ROBUST OBJECT TRACKING AGAINST TEMPLATE DRIFT

Jiyan Pan, Student Member, IEEE, and Bo Hu, Member, IEEE

Department of Electronic Engineering, Fudan University, Shanghai 200433, China

ABSTRACT

We propose a new method addressing the problem of template drift, a common phenomenon in which the target gradually shifts away from the template in object tracking. Much effort has been devoted to this problem, but the results are not satisfactory enough due to the lack of quantitative analysis of its cause. In this paper, after carefully examining where template drift stems from and how it influences template update, we derive expressions that accurately evaluate the model noises of the Kalman appearance filter employed to update the template. The appearance filter therefore achieves an optimal balance between reducing template drift and keeping track of target appearance variations. We perform experiments on a wide range of real-world video sequences containing diverse degrees of target appearance variations. All the experimental results confirm the effectiveness of our algorithm.

Index Terms**—** Object tracking, template drift, template matching, adaptive Kalman filtering, noise evaluation.

1. INTRODUCTION

Object tracking has a wide range of applications in robotic control, visual surveillance, video retrieval, and homing technologies. Much research has been devoted to this field, and the algorithms based on template matching draw much attention [1]-[6]. In such algorithms, target is modeled by a template, and is tracked in a video sequence by matching candidate image regions with the template through coordinate transformations. The set of the transformation parameters that yields the best match with the template represents the geometric information of the target.

In order to take into account the changing target appearance, the template ought to be updated in one way or another. The most straightforward method which replaces the template every frame (or every *n* frames) with the image region believed to be the target [7], [8] is found to suffer from gradual drift of the target out of the template, eventually resulting in the loss of the target. This phenomenon is referred to as template drift [1], [9].

The cause of template drift has been *preliminarily* and *qualitatively* investigated in the literature [1]-[3], where template drift is ascribed to the accumulation of small errors introduced in the location of the template each time the template is updated. Faster updating of the template results in severer template drift, and a tricky situation is thus formed: frequent template renewal is required to keep the template up-to-date with the changing target appearance; on the other hand, hasty update of the template will damage the integrity of the template in face of drift errors. In order to obtain a good trade-off for the situation, template-updating strategies should be carefully designed.

Some methods utilize the first template extracted from the first frame as a benchmark for appropriate realignment and update of the current template [1], [2], [9]. These approaches eradicate template drift when tracking objects over a short period of time during which the appearance of the target remains almost the same. However, their resort to the first template is unreliable when the target appearance undergoes major changes.

After comparing various template update strategies *without* resorting to the first template, a conclusion is reached in [3] that Kalman filtering of the template shows the strongest robustness against drift and noise. The choice of Kalman gain for template filtering is further investigated in [6]. However, as the Kalman gain keeps fixed throughout the sequence, the Kalman template filter mentioned in [3] and [6] is unable to adjust its updating rate according to the degree of the target appearance variations and the extent of possible template drift.

Further improvements are made in [4] and [5] by allowing the Kalman gain to fluctuate according to how intensively the target appearance changes. This is achieved by estimating one of the two model noises in the Kalman filter online. Nevertheless, Reference [4] and [5] either assume the state transition noise or the measurement noise to be constant, which are not well justified. As a consequence, their performance of reducing template drift is still not satisfactory enough.

In order to properly estimate the model noises, the contribution that template drift makes towards the measurement noise should be *quantitatively* obtained. After a careful analysis of the cause of template drift, we identify a large component of the measurement noise as “drift noise” introduced by the precision limit of searching for the optimal coordinate transformation parameters. The drift noise is then acquired quantitatively by calculating the probability

This work is supported by National Basic Research Program 973 under Grant 2006CB705700.

distributions of the true values of the template pixels. Combining the estimation of camera noise, we therefore arrive at an expression to evaluate the measurement noise online, which is crucial in obtaining an optimal Kalman gain in the sense of optimally balancing between keeping in pace with the target appearance variations and preventing template drift.

The remainder of this paper is organized as follows. Section 2 gives a brief review of the use of template matching and Kalman appearance filtering in tracking objects. In Section 3, we elaborate on the quantitative analysis of the measurement noise and the role it plays in preventing template drift. Experimental results are shown in Section 4. Section 5 concludes this paper.

2. KALMAN APPEARANCE FILTERING IN TEMPLATE-MATCHING BASED TRACKING

The object (or target) to be tracked is characterized by an image called *template* that is generally initialized by extracting from the first frame of a video sequence. In subsequent frames, the estimated template $\hat{T}$ is mapped to the frame coordinate system by the coordinate transformation $\phi(\boldsymbol{x};\boldsymbol{a})$, where $\boldsymbol{a}$ is the transformation parameter vector. The location of the target in a certain frame n is determined by performing

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \frac{1}{N} \sum_{\boldsymbol{x}\in\boldsymbol{\Omega}_T} \left| I_n[\phi(\boldsymbol{x};\boldsymbol{a})] - \hat{T}(\boldsymbol{x}) \right|, \qquad (1)$$

where $\hat{\boldsymbol{a}}$ is the estimated transformation parameter vector, $I_n(\boldsymbol{x})$ is the grayscale of the pixels in frame n, $\boldsymbol{\Omega}_T$ represents the ensemble of the template pixels in the template coordinate system, and N is the number of pixels in the template. As ϕ might not necessarily generate integer coordinates, bilinear interpolation is employed here to calculate $I_n[\phi(\boldsymbol{x};\boldsymbol{a})]$.

Ideally, the $\hat{\boldsymbol{a}}$ found by (1) reflects the true geometric status of the target. However, as actual implementation of (1) is always conducted in a *discrete* vector space, the coordinate transformation parameter vector acquired by (1) is always somewhat different from its true value due to non-infinitesimal searching steps. As a consequence, small deviation of the *measured* target $I_n[\phi(\boldsymbol{x};\hat{\boldsymbol{a}})]$ from the *true* target $I_n[\phi(\boldsymbol{x}; \boldsymbol{a_0})]$ constantly occurs in each frame, which, along with the camera noise, forms the measurement noise. We refer to the former component as *drift noise*. Accumulation of the drift noise is the ultimate cause of template drift.

In order to ensure an optimal estimation of the true target appearance in face of the measurement noise, Kalman filtering is employed to update the template, where the state equation and the measurement equation for a template pixel are

$$T(\boldsymbol{x},n) = T(\boldsymbol{x},n-1) + \varepsilon_S(\boldsymbol{x},n-1)\,, \qquad (2)$$

$$I_n[\phi(\boldsymbol{x},\hat{\boldsymbol{a}})] = T(\boldsymbol{x},n) + \varepsilon_M(\boldsymbol{x},n)\,. \qquad (3)$$

Here, $T(\boldsymbol{x},n)$ denotes the grayscale of a template pixel $\boldsymbol{x}$ at frame n. $\varepsilon_S(\boldsymbol{x},n-1)$ is the state transition noise which actually reflects the variation of the appearance of the target *itself* from frame $n-1$ to frame n. It is reasonable to assume that $\varepsilon_S(\boldsymbol{x},n)$ is a zero-mean white noise with the power spectrum $\sigma_S{}^2(\boldsymbol{x},n)$. $\varepsilon_M(\boldsymbol{x},n)$ represents the measurement noise which is also white and zero-mean. The power of $\varepsilon_M(\boldsymbol{x},n)$ is $\sigma_M{}^2(\boldsymbol{x},n)$, which consists of the power of the drift noise and the camera noise.

According to the theory of Kalman filtering [10], equations (4) to (7) form a complete iteration to update the estimated value of the template pixel:

$$\sigma_P^2(\boldsymbol{x},n) = \sigma_E^2(\boldsymbol{x},n-1) + \sigma_S^2(\boldsymbol{x},n-1)\,, \qquad (4)$$

$$G(\boldsymbol{x},n) = 1\Big/\left[1+\sigma_M^2(\boldsymbol{x},n)\big/\sigma_P^2(\boldsymbol{x},n)\right], \qquad (5)$$

$$\sigma_E^2(\boldsymbol{x},n) = [1-G(\boldsymbol{x},n)]\sigma_P^2(\boldsymbol{x},n)\,, \qquad (6)$$

$$\hat{T}(\boldsymbol{x},n+1) = \hat{T}(\boldsymbol{x},n) + G(\boldsymbol{x},n)\left\{I_n[\phi(\boldsymbol{x};\hat{\boldsymbol{a}})] - \hat{T}(\boldsymbol{x},n)\right\}. \qquad (7)$$

Here, $\sigma_P{}^2$ and $\sigma_E{}^2$ are the powers of the prediction error and the estimation error, respectively. They are automatically calculated in the iterations of the Kalman filtering. What should be estimated is the power of the two model noises, $\sigma_S{}^2$ and $\sigma_M{}^2$, which are related by the following equation [4]:

$$\sigma_V^2(\boldsymbol{x},n) = \sigma_E^2(\boldsymbol{x},n-1) + \sigma_S^2(\boldsymbol{x},n-1) + \sigma_M^2(\boldsymbol{x},n)\,, \qquad (8)$$

where $\sigma_V{}^2$ is the power of the innovation which can be approximated as the spatio-temporal mean squared differences:

$$\sigma_V^2(\boldsymbol{x},n) \approx \frac{1}{N_L} \sum_{k=n-L+1}^{n} \sum_{z\in\boldsymbol{\Omega}_L(\boldsymbol{x})} \left\{I_k[\phi(z;\hat{\boldsymbol{a}})] - \hat{T}(z;k)\right\}^2, \qquad (9)$$

where L is the length of the temporal moving-average window, $\boldsymbol{\Omega}_L(\boldsymbol{x})$ is a spatial neighborhood centered at $\boldsymbol{x}$, and N_L is the number of pixels involved in the averaging process.

According to (8), if one of the model noise powers is obtained, the other one can be trivially calculated. Both [4] and [5] assume one of the model noise powers to be constant. This assumption, however, is appropriate only in very specific tracking cases.

3. EVALUATING MEASUREMENT NOISE POWER

As has been discussed in the previous section, the discrepancy between $\hat{\boldsymbol{a}}$ and $\boldsymbol{a_0}$ leads to the inaccuracy of the transformed coordinate $\phi(\boldsymbol{x};\hat{\boldsymbol{a}})$ and hence the drift error in $I_n[\phi(\boldsymbol{x};\hat{\boldsymbol{a}})]$. As is shown in Fig. 1, the *true* position of a template pixel $\boldsymbol{x}$ might reside in a region $\boldsymbol{\Omega_u}$ which is centered at $\phi(\boldsymbol{x};\hat{\boldsymbol{a}})$ in the current frame I_n, and the *true* value of the pixel $\boldsymbol{x}$ is therefore equal to the value of a certain point within $\boldsymbol{\Omega_u}$, which is probably *not* $\phi(\boldsymbol{x};\hat{\boldsymbol{a}})$. Increasing precision of (1) results in smaller size of $\boldsymbol{\Omega_u}$ and thus less drift noise.

For the simplicity of notation, we use $\boldsymbol{a}$ instead of $\boldsymbol{a_0}$ to denote the *true* values of the transformation parameters. The drift noise power of a template pixel located at $\boldsymbol{x}$ can be formulated as

$$\sigma_D^2(\boldsymbol{x},n) = \int_{\boldsymbol{a}} \left\{I_n[\phi(\boldsymbol{x};\boldsymbol{a})] - I_n[\phi(\boldsymbol{x};\hat{\boldsymbol{a}})]\right\}^2 p_a(\boldsymbol{a}\,|\,\hat{\boldsymbol{a}})d\boldsymbol{a}\,, \qquad (10)$$

where $\sigma_D{}^2(\boldsymbol{x},n)$ is the drift noise power of pixel $\boldsymbol{x}$ at frame n, and p_a is the joint posterior distribution of the components of $\boldsymbol{a}$ conditioned on the value of $\hat{\boldsymbol{a}}$. The posterior distributions of the components of $\boldsymbol{a}$ are independent of one another when $\hat{\boldsymbol{a}}$ is in the close vicinity of $\boldsymbol{a}$, and (10) is therefore written as

$$\sigma_D^2(\boldsymbol{x},n) = \iint\cdots\int_{\boldsymbol{a}_1\boldsymbol{a}_2\cdots\boldsymbol{a}_m} \{I_n[\phi(\boldsymbol{x};\boldsymbol{a})] - I_n[\phi(\boldsymbol{x};\hat{\boldsymbol{a}})]\}^2 \prod_{i=1}^{m} p_i(a_i \mid \hat{a}_i)da_i \, . \quad (11)$$

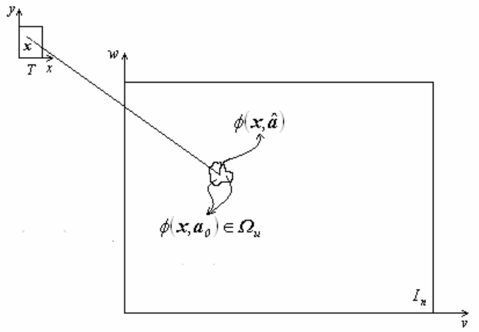
y
x
T
x
w
$\phi(\boldsymbol{x},\hat{\boldsymbol{a}})$
$\phi(\boldsymbol{x},\boldsymbol{a}_0) \in \Omega_u$
I_n
v

Fig. 1. Template drift occurs when the true mapped position $\phi(\boldsymbol{x}; \boldsymbol{a_0})$ lies in a region around the searching result $\phi(\boldsymbol{x}; \hat{\boldsymbol{a}})$.

Here, p_i is the posterior distribution of a_i, the i-th component of $\boldsymbol{a}$, and m is the number of transformation parameters that ϕ contains.

Now we focus on the calculation of p_i. As $\hat{a}_i$ can only take discrete values, the conditional probability of $\hat{a}_i$ is

$$P_i(\hat{a}_i \mid a_i) = \begin{cases} 1, & |\hat{a}_i - a_i| \le \Delta_i/2 \\ 0, & else \end{cases}, \quad (12)$$

where P_i is the conditional probability of $\hat{a}_i$ after a_i is given, and Δ_i is the final step size with which (1) searches for $\hat{a}_i$. From the Bayesian theory, the posterior distribution of a_i can be expressed as

$$p_i(a_i \mid \hat{a}_i) = \frac{P_i(\hat{a}_i \mid a_i)p(a_i)}{\int P_i(\hat{a}_i \mid a_i)p(a_i)da_i} \, . \quad (13)$$

Substituting (12) into (13) yields

$$p_i(a_i \mid \hat{a}_i) = \begin{cases} \dfrac{p(a_i)}{\int_{\hat{a}_i-\Delta_i/2}^{\hat{a}_i+\Delta_i/2} p(a_i)da_i}, & |a_i - \hat{a}_i| \le \Delta_i/2 \\ 0, & else \end{cases} \, . \quad (14)$$

Although it is difficult to acquire the exact value of $p(a_i)$, we can reasonably assume that $p(a_i)$ is approximately constant within the integral interval, since $p(a_i)$ is relatively flat near its maximum and Δ_i is relatively small. Equation (14) is therefore reduced to

$$p_i(a_i \mid \hat{a}_i) = \begin{cases} 1/\Delta_i, & |a_i - \hat{a}_i| \le \Delta_i/2 \\ 0, & else \end{cases} \, . \quad (15)$$

While it is a challenging task to arrive at an analytical expression of $\sigma_D{}^2(\boldsymbol{x},n)$ from (11) and (15), we can still obtain a numerical result by replacing the integral with summation. Let $\Delta a_1\cdots\Delta a_m$ forms an elementary hypercube in the hyperspace of $\boldsymbol{a}$ as a unit for summation, and define $\boldsymbol{a_k}$ as $[k_1\Delta a_1\cdots k_m\Delta a_m]^T$, we have

$$\begin{aligned}\sigma_D^2(\boldsymbol{x},n) &\approx \sum_{k_1}\cdots\sum_{k_m} \{I_n[\phi(\boldsymbol{x};\boldsymbol{a_k})] - I_n[\phi(\boldsymbol{x};\hat{\boldsymbol{a}})]\}^2 \prod_{i=1}^{m} p_i(k_i\Delta a_i \mid \hat{a}_i)\Delta a_i \\ &= \left(\prod_{i=1}^{m}\frac{\Delta a_i}{\Delta_i}\right)\cdot \sum_{k_1}\cdots\sum_{k_m} \{I_n[\phi(\boldsymbol{x};\boldsymbol{a_k})] - I_n[\phi(\boldsymbol{x};\hat{\boldsymbol{a}})]\}^2\end{aligned} . \quad (16)$$

Here, the range of the integer k_i in the summation satisfies

$$|k_i\Delta a_i - \hat{a}_i| \le \Delta_i/2 \, , \; i = 1,2,\cdots,m \, . \quad (17)$$

The other component of the measurement noise power is the camera noise power $\sigma_C{}^2$, which is assumed to be constant and can be acquired in advance. The final estimate of the measurement noise power for the template pixel $\boldsymbol{x}$ is therefore obtained as follows:

$$\sigma_M^2(\boldsymbol{x},n) = \sigma_D^2(\boldsymbol{x},n) + \sigma_C^2 \, . \quad (18)$$

From the discussion above, it can be seen that the measurement noise power is heavily dependent on the target appearance: higher density of textures or edges contained in target appearance results in larger measurement noise power. This is not surprising, because the same amount of deviation of the target location will cause greater appearance errors and (hence) severer template damage to the template pixels surrounded by complex target appearance than simple target appearance. As a result, template drift is more prone to occur when the target appearance contains more details. In our algorithm, more appearance details lead to higher measurement noise power which precludes the Kalman gain of the appearance filter from getting too large, and consequently template drift can be significantly reduced.

4. EXPERIMENTAL RESULTS

We perform experiments on a wide range of real-world video sequences in which the targets undergo various degrees of changes in their appearances. As the experiments on all the sequences have similar results, we only present three of them in this paper, which are demonstrated in Fig. 2. Each row shows the result of one sequence, and the degrees of the target appearance variations increase from the first row to the third row. We compare the performances of different algorithms: the algorithm in [1], the algorithm in [5], and our proposed algorithm. In each row, the leftmost image displays the common initialization for all the algorithms; the subsequent three images from left to right are the final tracking results of the algorithm in [1], the algorithm in [5], and our algorithm, respectively. The current template is overlapped in the lower-right corner of each image.

It is observed in the experiments that when the target appearance varies little, the algorithm in [5] suffer from severe template drift as a result of unnecessarily high Kalman gain (see Fig. 2-a$_3$). When the variation of the target appearance becomes more intensive, the performance of the algorithm in [1] deteriorates a lot due to the invalidity of the first template to serve as a benchmark (see Fig. 2-b$_2$ and -c$_2$). In this case, the algorithm in [5] still incurs some template drift (see Fig. 2-b$_3$ and -c$_3$), because the Kalman gain calculated by [5] is sub-optimal.

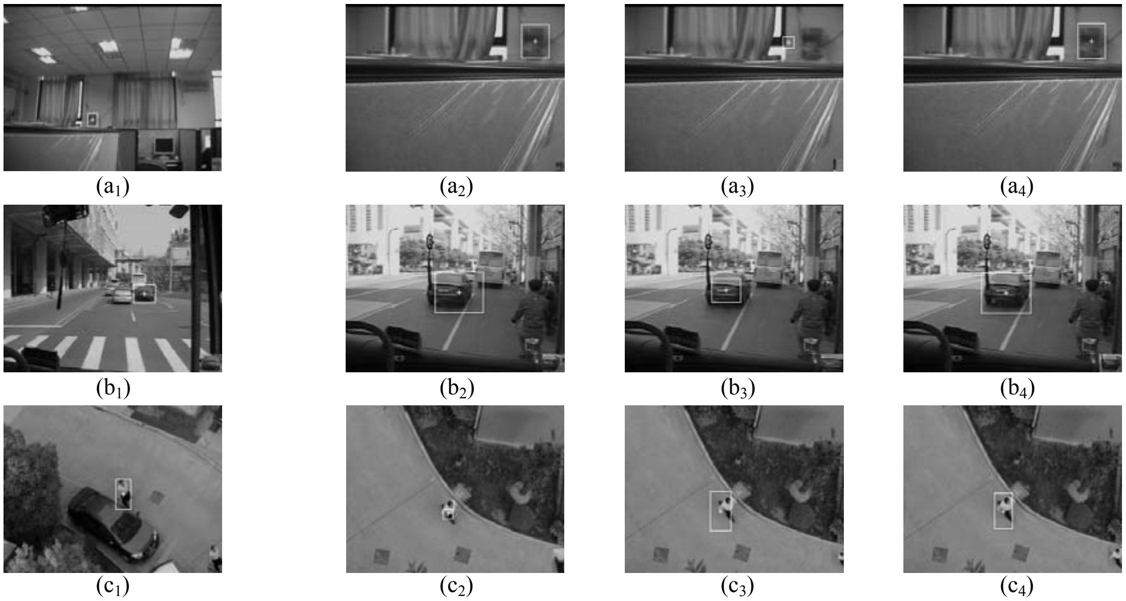

Fig. 2. Comparison of the robustness against template drift for various algorithms. The first column is the common initialization for all the algorithms. The subsequent columns from left to right are the results for the algorithm in [1], the algorithm in [5], and our algorithm, respectively.

Our proposed algorithm achieves the highest tracking accuracy throughout the experiments. We observe that even when there is *little* change in the target appearance, the innovation is much larger when the target is smaller in scale and thus compact with features. Evidently, such large innovation results from matching inaccuracy, not the variation of the target appearance. Our algorithm takes this into account by raising the measurement noise power and keeping the Kalman gain low. As a result, almost no template drift occurs when the target appearance remains fixed (see Fig. 2-a$_4$). On the other hand, when the target has a changing appearance, our algorithm effectively reduces template drift by updating the template just in time and just in place to keep up with the target appearance variations while refraining from over-updating the template (see Fig. 2-b$_4$ and -c$_4$). To sum up, our algorithm is robust against template drift under any circumstance.

5. CONCLUSION

In this paper, we propose an algorithm that is robust against template drift when tracking objects. This purpose is achieved by correctly calculating the measurement noise power online, in which the key is to evaluate the drift noise power after obtaining the distribution of the true values of template pixels. Experiments conducted on various types of real-world video sequences have demonstrated that our algorithm achieves the best performance of reducing template drift in all cases.

6. REFERENCES

[1] I. Matthews, T.Ishikawa, and S. Baker, “The Template Update Problem,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810-815, 2004.

[2] T. Kaneko and Osamu Hori, “Template Update Criterion for Template Matching of Image Sequences,” *Proc. IEEE Int’l Conf. Pattern Recognition*, vol. 2, pp. 1-5, 2002.

[3] A.M. Peacock, S. Matsunaga, D. Renshaw, J. Hannah, and A. Murray, “Reference Block Updating When Tracking with Block Matching Algorithm,” *Electronic Letters*, vol. 36, pp. 309-310, 2000.

[4] H.T. Nguyen, M. Worring, and R. van den Boomgaard, “Occlusion Robust Adaptive Template Tracking,” *Proc. IEEE Int’l Conf. Computer Vision*, vol. 1, pp. 678-683, 2001.

[5] H.T. Nguyen and A. W.M. Smeulders, “Fast Occluded Object Tracking by a Robust Appearance Filter,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1099-1104, 2004.

[6] C. Haworth, A.M. Peacock, and D. Renshaw, “Performance of Reference Block Updating Techniques When Tracking with the Block Matching Algorithm,” *Proc. IEEE Int’l Conf. Image Processing*, vol. 1, pp. 365-368, 2001.

[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] Z. Jia, A. Balasuriya, and S. Challa, “Target Tracking with Bayesian Fusion Based Template Matching,” *Proc. IEEE Int’l Conf. Image Processing*, vol. 2, pp. II - 826-9, 2005.

[10] R.G. Brown and P.Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley, 1992.