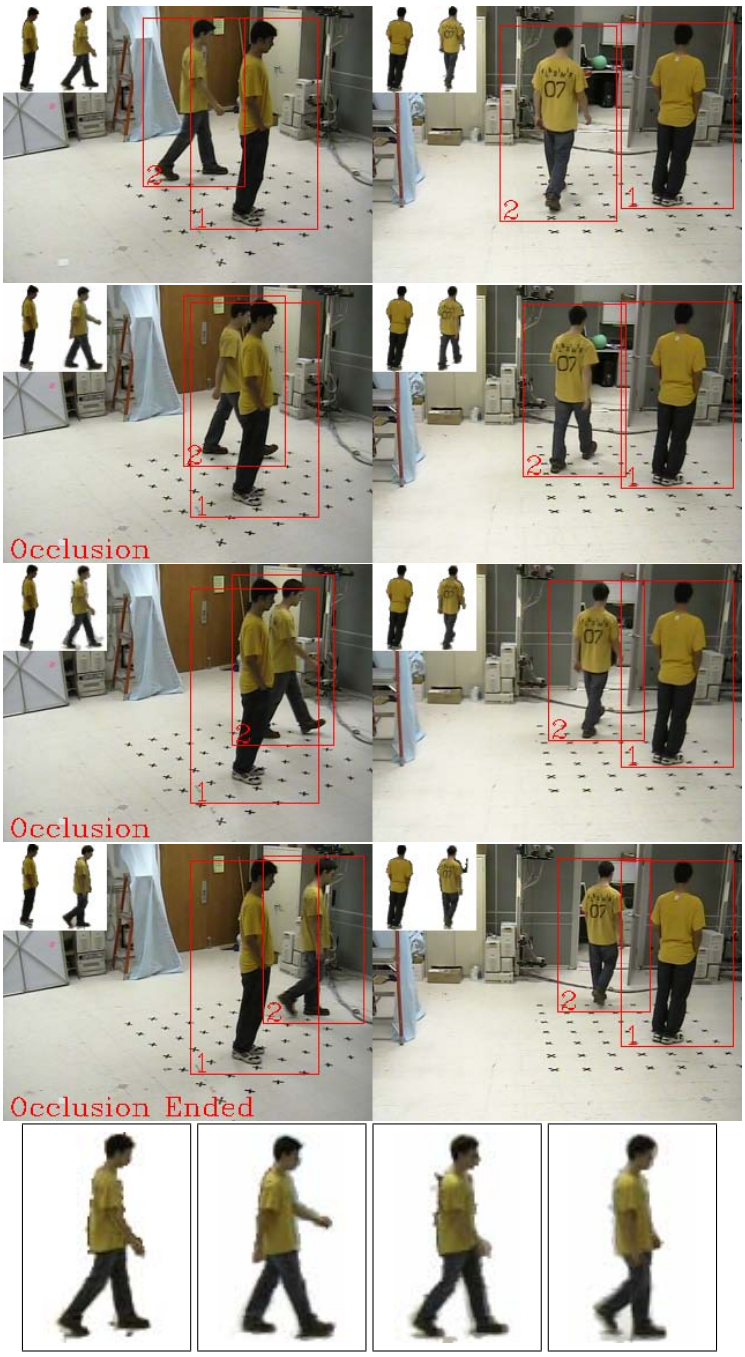


Fig. 15. Learned correspondence is used to generate appearance of occluded person and to maintain identity. Top: tracker views. Bottom: templates of occluded person.