Enhancing Correlation Filter based Trackers with Size Adaptivity and Drift Prevention

Emre Tunali, Sinan Öz and Mustafa Eral
Software Design Department, ASELSAN Inc. Microelectronics, Guidance and Electro-Optics Division, Ankara, Turkey

Keywords: Real-time Object Tracking, Size Adaptive Tracking, Track Drift Compensation, Track Initialization Error Compensation, Joint Segmentation and Tracking.

Abstract: To enhance correlation filter (CF) based trackers with size adaptivity and more robustness; we propose a new strategy which integrates an external segmentation methodology with CF based trackers in a closed feedback loop. Employing this strategy both enables object size disclosure during tracking; and automatic alteration of track models and parameters online in non-disturbing manner, yielding better target localization. Obviously, consolidation of CF based trackers with these properties introduces much more robustness against track center drifts and relaxes widespread perfectly centralized track initialization assumption. In other words, even if track window center is given with certain offset to center of target object at track initialization; proposed methodology achieves target centralization by aligning tracker template center with target center smoothly in time. Experimental results indicates that proposed algorithm increases performance of CF trackers in terms of accuracy and robustness without disrupting their real-time processing capabilities.

1 INTRODUCTION

Visual object tracking is a fundamental task in computer vision and has wide range of applications including surveillance, motion analysis, activity recognition, and human-computer interaction. Since target can belong to any object class that is requiring further analysis (i.e. vehicles on a road, pedestrians in the street, planes in the air etc.); trackers should handle large variety of appearance changes and that makes visual object tracking a challenging task in realistic scenarios. Moreover, many real-life application requires real-time processing; hence challenges including partial occlusions, pose variations, background clutter should be solved by using limited computational load.

To handle these challenges, object tracking has been studied for several decades; hence many surveys and benchmarking efforts exist to identify general trends, categorize various solutions and compare their performances (Yilmaz et al., 2006; Smeulders et al., 2014; Wu et al., 2013). Examining the visual tracker literature, correlation filter (CF) based tracking approaches are known to be one of the solution families that can achieve real-time tracking with comparable performance to other popular algorithms. However, this solution family suffers from two general restrictions. Firstly, CF based visual tracking algorithms keep track window size fixed through the scenario even if the target size changes. Static track window size not only restricts the adaption of tracker, but also strips off size information from tracker output which can be beneficial for many other tasks including automated surveillance and motion-based recognition. Secondly, since CF based trackers are not aware of the target; their templates (filters) can be easily polluted by track drifts which can be either observed during tracking, maneuvering targets, or introduced at the beginning of scenario due to imperfect track initializations. Most of the tracker algorithms evade from drifts at the first frame by requiring perfect initialization, i.e. perfectly sized target bounding box that is centralized at the target object center, from human users. However, in many real-life tracking systems this requirement can not be fulfilled and tracker might be initialized with a shifted track window. Even in ideal conditions, CF based trackers proposed in literature are expected to carry on tracking with the same amount of shift with the initialization throughout the scenario. In any case, to reduce undesired bias in localization error (due to track window shifts) and decrease track loss probability, these restrictions should be addressed. To handle these issues, we propose a new methodology that integrates an external

Tunali, E., Oz, S. and Eral, M.
Enhancing Correlation Filter based Trackers with Size Adaptivity and Drift Prevention.
DOI: 10.5220/0006680404720480
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 5: VISAPP*, pages 472-480
ISBN: 978-989-758-290-5

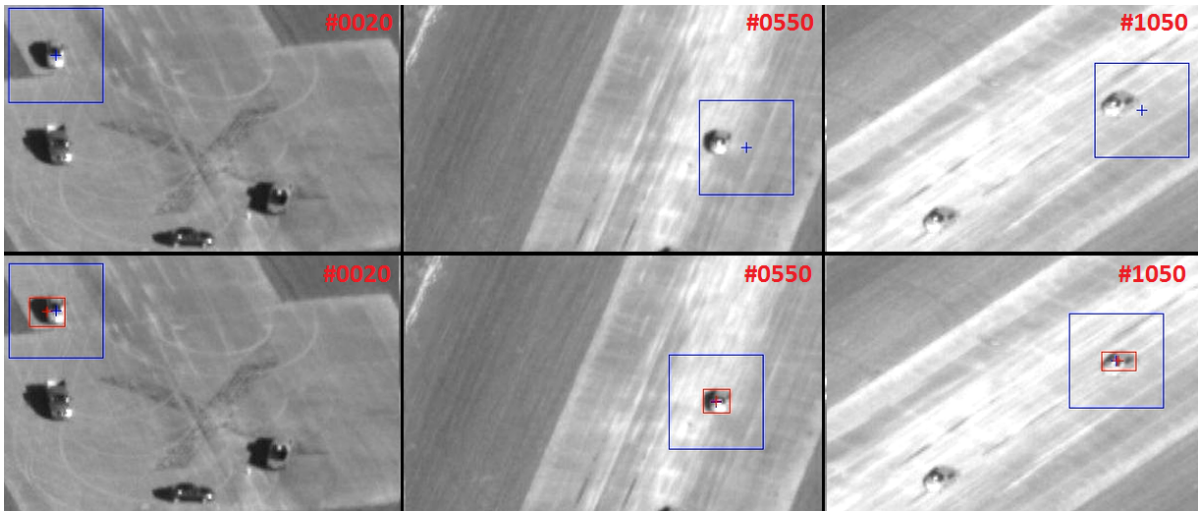

Figure 1: Top Row: CF tracker (Bolme et al., 2010) using static track window size (blue) suffers from track drift. Bottom Row: Enhanced tracker adapts to target size (red) and compensates drifts.

segmentation strategy with CF based trackers in a closed feedback loop. Employing this strategy, any CF based tracker becomes capable of revealing target bounding box within the track window at each frame and allows tracker to update its models, parameters online in non-disturbing manner; yielding better target localization. Cooperation introduced in the proposed strategy also proposes a remedy for erroneous track initializations can be made by human users; by aligning tracker template center with target center during tracking.

In summary, main contributions of this paper are three-fold: (1) developing an external segmentation method for revealing target bounding box information at each frame, (2) integrating this information with CF based tracking for size adaptivity and drift prevention, and (3) relaxing the widespread perfect track initialization assumption. It is important to emphasize that all these contributions are achieved without violating real-time processing constraints. The remainder of the paper is organized as follows: We first review related work in Section 2. In Section 3, the main framework of our algorithm is introduced. Then, the experimental results are presented in Section 4, and followed by conclusions in Section 5.

2 RELATED WORK

CF based Visual Tracking. Correlation filters have been investigated for three decades due to their attractive properties (shift invariance, robustness to graceful degradation, distortion tolerance) and employed in many applications. The basic idea behind learning scheme of CFs is to learn filters that optimally map input images to their ideal output. The ideal output is a peak (or a value of one) at position of the target and zeroes for all other locations in the image (Dirac Delta Function). Filters trained in this way, not only produce high responses for targets but also learn to suppress the response to common background distracters. In other words, correlation filters are designed to identify patterns that are consistent through the video sequence. Hence, they are more tolerant of common appearance changes than simple template matching and produce more prominent peaks in the target locations.

Minimum Average Correlation Energy (MACE) filter (Mahalanobis et al., 1987) is one of the first examples of CFs that are trained for localization purposes by using various target samples. This successful matching scheme lead design of other "constrained" filters including Optimal Tradeoff Synthetic Discriminant Function (OTSDF) (Refegier and Figue, 1991), Minimum Squared Error Synthetic Discriminant Function (MSESDF) (Kumar et al., 1992), and the Minimum Noise and Correlation Energy (MINACE) (Ravichandran and Casasent, 1992). However, applying constrains in filter learning, restricted generalizations to appearance changes which is a must for better localization. By relaxing constraints in training "Unconstrained" MACE (UMACE) (Mahalanobis et al., 1994) achieved higher average responses; hence fitted better for tracking applications due to improved generalization capability. "Optimized Correlation Output Filters" are the most recent examples of the CF family. Unlike prior training methods that recombine templates, these filters consider image to image mapping that is performed during cor-

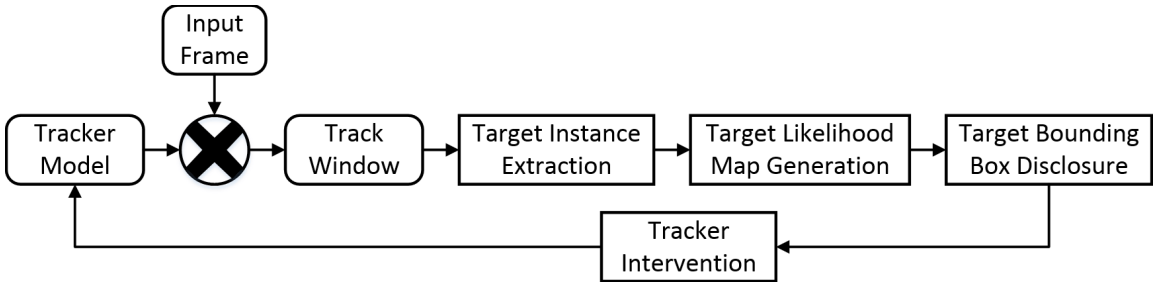

Figure 2: Overview of the proposed algorithm.

relation with synthetic outputs and inverts this mapping to produce ideal filters. Average of Synthetic Exact Filter (ASEF) (Bolme et al., 2009a) and Minimum Output Sum of Squared Error (MOSSE) filter (Bolme et al., 2010) are well-known examples of this approach which results in complete control over correlation plane rather than specifying a single peak as is done in previous CFs, hence fits better for the visual tracking purposes.

The superiority of CF based visual tracking comes from not only their unique way of template (filter) learning; but also their speed in matching. The simplest form of matching utilizes following steps. Firstly, the carefully designed template and query image is cross-correlated. Then, correlation output is searched for the most prominent peak by using a metric, such as peak-to-sidelobe ratio (PSR) or peak-to-correlation energy (PCE), which is designed to indicate likelihood of the target presence. More prominent the peak, target existence likelihood is higher at the indicated location. If the match quality is above a threshold, location of the prominent peak reveals location of the target. If not, target is stated to be occluded or lost. This matching scheme is employed by (Bolme et al., 2009b; Bolme et al., 2010; Henriques et al., 2015) and can be applied regardless of filter learning mechanism.

Joint Segmentation and Tracking. Image segmentation is the task of assigning each pixel of an image to a particular class label and has been extensively studied (Donoser and Schmalstieg, 2014; Taylor, 2013) since it is considered as a critical task for scene understanding. Benefiting from temporal information, video segmentation extends this idea to video volumes (Galasso et al., 2014). Hence, the main objective turns into assigning consistent pixel labels throughout scene that is being analyzed.

Tracking and segmentation can be considered as related issues since a successful object/background segmentation also means successful tracking. Similarly, tracking also provides strong cues for object/background segmentation especially in videos that are taken by moving camera. Therefore, literature includes examples of joint segmentation and tracking (Wen et al., 2015). In these methods, segmentation is mainly achieved for revealing exact contours of the object of interest. However, this task can be considered as over complex for our objectives. Actually, our goal is to give feedback to tracker for both enhanced localization and adaptation of track window size in real-time. Therefore, obtaining bounding box of object would be sufficient rather than revealing fine boundaries which is more appropriate for real-time processing. (Stalder et al., 2012), benefits from "objectness" definition to give feedback to tracker for better localization and drift prevention. Although size adaptive tracking is achieved in (Stalder et al., 2012), algorithm is limited to 5 fps and no effort is made for generalizing solution to other trackers.

3 METHODOLOGY

Our method integrates CF based trackers with proposed adaptive segmentation strategy with a feedback loop to adjust tracker models and parameters in online and non-disturbing manner. Procedure starts with the resultant track window produced by any CF based tracker. Then, target instance is obtained by executing (Zhu et al., 2014) within this track window. Superimposing instances according to proposed adaptive learning scheme, target likelihood map is obtained and used for target bounding box disclosure. This bounding box is not only reported as tracker output but also used for aligning track window with tracker templates (CFs) and adjusting other size dependent tracker parameters. Although tracker intervention has numerous merits, non-smooth changes of templates or track parameters may abruptly decrease track quality. Hence, adjustment pace is arranged in order not to disturb own flow of the tracker. General overview of algorithm flow is illustrated in Fig. 2.

3.1 Target Instance Extraction

Proposed solution benefits from two simple yet effective priors: (1) Track window always contains an object of interest; (2) Target object is more frequently visible than background. Actually, both of these assumptions are very natural for numerous tracking scenarios due to general tendency that tracker is initiated to observe specific objects in varying conditions.

To achieve target/background segmentation independent of object models, visual saliency exploration is one of the most commonly used strategy in the literature. Although a recent visual saliency benchmark (Borji et al., 2015) reveals various solutions, we employed RBD (Zhu et al., 2014) since it has unique benefits that are fitting better for our task. First of all, apart form many conventional methods, RBD does not depend only on contrast difference; it also benefits from the geometrical interpretation of the given image by examining boundary connectivity. This metric is based on the statement "object regions are much less connected to image boundaries than background ones" which fits perfectly for our application since the proposed tracker aims centralization of target resulting in decreased probability of target contact with track window boundaries. Secondly, usage of SLIC super pixels, (Achanta et al., 2012), in calculation of saliency map results in better fits for target boundaries leading to better target bounding box estimation. Finally, according to (Borji et al., 2015) and experimental results, it is revealed that RBD can achieve real-time processing constraints with less computational power requirement than other methods with comparable performance. Therefore, at each frame we employed RBD in track window for achieving target instances as saliency maps independent from temporal information. Each instance is treated as new information about target and superimposed in an adaptive manner for target likelihood map generation.

3.2 Target Likelihood Generation via Instance Quality Assessment

In order to reveal target likelihood, all extracted target instances should be examined. However, effect of each instance on likelihood map should differ since instances cannot represent the common properties of the target with the same quality or they are aged over time. For this reason, target likelihood is generated by following 1^{st} order IIR filter structure with adaptive learning rate $\lambda[n]$. By applying Eqn.1, instances ($S_{instance}[n]$) are superimposed in form of saliency maps to disclose target likelihood ($S_{target}[n]$) at each frame (n).

$$S_{target}[n] = (1-\lambda[n]) \cdot S_{target}[n-1] + \lambda[n] \cdot S_{instance}[n] \,. \tag{1}$$

Learning rate is directly proportional to quality of instances which is measured based on two properties: distinctiveness and consistency. Distinctiveness metric is designed to measure distribution of saliency between foreground and background pixels. In other words, higher saliency values in target pixels rather than background indicates clear representation of target instance which should be benefited in higher rates. Distinctiveness ($d_s(t)$), is calculated as in Eqn.2 where target instance is binarized and foreground is classified as the pixels having saliency values greater than the binarization threshold achieved by Otsu's (Otsu, 1979).

$$d_s[n] = \frac{\sum_{x \in Foreground} S_{instance}(x)}{\sum_{\forall x} S_{instance}(x)} \,. \tag{2}$$

One should note that distinctiveness metric benefits from target instance only; hence does not include any temporal information. However, consistency of target instances inherently shows the absence of abrupt changes indicating high quality of instances. In this manner, consistency score $c_s[n]$, is calculated as the maximum value of the normalized cross correlation score between target instance and target likelihood. Since high consistency is signature of confiding target, consistency metric also determines when tracker should be intervened.

Using these metrics, adaptive learning rate $\lambda[n]$ is calculated as in the Eqn.3 at each frame where α is maximum learning rate constant, β is penalization constant which is defined to prevent mislearning of target model in the presence of inconsistent target instances and C_{thres} is the consistency threshold.

$$\lambda[n] = \begin{cases} \alpha \cdot d_s[n] \cdot c_s[n], & c_s[n] > C_{thres} \\ \beta \cdot \alpha \cdot d_s[n] \cdot c_s[n], & c_s[n] \leq C_{thres} \end{cases} \,. \tag{3}$$

3.3 Target Bounding Box Disclosure

Although likelihood for intended target pixels is known to be high, all pixels having high likelihood do not necessarily belong to target since the track window may also contain other objects or their parts in the vicinity of intended target. In other words, after binarization, target pixels should be selected among the foreground pixels. Intuitively speaking, target bounding box should contain most salient region with minimum distance to the center. *Target Bounding Box*(TBB) is disclosed by revealing the *Bounding Box*(BB) of the connected component

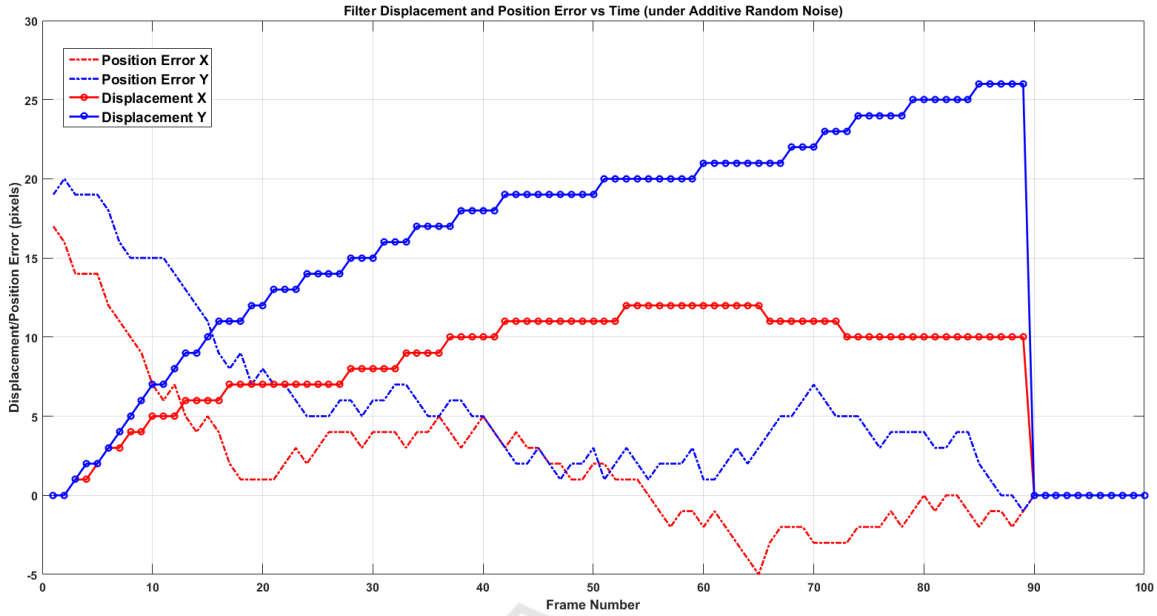

Figure 3: Exemplary filter shift pace to cover [17,19] pixel distance. Random noise in range [-2,2] is introduced at each frame to simulate localization error of tracker.

maximizing the targetness cost given by Eqn.4 as proposed in (Tunali and Öz, 2014).

$$TBB = BB\left[argmax_{C_i}\frac{\sum_{x\in C_i} S_{target}[n](x)}{\sqrt{(x_i-x_c)^2+(y_i-y_c)^2}}\right], \quad (4)$$

Note that, C_i is the 2D label matrix with values 1 for i^{th} connected component and 0 otherwise, (x_i, y_i), (x_c, y_c) are centers of each connected components and track window.

3.4 Tracker Intervention

In CF based trackers, filters (templates) are learned being unaware of the target. Therefore, when track is initiated imperfectly, unaligned with the target, or tracker faces with drifts during the process, generated filters become misplaced and that makes track drifts permanent. To achieve drift prevention, filters should be shifted back to their ideal locations without poisoning their history and nature. By aligning filter center with target bounding box obtained in Sec.3.3, CF filters achieves awareness on shape of target object.

Even though energy of the filter can be kept same by applying circular shifts, shifted filter would include some artificial responses that can only be recovered through combining with natural samples in time. In this sense, applying shifts in large amounts or repeatedly in consecutive frames would poison naturality of filter yielding abrupt decreases in track quality. On the contrary, limiting shifts too harshly would reduce convergence rate in filter alignment. To achieve proper shifting pace; adaptive step sizes, proportional to target likelihood learning rate $\lambda[n]$, are utilized while covering distance between current and ideal filter position. Maximum velocity or acceleration thresholds are used for allowing consecutive shifts in minimum N frames. Filter alignment process is summarized in Algorithm 1 where $\vec{v}[n]$, $\vec{a}[n]$ are velocity and acceleration; $\Delta\vec{x}_{integer}[n]$ is integer part of accumulated desired filter shift ($\Delta\vec{x}_{acc}[n]$) and

Algorithm 1: Target Center and Correlation Filter (Template) Alignment Procedure.

INPUT: *position error:* $\vec{e}[n] \in \Re^2$, *step size:* $\vec{\mu}[n] \in \Re^2$, *Template:* $H[n](u,v) \in \Re^{NxM}$, *consistency score:* $c_s[n] \in \Re$

while $(c_s[n] > C_{thres})$ **do**
  $\vec{v}[n] \leftarrow \vec{\mu}[n] \cdot \vec{e}[n]$
  $\vec{a}[n] \leftarrow \vec{v}[n] - \vec{v}[n-1]$
  if $(\vec{a}[n] > a_{thres})$ **then** $\vec{v}[n] \leftarrow \vec{v}[n-1] + a_{thres}$
  if $(\vec{v}[n] > v_{thres})$ **then** $\vec{v}[n] \leftarrow v_{thres}$
  $\Delta\vec{x}_{acc}[n] \leftarrow \Delta\vec{x}_{acc}[n-1] + \vec{v}[n]$
  $\Delta\vec{x}_{integer}[n] \leftarrow fix(\Delta\vec{x}_{acc}[n])$
  if $(\Delta\vec{x}_{integer}[n] > v_{thres})$ **then** $\Delta\vec{x}_{integer}[n] \leftarrow v_{thres}$
  $\Delta\vec{x}_{acc}[n] \leftarrow \Delta\vec{x}_{acc}[n] - \Delta\vec{x}_{integer}[n]$
  $H[n](u,v) = H[n](u,v). * \exp(-j2\pi \cdot [u/N, v/M] \cdot \vec{x}_{integer}[n])$
end while
return $H[n](u,v)$

Table 1: Real-time processing constraints can be satisfied up to 128x128 target sizes.

Track Window Size	32x32	64x64	128x128	256x256
Processing Time (ms)	2.272	10.129	35.961	151.056

represents amount of filter shift applied at each frame.

Online parameter tuning is another interaction with tracker. In many CF based tracker, size dependent parameters are set at the track initialization and kept fixed since target size is not known. Although size dependent parameters can vary from different trackers, our method allows their online adjustments resulting in better adaptability and increased capability of rejection of similar targets in the vicinity. For (Bolme et al., 2010), density distribution of preprocessing window and size of PSR window are such parameters. Unified effect of online parameter update together with filter alignment are given in Sec.4.

4 EXPERIMENTAL RESULTS

Testing our solution in the presence of scale changes, track drifts and erroneous track initialization is crucial since these issues are primarily addressed in this paper. Therefore, we evaluated performance on scenarios from three different datasets. From Vivid (Collins et al., 2005) *egtest01-02-03*; from Aircraft tracking (Mian, 2008) *small1, occlusion1*; and from CVPR2013 benchmark (Wu et al., 2013) *Sylvester, Walking, Walking2* are selected. These datasets fits for testing amendments of drift prevention and size adaptivity since they are dominated by targets having maneuvers, in-plane and out-of plane rotations, scale changes and deformations that generally causes track drifts and loses. To achieve more solid results, number of scenarios obtained from *egtest01-02-03* is increased to 10 in total (5, 3, and 2 respectively) by tracking auxiliary targets,whose ground truths are manually labeled, in addition to main targets given in (Collins et al., 2005).

During the experiments main attention is paid on quantifying the performance improvement by comparing base trackers with their enhanced versions. For evaluating performance of trackers, methodology and metrics proposed in (Wu et al., 2013) is followed. Hence, success and precision plots are generated to reveal track success rates (percentage of frames in which tracking is maintained) by measuring two different error types; target bounding box overlap and centralization errors. To be more precise, success plots uses a common overlap score which is defined as $S = \frac{|r_t \cap r_g|}{|r_t \cup r_g|}$ where $r_t$is output target bounding box and r_g ground truth bounding box. Although comparing overlap score with a fixed threshold is enough to obtain track success rate, success plot is generated by sweeping this threshold from 0 to 1 for better characterization. In precision plot, track success rate is disclosed based on center location error (CLE) that measures euclidean distance between ground truth and output track window centers. Similar to success plot, precision plot is also generated by comparing distance with a threshold ranging from 0 to 50. In order to rank trackers in precision plot, performances at CLE 15 is used while 0.5 is selected for success plot. To investigate whether the proposed scheme introduces robustness to initializations, temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE) are carried out together with one-pass evaluation (OPE) as is proposed in (Wu et al., 2013). OPE is the conventional scheme in which initialization is achieved perfectly at the first frame and tracker runs through whole scenario. In TRE analysis, scenario is divided into 20 segments and tracks are perfectly initialized at the first frame of these segments. In SRE analysis, erroneous track initializations are simulated by giving 8 spatial shifts including 4 center shifts and 4 corner shifts, and 4 scale variations. Spatial shifts are given in 10% of target size while scale ratios are 0.8, 0.9, 1.1 and 1.2 to the ground truth. For the parameter setting of the base trackers, we set them as default. Target instance extraction requires single parameter that is slic super pixel area and set to 65. For target likelihood map generation maximum learning rate constant $\alpha = 0.05$, penalization constant $\beta = 0.3$ and *consistency threshold = 0.85* are used. For filter alignment maximum acceleration (a_{thres}) and velocity (v_{thres}) thresholds are set to 0.2, and 2 while step size $\mu[n]$ is set to $0.3\lambda[n]$.

Figure 4 illustrates success and precision plots of 6 base trackers together with their enhanced versions while Table 2 quantitatively compares base trackers directly with their enhanced versions to disclose the impact of proposed solution on each of the trackers and the effect on the average. According to Table 2, proposed solution boosts performance of almost each tracker at each performance aspect. Achieved improvement on CLE and overlap metrics indicates that proposed solution is successful at both centralization and size disclosure. Smallest performance increase is achieved in TRE (overlap 6.2%, CLE 5.1) since base trackers also cannot achieve high track success rates due to low contrast and frequent occlusions. Obviously, boost in OPE (overlap 10.2%, CLE 12.6) is much better than TRE since base trackers have higher track success rates which provides proposed solution additional time for better target learning. SRE is the

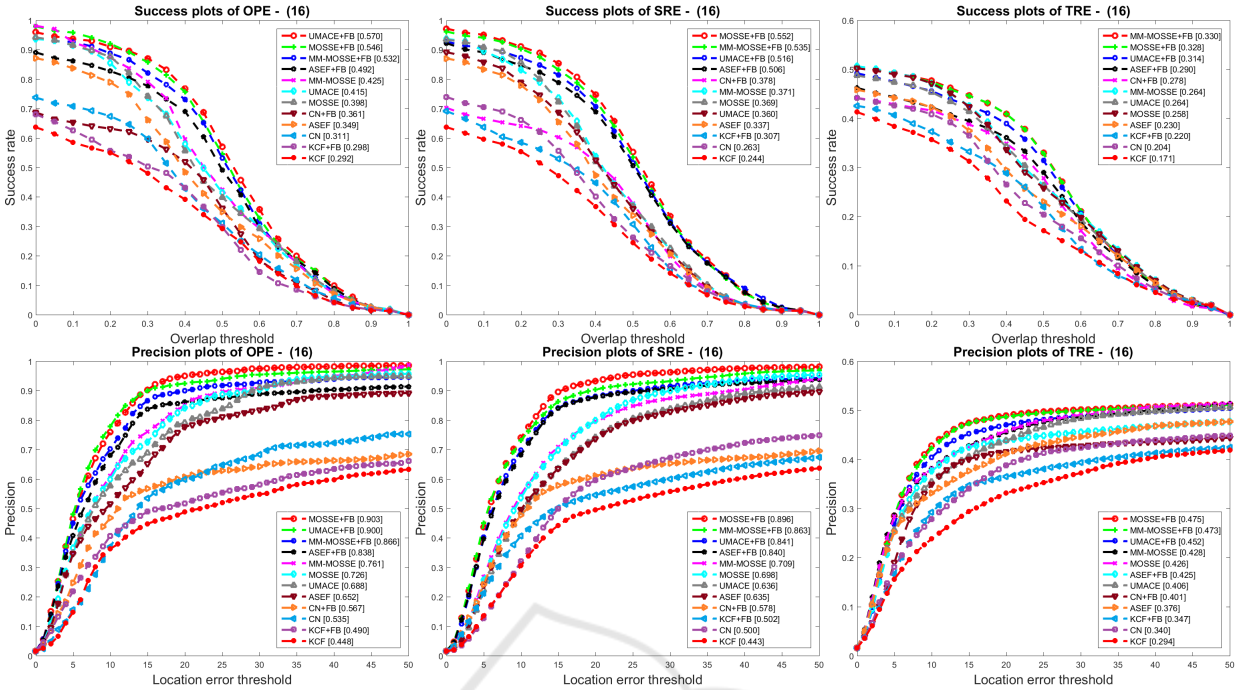

Figure 4: Success and precision plots for all base tracker and enhanced versions on 16 different scenarios from Vivid (Collins et al., 2005), Aircraft tracking (Mian, 2008) and CVPR2013 benchmark (Collins et al., 2005) datasets. Performance scores are shown in the legend (best viewed in color).

Table 2: Performance gains obtained in OPE, SRE and TRE analysis by proposed solution for each tracker. Highest and second highest gains are represented by red and blue.

All Scenarios		MM-Mosse (Tanisik and Gundogdu, 2015)	Mosse (Bolme et al., 2010)	ASEF (Bolme et al., 2009a)	UMACE (Mahalanobis et al., 1994)	CN (Danelljan et al., 2014)	KCF (Henriques et al., 2015)	Average
OPE-overlap	Base	0.425	0.398	0.349	0.415	0.311	0.292	0.365
	Base+FB	0532	0.546	0.492	0.570	0.361	0.298	0.467
	Gain	0.107	**0.148**	0.143	**0.155**	0.050	0.006	0.102
SRE-overlap	Base	0.371	0.369	0.337	0.360	0.263	0.244	0.324
	Base+FB	0.535	0.552	0.506	0.516	0.378	0.307	0.466
	Gain	0.164	**0.183**	**0.169**	0.156	0.115	0.063	0.142
TRE-overlap	Base	0.264	0.258	0.230	0.264	0.204	0.171	0.232
	Base+FB	0.330	0.328	0.290	0.314	0.278	0.220	0.293
	Gain	0.066	**0.070**	0.060	0.050	**0.074**	0.049	0.062
OPE-distance	Base	0.761	0.726	0.652	0.688	0.535	0.448	0.635
	Base+FB	0.866	0.903	0.838	0.900	0.567	0.490	0.761
	Gain	0.105	0.177	**0.186**	**0.212**	0.032	0.042	0.126
SRE-distance	Base	0.709	0.698	0.635	0.636	0.500	0.443	0.604
	Base+FB	0.863	0.896	0.840	0.841	0.578	0.502	0.753
	Gain	0.154	**0.198**	**0.205**	**0.205**	0.078	0.059	0.149
TRE-distance	Base	0.428	0.426	0.376	0.406	0.340	0.294	0.378
	Base+FB	0.473	0.475	0.425	0.452	0.401	0.347	0.429
	Gain	0.045	0.049	0.049	0.046	**0.061**	**0.053**	0.051

most significant analysis type for proposed solution since it directly evaluates robustness gained against erroneous initialization and track drifts, which are the major contributions of the paper. In addition to significant average SRE gains (14.2%, 14.9%), comparing OPE and SRE scores also reveals impact of proposed solution. To be more precise, for any tracker, SRE scores are expected to be lower than OPE since perfect initialization is achieved in OPE while SRE exposes perturbed initializations. Examining Table 2 reveals that perturbed initializations yields less performance decrease in trackers enhanced with feedback. Obviously, these improvements requires computational load proportional to track window size. Table 1, indicates required average processing times for various target sizes. Results are obtained from a single core of an Intel i7-2670QM CPU @2.20 GHz processor with an unoptimized MATLAB code.

5 CONCLUSIONS

We presented a novel adaptive segmentation and feedback mechanism to enhance any CF based tracker with target size output and more robustness. Key to the achieved performance boost is benefiting from target bounding box to align target and template (filter) centers by applying gradual shifts in a non-disturbing manner. Experiments revealed that proposed solution makes CF based trackers more practical in real-life scenarios by tolerating erroneous track initializations. It would be interesting to investigate effect of rota-

Figure 5: Exemplary results of performance boosts. Base and boosted versions are illustrated in blue and red. Information rows on each subfigure includes Slic superpixels, correlation filter, correlation output, target likelihood map and target instance, respectively.

ting and scaling filters to achieve even more robustness and extend our experiments to other challenging datasets.

REFERENCES

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Ssstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282.

Bolme, D. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M. (2010). Visual object tracking using adaptive correlation filters. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2544–2550. IEEE.

Bolme, D. S., Draper, B. A., and Beveridge, J. R. (2009a). Average of synthetic exact filters. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2105–2112. IEEE.

Bolme, D. S., Lui, Y. M., Draper, B. A., and Beveridge, J. R. (2009b). Simple real-time human detection using a single correlation filter. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pages 1–8. IEEE.

Borji, A., Cheng, M.-M., Jiang, H., and Li, J. (2015). Salient object detection: A benchmark. *Image Processing, IEEE Transactions on*, 24(12):5706–5722.

Collins, R., Zhou, X., and Teh, S. K. (2005). An open source tracking testbed and evaluation web site. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, volume 2, page 35.

Danelljan, M., Khan, F., Felsberg, M., and Weijer, J. (2014). Adaptive color attributes for real-time visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1090–1097.

Donoser, M. and Schmalstieg, D. (2014). Discrete-continuous gradient orientation estimation for faster image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3158–3165.

Galasso, F., Keuper, M., Brox, T., and Schiele, B. (2014). Spectral graph reduction for efficient image and streaming video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–56.

Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. (2015). High-speed tracking with kernelized correlation filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(3):583–596.

Kumar, B. V., Mahalanobis, A., Song, S., Sims, S. R. F., and Epperson, J. F. (1992). Minimum squared error synthetic discriminant functions. *Optical Engineering*, 31(5):915–922.

Mahalanobis, A., Kumar, B., and Casasent, D. (1987). Minimum average correlation energy filters. *Applied Optics*, 26(17):3633–3640.

Mahalanobis, A., Vijaya Kumar, B., Song, S., Sims, S., and Epperson, J. (1994). Unconstrained correlation filters. *Applied Optics*, 33(17):3751–3759.

Mian, A. S. (2008). Realtime visual tracking of aircrafts. In *Digital Image Computing: Techniques and Applications (DICTA), 2008*, pages 351–356. IEEE.

Otsu, N. (1979). A threshold selection method from gray-level histogram. *IEEE Transactions on System Man Cybernetics*, SMC-9.

Ravichandran, G. and Casasent, D. (1992). Minimum noise and correlation energy optical correlation filter. *Applied Optics*, 31(11):1823–1833.

Refegier, P. and Figue, J. (1991). Optimal trade-off filters for pattern recognition and their comparison with the wiener approach. *Optical Computing and Processing*, 1(3):245–266.

Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., and Shah, M. (2014). Visual tracking: An experimental survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1442–1468.

Stalder, S., Grabner, H., and Van Gool, L. J. (2012). Dynamic objectness for adaptive tracking. In *ACCV (3)*, pages 43–56.

Tanisik, G. and Gundogdu, E. (2015). Multiple model adaptive visual tracking with correlation filters. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 661–665. IEEE.

Taylor, C. (2013). Towards fast and accurate segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1916–1922.

Tunali, E. and Öz, S. (2014). A real-time, semi-automatic method for discriminant target initialization in thermal imagery. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 5117–5121.

Wen, L., Du, D., Lei, Z., Li, S. Z., and Yang, M.-H. (2015). Jots: Joint online tracking and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2226–2234.

Wu, Y., Lim, J., and Yang, M.-H. (2013). Online object tracking: A benchmark. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yilmaz, A., Javed, O., and Shah, M. (2006). Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13.

Zhu, W., Liang, S., Wei, Y., and Sun, J. (2014). Saliency optimization from robust background detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.