ROBUST TWO-CAMERA TRACKING USING HOMOGRAPHY

Zhanfeng Yue, Shaohua Kevin Zhou and Rama Chellappa

Center for Automation Research and
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD, 20742
{zyue, shaohua, rama}@cfar.umd.edu

ABSTRACT

This paper introduces a two view tracking method which uses the homography relation between the two views to handle occlusions. An adaptive appearance-based model is incorporated in a particle filter to realize robust visual tracking. Occlusion is detected using robust statistics. When there is occlusion in one view, the homography from this view to other views is estimated from previous tracking results and used to infer the correct transformation for the occluded view. Experimental results show the robustness of the two view tracker.

1. INTRODUCTION

Multi view tracking has the obvious advantage over single view tracking because of its wide coverage range. When a scene is viewed from different viewpoints there are often regions which are occluded in some views but visible in other views. A visual tracking system must be able to track objects which are partially or even fully occluded. In this paper we present a wide baseline two view visual tracking method which handles occlusions using the homography relation between the two views. An adaptive appearance model is incorporated in Sequential Monte Carlo (SMC) framework to accomplish the single view tracking. Occlusion is detected using robust statistics. If the target to be tracked is far enough from the cameras, it can be assumed that the target moves on a dominant plane which induces a homography relation between the two views. When occlusion is detected in one view, the homography between the two views is estimated from previous tracking results. Correct transformation of the target in the occluded view can be inferred with the homography and the tracking result of the un-occluded view.

Some work has been done in handling occlusion for both single view tracking [9, 10] and multi view tracking [1, 2, 3]. In [9], an appearance model is used to accomplish tracking. When occlusion is detected, the "disputed" pixels are classified using a maximum likelihood classifier to infer the depth order of the objects, and update the appearance model accordingly. In [10], a dynamic Bayesian network which accommodates an extra hidden process for occlusion is used to cope with occlusion. Both [9] and [10] assume that the target is occluded by a known object, which gives a clue to infer the depth ordering or compute the observation likelihood. [1] presents a multi view tracking method using a set of calibrated cameras. The Kalman filter is used to track each object in 3D world coordinates and 2D image coordinates. In [2], the correlation of visual information between different cameras is learnt using Support Vector Regression and Hierarchical PCA to estimate the subject appearance across cameras. When occlusion is detected for one camera, correspondences across cameras are built using the appearance models acquired during training, and different cues are fused based on a Bayes's theorem to make a final tracking report. [3] uses a Bayesian network to fuse the independent observations from multiple cameras and produce the most likely 3D state estimates.

The method we propose in this paper uses the homography relation between two views to infer the transformation for the occluded view. Even when the target is partially or fully occluded by an unknown object, the tracker still can follow the target as long as it is visible from another view. No complicated inference scheme is used to fuse the multiple camera observation, nor 3D information needs to be explicitly recovered. The homogrphy can be robustly estimated from previous tracking results, and the motion inference for the target in the occluded view is also estimated robustly by utilizing all the points inside the tracking region. The computation is simple and fast. The result is satisfactory as shown in the experimental results.

The remainder of this paper is organized as follows. In section 2 we present the single view tracking using an appearance model that can handle occlusion detection. Section 3 describes how to handle occlusion with homography in a multiple view tracking system. Experimental results are shown in section 4, and section 5 concludes the paper.

2. SINGLE VIEW APPEARANCE TRACKING

This section presents an appearance model-based tracking system for a single view. The system processes the video frames captured under one single view and produces the tracking parameters for later use. The task of an appearance tracker is to infer the deformation (or tracking) parameter best describing the differences between the observed appearances and the appearance model. To accommodate the dynamics embedded in the video sequence, we employ a state space time series model.

Suppose $\{Y_1, ..., Y_t, ...\}$ are the observed video frames containing the appearances of the object to be tracked. We use an affine transformation $\mathcal{T}$ parameterized by θ_t and denote the appearance model by A_t. Our time series model is fully defined by

Prepared through collaborative participation in the Advanced Sensors Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0008. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

(a) *a state transition equation* and (b) *an observation equation*.

$$(a)\ \theta_t = \theta_{t-1} + U_t,\ (b)\ Z_t \doteq \mathcal{T}\{Y_t; \theta_t\} = A_t + V_t, \quad (1)$$

where U_t is the system noise and V_t is the observation noise. Our goal is to compute the posterior probability $p(\theta_t|Y_{1:t})$, which is used to estimate the 'best' parameter $\hat{\theta}_t$. Because this model is nonlinear (e.g. the affine transformation part), we use SMC technique [7, 8] to approximate $p(\theta_t|Y_{1:t})$ using a set of particles. We now specify the actual model choices.

2.1. Appearance model A_t

The appearance model A_t is crucial in a tracker. If a fixed template, say $A_t \equiv A_0$, is used, it is difficult to handle appearance changes in the video. On the other hand, one could use a rapidly changing model, say $A_t = \hat{Z}_t \doteq \mathcal{T}\{Y_t; \hat{\theta}_t\}$, i.e., the 'best' patch of interest in the previous frame, but this is susceptible to drift. Thus, it is necessary to have a model which is a compromise between these two cases. Mixture models are used in [6, 11]. In this paper, we simply adapt the appearance model to the changing appearances at a moderate pace.

We assume that (i) the appearance model A_t is associated with a mean image μ_t (the actual A_t in (1)) and a variance image σ_t^2 (included in V_t in (1)), and (ii) A_t summarizes the past observations under an exponential envelop with a forgetting factor α. When the appearance in the current frame has been tracked, i.e. $\hat{Z}_t$ is ready, we compute an updated appearance model A_{t+1} and use it to track in the next frame. After a maximum likelihood (ML) reasoning (skipped here due to space limitations), one can show that μ_{t+1} and σ_{t+1}^2 is updated in the following manner:

$$\mu_{t+1} = \alpha\mu_t + (1-\alpha)\hat{Z}_t;\ \sigma_{t+1}^2 = \alpha\sigma_t^2 + (1-\alpha)(\hat{Z}_t - \mu_t)^2. \quad (2)$$

Notice in the above equations, all μ's and σ^2's are vectorized and the operation is element-wise. Also, V_t is distributed as a multivariate normal density $\mathcal{N}(0, D(\sigma_t^2))$, where $D(\sigma_t^2)$ denotes a diagonal matrix with diagonal elements σ_t^2.

2.2. Adaptive noise U_t

The system noise U_t constrains the particle coverage. It is ideal to draw particles such that they are close to the object. In addition, the particle coverage should also accommodate the extent of clutter in the observation. To this end, we use $U_t \sim \mathcal{N}(\nu_t, r_t I)$, where ν_t is the 'instantaneous' velocity in the tracking parameter, r_t is the noise variance measuring the extent of clutter, and I is an identity matrix.

However, we have no knowledge of ν_t and r_t. We use a linear prediction scheme to estimate them. This prediction scheme is in spirit similar to finding an affine flows for the current 'best' patch in the next frame. Refer to [11] for details. As a consequence, the prediction scheme produces an estimate of ν_t and a prediction error ϵ_t. We take r_t as a monotone function of ϵ_t. Also, we vary the number of particles according to r_t.

2.3. Occlusion detection and cancellation

When occlusion happens in one view, we need a mechanism to detect it. We assume that occlusions produce large image differences which can be treated as 'outlier'. Outlier pixels cannot be explained by the underlying process. If a pixel x satisfies $|\hat{Z}_t(x) - \mu_t(x)|/\sigma_t(x) > c$ (we take $c = 0.75$), we declare the pixel an outlier. This actually corresponds to using a robust statistics [5]. If the number of the outlier pixels in $\hat{Z}_t$, say d_{out}, exceeds a certain threshold, i.e., $d_{out} > \lambda d_{total}$ (we take $\lambda = 0.13$), we declare an occlusion. Once occlusion is declared, we stop updating the appearance model and estimating the motion velocity and start using the information derived from other views to maintain tracking. To cancel an occlusion alarm, we compare the image warped from the other views with our observation till the error is consistently small. Tracking is then resumed.

3. OCCLUSION HANDLING WITH HOMOGRAPHY

We consider a wide baseline two view tracking system. Suppose the distance between the cameras and the object to be tracked is far enough to assume that the object moves on a dominant plane (e.g., consider a surveillance system in a parking lot and the dominant plane is the ground plane). We then resort to the homography between the two views to handle occlusions in tracking. Suppose P is a scene point lying on a plane π. Let p and p' be the projections of P in view 1 and view 2 respectively. Then there exists a 3×3 matrix H_π such that $p' \cong H_\pi p$ where H_π is called the homography matrix of the plane π [4]. For simplicity we will omit the subscript of H_π if there is no confusion in the following parts.

3.1. Homography estimation

Given a set of corresponding points $\mathbf{x_i} \leftrightarrow \mathbf{x_i'}$, where $\mathbf{x_i}$ come from view 1 and $\mathbf{x_i'}$ come from view 2, and writing $\mathbf{x_i'} = (x_i', y_i', \omega_i')^T$ with homogeneous coordinate, we can estimate the homography H between the two views using $\mathbf{x_i'} \times H\mathbf{x_i} = 0$ [4]. For each pair of corresponding points, three linear equations are written as

$$\begin{bmatrix} \mathbf{0}^T & -\omega_i'\mathbf{x_i}^T & -y_i'\mathbf{x_i}^T \\ \omega_i'\mathbf{x_i}^T & \mathbf{0}^T & -x_i'\mathbf{x_i}^T \\ y_i'\mathbf{x_i}^T & x_i'\mathbf{x_i}^T & \mathbf{0}^T \end{bmatrix} \begin{pmatrix} \mathbf{h_1} \\ \mathbf{h_2} \\ \mathbf{h_3} \end{pmatrix} = \mathbf{0} \quad (3)$$

where $\mathbf{h_i}, i = 1, 2, 3$ is a 3×1 vector made up of the entries in the i^{th} row of H.

By stacking the coordinates of all the corresponding points into a coefficient matrix A as shown in (3), H is the solution to the linear equation $A\mathbf{h} = 0$ where $\mathbf{h} = (\mathbf{h_1}^T, \mathbf{h_2}^T, \mathbf{h_3}^T)^T$. For a more accurate result, robust estimation methods like RANSAC or LMedS estimation can be used. Before feeding into the linear equation, the coordinates of all the points are normalized such that the centroid of the points is the coordinate origin $(0, 0)^T$, and their average distance from the origin is $\sqrt{2}$.

Finding correspondences is always challenging, especially for wide baseline views. Although H can be estimated from at least 4 pairs of corresponding points (the more we can find, the more robust H will be) in the initial frame, it is more robust to utilize the corresponding points in all frames. Assuming that the object moves on the same dominant plane for all the frames, it is clear that the corresponding points in all frames will contribute in estimating H. Suppose n pairs of corresponding points $\mathbf{x_i} \leftrightarrow \mathbf{x_i'}, i = 1, 2, \ldots, n$ on the object were picked in the initial frame, then their corresponding relation is kept for all the frames (through the inter-frame affine transformation $\mathcal{T}$'s known from the tracking result) and can be used to estimate H. One assumption used here is that for the corresponding points in the previous frames, after taking the affine transformations in both views for the current frame

Fig. 1. Two view tracking result with the target partially occluded by an unknown object, with the appearance model A_t shown at the upper right corner. Top row: tracking result for the un-occluded view. Middle row: tracking result for the partially occluded view without occlusion handling. Bottom row: tracking result for the partially occluded view with occlusion handling

(i.e., we have $\mathbf{y_i} \leftrightarrow \mathbf{y'_i}$ where $\mathbf{y_i} = \mathcal{T}_1\mathbf{x_i}, \mathbf{y'_i} = \mathcal{T}_2\mathbf{x'_i}$), they are still linked to each other with the same homography H as in the previous frames. This assumption usually will not hold since an affine transformation concatenated with a homography gives another homography instead of another affine transformation. Considering this, we do not directly assume $\mathbf{y_i} \leftrightarrow \mathbf{y'_i}$ as the true corresponding points. Instead, after getting $\mathbf{y_i}$'s in view 1, we do a random local search around $\mathbf{y'_i}$'s in view 2 to find the correct corresponding points for $\mathbf{y_i}$'s. Nevertheless, since the tracker works well enough in our experiment, which means the difference between the two frames can be satisfactorily described with an affine transformation, we can always find the correct correspondences in a very close neighborhood around $\mathbf{y'_i}$'s.

3.2. Transformation inference for occluded view

Suppose at frame j occlusion is detected for view 2, but not for view 1. Denote $\mathcal{T}_1^j$ and $\mathcal{T}_2^j$ as the affine transformations from frame $j-1$ to frame j for view 1 and view 2, respectively. We need to derive $\mathcal{T}_2^j$ from H and $\mathcal{T}_1^j$. Let $\mathbf{x}^{j-1}$ and $\mathbf{x'}^{j-1}$ be a pair of corresponding points at frame $j-1$ for view 1 and view 2 respectively. Then we have

$$\mathbf{x}^j = \mathcal{T}_1^j\mathbf{x}^{j-1};\ \mathbf{x'}^j = \mathcal{T}_2^j\mathbf{x'}^{j-1}, \tag{4}$$

and

$$\mathbf{x'}^{j-1} = H\mathbf{x}^{j-1};\ \mathbf{x'}^j = H\mathbf{x}^j. \tag{5}$$

Knowing H and $\mathcal{T}_1^j$, it is easy to derive from (4) and (5) that

$$\mathcal{T}_2^j = H\mathcal{T}_1^j H^{-1}. \tag{6}$$

Although (6) gives a theoretically correct solution for $\mathcal{T}_2^j$, it gives a homography while the sought solution is an affine transformation in accordance with the tracker. Practically $\mathcal{T}_2^j$ can be obtained from $\mathbf{x'}^{j-1}$'s and the inferred $\mathbf{x'}^j$'s. Writing $\mathbf{x'}^k = (x'^k, y'^k, 1)^T, k = j-1, j$, and $\mathcal{T}_2^j = \begin{pmatrix} \alpha_1 & \alpha_2 & t_x \\ \alpha_3 & \alpha_4 & t_y \\ 0 & 0 & 1 \end{pmatrix}$, we have

$$\begin{pmatrix} x'^j \\ y'^j \end{pmatrix} = \begin{pmatrix} x'^{j-1} & y'^{j-1} & 0 & 0 & 1 & 0 \\ 0 & 0 & x'^{j-1} & y'^{j-1} & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ t_x \\ t_y \end{pmatrix}. \tag{7}$$

A minimum of 3 pairs of corresponding points is needed to solve for $\mathcal{T}_2^j$ from (7). To get a more robust solution, we want to use all the points inside the tracking region to form an over-constraint linear equation and seek the least square estimate. To this end, we have to infer the coordinates for all the points inside the tracking region at frame j. Given 3 non-collinear points $\mathbf{p_i}, i = 1, 2, 3$ on the image of an object, the relation between $\mathbf{p_i}$'s and any other image point $\mathbf{q}$ on the object stays invariant under affine transformation $\mathcal{T}$, i.e., if $\mathbf{q}-\mathbf{p_1} = \beta_1(\mathbf{q}-\mathbf{p_2})+\beta_2(\mathbf{q}-\mathbf{p_3})$, then we have $\mathcal{T}(\mathbf{q}-\mathbf{p_1}) = \beta_1\mathcal{T}(\mathbf{q}-\mathbf{p_2})+\beta_2\mathcal{T}(\mathbf{q}-\mathbf{p_3})$. Recall that up until frame j we have stored $n(j-1)$ pairs of corresponding points in order to estimate H. With H and $\mathbf{x_i}^j, i = 1, 2, \ldots, n$, we can compute $\mathbf{x'_i}^j, i = 1, 2, \ldots, n$ with equation (5). Then the coordinates for all the other points inside the tracking region can be obtained accordingly. Here the number of initially picked correspondence pairs n can be as few as 3 if they are non-collinear, so the difficulty of finding enough number of correspondence points in the initial frame is greatly reduced.

4. EXPERIMENTAL RESULTS

Experiments were conducted on the PETS2001 test sequence [12]. Fig. 1 shows the sequence that three walking people are visible in all the frames for view 1, and are partially occluded by an in-

Fig. 2. Two view tracking result with the target fully occluded by an unknown object, with the appearance model A_t shown at the upper right corner. Top row: tracking result for the un-occluded view. Middle row: tracking result for the fully occluded view without occlusion handling. Bottom row: tracking result for the fully occluded with occlusion handling.

coming vehicle in some frames and reappear afterwards for view 2. The appearance model A_t is shown at the upper right corner of each frame. The top row of Fig. 1 shows the tracking result for view 1 (the un-occluded view). The middle row shows the tracking result for view 2 (the partially occluded view) without using homography to handle occlusion. We see that the appearance model keeps updating even when there is occlusion, and the tracker stays with the vehicle instead of the walking people. The bottom row of Fig. 1 shows the tracking result for view 2 for handling occlusion using homography. If there is no occlusion detected, the two views are tracked independently. When occlusion is detected in view 2, the appearance model ceases to update, and the affine transformation is inferred from the tracking result for view 1 and the computed H. It is clear that the tracker in view 2 still tracks the walking people even when they are partially occluded by the vehicle and regains control as soon as the people fully reappear.

Fig. 2 shows similar experiment results, except that the to-be-tracked walking person is fully occluded by the tree in view 2. The tracking results for view 1 (un-occluded view), view 2 (occluded view) without using occlusion handling, and view 2 using homography to handle occlusion are shown in the top, middle and bottom rows of Fig. 2, respectively. We can see from the bottom row that the tracker can track the person even though he is fully occluded by the tree, while the tracker stays where the tree is when the occlusion is not handled (as shown in the middle row).

5. CONCLUSIONS

We have described a two view tracking approach which uses the homography relation between two views to handle occlusions. An adaptive appearance model is used in a particle filter to accomplish single view tracking. We showed how to robustly estimate the homography with the previous tracking results and how to infer the correct transformation for the occluded view with the estimated homography and the tracking result for the un-occluded view. Experimental results show that the proposed multiple view tracking method can follow the target when it is partially or fully occluded by an unknown object. Our future work will extend this work to multiple view tracking by fusing every two-view's tracking result.

6. REFERENCES

[1] J. Black, T. Ellis and P. Rosin, "Multi View Image Surveillance and Tracking ," *Proceedings of the Workshop on Motion and Video Computing*, 2002.

[2] T. Chang, S. Gong and E. Ong, "Tracking Multiple People under Occlusion Using Multiple Cameras ," *BMVC*, 2000.

[3] S. Dockstader and A. Tekalp, "Multiple Camera Fusion for Multi-Object Tracking" *Proc. IEEE Workshop on Multi-Object Tracking* , pages 95–102, 2001.

[4] R. Hartley, and A. Zisserman "Multiple View Geometry in Computer Vision ," Cambridge University Press, 2000.

[5] P. Huber, "Robust Statistics ," Wiley, 1981.

[6] A. D. Jepson, and D. J. Fleet, and T. El-Maraghi "Robust Online Appearance Model for Visual Tracking," *Proc. IEEE Conference on Computer Vision and Pattern Recognition,* 1:415-422,2001.

[7] G. Kitagawa "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models," *J. Computational and Graphical Statistics*, 5:1-25, 1996.

[8] J. S. Liu, and R. Chen "Sequential Monte Carlo for Dynamic Systems," *Journal of the American Statistical Association*, 93:1031-1041, 1998.

[9] A. Senior, A. Hampapur, Y. Tian, L. Brown, S. Pankanti and R. Bolle, "Appearance Models for Occlusion Handling," *2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance* , 2001.

[10] Y. Wu, T. Yu and G. Hua "Tracking Appearances with Occlusions," *CVPR*, 2003.

[11] S. Zhou, R. Chellappa, and B. Moghaddam "Visual tracking and recognition using appearance-based modeling in particle filters," *Accepted by IEEE Trans. Image Processing*, 2003.

[12] http://www.cvg.cs.rdg.ac.uk/PETS2001/pets2001-dataset.html.