J Real-Time Image Proc
DOI 10.1007/s11554-013-0329-2

SPECIAL ISSUE PAPER

A computationally efficient tracker with direct appearance-kinematic measure and adaptive Kalman filter

Rami Ben-Ari · Ohad Ben-Shahar

Received: 21 June 2012 / Accepted: 8 February 2013

Abstract Visual tracking is considered a common procedure in many real-time applications. Such systems are required to track objects under changes in illumination, dynamic viewing angle, image noise and occlusions (to name a few). But to maintain real-time performance despite these challenging conditions, tracking methods should require extremely low computational resources, therefore facing a trade-off between robustness and speed. Emergence of new consumer-level cameras capable of capturing video in 60 fps challenges this tradeoff even further. Unfortunately, state-of-the-art tracking techniques struggle to meet frame rates over 30 VGA-resolution fps with standard desktop power, let alone on typically-weaker mobile devices. In this paper we suggest a significantly cheaper computational method for tracking in colour video clips, that greatly improves tracking performance, in terms of robustness/speed trade-off. The suggested approach employs a novel similarity measure that *explicitly* combines appearance with object kinematics and a new adaptive Kalman filter extends the basic tracking to provide robustness to occlusions and noise. The linear time complexity of this method is reflected in computational efficiency and high processing rate. Comparisons with two recent trackers show superior tracking robustness at more than 5 times faster operation, all using naïve C/C++ implementation and built-in OpenCV functions.

R. Ben-Ari (✉)
Orbotech Ltd., Yavne, Israel
e-mail: benari.rami@gmail.com; benarir@windowslive.com

O. Ben-Shahar
Ben-Gurion University, Beersheba, Israel
e-mail: ben-shahar@cs.bgu.ac.il

1 Introduction

Visual tracking is used today in numerous applications such as surveillance, video communication, military missions, games, augmented reality and more. To be successful and practical, tracking algorithms need to satisfy two basic, though sometimes conflicting qualities: *robustness* and *speed*. Robustness implies the ability of the algorithm to track objects under various transformations of the visual signal, and in particular under changes of shape and appearance of the tracked object, due to factors such as illumination variations, occlusions, clutter, distractions, etc. The speed, on the other hand, relates to the number of frames per second (fps) that can be processed on a given computational resource. Typically, the robustness of a tracker is inversely related to the practical speed by which it can operate. Nowadays, common video cameras capable of capturing video at 60 fps produce a ramp up in requirements for computationally cheap trackers, while special equipment and specific applications (in sport or military) may demand trackers with even faster operation capability. Once computational resources are limited (as is often the case in camera clusters, mobile devices, embedded systems, or on board robotics systems) tracking performance may decrease severely, enough to prevent practical use or to critically limit the remaining resources needed for higher level tasks. When put in isolation, tracking methods must, therefore, exhibit "above real time" performance without compromising robustness. In this paper, we introduce such a method, which significantly improves the robustness/speed trade-off.

Tracking is certainly one of the most popular methods in computer vision [43], which motivated quantitative studies and complexity analysis of various classes of methods [32]. The operation of visual tracking can be thought of as the

Published online: 28 March 2013

recurrent detection of a predefined target in a sequence of images. In general, such target detection methods can be divided into *geometric* or *direct* methods. Geometric approaches require the extraction of a set of geometric primitives (e.g., points, contours, corners, etc.) from the two consecutive frames [17, 39]. Matching between frames is then obtained by classification. The computational complexity involved in both, the feature extraction and the classification phase, results in tracking speeds that rarely exceeds 30 fps [10, 20, 39]. Unlike geometric approaches, direct methods [14, 18, 19, 33, 34, 31, 41] exploit pixel intensities/colour without having to extract and match geometric features. Direct methods either depend only on colour histograms [1, 4, 38], disregarding the structural arrangement of pixels, or on appearance models, which ignore the statistical properties. There are several shortcomings for these representations. On the one hand, populating higher dimensional histograms by a small number of pixels results in incomplete representation. On the other hand, appearance models are sensitive to geometric and photometric transformations. Attempts to cope with this weakness of appearance models have driven the *template update* approach that has raised in turn the well-known *drifting* problem [26]. Many techniques thus avoid template update and deal with appearance variations via more sophisticated template matching such as advanced histogram modelling [4] or learning and classification [5]. Often, direct methods provide better robustness/speed trade-off presenting higher frame rates, e.g., [41] reaching 50–100 fps, on small frame sizes (typically up to VGA).

One of the powerful constraints for rejecting false positives during tracking is target kinematics. Tracking methods commonly incorporate kinematic priors *implicitly* using model-based estimators such as Kalman and Particle filters. One such prior relies on certain assumptions of the target's kinematic properties. For example, Matei et al. [25] recently used kinematic features with appearance cues to train classifiers and increase the discrimination between vehicles in their tracking scenarios. In this paper, we use a direct combination of appearance and kinematic priors. Our kinematic constraint is based on the *soft* assumption that target velocity is locally constant between consecutive frames. More specifically, we suggest a new measure of similarity based on normalized cross correlation (NCC) endowed with a penalty according to the deviation from the predicted target path (which is updated at each frame). The resultant similarity measure, integrated with a Kalman filter, yields a smoother cost function, tolerant to noise and appearance changes in a computationally efficient way.

In years, tracker's capabilities have been enhanced with model-based estimators to cope with noise, clutter, distractions and occlusions. The two main strategies have been Particle filters [13, 27, 38] and Kalman filters [17, 19, 24, 41]. Particle filters (PF) can estimate the state function directly from data without any prior assumptions about the associated noise distribution. One main drawback of PF methods, however, is the *curse of dimensionality* [16]. Solutions can become more accurate by increasing the number of particles but at the expense of extensive grow in the computational workload. While parallel processors and massively parallel devices such as GPUs can speed up the number crunching, the considerably higher computation effort will elevate power consumption, a crucial outcome in many applications including mobile devices.

A classical alternative to PF is the Kalman filter [12, 40]. The linear Kalman estimator provides an optimal solution (in least squares sense) when the state model and measurements are deviated from their true values by unbiased, uncorrelated, additive, Gaussian noise [12]. Although these optimal conditions are rarely satisfied in visual tracking, the Kalman filter (KF) provides a computationally efficient and robust solution, as demonstrated in many studies, e.g., [9, 11, 17, 19, 37].

In order to adjust the tracker to extreme changes in target appearance, the Kalman Filter covariance matrices are often varied during the course of track yielding what is known as *Adaptive* Kalman Filter (AKF) [14, 19, 34, 41, 42]. Different studies suggest various updating functions and arguments for the adaptive control. For instance, Falvio et al. [19] endowed their KF with a SSD measure applied on grey levels. In Weng et al. [41] the status of the KF was changed by threshold over the amount of motion in the scene, therefore working under the restrictive assumption of a static camera scenario, allowing only for a limited moving targets in the scene. Similarly, in [34] the AKF is controlled by acceleration, introducing a noise-sensitive argument and an ambiguity for targets having constant velocity. While most previous works in visual tracking concentrate on the argument of the adaptive control, the important characteristics of the control function was overlooked. For instance, in [14, 19, 41] the status of the KF was controlled by a *binary* function assigning the AKF two discrete values, *visible* and *occluded* states. In this work we use a mapping of the appearance similarity score to adaptively update the noise covariance matrices and *continuously* change the status of the tracker between visible and occluded stages. This approach is motivated by the reasoning that occlusions strongly affect the appearance and are generated gradually. Our approach can be related to the recently published study in [14] where the noise covariance matrix was controlled by a kernel response. However, the occlusion in this recent study was handled separately from the KF, using, again, a binary function.

Performance assessment shows tracking success and computational speeds on various test cases. The test bed includes public data processed by state-of-the-art tracking

approaches, e.g., [6, 7, 23]. The capability of the suggested tracker is further demonstrated under notorious tracking conditions such as specularities, change in viewing angle, occlusion and noise.

To summarize this introduction, our contribution boils down to a new visual tracking method that significantly improves the robustness/speed tradeoff. We first suggest a new measure for target-template similarity that endows appearance metric with motion as prior. Proper normalization maps these disparate measures to identical range, allowing robustness to appearance and motion variations under fixed setting. This somewhat non-standard integration of appearance and kinematics proves to significantly enhance tracking performance in a computationally efficient manner. Additionally, we introduce a *continuously-controlled* adaptive Kalman filter that yields greater tolerance to noise, occlusions and appearance changes. The resultant method yields an effective tracker to run in excess of 150 fps on a single core and standard implementation, more than five times as fast as existing tracking methods that still struggle to achieve above real time performance.

The rest of this paper is organized as follows: Sect. 2 presents our robust dissimilarity measure for appearance correlation endowed with a motion prior. Section 3 then discusses our novel adaptive Kalman filter while computational complexity is addressed in Sect. 4. This is followed by experimental evaluation, performance assessment and run-time comparison in Sect. 5. Finally the paper is concluded with a summary and discussion in Sect. 6.

As a notational convention, we denote scalar and vector quantities, respectively, by regular and bold face lower case letters. Matrices are denoted by regular upper case letters.

2 Object dissimilarity measure

In the template matching, a.k.a the sliding window approach, the position of an object is found by minimizing a dissimilarity measure between the appearance of an image patch and a predetermined reference template. To make the tracking algorithm robust to false detection and reduce the computational load, we consider a region of interest Ω at each frame, being $\mathcal{K}$ times the template size, and search for the best matching criteria at this restricted region. It is important to note that the search window is forwarded at each frame to lower the chance for losing the target by departing from the search window.

2.1 Appearance correlation

As the appearance dissimilarity measure we start with the normalized cross correlation (NCC), which is invariant to affine photometric transformations of the appearance pattern and acknowledged for its tolerance for varying illumination conditions and noise [21].

Let $\gamma(x, y)$ denote the NCC measure of a multi-channel (e.g., R,G,B) template in a search window. By definition, $-1 \leq \gamma \leq 1$ and hence bounded. Since zero and negative correlations indicate poor matches (i.e., a non-target sub-image), we rectify NCC negative values and apply an algebraic manipulation to produce a revised measure that exhibits smaller values to higher match measures:

$$\gamma_r(x,y) = \begin{cases} 1 - \gamma(x,y) & \gamma(x,y) \geq 0 \\ 1 & \text{Otherwise.} \end{cases} \quad (1)$$

This template matching process can be considered now as the data fidelity part of an energy minimization, which we later endow with a prior. Although at the first glance the rectification of the NCC may seem unnecessary, this action affects the relative weight between the data-fidelity term and the prior and yields robustness (invariancy) of this relative weight to the input (see also Sect. 2.2). The idea of rectification (a.k.a truncation) for the matching measure has been previously used in other domains [35].

Image colour is typically considered in terms of the three *RGB* components. However, the *RGB* space is far from being photometric invariant [22]. Indeed, a slight change in the colour (in terms of human perception) often yields a significant shift in RGB space. There are colour spaces that offer (nearly) invariancy to photometric variations (e.g, [29, 41]). One such colour space is the HSV (Hue-Saturation-Value) colour space typically described as a conical volume and shown to be more resistant to lighting changes since its three components are nearly uncoupled. Both hue and saturation are robust to shadows and shading while the hue is also invariant to specularities [29]. We, therefore, transform the RGB colour components of the captured colour frame to HSV space, prior to correlation stage. Since the HSV space combines coordinates of different nature, and in particular, its H dimension is angular rather than longitudinal, we represent the HSV space in the Cartesian coordinates via projection.

2.2 Joint kinematic and appearance matching measure

Employing the above variations on standard tracking building blocks is far from solving all tracking confounds that one is likely to encounter, in typical video sequences. Often tracking fails due to *distractions*, namely the existence of target-like image patches that enter the search window and possess better appearance match with the reference template than the real target patch. This possibility becomes more likely as the target appearance is drifted away in time and departs from the reference template. To handle these so-called "distracters", causing a

false-positive result, one must therefore, compensate the matching procedure with other means. This section introduces a new matching criteria utilizing our knowledge on the objects motion pattern to be incorporated directly into the matching measure.

Obviously, what differentiates the true target from a distracter in any particular frame is its spatial coherence with location in the previous frame. Assuming the target was detected correctly in the past frames, one can exploit this information, and the tendency of velocities to be smooth, in the target similarity measure. As illustrated in Fig. 1, let $\mathbf{p}(k-1)$ and $\mathbf{p}(k)$ denote the instantaneous target position at two consecutive time steps (or frames) $k-1$ and k, respectively. Moreover, let $\mathbf{v}(k)$ denote the *measured* (or observed) velocity of a candidate pattern at time k while $\mathbf{v}_P(k)$ is its *predicted* velocity.[1]

Given these quantities, one can, therefore, define an "acceleration vector" that represents the deviation of the target from the predicted pattern:

$$\mathbf{a}(k) := (\mathbf{v}(k) - \mathbf{v}_P(k))/\Delta t. \tag{2}$$

with $\Delta t = 1$ as the interval time unit between frames.

We assume a smooth object path with low accelerations with respect to the frame rate. Hence, object positions deviating from a constant velocity pattern (see Fig. 1) are associated with lower probability for target location. We modify the appearance-based NCC measure from Eq. 1 with this velocity coherence assumption. The new target detection model then becomes

$$(x_T, y_T) = \underset{(x,y)\in\Omega}{\operatorname{argmin}} \left(\gamma_r(x,y) + \beta \frac{\|\mathbf{a}(k)\|}{a_{\max}} \right) \tag{3}$$

where β is a relative weight between the appearance score $\gamma_r(x, y)$ and the motion prior while $a_{\max}$ is a normalization factor that reflects the maximum observable acceleration (see Fig. 1).

In practice when the search window is advanced according to the prediction model (see the dashed rectangle in Fig. 1, the acceleration term $\mathbf{a}(k)$ can be calculated directly by the distance of the patch from the centre of the window. Setting $a_{\max}$ according to window size entails $0 \leq \frac{\|\mathbf{a}(k)\|}{a_{\max}} \leq 1$. In practice $a_{\max}$ can be determined as a factor of the target size. Hence, the motion prior term possesses a constant dynamic range identical to the appearance term, making the choice of the weight β insensitive to the track conditions. Indeed, in our experiments we have set it once for all of our test cases. Finally, the minimization of the energy function in Eq. 3 can be done in a brute force manner due to the small size of the search window.

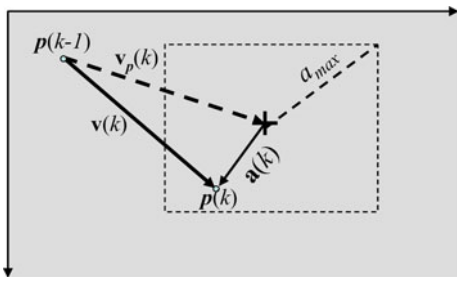

Fig. 1 A vector diagram presenting the kinematics of target prediction against measured (observed) values. Note that $\mathbf{a}(k)$ indicates deviation from instantaneous constant velocity pattern

Figure 2 demonstrates the effect of our combined appearance-motion dissimilarity measure in handling template-like distractions where a tracker without motion prior fails.

3 A novel adaptive Kalman filtering for visual tracking

During the course of track the object is expected to obtain appearance changes and can even be occluded for a period of time. The challenge for a robust tracker is to predict the target position reliably, during the occlusion period and recapture the target when it appears again. To this end, we embed the aforementioned energy minimization model into a novel Adaptive Kalman Filter (AKF) [12].

3.1 Kalman filter fundamentals

The Kalman filter mathematical model is described in terms of *state-space* variables [42]. Suppose: the vector $\mathbf{x} \in \mathbb{R}^m$ describes the state of our system, $\mathbf{z} \in \mathbb{R}^l, l \leq m$ is the measurement, given by our object dissimilarity measure and w, v are two random variables representing the process and measurement noise, respectively, or equivalently the discrepancies from the true values. The Kalman filter comprises of two stages, prediction and correction. The prediction is responsible for projecting forward the current state, obtaining *a-priori* estimate of the state while the correction step incorporates an actual measurement (observation) into the a-priori estimate to yield an improved *a-posteriori* evaluation. Assuming a linear model the state is predicted by the transition matrix A:

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + \boldsymbol{w_k} \tag{4}$$

The measurement $\mathbf{z}$ modelled by a linear operator with additive noise is given by:

$$\mathbf{z}_k = H\mathbf{x}_k + \boldsymbol{v_k} \tag{5}$$

[1] Predictions such as $\mathbf{v}_P(k)$ is part of our modified Kalman filtering approach discussed in Sect. 3 and at this point assume it is provided by an external oracle.

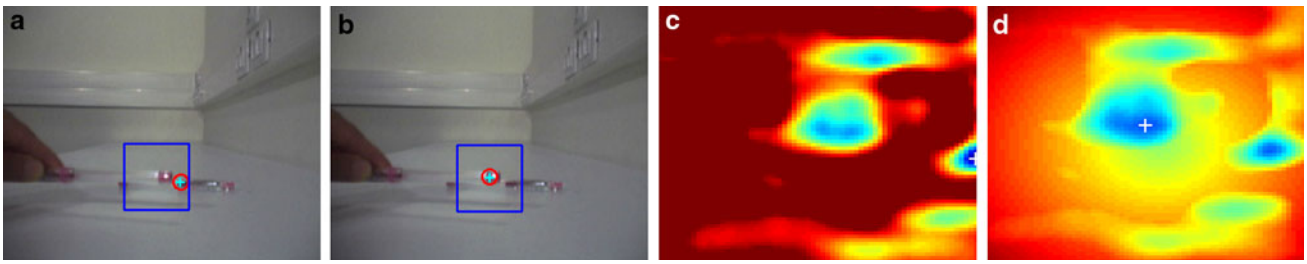

Fig. 2 Handling distractions by motion prior. **a** Demonstration of tracking failure due to a distracter, appearing in the search window. While tracking was supposed to follow the tip of the pen, a specular highlight made the real target less similar to the initial template, distorted by a peripheral target-like pattern. Tracking location is marked with a *circled plus sign*. **b** Successful tracking is obtained once motion prior is incorporated. **c** A *colour-coded* similarity map based on appearance. Note the global minimum at a false location, corresponding to **a**. **d** The new similarity map with the motion prior incorporated resulting a minimum correctly positioned at the real target, as indicated in **b**

where H is called the *measurement* matrix. Let us refer to the prediction and measurement noise covariance matrices Q and R, respectively, as the *uncertainties* in these two processes. Commonly Q, R are assumed to be constant. However, in an adaptive Kalman filter these terms are controlled and varied in time [30].

The Kalman filter estimates the state $\mathbf{x}$ at time k and corrects the prediction after each step of measurement by the following recursive stages [12, 40]:

Time update (prediction)

$$\hat{\mathbf{x}}_k = A\mathbf{x}_{k-1} \tag{6}$$

$$\hat{P}_k = AP_{k-1}A^T + Q_k \tag{7}$$

Measurement update (correction):

$$K_k = \hat{P}_k H^T \left(H\hat{P}_k H^T + R_k\right)^{-1} \tag{8}$$

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + K_k(\mathbf{z}_k - H\hat{\mathbf{x}}_k) \tag{9}$$

$$P_k = (I - K_k H)\hat{P}_k \tag{10}$$

where K_k is the Kalman *gain*, P_k is the state error covariance matrix and the values indicated with hat are *a-priori* estimates. The difference $(\mathbf{z}_k - H\hat{\mathbf{x}}_k)$ in Eq. 9 is called measurement *innovation* or the *residual*, indicating the discrepancy between the actual measurement and estimates, calculated by prediction. The *prediction-correction* steps are repeated recursively in time till convergence.[2] The gain K_k is a key parameter in Kalman filtering and presents the relative weight between the prediction and residual at each time step.

Assuming the statistical independence of the error (noise) associated with the state variables yields diagonal covariance matrices. Considering a unit time step (without loss of generality) allows us to have a single value diagonal entry for each covariance matrix, σ_p^2, σ_m^2 for Q, R, respectively, presenting the variance of the error distribution in terms of squared pixels. As observed in Eq. 8, when the measurement uncertainty σ_m^2 goes to zero, the gain K_k increases and weights the residual in Eq. 9 more heavily. Specifically:

$$\lim_{\sigma_m^2 \to 0} K_k = H^{-1} \Leftrightarrow \lim_{\sigma_m^2 \to 0} \mathbf{x}_k = \mathbf{z}_k. \tag{11}$$

i.e., the state vector converges to the measurement. On the other hand, increase in the measurement uncertainty is reflected by the growth of R_k yielding

$$\lim_{\sigma_m^2 \to \infty} K_k = [0] \Leftrightarrow \lim_{\sigma_m^2 \to \infty} \mathbf{x}_k = \hat{\mathbf{x}}_k. \tag{12}$$

while the left equation indicates convergence of K_k to *Null* matrix. The prediction is now highly trusted (in comparison to the measurement) resulting the convergence of the state vector to the predicted estimates (i.e. $P_k = \hat{P}_k$ in Eq. 10).

3.2 The Adaptive Kalman filter model

Let us define our state-space vector by $\mathbf{x}_k = [\mathbf{p}_k\, \mathbf{p}_{k-1}\, \mathbf{v}_k]^T$. We include the lagged position vector in the state-space, to allow for recursive update of the target *velocity*. Considering a unit time step, yields the velocity prediction as the displacement between the last two positions of the target (divided by unit time interval). The following prediction and measurement model are, therefore, formed (see Eqs. 4, 5):

$$\begin{aligned} \mathbf{x}_k &= \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \boldsymbol{w_k}, \\ \mathbf{z}_k &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & -1 & 0 \end{bmatrix} \mathbf{x}_k + \boldsymbol{v_k} \end{aligned} \tag{13}$$

Note the notation abuse in Eq. 13 for compact representation. The transition and measurement matrices are practically 6×6. This relation expresses a constant velocity model between each two consecutive frames.

[2] The interested reader is referred to [40] for more details on Kalman filter model.

Variation in the motion and appearance of the target affects the prediction and measurement reliability. Between the two extreme cases of perfect appearance match and occlusion lies a continuous domain (e.g. partial occlusion) where the decrease of confidence in the measurement should be compensated by prediction. We adjust our Kalman filter adaptively by controlling σ_m^2 in time, while keeping the prediction covariance σ_p^2 constant. Since the *appearance* is a strong feature for visibility of the target, we use the appearance dissimilarity metric γ_r as a confidence measure for control of the Kalman filter. An admissible function for the controller $\sigma_m^2(\gamma_r)$ must fulfill the following set of conditions:

$$\sigma_m^2 : \gamma_r \in [0,1] \rightarrow \mathbb{R}^+ \tag{14}$$

$$\lim_{\gamma_r \to 0} \sigma_m^2 = 0 \tag{15}$$

$$\lim_{\gamma_r \to 1} \sigma_m^2 = \infty \tag{16}$$

$$\sigma_m^2(x) \geq \sigma_m^2(y), \quad \forall x > y \tag{17}$$

$$\sigma_m^2 \in C^0[0,1] \tag{18}$$

The first term in Eq. 14 assures non-negativity for the uncertainty measure σ_m^2. The second and third requirements specified in Eqs. 15–16 comply with Eqs. 11–12 implying that low dissimilarity in the appearance ($\gamma_r \to 0$) yields state estimations based mainly on the measurement while for high dissimilarity scores ($\gamma_r \to 1$), the tracking is dominated by prediction. The next constraint in Eq. 17 requires the control function to be monotonically increasing, since loss of confidence in the appearance match should be compensated by higher uncertainty in the measurement and vice versa. The last condition in Eq. 18 demands for *continuity*, since between the stage of appearance and full occlusion lies a continuous domain, where, for instance partial occlusion can be characterized by an intermediate confidence measure. Note that previously used binary functions [14, 19, 41] do not obey this condition, often resulting in unstable controllers (as will be demonstrated in the experimental tests).

Obviously the stated constraints still allow for a large space of admissible functions. Setting of additional constraints depends on definition of the optimal control function and is application-dependent. Here we approach this problem empirically and suggest the following function satisfying the stated requirements (14–18):

$$\sigma_m^2 = \begin{cases} 0 < \lambda_1 \ll 1 & 0 \leq \gamma_r \leq \gamma_1 \\ \text{linear} & \gamma_1 < \gamma_r \leq \gamma_2 \\ \text{exponential} & \gamma_2 < \gamma_r \leq \gamma_3 \\ \lambda_2 \gg 1 & \gamma_r > \gamma_3 \end{cases} \tag{19}$$

where λ_1, λ_2 and γ_1, γ_2, γ_3 are constants. The suggested function, depicted in Fig. 3 consists of four domains. According to Eq. 19, low dissimilarity values $0 \leq \gamma_r \leq \gamma_1$, are associated with decreased uncertainty in the measurement, while at high dissimilarity region $\gamma_3 < \gamma_r < 1$, the controller commands a high value (a practical bound) for measurement uncertainty. Note that the two constant branches determined by λ_1, λ_2 are set to avoid numerical instability. As for the central region one may choose an exponential function (i.e. linear in logarithmic scale). Although such function showed high performance in our tests it was found that a small linear region near $\gamma_r \approx \gamma_1$ attenuates the rapid exponential change and yields more robust results. To present the practical advantage of the above approach, we use a constant set of parameters showing the robustness of this setting to various scenarios.

4 Computational complexity

Considering M as the size of the ROI search window, the complexities involved with each stage (per-frame) are

1. Template matching.
2. Rectification of NCC.
3. Calculation of the motion prior.
4. Determination of the measurement uncertainty.
5. The Kalman Filter computation.

Steps 1–4 are carried out at $\mathcal{O}(M)$ operations while step 5 requires number of operations proportional to the size of the state vector (6 in this work) and therefore is $\mathcal{O}(1)$ in complexity. The computational complexity of our method is dominated by step 1, namely template matching. Pixel-by-pixel template matching can be very time-consuming. For a template of size N in a search window of size M the computational complexity is $\mathcal{O}(MN)$. However, this can be reduced to $\mathcal{O}(M)$ via efficient algorithms and, therefore, become invariant to the template size [36]. Since our search window is linearly related to the template size, the total complexity per frame is, therefore, $\mathcal{O}(N)$, i.e. linear in the number of pixels in the template. Experimental results

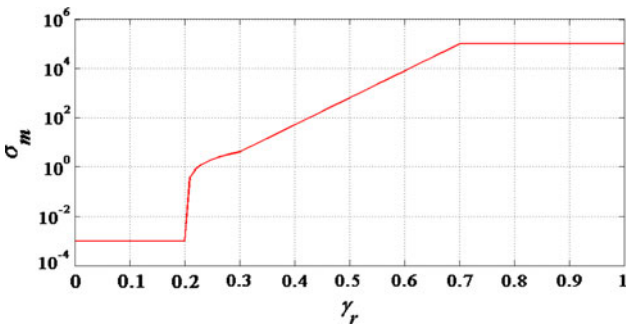

Fig. 3 Variation of the measurement covariance as the function of appearance (dissimilarity) measure. Note that the *plot* is semi-logarithmic

show that the constant associated with this linear order is low enough to provide a high frame rate performance that significantly outperforms present trackers.

5 Experimental results

In this section we assess the performance of our tracker on challenging image sequences in different environments and applications. The test set includes real sequences from public data as well as out-of-the-lab video clips captured by a off-the-shelf webcam. Our lab video clips introduce tracking hazards such as specular highlights, global colour transformations and narrow occlusions.

To gauge absolute performance, the tracking results are also compared with those produced by popular trackers—a colour based mean-shift approach [15], a window-matching technique with Kalman filtering [19] and an advanced histogram-based approach [4]. Furthermore, the validity of two main contributions in this paper, i.e the motion prior and the continuous appearance-based KF, are demonstrated.

The test sets chosen for our evaluation were selected for their challenging characteristics, among which are

1. Illumination variation and specular highlights that entail significant appearance changes.
2. Presence of similar nearby objects (namely distractions).
3. Geometric transformations (including high scale variations).
4. Partial, full or narrow occlusions.
5. Severe noise.
6. Single- and Multi-Object tracking.

Parameter setting: In general, we use a fixed parameter setting with a single user parameter, including the ratio between the template size and search window. Note that in Multi-object tracking we use the same ratio for all the targets. Variation of this value in a small range of [2, 4] was found to be sufficient to fit the parameter setting to all of our tested scenarios. However, in cases without occlusions, execution with constant covariance matrices can yield improved results. This allows a higher tolerance to appearance changes, a property of one of our test cases (see in Sect. 5.3). Otherwise, in such case, high manoeuvres involved with extended appearance change may cause track loss since changes in appearance result in higher measurement covariance, which further leads to increased prediction weights in the KF correction equation.

The parameters in our test bed were assigned the following values: $\beta = 0.75$ (Eq. 3), $\lambda_1 = 10^{-3}$, $\lambda_2 = 10^5$, $\gamma_1 = 0.2$, $\gamma_2 = 0.3$, $\gamma_3 = 0.7$ (Eq. 19) and the prediction uncertainty measure was set to $\sigma_p^2 = 2$. This parameter setting mostly related to our adaptive KF function were obtained by first choosing an anchor point such as $\gamma_r = 0.3$ and fixing its mapped value to $\sigma_m^2 = 4$, according to a reasonable ratio of $\sigma_m^2/\sigma_p^2 = 2$. From this point one can follow the narrative explained in Sect. 3 to reach the aforementioned parameter values.

5.1 Single object tracking

In this section we present the results for scenarios where a single object is tracked. The results for our first four experiments are presented in Fig. 4, showing three representative snapshots of the sequence along with the corresponding plots of the control function σ_m, as evaluated by our method. The first (top) test case in Fig. 4 is a 200-frame sequence captured by a common 480 × 640 pixel webcam. This sequence exhibits significant variations in target appearance. Note, for example, the effect of highlights in the target's image shown in the second snapshot and global colour variations in the frames, caused by the camera imperfections. This scenario additionally demonstrates a case with presence of a distraction (i.e., target-like regions). Despite these confounds, tracking is successful. Note how the tracking without motion prior fails on the very same sequence.

Our second case (in the second row of the figure) is a typical human tracking scenario from the public PETS data base.[3] The target is successfully tracked along a dynamic scene and is not lost despite its occlusion behind the green sign.

The third, semi synthetic, test case in Fig. 4 presents a scenario with repetitive occlusion a.k.a *narrow* occlusion. Here a toy car accelerates and decelerates behind a synthetic fence. Again, the target is successfully tracked along the entire sequence overcoming this periodic (partial) occlusion, varying velocities and motion blur (as observed in the target instantaneous appearance at the top right corner).

In the next experiment we examine our method on the *Distracted Pen* sequence while contaminated with severe noise (with Gaussian distribution). The results show that despite this high level of noise the object is tracked successfully. For full video clip results we refer the reader to the project page at [3]. The last column in Fig. 4 shows the control function (measurement covariance) and the appearance similarity measure. Note the correlation between the two, corresponding to events in the sequence, e.g., occlusion, abrupt appearance changes, etc.

Next, in Fig. 5, we present the results of our method on three longer sequences (230–1,500 frames long), all involving tracking a ground vehicle, either from stationary

[3] Publicly available at http://www.cvg.rdg.ac.uk/PETS2009/a.html.

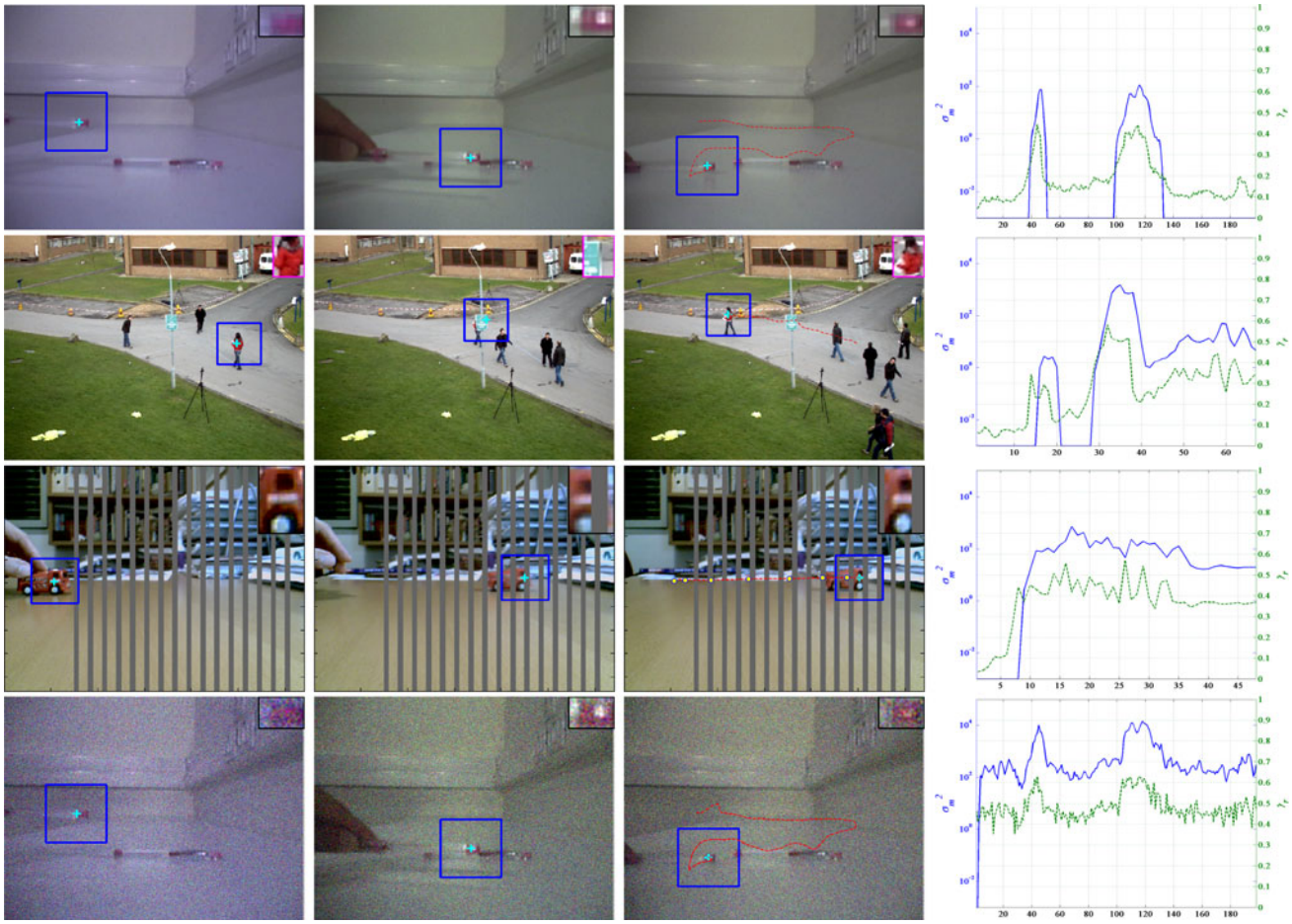

Fig. 4 Single object tracking. The *inset* in the *left column* shows the first frame with the enlarged target appearance superimposed on the *top right corner*. The *second* and *third columns* show an intermediate and the *last frame*. Note the correct target labelling despite various confounds. The *last column* presents the plot of log-uncertainty and the appearance dissimilarity measure. Note their correlation and the dynamic ranges. The occlusion responses are clearly seen in the *plots*. *First row*: Distracted Pen - Indoor scenario captured by a standard web cam. Frames #1,105, 200 are shown. Note the high appearance variations due to specular highlight, and global colour variations. *Second row*: Crowd from the public data base PETS 2009. Frames #445, 479, 513 are shown. Note the pose change and occlusion. *Third row*: Narrow Occl. A semi-synthetic test of a narrow occlusion scenario. Frames #1, 28, 50 are shown. Note the periodic partial occlusion, motion blur and accelerated motion validated by the yellow circles plotted at a constant time interval. Fourth row: Test under severe noise: contaminated with i.i.d Gaussian noise of zero mean and 25 intensity level STD. The resulting clips are available at [3]

or airborne controlled cameras. The first video is from the Next Generation Simulation (NGSIM) Peachtree street.[4] In this case the target is just 24 × 16 pixels in size and it undergoes occlusion, deceleration to full stop and initiating motion again.

The second video is extracted from YouTube named Dallas Police Chase[5] to describe an aerial car chase, where the target is in low contrast and passes through multiple long occlusions. One can observe the smooth transitions between visible and occlusion domains in the video result [3].

Finally, the third successfully tracked target is embedded in a 1,500 frames (352 × 240) long video of the DARPA Vehicle captured by a camera from an aerial carrier[6] (third row in Fig 5). Here, significant changes in velocity, scale (up to factor 5), and view angle are observed. The zooming effect introduces a rapid change in the target appearance noticeable in Fig. 5.

[4] Used to be at http://ngsim.fhwa.dot.gov.

[5] Available at http://www.youtube.com/watch?v=omI094GcZcw&feature=related.

[6] Used to be at http://www.vividevaluation.ri.cmu.edu/datasets/datasets.html.

Fig. 5 Single object tracking. Second test set including car tracking scenarios involving a long sequence, background clutter, accelerating targets and long occlusions. Inset in the first (*left*) frame shows the reference model (*top right corner*). Other columns depict intermediate and the last frame along with the instantaneous target instance at the top right. *First row*: NGSIM sequence, 230 frames long. Frames #1,12,25,230 are shown. The superimposed *curve* in the last frame shows the tracked path (**circles** plotted at 5 frame intervals). *Second row*: Dallas Police Chase, aerial sequence. 298 frames long, acquired from YouTube. Frames #1, 266, 279, 298 are shown. Note the long occlusions and low target contrast. *Third row*: DARPA Vehicle, aerial sequence, 1,500 frames long. Frames #1, 500, 1,300, 1,500 are shown. Note the drastic changes in scale and viewing angles

5.2 Quantitative performance

In this section we compare our approach to three other methods. Our metric is the fraction of the video sequence in which the target was tracked successfully, i.e., the ratio of the number of successfully tracked frames to the total number of frames. Tracking is considered to be lost at a point where a drift is observed and the evaluated target location missed the true position by a distance of at least one template size. Using this criterion, we quantitatively compare our method by the sequences presented in Figs. 4 and 5. In particular, the tracking results were compared with a colour mean shift tracker[7] in [15] (without occlusion handling capability), FragTrack[8] that uses an advanced histogram-based matching with image fragments [4] and the Kalman filter-based method of [19]. The FragTrack considers grey level sequences and has been shown to be very efficient in [4, 5]. Partial occlusions are handled here by patch-based matching.

The comparison is further elaborated by two variants of our approach, one with disabled motion prior and another where the *continuous* adaptive KF is replaced with a *binary* controller. The results of this performance assessment are summarized in Table 1. For fair comparison the Mean-Shift comparison is conducted on the non-occluding sequences. However, the FragTrack approach [4] proposes a partial occlusion capability and, therefore, participates also in the Narrow occlusion comparison. Note that the approach in [19] is an occlusion handling method as well as the variants of our proposed method.

As can be seen, the proposed tracker exhibits the best results with flawless tracking record in all the test sequences while failures are observed in the compared trackers. Considering just the non-occluding scenarios, the mean-shift tracker still exhibits poor performance in all test cases due to intolerance to dynamic appearances and noise. Although Flavio et al. [19] make use of Kalman filter, it still obtains inferior performance due to poor robustness of the SSD similarity measure used for appearance matching, and lack of a prior to regularize the energy space. The binary adaptive control in [19] further yields track losses, particularly under occlusions. The FragTrack [4] shows improved results with respect to aforementioned

[7] Code used to be at http://www.cs.bilkent.edu.tr/ismaila/MUSCLE/MSTracker.htm.

[8] Code available at http://www.cs.technion.ac.il/~amita/fragtrack/fragtrack.htm.

Table 1 Quantitative performance comparison with different trackers determined by the percentage of successfully tracked frames in the sequence

Sequence	Mean-shift	Flavio [19]	FragTrack [4]	Binary control	Without prior	Our method
Distracted pen	43	18	53	100	55	100
Crowd—PETS 2009	–	47	100	100	100	100
Narrow Occl.	–	100	30	100	100	100
NGSIM	–	11	–	35	6	100
Dallas police chase	–	84	–	94	13	100
DARPA vehicle	20	20	100	66	27	100
Noised dist. pen	15	14	12	68	100	100

Note that among this comparison the indicated Mean-Shift method lacks occlusion handling while FragTrack [4] is capable of handling partial occlusions. For fair comparison, scenarios where track failure are caused due to lack of capability are assigned by dash

methods. However, while this tracker well succeeds in human surveillance and copes with appearance changes (e.g., in DARPA data set), it tends to fail in other scenarios. Failures of this tracker are caused due to comparably small size of the target, lacking sufficient statistics, overlook of colour information and sensitivity to noise (see *Noised Dist. Pen example* in Table 1).

Our partial method with binary KF control fully succeeds in 3 out of 7 scenarios, with failures predominantly occurring in the NGSIM and the DARPA sequences. Removing the motion prior component from our method (but keeping the KF) yields failures in 4 sequences. The results in Table 1 show, therefore, the stabilizing influence of our motion prior on the KF while coping with challenging tracking scenarios.

5.3 Multi-object tracking

Whilst the system we describe was intended for single object tracking. The low computational load involved with our method can be exploited to track multiple objects simultaneously. In this section we demonstrate this capacity and present a quantitative analysis for scenarios that incorporate several objects interacting and overlapping in the field of view. Fig. 6 shows sample results for three publicly available data sets: one from CAVIAR data set[9] another from PETS 2009[10] and a test set called *Town Center*[11] recently introduced in [7].

Consider first the sequence at the two top rows in Fig. 6, describing two women walking down a corridor while a third person passes through, occluding each in turn. The second sequence runs multiple-object tracking on the example discussed in Sect. 5.1 from PETS 2009. In this case the three pedestrians are successfully tracked through the sequence (as always, tracked templates are extracted from the first frame and kept constant). Another scenario deals with human surveillance from a stationary HD (1,920 × 1,080/25 fps) camera viewing on a crowded street. In here, we track six individuals through 300 frames while a bird flying before the camera imposes a large occlusion. The results in Fig. 6 show the tracker recovery from the occlusion as well as coping with the varying view angle and scale, as the labelled targets pass from far field to near field camera position. In all cases tracking is sustained, despite occlusion, proximity of similar targets and high variation in appearance.

Three additional scenarios serve our quantitative tracking assessment. These multiple-object tracking sequences are shown by representative snapshots in Fig. 7. The first test from CAVIAR data set demonstrates simultaneous tracking of 8 individuals along 281 frames. This video sequence has been used as a benchmark in previous studies, including those that employ computationally expensive approaches such as [23]. Our tracker succeeds in tracking 7 out of the 8 targets throughout the entire sequence while encountering a single identity switch in this test. Despite the lower computational complexity, this performance is not inferior to the result reported in [23]. The second case involves two separate partial sequences, examined in [6], with 10–13 ping-pong balls entering and exiting the field of view (FOV). The balls were manually labelled on the entrance and removed as they exit the FOV. These two sequences incorporate several close passes and mutual impacts between the visually similar balls, raising a tracking challenge, particularly in respect of identity switches. The results reveal that the suggested tracker copes reasonably with these confounds (see clips at [3] and quantitative measures in Table 2), successfully tracking the balls. Despite the extreme and sudden motions, violating our KF model the track is sustained (see frame 55 at ping-pong balls 732-835).

[9] Publicly available at http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/.

[10] Publicly available at http://www.cvg.rdg.ac.uk/PETS2009/a.html.

[11] Publicly available at http://www.robots.ox.ac.uk/ActiveVision/Research/Projects/2009bbenfold_headpose/project.html#datasets.

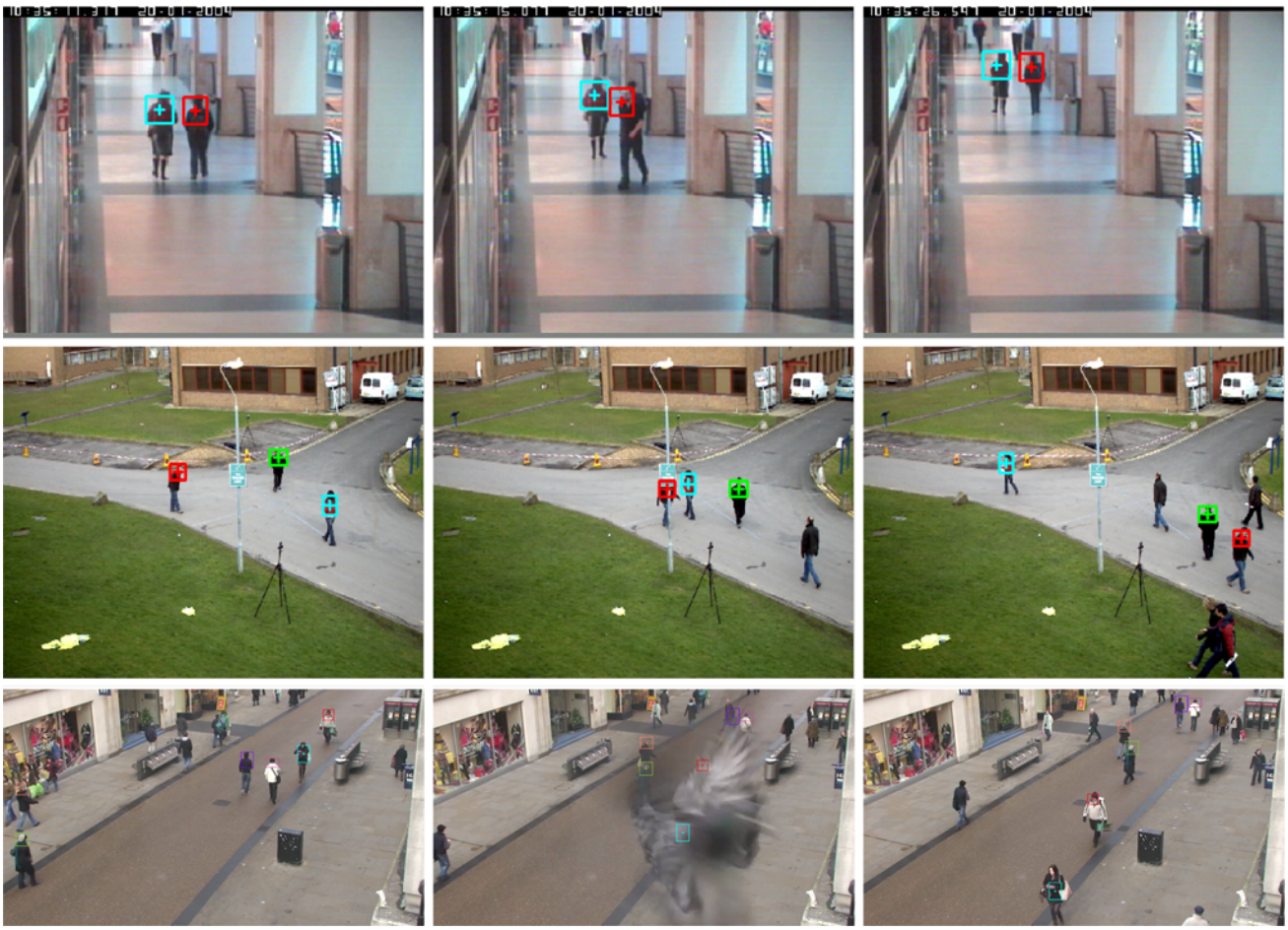

Fig. 6 Multi object tracking. *Top*: CAVIAR Enter Exit. *Middle*: PETS Crowd. *Down*: Town Center from [7]: Six individuals are tracked along 300 frames, incorporating a severe occlusion where a bird is blocking the line of sight, see movie and more results at [3]

Fig. 7 Snaphots of the sequences used for MOTA assessment. From *left* to *right*: CAVIAR One Stop Move, Pingpong balls (two sequences) from [6] and Monterey Car Race from YouTube

The last scenario is 783 frames long from YouTube, presenting a car race as viewed from a stationary camera. The video captures 27 vehicles successively entering the frame on top right corner and exiting on the low left. Since targets has not gone under occlusion, fixed covariance values (namely non-adaptive KF) was set to maximize performance. This allows a higher tolerance to appearance changes as this appears extensively here. Despite the significant change in the viewing angles and scale, the tracker succeeds to follow all the vehicles, except one, along the course (see movie at [3]). Note that since there is no template updating in our tracker, the appearance changes in

Table 2 Examples of MOTA tracking performance for our approach and two comparing methods of [19] and a variant of ours without the motion prior

Sequence	Flavio (%)	Without prior (%)	Our approach (%)
CAVIAR one stop move	71.7	82.2	91.5
Ping-pong balls (435-593)	67.5	91.6	99.1
Ping-pong balls (732-835)	92.4	92.4	92.4
Monterey car race	85.6	88.9	98.0*

* This result corresponds to constant (non-adaptive) covariance matrices. Execution with standard adaptive KF yields 90.6 % MOTA measure

the view angle cause the designated patch on some targets to move on the object. Evidently, the most similar patch, in terms of our new measure, still designates the target (see results at [3]). Small targets are more vulnerable to this effect than larger templates. Indeed, re-execution choosing small patches for all the targets resulted a 2 % decrease in the corresponding MOTA measure.

Next, we assess the performance of the suggested tracker in terms of MOTA (Multiple Object Tracking Accuracy) measure [8]. This is a combined measure which takes into account false positives, false negatives and identity switches (see [8] for details). Table 2 contains the the corresponding MOTA measures along with the figures obtained from two comparing methods of Flavio et al. [19] and a variant of our approach without the motion prior. The results show the improvement associated with our scheme as well as the positive impact of the motion prior, particularly restricting identity switches.

Although not targeted for multi-object tracking since multi-object interactions are ignored (in opposed to e.g., [28]) the obtained MOTA figures demonstrate the high capability of the suggested approach.

5.4 Runtime

Execution time is an important property of practical tracking applications and also a motivation for our proposed approach. A key contributor to our high frame rate is the computational efficiency expressed in the linear complexity (w.r.t the template size). We hereby compare our runtime with figures reported on several recently published methods. Prior to detail on the runtime figures the corresponding complexities of these methods are described:

Mean shift: The classic mean shift algorithm is time intensive, associated with time complexity given by $\mathcal{O}(Tn^2)$ where T is the number of iterations and n is the number of data points.

Adaptive mean shift [38]: The complexity here is dominated by the mean shift approach (see above) and computational complexity of particle filters, scaling exponentially with the number of particles.

Graph cuts [31]: In graph cuts the complexity is typically $\mathcal{O}(n^2m)$ with n being the no. of nodes, here pixels, and m the no. of edges.

Cascade particle filters [23]: The complexity is dominated by the Particle Filters growing exponentially with the number of particles.

Gaussian approximation [28]: Addresses the problem of interactions between multiple objects. The complexity is dominated by calculations running over all pair of interacting objects.

Novel Prob. observation [24]: This method is based on a probabilistic model. The computational cost is mainly spent in computing colour histograms on 400 particles, although this is a linear complexity but with a large constant.

FragTrack [4]: This method is based on histogram comparison on several sub-patches of the template. The histogram distances are based on Earth Mover Distance (EMD) having a super-cubic complexity.

For the sake of comparison we collect in in Table 3 the run-time of several state-of-the-art methods with the executing platforms, as reported in each manuscript. Note that the comparison for the mean-shift and FragTrack methods are computed here on the same platform. The results show that while the mean-shift obtains the highest frame rate, it is associated with relatively poor results. The FragTrack, on the other hand, shows improved tracking results in expense of higher computational cost, while the proposed method is shown to be superior than FragTrack in both accuracy and computational speed.

Although the computation speeds correspond to non-identical platforms, the devices in Table 3 are mostly similar. Furthermore, our results on Pentium 1.86 GHz shows a high frame rate on a device inferior to most of the others. The results show speed ups of $\times 4 - \times 45$ (excluding the naïve mean-shift approach with highly inferior performance—cf. Table 1). This observation is reinforced by the above complexity analysis.

The suggested tracker, therefore, runs in a high frame rate mainly due to its low complexity (see Sect. 4). Typical performance averages over 160–180 frames per second for template size of 20×20 pixels and search window of 60×60. These template and search window sizes were

Table 3 computation frame rate as reported by the authors of each method

Sequence	Image size	fps	Platform
Adaptive mean-shift [38]	640 × 480	24	Intel Centrino 1.6 GHz laptop, 1GB RAM
Graph cuts [31]	360 × 300	4	Standard PC
Cascade particle filters [23]	320 × 240	30	Intel PentiumD 2.8 GHz
Gaussian approximation [28]	720 × 576	43	Intel PentiumD 3 GHz, 2GB RAM
Novel Prob. observation [24]	320 × 480	30–40	Intel Pentium IV 3.2 GHz, 512MB RAM
Mean-shift [15]		1,000	
FragTrack [4]	640 × 480	1	Intel PentiumR 1.86 GHz, 1.5GB RAM
Our approach		**165**	
Our approach	1,920 × 1,080	**180**	Intel PentiumD 3.4 GHz, 3GB RAM

found to be sufficient for a stable track in all of our test cases. These frame rates correspond to a naïve C/C++ implementation based on the OpenCV open source library (Version 1.1) and it excludes image acquisition time and frame decompression where applicable. Frame rates in Table 3 relate to plugging OpenCV with IPP, gaining speed-up of 20 %. Therefore, The suggested tracker highly scores with respect to robustness-speed performance.

6 Summary and discussion

This paper presents a robust yet fast object tracking method for colour video sequences. The first novel feature of our tracker *explicitly* incorporates the motion with the appearance, yielding a measure, tolerant to target colour variations, background clutter, noise and distractions appearing in the course of track. The kinematic prior yields a smoother cost function for similarity measure tending to peak on the target location. This gamut is a key factor in our method allowing for a small patch of the target to be distinguishable and successfully tracked with a very low computational cost.

Our tracker is further endowed with an adaptive Kalman filter to handle partial and full occlusions and to cope with noise. The new adaptive KF is controlled by a mapping of the appearance similarity score considered as a strong measure for visibility. We analyse the requirements from a general adaptive KF control function when used for tracking and infer a set of constraints for the controller. One important constraint is the continuity of this function coping successfully with the failure prone to the transition domain, between visibility and occlusion. Upon these constraints a control function is suggested that yields improved results. We further demonstrate the validity of this conclusion by several tests.

The resultant tracker is demonstrated on various scenarios and target types. While the traditional mean-shift approach is superior in run-time, it exhibits a very low robustness. When compared with two additional tracking methods, the suggested approach showed improved results in stability, attainability as well as computational efficiency. These performances were obtained without being tailored to any specific application. Experimental tests from celebrated data sets as well downloaded YouTube videos show that our method well handles common tracking confounds such as varying appearance, dynamic backgrounds, changes in scale and view point, as well as partial and full occlusions. The robustness of our tracker is further emphasized by tolerance to severe image noise.

We show that tracking remains successful without the need for a template updating scheme, even in relatively long sequences. This can also fit into a paradigm where template updates are performed on temporally distant frames to lower the chance for drift. However, this should not obscure the fact that our tracking performance depends on the choice of a rather distinguishable template. While being capable to cope with common tracking confounds, failure reasons remain as drifting, abrupt appearance changes or severe accelerations, particularly before the KF stabilization (i.e. near the entrance to the FOV). These conclusions reflect our idea that common tracking scenarios contain a limited range of hazards, thus allowing for our relatively robust and highly efficient method yield satisfactory results.

A direction for improvement is to consider a dynamic search window. One can vary the size of the search domain according to the uncertainly, reflected in the adaptive measurement covariance. The impact of this enhancement on the tracking success is yet to be analysed, considering the fact that larger search window adds to the chance for false positives.

The linear complexity of the suggested method along with the allowed small size of the reference template in our approach creates a high computational efficiency. The resulting tracker runs, therefore, in excess of 180 frames per seconds (fps) on a standard CPU. In spite the high-speed performance on a single core, our method can

leverage a multi-core or GPU implementation in two senses:: one speeding up the single object track via distributing the computation of NCC and the other, in parallel computation of multi-object tracking.

The suggested method is also easy to implement with common toolboxes (e.g., OpenCV or MATLAB). In practice, the reported computational speeds were obtained by a naïve C/C++ implementation, using standard OpenCV functions.

References

1. The Mean Shift based Tracker code. http://www.cs.bilkent.edu.tr/ismaila/MUSCLE/MSTracker.htm
2. The OpenCV library. http://opencv.willowgarage.com/wiki/
3. The Project Webpage. http://www.cs.bgu.ac.il/rba/Tracking_MPrior_AKF_Results/Project
4. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: Proceedings of the CVPR, vol. 1, pp. 798–805 (2006)
5. Babenko, B., Ming-Hsuan, Y., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE TPAMI **33**(8), 1619–1632 (2011)
6. Ben-Shitrit, H., Berclaz, J., Fleuret, F., Fua, P.: Tracking multiple people under global appearance constraints. In: Proceedings of the ICCV (2011)
7. Benfold, B., Reid, I.: Stable multi-target tracking in real-time surveillance video. In: Proceedings of the CVPR (2011)
8. Bernardin, K., Elbs, A., Stiefelhagen, R.: Multiple object tracking performance metrics and evaluation in a smart room environment. In: Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV, vol. 90 (2006)
9. Beymer, D., Konolige, K.: Real-time tracking of multiple people using continuous detection. In: Proceedings of the ICCV-Frame Rate Workshop (1999)
10. Boris Babenko Ming-Hsuan Yang, S.B.: Visual tracking with online multiple instance learning. In: Proceedings of the CVPR (2009)
11. Brodia, T., Chellappa, R.: Estimation of object motion parameters from noisy images. IEEE TPAMI **8**(1), 90–99 (1986)
12. Brown, R.G.: Random Signal Analysis and Kalman Filtering. Wiley, Hoboken (1983)
13. Bruce, A., Gordon, G.: Better motion prediction for people-tracking. In: Proceedings of the ICRA (2004)
14. Chu, C.T., Hwang, J.N., Wang, S.Z., Chen, Y.Y.: Human tracking by adaptive Kalman filtering and multiple kernels tracking with projected gradients. In: Proceedings of the Fifth ACM/IEEE International Conference on Distributed Smart Cameras, pp. 1–6 (2011)
15. Comaniciu, D., Ramesh, V., Meer, P.: Kernel based object tracking. IEEE TPAMI **25**, 564–575 (2003)
16. Daum, F.: Non-particle filters. SPIE—Signal and Data Processing of Small Targets **6326**, 614–623 (2006)
17. Dellaert, F., Thorpe, C.: Robust car tracking using Kalman filtering and bayesian templates. In: Conference on Intelligent Transportation Systems (1997)
18. Fieguth, P., Terezopoulos, D.: Color-based tracking of heads and other mobile objects at video frame rates. In: Proceedings of the CVPR, pp. 21–27 (1997)
19. Flávio, B.: Window matching techniques with Kalman filtering for an improved object visual tracking. In: Proceedings of the IEEE Conference on Automation Science and Engineering (2007)
20. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Proceedings of the ECCV (2008)
21. Hii, A.J.H., Hann, C.E., Chase, J.G., Van- Houten, E.W.: Fast normalized cross correlation for motion tracking using basis functions. Comput. Methods Programs Biomed. **82**(2), 144–156 (2006)
22. Jameson, D., Huvich, L.: Complexities of perceived brightness. Science **133**, 174–179 (1961).
23. Li, Y., Ai, H., Yamashita, T., Lao, S., Kawade, M.: Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans. IEEE TPAMI **30**(10), 1728–1740 (2008)
24. Liang, D., Huang, Q., Yao, H., Jiang, S., Ji, R., Gao, W.: Novel observation model for probabilistic object tracking. In: Proceedings of the CVPR, pp. 1387–1394 (2010)
25. Matei, B., Sawhney, H., Samarasekera, S.: Vehicle tracking across nonoverlapping cameras using joint kinematic and appearance features. In: Proceedings of the CVPR (2011)
26. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. IEEE TPAMI **26**(6), 810–815 (2004)
27. Mauthner, T., Donoser, M., Bischof, H.: Robust tracking of spatial related components. In: Proceedings of the ICPR (2008)
28. Medrano, C., Martinez, J., Igual, R.: Gaussian approximation for tracking occluding and interacting targets. J. Math. Imaging Vis. **36**(2), 241–253 (2010)
29. Mileva, Y., Bruhn, A., Weickert, J.: Illumination-robust variational optical flow with photometric invariants. In: Proceedings of the DAGM Symposium (2007)
30. Oussalah, M., Schutter, J.D.: Adaptive Kalman filter for noise identification. In: Proceedings of the International Conference on Noise and Vibration Engineering, pp. 1225–1232 (2000)
31. Papadakis, N., Bugeau, A.: Tracking with occlusions via graph cuts. IEEE TPAMI **33**(1), 144–157 (2011)
32. Porikli, F.: Achieving real-time object detection and tracking under extreme conditions. J. Real-Time Image Proc. **1**, 33–40 (2006)
33. Sebastian, P., Voon, Y.V.: Tracking using normalized cross correlation and color space. In: Proceedings of the International Conference on Intelligence and Advanced Systems (2007)
34. Shahrimie, M., Asaari, M., Suandi, S.A.: Hand gesture tracking system using adaptive Kalman filter. In: Proceedings of the International Conference on Intelligent Systems Design and Applications, pp. 166–171 (2010)
35. Sun, J., Li, Y., Kang, S.B., Shum, H.Y.: Symmetric stereo matching for occlusion handling. In: Proceedings of theInternational Conference on Computer Vision and Pattern Recognition, pp. 399–406 (2005)
36. Tsai, D.M., Lin, C.T.: Fast normalized cross correlation for defect detection. Pattern Recogn. Lett. **24**, 2625–2631 (2003)
37. Van Leuven, J., Van Leeuwen, M., Groen, F.: Real-time vehicle tracking in image sequences. In: Proceedings of the IEEE Instrumentation and Measurement Technology (2001)
38. Wang, J., Yagi, Y.: Adaptive mean-shift tracking with auxiliary particles. IEEE Trans. Syst. Man Cybernet. B **39**(6), 1578–1589 (2009)
39. Wang, Y., Teoh, E.K., Shen, D.: Lane detection and tracking using B-snake. Image Vis. Comput. **22**(10), 269–280 (2004)
40. Weltch, G., Bishop, G.: An introduction to the Kalman filter. Technical Report 95-041, Department of Computer Science University of North Carolina at Chapel Hill (2006)

41. Weng, S.K., Kuo, C.M., Tu, S.K.: Video object tracking using adaptive Kalman filter. J. Vis. Commun. Image Represent. **17**, 1190–1208 (2006)
42. Williams, R.L., Lawrence, D.A.: Linear State-Space Control Systems. Wiley, USA (2007)
43. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. **38**(4) (2006)

Author Biographies

Rami Ben-Ari is a computer vision expert at Orbotech Ltd., the world leader in PCB inspection and imaging. He received his B.Sc. (1990) and M.Sc. (1993) degrees in Aerospace Engineering from Israel Institute of Technology-Technion, and Ph.D. degree in Applied Mathematics from Tel-Aviv University, in 2008. During 2009, he was an associate researcher at the Technion-Israel Institute of Technology and Ben-Gurion University. His research interests include PDE methods in computer vision, advanced numerical schemes adapted to parallel platforms (GPU), statistical filtering and visual tracking.

Ohad Ben-Shahar received the B.Sc. and M.Sc. degrees in Computer Science from the Technion, Israel Institute of Technology, and the Ph.D. degree in Computer Science from Yale University, New Haven, Connecticut, in 2003. He is an associate professor in the Department of Computer Science at Ben-Gurion University of the Negev and the Director of the BGU Interdisciplinary Computational Vision Lab. His main area of research is in computational vision and image analysis, where he is focusing primarily on issues related to the differential geometrical foundations of perceptual organization and early vision. His work is explicitly multidisciplinary and his computational research is often endowed by investigations into human perception, visual psychophysics and computational neuroscience of biological vision. He is the recipient of the 2007 Psychobiology Young Investigator Award and his research is funded by the Israel Science Foundation (ISF), the US Air Force Office of Scientific Research (AFOSR), the Deutsche Forschungsgemeinschaft (DFG) in Germany, the US National Science Foundation (NSF), the US National Institute for Psychobiology and the European Union (FP7).