Fast Occluded Object Tracking by a Robust Appearance Filter

Hieu T. Nguyen and
Arnold W.M. Smeulders, *Member*, *IEEE*

Abstract—We propose a new method for object tracking in image sequences using template matching. To update the template, appearance features are smoothed temporally by robust Kalman filters, one to each pixel. The resistance of the resulting template to partial occlusions enables the accurate detection and handling of more severe occlusions. Abrupt changes of lighting conditions can also be handled, especially when photometric invariant color features are used. The method has only a few parameters and is computationally fast enough to track objects in real time.

Index Terms—Object tracking, occlusions, appearance tracking, robust Kalman filter.

1 Introduction

THIS paper is concerned with tracking rigid objects in image sequences, using template matching. In essence, object tracking is the process of updating object attributes over time. The complete set of attributes include position, motion, shape, and appearance. The appearance is comprised of a set of photometric features representing the object region in a frame. To suppress noise and to achieve tracking stability, the attributes are smoothed by a temporal filter like the Kalman filter or Monte Carlo filters.

Many existing methods smooth the position and motion of the object only [1], [2], [3]. In such an approach, the benefit of the Kalman filter is just the smoothing of the object trajectory [1], [2]. Smoothing by a Monte Carlo filter is useful in dealing with the problem of background clutter as it allows for the tracking of multiple hypotheses [3]. In both cases, however, the temporal filter has no effect on improving the object localization, the main task of the tracking. The object has to be detected by an independent technique that may be based on edge detection [1], a domain specific method [3], or matching with a fixed template acquired from the first frame [4], [5]. Note that such an approach requires a reliable a priori object appearance model.

In the absence of an a priori object appearance model, such a model has to be learned on the fly during tracking. In such an approach, the algorithm needs a memory to store the learned appearance. This memory serves as a template for the object localization in the next frame. The template is normally initialized by the user and then updated over time. The updating is necessary because the object appearance can change. The fast updating scheme that acquires the template from the preceding frame [6], [2], [7], [8] will fail at the presence of occlusions or abrupt changes in lighting condition. To make the tracking robust to these factors, an appropriate temporal smoothing of appearance is needed. The smoothing should provide a more persistent template updating scheme which would be insensitive to sudden changes of the object appearance and, at the same time, be able to adapt to slow changes.

Based on our previous preliminary presentations [9], [10], this paper presents a new template updating algorithm that satisfies the two qualities: simplicity and robustness. Simplicity implies that the algorithm is easy to implement and has the minimum number of parameters. Robustness implies the ability of the algorithm to track objects under difficult conditions which include:

1. severe occlusions and lighting changes,
2. changing of object orientation or viewpoint,
3. background clutter and the presence of other moving objects in the scene,
4. a moving camera, and
5. nontranslational object motion like zooms and rotations.

To reduce complexity, we restrict ourselves to motion types where the deformation of the object shape in the image plane is described by a global transformation like the affine transformation. This assumption is justified for a large range of tracking applications where the object motion is rigid or the object is sufficiently distant from the camera. By doing so, we do not need a separate model for the object shape since this can be determined from the object motion.

The paper is structured as follows: Section 2 provides an overview of the related methods. Section 3 is the main section, which describes the template matching and the tracking of pixel appearance features using the Kalman filter. This section also presents a method for parameter tuning. The handling of severe occlusions is discussed in Section 4. Section 5 shows experimental results.

2 Related Work

Some recent methods for object tracking also smooth the object appearance [11], [12], [13], [14]. The feature type ranges from gray value of pixels [11], phase data from wavelet filters [13], to global statistics like mean color [12] or histogram [14].

The method of [11] proposes a MAP framework for tracking a set of attributes of video layers, including motion parameters, layer labels, and intensities. In particular, the intensity of each layer is updated using a weighted sum between the old template and the current observation data, with the weights being the posterior probabilities of layer labels. The method can be powerful for the segmentation of an entire video frame into moving layers, but is expensive for the tracking of a single object. Furthermore, we remark that, while there is a clear dependence between motion, shape, and location, there is very little dependence between those attributes and appearance. We can therefore separate the filtering of appearance from the filtering of motion and velocity to get a much simpler model formulation without losing much of tracking performance. This separation is actually the case for the other methods [12], [13], [14], [9].

The method in [12] uses a particle filter to track global statistics of object shape and color. Inserting color to the state of the particle filter yields robustness to background clutter and occlusions. Sampling in the state space, however, is rather expensive. In fact, the clutter problem can also be overcome by using a more discriminative appearance model, for example, the collection of pixel features as used in this paper.

While [11], [12] use a single Gaussian to model the object appearance, a more sophisticated model is used in [13]. The model is a mixture of three Gaussians. The first one corresponds to stable image structures. The second Gaussian takes care of changes in segmentation labels, whereas the third Gaussian allows for control over the filter adaptability. Although the mixture model gained in expressibility over the one Gaussian, the optimal state has to be estimated via an iterative EM algorithm, requiring many more computations, as is also the case for [14].

In [15], the Kalman filter is used to track the pose and pixel intensity of a image patch. The approach is somewhat similar to our previous work [9]. The essential difference, however, is that the paper does not address occlusions. In particular, while the standard Kalman filter is able to adapt the template to changes of object appearance, it is vulnerable to outliers caused by occlusions. Reference [15] also does not give a recipe for setting the parameters of the Kalman filter.

• *The authors are with the Intelligent Sensory Information Systems Group, University of Amsterdam, Faculty of Science, Kruislaan 403, NL-1098 SJ, Amsterdam, The Netherlands. E-mail: {tat, smeulders}@science.uva.nl.*

Manuscript received 16 Dec. 2002; revised 28 Aug. 2003; accepted 29 Dec. 2003. Recommended for acceptance by S. Soatto.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117973.

In general, the above models indeed offer flexibility for the tracker to cope with partial occlusions and sudden changes in illuminating conditions and object pose. The flexibility, however, is at the cost of more computationally expensive schemes for model estimation. The number of magic parameters is usually large. And, the methods in the references handle partial occlusions only, failing for severe or complete occlusions.

3 Tracking a Multivalue Template

3.1 The Appearance Model

The major idea of the presented approach is to keep track of an object appearance model defined as the collection of *photometric feature vectors* for pixels inside the target region. As the image region of rigid moving objects can be obtained from a fixed template region Ω via a coordinate transformation, it is convenient to map the feature vector at a point $\tilde{\mathbf{x}}$ in the target region at time t to the *template feature vector* $\boldsymbol{f}_t(\mathbf{x})$ defined for the corresponding point $\mathbf{x} \in \Omega$.

The components of the feature vector may be the RGB intensity values:

$$\boldsymbol{f}_t^{RGB}(\mathbf{x}) = [R_t(\mathbf{x}), G_t(\mathbf{x}), B_t(\mathbf{x})]^T. \tag{1}$$

If one wishes to make an intensity independent tracker, the color invariants proposed in [16] can be used. A fast implementation can be achieved as follows [17]:

$$\boldsymbol{f}^{c_1c_2c_3} = \left[\frac{R}{\max\{G,B\}}, \frac{G}{\max\{R,B\}}, \frac{B}{\max\{R,G\}}\right]^T. \tag{2}$$

In a special case, the vector can be reduced to a scalar being the pixel gray value:

$$\boldsymbol{f}^I = \frac{1}{3}[R+G+B]. \tag{3}$$

3.2 Temporal Estimation of Object Appearance Model

This section presents a robust Kalman filter for estimating the template feature vectors.

The feature vectors are tracked independently by individual temporal filters. As the temporal filter is described for one pixel, through the section we use $\boldsymbol{f}_t$ instead of $\boldsymbol{f}_t(\mathbf{x})$.

Following the Bayesian approach, the feature vector $\boldsymbol{f}_t$ is estimated by maximizing its posterior probability $p(\boldsymbol{f}_t|z_{1:t})$ conditioned on the history of measurements $z_{1:t} = \{z_1, \ldots, z_t\}$. As will be explained in Section 3.3, the measurement z_t is obtained by matching the recent template to the current image. Under the Markov assumption on the state process, the estimation is possible if the following two probabilistic models are given: the prediction model $p(\boldsymbol{f}_t|\boldsymbol{f}_{t-1})$ and the observation model $p(z_t|\boldsymbol{f}_t)$.

As the object photometry slowly changes over time, we assume that the predicted state remains unaltered apart from Gaussian noise:

$$p(\boldsymbol{f}_t|\boldsymbol{f}_{t-1}) \sim \mathcal{N}(\boldsymbol{f}_{t-1}, \boldsymbol{W}), \tag{4}$$

where $\mathcal{N}(\boldsymbol{f}_{t-1}, \boldsymbol{W})$ denotes the Gaussian distribution with the mean $\boldsymbol{f}_{t-1}$ and the covariance matrix $\boldsymbol{W}$. This matrix measures the fluctuation of appearance features, which depends mostly on camera and surface reflection noise as well as changes in appearance due to object movement with respect to the camera and the light source. As the object points have similar motion, $\boldsymbol{W}$ is set the same for all template pixels.

The observation model should be able to cope with outliers which are assumed to be caused by occlusion. In this case, the Gaussian observation model of the standard Kalman filter could lead to a wrong estimation of the template appearance. To limit the impact of the outliers we replace the square in the Gaussian distribution with a robust error norm. First, we define the measurement error as the Mahalanobis distance between the state and the measurement:

$$\epsilon(z_t, \boldsymbol{f}_t) = \sqrt{[z_t - \boldsymbol{f}_t]^T \boldsymbol{R}^{-1} [z_t - \boldsymbol{f}_t]}, \tag{5}$$

where $\boldsymbol{R}$ is the scale matrix. Again, we use the same $\boldsymbol{R}$ for all template pixels. Estimation of $\boldsymbol{R}$ will be discussed in Section 3.4. The observation model is defined as:

$$p(z_t|\boldsymbol{f}_t) = \kappa^{-1} |\boldsymbol{R}|^{-1/2} \exp\{-\rho(\epsilon(z_t, \boldsymbol{f}_t))\}, \tag{6}$$

where $\kappa = \int \exp\{-\rho(\sqrt{\tau^T\tau})\} d\tau$ is the normalization constant, and $|\boldsymbol{R}|$ denotes the determinant of $\boldsymbol{R}$. Here, ρ is the robust function. We use Huber's function:

$$\rho(\epsilon) = \begin{cases} \epsilon^2/2 & \text{if } |\epsilon| < c \\ c(|\epsilon| - c/2) & \text{otherwise,} \end{cases} \tag{7}$$

where c is the cutoff threshold. Since (6) is identical to the Gaussian distribution for $|\epsilon| < c$, we should set c to the value where the measurement distribution deviates from the Gaussian distribution. To do this, note that, if z_t was normally distributed, the squared norm ϵ^2 would have a chi-square distribution with d-degrees of freedom, where d is the dimensionality of z_t. Thus, we set $c = \sqrt{\chi^2_{d,\delta}}$, where $\chi^2_{d,\delta}$ is the δth quantile of the chi-square distribution with d degrees of freedom and δ is the level of significance, typically set to 0.99. If the measurement error exceeds this threshold, it is likely that z_t falls in the non-Gaussian part of the distribution and, therefore, the squared error needs to be replaced by the linear norm.

Having defined the two models, the posterior probability of $\boldsymbol{f}_t$ is calculated via the recursive formula:

$$\begin{aligned} p(\boldsymbol{f}_t|z_{1:t}) &\propto p(z_t|\boldsymbol{f}_t) p(\boldsymbol{f}_t|z_{1:t-1}) \\ &\propto p(z_t|\boldsymbol{f}_t) \int_{\boldsymbol{f}_{t-1}} p(\boldsymbol{f}_t|\boldsymbol{f}_{t-1}) p(\boldsymbol{f}_{t-1}|z_{1:t-1}). \end{aligned} \tag{8}$$

As the observation model $p(z_t|\boldsymbol{f}_t)$ is non-Gaussian, the standard Kalman filter no longer applies. Moreover, it is difficult to derive an analytical solution for $\boldsymbol{f}_t$. We present here an approximation solution by approximating $p(\boldsymbol{f}_t|z_{1:t})$ by a Gaussian, while preserving the non-Gaussian form of $p(z_t|\boldsymbol{f}_t)$. This results in a robust Kalman filter.

Suppose the previous output distribution $p(\boldsymbol{f}_{t-1}|z_{1:t-1})$ is approximated by a Gaussian with mean $\hat{\boldsymbol{f}}_{t-1}$ and covariance $\boldsymbol{C}_{t-1}$. The approximation is reasonable in most cases where $p(\boldsymbol{f}_{t-1}|z_{1:t-1})$ is unimodal. Then, the distribution of the predicted state has a Gaussian form also:

$$\begin{aligned} p(\boldsymbol{f}_t|z_{1:t-1}) &= \int_{\boldsymbol{f}_{t-1}} p(\boldsymbol{f}_t|\boldsymbol{f}_{t-1}) p(\boldsymbol{f}_{t-1}|z_{1:t-1}) \\ &= \mathcal{N}(\hat{\boldsymbol{f}}_{t-}, \boldsymbol{C}_{t-}), \end{aligned} \tag{9}$$

where

$$\hat{\boldsymbol{f}}_{t-} = \hat{\boldsymbol{f}}_{t-1} \qquad \text{and} \qquad \boldsymbol{C}_{t-} = \boldsymbol{C}_{t-1} + \boldsymbol{W}. \tag{10}$$

Substituting (6) and (9) into (8) yields:

$$\begin{aligned} p(\boldsymbol{f}_t|z_{1:t}) \propto & |\boldsymbol{R}|^{-1/2} |\boldsymbol{C}_{t-}|^{-1/2} \exp\{-\rho(\epsilon(z_t, \boldsymbol{f}_t)) \\ & + \frac{1}{2}(\boldsymbol{f}_t - \hat{\boldsymbol{f}}_{t-})^T \boldsymbol{C}_{t-}^{-1} (\boldsymbol{f}_t - \hat{\boldsymbol{f}}_{t-})\}. \end{aligned} \tag{11}$$

The optimal value of $\boldsymbol{f}_t$ is obtained from minimizing the function

$$S(\boldsymbol{f}_t) = \rho(\epsilon(z_t, \boldsymbol{f}_t)) + \frac{1}{2}(\boldsymbol{f}_t - \hat{\boldsymbol{f}}_{t-})^T \boldsymbol{C}_{t-}^{-1} (\boldsymbol{f}_t - \hat{\boldsymbol{f}}_{t-}) \tag{12}$$

over $\boldsymbol{f}_t$. Setting the derivatives of S to zero, we find the update equation for $\boldsymbol{f}_t$:

$$\hat{\boldsymbol{f}}_t = [\psi(\hat{\epsilon})\boldsymbol{R}^{-1} + \boldsymbol{C}_{t-}^{-1}]^{-1} [\psi(\hat{\epsilon})\boldsymbol{R}^{-1} z_t + \boldsymbol{C}_{t-}^{-1} \hat{\boldsymbol{f}}_{t-}], \tag{13}$$

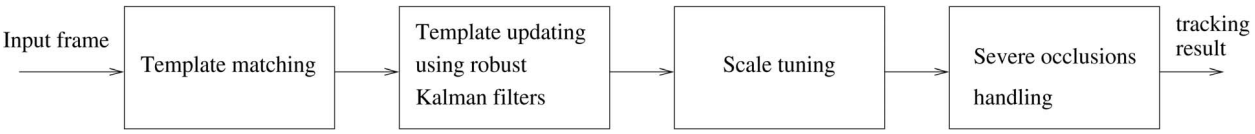

Fig. 1. The data flow diagram of one tracking iteration.

where $\hat{\epsilon} = \epsilon(z_t, \hat{\boldsymbol{f}}_t)$ and $\psi \equiv \rho'(\epsilon)/\epsilon$ is the influence function. Unlike the standard Kalman filter, (13) is to be iterated until stability. $\psi(\hat{\epsilon})$ is 1 if $|\hat{\epsilon}| \leq c$ and is $c/|\epsilon|$ if $|\hat{\epsilon}| > c$. So, if the measurement error ϵ is below the threshold, the standard Kalman equations are applied. Otherwise, the measurement is downweighted.

It remains to find a Gaussian approximation of $p(\boldsymbol{f}_t|z_{1:t})$, which will be used in the next tracking step. The mean of this Gaussian is $\hat{\boldsymbol{f}}_t$. An obvious choice of the covariance matrix is the inverse of the Hessian of S at $\hat{\boldsymbol{f}}_t$. The latter is approximated as $\psi(\hat{\epsilon})\boldsymbol{R}^{-1} + \boldsymbol{C}_{t-}^{-1}$, where the derivative of ψ is neglected. The update equation for the covariance matrix is:

$$\boldsymbol{C}_t = [\psi(\hat{\epsilon})\boldsymbol{R}^{-1} + \boldsymbol{C}_{t-}^{-1}]^{-1} = \psi(\hat{\epsilon})^{-1}\boldsymbol{R}[\psi(\hat{\epsilon})^{-1}\boldsymbol{R} + \boldsymbol{C}_{t-}]^{-1}\boldsymbol{C}_{t-}. \quad (14)$$

Equations (10), (13), and (14) in their order constitute the complete set of the update equations for the appearance tracking filter.

3.3 Template Matching

So far, we have not discussed how measurements are obtained. So, let us take a step back to the point before the appearance updating takes place. Given the collection of the predicted appearance feature vectors $\hat{\boldsymbol{f}}_{t-}(\mathbf{x})$, these vectors are matched to the current image in order to determine the measurement for the appearance filter at time t. Another goal of the matching is to locate the current object position.

Let φ be the transformation from the template region Ω to the object region and $\boldsymbol{a}_t$ be the parameter vector of this transformation at time t. $\mathbf{x}' = \varphi(\mathbf{x}; \boldsymbol{a}_t)$ then denotes the transformation of the point $\mathbf{x} \in \Omega$. We consider: translation, rotation, and scaling, so vector $\boldsymbol{a}$ has four components $a_1, \ldots, a_4$. The transformation equation can be written as:

$$\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = (1 + a_4)\begin{bmatrix} \cos a_3 & -\sin a_3 \\ \sin a_3 & \cos a_3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}. \quad (15)$$

The matching is performed by finding the image region that yields the maximal likelihood with respect to the distribution of $\boldsymbol{f}_{t-}$. This is achieved with the minimization:

$$\hat{\boldsymbol{a}}_t = \arg\min_{\boldsymbol{a}} \sum_{\mathbf{x}\in\Omega} \rho\Big(\epsilon\Big(\boldsymbol{I}_t(\varphi(\mathbf{x};\mathbf{a})), \hat{\boldsymbol{f}}_{t-}(\mathbf{x})\Big)\Big), \quad (16)$$

where $\boldsymbol{I}_t(\varphi(\mathbf{x};\mathbf{a}))$ is the feature vector observed in the current image at the point $\varphi(\mathbf{x};\mathbf{a})$. Target matching using robust error functions has been used in [4] and has been more heavily studied in [18]. The minimization is efficiently performed by the gradient descent algorithm in a coarse-to-fine manner. However, as this algorithm finds a local minimum only, it may result in a wrong location when the translation is large. To overcome this shortcoming, the gradient descent algorithm is repeated with a set of initial values of $\boldsymbol{a}$ with different translation vectors and the best resulting location is selected. Once $\hat{\boldsymbol{a}}_t$ has been obtained, the measurement of the appearance filter is defined as $z_t(\mathbf{x}) = \boldsymbol{I}_t(\varphi(\mathbf{x}; \hat{\mathbf{a}}_t))$.

Note that, to prevent sudden changes of $\mathbf{a}_t$ between successive frames, one could smooth $\mathbf{a}_t$ found in (16) by a Kalman filter together with the object velocity.

3.4 Scale Tuning

This section considers the setting of the scaling matrix $\boldsymbol{R}$. It is important to set the matrix properly since it decides whether a pixel is an outlier. Tuning $\boldsymbol{R}$ is necessary also because the matrix may vary with time and we should tune the filter accordingly. For tuning parameters of dynamical systems, powerful EM algorithms have been developed, see, for example, [19], [20]. However, these approaches are not efficient in online tracking as they require iteration over past frames.

We adopt a simpler method which is based on matching the covariances of the innovation $\boldsymbol{r}_t = z_t - \boldsymbol{f}_t$ with their prediction ([21], chapter 10). First, let us calculate the predicted distribution for the measurement z_t:

$$\begin{aligned} p(z_t|z_{1:t-1}) &= \int_{\boldsymbol{f}_t} p(z_t|\boldsymbol{f}_t)p(\boldsymbol{f}_t|z_{1:t-1}) \\ &\propto |\boldsymbol{R}|^{-1/2}|\boldsymbol{C}_{t-}|^{-1/2} \int_{\boldsymbol{f}_t} \exp\{-S(\boldsymbol{f}_t)\}, \end{aligned} \quad (17)$$

where S is given in (12). Again, we approximate this likelihood by a Gaussian. The Taylor expansion of S around $\hat{\boldsymbol{f}}_t$ allows us to integrate the quadratic part of $\boldsymbol{f}_t$ to obtain

$$\begin{aligned} p(z_t|z_{1:t-1}) \propto& \psi(\hat{\epsilon})^{-d/2}|\psi(\hat{\epsilon})^{-1}\boldsymbol{R} + \boldsymbol{C}_{t-}|^{-1/2} \times \\ & |\boldsymbol{C}_t|^{1/2} \exp\{-S(\hat{\boldsymbol{f}}_t)\}. \end{aligned} \quad (18)$$

Note that $S(\hat{\boldsymbol{f}}_t)$ depends on z_t. Let $G(z_t) = S(\hat{\boldsymbol{f}}_t)$. It can be shown that $G(z_t)$ attains its minimum at $z_t = \boldsymbol{f}_{t-}$ and the Hessian matrix of $G(z_t)$ is approximated by $\psi(\hat{\epsilon})^{-1}\boldsymbol{R} + \boldsymbol{C}_{t-}$. Therefore, (18) can be approximated by $\mathcal{N}(\boldsymbol{f}_{t-}, \psi(\epsilon)^{-1}\boldsymbol{R} + \boldsymbol{C}_{t-})$. A good parameter setting then should satisfy:

$$(z_t - \boldsymbol{f}_t^-)(z_t - \boldsymbol{f}_t^-)^T \approx \psi(\hat{\epsilon})^{-1}\boldsymbol{R} + \boldsymbol{C}_{t-}. \quad (19)$$

Since $\boldsymbol{C}_{t-} = \boldsymbol{C}_{t-1} + \boldsymbol{W}$, (19) relates the two matrices $\boldsymbol{R}$ and $\boldsymbol{W}$. Assuming that $\boldsymbol{W}$ is known, we can therefore adjust the matrix $\boldsymbol{R}$. Furthermore, the result is averaged over all pixels in the template region and over the last k frames, yielding:

$$\boldsymbol{R} \approx \frac{1}{kN} \sum_{i=t-k+1}^{k} \sum_{\boldsymbol{x}\in\Omega} \psi(\hat{\epsilon}(\boldsymbol{x}, i))[\boldsymbol{r}_i\boldsymbol{r}_i^T - \boldsymbol{C}_{i-}(\boldsymbol{x})] \quad (20)$$

Here, N is the number of pixels in the template. We have used $k = 25$ frames. Note that, in the first tracking steps, $\boldsymbol{R}$ should be made positive-definite as a reliable estimate of $\boldsymbol{C}_t$ is not yet available.

4 Severe Occlusion Handling

One major goal of our algorithm is to handle occlusions, the most unwanted events that often happen in video. The robust Kalman filter described in Section 3 makes the template resistant against

TABLE 1
Statistics of the Tracking Results for a Test Data Set

algorithm	A	B	C	D	E
type of template	Kalman-based template			template from	
	intensity	RGB	$c_1c_2c_3$	recent frame	first frame
lost tracks	6/25	4/25	6/25	15/25	22/25
missed occlusions	6/29	4/29	6/29	21/29	27/29
false occlusions	9	27	14	2	0

The data set consists of 25 clips containing 29 complete occlusions. The average duration for each clip is 150 frames.

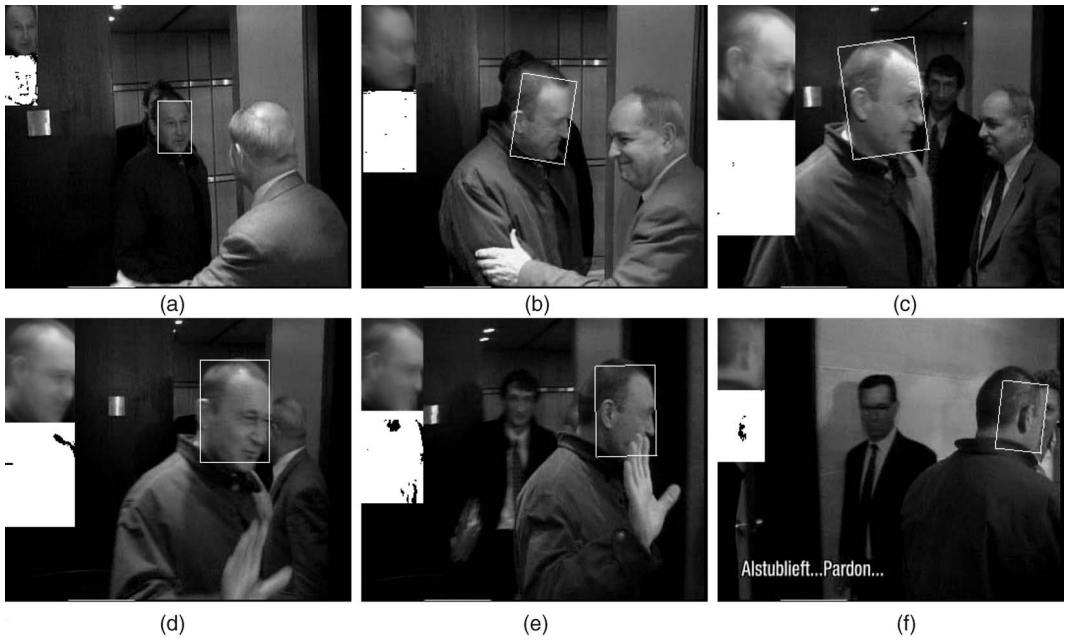

Fig. 2. Tracking results with several kinds of appearance changes. Gray value was used, (3). The current template is shown at the upper left corner and the outlier map is shown middle left. Outliers are indicated by black.

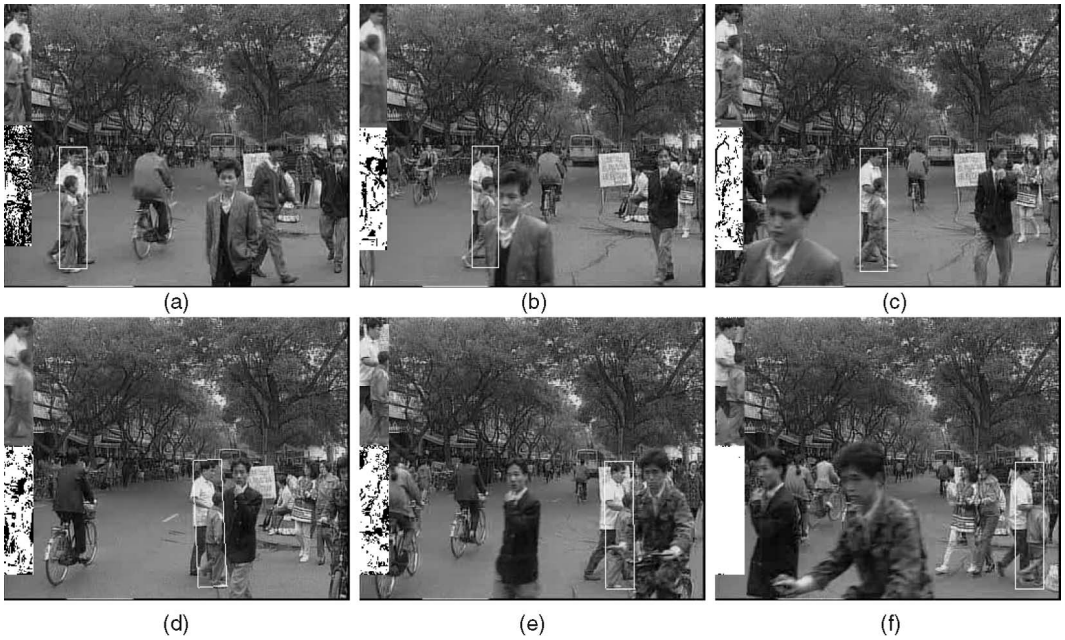

Fig. 3. Tracking results with a cluttered background and complete occlusions. RGB values were used, (1). Note the insensitivity of the template at the start of the occlusions.

short-time partial occlusions. It will, however, fail at severe and complete occlusions where the number of outliers is high. In this case, it is better to turn off the tracking for the entire template. A template pixel is regarded as outlier if the measurement error ε exceed the threshold c defined in Section 3.2. An occlusion is declared when the fraction of outliers exceeds a predefined percentage γ. During the occlusion, the template and parameters are not updated.

An important point is how to detect the end of the occlusion. For short-time complete occlusions, this relies on the assumption that the maximal duration of the occlusion is limited to L frames. Let t_o be the time the occlusion is detected. The template is then matched with the frames from t_o to $t_o + L$. The end of the occlusion is the frame, yielding the minimum cost in (16). Since the object appearance can change a lot after the occlusion, it is appropriate to reinitialize the template from the observation data, once the end of occlusion has been determined.

The proposed algorithm handles short-time occlusions. In general, handling long-time occlusions is difficult. When a partial occlusion lasts for a long time, the temporal filter will slowly accept the data at the occluded area. At a long time complete occlusion, the object can be recaptured only with more powerful tools for object recognition.

5 Experiments

The data flow diagram for one tracking iteration is shown in Fig. 1.

We have implemented the algorithm for three kinds of features: 1) image intensity as in (3), 2) the (R, G, B) vector as in (1), and 3) the color invariant features $c_1c_2c_3$ as in (2). These versions are named A,

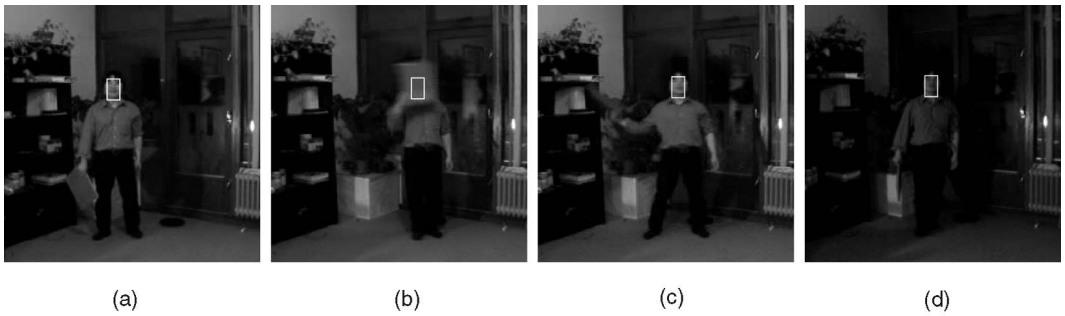

Fig. 4. Tracking results with occlusions and abrupt lighting changes. Color invariants were used, (2).

B, and C, respectively. They were applied for tracking objects in 25 video clips selected from TV news and movies. We deliberately selected clips that include the difficult conditions mentioned in the introduction, especially those with complete occlusions. For the comparison, we also used two other tracking algorithms. The algorithm D uses a template constructed from the most recent frame. The algorithm E on the contrary fixes the template as in the first frame of the sequence. The algorithms D and E localize the target by minimizing the sum of squared intensity differences between the template and image data. Occlusions are declared when the average residual exceeds 2.5 times the average residual over the last 25 frames.

The tracking results for the five algorithms were collected in Table 1. A lost track is declared when the target occupies less than 25 percent of the region found by the algorithm. As observed in the table, algorithms A, B, and C outperform the algorithms D and E regarding the number of lost tracks. Algorithm B that uses RGB appears to be the most successful despite the high number of falsely detected occlusions. In fact, false occlusion alarms are not a serious problem. They only slow the algorithm but they do not cause the tracking failure.

The method has only a few parameters, which are set as follows:

1. The state transition noise covariance $\boldsymbol{W} = 5\boldsymbol{I}$, where $\boldsymbol{I}$ denotes the identity matrix,
2. the maximal outlier percentage in partial occlusions $\gamma = 30\%$, and
3. the maximal duration of a severe occlusion $L = 25$ frames, which is sufficient for most videos we have.

Some results are shown in Figs. 2, 3, and 4 to provide insights to the tracking performance. Fig. 2 illustrates the tracking performance, including changes of object orientation and lighting, zooms, and partial occlusions. At the start, the target is in a dark area due to shadow. As the man enters the room, he turns left and shows his zoomed side-view. In addition, his face becomes strongly illuminated. The man still changes the orientation of his face a few more times. In the second half of the sequence, the lighting gradually decreases. The camera pans to the right. At some moment, the face is partially occluded by the hand. The algorithm E lost track in this sequence since the target appearance has totally changed in comparison with the first frame. The success of the Kalman-based templates was due to the ability to adapt to new appearances of the target. On the other hand, the Kalman filters do not accept outright the sudden changes due to the partial occlusion. This can be observed in Fig. 2e, where the hand is not included in the template but it appears in the outlier map.

Fig. 3 illustrates the tracking performance under the condition of a cluttered background and severe occlusions. The algorithm tracks two pedestrians crossing a street. The background is cluttered and contains many other moving objects with diverse motion directions. During the sequence, the objects are occluded completely three times. All the occlusions have been detected and handled successfully. Algorithm D lost track at the first occlusion due to the hasty adaptation.

Fig. 4 illustrates the tracking performance under abrupt changes of the lighting conditions. The sequence contains one occlusion and several abrupt changes of the lighting condition created intentionally by turning one of the light sources on and off. While algorithms A and B successfully detected and handled the occlusion, they lost track at the moment of the lighting change, see Fig. 4d. Algorithm C is the only one that succeeded in following the object till the end in this case. Note, however, that the color invariants have an inferior performance compared to RGB over the entire data set, as shown in Table 1. The reason is that the invariants throw away some information of object appearance.

6 Conclusion

This paper proposes a method for tracking objects in image sequences using template matching. It has revealed that tracking on object appearance rather than geometry is easier due to better identification power of appearance features. The multivalue appearance template is smoothed temporally by robust Kalman filters during tracking. In particular, outliers due to partial occlusions are downweighted by an observation model using a robust error norm and the Mahalanobis distance. The residual information is exploited to tune the scale parameters automatically. When photometric invariants are used, the method can achieve the insensitivity to shadow and abrupt changes of illumination conditions.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that have significantly improved Section 3. They would also like to thank Marcel Worring and Rein van den Boomgaard for their contribution at the early stage of this work.

References

[1] A. Blake, R. Curwen, and A. Zisserman, "A Framework for Spatio-Temporal Control in the Tracking of Visual Contour," *Int'l J. Computer Vision,* vol. 11, no. 2, pp. 127-145, 1993.
[2] N.P. Papanikolopoulos, P.K. Khosla, and T. Kanade, "Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision," *IEEE Trans. Robotics and Automation,* vol. 9, pp. 14-35, 1993.
[3] M. Isard and A. Blake, "CONDENSATION—Conditional Density Propagation for Visual Tracking," *Int'l J. Computer Vision,* vol. 29, no. 1, pp. 5-28, 1998.
[4] G.D. Hager and P.N. Belhumeur, "Efficient Region Tracking with Parametric Models of Geometry and Illumination," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 10, pp. 1025-1039, Oct. 1998.
[5] B. Li and R. Chellappa, "Simultaneous Tracking and Verification via Sequential Posterior Estimation," *Proc. Computer Vision and Pattern Recognition,* vol. 2, pp. 110-117, 2000.

[6] G.R. Legters Jr. and T.Y. Young, "A Mathematical Model for Computer Image Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 4, no. 6, pp. 583-594, June 1982.
[7] M.J. Black and Y. Yacoob, "Recognizing Facial Expressions in Image Sequences Using Local Parameterized Models of Image Motion," *Int'l J. Computer Vision,* vol. 25, no. 1, pp. 23-48, 1997.
[8] H. Sidenbladh, M.J. Black, and D.J. Fleet, "Stochastic Tracking of 3D Human Figures Using 2D Image Motion," *Proc. European Conf. Computer Vision,* vol. 2, pp. 702-718, 2000.
[9] H.T. Nguyen, M. Worring, and R. van den Boomgaard, "Occlusion Robust Adaptive Template Tracking," *Proc. Int'l Conf. Computer Vision,* vol. 1, pp. 678-683, 2001.
[10] H.T. Nguyen and A.W.M. Smeulders, "Template Tracking Using Color Invariant Pixel Features," *Proc. Int'l Conf. Image Processing,* vol. 1, pp. 569-572, 2002.
[11] H. Tao, H.S. Sawhney, and R. Kumar, "Dynamic Layer Representation with Applications to Tracking," *Proc. Computer Vision and Pattern Recognition,* vol. 2, pp. 134-141, 2000.
[12] Y. Wu and T.S. Huang, "A Co-Inference Approach to Robust Visual Tracking," *Proc. Int'l Conf. Computer Vision,* vol. 2, pp. 26-33, 2001.
[13] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi, "Robust Online Appearance Models for Visual Tracking," *Proc. Computer Vision and Pattern Recognition,* 2001.
[14] J. Vermaak, P. Perez, M. Gangnet, and A. Blake, "Towards Improved Observation Models for Visual Tracking: Selective Adaptation," *Proc. European Conf. Computer Vision,* vol. 1, pp. 645-660, 2002.
[15] F. Dellaert, S. Thrun, and C. Thorpe, "Jacobian Images of Superresolved Texture Maps for Model-Based Motion Estimation and Tracking," *Proc. Fourth Workshop Applications of Computer Vision,* 1998.
[16] J.M. Geusebroek, R. van den Boomgaard, A.W.M. Smeulders, and H. Geerts, "Color Invariance," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 12, pp. 1338-1350, Dec. 2001.
[17] T. Gevers and A.W.M. Smeulders, "Pictoseek: Combining Color and Shape Invariant Features for Image Retrieval," *IEEE Trans. Image Processing,* vol. 9, no. 1, p. 102, 2000.
[18] S. Baker, R. Gross, and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework: Part 3," Technical Report: CMU-RI-TR-03-35, Carnegie Mellon Univ., 2003.
[19] R.H. Shumway and D.S. Stoffer, "An Approach to Time Series Smoothing and Forecasting Using the em Algorithm," *J. Time Series Analysis,* vol. 3, no. 4, pp. 253-264, 1982.
[20] Z. Ghahramani and G.E. Hinton, "Parameter Estimation for Linear Dynamical Systems," Technical Report: CRG-TR-96-2, Univ. of Toronto 1996.
[21] P.S. Maybeck, *Stochastic Models, Estimation and Control,* vol. 2. New York: Academic Press, 1982.