Visual Tracking Extensions for Accurate Target Recovery in Low Frame Rate Videos

Yoav Liberman and Adi Perry

Elbit Systems Ltd., Haifa, Israel

Abstract. Visual tracking in low frame rate videos has many inherent difficulties for achieving accurate target recovery, such as occlusions, abrupt motions and rapid pose changes. Thus, conventional tracking methods cannot be applied reliably. In this paper, we offer a new scheme for tracking objects in low frame rate videos. We present a method of integrating multiple metrics for template matching, as an extension for the particle filter. By inspecting a large data set of videos for tracking, we show that our method not only outperforms other related benchmarks in the field, but it also achieves better results both visually and quantitatively, once compared to actual ground truth data.

1 Introduction

Object tracking is a core component of many applications such as surveillance and reconnaissance. Low frame rate (LFR) systems are common mainly due to hardware and processing bandwidth limitations. For these systems, tracking poses a challenge due to abrupt motion, rapid pose changes and partial or even absolute occlusions. These problems are encountered frequently and are difficult to cope with, as demonstrated in Fig. 1.

Although the literature regarding visual tracking is of large extent, most existing approaches cannot be reliably applied to LFR videos (e.g. [1], [2]), because of the uncertainties caused by its data (e.g., [3]). One of the main reasons for the limitation of traditional tracking approaches in effectively handling these situations lies on the fact that they heavily depend on motion continuity, which not necessarily hold in LFR videos. For example, particle filters [4] use particles for guiding the target propagation within a limited sub-space. Other methods such as the ones by Tomasi and Kanade [5] and Comaniciu *et al.* [6] are based on an iterative optimization scheme, which generally require the knowledge of feature patches in consecutive frames to overlap in a very close vicinity of each other. However, these presumptions are too expensive for LFR tracking. Therefore, we propose a new approach which can not only cope with the LFR complexities, but also presents consistency, and performs well over long term videos.

During the past half decade some methods were proposed for the purpose of LFR tracking (e.g., [7], [8], [9]). Beside these, may LFR tracking be their motive or not, Liu *et al.* [10] and Okuma *et al.* [11] offered to use a boosted detector to amend the particle filter distributions for handling the pitfalls of

G. Bebis et al. (Eds.): ISVC 2014, Part I, LNCS 8887, pp. 128–139, 2014.

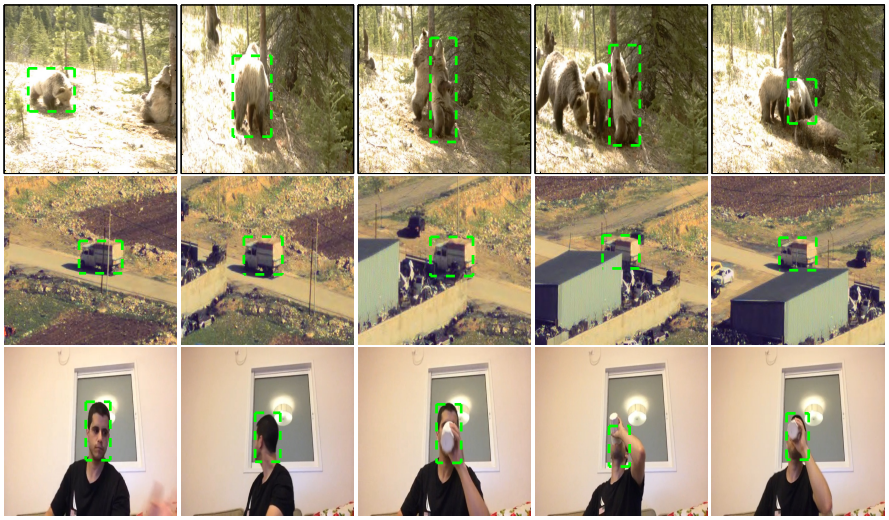

Fig. 1. Examples of LFR tracking for 3 frames per second videos with target objects marked by a dashed line rectangle

visual tracking, which can be quite effective even for LFR videos. Still, these methods might be limited because they mainly depend upon detection, thus they do not necessarily provide a sufficient remedy for the complexities that are often encountered (e.g., two similar proximate objects, object and background resemblance, partial occlusions etc.). Hence, a different, more robust approach that can reliably deal with these kind of complexities is of great importance.

The main notion of our method is that matching and tracking can be integrated in a complementary way in order to overcome the difficulties in LFR tracking. Regrading the matching, we consider multiple metrics, both statistic and deterministic, in order to recover the target, while combining both multiple notions of the tracked template, and taking into account its distinct color properties. Then, by assimilating all the templates matchings from all inspected metrics into multiple particle filters, the target object can be accurately tracked.

This paper is organized as follows: Section 2 details the proposed object tracking algorithm for target recovery in LFR videos. In Section 3 we demonstrate some visual results of our proposed technique, compared to other renowned methods. The performance of our suggested approach and other benchmarks in the field with respect to actual ground truth, given a large data set of object tracking cases, is provided in Section 4. This paper is concluded in Section 5.

2 LFR Visual Tracking Extensions Algorithm

In this section we first reveal the deficiencies of the particle filter for LFR tracking. Then we manifest the matching phase of our method and how it is integrated into the detection phase for the purpose of target recovery in LFR videos.

2.1 The Particle Filter Deficiencies in LFR Tracking

The basic idea of the particle filter for visual tracking lies in its ability to provide an estimate of the state given the observations at each time step. This can be achieved by estimating the distribution $p(x_t|\mathbf{z}_t)$ of the target state given all observations up to time t, denoted as $\mathbf{z}_t = (z_1, z_2, \ldots, z_t)$, and x_t denotes the object's state at time t. The particle filter for visual tracking (also known as Condensation [4]) estimates this distribution in a two-step recursion scheme:

$$P(x_t|\boldsymbol{z}_{t-1}) = \int P(x_t|x_{t-1})P(x_{t-1}|\boldsymbol{z}_{t-1})dx_{t-1} \tag{1a}$$

$$P(x_t|\mathbf{z}_t) = \frac{P(z_t|x_t)P(x_t|\mathbf{z}_{t-1})P(\mathbf{z}_{t-1})}{P(\mathbf{z}_t)} \propto P(z_t|x_t)P(x_t|\mathbf{z}_{t-1}), \tag{1b}$$

where (1a) and (1b) indicate the prediction and the update steps, respectively.

The object tracking algorithm using particle filter is detailed in Algorithm 1, where in the algorithm S, π and C denote the particles, the probability and the cumulutive distribution, respectively. This approach is further discussed in [12].

Algorithm 1. Particle Filter for Object Tracking.

1. **Input**: $S_{t-1}^{(n)}$, $\boldsymbol{z}_t$. **Output**: $S_t^{(n)}, \pi_t^{(n)}, C_t^{(n)}$, with particle number n and time t.
2. Predict the particles state by applying: $S_{t'}^{(n)} = f(S_{t-1}^{(n)}) + W_t^{(n)}$; where $W_t^{(n)}$ is distributed normally and f is a known model function.
3. Calculate the particles likelihood: $\pi_{t'}^{(n)} = P(S_{t'}^{(n)}|\boldsymbol{z}_t)$ and $C_{t'}^{(n)} = \sum_{m=1}^{n} \pi_{t'}^{(m)}$.
4. Resample the particles $S_{t'}^{(n)}$ and obtain $S_t^{(n)}$:
    1. From a random uniform distribution draw r, such that: $r \in [0, 1]$.
    2. Find smallest j s.t.: $C_{t'}^{(j)} >= r$.
    3. Set $S_t^{(n)} = S_{t'}^{(j)}$.
5. Given the new particles, update $\pi_t^{(n)} = P(S_t^{(n)}|\boldsymbol{z}_t)$.
6. Normalize and re-evaluate $\pi_t^{(n)}$ and $C_t^{(n)}$: $\pi_t^{(n)} = \frac{\pi_t^{(n)}}{\sum_n \pi_t^{(n)}}$, $C_t^{(n)} = \sum_{m=1}^{n} \pi_t^{(m)}$.

For LFR tracking, the particle filter as in Algorithm 1 is not sufficient for achieving accurate target recovery, since tracking loss may occur due to the gradual departure of the sample set from the actual target, as illustrated in Fig. 2. A possible remedy for this pitfall is to increase the number of samples [13]; however this kind of solution is not stable under LFR conditions. Therefore, we offer a new integration scheme of two important tools: matching and tracking based on particle filters, for handling the complex cases of LFR tracking. These tools phases are detailed in Section 2.2 and Section 2.3.

2.2 The Matching Phase

The first step of the matching phase is to set initial conditions. That is, in the $1st$ frame the desired template object to track (defined as T_0) is provided, along with

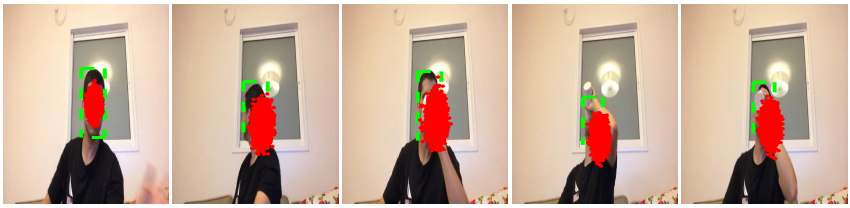

Fig. 2. Example of the particle filter for visual tracking in LFR video

the template's center of mass position in the frame, denoted as - $[cm_{x;0}, cm_{y;0}]$. Then, in the $2nd$ frame, we acquire information regarding the tracked object trajectories, this could be achieved by either having a prior information or by doing an exhaustive search around the object's previous location. From now on, for all the given objects, we assume that in the first two frames the desired tracked object is clearly visible and no abrupt changes had occurred, that is, the object's center of mass location (in both frames) is assumed to be exact.

Since a single-metric based template matching occasionally fails, we propose to incorporate multiple metrics altogether in order to form a stronger matching. These metrics consist of statistical measures (i.e., histogram comparisons which can capture the similarity between two consecutive appearances of the same object in case of partial occlusions and pose changes) and deterministic measures (i.e., correlation), which are all defined as follows, with respect to a given ROI:

$$NRMSE(x,y) = \frac{\sum_{x',y'}(T(x',y') - I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \sum_{x',y'} I(x+x',y+y')^2}} \tag{2a}$$

$$\rho(x,y) = \frac{\sum_{x',y'}((T(x',y') - \bar{T})(I(x+x',y+y') - \bar{I}(x,y))}{\sqrt{\sum_{x',y'}(T(x',y') - \bar{T})^2 \sum_{x',y'}(I(x+x',y+y') - \bar{I}(x,y))^2}} \tag{2b}$$

$$MAD(x,y) = \frac{\max_{x',y'} |T(x',y') - I(x+x',y+y')|}{\sqrt{\max_{x',y'} |T(x',y')| \max_{x',y'} |I(x+x',y+y')|}} \tag{2c}$$

$$\chi^2 = \sum_m \frac{(H_T(m) - H_I(m))^2}{H_T(m)} \tag{2d}$$

$$\rho_h = \frac{\sum_m (H_T(m) - \bar{H}_T)(H_I(m) - \bar{H}_I)}{\sqrt{\sum_m (H_T(m) - \bar{H}_T)^2 \sum_I (H_I(m) - \bar{H}_I)^2}} \tag{2e}$$

$$EMD = \frac{\sum_{i,j} d_{ij} f_{ij}}{\sum_{i,j} f_{i,j}} \tag{2f}$$

where t is the current time frame, $[x+x', y+y'] \in ROI$, T denotes the tracked template and I is the frame in which the template is tracked. The $NRMSE$, ρ and MAD are the normalized RMSE, correlation coefficient and the maximum absolute deviation metrics, respectively. In (2b) $\bar{T}$, $\bar{I}$ are the template and ROI

means, respectively. χ^2 and ρ_h are the chi-square and correlation histogram metrics, where H_T and H_I stand for the template and ROI histograms, respectively. The EMD in (2f) is defined as the earth movers distance metric [14], where d_{ij} indicates the movement distance and f_{ij} stands for the amount of movement between bin i and bin j in the inspected histograms. Thus, for each one of the metrics we derive a scores matrix, for all of the $[x, y]$ locations within the ROI. Regarding (2d), (2e) and (2f) we adopt a multi-color observation model based on the Hue-Saturation-Value (HSV), since it provides more delicate histogram information than other color spaces histograms, [15]. By assimilating all these metrics together, we increase the probability to achieve an accurate matching, rather than what a single metric could accomplish. It should be noted that there is no limitation on the metrics that are used, that is, other metrics might be considered as well in order to increase the matching accuracy.

To further cope with partial occlusions and pose changes, we offer to use multiple sub-templates of different sizes acquired from the original template (of size $w \times h$) with respect to the estimated center of mass, as shown in Fig. 3. In the figure, we demonstrate the derivation of the sub-templates (e.g., of sizes: $w/2 \times h/2$, $w/4 \times h/4$ and $w/8 \times h/8$). Afterwards, a ROI around the center of mass, from the previous frame, is constructed for each sub-template, in which each metric is then applied. By doing so we reduce the algorithm computational effort as well as omitting irrelevant areas from matching.

Additionally, because the tracked template may have distinct color properties (e.g., a red vehicle), we offer to utilize its color characteristics, which may increase the matching accuracy. Hence, a weighting strategy is proposed between the individual color channels, achieved by first smoothing each one of the template's RGB channels, and then taking the mean value of them. Finally, the values are normalized in order to obtain proper weights. All these steps are combined together for achieving a reliable matching, as illustrated in Fig. 3.

To detect a possible matching failure (e.g., target is occluded), we define two thresholds (θ_H, θ_L), which are adjusted in context of the data. The matching scores are compared with these thresholds for determining the level of confidence which the matching achieved. The thresholds are defined as follows:

$$\theta_H = \frac{\alpha}{N_m} \sum_{Met} best_{Met;val} \;;\; \theta_L = \beta \cdot \theta_H, \tag{3}$$

where Met indicates each metric with total of N_m metrics and $0 < \alpha, \beta < 1$, $\alpha > \beta$. The best metric values (denoted as $best_{Met;val}$), were initialized with respect to the first two frames, since a reliable prior information was available.

Also, we define a "fallback template" to be used in case of a matching failure (i.e., the thresholds in (3) were not satisfied) by: $T_{FB} \triangleq T_0$, where T_0 is the initial template and is updated in each frame according to the achieved matching scores. The algorithm for the matching phase between two sequential frames (i.e., between frames $t-1$ and t) is detailed in Algorithm 2.

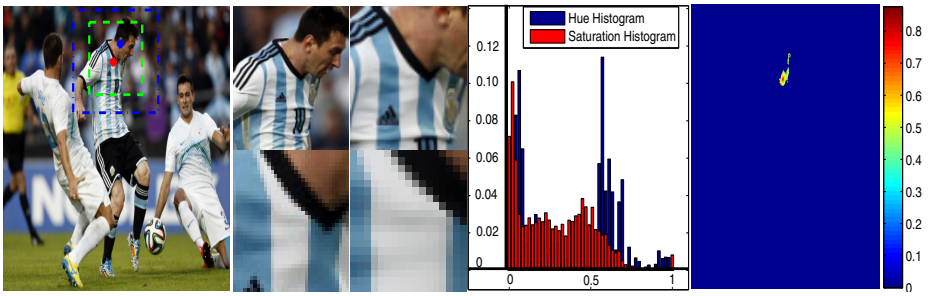

Fig. 3. An illustration of the matching phase. **1st column** (from left to right): the original frame along with the marked template to track indicated by a dashed rectangle and the inspected ROI given by a dashed-dotted rectangle. The two dots indicate the desired center of mass and the estimated center of mass. **2nd column**: The obtained sub-templates (from the original template - top left) of different sizes. **3rd column** - the Hue and Saturation histograms of the original template. **4th column**: the matching result for the original template with respect to the ROI, given the $\rho(x, y)$ metric, (2b).

Algorithm 2. The Matching Algorithm.

1. **Input**: The tracked object's center of mass $[cm_{x;t}, cm_{y;t}]$ at t and the original template T_{t-1} of size $w_{t-1} \times h_{t-1}$. **Output**: The scores matrices for each metric $\boldsymbol{R}_{Met}$, each metric's best score, location, and std: $best_{Met;val}$, $\boldsymbol{best}_{Met;Loc}$, σ_{Met}.
2. From T_{t-1} obtain a total of N_t sub-templates.
3. Define the ROI with respect to each sub template's width and height centered around $[cm_{x;t}, cm_{y;t}]$.
4. Apply a smoothing kernel on each sub-template.
5. Obtain each sub-template's normalized weights $\{\boldsymbol{w}_i\}_{i=1}^{N_t}$.
6. Apply each metric in (2) on each sub-template's color channel with respect to its ROI. The vector of the matrices scores is denoted as $\{\boldsymbol{\Phi}_{Met;i}\}_{i=1}^{N_t}$.
7. calculate each metric's matrix result: $\boldsymbol{R}_{Met} = (1/N_t) \sum_{i,m} w_{m;i} \cdot \boldsymbol{\Phi}_{m;i;Met}$.
8. Calculate each metric's standard deviation: $\sigma_{Met} = std(\boldsymbol{R}_{Met})$.
9. Obtain best value ($best_{Met;val}$) and location ($\boldsymbol{best}_{Met;Loc}$) from each $\boldsymbol{R}_{Met}$.
10. **if** $(1/N_m) \sum_{Met} best_{Met;val} > \theta_L$ **then**
11. **if** $(1/N_m) \sum_{Met} best_{Met;val} > \theta_H$ **then**
12. Update the fallback template: $T_{FB} = T_t$ and θ_H, θ_L as in (3).
13. **end if**
14. **return success**.
15. **else**
16. **if** $T_{t-1} \neq T_{FB}$ **then**
17. Set $T_{t-1} = T_{FB}$ and go to step (2).
18. **else**
19. **return failure**.
20. **end if**
21. **end if**

2.3 The Tracking Phase

For the tracking phase we adopt and extend the particle filter algorithm (as detailed in Algorithm 1) so it can be reliably applied for LFR tracking. From now on the particles will be regarded as $\boldsymbol{S}_t$ of size $4 \times N_p$, with N_p particles for time step t, where each row in $\boldsymbol{S}_t$ stands for each particle's: $x_{pos;t}$, $y_{pos;t}$, $x_{move;t}$ and $y_{move;t}$, respectively. The particles are initialized given the exact state of the template in $frame_{0,1}$: $\boldsymbol{S}_1 = [cm_{x;1} \cdot \mathbf{1};\ cm_{y;1} \cdot \mathbf{1};\ x_{move;1} \cdot \mathbf{1};\ y_{move;1} \cdot \mathbf{1}]$, with the unity vector $\mathbf{1}$ of size $1 \times N_p$. The particles distribution at each time step t is updated given the template's width and height at $t-1$ (i.e., $w_{t-1} \times h_{t-1}$) , according to a normal distribution, in the following manner:

$$\boldsymbol{S}_t = \boldsymbol{D}\boldsymbol{S}_{t-1} + \left[\sqrt{w_{t-1}/2} \cdot \boldsymbol{n}_x; \sqrt{h_{t-1}/2} \cdot \boldsymbol{n}_y; s_x \cdot \boldsymbol{n}_{x_{move}}; s_y \cdot \boldsymbol{n}_{y_{move}}\right], \quad (4)$$

where each element in $\boldsymbol{n}$ is a random Gaussian variable, i.e. $n_i \sim \mathcal{N}(0,1)$, and s_x, s_y are predefined constants. The matrix $\boldsymbol{D}$ is the transition matrix which is used for the prediction of the particles position in the current frame.

Given the observations metric results ($\boldsymbol{R}_{Met}$) along with the best location ($\boldsymbol{best}_{Met;loc}$) and best value ($best_{Met;val}$) from Algorithm 2, we define the likelihood of the predicted particles at frame t, that is:

$$\pi_t^{(n)} = P(S_t^{(n)}|R_{Met}^{(n)}) = Ce^{-\left(\frac{R_{Met}(x_n,y_n) - best_{Met;val}}{\sigma_{Met}}\right)^2}, \quad (5)$$

where x_n and y_n indicate the x, y location of the nth updated particle. By applying this distribution on each inspected metric (total of N_m) we have $N_m \cdot N_p$ probabilities with respect to all particles. The next step is to acquire the most dominant N_p particles (highest probabilities). Then, the probabilities are normalized so the particles can be resampled.

Algorithm 3. The Tracking Algorithm

1. **Input**: Particles positions and states at frame $t-1$ ($\boldsymbol{S}_{t-1}$), each inspected metric results at frame t ($\boldsymbol{R}_{Met}$). **Output**: Estimated center of mass, width and height.
2. Predict the particles position using (4) to obtain $\boldsymbol{S}_{t''}$.
3. Calculate each nth predicted particle's likelihood given the observations ($\boldsymbol{R}_{Met}$), i.e.: $\pi_{t''}^{(n)} = P(S_{t''}^{(n)}|R_{Met}^{(n)})$ by using σ_{Met} relevant to each metric.
4. **if** Algorithm 2 returned **Success then**
5. Consider the N_p most dominant particles, denoted as $\boldsymbol{S}_{t'}$, with probabilities $\boldsymbol{\pi_{t'}}$.
6. Normalize $\boldsymbol{\pi}_{t'}$ and obtain the commutative distribution: $\boldsymbol{C}_{t'} = \sum_{m=1}^{n} \pi_{t'}^{(m)}$.
7. Resample the particles $\boldsymbol{S}_{t'}$ as in step (4) of Algorithm 1 to yield $\boldsymbol{S}_t$.
8. Calculate the resampled particles likelihood $\boldsymbol{\pi}_t$ given the observations.
9. Evaluate the center of mass ($\boldsymbol{cm}_t$) given $\boldsymbol{S}_t$ and $\boldsymbol{\pi}_t$ by: $cm_{x,y} = \sum_n \pi_t^{(n)} S_{x,y;t}^{(n)}$.
10. Attain the new template's width and height by applying a detection scheme.
11. **else**
12. Set $\boldsymbol{S}_t = \boldsymbol{S}_{t''}$ and evaluate $\boldsymbol{cm}_t$ from step (9) given $\boldsymbol{\pi}_{t''}$ from step (3).
13. **end if**

Finally, in order to attain the most plausible position and state of the desired tracked object, the new particles positions and their probabilities are utilized. Also, if satisfactory scores were achieved, a detection scheme may be applied around the estimated center of mass with respect to the template's current height and width (e.g., Active Contours based algorithm as discussed in [16]). This procedure allows to derive an updated template for the frames to come. The algorithm for the tracking phase is detailed in Algorithm 3.

From step (12) in Algorithm 3, one can see that in cases when the matching did not provide satisfactory results , the algorithm estimates the object's position only by the particles prediction as in step (2). This course is very beneficial in: full occlusions, unpredictable change of scenery, extreme pose changes and more.

3 Results

We have tested our tracking algorithm on LFR videos with 3 to 5 FPS rate, and compared our results to the implementations of Matlab's KLT tracker from the CV Toolbox and a Condensation based visual tracking algorithm, as well as comparison to ground truth data. In Fig. 4 we present selected snapshots for tracking various kind of targets. The first four examples demonstrate the results of tracking under pose changes, abrupt motion and partial occlusions. The last example demonstrates our method's results of a vehicle that undergoes a long full occlusion (17 frames) along with the typical LFR tracking complexities.

In all frames shown in Fig. 4 the improvement of the proposed method with respect to the examined benchmarks is distinct while comparing to the ground truth. In order to further investigate our method's performance, we examined the recovery of each target in large scale of frames, as elaborated in Section 4.

4 Performance Evaluation

In order to quantify the performance of our algorithm, we used two data sets of LFR videos: the first is our own diverse set, manually annotated, including vehicles and humans in various scenarios and different resolutions; the latter is the PETS2009 data set. The videos were dilated to a 3 to 5 frames per second rate, varying from 50 - 200 frames, with total of 500 different objects.

To compare the results of our method and the other algorithms with the ground truth, we suggest three measures that capture the similarity of the recovered objects to the annotated templates, which are defined as follows:

$$CM_{RMSE}(t) = ((\hat{cm}_{x;t} - cm_{x;t})^2 + (\hat{cm}_{y;t} - cm_{y;t})^2)^{\frac{1}{2}} \tag{6a}$$

$$\rho(t) = \frac{\sum_{x,y}(\hat{T}(x,y;t) - \mu_{\hat{T};t})(T(x,y;t) - \mu_{T;t})}{(\sum_{x,y}(T(x,y;t) - \mu_{T;t})^2 \sum_{x,y}(\hat{T}(x,y;t) - \mu_{\hat{T};t})^2)^{\frac{1}{2}}} \tag{6b}$$

$$\chi^2(t) = \sum_I \frac{(H_{T;t}(I) - H_{\hat{T};t}(I))^2}{H_{T;t}(I)}, \tag{6c}$$

where the CM_{RMSE} is the center of mass (position) RMSE, ρ is the correlation, and χ^2 is the chi-square measure. $\mu_{\hat{T};t}$ and $\mu_{T;t}$ indicate the mean of the estimate $\hat{T}_t$ and the ground truth T_t, respectively. $H_{T;t}$ and $H_{\hat{T};t}$ stand for the HSV histograms of the ground truth T_t and the estimation $\hat{T}_t$, respectively. The index t refers to the frame number and $x,\ y$ indicate pixel location.

Frame by frame performance evaluations for the first four examples that were presented in Fig. 4 are illustrated in Fig. 5, where in the figure each method performance was assessed using (6). The mean scores of all 500 objects in total, for all frames, from both data sets are summarized in Table 1.

Given the results in Table 1 and Fig. 5, we have further manifested the consistency, accuracy and effectiveness of the proposed method for LFR tracking.

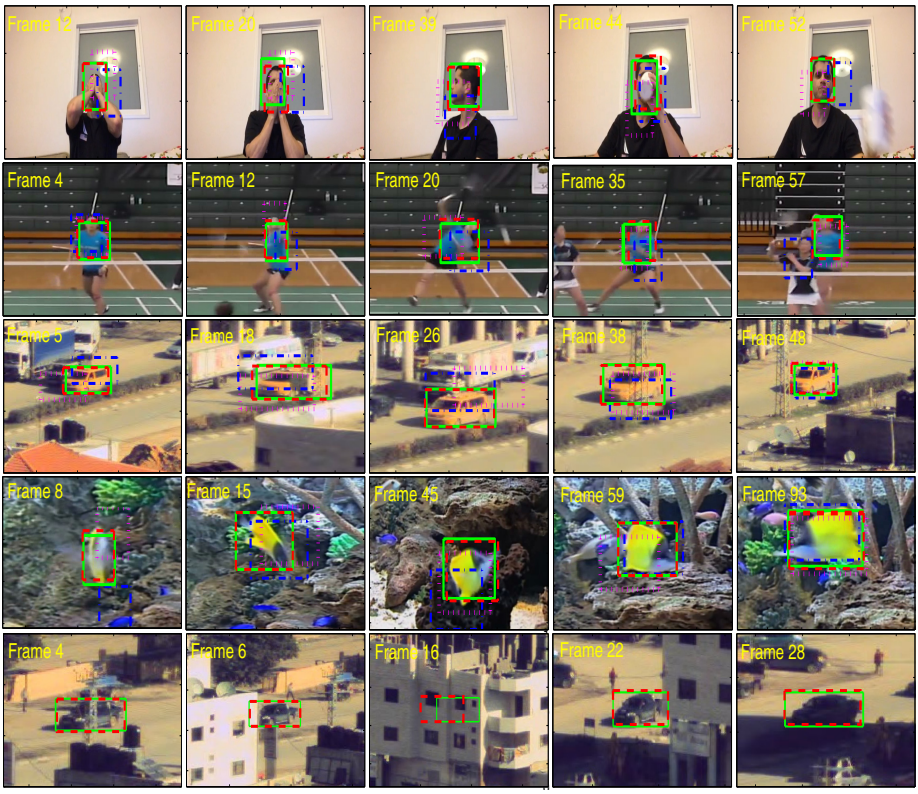

Fig. 4. Visual examples of LFR tracking for 3 and 5 frames per second videos with target objects marked by: ground truth - dashed line ; The proposed method - regular line ; KLT - dotted line ; particle filter - dashed-dotted line. The first four examples compare all the methods in different time frames. In the last example we compared our method to the ground truth.

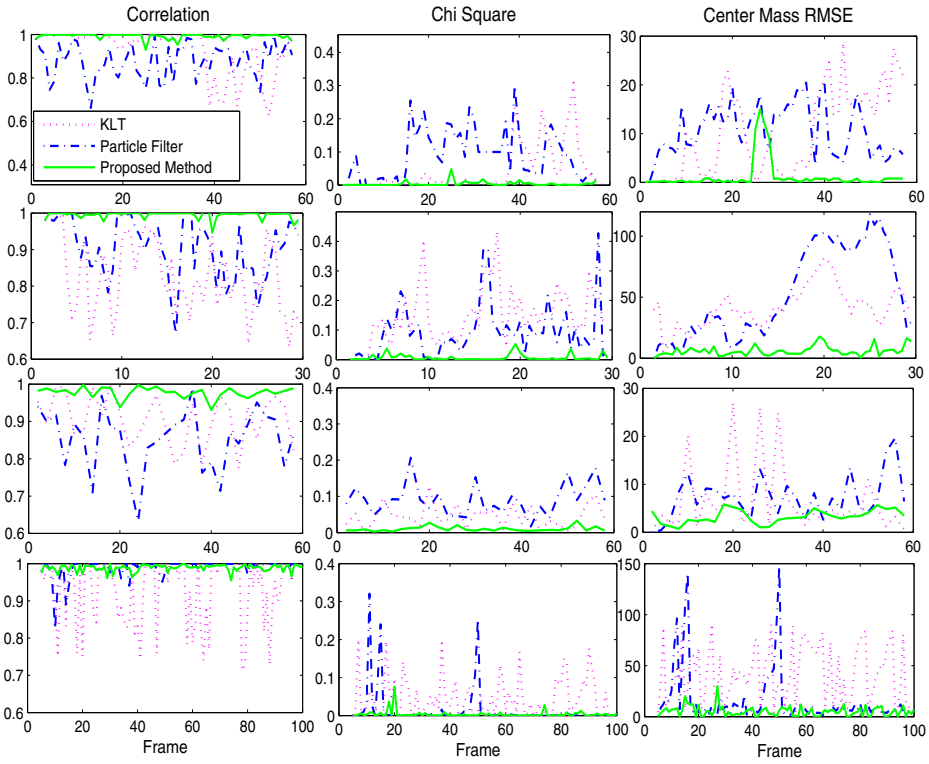

Fig. 5. Frame by frame performance evaluations of all methods, considering the Correlation coefficient (left), the χ^2 (middle) and RMSE (right) measures with respect to ground truth. The rows correspond to the first four objects from Fig. 4, respectively. The proposed method is depicted by a regular line ; the KLT - dotted line ; and the particle filter - dashed line.

Table 1. Performance evaluation

Method	Center Mass RMSE	Correlation (ρ)	Chi Square (χ^2)
Proposed Method	4.86	0.964	0.072
KLT	12.25	0.838	0.158
Particle Filter	13.97	0.843	0.201

5 Conclusions

This paper explores the concept of low frame rate tracking for the purpose of accurate target recovery by proposing a new scheme which in-cooperates matching and tracking in a complementary way. Regarding the matching, both statistical and deterministic comparison measures are considered, which are then integrated into an extended tracking methodology based on particle filtering.

The performance of the proposed method has been assessed by comparing its abilities for recovering a large data set of objects, to other benchmarks, KLT and particle filtering for visual tracking, with respect to ground truth annotated data, while inspecting different measures for comparison. Because object tracking in LFR is subject to many uncertainties, standard tracking methods make it hard to cope with, thus providing rather poor performance compared to the proposed tracking method, as shown in Table 1.

Regarding the particle filter performance with respect to that of the KLT, while a slight improvement was achieved in the correlation measure, the KLT accomplished better scores in terms of CM_{RMSE} and χ^2. Still, the suggested technique outperformed them, almost uniformly, while showing a clear improvement in all measures. The ability of the method to perform well even in extreme cases, in which the target object was completely occluded (for a long period, in a 3 FPS video) was also illustrated in Fig. 4, where the recovery was compared with that of the ground truth.

The method proposed in this paper is computationally fast, mainly due to the fact that the ROI used for the matching was generically limited, by taking into account the sizes of each template (and sub-templates) to be tracked.

Although the thresholds for verifying the matching confidence are derived in context of the data, they imply a large impact on the overall performance, thus presenting a possible limitation for tracking under different settings. In addition, since a well defined prior information is available, it can be used in order to set a better probability distribution for the particles, than the normal distribution.

In the future, improvement of the method and further robustness will surely be the focus, by possibly extending the detection mechanism for the template updating under LFR conditions, and considering other color and feature spaces.

Even though the proposed approach in this paper was used for the application of accurate recovery of target objects in LFR videos, it may also be used in other object tracking or video processing applications as well.

References

1. Zhong, B., Yao, H., Chen, S., Ji, R., Chin, T.J., Wang, H.: Visual tracking via weakly supervised learning from multiple imperfect oracles. Pattern Recognition 47, 1395–1410 (2014)
2. Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1940–1947. IEEE (2012)
3. Porikli, F., Tuzel, O.: Object tracking in low-frame-rate video. In: International Society for Optics and Photonics Electronic Imaging 2005, pp. 72–79 (2005)
4. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. International Journal of Computer Vision 29, 5–28 (1998)
5. Tomasi, C., Kanade, T.: Detection and tracking of point features. School of Computer Science, Carnegie Mellon Univ. Pittsburgh (1991)
6. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 142–149. IEEE (2000)

7. Ling, H., Wu, Y., Blasch, E., Chen, G., Lang, H., Bai, L.: Evaluation of visual tracking in extremely low frame rate wide area motion imagery. In: 2011 Proceedings of the 14th International Conference on Information Fusion (FUSION), pp. 1–8. IEEE (2011)
8. Palaniappan, K., Bunyak, F., Kumar, P., Ersoy, I., Jaeger, S., Ganguli, K., Haridas, A., Fraser, J., Rao, R.M., Seetharaman, G.: Efficient feature extraction and likelihood fusion for vehicle tracking in low frame rate airborne video. In: 2010 13th Conference on Information fusion (FUSION), pp. 1–8. IEEE (2010)
9. 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 Tran. on Pattern Analysis and Machine Intelligence 30, 1728–1740 (2008)
10. Liu, C., Shum, H.-Y., Zhang, C.: Hierarchical shape modeling for automatic face localization. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part II. LNCS, vol. 2351, pp. 687–703. Springer, Heidelberg (2002)
11. Okuma, K., Taleghani, A., de Freitas, N., Little, J.J., Lowe, D.G.: A boosted particle filter: Multitarget detection and tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
12. Chang, C., Ansari, R.: Kernel particle filter for visual tracking. IEEE Signal Processing Letters 12, 242–245 (2005)
13. Jiang, Y., Li, Z., Fang, J., Yue, J., Li, D.: Automatic video tracking of chinese mitten crabs based on the particle filter algorithm using a biologically constrained probe and resampling. Computers and Electronics in Agriculture 106
14. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: 6th International Conference on Computer Vision, pp. 59–66. IEEE (1998)
15. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
16. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision 22, 61–79 (1997)