SIViP (2015) 9:1567–1585
DOI 10.1007/s11760-014-0612-0

ORIGINAL PAPER

Correlation, Kalman filter and adaptive fast mean shift based heuristic approach for robust visual tracking

Ahmad Ali · Abdul Jalil · Javed Ahmed · Muhammad Aksam Iftikhar · Mutawarra Hussain

Received: 18 February 2013 / Revised: 27 December 2013 / Accepted: 2 January 2014 / Published online: 31 January 2014

Abstract Correlation tracker is computation intensive (if the search space or the template is large), has template drift problem, and may fail in case of fast maneuvering target, rapid changes in its appearance, occlusion suffered by it and clutter in the scene. Kalman filter can predict the target coordinates in the next frame, if the measurement vector is supplied to it by a correlation tracker. Thus, a relatively small search space can be determined where the probability of finding the target in the next frame is high. This way, the tracker can become fast and reject the clutter, which is *outside* the search space in the scene. However, if the tracker provides wrong measurement vector due to the clutter or the occlusion *inside* the search region, the efficacy of the filter is significantly deteriorated. Mean-shift tracker is fast and has shown good tracking results in the literature, but it may fail when the histograms of the target and the candidate region in the scene are similar (even when their appearance is different). In order to make the overall visual tracking framework robust to the mentioned problems, we propose to combine the three approaches heuristically, so that they may support each other for better tracking results. Furthermore, we present novel method for (1) appearance model updating which adapts the template according to rate of appearance change of target, (2) adaptive threshold for similarity measure which uses the variable threshold for each forthcoming image frame based on current frame peak similarity value, and (3) adaptive kernel size for fast mean-shift algorithm based on varying size of the target. Comparison with nine state-of-the-art tracking algorithms on eleven publically available standard dataset shows that the proposed algorithm outperforms the other algorithms in most of the cases.

Keywords Object tracking · Template drift · Clutter · Occlusion · Fast mean shift · Kalman filter · Normalized correlation

1 Introduction

The aim of visual object tracking is to consistently labeling the position of the object of interest (OI) in the consecutive frames of a video [1]. It is an important field of computer vision and its usefulness as well as usability spans from commercial to military applications such as visual surveillance and security systems [2–6], activity recognition [7–9], motion capture and animation [10,11], video games [12], vehicle tracking [13,14], traffic monitoring [15], human-machine interaction [16], industrial robotics [17], and medical diagnosis systems [18–20]. Although a lot of research has been carried out in this field for many years [21,22], it is still an open avenue for computer vision community due to various complex issue; a few of them include clutter (presence of other objects or high texture in background), occlusion (hiding of OI by other objects in the scene), complex motion, out-of-plane rotation, variation in size of the object as it moves, and rapid change in its appearance, especially in case of non-

A. Ali (✉) · A. Jalil · M. A. Iftikhar · M. Hussain
Pakistan Institute of Engineering and Applied Sciences,
Islamabad, Pakistan
e-mail: ahmadali1655@hotmail.com

A. Jalil
e-mail: jalil@pieas.edu.pk

M. A. Iftikhar
e-mail: aksam.iftikhar@gmail.com

M. Hussain
e-mail: mutawarra@pieas.edu.pk

J. Ahmed
NUST Military College of Signals, Islamabad, Pakistan
e-mail: javed@mcs.edu.pk

rigid object. We present correlation, Kalman filter (KF), and adaptive fast mean shift based heuristic approach to make the visual object tracking robust to these issues. The contributions of the proposed approach include (1) integration of correlation, KF and mean-shift tracker, and switching back and forth between the correlation tracker and the KF tracker based on the closeness of the mean-shift tracker results to either measured or predicted object position, respectively, (2) algorithm to update the template according to rate of appearance change of the target, (3) use of variable thresholding for similarity measure in forthcoming image frame based on peak correlation value in the current image frame, and (4) adaptive kernel size for fast mean-shift algorithm according to changing target size.

Correlation based visual template tracking is in use since very start of this field [22–25], and it has shown its strength for long-term tracking session [26,27], but classically, there are a few inherent issues with this approach which are as follows: (1) It is computation intensive, (2) it has serious template drift problem, and (3) it may fail in case of fast maneuvering OI, rapid changes in its appearance, or occlusion and clutter in the scene. These issues are handled, to some extent, by integrating KF with the correlation based tracking and temporarily updating the template [28,29]. Considering the position of peak correlation value as position of OI in the current image frame, KF predicts its position in the upcoming image frame. Thus, a relatively small search window can be determined where the occurrence of OI is highly likely [30]. Moreover, KF helps tracker get out of occlusion faced by the target. Occlusion is assumed to be happening if correlation value of the target in search window falls below a threshold. Therefore, choosing the right value of the threshold is very important. Many papers [27,29–32] use fixed threshold, but it does not work as complexity of tracking scenario changes. We propose a new method for adaptive threshold based on the current frame peak correlation value. During occlusion, correlation based measurement vector is ignored and KF-predicted vector is used as next measurement vector. This way, the tracker becomes (1) fast, (2) its performance remains safe from a lot of clutter *outside* the search window, and (3) it shows robustness to occlusion as well. However, due to all or any of the above-mentioned issues occurring *inside* the search window, tracker may provide wrong measurements to KF, which in return generate wrong predictions and whole tracking process is deteriorated. Now the question arises, how to get to know automatically that this situation has happened? In order to answer this question, we propose to use difference between KF-predicted and correlation based measured coordinates; this difference is checked against another adaptive threshold based on the target size. The next question that comes, intuitively, in mind is whether the tracker should go with the predicted or the measured coordinates? We use adaptive fast mean-shift algorithm to answer this question. It is applied to find out the clusters in difference of the search windows in the two consecutive frames of the video. These clusters are moving regions in the video, and thus, they become potential candidates for being OI. Nearest neighborhood technique is used to check whether a candidate OI is close to predicted or measured coordinates. In such a way, fast mean-shift algorithm acts as an arbitrator for the validity between KF- and correlation based results. Thus, KF can be protected from being misled by wrong measurement vector. The size of the kernel for mean shift is set adaptively according to the changing target size. To tackle the issue of rapid change of target appearance, we present a novel method which updates the target model according to rate of appearance change in the target. In general, the proposed tracking strategy can be considered as an ensemble of the three techniques, complementing each other in complex situations. The switching from one technique to other technique is decided heuristically as described above.

The paper is organized as follows: Sect. 2 describes the related work, Sect. 3 explains the proposed visual object tracking algorithm, Sect. 4 presents the experimental results, and Sect. 5 concludes the paper.

2 Related work

Brief summary of different tracking techniques can be studied from [1]. In this section, we describe only the tracking algorithms related to correlation, Kalman filter, and mean-shift algorithms followed by different template updating strategies. Different correlation based similarity measuring metrics, e.g., phase correlation, normalized correlation, and normalized correlation coefficient, are used for visual tracking. Phase correlation has been used by [33,34], and [35] for image registration and tracking, but it is not robust to noise [36] and sometimes produce higher peaks at wrong positions resulting in false alarms [30,37,38]. This problem was overcome by [31] by using edge-enhanced image instead of gray-scale image. Ahmed et al. [39] used extended flat-top Gaussian weighting function with gray-scale image to handle it. Some other papers such as [40–42] also propose algorithms to enhance the performances of phase correlation. All these methods do not produce as much good tracking results in case of changes in appearance, shape, and brightness etc, as normalized correlation does [28,30,31]. Normalized correlation coefficient (NCC) is another widely used similarity measure for object localization [43–47]. NCC imposes constraint of nonuniformity on template and search window. The issue of occlusion handling using NCC was tackled in [29] with the help of Kalman filter. It checks the value of NCC against an empirically determined threshold; if NCC is less than the threshold, it is considered as occlusion and next position of target is calculated by Kalman filter. Similar technique for occlusion handling was used in [27,30] with normalized correlation which is computationally more efficient than NCC in spatial domain and does not restrict the template as well as search window to be nonuniform. It was shown in [28] that normalized correlation produces better results than NCC when edge-enhanced image is used for matching instead of gray-scale images. Ali et al. [32] combined NCC with Kalman filter and fast mean shift to handle complex object motion only; their technique was not robust to clutter and occlusion.

Baleznai et al. used fast mean-shift algorithm for the detection of humans in groups [48,49]. They further extended their work to track humans [50–52] . Wang et al. [53] used multicue fusion-based mean-shift algorithm to track a human in infrared imagery. Sutor et al. [54] presented efficient mean-shift clustering to detect and track humans. Shan et al. [55] proposed mean-shift embedded particle filter for hand tracking. Yilmaz et al. [56] used mean shift with motion compensation to track target in forward-Looking infra red (FLIR) imagery. Comaniciu et al. [57] employed color histogram for real-time visual object tracking of nonrigid objects using mean shift. They used Bhattacharya coefficient as similarity metric to find out the candidate target, obtained by mean-shift algorithm, that is the most similar to OI. Afterward, Comaniciu and Ramesh combined mean shift and KF for object tracking based on color histogram [58]. They used mean-shift iterations to get best candidate target, and KF is used for next target position in the upcoming image frame. When the next frame arrives, mean shift is initialized at the target position predicted from the previous frame. Li et al. [59] suggested adaptive KF with mean shift for object tracking. It adaptively updates the parameters of KF as opposed to previous techniques that keep KF parameters constant. Similar to [57] and [58] color histogram-based target representation is considered in [59]. Since color histogram does not carry spatial information of pixels [60], it is likely to detect a wrong target with similar histogram as that of OI [30]. Therefore, we propose the idea of heuristically combining correlation, Kalman filter, and adaptive kernel fast mean-shift algorithm for better visual tracking results.

2.1 Appearance model updating

Template or appearance model updating is the most important and one of the most difficult steps in a tracking process. By the time, target may suffer from noise, illumination, shape, and appear (especially in case of articulated objects) changes in image plane, so it necessitates the requirement of appearance model updating for a long and successful tracking session. Whenever the template is updated, it is likely to contain some background pixels because spatial location based on peak correlation value might not be a true position of the target; these flawed pixels in template model may result in template walk-off (drift) problem. There are numerous and different template updating schemes available in literature, and each depends upon its underlying tracking algorithm. In this section, we will first discuss the some already presented template updating techniques relevant to correlation based tracking, and then, the proposed algorithm will be presented.

2.1.1 *Naive template updating method*

In this method, template is updated in every next frame, or after a number of frames provided, peak correlation value is greater than a threshold. Equation (1) is the mathematical formulation of this scheme,

$$T_{n+1} = \begin{cases} b_n & \text{if } C_{\text{peak}} \geq t \\ T_n & \text{otherwise} \end{cases} \tag{1}$$

where b_n is the best candidate target for new template in current image, T_n and T_{n+1} show the current and updated templates, respectively, C_{peak} is the peak correlation value, and t is the fixed threshold value. This scheme always assumes b_n as the true target (which is not the case in reality) and completely replaces the current template. Furthermore, there is no mechanism to check whether the template is updated correctly or not. Therefore, it is highly prone to template drift problem as shown in the first row of Fig. 1.

2.1.2 *α-Template updating method*

This method does not replace the current template with the new template at once, rather it introduces a parameter $\alpha, 0 \leq \alpha \leq 1$, to smoothly update the template. Equation (2) explains this method,

$$T_{n+1} = \begin{cases} T_n + \alpha(b_n - T_n) & \text{if } C_{\text{peak}} \geq t \\ T_n & \text{otherwise} \end{cases} \tag{2}$$

For $\alpha = 1$, this method becomes the naive method and if $\alpha = 0$, there will be no template updating. If α is assigned a small value (e.g., 0.02) as in [23,61], it does not cater for rapid changes in appearance of the target due to stagnation to target old state [31]. In order to handle this issue, the idea of using $\alpha = C_{\text{peak}}$ was presented [31]. However, during normal tracking, the value of C_{peak} is greater than 0.9, so it behaves the same way as the naive method does, as shown in the second row of Fig. 1.

2.1.3 *β-Template updating method*

This method was proposed by Ahmed et al. [27,30] and has the same mathematical formulation [see Eq. (3)] as that of α method for template updating. The only difference is the assignment of value to the filter coefficient (which is β in this case). Here, $\beta = 0.15C_{\text{peak}}$, which helps to cope with stagnation as well as drift problem of α-template updating

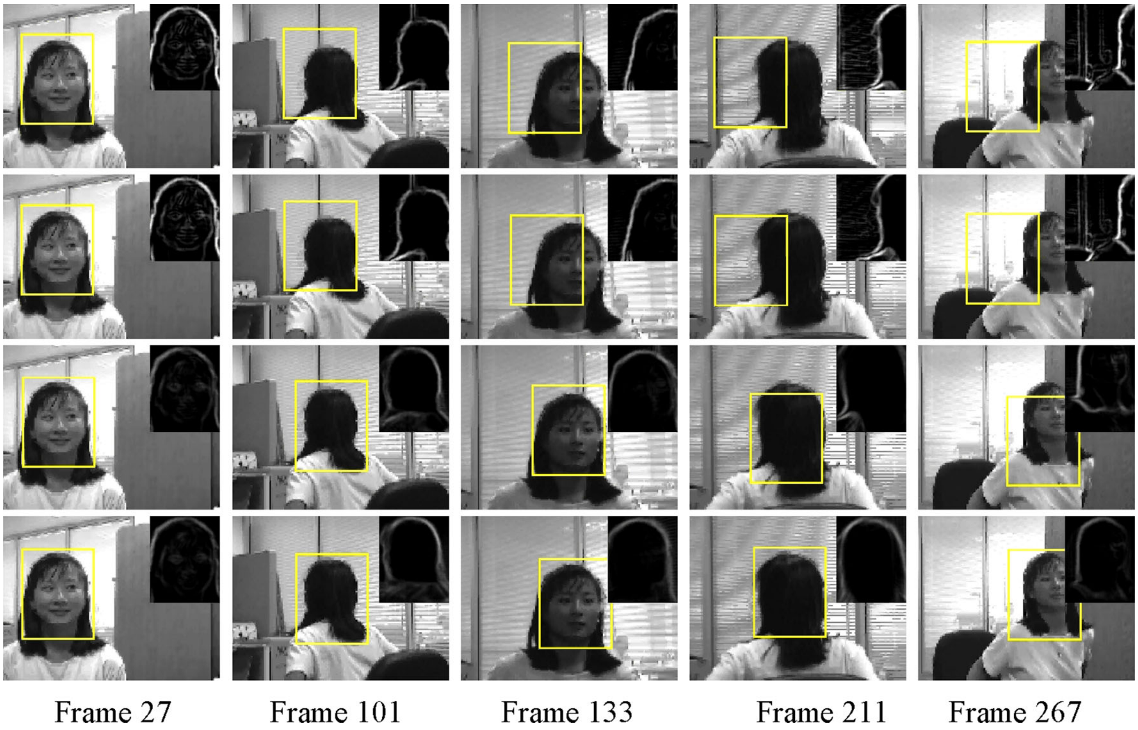


Fig. 1 Comparison of different updating schemes (i.e., Naive, α, and β methods in first, second, and third row, respectively) with the proposed method (fourth row) on girl sequence. It shows that the proposed method has the lowest, negligible template drift and keeps locking the target successfully

method. The performance of this technique is better than above-mentioned both methods as can be verified by the third row of Fig. 1.

$$T_{n+1} = \begin{cases} T_n + \beta(b_n - T_n) & \text{if } C_{\text{peak}} \geq t \\ T_n & \text{otherwise} \end{cases} \tag{3}$$

3 Proposed visual object tracking

The proposed VOT consists of correlation- and KF-based tracking with novel approach of template updating, adaptive thresholding, and adaptive kernel fast mean-shift algorithm. These techniques are heuristically combined to reinforce each other for better performance of robust visual object tracking. The detail of each is given as follows.

3.1 Correlation- and KF-based tracking

In order to initiate the correlation based tracking process, (a) target may be selected by the user, (b) it may already be stored in the system, or (c) it may be selected automatically by some target detection system. The subimage of the target is taken as the template. Edge-enhanced representation of template is used as target appearance model. Search window of the template is also made edge-enhanced for better similarity measure results. The process of edge enhancement starts with Gaussian smoothing to remove the noise in the image, followed by the operations of gradient magnitude, normalization, and thresholding. Interested readers may study [30] for its further detail. The size of search window is not kept constant rather it is made dynamic with the help of KF throughout the tracking session. This way, tracker becomes computationally efficient, rejects a lot of clutter outside the search window, and generates better tracking results. Detailed explanation about the dynamic search window can be studied from [27]. Varying size of the target is an important issue for robust and successful tracking, it is handled by following two processes: (1) correlating the original template as well as 10 % smaller and 10 % larger templates with the search space. The size of the template with highest peak correlation value is considered as the new template size. Similar technique has been used for scale handling in many other papers [27,62–64]. (2) Best-match rectangle adjustment algorithm [65] is used to resize the template according to target size and to keep the target at the center of the template. It divides the template into nine nonoverlapping fragments and checks the statistical properties of each fragment. Combined voting scheme is used for adjustment of best-match rectangle. Thus, it helps to keep target at center and to tackle the problem of template drift, especially in case of tracking an airborne object such as airplane, flying kites, birds, and helicopter. The detail of best-match rectangle adjustment algorithm can be studied from [65]. The template is matched in the search window of next image frame using normalized correlation, and the spatial location of the peak correlation value is considered as the current position of the target in the search window. The matching is considered successful if the peak value of the normalized correlation is greater than a threshold. The reason for using normalized correlation is that it has shown its strength over the other similarity measures for object localization in case of edge-enhanced images [28]. The next step after matching is to update the template. It is explained in the following subsection. Algorithm 1 summarizes the correlation- and KF-based tracking methodology.

Algorithm 1: Correlation and Kalman Filter Tracking

Input: Video sequence of *n* frames, template image of the target, *T*, and target bounding rectangle in the1st frame, *R*

Output: target position in each frame of the video sequence

```
for 1st frame to n frames
    1. Make the template, T, edge enhanced
    2. Extract search window, S
    3. Make the search window, S, edge enhanced
    4. Match T with S using normalized correlation (NC)
    5. $C_{peak} \leftarrow \max(NC)$
    6. Update size of the T
    7. Occlusion handling using Kalman filter
    8. Update T
    9. Output bounding rectangle of T according to $C_{peak}$
end for
```

3.1.1 Proposed template updating method

A good template updating scheme should handle the problems of *template drift* as well as *stagnation to old appearance*. For this, the updating scheme should be such that (1) it may incorporate maximum target changes, i.e., template updating scheme should be dynamic based on the fact that whether target is changing its appearance rapidly or slowly, (2) it should contain as small background as possible, and (3) if the template is poorly updated with some background or noisy pixels, the updating scheme should have the ability to restore the template to a better representation.

In the proposed method, the first template (which is selected by the user) is considered as the most trusty one, and it is kept in buffer throughout the tracking session. Let it be denoted by T_1. The last updated template, say at $(n-1)$th frame, is assumed to contain the maximum changes in the target within itself. It is represented by T_n, where subscript shows frame number where template will be used to find target. It is possible that T_n may have noise, background pixels, sudden illumination changes, or blurriness of the target changing its appearance *heavily and rapidly*. Therefore, the second last template, symbolized by T_{n-1}, is also kept in memory to validate whether the updated template is the correct one or to revert it back to the second last template in case of poor updating. At frame n, we proceed with the process to make template representation better (if needed) followed by the template updating process for the next frame (i.e., $n+1$). The former process consists of two steps. At first, the both templates, T_n and T_{n-1}, are correlated with the search window. We, respectively, represent their peak correlation values as $C_{(n)\text{peak}}$ and $C_{(n-1)\text{peak}}$. If $C_{(n)\text{peak}} \geq C_{(n-1)\text{peak}}$, it is considered that the last updated template is the correct one and $C_{\text{peak}} = C_{(n)\text{peak}}$, otherwise T_n is replaced by T_{n-1} and $C_{\text{peak}} = C_{(n-1)\text{peak}}$. As a second step, T_1 is correlated with the search window. Its peak correlation value is represented by $C_{(1)\text{peak}}$. If T_1 fails to find, at least, its 50 % match in the search window, it is assumed that *slowly occurring* occlusion is being faced by the target, which corrupted both T_n and T_{n-1}. Therefore, we start updating T_n partly by T_1. Equations (4)–(6) describe the process mathematically,

$$a_n = \begin{cases} T_{n-1} & \text{if } C_{(n)\text{peak}} < C_{(n-1)\text{peak}} \\ T_n & \text{otherwise} \end{cases} \tag{4}$$

$$C_{\text{peak}} = \begin{cases} C_{(n-1)\text{peak}} & \text{if } C_{(n)\text{peak}} < C_{(n-1)\text{peak}} \\ C_{(n)\text{peak}} & \text{otherwise} \end{cases} \tag{5}$$

$$d_n = \begin{cases} \omega a_n + (1-\omega)T_1 & \text{if } C_{(1)\text{peak}} < 0.50 \\ a_n & \text{otherwise} \end{cases} \tag{6}$$

where $0 < \omega \leq 1$. We have used $\omega = 0.25$ in our experiments for satisfactory results. The latter process of template updating is mathematically represented by Eqs. (7)–(11).

$$T_{n+1} = \begin{cases} d_n + \gamma(b_n - d_n) & \text{if } C_{\text{peak}} \geq t \\ d_n & \text{if } (C_{\text{peak}} < t) \wedge (f \leq \lambda) \\ \sigma d_n + (1-\sigma)T_1 & \text{if } (C_{\text{peak}} < t) \wedge (f > \lambda) \end{cases} \tag{7}$$

$$f = \begin{cases} 0 & \text{if } C_{\text{peak}} \geq t \\ f+1 & \text{otherwise} \end{cases} \tag{8}$$

$$\gamma = \delta^{*}\Delta C_{\text{ref}} + (1-\delta)\Delta C \tag{9}$$

$$\Delta C_{\text{ref}} = 1 - C_{\text{peak}} \tag{10}$$

$$\Delta C = |C_{\text{peak}} - oldC_{\text{peak}}| \tag{11}$$

where $oldC_{\text{peak}}$ is the peak correlation value of the last frame, $0 \leq \gamma \leq 1, 0 \leq \sigma \leq 1, 0 \leq \delta \leq 1$, and $\lambda > 0$, t is threshold, which is made adaptive by the proposed method. We set the values of these parameters as follows: $\sigma = 0.035$, $\delta = 0.3$, and $\lambda = 3$. The pseudocode for the proposed template updating method is given in Algorithm 2. Equations (6)–(11) are explained as follows.

Case I (When $C_{\text{peak}} \geq t$): In this case, template is updated as weighted average of current template and best match found in the image. The weight, γ, is calculated dynamically as described by Eq. (9). It is made as function of difference of peak correlations, ΔC, in the two latest frames and difference of peak correlation from its upper limit (which is 1), ΔC_{ref} . For an object changing its appearance *heavily and rapidly,* ΔC, calculated by Eq. (11),will be higher, otherwise it will have smaller value. Thus, the template will be updated according to rate of change in appearance of the target. Ideally, updated template should have 100 % or, at least, near

100 % match in the next image. This is achieved by incorporating ΔC_{ref} term in Eq. (9). Furthermore, ΔC_{ref} accelerates the updating process, which is normally slow due to very small value of ΔC in consecutive image frames.

Algorithm 2: Proposed template updating method

Input: Current template, T_n , previous template, T_{n-1}, Initial template, T_1, Search window, previous peak correlation value $oldC_{peak}$
Output: Updated template

```
Initialize $\sigma$, $\delta$, and $\lambda$
Correlate $T_n$ , with the search window to
    calculate $C_{peak}$, and $b_n$
Correlate $T_{n-1}$, and $T_1$ with the search window
    and calculate $C_{(n-1)peak}$, and $C_{(1)peak}$ respectively.
if $C_{peak} < C_{(n-1)peak}$
  $T_n \leftarrow T_{n-1}$
  $C_{peak} \leftarrow C_{(n-1)peak}$
end if
if $C_{(1)peak} < 0.50$
    $T_n \leftarrow \omega T_n + (1 - \omega)T_{n-1}$
end if
$\Delta C_{ref} \leftarrow 1 - C_{peak}$
$\Delta C \leftarrow \mathrm{abs}(C_{peak} - oldC_{peak})$
$\gamma \leftarrow \delta\Delta C_{ref} - (1 - \delta)\,\Delta C$
$oldC_{peak} \leftarrow C_{peak}$
if $C_{peak} \geq t$
  $f \leftarrow 0$
  $T_n \leftarrow T_n + \gamma(b_n - T_n)$
else
  $f \leftarrow f + 1$
  if $f > \lambda$
    $T_n \leftarrow \sigma\, T_n + (1 - \sigma)\, T_1$
  end if
end if
```

Case II (When $C_{\text{peak}} < t$ and $f \leq \lambda$): In this case, we do not update the template considering that the template has been occluded. If the situation holds for a certain number of frames, f, then it may be due to the following reasons: (1) The template is poorly updated, and therefore, it is consistently failing to find good match in upcoming image frames; (2) the target moved so fast that it had gone outside the search window.

Case III (When $C_{\text{peak}} < t$ and $f > \lambda$): In order to deal with the issues mentioned in Case II, we smoothly update the template with the most trusty one, i.e., T_1, to solve the first issue and start increasing the search area of template to handle the second issue. Figure 1 (row four) shows that the proposed strategy updates the template (shown at top-right corners of frames) better than the any of the other updating methods. Yellow rectangle in Fig. 1 represents the current best match in the frame. It is clear that naive method and α-method start sliding off at frame 101, and β-method suffers from drifting problem at frame 211, while the proposed method does not face any such issue and keeps locking the target more accurately. We will discuss the efficacy of our algorithm further in Sect. 4.

3.1.2 Adaptive threshold

Fixed threshold scheme is presented in many papers such as [27,29,30,32]. This scheme sets a global fixed value for all frames of a video and does not take into account any local information obtained through correlation surface at each image frame. Thus, it always uses same barrier at every image frame without considering scene and target dynamics. Therefore, the method is more likely to fail in case of very fast maneuvering object changing its appearance *heavily and rapidly*. Peak correlation value indirectly provides clue about changes in target and acts as heuristic information to introduce adaptability in the threshold; e.g., if current peak value of normalized correlation is 0.85, it means this value may drop more in next image frame . So, threshold value should be set well below the current peak correlation value for upcoming frames. In this way, the scheme uses local information of target matching score to set its threshold at each frame instead of using global value for all the frames. To avoid the possibility of too low threshold value to be accepted as good matching, we put a lower limit on adaptive threshold. Mathematically, the process is described by Eq. (12).

$$t = \begin{cases} C_{\text{peak}} - \tau & \text{if } t \geq t_l \\ t_l & \text{otherwise} \end{cases} \tag{12}$$

where $0.10 \leq \tau \leq 0.17$ and $0 < t_l < 1$, i.e., it is being assumed that the target may change its appearance by at most 17 % in the next image frame. We have found this limit empirically, and it works well for slow and fast maneuvering object changing its appearance slowly or rapidly. Moreover, we use $\tau = 0.12$ and $t_l = 0.65$. Pseudocode of the adaptive threshold method is presented in Algorithm 3.

Algorithm 3: Adaptive Threshold

Input: current threshold, t, peak correlation value, C_{peak}
Output: updated threshold

```
Initialize $t_l$ and $\tau$
if $t \geq t_l$
   $t \leftarrow C_{peak} - \tau$
else
   $t \leftarrow t_l$
endif
```

3.2 Occlusion handling with Kalman filter

When the target is hidden by another object in the scene, it is said that the occlusion has occurred. It is a critical task for all visual object tracking algorithms to handle this situation. Peak correlation value may be used as an occlusion indicator because its value drops as target suddenly get occluded by another object. As its value becomes less than the threshold, we stop updating the template and assume that the target coordinates provided by the correlation tracker are no more trust worthy. Previously, Kalman filter predicted position is considered as current position of the target, and Kalman filter is updated according to its own prediction. The value of threshold is iteratively reduced. It is due to the fact that changes in target during occlusion are not incorporated in template, so the peak correlation value may drop below the threshold. Moreover, size of dynamically created search window is made larger and larger at each iteration to take into account the possible variations in the direction and speed of the target during occlusion. The template is correlated with the search window at each image frame but tracker remains in Kalman mode (i.e, the bounding box for the target is decided by Kalman predicted coordinates) till best-match score exceeds the threshold. Algorithm 4 summarizes these steps.

Algorithm 4: Occlusion Handling with Kalman Filter

```
1. Consider previously Kalman Filter predicted position as current
   position of the target.
2. Kalman filter is updated according to its own prediction in the previous
   iteration.
3. Template is not updated during occlusion.
4. Value of threshold is iteratively reduced.
5. Size of dynamically created search window is made larger and larger at
   each iteration.
```

3.3 Adaptive fast mean-shift algorithm

Mean shift is used for segmentation and tracking due to its clustering and mode seeking capability. It is an iterative algorithm centered at a random point, finds mean value in its neighborhood and shifts center point to the newly found mean position. The process ends up when change in position is extremely small, or maximum number of iterations is reached. Mathematical detail of mean shift is simple, and it is easy to apply on images; interested readers may study it in [48]. In order to find weighted mean of data points, a kernel function is used to assign weight to each data point. In case of uniform kernel, integral image technique can be exploited for fast calculation of mean shift [48]. Difference of two consecutive frames usually shows moving regions; the regions can be considered as potential candidates for target in tracking scenario. Mean-shift technique can be used to find these regions in difference image. Beleznai et al. exploited fast mean shift approach with uniform kernel for human detection and tracking [48–52]. We manipulate the same technique but we introduce novelty by making size of kernel adaptive at each frame. Moreover, we compute the *difference of search windows* instead of full frames. For this, the size of the both search windows is kept same, and their difference is obtained by subtracting the previous search window from the current one. In this way, too many moving regions and outliers in difference image can be avoided. Furthermore, the process becomes more efficient, computationally, because mean shift is now calculated in the search window only. We set the size of kernel at each image frame equal to the size of the template. The template size is made adaptive by the following two methods: (1) correlating the original template as well as 10 % smaller and 10 % larger templates with the search space. The size of the template, which provided the highest peak correlation value, is considered the new template size. Similar technique was used for scale handling in many other papers too; some of which are [27,62–64]. (2) Best-Match Rectangle Adjustment (BMRA) algorithm is used to resize the template according to the target size and keep the target at the center of the template. It divides the template into nine nonoverlapping fragments and checks the energy contents in each fragment. A voting scheme is used for adjustment of best-match rectangle [65]. Algorithm 5 summarizes these steps.

Algorithm 5: Adaptive Fast Mean Shift Algorithm

```
1. Calculate difference of search windows.
2. Calculate size of the template in the current image by BMRA as well
   as correlating 10 % larger and 10 % smaller template with the search
   window.
3. The size of the Kernel is set as the size of the template calculated in
   step 2.
4. Apply fast mean shift algorithm with the difference image calculated
   at step 1 and the rectangular Kernel calculated at step 3.
```

3.4 Combining correlation, Kalman filter, and adaptive kernel fast mean-shift algorithms

Kalman filter is measurement follower. It predicts the position of the target in next frame based on the position of the target (determined by the correlation tracker) in the current and the previous frames. It works in prediction-correction cycle. That is, it predicts the next position of the target and corrects itself exploiting the actual position of the target. It works in recursive manner and takes a few number of measurements to come to its steady state. After that, its accuracy is determined by the closeness of its predicted value with the measurement value at each image frame. When difference between predicted and measured values gets larger than a threshold, it indicates an alarming situation for tracking scenario. It may be due to any of following reasons: (1) correlation tracker provided wrong measurement due to clutter, blurriness, occlusion (especially slowly occurring long-term occlusion), out-of-plane rotation of target, or any other issue in search window, or (2) target has suddenly changed its direction (e.g., target may be moving back and forth briskly); correlation measurement is correct one in this case. The problem becomes worse when there is no significant decrease in peak correlation value, i.e., no indication of occlusion. To grip this problematic issue and to decide whether to follow Kalman filter prediction or correlation tracker measurement, we propose to combine the strengths of correlation, Kalman filter, and adaptive kernel fast mean-shift algorithm. For this, we calculate the difference between the measured and the predicted target position at each image frame. If the difference is greater than a threshold, the difference of the current and the previous search window is calculated. The adaptive fast mean-shift algorithm is applied on the difference search window to find the position of the potential candidate for the target. It is checked whether the measured or the predicted target position is the nearest neighbor to this value. If it is the measured one, we consider it the correct position of the target, otherwise, the predicted position is considered correct. Moreover, the template is not updated in this case, and the area of the search window is increased iteratively so that the possibility of the target going out of the search window may be avoided. Algorithm 6 presents these steps briefly, and Fig. 19 shows the proposed tracking algorithm diagrammatically.

Algorithm 6: Combining Correlation, Kalman Filter and Adaptive Fast Mean Shift Algorithms

1. Calculate difference between measured and predicted target position at each image frame.
2. If the difference is greater than a threshold, get the difference search window by subtracting the current search window from the previous one.
3. Apply the proposed adaptive fast mean shift algorithm in difference search window and find the position of potential candidate for the target, i.e. the candidate with the highest correlation value with the template.
4. Check whether the position calculated in step 3 is the nearest neighbor of measured value or the predicted value. If it is measured value, we consider it correct position, otherwise, confidence is given to the predicted value.
5. Template is not updated during this scenario.
6. Area of search window is iteratively increased to avoid the possibility of getting the target out of the search window.

4 Results and discussion

In this section, we present (1) the efficacy of the proposed template updating method, (2) effect of different values of τ for adaptive threshold on tracking results, (3) the comparison of correlation tracker, correlation and KF tracker, and correlation, KF, and adaptive fast mean shift based proposed tracker, and (4) the comparison of the proposed tracking strategy with nine state-of-the-art tracking methods on different publically available videos.

4.1 Dataset

We use eleven publically available challenging videos for different experimentation to show the robustness of our algorithm. The videos are *Girl*, *Faceocc*, *Faceocc2*, *ThreePastShop2Cor2* (from Caviar dataset), *Woman*, *Car11*, *David*, *Singer*, *Board*, *Box*, and *Liquor*. Several papers have used these videos for benchmarking their algorithms in recent years, some of which are [26,60,66–71]. Thus, the videos may be considered as de-facto standard videos for tracking algorithm evaluation. *Girl, Faceocc, Faceocc2*, and David videos can be downloaded from [72], *Board, Box*, and *Liquor* videos are available at [73], and *Woman, ThreePastShop2Cor2, Singer, Car11* videos can be downloaded from [74–77], respectively. Table 1 provides description of these videos.

4.2 Analysis for proposed tracking algorithm

We have analyzed the proposed algorithm qualitatively as well as quantitatively. For qualitative analysis, sample tracked frames of the proposed method are shown and

Table 1 Description of test videos

Sequence	No. of frames	Challenges involved
Faceocc2	812	Slowly occurring heavy occlusions, high appearance changes
ThreePastShop2Corr2	351	Similar objects, Heavy occlusion, appearance and scale changes
woman	552	Occlusions, appearance changes
Car11	393	Low light conditions
David	462	Illuminations changes, appearance changes
Singer	351	Illuminations changes, scale changes
Board	698	3D motion, cluttered background
Box	1161	Fast 3D motion, occlusions, motion blur, cluttered background, scale changes
Liquor	1741	Fast 3D motion, occlusions, motion blur
Faceocc	887	Slow occurring long-term occlusions, high appearance changes
Girl	502	360° out-of-plane rotation, appearance change, occlusion

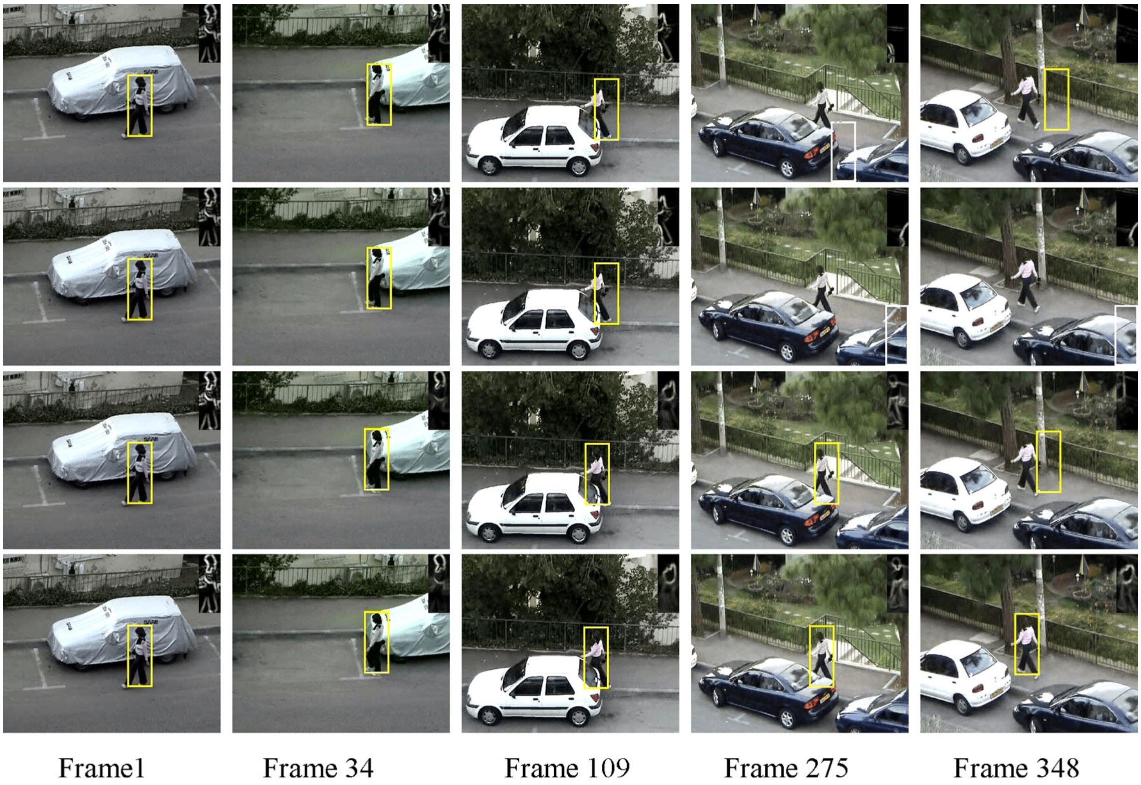


Fig. 2 Comparison of different updating schemes (i.e., Naive, α, and β methods in first, second, and third row, respectively) with the proposed method (fourth row). It is clear that the proposed method works better than any of the other methods

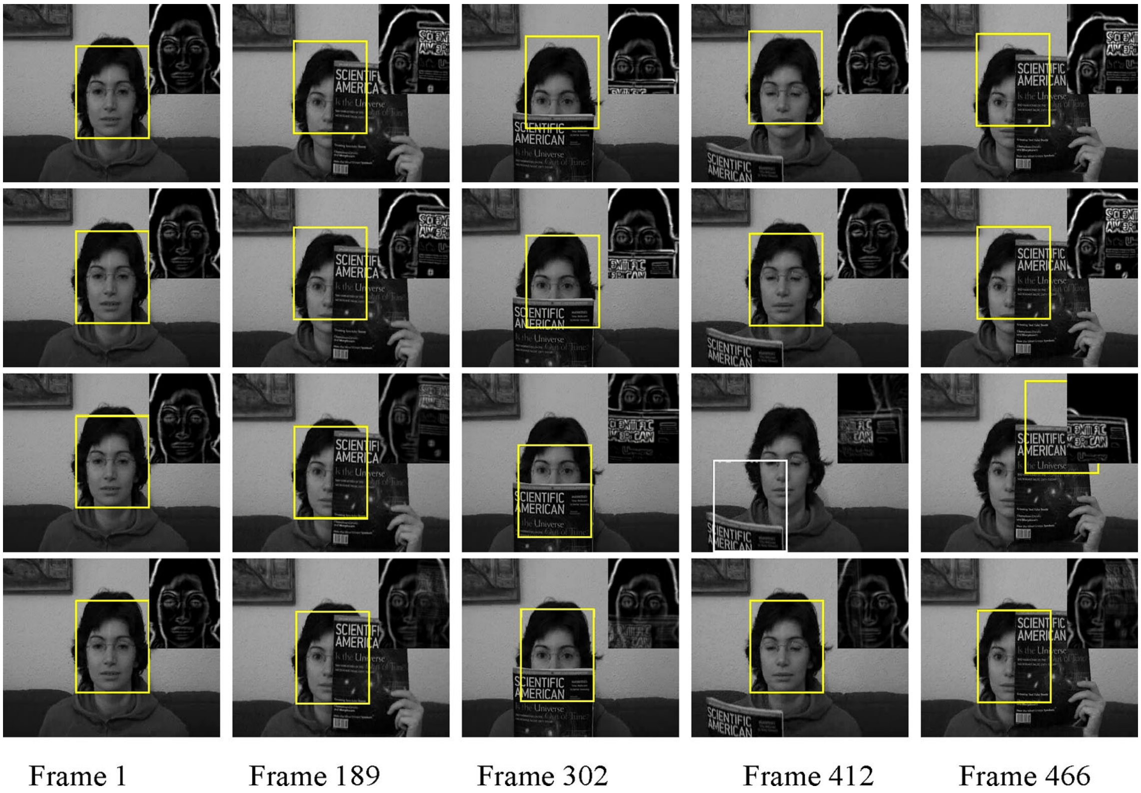


Fig. 3 Comparison of different updating schemes (i.e., Naive, α, and β methods in the first, second, and third row, respectively) with the proposed method (fourth row). The proposed method successfully handles slow occurring long-term occlusion

Table 2 Pascal score on test video sequences with different values of τ

Value of τ sequence	0.1	1.2	1.4	1.6	1.7
Faceocc2	1.00	1.00	1.00	1.00	1.00
Caviar	0.629	0.894	0.211	0.211	0.731
woman	0.091	1.00	1.00	1.00	1.00
Car11	1.00	1.00	1.00	1.00	1.00
David	1.00	1.00	1.00	1.00	1.00
Singer	0.277	1.00	1.00	1.00	1.00
Board	0.312	0.75	0.77	0.78	0.78
Box	0.927	0.923	0.884	0.901	0.901
Liquor	0.831	0.954	0.736	0.711	0.562
Faceocc	0.983	0.933	0.865	0.865	0.865
Girl	0.663	0.891	0.743	0.743	0.743

Table 3 Mean distance error on test video sequences with different values of τ

Value of τ sequence	0.1	1.2	1.4	1.6	1.7
Faceocc2	9.463	9.463	9.463	9.463	9.463
Caviar	43.131	4.963	66.539	66.315	24.805
woman	111.211	2.353	2.353	2.353	2.353
Car11	1.559	1.559	1.559	1.559	1.559
David	6.079	6.079	6.079	6.079	6.079
Singer	88.129	2.630	2.630	2.630	2.630
Board	75.571	34.960	33.524	33.125	33.125
Box	10.703	12.122	13.818	13.130	13.130
Liquor	35.566	20.469	62.426	63.640	73.473
Faceocc	6.357	11.066	17.321	17.321	17.321
Girl	40.236	21.428	25.017	25.017	25.017

compared visually with the results of benchmark algorithms. The visually better results are considered those which have tracked rectangle closer to the target of interest. Qualitative analysis does not provide fair comparison between different algorithms. Therefore, quantitative solution is calculated to have better understanding of robustness of the proposed algorithm. For this, two measures have been employed: one is the mean distance from center location, it provides the error between center location of tracked rectangle and its ground truth value, and the other is Pascal VOC criteria [78], which outputs the number of correctly tracked frames. Pascal score can be computed using Eq. (13):

$$s = \frac{area(R_t \cap R_g)}{area(R_t \cup R_g)} \tag{13}$$

where R_t is target tracked rectangle, and R_g is its ground truth. A frame is considered as correctly tracked if $s > 0.5$.

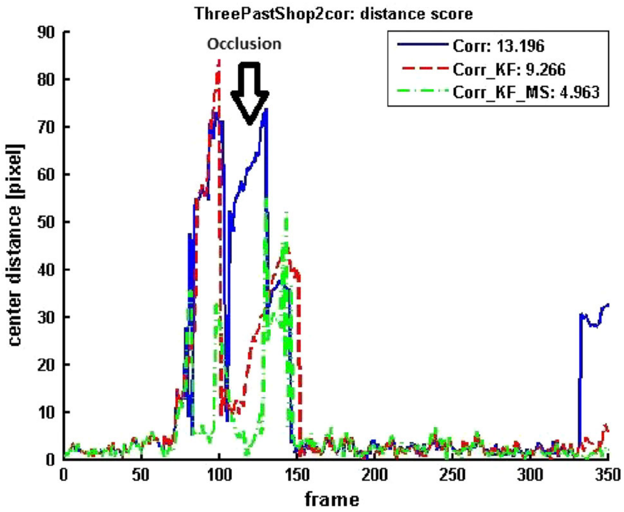


Fig. 4 Comparison of results for simple correlation tracker, correlation and KF tracker, and adaptive fast mean shift embedded with correlation and KF tracker. It proves the claim that adding mean-shift approach in the proposed way with correlation and KF tracker improves the results

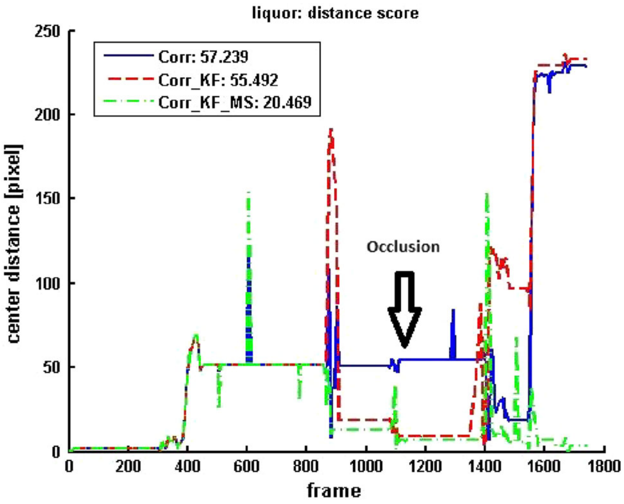


Fig. 5 Comparison of results for simple correlation tracker, correlation and KF tracker, and adaptive fast mean shift embedded with correlation and KF tracker. It proves the claim that adding mean-shift approach in the proposed way with correlation and KF tracker improves the results

4.3 Performance Of proposed template updating method

Figures 1, 2, and 3 highlight a few frames of *Girl*, *Woman*, and *Faceocc* sequences, respectively. *Girl* sequence mainly contains challenges of fast appearance change and out-of-plane rotation, *Woman* sequence provides challenges of high appearance change as well as heavy and long-term occlusion, and *Faceocc* sequence has slowly occurring heavy occlusion. The first three rows in each figure are glimpses of results of naive, α, and β methods, respectively, and the fourth row represents the results of the proposed method. Template has

Table 4 Comparison of correlation–KF tracker with and without adaptive fast mean-shift algorithm

Sequence	Pascal VOC score		Mean distance error	
	Without mean shift	With mean shift	Without mean shift	With mean shift
Faceocc2	0.939	1.00	14.014	9.463
Caviar	0.477	0.894	9.266	4.963
Woman	1.000	1.00	2.353	2.353
Car11	1.000	1.00	1.608	1.559
David	0.645	1.00	15.735	6.079
Singer	1.000	1.00	3.035	2.630
Board	0.064	0.75	206.746	34.960
Box	0.335	0.923	213.631	12.122
Liquor	0.756	0.954	55.492	20.469
Faceocc	0.908	0.933	16.961	11.066
Girl	0.713	0.891	23.282	21.427

been shown at top-right corner of each image frame. The yellow rectangle in the figures shows the position of highest match of template in the image, and white rectangle in some image frames explains that the peak correlation value has been dropped below the threshold and tracker is now in Kalman filter prediction mode. β-method in *Girl* and *Woman* sequences gives better results than naive and α methods but in case of *Face-occlusion* sequence it does not. The reason is that *Faceocc* sequence has fixed background as well as stationary target while the other two sequences contain non-stationary background as well as target. It is obvious from the figures that the proposed method performs better than each of the other methods in all the sequences.

4.4 Adaptive threshold with different parameter values

Value of τ plays a pivot role in adaptive threshold. We have performed various experiments with different values of τ in the range of 0.10–0.17 (both ends closed) and calculate Pascal score and mean distance error of all test video sequences. The results are summarized in Tables 2 and 3, respectively. It can be concluded from these results that $\tau = 0.12$ provides better results for most of the test video sequences.

4.5 Comparison of correlation tracker, correlation and KF tracker, correlation, KF, and adaptive fast mean shift based tracker

In this section, we compare the results of three tracking algorithms, i.e., (1) simple correlation tracker, (2) correlation- and KF-based tracker, and (3) the proposed correlation, KF, and adaptive fast mean-based tracking algorithm. This way, we may certify that our claim of heuristically switching (with the help of mean shift approach) between correlation based measured and KF-based predicted target coordinates makes the tracking robust and hence improves its results. The proposed adaptive threshold and template updating methods have been used in these three methods. Figures 4 and 5 show the center location error for *ThreePastShop2Corr2* and *Liquor* videos, respectively. It is clear from Fig. 4 that the occlusion occurring during frames 107–130 is not handled by simple correlation tracker and produces mean center error of 13.196, KF helps in this situation and reduces the average center distance to 9.266, and the performance of correlation–KF tracker improves significantly when embedded with adaptive fast mean-shift approach with the average score center distance to 4.963. Similar situation can be seen in Fig. 5 for *Liquor* video with mean center errors of 57.239, 55.492, and 20.469 for these three approaches, respectively. We have marked the occluded regions in both Figures by downward directed arrow with the labels of **occlusion**. In order to elaborate the advantage of integrating adaptive fast mean-shift approach in correlation–KF tracker, we have calculated the mean distance error as well as Pascal score on all the test

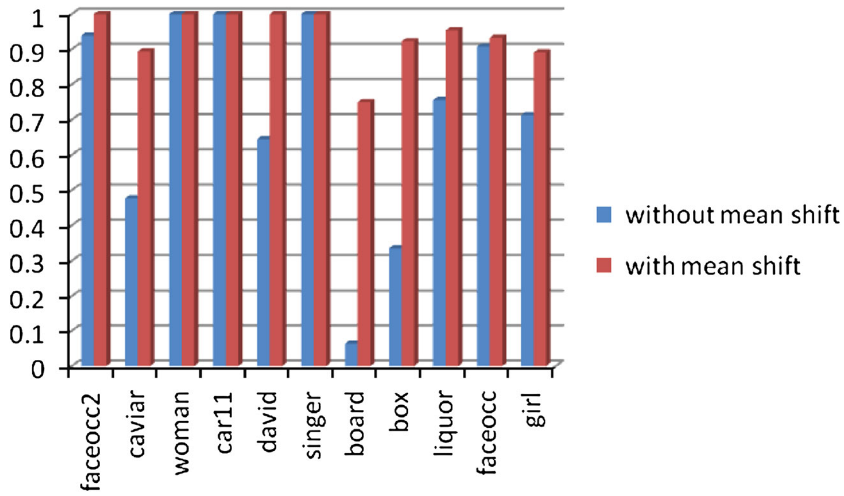


Fig. 6 Comparison of Pascal score of correlation–KF tracker with and without adaptive fast mean-shift algorithm

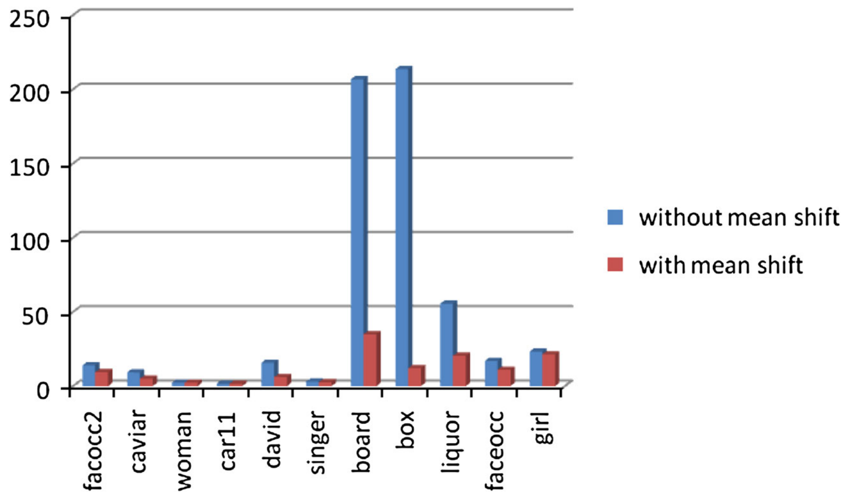


Fig. 7 Comparison of mean distance error of correlation–KF tracker with and without adaptive fast mean-shift algorithm

Table 5 Mean center location error

	IVT	L1	PN	VTD	MIL	FragTrack	LSAM	EENC	PROST	The
	(2008)	(2009)	(2010)	(2010)	(2011)	(2006)	(2012)	(2008)	(2010)	proposed
Faceocc2	*10.2*	11.1	18.6	10.4	14.3	15.5	**3.8**	41.309	17.2	*9.463*
Caviar	66.2	65.9	*53.0*	60.9	83.9	94.2	**2.3**	91.867	–	*4.963*
Woman	167.5	131.6	*9.0*	136.6	122.4	113.6	*2.8*	104.549	–	**2.353**
Car11	*2.1*	33.3	25.1	27.1	43.5	63.9	*2.0*	2.332	–	**1.559**
David	**3.6**	*7.6*	9.7	13.6	15.6	46.0	**3.6**	17.418	15.3	*6.079*
Singer	8.5	*4.6*	32.7	4.1	15.2	22.0	4.8	15.589	–	**2.630**
Board	165.5	177.0	97.0	96.1	51.2	90.1	**7.3**	165.347	*37.0*	*34.960*
Box	–	196.0	–	–	104.6	*57.4*	–	117.866	*12.696*	**12.122**
Liquor	–	–	–	–	115.1	*30.7*	–	100.733	*21.487*	**20.469**
Faceocc	–	–	–	–	18.4	**6.5**	–	48.641	*7.0*	*11.066*
Girl	48.5	62.5	23.2	21.5	31.5	*26.5*	–	53.711	**19.0**	*21.427*
Average	59.013	76.622	33.537	46.287	55.973	51.491	**3.8**	69.033	*18.526*	*11.554*

Table 6 Pascal VOC score

	IVT	L1	PN	VTD	MIL	FragTrack	LSAM	EENC	PROST	The
	(2008)	(2009)	(2010)	(2010)	(2011)	(2006)	(2012)	(2008)	(2010)	proposed
Faceocc2	0.59	0.84	0.49	0.59	*0.96*	0.60	*0.82*	0.515	*0.82*	**1.00**
Caviar	*0.21*	0.20	*0.21*	0.19	0.19	0.19	*0.84*	0.309	–	**0.894**
Woman	0.19	0.18	0.60	0.15	0.16	*0.20*	*0.78*	0.182	–	**1.00**
Car11	*0.81*	0.44	0.38	0.43	0.17	0.09	*0.81*	*0.886*	–	**1.00**
David	0.72	0.63	0.60	0.53	0.70	0.47	*0.79*	0.742	*0.80*	**1.00**
Singer	0.66	0.70	0.41	*0.79*	0.33	0.34	*0.74*	0.246	–	**1.00**
Board	0.17	0.15	0.31	0.36	*0.679*	*0.679*	*0.74*	0.136	**0.75**	**0.75**
Box	–	0.05	–	–	0.245	*0.614*	–	0.506	*0.914*	**0.923**
Liquor	–	–	–	–	0.206	*0.799*	–	0.504	*0.854*	**0.954**
Faceocc	–	–	–	–	*0.93*	**1.00**	–	0.449	**1.00**	*0.933*
Girl	0.42	0.32	0.57	0.51	*0.70*	*0.70*	–	0.287	*0.89*	**0.891**
Average	0.471	0.390	0.446	0.444	0.479	0.516	*0.789*	0.433	*0.861*	**0.940**

videos with and without adaptive fast mean-shift algorithm as shown in Table 4. It is clear from the table that integration of mean-shift approach into correlation– KF tracker significantly improves the results. Figures 6 and 7 summarize these results for Pascal score and mean distance error, respectively.

4.6 Performance comparison of proposed tracking method with other methods

Tracking results of the proposed method are compared with the nine state-of-the-art tracking methods, i.e., incremental visual tracking (IVT) [79], λ1 tracker [80], PN learning [81], visual tracking decomposition (VTD) [70], MIL tracker [66], FragTrack [60], local sparse appearance model (LSAM) [69], PROST [67], and EENC tracker [27,30]. We have mentioned the already cited results of these trackers from the papers [67,69] (except EENC tracker, it was run on all the videos). Therefore, if the result on a certain video is not found in the papers, we do not mention it.

Table 5 summarizes the results of mean center location error in pixels, and Table 6 shows the mean Pascal score. First row of both tables show the name of the algorithm with its publishing year. The best result for each video is shown in bold–underline, the second best is in italic–underline, and the third best result is in italic format. The last row of each table shows the average score of the algorithms for all 9 videos. It is clear from the tables that the proposed algorithm, overall, performs better than each of the other algorithms. The algo-

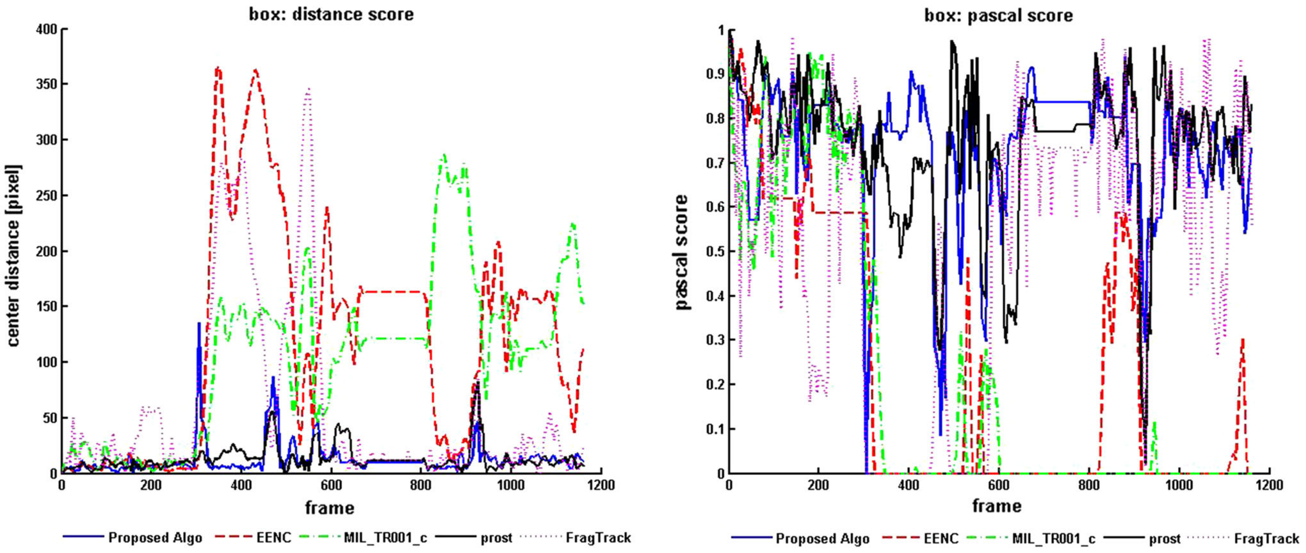


Fig. 8 Distance and Pascal score for box video sequence

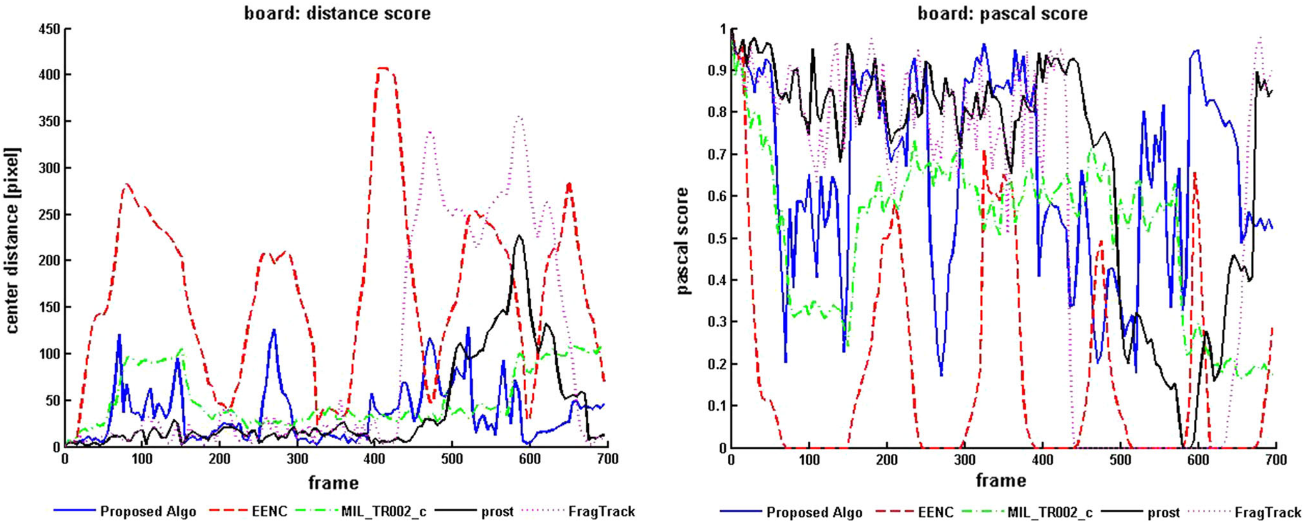


Fig. 9 Distance and Pascal score for board video sequence

rithm was implemented using OpenCV on Core i5 machine with 4 GB RAM. The number of frames processed per second (fps) depends upon the size of template and search window. The normalized correlation is calculated in Fourier domain or spatial domain depending on the sizes of the template and the search window for fast processing. Adaptive fast mean shift is also efficient as compared to the original mean-shift algorithm due to usage of integral histogram technique. Furthermore, it is calculated only when there is no overlap between predicted and measured target coordinates, or the peak cor-

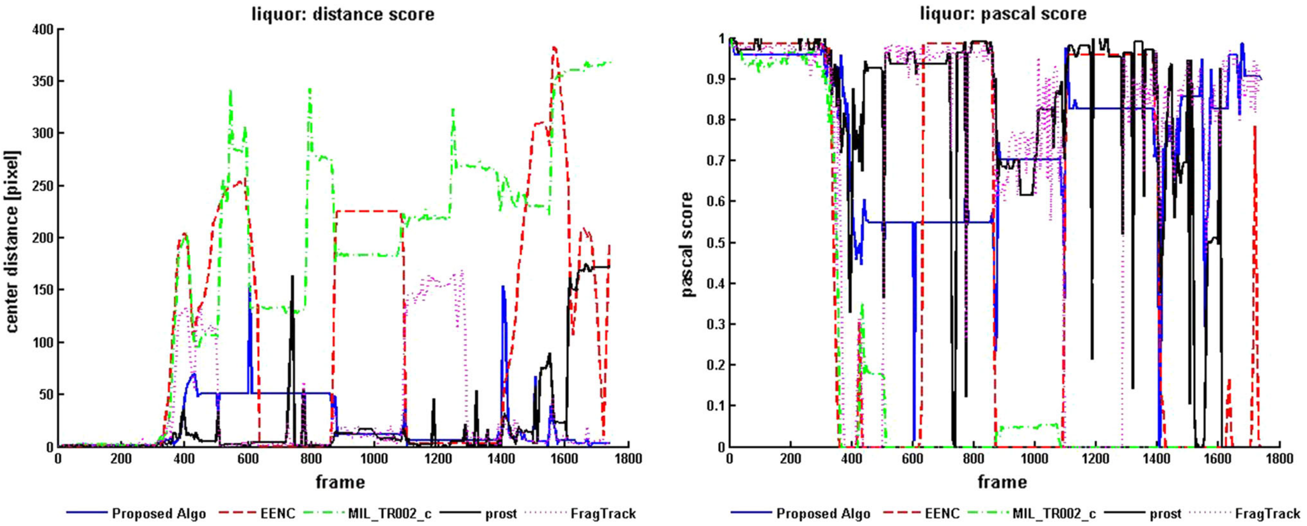
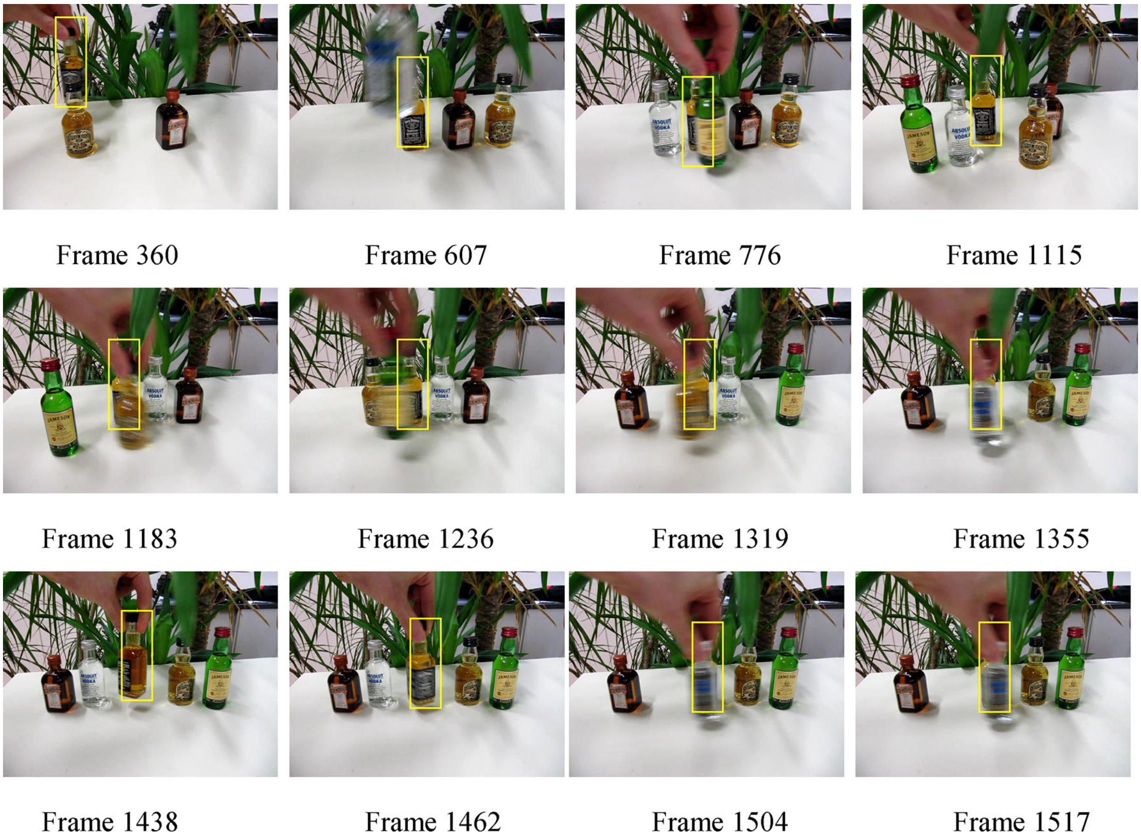


Fig. 10 Distance and Pascal score for liquor video sequence


Frame 360

Frame 607

Frame 776

Frame 1115

Frame 1183

Frame 1236

Frame 1319

Frame 1355

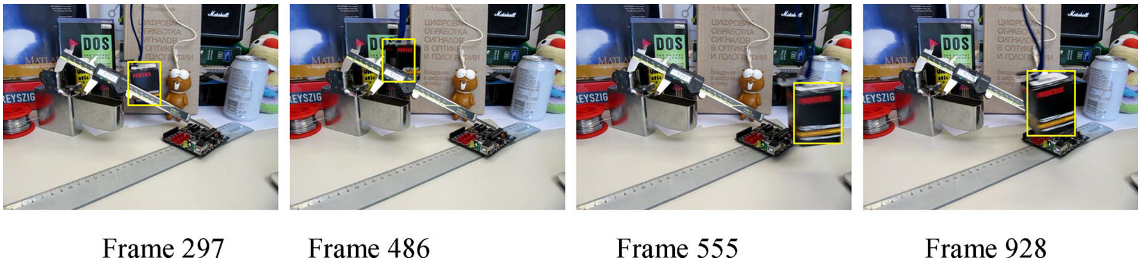
Frame 1438

Frame 1462

Frame 1504

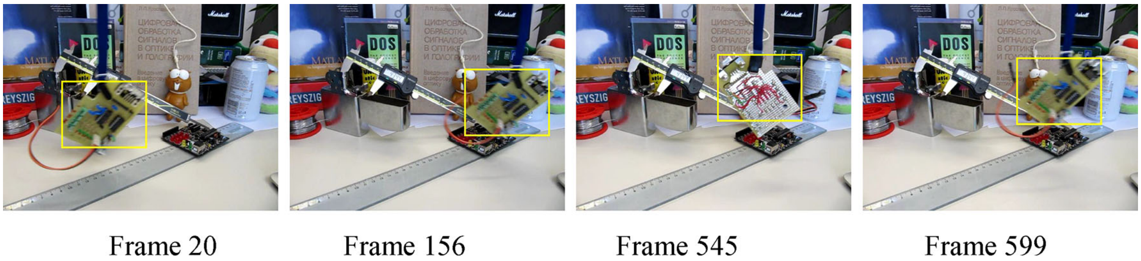
Frame 1517

Fig. 11 A few tracked frames of Liquor video sequence. The proposed approach successfully tracks during occlusions, 3D motion causing blurriness, and background clutter



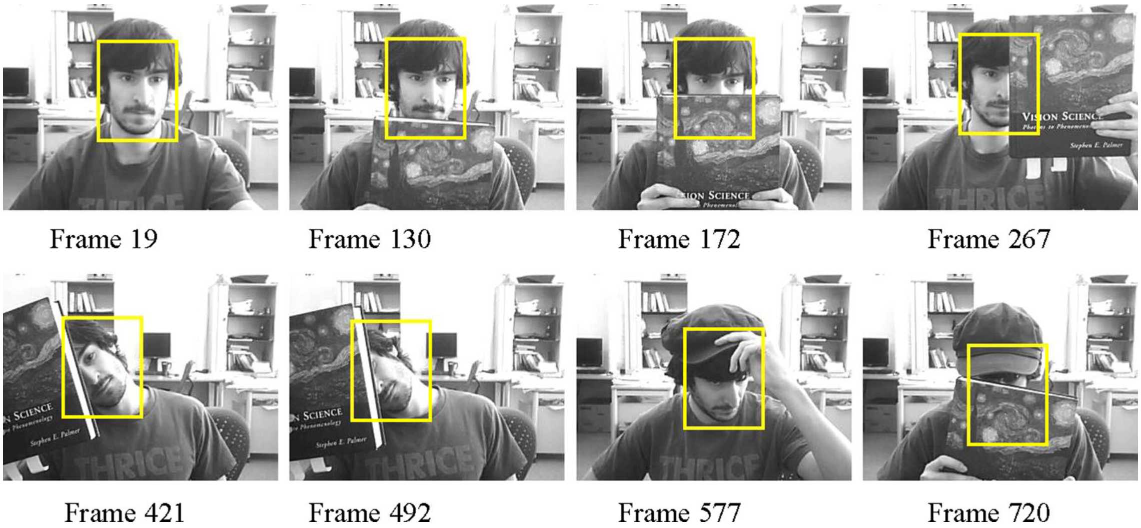
Frame 297 Frame 486 Frame 555 Frame 928

Fig. 12 Sample tracked frames of Box video sequence. The proposed algorithm successfully tracks the target during occlusions, scale changes, 3D motion causing blurriness, and clutter background



Frame 20 Frame 156 Frame 545 Frame 599

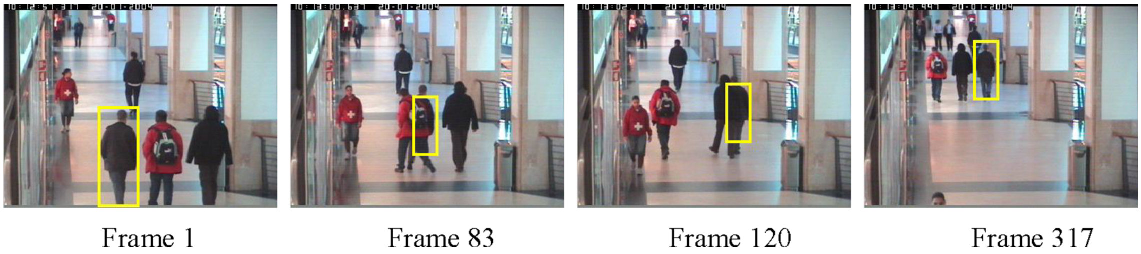
Fig. 13 Results for Board video sequence. The proposed algorithm successfully handles the out-of-plane motion of the target in cluttered background



Frame 19 Frame 130 Frame 172 Frame 267

Frame 421 Frame 492 Frame 577 Frame 720

Fig. 14 A few frames of Faceocc2 video sequence. The proposed algorithm tracks the target with large appearance changes and slowly occurring heavy occlusions



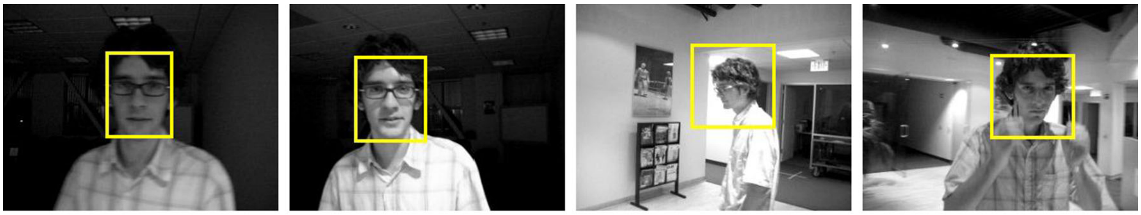
Frame 1 Frame 83 Frame 120 Frame 317

Fig. 15 Some tracked frames from the sequence ThreePastShop2Cor2 (Caviar dataset). The main challenges in the sequence are (1) the similar objects and (2) the occlusions, which occur while the persons in the sequence cross each other. The proposed method successfully tracks the target

relation value is less than the threshold. On the average, the whole algorithm runs in real time (i.e., 25 fps).

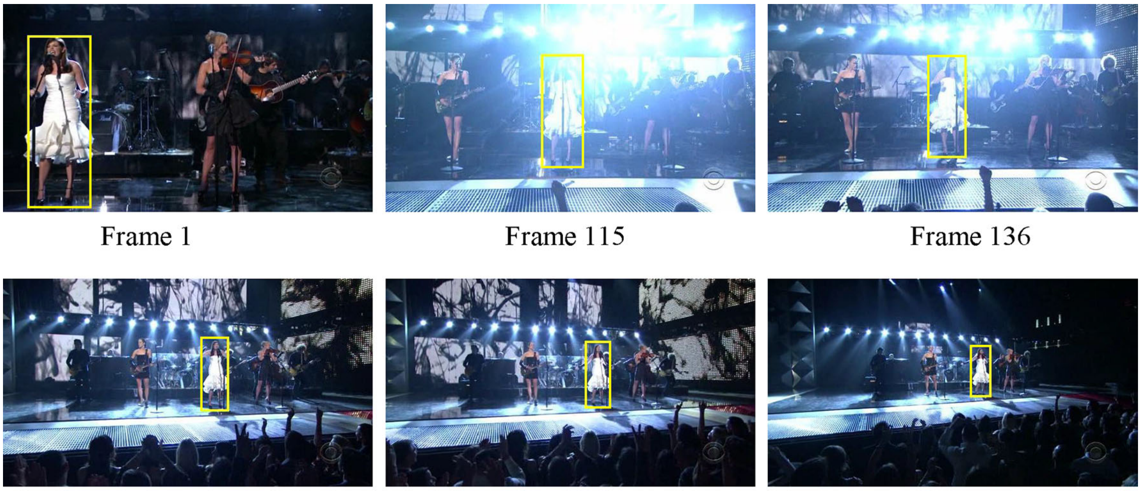
Figures 8, 9, and 10 depict graphically the center distance error and Pascal score for the *Board*, *Box* and *Liquor* videos, respectively, for each fifth frame (as ground truth is available for only these frames). The graphs show the results of the proposed, EENC, MIL, PROST, and FragTrack in blue, red, green, black, and magenta colors, respectively.

Figure 11 shows a few sample tracked frames for *Liqour* video sequence. The proposed algorithm successfully tracks the target during occlusions (as shown in Frames 360, 607, 776, 1115, 1183, 1236, 1319, 1355, 1438, and 1462) and 360° rotation causing motion blur (e.g., frames 1404 and 1407). Figure 12 shows the performance of the proposed tracking method for *Box* video during occlusions (e.g., Frames 297 and 486), scale changes, complex target motion including its 3D rotation creating motion blur in cluttered background (for example, Frame 555 and 928). Figure 13 explains that the proposed algorithm successfully handles the out-of-plane rotation of the target with cluttered background in *Board* video. Figure 14 depicts the results of the proposed algorithm on *Faceocc2* video. The video contains large appearance changes (for example, appearance in Frame 19 and Frame 577) and slowly occurring heavy occlusions (e.g., more than 90 % of the face is occluded as shown in Frame 720). The proposed template updating and tracking strategy keeps lock-



Frame 1 Frame 25 Frame 160 Frame 383

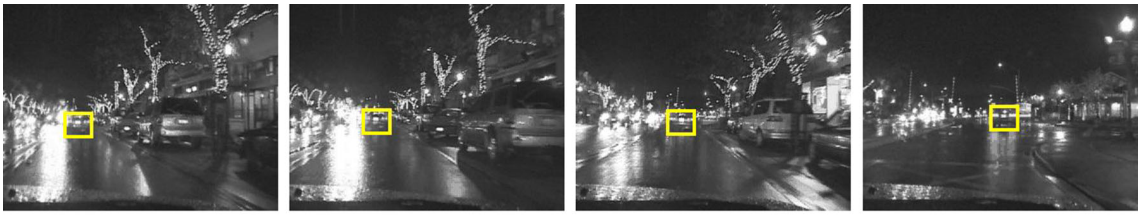
Fig. 16 Some frames from David video sequence. The proposed algorithm tracks the target in varying illuminations and appearance changes



Frame 1 Frame 115 Frame 136

Frame 240 Frame 265 Frame 333

Fig. 17 A few frames from Singer video sequence. The proposed algorithm successfully handles high illumination effects as well as large scale changes



Frame 1 Frame 25 Frame 200 Frame 305

Fig. 18 Frames of Car video sequence. The proposed algorithm successfully tracks the target in low light conditions

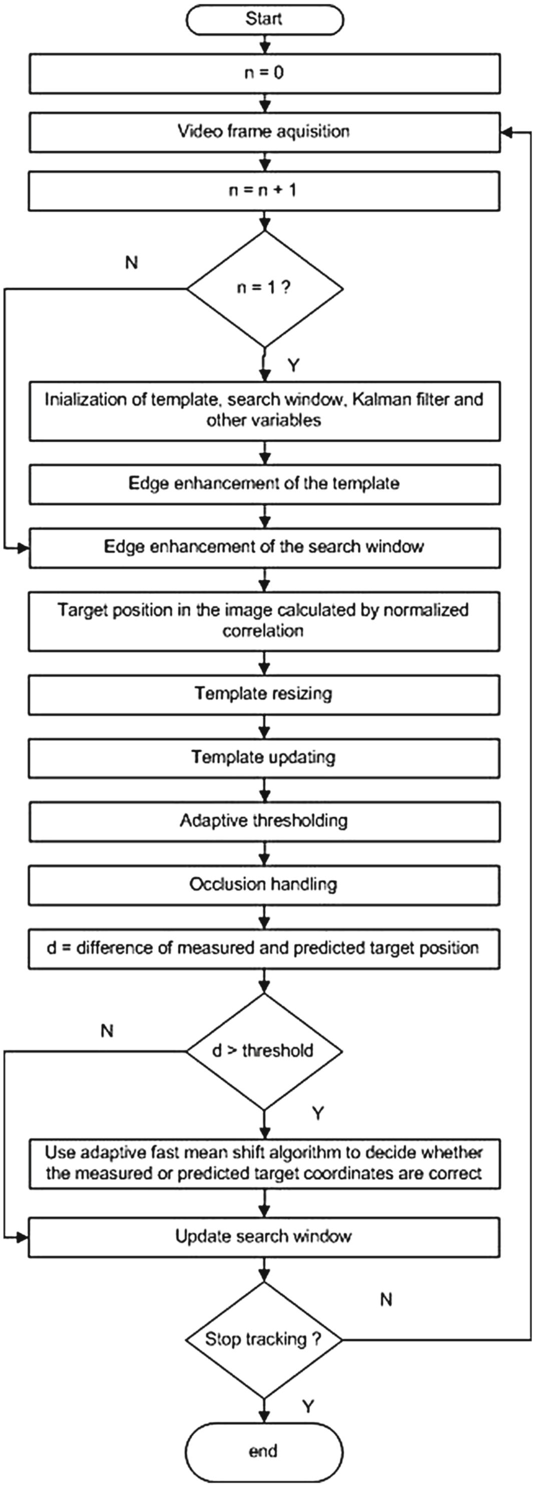

Start
n = 0
Video frame aquisition
n = n + 1
n = 1 ?
N
Y
Inialization of template, search window, Kalman filter and other variables
Edge enhancement of the template
Edge enhancement of the search window
Target position in the image calculated by normalized correlation
Template resizing
Template updating
Adaptive thresholding
Occlusion handling
d = difference of measured and predicted target position
d > threshold
N
Y
Use adaptive fast mean shift algorithm to decide whether the measured or predicted target coordinates are correct
Update search window
Stop tracking ?
N
Y
end

Fig. 19 Proposed Tracking Algorithm

ing the target successfully. Figure 15 shows some frames of *ThreePastShop2Cor2* video from Caviar dataset. The video contains similar objects, which makes it difficult to track the target. The situation becomes worse due to the occlusions which occur when the target and the other objects in the video cross each other (e.g., Frames 83 and 120). The proposed method shows prominent results and successfully tracks the target. Figure 16 shows a few frames of *David* video. The proposed algorithm handles varying illumination conditions (e.g., Frame 1 and 25), complex target motion (e.g., Frame 160), and target appearance changes (e.g., Frame 383). Figure 17 shows some frames of *Singer* video. The proposed algorithm successfully handles the high illumination effects on the target (e.g., Frame 115) and large change in its scale (e.g., Frame 333). Figure 18 (*Car11* video frames) shows that the proposed algorithm tracks the target in low light conditions (Fig. 19).

5 Conclusion and future work

The contribution of the research presented in the paper is as follows: (1) combining of correlation, Kalman filter, and fast mean shift algorithms for robust object tracking, (2) the novel template updating method, which updates the template according the rate of change in target appearance, (3) adaptive threshold for peak correlation value; the threshold varies for each upcoming image frame according to the current frame peak correlation value, (4) adaptive kernel size for fast mean shift; the size of kernel varies according to changing size of target. Comparison of the proposed method with nine other methods on eleven different challenging videos shows the efficacy of the algorithm. We conclude from our results that the proposed tracking algorithms handles high illumination effects (*Singer* video), low light conditions (*Car11* video), varying illumination with target appearance changes (*David* video), slowly occurring heavy occlusions (*Faceocc* video) with appearance changes (*Faceocc2* and *Woman* videos), out-of-plane rotation of the target (*Girl* and *Board* videos), motion blur, complex object motion, 360° rotation of the target, clutter in background, and occlusions (*Liquor* and *Box* videos). The algorithm assumes that the target does not change its appearance more than fifty percent during the course of tracking. This constraint holds true for all the test videos discussed in the paper, but there might be cases where the assumption does not remain valid (e.g., airborne objects such as kite, aero plane moving away or toward the camera and taking turn). Parameters tuning may also be required for some other videos. Future extension of the algorithm includes the development of constraint free tracking technique. The assumption of appearance change in the target less that 50 % may be handled by considering current template after certain numbers of frames as the most trusty template if the peak correlation value is greater than a certain value, e.g., 0.95, but it requires adding two more parameters in the algorithm.

Acknowledgments This research work is supported by PIEAS-administered HEC Endowment Fund for Higher education and R&D for IT and Telecom Sector Fund.

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Surv. (CSUR) **38**(4), 1–45 (2006)
2. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Trans. Syst. Man Cybern. **34**, 334–352 (2004)
3. Kettnaker, V., Zabih, R.: Bayesian multi-camera surveillance. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 23–25 June 1999, pp. 1–18 (1999)
4. Collins, R.T., Lipton, A.J., Fujiyoshi, H., Kanade, T.: Algorithms for cooperative multisensor surveillance. Proc. IEEE **89**(10), 1456–1477 (2001)
5. Greiffenhagen, M., Comaniciu, D., Niemann, H., Ramesh, V.: Design, analysis, and engineering of video monitoring systems: an approach and a case study. Proc. IEEE **89**(10), 1498–1517 (2001)
6. Kumar, R., Sawhney, H., Samarasekera, S., Hsu, S., Tao, H., Guo, Y., Hanna, K., Pope, A., Wildes, R., Hirvonen, D., Hansen, M., Burt, P.: Aerial video surveillance and exploitation. Proc. IEEE **89**(10), 1518–1539 (2001)
7. Decarlo, D., Metaxas, D.: Optical flow constraints on deformable models with applications to face tracking. Int. J. Comput. Vis. **38**(2), 99–127 (2000)
8. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **24**(1), 34–58 (2002)
9. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 747–757 (2000)
10. Fablet, R., Black, M.J.: Automatic detection and tracking of human motion with a view-based representation. In: European Conference on Computer Vision (ECCV'02) 2002, pp. 476–491 (2002)
11. Agarwal, A., Triggs, B.: Learning to track 3D human motion from silhouettes. In: International Conference on Machine Learning (ICML'04), Banff, Canada 2004, pp. 9–16 (2004)
12. Rand, D., Kizony, R., Weiss, P.T.L.: The Sony PlayStation II EyeToy: low-cost virtual reality for use in rehabilitation. J. Neurol. Phys. Ther. **32**(4), 155–163 (2008)
13. Handmann, U., Kalinke, T., Tzomakas, C., Werner, M., von Seelen, W.: Computer vision for driver assistance systems. In: International Society for Optics and Photonics: Aerospace/Defense Sensing and Controls 1998, pp. 136–147 (1998)
14. Avidan, S.: Support vector tracking. IEEE Trans. Pattern Anal. Mach. Intell. **26**(8), 1064–1072 (2004)
15. Coifman, B., Beymer, D., McLauchlan, P., Malik, J.: A real-time computer vision system for vehicle tracking and traffic surveillance. Transp. Res. Part C: Emerg. Technol. **6**(4), 271–288 (1998)
16. Bradski, G.R.: Real time face and object tracking as a component of a perceptual user interface. In: Fourth IEEE Workshop on Applications of Computer Vision (WACV'98). 1998, pp. 214–219 (1998)
17. Papanikolopoulos, N.P., Khosla, P.K.: Adaptive robotic visual tracking: theory and experiments. IEEE Trans. Autom. Control **38**(3), 429–445 (1993)
18. Amini, A., Owen, R., Anandan, P., Duncan, J.: Non-rigid motion models for tracking the left-ventricular wall. In: Information Processing in Medical Imaging 1991, pp. 343–357 (1991)
19. Vasconcelos, M.J.M., Ventura, S.M.R., Freitas, D.R.S., Tavares, J.M.R.S.: Using statistical deformable models to reconstruct vocal tract shape from magnetic resonance images. Proc. Inst. Mech. Eng. Part H: J. Eng. Med. **224**(10), 1153–1163 (2010)
20. Vasconcelos, M.J., Rua Ventura, S.M., Freitas, D.R.S., Tavares, J.M.R.S.: Towards the automatic study of the vocal tract from magnetic resonance images. J. Voice **25**(6), 732–742 (2010)
21. Cafforio, C., Rocca, F.: Tracking moving objects in television images. Signal Process. **1**(2), 133–140 (1979)
22. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: 7th International Joint Conference on Artificial Intelligence 1981 (1981)
23. Fitts, J.M.: Precision correlation tracking via optimal weighting functions. In: 18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes 1979, pp. 280–283 (1979)
24. Asgarizadeh, M., Pourghassem, H.: A robust object tracking synthetic structure using regional mutual information and edge correlation-based tracking algorithm in aerial surveillance application. Signal Image Video Process. 1–15 (2013)
25. Wang, Y., Zhao, Q.: Robust object tracking via online principal component-canonical correlation analysis (P3CA). Signal Image Video Process. 1–16 (2013)
26. Khan, M.I., Ahmed, J., Ali, A., Masood, A.: Robust edge-enhanced fragment based normalized correlation tracking in cluttered and occluded imagery. Signal Process. Image Process. Pattern Recogn. **12**, 169–176 (2009)
27. Ahmed, J., Ali, A., Khan, A.: Stabilized active camera tracking system. J. Real-Time Image Process. 1–20 (2012)
28. Ahmed, J.: Adaptive Edge-Enhanced Correlation Based Robust And Real-Time Visual Tracking Framework and Its Deployment in Machine Vision Systems. Research, National University of Science and Technology (NUST), Karachi (2008)
29. Ali, A., Kauser, H., Khan, M.I.: Automatic Visual Tracking and Firing System for Anti-Aircraft Machine Gun. In: 6th International Bhurban Conference of Applied Science and Technology, Islamabad, Pakistan, 2009, pp. 253–257 (2009)
30. Ahmed, J., Jafri, M.N., Shah, M., Akbar, M.: Real-time edge-enhanced dynamic correlation and predictive open-loop car following control for robust tracking. Mach. Vis. Appl. J. **19**(1), 1–25 (2008)
31. Wong, S.: Advanced correlation tracking of objects in cluttered imagery. In: Defense and Security:International Society for Optics and Photonics 2005, pp. 158–169 (2005)
32. Ali, A., Mirza, S.M.: Object tracking using correlation, Kalman filter and fast means shift algorithms. In: International Conference on Emerging Technologies, 2006. ICET'06, Islamabad, pp. 174–178 (2006)
33. Wilson, J.N., Ritter, G.X.: Handbook of Computer Vision-Algorithms in Image Algebra. CRC Press, Boca Raton (2001)
34. Kuglin, C., Hines, D.: The phase correlation image alignment method. In: International Conference on Cybernetics and Society 1975, pp. 163–165 (1975)
35. Chen, Q., Defrise, M., Deconinck, F.: Symmetric phase-only matched filtering of Fourier–Mellin transforms for image registration and recognition. IEEE Trans. Pattern Anal. Mach. Intell. **16**(12), 1156–1168 (1994)
36. Manduchi, R., Mian, G.A.: Accuracy analysis for correlation-based image registration algorithms. In: IEEE International Symposium on Circuits and Systems (ISCAS'93) 1993, pp. 834–837 (1993)
37. Stone, H.S., Tao, B., McGuire, M.: Analysis of image registration noise due to rotationally dependent aliasing. J. Vis. Commun. Image Represent. **14**(2), 114–135 (2003)
38. Stone, H.S.: Fourier-based image registration techniques. NEC Research (2002)
39. Ahmed, J., Jafri, M.N.: Improved phase correlation matching. In: ICISP-08: International Conference on Image and Signal Processing, France 2008, pp. 128–135 (2008)
40. Jingying, J., Xiaodong, H., Kexin, X., Qilian, Y.: Phase correlation-based matching method with sub-pixel accuracy for translated

and rotated images. In: IEEE International Conference on Signal Processing (ICSP'02) 2002, pp. 752–755 (2002)
41. Foroosh, H., Zerubia, J.B., Berthod, M.: Extension of phase correlation to subpixel registration. IEEE Trans. Image Process. **11**(3), 188–200 (2002)
42. Keller, Y., Averbuch, A., Miller, O.: Robust Phase Correlation. In: 17th International Conference on Pattern Recognition (ICPR'04) 2004, pp. 740–743 (2004)
43. Blackman, S., Popoli, R.: Design and Analysis of Modern Tracking Systems. Artech House, Boston (1999)
44. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice-Hall, Englewood Cliffs (2002)
45. Lewis, J.P.: Fast Normalized Cross-Correlation. In: Vision Interface 1995, pp. 120–123 (1995)
46. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing Using MATLAB. Pearson Education Pte. Ltd., Delhi (2004)
47. Nixon, M., Aguado, A.: Feature Extraction and Image Processing. Newnes, Oxford (2002)
48. Beleznai, C., Frühstück, B., Bischop, H.: Human detection in groups using a fast mean shift procedure. In: International Conference on Image Processing (ICIP), October 2004, pp. 349–352 (2004)
49. Beleznai, C., Frühstück, B., Bischop, H.: Detecting humans in groups using a fast mean shift procedure. In: Proceedings of the 28th Workshop of the Austrian Association for Pattern Recognition (AAPR), June 2004, pp. 71–78 (2004)
50. Beleznai, C., Frühstück, B., Bischop, H.: Tracking multiple humans using fast mean shift mode seeking. In: IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, January 2005, pp. 25–32 (2005)
51. Beleznai, C., Frühstück, B., Bischop, H.: Human tracking by mode seeking. In: Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA), September 2005, pp. 1–6 (2005)
52. Beleznai, C., Frühstück, B., Bischop, H.: Human tracking by fast mean shift mode seeking. Trans. J. Multimed. **1**(1), 1–8 (2006)
53. Wang, X., Liu, L., Tang, Z.: Infrared human tracking with improved mean shift algorithm based on multi-cue fusion. Trans. J. Appl. Otics **48**(21), 4201–4212 (2009)
54. Sutor, S., Röhr, R., Pujolle, G., Reda, R.: Efficient mean shift clustering using exponential integral kernels. Trans. Int. J. Electric. Comput. Eng. **4**(4), 206–210 (2009)
55. Shan, C., Tan, T., Wei, Y.: Real-time hand tracking using a mean shift embedded particle filter. Trans. Pattern Recogn. **40**, 1958–1970 (2007)
56. Yilmaz, A., Shafique, K., Lobo, N., Li, X., Olson, T., Shah, M.: Target tracking in FLIR imagery using mean shift and global motion compensation. In: IEEE Workshop on Computer Vision Beyond Visible Spectrum, Kauai, Hawaii 2001, pp. 54–58 (2001)
57. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: IEEE Conference on Computer Vision and Pattern Recognition, June 2000, pp. 142–149. Hilton Head, SC (2000)
58. Comaniciu, D., Ramesh, V.: Mean shift and optimal prediction for efficient object tracking. In: IEEE International Conference on Image Processing (ICIP) 2000, pp. 70–73 (2000)
59. Li, X., Zhang, T., Shen, X., Sun, J.: Object Tracking using an Adaptive Kalman Filter combined with Mean Shift. Opt. Eng. **49**(2), 31–33 (2010)
60. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral Histogram. In: IEEE Conference on Computer Vision and Pattern Recognition (ICPR) 2006, pp. 798–805 (2006)
61. Brunson, R.L., Boesen, D.L., Crockett, G.A., Riker, J.F.: Precision trackpoint control via correlation track referenced to simulated imagery. In: International Society for Optics and Photonics: Aerospace Sensing 1992, pp. 325–336 (1992)
62. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. **25**(5), 564–577 (2003)
63. Collins, R.T.: Mean-shift blob tracking through scale space. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2003, pp. 234–240 (2003)
64. Ahmed, J., Shah, M., Miller, A., Harper, D., Jafri, M.N.: A Vision-based System for a UGV to Handle a Road Intersection. In: Proceedings of the National Conference on Artificial Intelligence 2007. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999
65. Ahmed, J., Jafri, M.N.: Best-match rectangle adjustment algorithm for persistent and precise correlation tracking. In: IEEE International Conference on Machine Vision (ICMV), Islamabad, Pakistan, 28–29 December 2007 (2007)
66. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1619–1632 (2011)
67. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: PROST: Parallel robust online simple tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2010, pp. 723–730 (2010)
68. Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012, pp. 1940–1947 (2012)
69. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012, pp. 1822–1829 (2012)
70. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2010, pp. 1269–1276 (2010)
71. Liu, B., Huang, J., Yang, L., Kulikowsk, C.: Robust tracking using local sparse appearance model and k-selection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011, pp. 1313–1320 (2011)
72. http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml
73. http://gpu4vision.icg.tugraz.at/index.php?content=subsites/prost/prost.php
74. http://www.cs.technion.ac.il/~amita/fragtrack/fragtrack.html
75. http://groups.inf.ed.ac.uk/vision/caviar/caviardata1/
76. http://cv.snu.ac.kr/research/~vtd/
77. http://www.cs.toronto.edu/~dross/ivt/
78. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
79. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**(1), 125–141 (2008)
80. Mei, X., Ling, H.: Robust visual tracking using ℓ 1 minimization. In: IEEE 12th International Conference on Computer Vision 2009, pp. 1436–1443 (2009)
81. Kalal, Z., Matas, J., Mikolajczyk, K.: Pn learning: Bootstrapping binary classifiers by structural constraints. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2010, pp. 49–56 (2010)