TEMPLATE TRACKING USING COLOR INVARIANT PIXEL FEATURES

Hieu T. Nguyen and Arnold W. M. Smeulders

Intelligent Sensory Information Systems
University of Amsterdam, Faculty of Science
Kruislaan 403, NL-1098 SJ, Amsterdam, The Netherlands
Email: { tat,smeulder }@science.uva.nl

ABSTRACT

In our method for tracking objects, appearance features are smoothed by robust and adaptive Kalman filters, one to each pixel, making the method robust against occlusions. While the existing methods use only intensity to model the object appearance, our paper concentrates on multivalue features. Specifically, one option is to use photometric invariant color features, making the method robust to illumination effects such as shadow and object geometry. The method is able to track objects in real time.

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 the object attributes over time. To suppress noise and achieve tracking stability, the attributes are smoothed by a temporal filter like the Kalman filter or Monte-Carlo filters. In contrast to many early methods that smooth position, motion and shape of the object only, in recent years several researchers [1, 2, 3, 4] emphasize object appearance as important attributes to track. The temporal smoothing of object appearance enables the reliable detection of the object in new coming frames. In case of tracking rigid objects, the method of [4] has several advantages over the other methods in terms of robustness to occlusions, the automatic tuning of filter parameters and the implementation simplicity.

The existing methods use scalar features like grey value [1, 2, 4] or phase data [3] to describe the object appearance. Such features have a limited description power as they ignore the color information. Furthermore, the use of grey value suffers from the sensitivity to illumination change. The phase data [3] has some illumination independence but the application of this kind of feature is still limited due to its scalar nature. In this paper, based on the framework of [4], we aim to develop an algorithm for tracking color objects, which is insensitive to strong and abrupt illumination variations. To achieve this, the method referenced needs to be extended to cope with multivalue appearance features. Furthermore, in many cases the component features are highly correlated and the tracking algorithm should also take this into account.

2. TRACKING A MULTIVALUE TEMPLATE USING A ROBUST ADAPTIVE KALMAN FILTER

2.1. Template matching based tracking

Let $\Omega(t)$ be the image region, occupied by the tracked object at any time moment t. When the object is rigid, $\Omega(t)$ is obtained from a template region Ω_0 via a coordinate transformation $\varphi : \Omega_0 \mapsto \Omega(t)$ with a reasonable number of parameters. Examples are the translational, affine or quadratic transformations. This implies that every point $\mathbf{x} = (x, y)$ in the target region $\Omega(t)$ is obtained from a corresponding point $\mathbf{p} = (p_x, p_y)$ in the template Ω_0 as follows:

$$\mathbf{x} = \varphi(\mathbf{p}; \mathbf{a}(t)) \tag{1}$$

where $\mathbf{a}(t)$ denotes the parameter vector of the transformation, which is specific for $\Omega(t)$. This vector determines the position of the object in the current frame. The object motion is characterized by the deformation of $\Omega(t)$ between two consecutive frames, and can usually be modeled by the same type of transformation.

The object appearance is represented by the collection of *feature vectors* for pixels inside $\Omega(t)$. The components of such vectors may be RGB values or color invariants [5] at the pixel considered. Let d be the number of components. We therefore define for each point $\mathbf{p}$ in Ω_0 a *template feature vector* $\mathbf{g}(\mathbf{p}, t) \in \mathbb{R}^d$, which represents the image features at the corresponding point $\mathbf{x}$ given in eq. (1).

Let $\mathbf{f}(\mathbf{x}, t)$ denote the observed feature vector of pixel $\mathbf{x}$ at time t. The vector $\mathbf{a}(t)$ is estimated by matching the template $\mathbf{g}(\mathbf{p}, t')$, obtained at some earlier point in time $t' < t$, with the current image $\mathbf{f}(\mathbf{x}, t)$. Usually, the previous template is used, i.e. $t' = t - 1$. During an occlusion, t' is the moment where the occlusion is detected. Let

$$\mathbf{r}(\mathbf{p}) = \mathbf{f}(\varphi(\mathbf{p}; \mathbf{a}), t) - \mathbf{g}(\mathbf{p}, t') \tag{2}$$

This is the residual vector between the template value $\mathbf{g}(\mathbf{p}, t')$ and the observed data $\mathbf{f}(\varphi(\mathbf{p}; \mathbf{a}), t)$. The matching error at pixel $\mathbf{p}$ can be defined by the Mahalanobis distance: $\epsilon(\mathbf{p}) = \sqrt{\mathbf{r}(\mathbf{p})^\top \bar{\mathbf{R}}^{-1} \mathbf{r}(\mathbf{p})}$ where $\bar{\mathbf{R}}$ is the covariance matrix of the residual $\mathbf{r}(\mathbf{p})$. Furthermore, in order to make the matching robust against partial occlusions, we should downweight too large residuals, i.e. outliers. This is achieved by using a robust error norm $\rho(\epsilon)$ where ρ is a robust function. We use Huber's function, although other functions in [6] can be used as well:

$$\rho(\epsilon) = \begin{cases} \epsilon^2/2 & \text{if } |\epsilon| < c \\ c(|\epsilon| - c/2) & \text{otherwise} \end{cases} \quad (3)$$

where c is the cutoff threshold. Since the minimization of the matching error requires the differentiation of ρ, the bounded derivative of ρ effectively removes the influence of outliers to the minimization of eq.(4) to follow. Under the assumption that the residual $\mathbf{r}(\mathbf{p})$ has a normal distribution with zero mean, $\mathbf{r}(\mathbf{p})^\top \bar{\mathbf{R}}^{-1} \mathbf{r}(\mathbf{p})$ has a chi-square distribution with d- degrees of freedom. Thus, we can set $c = \sqrt{\chi^2_{d,\delta}}$, where $\chi^2_{d,\delta}$ is the $\delta - th$ quantile of the chi-square distribution with d degrees of freedom, and δ is the level of significance, typically set to 0.99.

Having defined the matching error for one pixel, the parameters $\mathbf{a}(t)$ is estimated by minimizing the total error over the template:

$$\mathbf{a}(t) = \arg\min_{\mathbf{a}} \sum_{\mathbf{p} \in \Omega_0} \rho\left(\sqrt{\mathbf{r}(\mathbf{p})^\top \bar{\mathbf{R}}^{-1} \mathbf{r}(\mathbf{p})}\right) \quad (4)$$

For the computational efficiency, we consider only the two kinds of motion, encountered most frequently in video: translation and scaling. $\mathbf{a}(t)$ is then found by exhaustive search in the quantized parameter space in a coarse-to-fine manner [4]. For the stability, the solution of eq.(4) is further smoothed by a Kalman filter together with the object velocity. This smoothing is standard and can be found in many traditional methods. See [7] for an example.

In conclusion, template matching is described by eq. (4), once methods for estimating image features $\mathbf{g}(\mathbf{p}, t)$ and residual covariances $\bar{\mathbf{R}}$ are given.

2.2. Kalman filter for tracking intensity

Following [4], the Kalman filter is employed to estimate $g(\mathbf{p}, t)$. We assume here that image features $\mathbf{g}(\mathbf{p}, t)$ for different pixels $\mathbf{p}$ are independent so that they can be tracked independently by individual Kalman filters.

The prediction and observation models for the filters are as follows:

$$\mathbf{g}(\mathbf{p}, t) = \mathbf{g}(\mathbf{p}, t-1) + \boldsymbol{\varepsilon}_w(\mathbf{p}, t) \quad (5)$$

$$\mathbf{f}(\varphi(\mathbf{p}; \mathbf{a}(t)), t) = \mathbf{g}(\mathbf{p}, t) + \boldsymbol{\varepsilon}_f(\mathbf{p}, t) \quad (6)$$

where $\mathbf{a}(t)$ is the result of eq. (4). Here, $\boldsymbol{\varepsilon}_w(\mathbf{p}, t)$ and $\boldsymbol{\varepsilon}_f(\mathbf{p}, t)$ denote the vectors of state noise and measurement noise respectively. $\boldsymbol{\varepsilon}_w$ models changes of object appearance due to factors such as change of the illumination condition or the object orientation, and $\boldsymbol{\varepsilon}_f$ models the noise in the image signal. As common in Kalman filtering, the two noise processes are assumed to be independent gaussians: $\boldsymbol{\varepsilon}_w(\mathbf{p}, t) \sim N(0, \mathbf{C}_w)$ and $\boldsymbol{\varepsilon}_f(\mathbf{p}, t) \sim N(0, \mathbf{C}_f)$. Furthermore, the covariance matrices $\mathbf{C}_w$ and $\mathbf{C}_f$ are assumed to be the same for all $\mathbf{p}$. This assumption is usually valid since all points $\mathbf{p}$ have a similar motion. Thus, all filters share the same parameters.

We now derive equations for the Kalman filters constructed from eq. (5) and (6). At $t = 0$, the template is boostraped from the observed data. We use $\mathbf{g}(\mathbf{p}, t^-)$ to denote the prediction of $\mathbf{g}(\mathbf{p}, t)$ at time t, reserving $\mathbf{g}(\mathbf{p}, t)$ for the estimate after the filter takes the current measurement $\mathbf{f}(\varphi(\mathbf{p}; \mathbf{a}(t)), t)$ into account. Let $\mathbf{C}_g(t^-)$ and $\mathbf{C}_g(t)$ be the covariance matrices of $\mathbf{g}(\mathbf{p}, t^-)$ and $\mathbf{g}(\mathbf{p}, t)$ respectively. Let $\mathbf{r}(\mathbf{p}, t)$ be the residual defined by eq. (2) with $t' = t - 1$. The template is updated as follows:

$$\mathbf{g}(\mathbf{p}, t^-) = \mathbf{g}(\mathbf{p}, t-1) \quad (7)$$

$$\mathbf{C}_g(t^-) = \mathbf{C}_g(t-1) + \mathbf{C}_w \quad (8)$$

$$\mathbf{K}(t) = \mathbf{C}_g(t^-)[\mathbf{C}_g(t^-) + \mathbf{C}_f]^{-1}$$

$$\mathbf{g}(\mathbf{p}, t) = \mathbf{g}(\mathbf{p}, t^-) + \mathbf{K}(t)\mathbf{r}(\mathbf{p}, t) \quad (9)$$

$$\mathbf{C}_g(t) = \mathbf{C}_g(t^-) - \mathbf{K}(t)\mathbf{C}_g(t^-) \quad (10)$$

Eq. (9) yields the optimal estimates for the template features $\mathbf{g}(\mathbf{p}, t)$, provided the residual is gaussian. In practice, this assumption is often violated due to occlusions or imperfections of the motion model used. To produce reliable feature estimates, template pixels with large residual should be removed from the filter state estimation.

Again, the criteria for outlier detection is based on checking whether the Mahalanobis distance $\mathbf{r}(\mathbf{p})^\top \bar{\mathbf{R}}^{-1} \mathbf{r}(\mathbf{p})$ exceeds a certain threshold. On the other hand, to prevent the possibility that $\mathbf{g}(\mathbf{p})$ may never be updated, we do not allow the algorithm to declare a pixel as outlier for long time. For each pixel $\mathbf{p}$, a counter $n_o(\mathbf{p})$ is introduced, that counts the number of successive frames where $\mathbf{p}$ is declared outlier. When $n_o(\mathbf{p})$ exceeds a maximally allowed value n_{omax}, the template value $\mathbf{g}(\mathbf{p})$ is re-bootstrapped from the observed value $\mathbf{f}(\varphi(\mathbf{p}; \mathbf{a}(t)))$. Thus, eq. (9) is replaced by:

$$\mathbf{g}(\mathbf{p}, t) = \begin{cases} \text{as in eq. (9) if } \mathbf{r}(\mathbf{p}, t)^\top \bar{\mathbf{R}}^{-1} \mathbf{r}(\mathbf{p}, t) < \chi^2_{d,\delta} \\ \mathbf{f}(\varphi(\mathbf{p}; \mathbf{a}(t)), t) \text{ if } n_o(\mathbf{p}) \geq n_{omax} \\ \mathbf{g}(\mathbf{p}, t^-) \qquad \text{otherwise} \end{cases} \quad (11)$$

From now on, whenever the updating of the template is mentioned, it refers to eq. (11).

The turning off of the tracking at outliers is useful not only for making the template insensitive against short-time and partial occlusions. It is also useful in case the template does not match exactly the object shape and contains also pixels from the background. In such a case, background pixels are treated as outliers

2.3. Adaptive filtering

This section considers the proper parameter settings.

The Kalman filter described requires the following parameters be known: the covariance matrix for the initial state $\mathbf{C}_g(0)$, for the state noise $\mathbf{C}_w$, and for the measurement noise $\mathbf{C}_f$. Among these, the matrices $\mathbf{C}_w$ and $\mathbf{C}_f$ are most critical. In practice, they are seldom known and not even constant in time. Therefore, one would like to estimate these parameters simultaneously with the states. We use the covariance matching method [8, p. 141] which suggests to compare the estimated variance of the residual with their theoretical variance.

Let Ω_0' be the subset of Ω_0 without outliers, and N' the number of pixels in Ω_0'. The covariance of the residuals is estimated by averaging $\mathbf{r}(\mathbf{p},t)\mathbf{r}(\mathbf{p},t)^\top$ over Ω_0' and over the last K frames:

$$\bar{\mathbf{R}} = \frac{1}{K}\sum_{i=t-K+1}^{t}\mathbf{R}(t) \tag{12}$$

where

$$\mathbf{R}(t) = \frac{1}{N'}\sum_{\mathbf{p}\in\Omega_0'}\mathbf{r}(\mathbf{p},t)\mathbf{r}(\mathbf{p},t)^\top \tag{13}$$

The matrix $\bar{\mathbf{R}}$, given by eq. (12), is used in eq. (4) and (11).

By comparing $\bar{\mathbf{R}}$ with the theoretical variance of $\mathbf{r}(\mathbf{p},t)$, which is $\mathbf{C}_g(t^-)+\mathbf{C}_f$, one of the two noise covariance matrices can be readjusted if the other one is known beforehand. Tuning one matrix is usually sufficient for the filter to adapt to changes of object orientation or illumination. Let us assume the measurement noise $\mathbf{C}_f$ is known, then the state noise is estimated as:

$$\mathbf{C}_w = \bar{\mathbf{R}} - \mathbf{C}_f - \mathbf{C}_g(t-1) \tag{14}$$

This re-estimation of $\mathbf{C}_w$ is especially useful when the object orientation or the illumination condition changes. In these cases, object appearance features change faster, leading to the increase of $\bar{\mathbf{R}}$, and hence, the increase of $\mathbf{C}_w$ as well. The higher value of $\mathbf{C}_w$ actually puts more weights for the observation data in the output of the Kalman filter, and therefore, keeps the template up-to-date with the object appearance.

It remains to specify $\mathbf{C}_f$ and the initial values for $\mathbf{C}_w$ and $\mathbf{C}_g$. They are set such that initially the states and measurements have equal weights:

$$\mathbf{C}_f = 0.5\mathbf{R}(1),\ \ \mathbf{C}_w(0) = 0 \ \text{ and } \ \mathbf{C}_g(0) = 0.5\mathbf{R}(1) \tag{15}$$

Using eq. (14) and (15), all noise parameters are set automatically.

2.4. Severe occlusion handling

The rejection of outliers, described in eq. (11), makes the template robust against short-time and partial occlusions. Severe occlusions are usually indicated by high number of outliers. In this case, it is better to turn off the tracking for the entire template. An occlusion is declared when the fraction of outliers exceeds a predefined percentage γ:

$$\frac{N-N'}{N} > \gamma \tag{16}$$

where N is the number of pixels in R, and as before, N' is the number of pixels in R'. During the occlusion, the template and parameters are not updated. Finding the end of the occlusion 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 (4). To save computations, we do not consider all L frames and skip frames with exponentially increasing steps. The typical sequence of frame numbers to visit is then 5,7,11,19,35 etc. The template is re-initialized from the new object features, once the end of occlusion has been determined.

There is a relation between γ and n_{omax} in eq. (11). n_{omax} must be large enough so that the template remains unaffected at first frames of the occlusion, where the fraction of outliers is still below γ. Thus, we set:

$$n_{omax} = \frac{\gamma}{\kappa} \tag{17}$$

where κ is the ratio of the minimal occlusion speed to the template width. We set $\kappa = 5\%$ and $\gamma = 25\%$. Hence, $n_{omax} = 5$.

3. EXPERIMENTS

We applied the presented method for tracking three kinds of features: image intensity $R+G+B$ as proposed in [4], the (R,G,B) vector, and the photometric features suggested by [5]. In the latter case features of a pixel are computed as:

$$c_1 = \frac{R}{\max\{G,B\}};\ c_2 = \frac{G}{\max\{B,R\}};\ c_3 = \frac{B}{\max\{R,G\}}; \tag{18}$$

where R,G,B are the usual color values. These features have been shown to be invariant to shadow and object geometry orientation with respect to camera while retaining intrinsic object properties [5].

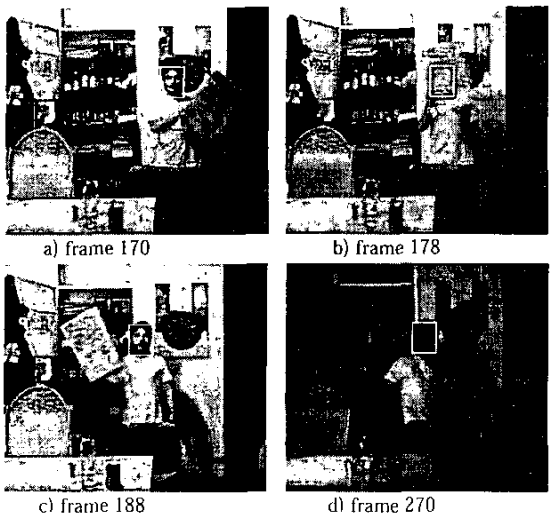

Fig. 1. *Tracking results using color invariants. a),b),c): with an complete occlusion. d) with an abrupt change of illumination, created by turning off one of the light sources.*

algorithm	number of clips where the tracking is successful	correctly detected occlusions
intensity	16	17
RGB	16	17
$c_1c_2c_3$	11	14

Table 1. *Tracking in movie. In total, there are 20 clips which contain 21 complete occlusions. The average duration for each clip is 150 frames.*

In the first experiment, the algorithms were tested for a video sequence created by ourselves. This sequence contains several complete occlusions and abrupt changes of illumination. The tracking result with the algorithm using color invariants is shown in Figure 1. While the algorithms using intensity and RGB lost track at the moment of the abrupt change of illumination, the tracker with color invariants vector (18) successfully tracked the object over the whole sequence.

In the second experiment, we tested the algorithms with several clips in an action movie. These clips contain many occlusions but not much abrupt illumination changes. The results are shown in Tab.1. As observed, while both the algorithms using intensity and RGB values exhibit a good and comparable performance, the algorithm using color invariants has an inferior performance. The reason is that the invariants throw away some information of object appearance. Further research is therefore needed to determine the criteria of switching to a specific feature type.

In our PC (Pentium II, 400 MHz) the average tracking time is approximately 0.005 seconds per frame, and hence, fast enough for real time applications.

4. CONCLUSION

This paper proposes a method for tracking rigid objects in image sequences using template matching. While shape and motion are smoothed in a similar manner as traditional methods, multivalue appearance features are smoothed independently by robust and adaptive Kalman filters, allowing for the accurate detection of the object. In particular, the rejection of outliers in observations using the Mahalanobis distance allows the efficient handling of occlusions. At the same time, the tracker can tune its parameters to adapt to changes of the object orientation or illumination conditions. The usefulness of the algorithm has been illustrated with the tracking of color invariants.

5. REFERENCES

[1] H. Tao, H.S. Sawhney, and R. Kumar, "Dynamic layer representation with applications to tracking," in *Proc. IEEE CVPR'2000*, pp. II:134–141.

[2] H. Sidenbladh, M.J. Black, and D.J. Fleet, "Stochastic tracking of 3D human figures using 2D image motion," in *Proc. of ECCV'2000*, pp. II:702–718.

[3] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi, "Robust online appearance models for visual tracking," in *Proc. IEEE CVPR'2001*.

[4] H.T. Nguyen, M. Worring, and R. van den Boomgaard, "Occlusion robust adaptive template tracking," in *Proc. IEEE Conf. on Computer Vision*, 2001, pp. I: 678–683.

[5] T. Gevers and A.W.M. Smeulders, "Pictoseek: Combining color and shape invariant features for image retrieval," *IEEE Trans. on Image Proc.*, vol. 9, no. 1, pp. 102, 2000.

[6] Z.Y. Zhang, "Parameter-estimation techniques: A tutorial with application to conic fitting," *Image and Vision Computing*, vol. 15, no. 1, pp. 59–76, January 1997.

[7] A. Blake, R. Curwen, and A. Zisserman, "A framework for spatio-temporal control in the tracking of visual contour," *Int. J. Computer Vision*, vol. 11, no. 2, pp. 127–145, 1993.

[8] P.S. Maybeck, *Stochastic models, estimation and control*, vol. 2, Academic Press, NewYork, 1982.