Honda Research Institute EU

Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Robust Tracking by Means of Template Adaptation with Drift Correction

Chen Zhang, Julian Eggert, Nils Einecke

2009

Preprint:

This is an accepted article published in Proceedings of the 7th International Conference on Computer Vision Systems (ICVS). The final authenticated version is available online at: https://doi.org/[DOI not available]

Robust Tracking by Means of Template Adaptation with Drift Correction

Chen Zhang[1], Julian Eggert[2], and Nils Einecke[2]

[1] Darmstadt University of Technology, Institute of Automatic Control, Control Theory and Robotics Lab, Darmstadt D-64283, Germany
[2] Honda Research Institute Europe GmbH, Offenbach D-63073, Germany

Abstract. Algorithms for correlation-based visual tracking rely to a great extent on a robust measurement of an object's location, gained by comparing a template with the visual input. Robustness against object appearance transformations requires template adaptation - a technique that is subject to drift problems due to error integration. Most solutions to this "drift-problem" fall back on a dominant template that remains unmodified, preventing a true adaptation to arbitrary large transformations. In this paper, we present a novel template adaptation approach that instead of recurring to a master template, makes use of object segmentation as a complementary object support to circumvent the drift problem. In addition, we introduce a selective update strategy that prevents erroneous adaptation in case of occlusion or segmentation failure. We show that using our template adaptation approach, we are able to successfully track a target in sequences containing large appearance transformations, where standard template adaptation techniques fail.

1 Introduction

Visually tracking arbitrary moving objects in a dynamic real-world environment is a difficult task. Tracking systems face many challenges like cluttered background, fluctuating environment conditions, occlusions, large object movements due to limited processing frame rate and not at least object appearance transformation. A robust tracking system must be able to cope with all these difficulties, in order to succeed in real-world applications

Modern Bayesian tracking systems (as e.g. described in [1], [2]) track arbitrary moving objects in a way that they estimate the object's parameters by first predicting the state and consecutively confirming the predicted state by incorporating a measurement of the state. In our case, the measurement of a target object is gained by comparing a template (as an appearance model) of the target object with the input image containing the sought object, using template matching techniques (see e.g. [3]). The quality of the measurement essentially depends on how up-to-date the template is. In order to make template matching robust in a sequence of images with changing object appearance, the template needs to be continuously updated towards the current appearance.

M. Fritz, B. Schiele, and J.H. Piater (Eds.): ICVS 2009, LNCS 5815, pp. 425–434, 2009.
© Springer-Verlag Berlin Heidelberg 2009

Under the assumption of small object appearance changes, different approaches were proposed to update the template towards the current appearance. These approaches apply parametric transformations depending on the type of assumed appearance change ([4]), e.g. using a linear appearance transformation (e.g. in [5], [6]) or an active appearance model (e.g. in [7]). Apart from the fact that these methods are gradient-descent based and therefore not suited for large transformations, they are not designed to cope with arbitrary appearance transformations.

However, in order to cope with large appearance transformation of arbitrary type, the most generic way is rather simple. The template is updated using the current input image at the position where the object is currently found. Unfortunately, due to small errors in the position estimation, the great drawback of this naive approach is the risk of a loss of the target by a systematic drift introduced by the template update process. Previous authors have addressed this problem (see e.g. [4], [8]), but the solutions are based on falling back on a master reference template in regular intervals, which is not suitable for large, irreversible transformations.

To tackle the template drift problem, we propose in this paper a novel method for template adaptation by cutting out a new template with drift correction. The drift problem is solved in the way that the drift is first detected by analyzing the segmented object mask and then the drift is immediately corrected after each cut-out of the new template. A further update strategy prevents an erroneous adaptation, e.g. in case of occlusion or segmentation failure. In this paper, we first present our tracking framework with its three constituting components: I) multiple-cue based measurement by template matching, II) Bayesian based dynamic state estimation and III) template adaptation with drift correction. Then we demonstrate the performance of our tracking system by means of an experimental evaluation using different template adaptation methods.

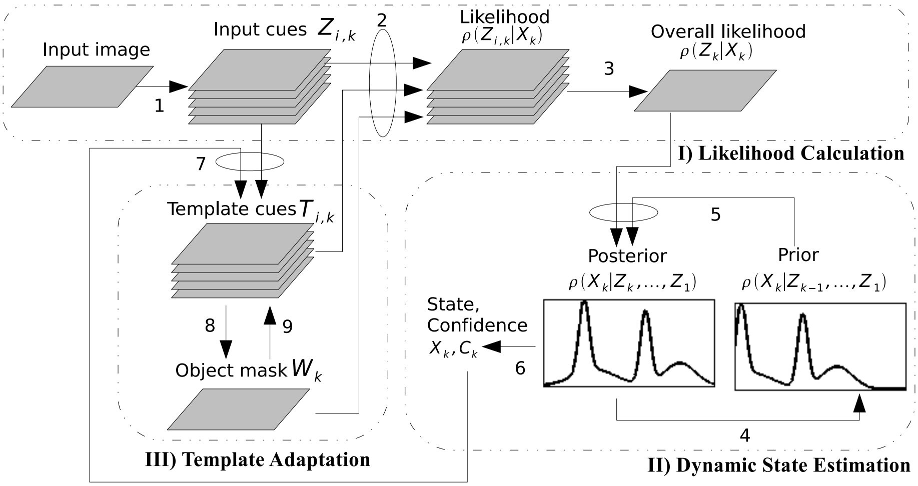
2 The General Tracking Framework

2.1 Framework Overview

We want to track an arbitrary object in 2D images by estimating its state vector $\mathbf{X}_k = (\mathbf{x}_k, \mathbf{v}_k, \mathbf{a}_k)$ - comprising position, velocity and acceleration - at time step k, given all its past and current measurements $\mathbf{Z}_k, \mathbf{Z}_{k-1}, ..., \mathbf{Z}_1$.

The general tracking framework can be separated into three main components: I) The multiple-cue based measurement part for calculating the likelihood, II) the dynamic state estimation part for obtaining $\mathbf{X}_k$, and III) the template adaptation part for keeping the template up-to-date in favor of a robust likelihood calculation.

Part I: This part starts with a color image as input. Indicated as step *1* in Fig. 1, the image is then preprocessed into an array $\mathbf{Z}_{i,k}$ of N 2D planes corresponding each to an activity field that indicates the degree of response to a given cue. In step *2*, a tracking template array $\mathbf{T}_{i,k}$ and an object mask $\mathbf{W}_k$ are used to find positions in the 2D planes that indicate a high probability for the presence of the tracked pattern. This template array consists of the same number of 2D planes as the preprocessed input after step 1, so that all the different cues



1. cue extraction	4. state prediction	7. template adaptation
2. likelihood calculation	5. state confirmation	8. mask segmentation
3. cue fusion	6. state estimation	9. drift correction

Fig. 1. Overview of tracking framework: the entire tracking system consists of nine processing steps within three main components: I) Calculation of measurement likelihood. II) Dynamic state estimation. III) Template adaptation. Details about the steps are described in sections 2.1-2.4.

are taken into consideration. The result is the likelihoods $\rho(\mathbf{Z}_{i,k}|\mathbf{X}_k)$ which are fused in step *3* to an overall likelihood $\rho(\mathbf{Z}_k|\mathbf{X}_k)$ as a single 2D plane.

Part II: The overall likelihood is used for a dynamic estimation of the tracking state $\mathbf{X}_k$ in a Bayesian manner in steps *4*, *5* and *6* of Fig. 1.

Part III: Depending on the confidence of the state vector (which mainly depends on the quality of the likelihood and therefore, of the current template), a decision is made in step *7* whether to adapt the template. In the case that a new template is extracted, a segmentation of the target object from the new template provides an up-to-date object mask (step *8*) which is used to prevent the drift during the template adaptation (step *9*).

Steps 1-9 are iterated, so that the state estimation and if necessary also the template and object mask adapt to changing input. The system has to be initialized with a starting state vector $\mathbf{X}_0$, a template $\mathbf{T}_{i,0}$ and an object mask $\mathbf{W}_0$ for the target object. This is assumed to occur from outside of the system, either by human interaction or by some higher-level processing centers that communicate down a hypothesis on which parts of the scene should be tracked.

2.2 Likelihood Calculation

In the evaluations, we used RGB color channels and structure tensors (see e.g. [9]) as cues. This leads to $N = 6$ cue planes which are correlated with a template pattern consisting of N corresponding planes which specifies the particular properties of the selected 2D region that is being tracked. We perform a windowed, normalized cross correlation of the cue planes with the template pattern.

We basically assume that the sought object/pattern in the input planes $\mathbf{Z}_{i,k}$ of time step k can be found by comparing the planes with their corresponding tracking template planes $\mathbf{T}_{i,k}$, so that

$$\mathbf{W}_k \odot \mathbf{T}_{i,k} \approx \mathbf{W}_k \odot \mathbf{Z}^{\mathbf{x}}_{i,k} \quad . \tag{1}$$

Here, $\mathbf{W}_k$ is a window operating in the two spatial coordinates that indicates the approximate limits of the tracked pattern. The contribution of the window is to suppress background clutter. $\mathbf{Z}^{\mathbf{x}}_{i,k}$ are cue planes of the input image which are centered around the spatial position $\mathbf{x}$ which is part of the overall state vector $\mathbf{X}_k$. Eq. (1) expresses that we expect the pattern of activity of the input around position $\mathbf{x}$ to be approximately matching the tracking template within the considered window $\mathbf{W}_k$.

We additionally assume that the input may contain a shifted and scaled (in terms of cue intensity) version of the tracking template, to make the template comparison and the tracking procedure contrast and brightness invariant. This means, that instead of Eq. (1), we consider

$$\mathbf{W}_k \odot \mathbf{T}_{i,k} \approx \lambda_i \mathbf{W}_k \odot \mathbf{Z}^{\mathbf{x}}_{i,k} - \kappa_i \, \mathbf{W}_k + \eta_i \, \mathbf{1} \quad . \tag{2}$$

Here, λ_i is a contrast scaling parameter and κ_i a brightness bias. The factor η_i is additive Gaussian noise with variance σ_i^2. It is important that all the input image patches $\mathbf{Z}^{\mathbf{x}}_{i,k}$ centered around different positions $\mathbf{x}$ get scaled to the corresponding reference template $\mathbf{T}_{i,k}$, and not the other way round.

The likelihood that the input $\mathbf{Z}^{\mathbf{x}}_{i,k}$ centered around position $\mathbf{x}$ matches the tracking template $\mathbf{T}_{i,k}$ within a window $\mathbf{W}_k$ can then be expressed as

$$\rho(\mathbf{Z}_{i,k}|\mathbf{X}_k) \sim e^{-\frac{1}{2\sigma_i^2}\left\|\lambda_i \mathbf{W}_k \odot \mathbf{Z}^{\mathbf{x}}_{i,k} - \kappa_i \, \mathbf{W}_k - \mathbf{W}_k \odot \mathbf{T}_{i,k}\right\|^2} \quad . \tag{3}$$

Eq. (3) constitutes an observation model that expresses the similarity between the real measurements $\mathbf{Z}_{i,k}$ and the measurements expected from knowing the state $\mathbf{X}_k$.

We now proceed to make Eq. (3) independent of λ_i and κ_i. For this purpose, we maximize the likelihood Eq. (3) with respect to the scaling and shift parameters, λ_i and κ_i. This amounts to minimizing the exponent, so that we want to find

$$\{\lambda_i^*(\mathbf{x}), \kappa_i^*(\mathbf{x})\} := \mathrm{argmin}_{\lambda_i,\kappa_i} \left\|\mathbf{W}_k \odot \left(\lambda_i \, \mathbf{Z}^{\mathbf{x}}_{i,k} - \kappa_i \mathbf{1} - \mathbf{T}_{i,k}\right)\right\|^2 \quad . \tag{4}$$

This leads to:

$$\lambda_i^*(\mathbf{x}) = \frac{\varrho_{\mathbf{Z}^{\mathbf{x}}_{i,k},\mathbf{T}_{i,k}} \cdot \sigma_{\mathbf{T}_{i,k}}}{\sigma_{\mathbf{Z}^{\mathbf{x}}_{i,k}}} \qquad \text{and} \tag{5}$$

$$\kappa_i^*(\mathbf{x}) = \lambda_i^*(\mathbf{x}) \, \mu_{\mathbf{Z}^{\mathbf{x}}_{i,k}} - \mu_{\mathbf{T}_{i,k}} \qquad \text{with} \tag{6}$$

$$\mu_{\mathbf{A}} = \langle \mathbf{A} \rangle := \frac{\mathbf{1}^T \mathbf{A} \odot \mathbf{W}_k{}^{②} \mathbf{1}}{\mathbf{1}^T \mathbf{W}_k{}^{②} \mathbf{1}} \;, \tag{7}$$

$$\sigma_{\mathbf{A}}^2 = \langle \mathbf{A}^{②} \rangle - \langle \mathbf{A} \rangle^2 \text{ , and} \tag{8}$$

$$\varrho_{\mathbf{A},\mathbf{B}} = \frac{1}{\sigma_{\mathbf{A}} \cdot \sigma_{\mathbf{B}}} \big\langle (\mathbf{A} - \mu_{\mathbf{A}} \mathbf{1}) \odot (\mathbf{B} - \mu_{\mathbf{B}} \mathbf{1}) \big\rangle \ , \tag{9}$$

where $\mathbf{A}$ and $\mathbf{B}$ have to be replaced by $\mathbf{Z}_{i,k}^{\mathbf{x}}$ and $\mathbf{T}_{i,k}$, respectively. Here, the operators $\odot$ and ^{②} indicate element-wise multiplication and element-wise squaring, respectively.

Inserting Eqs. (5) and (6) into Eq. (3), so that $\lambda_i = \lambda_i^*(\mathbf{x})$ and $\kappa_i = \kappa_i^*(\mathbf{x})$, leads to the final likelihood formulation:

$$\rho(\mathbf{Z}_{i,k}|\mathbf{X}_k) \sim e^{-\frac{1}{2}\left(\frac{\sigma_{\mathbf{T}_{i,k}}}{\sigma_i}\right)^2 \left(1 - \varrho_{\mathbf{Z}_{i,k}^{\mathbf{x}}, \mathbf{T}_{i,k}}\right)} \ . \tag{10}$$

For all cues, Eq. (10) is calculated, which basically amounts to the computation of the normalized, locally windowed correlation $\varrho_{\mathbf{Z}_{i,k}^{\mathbf{x}}, \mathbf{T}_{i,k}}$ of a cue with its template. This is done for each pair of cue/template planes separately, and the overall likelihood $\rho(\mathbf{Z}_k|\mathbf{X}_k)$ is then additively composed of all the planes computed according to Eq. (10)

$$\rho(\mathbf{Z}_k|\mathbf{X}_k) \sim \bigoplus_{i=1}^{N} \rho(\mathbf{Z}_{i,k}|\mathbf{X}_k) \ . \tag{11}$$

From Eq. (11) it is expected that, for cues which temporarily characterize the tracked object to some extent, there will be a pronounced peak at the position of the tracked object.

2.3 Dynamic State Estimation

The fundamental problem that a tracking system has to solve is that of recursive, dynamic target state estimation. This means that it has to estimate continuously the state $\mathbf{X}_k$ of the dynamic system

$$\mathbf{X}_k = f_{k-1}(\mathbf{X}_{k-1}) + v_{k-1} \tag{12}$$

$$\mathbf{Z}_k = h_k(\mathbf{X}_k) + w_k \tag{13}$$

using a series of measurements $\mathbf{Z}_k, \mathbf{Z}_{k-1}, ..., \mathbf{Z}_1$ gained from an observable that can be put in relation to the state $\mathbf{X}_k$. Here, $f_{k-1}(.)$ and $h_k(.)$ are models for state transition and object observation, respectively, and v_{k-1} and w_k are additive noise.

Bayesian tracking equations (see e.g. [1], [2], [10]) express this formally in two stages of the filtering process, usually termed *prediction* and *update* stages

$$\rho(\mathbf{X}_k|\mathbf{Z}_{k-1}, ..., \mathbf{Z}_1) = \int \rho(\mathbf{X}_k|\mathbf{X}_{k-1}) \rho(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}, ..., \mathbf{Z}_1) d\mathbf{X}_{k-1} \tag{14}$$

$$\rho(\mathbf{X}_k|\mathbf{Z}_k, ..., \mathbf{Z}_1) \propto \rho(\mathbf{Z}_k|\mathbf{X}_k) \rho(\mathbf{X}_k|\mathbf{Z}_{k-1}, ..., \mathbf{Z}_1) \tag{15}$$

with

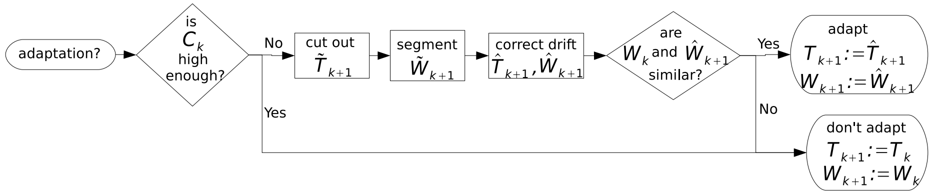


Fig. 2. Template adaptation strategy: a template adaption is only done for small calculated confidences C_k. For each template adaptation an object segmentation mask is calculated, which is used to correct the drift of target object in the template. A plausibility check between the newly segmented mask and the currently used mask is applied to prevent a possible wrong segmentation, which can happen e.g. due to occlusion or if the segmentation algorithm fails.

- $\rho(\mathbf{Z}_k|\mathbf{X}_k)$ as the measurement likelihood expressing the comparison between measurement and expected measurement (given by Eqs. (10), (11) for our correlation-based tracker),
- $\rho(\mathbf{X}_k|\mathbf{Z}_k, ..., \mathbf{Z}_1)$ as the posterior of the state given all past and current measurements, and
- $\rho(\mathbf{X}_k|\mathbf{Z}_{k-1}, ..., \mathbf{Z}_1)$ as the prior which is gained from the posterior of the last step using a state transition model $\rho(\mathbf{X}_k|\mathbf{X}_{k-1})$.

For coping with the six dimensional state space and multimodal distributions of the posterior, we use a *particle filter* approach in our framework for the estimation problem, with a linear prediction model for $\rho(\mathbf{X}_k|\mathbf{X}_{k-1})$. The particle filter produces estimates for $\mathbf{X}_k$ from the posterior $\rho(\mathbf{X}_k|\mathbf{Z}_k, ..., \mathbf{Z}_1)$.

2.4 Template Adaptation

In real-world applications objects may undergo arbitrary types of appearance transformations. To ensure a robust likelihood calculation $\rho(\mathbf{Z}_k|\mathbf{X}_k)$, the template $\mathbf{T}_{i,k}$ and corresponding object mask $\mathbf{W}_k$ need to be adapted to the current object appearance.

The most generic way to do template adaptation for coping with all types of appearance transformations consists in cutting out the patch at the determined object position $\mathbf{x}_k$ which is part of $\mathbf{X}_k$ and set it as the new template $\mathbf{T}_{i,k+1}$. However, apart from the fact that the object mask $\mathbf{W}_{k+1}$ must be recalculated after every template adaptation, the weakness of this method is that the target may drift from the template in course of time, because of the integration of small inaccuracies in the object measurement. In order to fix this problem, in our framework the drift problem is solved in a correction step after each cut-out.

As Fig. 2 illustrates, the first step of our template adaptation strategy is to check if it is necessary to adapt the template. For high confidences of state vector $\mathbf{X}_k$ there is no need to adapt template, since likelihood calculation works robustly. A possible confidence is for instance the value of the highest posterior peak. In the case if the confidence is starting to get worse, the new template

$\tilde{\mathbf{T}}_{i,k+1}$ on position $\mathbf{x}_k$ is cut out. A segmentation algorithm is applied to the newly cut-out template patch in order to generate a new object mask $\tilde{\mathbf{W}}_{k+1}$. Different segmentation algorithms are suitable for calculating the object mask, here we have chosen a Level-Set based segmentation algorithm as described in [11]. From the mask, the center of gravity of the object mask $\mathbf{x}_{\tilde{\mathbf{W}}_{k+1}}$ is calculated. The distance

$$\mathbf{d}_k = \mathbf{x}_{\tilde{\mathbf{W}}_{k+1}} - \mathbf{x}_k \tag{16}$$

is considered as the template drift which is therefore used to recenter the template and the mask in order to get the corrected template $\hat{\mathbf{T}}_{i,k+1}$ and mask $\hat{\mathbf{W}}_{k+1}$.

As a second criterion for excluding detrimental template adaptations, e.g. after an object loss, occlusion or failure of the segmentation algorithm, the newly gained mask $\hat{\mathbf{W}}_{k+1}$ undergoes a plausibility check, in the way that it must be similar to the current mask $\mathbf{W}_k$. A possible similarity measure is the correlation coefficient between $\hat{\mathbf{W}}_{k+1}$ and $\mathbf{W}_k$. Only in case of high similarity, the new template and mask are updated, otherwise the current template and mask are kept for the next frame, according to:

$$\mathbf{T}_{i,k+1} = \hat{\mathbf{T}}_{i,k+1} \text{ and } \mathbf{W}_{k+1} = \hat{\mathbf{W}}_{k+1} \text{ , if both adaptation criteria fulfilled} \tag{17}$$

$$\mathbf{T}_{i,k+1} = \mathbf{T}_{i,k} \text{ and } \mathbf{W}_{k+1} = \mathbf{W}_k \text{ , otherwise} \tag{18}$$

On applying these two criteria during template adaptation, a persistently low confidence value C_k of state $\mathbf{X}_k$ is a strong indication for target object loss, e.g. because of occlusion. In this case, the second criterion prevents to update the template $\hat{\mathbf{T}}_{i,k+1}$ in such cases.

3 Evaluation

This section demonstrates the evaluation results of our tracking system with the template adaptation mechanism as described in section 2.4.

In a sequence of 250 frames (illustrated in Fig. 3) a hand is tracked. As listed in the table of Fig. 3, this sequence contains four phases of different motions (each with translation and/or other types of appearance transformations, like 2D/3D rotation or change of hand shape). The appearance transformation that the template update system has to adapt to is easiest in phase I and gets increasingly harder for phases II-IV.

In the evaluation three methods for template adaptation are compared. In *method 1* **no** template and mask adaptation is applied. The template $\mathbf{T}_{i,0}$ and the object mask $\mathbf{W}_0$ from the first frame are used throughout the rest of the sequence. *Method 2* adapts template and its object mask as described in section 2.4 but **without** drift correction. That means, if the plausibility check is fufilled, the cut-out template $\tilde{\mathbf{T}}_{i,k+1}$ and its object mask $\tilde{\mathbf{W}}_{k+1}$ are used to update template and mask instead of using $\hat{\mathbf{T}}_{i,k+1}$ and $\hat{\mathbf{W}}_{k+1}$. *Method 3* applies the full template adaptation strategy described in section 2.4 according to Eqs. (17) and (18) which corresponds to method 2 but **with** drift correction.

Phase	2D translation	2D rotation	3D rotation	change of hand shape
I	from the left to the right	no	no	no
II	from the lower right to the upper left	yes	no	no
III	from the upper left the middle	yes	yes	no
IV	from the middle to the left and then to the right	yes	yes	between extended hand and clenched fist

Fig. 3. This figure illustrates selected frames of an evaluation sequence composed of 250 frames. The images are taken with a frame rate of 10Hz and have a size of 256×256 pixels. The sequence can be approximatively separated into four phases with individual motion types which are described in the table.

The evaluation results are shown in Fig. 4. They reveal that the tracking system fails to track the target object in cases using method 1 and method 2. Using method 1 the object is already lost at around frame 50 when the object starts to perform 2D rotation, because the correlation based measurement method as presented in section 2.2 can only cope with 2D translation, scaling intensity and baseline shift. Using method 2 the object is lost in phase II. In this case, the object smoothly drifts away from the template. After a while the template hardly contains any object parts so that the tracker sticks on the background. This drift process is illustrated in Fig. 5. Only by using method 3 the object is successfully tracked throughout all 250 images of the sequence. In no single frame the target object is lost or misplaced. Template adaptation is mainly triggered in phases III and IV when object appearance transformation is strongest.

In further evaluations (not shown) this framework was tested during occlusion conditions. In the cases of occlusion, the mask plausibility check failed, which prevented a template adaptation, although the measurement confidence was below threshold.

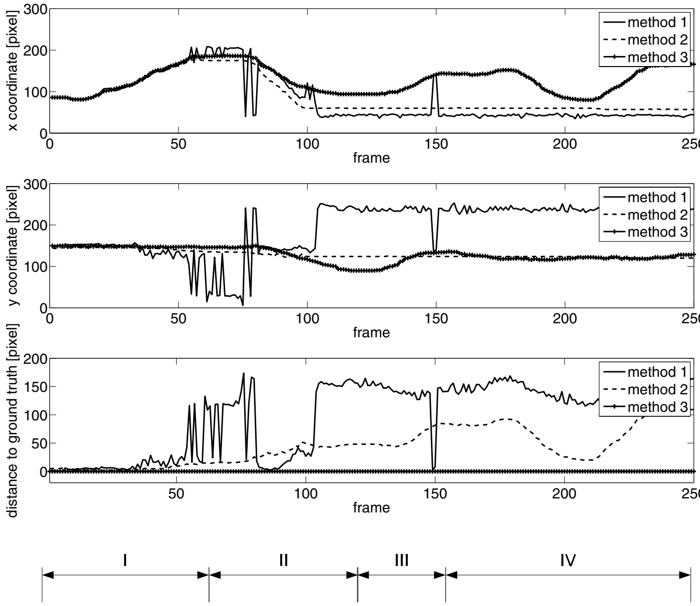


Fig. 4. Comparison of the three methods from section 3. Method 1 looses the target when the tracked object starts to perform 2D rotation. Method 2 looses the object in phase II, after the object more and more drifted out of the template and the tracker therefore confuses the background with the object. Only method 3 can successfully track the object throughout the entire sequence. For a more detailed description refer to section 3.

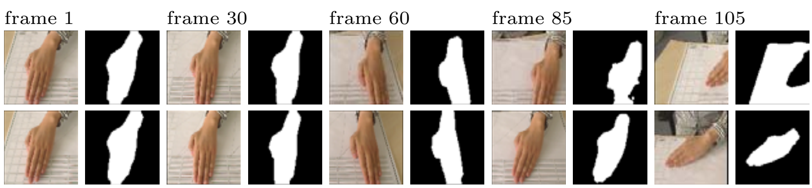


Fig. 5. This figures shows the template and its corresponding object mask during the tracking process in cases without and with drift correction. In the case of no drift correction (first row) one can see that the target object is lost from the template step by step, leading to a tracker failure. In the case with drift correction (second row) the template drift is corrected after every adaptation step using a newly calculated object mask, keeping the object steadily in the middle of template.

4 Conclusion

In this paper we presented a novel template adaptation approach for robust template-matching-based object tracking which allows large appearance transformations of arbitrary types. The approach consists of cutting out a new template at an estimated object position, where a segmentation algorithm is then applied to update the mask in order to apply a drift correction to prevent integration of position errors between template and object. By applying two criteria as conditions for template adaptation, we make sure that it is only applied if it is necessary and its result is plausible. A large change of mask is considered as an indication of segmentation failure, occlusion or loss of target. In a comparative setting using an image sequence containing four phases with different types of appearance transformations, a template-matching-based particle filter tracking system is evaluated according to three different template adaptation methods: no adaptation, adaptation without drift correction and adaptation with drift correction. The method with drift correction clearly proved to be the most robust one, allowing continuous tracking despite difficult appearance transformations whereas the other two methods failed at some point of the test sequence.

References

1. Arulampalam, S., Maskell, S., Gordon, N.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Transactions on Signal Processing 50, 174–188 (2002)
2. Ristic, B., Arulampalam, S., Gordon, N.: Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House, London (2004)
3. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice Hall International, Upper Saddle River (2007)
4. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. IEEE Trans. Pattern Anal. Mach. Intell. 26, 810–815 (2004)
5. Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework. International Journal of Computer Vision 56, 221–255 (2004)
6. Hager, G.D., Belhumeur, P.N.: Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 1025–1039 (1998)
7. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 681–685 (2001)
8. Kaneko, T., Hori, O.: Template update criterion for template matching of image sequences. In: 16th International Conference on Pattern Recognition, New York, pp. 1–5. IEEE Press, Los Alamitos (2002)
9. Bigun, J., Granlund, G.H.: Optimal orientation detection of linear symmetry. In: First International Conference on Computer Vision, Washington, DC,pp. 433–438. IEEE Computer Society Press, Los Alamitos (1987)
10. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. International Journal of Computer Vision 29, 5–28 (1998)
11. Weiler, D., Eggert, J.P.: Multi-dimensional histogram-based image segmentation. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part I. LNCS, vol. 4984, pp. 963–972. Springer, Heidelberg (2008)