TRACKING ALGORITHM USING BACKGROUND-FOREGROUND MOTION MODELS AND MULTIPLE CUES

Jie Shao[1]*, Shaohua Kevin Zhou*[2] *and Rama Chellappa*[1]

[1]Center for Automation Research and Department of ECE
University of Maryland, College Park, MD 20742
{shaojie,rama}@cfar.umd.edu
[2]Integrated Data System Department, Siemens Corporate Research
755 College Road East, Princeton, NJ 08540
{kzhou}@scr.siemens.com

ABSTRACT

We present a stochastic tracking algorithm for surveillance videos where targets are dim and of low resolution. Our tracker is mainly based on the particle filter algorithm. Two important novel features of the tracker include: A motion model consisting of both background and foreground motion parameters is built. Multiple cues are adaptively integrated in a system observation model when estimating the likelihood functions. Based on these features, the accuracy and robustness of the tracker has been improved, which is very important for surveillance problems. We present the results of applying the proposed algorithm to many videos.

1. INTRODUCTION

Visual tracking has many applications including robotics, video surveillance and image sequence analysis. Tracking small dim objects in video is an important and challenging research topic. However, developing an algorithm for tracking small dim objects is still an open problem. In the literature, many algorithms are available for precise object tracking in real time. Most of these algorithms work very well on large objects, but less so for small objects, which is important in surveillance applications, where there may only be tens of pixels on the target.

Many approaches have been suggested for addressing surveillance tracking problems. Four approaches, namely background subtraction [4], foreground tracking, [5], motion encoded tracking [1], and appearance-based tracking [10] have been taken. Each of these algorithms has its own emphasis on solving specific tracking problem; as a result, they all fail in many situations. For example, both intensity-based and edge-based trackers will be distracted by other objects or background clutters of similar intensity. Thus no single cue can perform efficiently in all kinds of situation. Another point is that usually in the non-static camera case, there exist two different movements caused by different sources. We separate them as background motion and foreground motion, which can be specified by two different sets of parameters. We present a new statistical method for tracking objects in surveillance videos, which has two features in the system model. 1) A dynamic motion model: we define different parameters for characterizing the background and foreground motions. 2) An observation model: we use a fused model with multiple cues. The former improves the tracking accuracy of the system, while the latter yields a higher system robustness.

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

The *dynamic model* in the system is a time series state space model parameterized by a tracking motion vector, denoted by θ. To describe the two types of motions, the motion vector θ for the entire system consists of two subsets of parameters for the background movement and foreground motions[6].

$$\theta = \{\theta^F; \theta^B\} = \{\alpha^F, dx^F, dy^F, s^F; \alpha^B, dx^B, dy^B, s^B\} \quad (1)$$

where θ^B represents the background parameters, θ^F represents the foreground parameters. Image rotation angle α^B, displacement dx^B, dy^B and scale s^B are four parameters in θ^B describing the background changes caused by the moving camera, while α^F,dx^F, dy^F,s^F in θ^F represent the movement of the foreground object motion. In most of surveillance scenarios when objects only occupy a small number of pixels, the displacements of the objects $\{dx^F, dy^F\}$ are good enough to describe the movements of targets.

The *observation model* fuses motion and intensity cues. Intensity information classifies objects from background due to different intensity values, while motion information helps to discriminate moving objects from relatively still background. When used separately, neither of them is robust enough to deal with all kinds of situations. Therefore in the proposed algorithm, the two cues are used together as measurements.

With dynamic and observation models defined, we use the particle filter as our basic tracking filter. This is under the realization that tracking targets in real world requires nonlinear models and non-Gaussian noise models, and particle filters are particularly appropriate. Since the system simultaneously estimates both background and foreground motion parameters using a single dynamic model, the efficiency of the algorithm is increased for: 1) segmenting background and foreground objects; 2) obtaining both motion information and intensity information from background motion vector θ^B; and 3) tracking the foreground target based on estimated foreground motion vector θ^F.

The rest of the paper is organized as follows. After a brief discussion of the proposed tracking algorithm in Section 2, we describe the tracking model that includes background and foreground models. The tracking observation model with multiple cues is introduced in Section 3. Experimental results and discussions are

presented in Section 4. Finally, in Section 5 we present the concluding remarks.

2. BACKGROUND-FOREGROUND TRACKING USING PARTICLE FILTER

2.1. Visual Tracking by Particle Filtering

The *particle filter* was originally proposed as a probability propagation model in [3] in the signal processing literature and has been used to solve many vision tasks [5]with a popular name as the *condensation* algorithm. The problem of tracking can be formulated as maximizing $p(\theta|Y_{t-1}) \propto \mathcal{L}(Y_t|\theta_t) \int p(\theta_t|\theta_{t-1})p(\theta_{t-1}|Y_{1:t-1})d\theta_{t-1}$. The particle filter approximates the current posterior distribution $p(\theta_t|Y_{1:t})$ by a set of weighted particles $S_t = \{\theta_t^{(j)}, \omega_t^{(j)}\}_{j=1}^{J}$. In each time instant, the weights are updated with the likelihood of the new observation combined with the former weights. Then, a resampling step is added to eliminate particles with lower weights.

2.2. Dynamic Model Components of Particle Filter

As already discussed in section 1, the motion in a video frame is caused by two different sources. One is due to the moving object itself, the other is due to the moving camera. Therefore, we define the state motion vector θ in (1), which is more appropriate to this kind of tracking scenario. Accordingly, the posterior distribution becomes a joint distribution of the background and the foreground, which is $p(\theta_t^B; \theta_t^F|Y_{1:t})$. Our goal is to find θ that maximizes the posterior probability, which require the solving of θ_t^B and θ_t^F simultaneously. We present an iterative algorithm to find a local solution by iteratively improving θ.

2.3. Iterative Optimization for Background Foreground Motion Estimation

Expectation-Maximization [2] is a technique for obtaining a maximum likelihood estimate (MLE) for a family of model parameters given some observed data. Though the EM algorithm is not necessary in our settings, the general idea that a local optimal solution can be achieved by iteratively optimizing the target function is adopted. Obviously, we have two different sets in the state vector: $\theta_t = \{\theta_t^B, \theta_t^F\}$, and the goal is to find θ_t that maximizes the posterior probability $\max \arg_{\theta_t} P(\theta_t|Y_{1:t}, \theta_{t-1})$ [8].

During each optimization stage, we first fix θ_t^F by using the previous estimated value θ_{t-1}^F, estimate θ_t^B, then estimate θ_t^F again. Multiple iterations may be required before proceeding to the next image, but based on experimental results, we have found that a single iteration may be sufficient. Using this assumption, time recursiveness and Markov properties, the relation between background and foreground distributions can be written as

$$p(\theta_t^F|\hat{\theta}_t^B, Y_{1:t}, \theta_{t-1}) \propto p(Y_t|\theta_t^F, \hat{\theta}_t^B)p(\theta_t^F|Y_{1:t-1}, \theta_{t-1}^F) \quad (2)$$

$$p(\theta_t^B|\hat{\theta}_t^F, Y_{1:t}, \theta_{t-1}) \propto p(Y_t|\theta_t^B, \hat{\theta}_t^F)p(\theta_t^B|Y_{1:t-1}, \theta_{t-1}^B) \quad (3)$$

2.4. Background Motion Model

How to find what has changed between two successive frames? As described in [6] , we include the background motion vectors into the system state vector which provide the necessary parameters for stabilization. Let $\mathbf{X} = (x, y)^T$, $d\mathbf{X}_t = (dx, dy)^T$. We then have a motion formula:

$$\mathbf{X}_t = s \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \mathbf{X}_{t-1} + d\mathbf{X}_t \quad (4)$$

where s is the scale factor, α is the rotation angle between the two frames, $d\mathbf{X}_t$ is the translation measured in the image coordinates at time t. According to the transform equation, four parameters $(\alpha_t^B, dx_t^B, dy_t^B, s_t^B)$ are used to describe the motion of the background between frame $t-1$ and t, where α_t^B is the rotation angle, dx_t^B and dy_t^B represent the translation parameters, and s_t^B is the scale factor. These parameters characterize the motion of the camera. The state transition is then approximated by using a first-order Markov chain and a mixture Gaussian noise model. One component of noise is the zero-mean Gaussian distribution, denoted as μ_t, accounting for sensor noise, digitization noise etc; the other is a non-zero-mean Gaussian distribution, denoted as ν_t, caused by the camera motion. The background motion equation is written as:

$$\theta_t^B = \theta_{t-1}^B + \nu_t + \mu_t = \widetilde{\theta}_t^B + \mu_t \quad (5)$$

Where $\widetilde{\theta}_t^B$ is the initial estimate derived using the stabilization algorithm proposed in [9].

2.5. Foreground Motion Model

Since the objects in surveillance videos usually tend to be very small, we simply apply a uniform distribution of the foreground motion vector $\theta_t^F = d\mathbf{X}_t^F = \{dx_t^F, dy_t^F\}$. Therefore, the new location of object equals $\mathbf{X}_t^F = \mathbf{X}_{t-1}^F + d\mathbf{X}_t^F$. If the objects are not small, we can also adopt an affine parameter motion model as the foreground motion model [10], as in (1).

3. MEASUREMENT MODEL USING MODALITY FUSION OBSERVATIONS

Our ultimate goal is to improve the tracking accuracy of foreground object space.In order to further improve the robustness of the tracking system in dynamically changing environments, we fuse multiple cues. In our system, all visual cues contribute simultaneously to the overall observation model, and the relative relevances of cues are determined by the frame context currently processed in the system, reflected as adaptive weights associated with cues in the integration process. That means, no cue is predetermined as the "optimal" cue, but the system itself will weigh the contributions of different cues according to the existing conditions. We designed a particle filter based visual tracker that fuses two principle cues, the appearance cue and the motion cue. For each cue, a likelihood model is constructed. These models are not entirely independent but are indirectly coupled by the result they agreed upon [7]. We further assume that the measurements are also conditionally independent given the state, so that the likelihood can be factorized as

$$\begin{aligned} \mathcal{L}(Y|x) &= \prod_{m=1}^{M} \mathcal{L}(Y^m|x) \\ \mathcal{L}(Y_t, \theta_t^B|\theta_t^F) &= \prod_{m=1}^{M} \mathcal{L}(Y_t^m, \theta_t^B|\theta_t^F) \end{aligned} \quad (6)$$

with M is the number of measurement sources. For each $p(Y^m|x)$, the construction of the likelihood model is same. The importance of each cue depends on the quality of the measurement the cue can provide to the tracker, thus the cues are incorporated in an adaptive manner.

3.1. Motion-encoded Observation Model

We obtain the motion-based observation as follows. Let $\Delta_t = I_t - T_{\hat{\theta}_t^B} I_{t-1}$, where Δ_t is the difference image and I_t and I_{t-1} are the original frames, $T_{\hat{\theta}_t^B}$ is the stabilizing function with $\hat{\theta}_t^B$. The advantage of using the motion cue Δ_t is: relative to the background, most of the targets in a surveillance video are in motion. In such cases, motion information turns out to be a decisive measurement to separate the foreground object from the relatively static background, especially when the intensities of background and foreground are not that different.

To analyze the motion of a foreground target, we process the difference images, instead of the original frames. For delineating the inside and the outside regions of the moving object, the edge gradient information of the Δ images is used. The edge image E is generated as $E = \Delta \otimes DoG$, where DoG is a 2D derivative of a Gaussian filter. We assume that the motion of one feature pixel on the target can describe the motion of the entire target. A support region is applied to each pixel to collect enough edge gradient measurements for computing the cost function. In addition, we assume that dx^F and dy^F are independent, and that their joint distribution is uniformly distributed in a specified 2D space $\mathcal{D}_2$. The function collects the match measurement in a pre-defined rectangle region $\mathbf{W_X}$ with $\mathbf{X}$ as its center.

The idea behind the foreground displacements estimation is: A particular pixel in Δ_{t-1} image will move by a distance in Δ_t image due to motion continuity. This is estimated by matching a region between successive Δ images using a cost function $D(\mathbf{X}; d)$ where d denotes the linear translation $d = (d_x, d_y)$. The cost function and the likelihood function are defined respectively as:

$$D(\mathbf{X}; d) = \sum_{\mathbf{Y} \in \mathbf{W_X}} \frac{|E_t(\mathbf{Y}) - E_{t-1}(\mathbf{Y} + d)|}{|\mathbf{W_X}|} \quad (7)$$

$$\mathcal{L}(Y_t^m, \theta_t^B | \theta_t^F) \propto \exp\left(-\frac{D(\mathbf{X}; d)}{2\sigma_1^2}\right) \quad (8)$$

where $|\mathbf{W_X}|$ is the number of pixels in the window $\mathbf{W_X}$, $\mathbf{X}$ represents a pixel position in the image $(x, y)^T$, and Y^m represents the motion cue.

3.2. Appearance-encoded Observation Model

Another principal cue is derived from the local-appearance. It is obtained by associating some reference template with the object of interest. Templates extracted from the candidate regions in the current frame are compared to this reference template, and smaller the discrepancy between the candidate and reference templates, higher is the probability that the object is located in the corresponding image region. The localization performance hinges on the reference template selection. To make it more robust, the template is set to contain two components, a stable component and a dynamic one. The stable component is the object model manually cropped in the initializing stage. The dynamic one is the tracked object in time instant $t-1$ with the same size of the stable template after accounting for the scale change, and is updated at each time instant. Therefore, the cost function and the likelihood function are defined respectively as:

$$D(\mathbf{X}; d) = \sum_{\mathbf{Y} \in \mathbf{W_X}} \frac{(T_t - I_t(\mathbf{Y} + d))^2}{|\mathbf{W_X}|} \quad (9)$$

$$(Y_t^a, \theta_t^B | \theta_t^F) \propto \exp\left(-\frac{D(\mathbf{X}; d)}{2\sigma_1^2}\right) \quad (10)$$

where Y^a represents the appearance cue, T_t is the template at time t, I_t is the original image at time t.

3.3. Adaptive Fusion

One important factor that affects the performance of multi-cue integration is the fusion weight set containing weights assigned to each cue respectively. The values of weights reflect the contributions of the corresponding cues to the overall tracking system. We denote them as $\alpha_{m,t}$. They are determined by the information reliability of the different cues and can be quantified by the likelihood functions of them. Hence, the multi-cue integration can be formulated as a weighted product of the likelihood functions, the basic rule is that each cue is associated with a score based on the error between the individual one's saliency and the average saliency, using a self-organized dynamic equation.

Our strategy is: in each frame, adjust the fusion weights according to the current visual context, allowing them to react to changing situation, and propagate them to the next frame as updated fusion weights. Thus, the cues with less reliable information get suppressed and those with reliable information contribute more to the fusion process. So (6) is reformulated as:

$$\begin{aligned} \mathcal{L}(Y_t, \theta_t^B | \theta_t^F) &= \prod_{m=1}^{M} \mathcal{L}(Y_t^m, \theta_t^B | \theta_t^F)^{\alpha_{m,t}} \\ &\propto \prod_{m=1}^{M} \exp\left(-\frac{D^m(\mathbf{X}; d)}{2\sigma_1^2}\right)^{\alpha_{m,t}} \end{aligned} \quad (11)$$

The weights are determined as [7]; the error of each cue to the estimated target is computed, then converted to a score q_m between 0 and 1. Then these scores are normalized, and as follows:

$$\xi \dot{\alpha}_{m,t} = q_{m,t} - \alpha_{m,t} \quad (12)$$

$$\alpha_{m,t} = \frac{\xi}{\xi + 1} \alpha_{m,t-1} + \frac{1}{\xi + 1} q_{m,t} \quad (13)$$

$$q_{m,t} = \frac{\gamma_{m,t}}{\sum_{i=1}^{M} \gamma_{i,t}} \quad (14)$$

$$\gamma_{m,t} = \exp(-a \bar{E}_{m,t-1}) \quad (15)$$

$$\bar{E}_{m,t-1} = D(\mathbf{X}_{m,t-1}; d_{m,t-1}) \quad (16)$$

In figure 1 we show how the adaptive weights change for different scenes. Sequence 1 shows a slow-moving pedestrian in a high contrast video, therefore, the intensity information is dominant. Sequence 2 shows a fast-moving pedestrian in a low-contrast video and as a result, the weights for the motion information are larger than those of the intensity information.

4. EXPERIMENT RESULTS

We applied our algorithm to different sets of outdoor surveillance video sequences, where most of the objects are moving people. Generally, the pedestrians are very small, occupying only 20-40

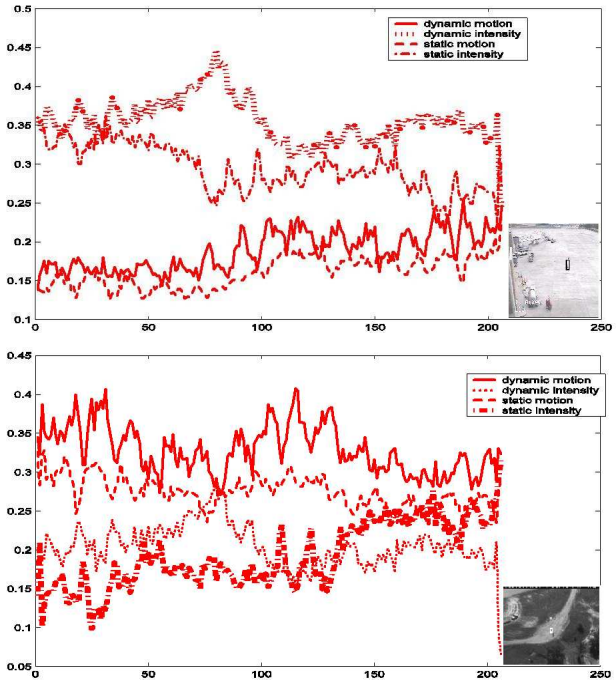

Fig. 1. Plots of adaptive weights plot for different information cues. The upper row shows a sequence containing slow-moving objects, where the weight plot shows a dominant intensity cue in the observation model; the bottom row shows a sequence containing a fast-moving object, where the weight plot reflects a strong effect from motion cues to the observation model. "dynamic" means the likelihood function is estimated with respect to a dynamic template while "static" means the likelihood function is estimated with respect to a static template.

pixels. It is also noticeable that some of the targets show a very poor intensity contrast. Some of the tracking results on different scenes are shown in Figure 2 . In these images, bounding boxes indicate target locations. The number of background motion particles is 150 with a Gaussian distribution as the proposal distribution, the number of foreground displacement particles is 25 with $[-2:2,-2:2]$ as the distribution space $\mathcal{D}_2$, the region window $\mathbf{W_X}$ is 5×5 , and the size of the derivative Gaussian filter used is 10×10.

5. CONCLUSION

We have addressed a surveillance tracking algorithm using the particle filter. The approach builds a robust motion model over a state-space of multiple-hypotheses for a moving object. The algorithm can simultaneously track both the background motion and the foreground motion, which improves the accuracy of the tracking result, especially in moving camera sequences; It constructs an integrated multi-cue observation model to make the system more robust. The experimental results demonstrate that the tracker reliably tracks multiple hypotheses, even under challenging conditions with low-contrast and small objects. We are now investigating its applications to several problems such as vehicle tracking and object classification.

Fig. 2. Tracking of pedestrians. The bounding box indicates the location of the target.

6. REFERENCES

[1] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion discontinuities. In *Proc. of IEEE International Conference on Computer Vision*, pages 551–558, Corfu, Greece, September 2000.

[2] A. Dempster, M. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc.*, B 39:1–38, 1977.

[3] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140:107–113, 1993.

[4] I. Haritaoglu, D. Harwood, and L. S. Davis. w^4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[5] M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. *Proc. of European Conference on Computer Vision, Freiburg, Germany , June 2-6, 1998*, pages 893–908, 1998.

[6] J. Shao, S. k. Zhou, and R. Chellppa. Simultaneous background and foreground modeling for tracking in surveillance video. In *IEEE Proc. on International Conference of Image Processing*, Singapore, Octobor 2004.

[7] M. Spengler and B. Schiele. Towards robust multi-cue integration for visual tracking. *Machine Vision and Application*, 14:50–58, 2003.

[8] H. Tao, H. Sawhney, and R. Kumar. Dynamic layer representation with applications to tracking. In *Proc. of Computer Vision and Pattern Recognition*, pages 532–539, 2000.

[9] Q. Zheng and R. Chellappa. A computational vision approach to image registration. *IEEE Trans. Image Processing*, 2:311–326, 1993.

[10] S. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *Accepeted for IEEE Trans. Image Processing*, 2003.