Appearance Learning by Adaptive Kalman Filters for FLIR Tracking

Vijay Venkataraman, Guoliang Fan, Xin Fan
School of Electrical & Computer Engineering
Oklahoma State University, Stillwater, OK 74078
vvenka@okstate.edu, guoliang.fan@okstate.edu

Joseph P. Havlicek
School of Electrical & Computer Engineering
University of Oklahoma, Norman, OK 73019
joebob@ou.edu

Abstract

This paper addresses the challenging issue of target tracking and appearance learning in Forward Looking Infrared (FLIR) sequences. Tracking and appearance learning are formulated as a joint state estimation problem with two parallel inference processes. Specifically, a new adaptive Kalman filter is proposed to learn histogram-based target appearances. A particle filter is used to estimate the target position and size, where the learned appearance plays an important role. Our appearance learning algorithm is compared against two existing methods and experiments on the AMCOM FLIR dataset validate its effectiveness.

1. Introduction

Target tracking in Forward Looking Infrared (FLIR) image sequences is a challenging problem since FLIR images are often characterized by low signal-to-noise (SNR) ratios, strong ego-motion of the sensor, poor target visibility, and time varying target signatures, as shown in Fig. 1. Tracking failure can often be attributed to the deterioration of the appearance model, *viz*., the "drifting problem" [4]. Therefore, appearance modeling and learning are two key related issues that affect tracker accuracy and robustness. In [1, 17], the use of both foreground and local background information is shown to be beneficial for histogram-based appearance models. However, the tracker is prone to failure in cases with low background-foreground contrast if no adaptive updating scheme is established for the target appearance model. The example in Fig.1 shows several sample FLIR frames and the evolution of the intensity histograms of the target and local background area over time. The strong similarity between the foreground and background histograms renders the target barely distinguishable from the background. In such cases, increasing numbers of background pixels may creep into the target appearance model over time, eventually causing track loss. Therefore, it is essential to have an adaptive scheme which learns the target appearance "on-the-fly."

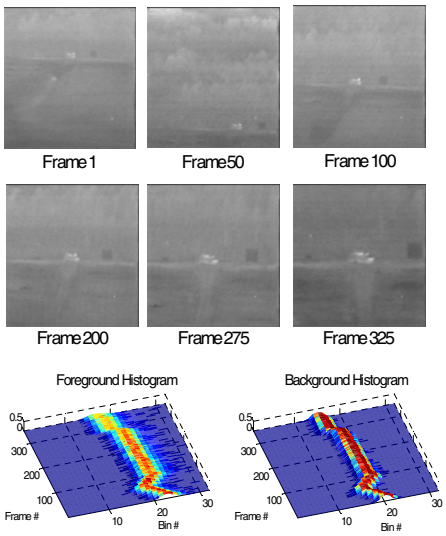

Figure 1. Sample FLIR frames and the variation of foreground and background intensity histograms for the sequence LW-17-01.

Significant efforts have been directed towards developing methods for online appearance learning. These methods often depend on the choice of the descriptive features. For histogram-based representations, there are two methods for appearance learning. The first combines the reference model with the current observation via a linear weighting scheme [22]. It is simple and straightforward, but highly susceptible to the "drifting problem." The second formulates appearance learning as a state estimation problem, where each histogram bin is treated as a linear system state and filtered by an Adaptive Kalman filter (AKF) [14]. Although this method is more robust, its applicability is limited by the potentially ill-conditioned estimation of system noise parameters which is key to the AKF. In this paper, we propose a new AKF-based appearance learning method where the the system noise is estimated via the time-varying Autocovariance Least-Squares (ALS) technique [12], which provides a well-conditioned and stable solution. Joint tracking and learning is formulated as a unified probabilistic inference problem, where two state estimation processes are integrated into a graphical model.

978-1-4244-3993-5/09/$25.00 ©2009 IEEE

Authorized licensed use limited to: University of Oklahoma Libraries. Downloaded on January 26, 2010 at 22:29 from IEEE Xplore. Restrictions apply.

2. Related Work

In the context of infrared target tracking, the use of a simple rectangle [3, 15, 18] is often preferred over the use of contours [20] to describe the target shape. The features commonly used for appearance modelling in FLIR sequences include simple shapes, edges [15] and local statistics like intensity and standard deviation (stdev) [3, 21] of the target area. The use of stdev as a feature helps greatly in localizing both small and dark targets. The lack of color information in IR images prohibits the use of color features as in [2]. Because they are scale invariant and tend to be slowly varying, intensity histograms are widely used for target representation [2, 7, 21]. Here, we employ a dual foreground-background appearance model [17] which incorporates simple pixel statistics (intensity and local stdev) from both the target and the surrounding background.

The value of appearance learning in tracking is well recognized. Generally, the appearance learning/update method is strongly influenced by the choice of features which characterize the appearance model. In the case of templates, a drift correction approach is proposed in [8]. Templates, however, cannot handle appearance variations and view changes of the target. In [5], a more sophisticated model that involves three components (stable, wandering and outlier) is proposed. These three components are combined into a Gaussian mixture model with parameters that are updated using an EM algorithm. For small targets, there may not be enough pixels on the target to support meaningful estimation of the elaborate parameter set.

Kalman filters have traditionally been used in target tracking where the objective is to estimate target kinematics (positions and velocities). They have also been used recently for appearance learning. A Kalman-based approach was proposed to update pixel values of the target template in [11]. This idea was extended to histogram-based appearance modeling in [14]. However, Kalman filtering requires complete knowledge of the system model, including the statistics of the system noises. Designing optimal filters without knowledge of the noise components is a well studied topic in the field of control systems where it is referred to as AKF, [6, 9]. Two of the most popular AKF algorithms are covariance matching and autocorrelation based methods, which will be discussed and compared in this paper. These methods are based on the principle of deriving a set of constraints that relate the covariance or autocorrelation of the filter residues with the unknown noise parameters. The covariance method seeks to obtain filter residuals that are consistent with the theoretical covariances by adjusting the noise parameters. This is based on a single constraint that does not guarantee a feasible solution. Alternatively, the ALS method, which will be our main focus here, is robust in the sense of involving multiple autocorrelation constraints at different time lags.

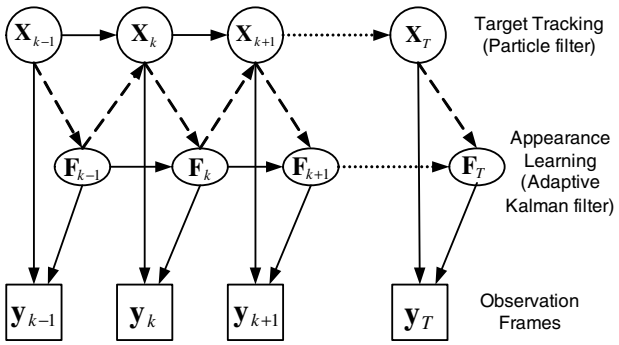

Figure 2. Framework of the proposed algorithm

3. Problem Formulation

The proposed tracking algorithm along with appearance learning is shown in Fig. 2, where two state estimation problems are involved. Let $\mathbf{x}_k$ and $\mathbf{F}_k$ represent the unknown kinematics (position and size) and appearance model at time step k and let $\mathbf{y}_k$ represent the kth observed video frame from which we want to infer $\mathbf{x}_k$ and $\mathbf{F}_k$. The goal of the inference is to estimate the posterior densities $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ and $p(\mathbf{F}_k|\mathbf{y}_{1:k})$. Formulating the conditional dependencies of Fig. 2 in a recursive Bayesian framework, we have

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \propto \int_{\mathbf{F}_{k-1}} p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{F}_{k-1}) p(\mathbf{F}_{k-1}|\mathbf{y}_{1:k-1}) \mathrm{d}\mathbf{F}_{k-1} \cdot \int_{\mathbf{x}_{k-1}} p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \mathrm{d}\mathbf{x}_{k-1}, \quad (1)$$

$$p(\mathbf{F}_k|\mathbf{y}_{1:k}) \propto \int_{\mathbf{x}_k} p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{F}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k}) \mathrm{d}\mathbf{x}_k \cdot \int_{\mathbf{F}_{k-1}} p(\mathbf{F}_k|\mathbf{F}_{k-1}) p(\mathbf{F}_{k-1}|\mathbf{y}_{1:k-1}) \mathrm{d}\mathbf{F}_{k-1}. \quad (2)$$

Similar to the co-inference algorithm in [19], we substitute the first integrals on $\mathbf{F}_{k-1}$ and $\mathbf{x}_k$ in (1) and (2) with their expectations $\hat{\mathbf{F}}_{k-1}$ and $\hat{\mathbf{x}}_k$. Thus we have

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \propto p(\mathbf{y}_k|\mathbf{x}_k, \hat{\mathbf{F}}_{k-1}) \cdot \int_{\mathbf{x}_{k-1}} p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \mathrm{d}\mathbf{x}_{k-1}, \quad (3)$$

$$p(\mathbf{F}_k|\mathbf{y}_{1:k}) \propto p(\mathbf{y}_k|\mathbf{F}_k, \hat{\mathbf{x}}_k) \cdot \int_{\mathbf{F}_{k-1}} p(\mathbf{F}_k|\mathbf{F}_{k-1}) p(\mathbf{F}_{k-1}|\mathbf{y}_{1:k-1}) \mathrm{d}\mathbf{F}_{k-1}. \quad (4)$$

Here, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ represents the kinematics evolution and $p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{F}_{k-1})$ is the observation likelihood given the kinematics and appearance model. Because (4) is nonlinear, we approximate it using a particle filtering approach in Section 5. Under the assumption that the appearance histograms evolve linearly, we introduce an AKF in Section 4 to estimate (4). The particle filter uses the appearance model $\mathbf{F}_{k-1}$ to localize the target at time step k. In turn, the appearance model is updated to $\mathbf{F}_k$ using information from the tracker output at time step k, as shown in Fig. 2.

Authorized licensed use limited to: University of Oklahoma Libraries. Downloaded on January 26, 2010 at 22:29 from IEEE Xplore. Restrictions apply.

4. Histogram-based Appearance Learning

In this section, we present three appearance learning techniques. Let the histograms of the target appearance and of the track gate at time k be given by

$$\begin{aligned}&\text{appearance model}: \ \mathbf{f}_k = \{f_k^b\}_{b=1\cdots N_b};\ \textstyle\sum_{b=1}^{N_b} f_k^b = 1,\\ &\text{tracker hypothesis}: \ \mathbf{g}_k = \{g_k^b\}_{b=1\cdots N_b};\ \textstyle\sum_{b=1}^{N_b} g_k^b = 1,\end{aligned}$$

where N_b is the number of bins of the appearance histogram. Our objective is to learn the one-step future appearance model $\mathbf{f}_{k+1}$ by incorporating the current tracker information $\mathbf{g}_k$ into the present appearance model $\mathbf{f}_k$.

4.1. Linear Combination Method

The linear combination method is based on the following:

$$\mathbf{f}_k = \xi_k \mathbf{f}_{k-1} + (1-\xi_k)\mathbf{g}_k, \tag{5}$$

where ξ_k is defined by

$$\xi_k = \mathrm{d}(\mathbf{f}_{k-1}, \mathbf{g}_k), \tag{6}$$

$0 \leq \xi_k \leq 1$, and where d is a distance function. A commonly used distance measure is the Bhattacharyya Coefficient (BC) [2, 7]. However, we have found that the histogram intersection metric [16] is more suitable for the problem considered here. When the two histograms are very similar (large ξ_k) very little information from the tracker hypothesis is incorporated in the learning step. However, when there is a sudden change in target appearance and the two histograms become less similar (small ξ_k), the new appearance is quickly incorporated into the model. This rapid adaptation of the tracker hypothesis can be a disadvantage and lead to appearance drift if the image is corrupted by noise or if the tracker is distracted by similar looking targets or background.

4.2. Adaptive Kalman Filtering

The Kalman filter is a linear method where the filter coefficients are optimal in the MSE sense under appropriate assumptions. For Kalman filter-based histogram appearance learning, the state and observation models for the bth bin are given by

$$f_k^b = f_{k-1}^b + w_{k-1}^b, \tag{7}$$
$$g_k^b = f_k^b + v_k^b, \tag{8}$$

where w_{k-1}^b and v_k^b are the system and observation noises that are assumed to be zero mean IID Gaussian with variances σ_{wb}^2 and σ_{vb}^2 respectively. In (7) and (8), it is assumed that both the histogram evolution process and its observation are driven by white noise alone. The state prediction and update equations for the above system based on the Kalman Filter are given by:

$$\text{State prediction:} \quad f_{k|k-1}^b = f_{k-1}^b. \tag{9}$$
$$\text{Covariance prediction:} \quad p_{k|k-1}^b = p_{k-1}^b + \sigma_{wb}^2. \tag{10}$$
$$\text{Compute Kalman gain:} \quad K_k^b = \frac{p_{k|k-1}^b}{p_{k|k-1}^b + \sigma_{vb}^2}. \tag{11}$$
$$\text{Compute residue:} \quad r_k^b = g_k^b - f_{k|k-1}^b. \tag{12}$$
$$\text{State update:} \quad f_k^b = f_{k|k-1}^b + K_k^b r_k^b. \tag{13}$$
$$\text{Covariance update:} \quad p_k^b = (1-K_k^b) p_{k|k-1}^b. \tag{14}$$

Computation of the optimal Kalman gains in (9) - (14) requires knowledge of the variances σ_{wb}^2 and σ_{vb}^2. Estimation of these unknown variances is the main objective of adaptive Kalman filtering. In the following, we will review the application of covariance and autocovariance methods for appearance histogram learning.

4.3. AKF: Covariance matching

Covariance matching methods are based on making the filter residuals, computed in (12), consistent with their theoretical values by adjusting the noise parameters. Assuming all bins share the same noise statistics (*e.g.*, $\sigma_{vb}^2 = \sigma_v^2$ and $\sigma_{wb}^2 = \sigma_w^2 \ \forall\ b$), the theoretical value of the residual covariance for the system defined in (7) and (8) is given by [9]

$$E[r_k r_k^T] = p_{k|k-1} + \sigma_v^2 = p_{k-1} + \sigma_w^2 + \sigma_v^2. \tag{15}$$

The sample based covariance of the filter residues r_k is computed using the residual values of all bins over the last L frames according to

$$E[r_k r_k^T] = \frac{1}{LN_b} \sum_{l=k-L+1}^{k} \sum_{b=1}^{N_b} (r_l^b)^2. \tag{16}$$

The error covariance p_{k-1} is estimated by

$$p_{k-1} = \frac{1}{N_b} \sum_{b=1}^{N_b} p_{k-1}^b. \tag{17}$$

Note that (15) involves both of the unknown noise variances σ_v^2 and σ_w^2 and can therefore be used to solve for one of them only if the other is known. If σ_v^2, $E[r_k r_k^T]$, and p_{k-1} are known, for example, then σ_w^2 can be determined using (15). Many recent works [11, 13, 14] employ this method to update the filter noise covariances. In [13], the authors attribute the observation noise to the precision of the template transformation parameters and obtain an expression to explicitly evaluate it. In [11] and [14], the observation noise σ_v^2 is initialized in the first frame and then held constant for the rest of the tracking process.

Authorized licensed use limited to: University of Oklahoma Libraries. Downloaded on January 26, 2010 at 22:29 from IEEE Xplore. Restrictions apply.

The covariance matching method makes two important but potentially problematic assumptions: (1) all histogram bins share same noise statistics, and (2) the measurement noise is a known constant. In addition, there is no guarantee to ensure that the estimates of (15) always result in positive values for σ_w^2, the process noise variance. Since the term p_{k-1} computed in (15) is only an approximation of the actual error covariance, convergence of σ_w^2 to the optimal value cannot be guaranteed.

4.4. Autocovariance based Least Squares (ALS)

Autocovariance based methods typically derive a set of equations that relate the residual autocorrelations at different lags with the unknown noise statistics. Pioneering work in this field was done by Mehra [9], who first proposed the use of residual autocorrelation for adaptive filtering. Neethling and Young [10] pointed out that Mehra's method yields estimates with large variances and does not consider the positive semidefinite (PSD) requirement of the unknown noise parameters. Recently, Odelson *et al.*[12] presented an Autocovariance Least Squares (ALS) method which estimates both the process and measurement noise covariances and ensures that they are non-negative. In addition, estimates from the ALS method have lower variance in comparison to Mehra's method and converge asymptotically to the optimal value with increasing sample size.

We next present a brief overview of the ALS method as applicable to appearance histograms. Consider the state space model given by (7) and (8). The noises w_k^b and v_k^b are assumed to be statistically independent of each other and of the other bins. In the following, the superscript b is omitted for brevity. Given a random stable (suboptimal) Kalman filter gain K, the state estimates are given by

$$\hat{f}_{k+1} = \hat{f}_k + K(g_k - \hat{f}_k), \tag{18}$$

where the estimation error is defined by $\epsilon_k = f_k - \hat{f}_k$. The evolution of this error over time is given by

$$\epsilon_{k+1} = \overbrace{(1-K)}^{\bar{A}}\epsilon_k + \overbrace{\begin{bmatrix} 1 & -K \end{bmatrix}}^{\bar{G}}\begin{bmatrix} w_k \\ v_k \end{bmatrix}, \tag{19}$$

$$r_k = \epsilon_k + v_k, \tag{20}$$

where r_k is the residue defined as $r_k \triangleq g_k - \hat{f}_k$. Let $\mathscr{C}_j = E[r_k r_{k+j}^T]$ be the autocorrelation of the residues at lag j. Considering autocorrelations up to a lag of N and based on the derivation in [12], we obtain

$$\begin{bmatrix} 1 \\ \bar{A} \\ \vdots \\ \bar{A}^{N-1} \end{bmatrix} P + \begin{bmatrix} 1 \\ -K \\ \vdots \\ -\bar{A}^{N-2}K \end{bmatrix} \sigma_v^2 = \begin{bmatrix} \mathscr{C}_0 \\ \mathscr{C}_1 \\ \vdots \\ \mathscr{C}_{N-1} \end{bmatrix} \triangleq \mathscr{C}, \tag{21}$$

Predict bin value $f_{k|k-1} = f_{k-1}$.
Obtain observation g_k based on tracker output.
Compute residue $r_k = g_k - f_{k|k-1}$.
for $j = 0$ to $N_d - 1$
  Compute $\hat{\mathscr{C}}_j = \frac{1}{L-j}\sum_{i=k-L+1}^{k-j} r_i r_{i+j}$.
end
Setup and optimize ALS problem to obtain σ_v^2 and σ_w^2.
Compute steady state Kalman gain K based on the estimated noise parameters.
Update bin value $f_k = f_{k|k-1} + Kr_k$.

Table 1. Pseudo-code of the ALS based AKF for a single bin of the histogram at time k.

where P is the steady state error covariance matrix given by the solution of the Lyapunov equation

$$P = \bar{A}P\bar{A}^T + \bar{G}\begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}\bar{G}^T. \tag{22}$$

The expressions (21) and (22) form the core of the ALS method as it relates the autocorrelation $\mathscr{C}_j$ to the unknown noise covariances σ_w^2 and σ_v^2 embedded within P. The autocorrelations in the RHS of (21) can be approximated using the residues r_k computed from the filter according to

$$\hat{\mathscr{C}}_j = \frac{1}{N_d - j}\sum_{i=1}^{N_d-j} r_i r_{i+j}^T. \tag{23}$$

In (21), if we can substitute for P in terms of the unknown covariances σ_w^2 and σ_v^2, then an equivalency of the form $\mathscr{A}x = \hat{\mathscr{C}}$ can be obtained. In [12] this is done by applying the stacked form of equation (22). Here x represents the stacked vector of the unknown variances and $\hat{\mathscr{C}}$ represents the approximated autocorrelation estimates. $\hat{\mathscr{C}}$ is obtained from (23) using the residues computed in (12). The expression for $\mathscr{A}$ is defined in [12]. The Least Squares problem can now be expressed in the form

$$\Phi = \min_{\sigma_{\mathbf{w}}^2, \sigma_{\mathbf{v}}^2} \left\| \mathscr{A}\begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} - \hat{\mathscr{C}} \right\|^2 \text{ st: } \sigma_{\mathbf{w}}^{\mathbf{2}}, \sigma_{\mathbf{v}}^{\mathbf{2}} \geq 0. \tag{24}$$

The inequalities are handled by appending the objective with a logarithmic barrier function according to

$$\Phi = \min_{\sigma_{\mathbf{w}}^2, \sigma_{\mathbf{v}}^2} \left\| \mathscr{A}\begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} - \hat{\mathscr{C}} \right\|^2 - \mu \log(\sigma_w^2 \sigma_v^2), \tag{25}$$

where μ is the barrier parameter. This optimization has been shown to be convex and can be solved using a simple Newton recursion [12]. Unlike the covariance matching method, the ALS method (1) estimates both process and measurement noise parameters simultaneously, (2) computes noise statistics for each individual bin of the histogram, (3) enforces PSD on the estimated parameters, and (4) is based on multiple constraints that are obtained by considering the autocorrelation of the residues at different lags.

Authorized licensed use limited to: University of Oklahoma Libraries. Downloaded on January 26, 2010 at 22:29 from IEEE Xplore. Restrictions apply.

5. Particle Filter-based Tracking

A particle filter is used to estimate the target kinematics. The state vector at time step k is defined as $\mathbf{x}_k$=[$\mathrm{x}_k, \mathrm{s}_k$], where x_k=[x_k, y_k] contains the position information and s_k=[s_k^x, s_k^y] represents the target size. The position dynamics are based on the model in [17] that can handle strong ego-motion of the sensor platform. To account for size changes, we employ a simple model that can increase or decrease the size by 20% at each time step.

Given $\mathbf{x}_k$, the corresponding target appearance, denoted by $\mathbf{G}(\mathbf{x}_k)$, is composed of four histograms: the foreground intensity $f_{fi}(\mathbf{x}_k)$, background intensity $f_{bi}(\mathbf{x}_k)$, foreground stdev $f_{fs}(\mathbf{x}_k)$ and background stdev $f_{bs}(\mathbf{x}_k)$. These histograms are computed in a way similar to that described in [17]. Then $\mathbf{G}(\mathbf{x}_k)$ is defined as

$$\mathbf{G}(\mathbf{x}_k) = \{f_{fi}(\mathbf{x}_k), f_{bi}(\mathbf{x}_k), f_{fs}(\mathbf{x}_k), f_{bs}(\mathbf{x}_k)\}. \quad (26)$$

The Histogram Intersection (HI) metric is used to measure the similarity between any two histograms $\boldsymbol{p}$ and $\boldsymbol{q}$ as

$$\mathrm{d}(\boldsymbol{p}, \boldsymbol{q}) = \sum_{i=1}^{N_b} \min(\boldsymbol{p}(i), \boldsymbol{q}(i)). \quad (27)$$

The similarity between $\mathbf{G}(\mathbf{x}_k)$ and the reference model $\mathbf{F}_{k-1}$ which also comprises four histograms is defined as

$$\mathrm{D}(\mathbf{G}(\mathbf{x}_k), \mathbf{F}_{k-1}) = \sum_{z \in Z} \mathrm{d}(f_z(\mathbf{x}_k), f_{z,k-1}), \quad (28)$$

where $Z = \{fi, bi, fs, bs\}$. The implication of (28) is that all the four histograms are given equal weight in the tracking process. The likelihood $p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{v}_{k-1})$ is defined based on the distance measure in (28) and is given by

$$p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{F}_{k-1}) \propto \exp(\lambda \cdot \mathrm{D}(\mathbf{G}(\mathbf{x}_k^j), \mathbf{F}_{k-1})), \quad (29)$$

where λ is a constant controlling exponential non-linear stretching. The pseudo code for the particle filter based tracker is given in Table 2.

Initialization: Draw $\mathbf{x}_0^j \sim N(X_0, 1)$, and set $\mathbf{F}_0 = \mathbf{G}(X_0)$,
where X_0 is the ground truth of the states in the initial frame.
For k=1,$\cdots$,T
For j=1,$\cdots$, N_p
Draw $\mathbf{x}_k^j \sim p(\mathbf{x}_k^j|\mathbf{x}_{k-1}^j)$ using position and size dynamics.
Compute $w_k^j = \exp(\lambda \cdot \mathrm{D}(\mathbf{G}(\mathbf{x}_k^j), \mathbf{F}_{k-1}))$.
End
Normalize the weights such that $\sum_{j=1}^{N_p} w_k^j = 1$.
Compute the mean of the states $\hat{\mathbf{x}}_k$=$\sum_{j=1}^{N_p} w_k^j \mathbf{x}_k^j$.
Set $\mathbf{x}_k^j$ =resample($\mathbf{x}_k^j, w_k^j$).
Update reference model to obtain $\mathbf{F}_{k+1}$.
End

Table 2. Pseudo-code of the particle filter algorithm with online appearance learning for tracking in real video sequences.

6. Experimental Results

The proposed algorithm was evaluated on the AMCOM dataset that contains challenging range closure sequences in grayscale format (128×128 pixels). Ground truth information about the target position and size is available and serves as a benchmark for algorithm evaluation.

6.1. Experimental Setup

Three appearance learning algorithms were integrated with the same particle filter for performance evaluation. The LC algorithm used the linear combination method defined in (5) and (27). $\mathrm{AKF}_{\mathrm{cov}}$ used covariance matching as described in Section 4.3 to determine the unknown system noise parameters. In $\mathrm{AKF}_{\mathrm{als}}$, the unknowns were estimated using the autocorrelation method of Section 4.4. It is important to realize that both of the AKF methods result in filtering of the form (5) and differ only in the choice of ξ_k. The LC algorithm relies on appearance similarity, whereas the AKF methods considers system noise statistics in deciding the appropriate value ξ_k. The number of bins N_b for the intensity and stdev histograms were set to 32 and 16, respectively. The main purpose of the sdev histograms is to aid in localizing small and hard-to-see targets. The number of particles used for tracking was $N_p = 200$. The number of frames L used to compute the residual covariance for $\mathrm{AKF}_{\mathrm{cov}}$ in (16) was set to 3. Note that the $\mathrm{AKF}_{\mathrm{cov}}$ algorithm averages over the number of bins to make covariance estimates and would have $L \times N_b$ data points. However, the $\mathrm{AKF}_{\mathrm{als}}$ algorithm performs separate computation for each bin. To ensure a fair comparison between the two algorithms, more past frames (N_d=7) are included for autocorrelation estimation. The number of time lags was set to $N = 5$. Note that the appearance learning algorithms were applied only to the intensity histograms, as the dynamics of stdev histograms do not have a well-defined structure. The stdev histograms in all cases were updated using LC.

6.2. Discussion

The three algorithms were evaluated on the basis of both appearance learning (Fig.3) and tracking performance (Fig.4). In Fig.3 the ground-truth appearance of the target is shown (foreground histogram). It can easily be observed that the results of $\mathrm{AKF}_{\mathrm{als}}$ closely match that of the true target histogram. Closer examination reveals that the LC and $\mathrm{AKF}_{\mathrm{cov}}$ algorithms result in histograms that slowly deviate or "drift" from the true ones. This is clearly evident in Fig.3(c), where the intensity variation in the latter part of the sequence is not captured by the LC and $\mathrm{AKF}_{\mathrm{cov}}$ algorithms. Therefore, the tracker includes a large portion of the background into the tracking gate as seen in frames 320, 360 of Fig.4 (c). The LC and $\mathrm{AKF}_{\mathrm{cov}}$ algorithms show strong affinity in maintaining a single mode even when the origi-

Authorized licensed use limited to: University of Oklahoma Libraries. Downloaded on January 26, 2010 at 22:29 from IEEE Xplore. Restrictions apply.

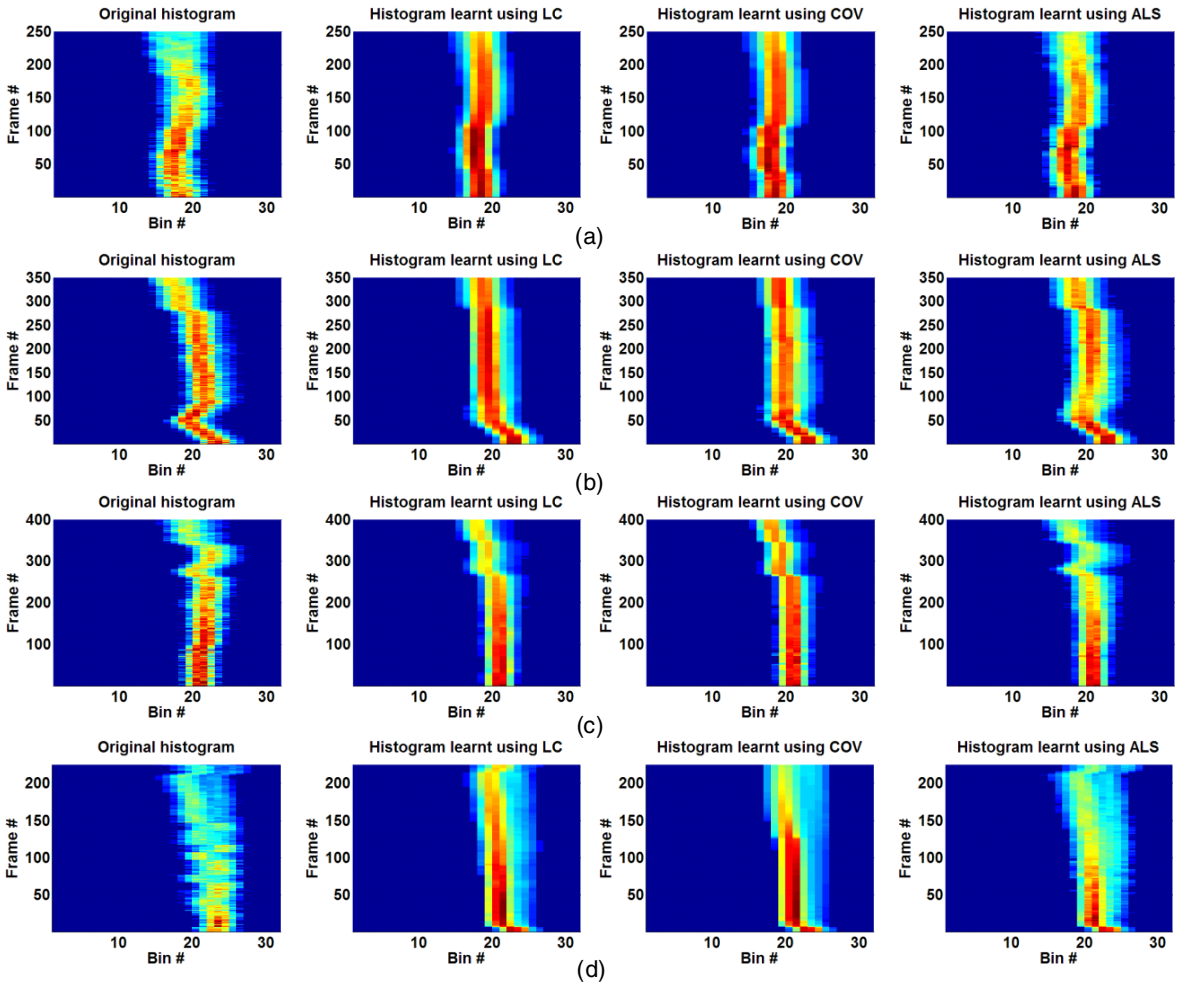

Figure 3. Comparison of appearance learning for four AMCOM sequences: (a) LW-15-NS (b) LW-17-01 (c) LW-21-15 and (d) LW-14-15.

Algorithm	LC				AKF_{cov}				AKF_{als}			
Sequence	x	y	s^x	s^y	x	y	s^x	s^y	x	y	s^x	s^y
LW-15-NS	1.019	1.817	1.906	2.732	0.860	1.511	1.644	2.396	**0.801**	**1.461**	**1.423**	**2.339**
LW-17-01	2.406	3.415	2.104	**3.016**	2.145	3.005	2.101	3.163	**1.213**	**2.110**	**1.376**	3.033
LW-21-15	0.970	1.653	**2.624**	2.941	1.135	1.812	2.799	3.113	**0.893**	**1.300**	2.786	**2.575**
LW-14-15	**0.889**	0.815	3.160	2.137	0.932	**0.787**	2.981	2.157	1.099	0.801	**2.660**	**1.787**
LW-19-06	1.977	1.545	1.566	1.544	0.797	0.764	1.681	1.454	**0.694**	**0.709**	**1.536**	**1.279**
Average	1.452	1.849	2.272	2.474	1.174	1.576	2.241	2.457	**0.940**	**1.276**	**1.956**	**2.202**

Table 3. Absolute error in estimated position and size. Averaged from 50 Monte Carlo runs for the three appearance learning algorithms.

nal histogram spreads slowly. This is illustrated in Fig.3 (a) and (d), the effect of this affinity is seen in Fig.4 (a) and (d) where the tracker is unable to estimate the target size accurately. Fig.3 and Fig.4 clearly indicate the positive recursive relationship between appearance learning and target tracking. In summary, LC easily corrupts the foreground appearance due to the drifting problem. The AKF_{cov} method, which assumes the same noise statistics for all histogram bins and estimates only the process noise without considering PSD conditions, resulting in a suboptimal Kalman gain estimate. Therefore its performance is only marginally better than that of LC. In contrast, the AKF_{als} algorithm, which estimates both process and measurement noise parameters with PSD conditions for each individual bin of the histograms, is able to follow the modes and variations of the original histogram accurately.

Authorized licensed use limited to: University of Oklahoma Libraries. Downloaded on January 26, 2010 at 22:29 from IEEE Xplore. Restrictions apply.

A detailed numerical comparison is shown in Table 3. It is observed that for most sequences $\mathrm{AKF}_{\mathrm{als}}$ produces the best results. The LC algorithm loses track of the target in the sequence LW-19-06 (2 runs) as indicated by large errors. The largest improvement in localization using $\mathrm{AKF}_{\mathrm{als}}$ is seen in the case of sequence LW-17-01, where the LC and $\mathrm{AKF}_{\mathrm{cov}}$ methods fail to capture the true histogram mode. This leads the tracker to lock-on to only a portion of the true target. In the last few frames of Fig.4 (b), the $\mathrm{AKF}_{\mathrm{als}}$ algorithm has difficulty in covering the whole target, since the left extreme of the target is very similar to the background.

7. Conclusion

We proposed an integrated framework for joint target tracking and appearance learning in FLIR sequences, where the problem was formulated as two interrelated state estimation processes. In particular, we presented a new AKF-based method which robustly learns histogram-based target appearances "on-the-fly." Experimental results on the AMCOM FLIR sequences show that the proposed technique significantly enhances the ability and robustness of appearance learning compared to two existing techniques, and consequently improves tracking performance.

References

[1] R. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.

[2] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[3] A. Dawoud, M. Alam, A. Bal, and C. Loo. Target tracking in infrared imagery using weighted composite reference function-based decision fusion. *IEEE Trans. on Image Processing*, 15(2):404–410, 2006.

[4] T. X. Han, M. Liu, and T. S. Huang. A drifting-proof framework for tracking and online appearance learning. In *Proc. the Eighth IEEE Workshop on Applications of Computer Vision*, 2007.

[5] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, 2003.

[6] X. Li and Y. Bar-Shalom. A recursive multiple model approach to noise identification. *IEEE Trans.on Aerospace and Electronic Systems*, 30(3):671–684, 1994.

[7] E. Maggio and A. Cavallaro. Hybrid particle filter and mean shift tracker with adaptive transition model. In *Proc. IEEE Intl Conf on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 221–224, 2005.

[8] L. Matthews, I. T., and S. Baker. The template update problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(6):810–815, 2004.

[9] R. Mehra. Approaches to adaptive filtering. *IEEE Trans.on Automatic Control*, 17(5):693–698, 1972.

[10] C. Neethling and P. Young. Comments on "identification of optimum filter steady-state gain for systems with unknown noise covariances". *IEEE Trans.on Automatic Control*, 19(5):623–625, 1974.

[11] H. Nguyen and A. Smeulders. Fast occluded object tracking by a robust appearance filter. *IEEE Trans.on Pattern Analysis and Machine Intelligence*, 26(8):1099–1104, 2004.

[12] B. Odelson, R. Rajamani, and B. Rawlings. A new autocovariance least-squares method for estimating noise covariances. *Automatica*, 42(2):303–308, 2006.

[13] J. Pan and B. Hu. Robust object tracking against template drift. In *IEEE Intl Conf on Image Processing*, pages 353–356, 2007.

[14] N. Peng, J. Yang, and Z. Liu. Mean shift blob tracking with kernel histogram filtering and hypothesis testing. *Pattern Recognition Letters*, 26(5):605–614, 2005.

[15] J. Shaik and K. Iftekharuddin. Automated tracking and classification of infrared images. In *Proc. Intl Joint Conf on Neural Networks*, volume 2, pages 1201–1206, 2003.

[16] M. Swain and D. Ballard. Indexing via color histograms. In *Proc. Intl Conf on Computer Vision*, pages 390–393, 1990.

[17] V. Venkataraman, G. Fan, and X. Fan. Target tracking with online feature selection in flir imagery. In *Proc. of IEEE Workshop on Object Tracking and Classification Beyond the Visible Spectrum (OTCBVS) (in conjunction with CVPR2007)*, pages 1–8, 2007.

[18] Z. Wang, Y. Wu, J. Wang, and H. Lu. Target tracking in infrared image sequences using diverse adaboostsvm. In *Proc. Intl Conf on Innovative Computing, Information and Control*, pages 233–236, 2006.

[19] Y. Wu and T. S. Huang. Robust visual tracking by integrating multiple cues based on co-inference learning. *International Journal of Computer Vision*, 58(1):55–71, 2004.

[20] A. Yilmaz, X. Li, and M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans.on Pattern Analysis and Machine Intelligence*, 26(11):1531–1536, 2004.

[21] A. Yilmaz, K. Shafique, and M. Shah. Tracking in airborne forward looking infrared imagery. *Image and Vision Computing*, 21(7):623–635, 2003.

[22] C. Zhang and Y. Rui. Robust visual tracking via pixel classification and integration. In *Proc. Intl Conf on Pattern Recognition (ICPR 2006)*, volume 3, pages 37–42, 2006.

Acknowledgments

The authors want to thank Prof. James B. Rawlings's research group for providing the ALS code.[1] They also acknowledge the anonymous reviewers for their constructive comments that have improved this paper. This work was supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under grants W911NF-04-1-0221 and W911NF-08-1-0293 and the 2009 Oklahoma NASA EPSCoR Research Initiation Grant (RIG).

[1] `http://jbrwww.che.wisc.edu/software/als/`

Authorized licensed use limited to: University of Oklahoma Libraries. Downloaded on January 26, 2010 at 22:29 from IEEE Xplore. Restrictions apply.

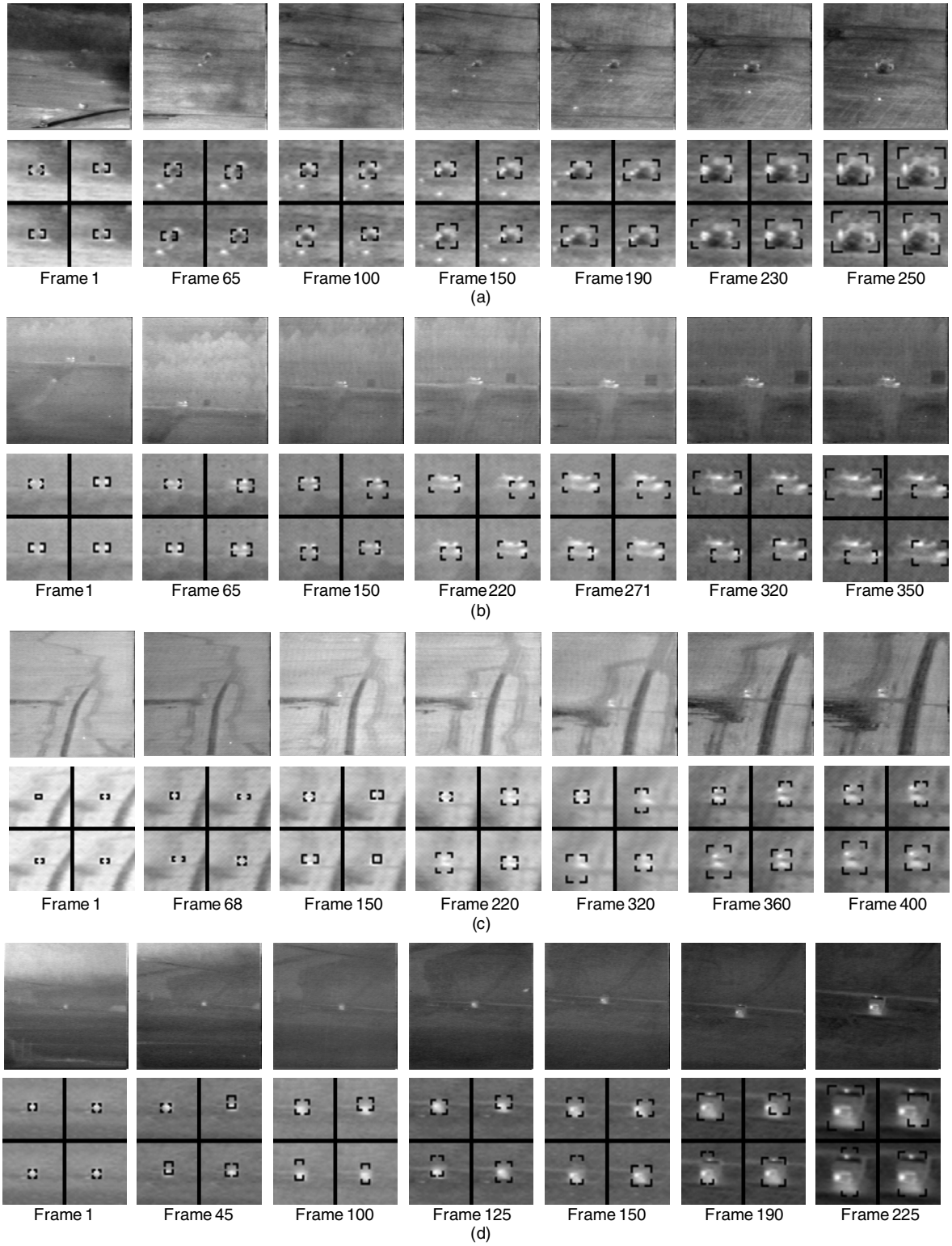

Figure 4. Tracking results of the three algorithm on four AMCOM sequences. The top row shows the sample frames of (a) LW-15-NS (b) LW-17-01 (c) LW-21-15 and (d) LW-14-15. The bottom row illustrates the tracking gates corresponding to the Ground truth (Top-Left), LC (Top-Right), $\mathrm{AKF}_{\mathrm{cov}}$ (Bottom-Left), $\mathrm{AKF}_{\mathrm{als}}$ (Bottom-Right).

Authorized licensed use limited to: University of Oklahoma Libraries. Downloaded on January 26, 2010 at 22:29 from IEEE Xplore. Restrictions apply.