APPEARANCE-BASED TRACKING AND RECOGNITION USING THE 3D TRILINEAR TENSOR

Jie Shao, Shaohua Kevin Zhou, and Rama Chellappa

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

ABSTRACT

The paper presents an appearance-based adaptive algorithm for simultaneous tracking and recognition by generalizing the transformation model to 3D perspective transformation. A trilinear tensor operator is used to represent the 3D geometrical structure. The tensor is estimated by predicting the corresponding points using the existing affine-transformation based algorithm. The estimated tensor is used to synthesize novel views to update the appearance templates. Some experimental results using airborne video are presented.

1. INTRODUCTION

Video-based object tracking and recognition is gaining more interest in both computer graphics and computer vision communities. The conventional methods usually attack these two tasks separately [4]. A new framework was proposed in [10], which ca

n not only simultaneously perform both tracking and recognition, but also improve the recognition rate over the conventional methods without any reduction in tracking accuracy. In order to make this algorithm more reliable an adaptive approach using the following strategies was proposed in [9]. (1) Employing an adaptive appearance model and an adaptive velocity motion model [4] for representing inter-frame appearance changes. (2) Constructing intra- and extra-personal spaces to model the appearance changes between the video frames and gallery images. (3) Exploiting the fact that the gallery images are in frontal views. By embedding these improvements in a particle filter, enhancements in both stability and accuracy are achieved when confronted by pose and illumination variations.

However, [9] still limits itself by using an affine transformation. In many applications, the affine transform is not enough to describe appearance changes in a 3D scene. In this paper, we extend [9] to a more general situation by representing the object motion using a 3D transformation. By doing this, we generalize our model assumption from an affine transformation to a perspective transformation. The core of the algorithm is to use a trilinear tensor operator that links point correspondences across three images to represent the 3D transformation among the different views [1]. We first use the affine model algorithm to get sets of corresponding points, which leads us to an estimated trilinear tensor. Then by synthesizing a novel view from the derived tensor, we can refine our motion prediction. Meanwhile, the desired virtual view is also used to update the appearance model. By repeatedly applying the tensor operations in a sequence of reference images in an estimate-refine manner, we extend the previous tracking and recognition algorithm to a more general model. The most significant aspect of our approach is that we obtain our 3D transformation model without invoking the burdensome process of explicitly recovering a depth map or camera parameters. This makes the entire system more robust and efficient.

The rest of the paper is organized as follows. After a brief review of the adaptive algorithm for tracking and recognition in Sec. 2, we describe in Sec. 3 the method using trilinear tensor operators. Experimental results and discussions are presented in Sec. 4, followed by Sec. 5, which concludes the paper.

2. REVIEW OF AFFINE-BASED ADAPTIVE TRACKING AND RECOGNITION

In this section, we give a brief summary of the original algorithm [9][10] including the basic components of tracking and recognition, the original algorithm for simultaneous tracking and recognition, and further improvements making the algorithm adaptive.

2.1. Basic Components of Tracking and Recognition

In the recognition stage, the propagation model has three major components. They are the motion transition equation, the identity equation and the observation likelihood respectively. The task of tracking and recognition is then formulated as a statistical inference problem, and solved using a particle filter.

Partially funded by the DARPA VIVID program through a subcontract from SRI international.

The motion transition parameter is denoted by θ_t, which describes the kinematics of object (affine transform model is assumed here). The equation is:

$$\theta_t = \theta_{t-1} + \upsilon_t + \mu_t, t \geq 1 \quad (1)$$

where θ_{t-1} is the previous estimate, υ_t is a first-order prediction of motion velocity , and μ_t as an noise variable. Under the assumption that the identity of object remains the same during the procedure, the identity equation is represented as $n_t = n_{t-1}, t \geq 1$, where $n_t \in \mathbf{N} = \{1, 2, ..., N\}$. The index set $\mathbf{N}$ denotes objects in the gallery. The observation likelihood $p(Z_t|n_t, \theta_t)$ measures the inter-frame appearance changes. Appearances are transformed to match a fixed template and the transformation producing the minimum cost gives the best motion transition parameters at time t .

2.2. Original Algorithm

Assuming statistical independence among all noise variables and prior knowledge of $p(\theta_0|z_0)$ and $p(n_0|z_0)$, we estimate the posterior probability $p(n_t|z_{0:t})$ by marginalizing $p(n_t, \theta_t|Z_{0:t})$ over θ_t. Notice that as the model is nonlinear and non-Gaussian, a particle filter is invoked to implement numerical approximations of [3]. The basic idea of the particle filter is to draw samples, update weights and estimate probability [10].

2.3. Improved Algorithm with Adaptive Properties

Some improvements have been made to make the original algorithm more reliable. First, an adaptive appearance model is used to represent the inter-frame appearance changes. Secondly, the motion transition model is reformed to be adaptive by predicting the shift υ_t in (1) with a first-order linear approximation. In order to determine the quality of prediction, a measurement ϵ_t indicating the distance between the patch of interest and the updated appearance model is calculated. Finally, at each time t, the number of particles is also adjusted based on the variance and the entropy measure of $p(n_t|z_{0:t})$ at that time. The adaptive particle-filtering algorithm is described in [9].

3. 3D-TRANSFORM ALGORITHM

As mentioned in Sec. 2, the kinematic model in the original algorithm uses an affine transform model, which can be represented as:

$$\bar{X}' = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{R} \bullet \bar{X}. \quad (2)$$

The first 6 elements in $\mathbf{R}$ comprise the parameter set θ, which are estimated at each time t. An affine transform model is not sufficient in many practical situations. This motivates us to extend the algorithm to a rigid 3D transformation model.

3.1. 3D Rigid Transformation with Trilinear Tensor

A standard 3D transformation is represented as

$$\bar{X}' = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \mathbf{R} \bullet \bar{X} \quad (3)$$

followed by a projection from 3D space to 2D. All rigid 3D transformation, such as 3D rotation, translation and scaling can be expressed in that way. In order to avoid the complicated procedure of 3D reconstruction, we introduce a trilinear tensor to describe the rigid 3D transformation.

3.1.1. Trilinear Tensor

The trilinear tensor approach is based on the following scheme: three views should satisfy certain matching constraints in a trilinear form, represented by a tensor [1]. Therefore, given two views in correspondence and a tensor, the corresponding third view can be generated uniquely. Given $\mathbf{I}$,$\mathbf{A}$,$\mathbf{B}$ as corresponding camera matrices of first, second and third views respectively, the tensor, which is a $3\times3\times3$ array of 27 entries, can be represented using a bilinear function of the $\mathbf{A}$,$\mathbf{B}$:

$$\mathrm{T}_i^{jk} = \nu'^j b_i^k - \nu''^k a_i^j, i, j, k \in \{1, 2, 3\} \quad (4)$$

where a_i^j,b_i^k are the elements of the homographes $\mathbf{A}$,$\mathbf{B}$ for the second and third views respectively, and ν',ν'' are the epipoles of the first view in second and third views. Equation (4) needs at least seven matching points across three images to determine the 27 parameters in T. Once the parameters are recovered, a corresponding third view can be reprojected from $\mathrm{T}_{1,2,3}$ and the two corresponding sets of points in the first and second views as follows: let p,p' be the matching points in first and second views, then

$$p''^k \cong p^i s_j^\mu \mathrm{T}_j^{jk}, i, j, k \in \{1, 2, 3\} \quad (5)$$

which provides a set of four independent equations for computation of p'', the matching point in third view.

3.1.2. Implementation of trilinear tensor method

The complete procedure of finding the tensor and then synthesize a corresponding view is separated into several steps according to (4) and (5) at each time t. Details are given in [1].

• Finding matching points in three views (t, $t-1$ and $t-2$) to estimate the tensor: the corresponding points here can

either be derived using an optical flow method [6], or a particle filter algorithm, which is efficient to provide approximately corresponding points for successive images. Among those hundreds of points, we select a subset referred to as "good" points, (usually those points with smaller eigenvalue in "optic-flow" matrix above some predefined threshold will be selected.) and normalize them.

• The reprojection process is performed after the tensor is obtained using (5), generating a novel view. The corresponding sets are known from the previous computation. As these corresponding points sets cannot provide a dense correspondence, a warping procedure has to be performed in order to overcome the forward-mapping problem.

•The warping process works in a backward-mapping way. The novel view is inversely mapped to the source image, which guarantees that each pixel in the novel view is assigned a value.

3.1.3. HEIV algorithm for robust tensor estimation

In practice, the tensor estimation is always based on noisy measurements, which seriously affect the final accuracy of the result; therefore, robustness is a very important factor we need to consider. Fundamental numerical scheme (FNS) [2] and Heteroscedastic EIV (HEIV)[7] are two popular approaches that have been employed to solve such a problem. We use the HEIV algorithm, because of its weaker dependence on the initial solution and faster convergence [7]. Before applying HEIV, we write the tensor estimation (4) into a linear equation with respect to a vector θ that consists of 27 elements in a $3\times3\times3$ tensor T as:

$$\mathbf{Z}_i \cdot \theta_i = \vec{0}, \mathbf{Z}_i \in \mathbf{R}^{m\times 27}, \theta_i \in \mathbf{R}^{27\times 1} \qquad (6)$$

With (6) and an initial value of θ_0 usually obtained from the generalized total least squares (GTLS) solution [5], an iteration implementating the HEIV algorithm gives a more robust and accurate solution.

3.2. 3D Transform Algorithm

By employing the trilinear tensor to represent the 3D transformation, we extend the original method to perform a tracking and recognition in a 3D scene. At each time t, the new algorithm estimates the object motion using the following steps:

Step.1. For the current frame I_t , estimate the motion transition and find the corresponding point set P_t (denotes 2D coordinates of point) with respect to points in the previous frame I_{t-1} using the adaptive particle filter approach. The result of this step is the same as that of the original particle filter algorithm, and is regarded as a coarse approximation to be refined in the following steps.

Step.2. Draw sample sets $P_{ti}, i = 1, 2, ...N$ with probability $\mathcal{N}(P_t, \Sigma_t)$, where the diagonal matrix Σ_t is the covariance matrix of point set P_t, under the assumption that each point is independent, N is the number of samples. For each sample set P_{ti}:

Step.2.1 Select "good" points in P_{ti} to form a feature point set π_t, together with known π_{t-2},π_{t-1} (from previous frames) to estimate the tensor $\mathrm{T}_{t-2,t-1,t}$.

Step.2.2 Reproject the frame $\tilde{I_{ti}}$ by using $\mathrm{T}_{t-2,t-1,t}$ and known corresponding set P_{t-2},P_{t-1}.

Step.2.3 Compute the square error of $\tilde{I_{ti}}$ and I_t.

Step.3. Pick the sample set $P_{tk}, k = \arg\min_i (\tilde{I_{ti}} - I_t)^2$ as the current corresponding point set derived from the 3D transition model, which indicates the target. The corresponding tensor T is the current tensor, which provides useful geometry information.

Step.4. Update the current-view templates by synthesizing from current tensor and the previous templates. Compute matching measurements from the current frame object to these new templates.

Step.5. Update the current data. $t \rightarrow t+1$, go to step.1.

Some remarks follow:

1) Initialization: A tensor operator usually hinges on two or three different views. In our case, we use three sets of feature points to get an estimate of the tensor, two sets of corresponding points to synthesize a novel view. Thus, in the initial stage, we need at least two sets of feature points and corresponding points before we can estimate the tensor. Then, the third set of feature points is exactly the current feature point set at each time t, which can be derived from Step.2.1. Updating is performed as in Step.5 at each time t and the data being updated includes feature point sets, corresponding point sets, templates and the frame being processed.

2) Synthesis of templates: Obviously, a still 2D object template is not enough for providing the novel view of sufficient quality, which will impair the accuracy of recognition when image view changes. This can be compensated by using the tensor to update the template in the current view. Since the tensor operator reflects 3D information, we can inexplicitly include the 3D information of current frame into the template, which contributes significantly to the accuracy of recognition.

4. EXPERIMENTAL RESULTS

We present experimental results using airborne video sequences of vehicles moving on a highway. In the particle filter estimation step, the motion is characterized by $\theta = \{r_{11}, r_{12}, r_{21}, r_{22}, t_x, t_y\}$, where the first four elements are deformation parameters and the last two represent the 2D translation. A zero-mean-unit-variance normalization is applied to partially compensate for contrast variations.

In our experiments, 600 corresponding points in each frame are chosen to depict the object appearance. Among them, 14-16 "good" points are picked up as feature points to do tensor estimation. We also estimate the covariance matrix

of these points and model the location of the estimated corresponding points using zero-mean Gaussian distributions. Based on the estimated probability, we draw samples around each estimated corresponding point provided from the previous step with the assumption of independent distribution of each point. When doing image warping to synthesize a novel template, a convenient strategy to implement the backward mapping step is as follows: we can use the known corresponding points between the destination and the source image as reference points to find all the matching-pixel pairs between images. Some results are presented here. In Fig.1, the tracked sequence is shown with a white box denoting the object. Fig.2 demonstrates some synthesized templates rendered using the tensor operator.

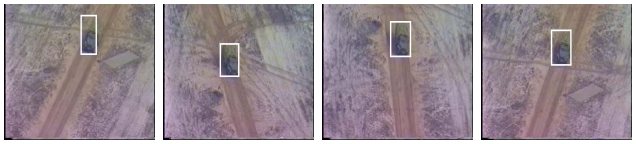

Fig. 1. Tracking results of a truck on the road (white box indicates the location of our target)

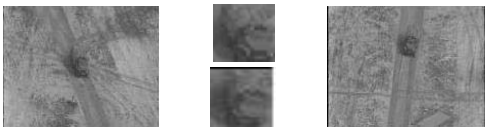

Fig. 2. Left column: first view in tensor estimation;middle column(top): the original template referred to first view; middle column(bottom): the synthesized template obtained using the estimated tensor; right column: the corresponding view.

As a comparison, we consider another feasible solution [8], which recovers the 3D model using a structure from motion algorithm (SfM) and then renders the appearance using texture mapping. We ran the two algorithms (3D-tensor, 3D-SfM) on video sequences for a 2-class problem. Fig. 3 presents the two gallery models ('Tank' and 'Ural') which are rendered by texture-mapping the 2D appearances onto the reconstructed 3D models via the SfM algorithm proposed in [8]. Fig.4 gives the plots of posterior probabilities. From the plots, both the algorithms give correct recognition results, but the 3D-tensor-model converges faster than the 3D-SfM-model.

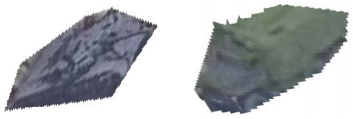

Fig. 3. The rendered appearance models by texture-mapping the 2D appearances onto the reconstructed 3D models. These two appearance models are used as the gallery images in the recognition stage. Left: 'Tank' model; Right:'Ural' model.

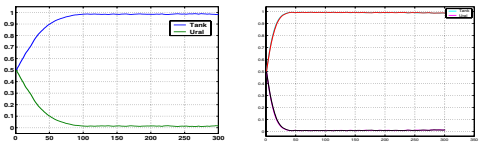

Fig. 4. The posterior distribution $p(n_t|y_{0:t})$ calculated using (left) 3D-SfM-model and (right) 3D-tensor-model.

5. CONCLUSION

We have extended the appearance-based adaptive tracking and recognition algorithm from an affine-transform model to the 3D-perspective-transform model using the trilinear tensor operator. The HEIV algorithm is adopted for estimating the tensor, and a warping technique is employed to handle novel-view-synthesis. Experiments give good results for tracking and recognition.

6. REFERENCES

[1] S. Avidan and A. Shashua. Novel view synthesis by cascading trilinear tensor. *IEEE Trans. Visualization and Computer Graphics*, 4:293–306, 1998.

[2] W. Chojnacki, M. Brooks, A. Hengel, and D. Gawley. On the fitting of surfaces to data with covariances. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 22:1294–1303, November 2000.

[3] M. Isard and A. Blake. Contour tracking by stochastec propagation of conditional density. *Proc. of European Conference on Computer Vision, Cambridge, UK, April, 1996*, pages 343–356, 1996.

[4] T. Jebara and A. Pentland. Parameterized structure from motion for 3d adaptive feedback tracking of faces. *Proc. of IEEE Computer Society Conf. on Computer Vision Pattern Recognition, Puerto Rico, June, 1997*, pages 144–150, 1997.

[5] Y. Leedan and P. Meer. Heteroscedastic regression in computer vision: Problems with bilinear constraint. *International Journal of Computer Vision*, 37:127–150, 2000.

[6] B. Lucas and T. Kanade. An iterative image registation technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[7] B. Matei and P. Meer. A general method for errors-in variables problems in computer vision. *Proc. of IEEE Computer Society's Conf. on Computer Vision and Pattern Recognition, Hilton Head, SC, June, 2000*, II:18–25, June 2000.

[8] G. Qian and R. Chellappa. Structure from motion using sequential monte carlo methods. *Proc. of Intl. Conf. on Computer Vision, Vancouver, Canada, July 2001*, pages 614–621, 2001.

[9] S. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-based modeling in particle filters. *Proc. Intl. Conf. on Multimedia and Expo., Baltimore, MD, July 2003, Accepeted for IEEE Trans. Image Processing*, 2003.

[10] S. Zhou, V. Kruger, and R. Chellappa. Probabilistic recognition of human faces from video. *Computer Vision and Image Understanding*, 91:214–245, July/Aug 2003.