Journal of Engineering and Applied Sciences 14 (1): 293-307, 2019
ISSN: 1816-949X

Kernelized Correlation Filters Parameters Optimization for Enhanced Visual Tracking

Chor Keat Ong and Parvathy Rajendran
School of Aerospace Engineering, Universiti Sains Malaysia, Engineering Campus,
14300 Nibong Tebal, Pulau Pinang, Malaysia

Abstract: Visual tracking has become one of the most important components in computer vision as the knowledge in this field can be applied into a wide range of applications in computer vision such as medical imaging, pattern recognition, video surveillance industrial robot, computer-human interaction, etc. A lot of researches have been conducted and many types of state-of-the-art methods and modifications such as sparse representation, online similarity learning, self-expressive, spatial kernel phase correlation filter and others are proposed in order to increase the robustness of the tracking. Despite of many methods has been demonstrated successfully but there are several issues that still need to be addressed. There still have some unsolvable difficulties in which they become a challenging task to track an object effectively and robustly and it will tend to decrease the accuracy of the results and hence. Until now, there are still no perfect algorithm to track the target flawlessly. In order to improve the performance, the main idea proposed is implementing optimization technique on the selected parameters and obtain a better performance. In this research, the tracking is proposed by using the Overlap Ratio (OR) and Centre Location Error (CLE). In our case, our target is to obtain a better accuracy which is higher OR and lower CLE than the result from the algorithms available in public. A simple optimization is used in here where the global best results with respect to the value of the parameters are selected through a range of values defined in our research. Through the optimization, the overall OR is increased to 0.554 and overall CLE is decreased to 19.803 pixels. Thus, the proposed method had increased the accuracy and robustness of the visual tracking on many of the video sequences.

Key words: Visual tracking, computer vision, kernelized correlation filters, parameter optimization, CLE, OR

INTRODUCTION

Visual tracking is one of the research field in computer science that increases in popularity due to the importance to many applications in computer vision such as medical imaging, pattern recognition, video surveillance industrial robot, computer-human interaction, etc. Basically, the main objective of visual tracking is to estimate and locate the target objects in consecutive video frames.

In general, the basis working principle of visual tracking is after inputting a video sequence, we require a description for the object to be tracked. For example, shape, colour model, texture or others can use to be the template image of the object. Next, some context is applied into the object as implementing a good and proper integration of such context information into a tracking framework will bring some positive effects to visual tracking. After context information integration, the classifier classifies the image patches, then updated from time to time which is also known as online learning in order to handle and adapt the new appearance changes in the subsequent frames. These steps are repeated to track the object and stop when it reaches the last frame of the video.

Although, visual tracking has been studied for several decades but it is remaining as a challenging topic to be researched as mainly due to abrupt object motion, appearance pattern change, non-rigid object structures, occlusion and camera motion. And thus, there are no a single comprehensive method to handle all these destabilizing factors where these destabilizing factors are mainly consist of 11 attributes which are shown in Table 1 with description respectively (Wu *et al*., 2013; Yang *et al*., 2011).

In recent years, different algorithms have been proposed in order to solve the challenging issues. One of the methods is choose the right features or the most desirable property of a visual feature in order to be distinguished in the feature space easily. So, feature descriptors are playing an important role in selecting the

Corresponding Author: Parvathy Rajendran, School of Aerospace Engineering, Universiti Sains Malaysia, Engineering Campus, 14300 Nibong Tebal, Pulau Pinang, Malaysia

Table 1: The 11 attributes annotated to test sequences with threshold values provided

Name	Description
Illumination Variation (IV)	The illumination in the target region is significantly changed
Scale Variation (SV)	The ratio of the bounding boxes of the first frame and the current frame is out of the range [$1/t_s$, t_s], t_s >1 (t_s = 2)
Occlusion (OCC)	The target is partially or fully occluded
Deformation (DEF)	Non-rigid object deformation
Motion Blur (MB)	The target region is blurred due to the motion of target or camera
Fast Motion (FM)	The motion of the ground truth is larger than tm pixels (t_m = 20)
In-Plane Rotation (IPR)	The target rotates in the image plane
Out-of-Plane Rotation (OPR)	The target rotates out of the image plane
Out-of-View (OV)	Some portion of the target leaves the view
Background Clutters (BC)	The background near the target has the similar color or texture as the target
Low Resolution (LR)	The number of pixels inside the ground-truth bounding box is less than (t_r = 400)

right features. For instances, gradient feature is proved to have advantageous in human detection (Dalal and Triggs, 2005; Sabzmeydani and Mori, 2007); colour features which are robust against certain photomatic changes, texture features where texture is used to measure the intensity of a surface and quantifies properties such as smoothness and regularity (Fergus *et al*., 2003; Shotton *et al*., 2009; Winn *et al*., 2005), spatio-temporal features which used as representation for action recognition and visual detection, multiple features fusion which is more robust for image and video retrieval, visual tracking and detection.

Despite these feature descriptors, visual tracking still requires online learning based tracking methods to handle appearance variations of a target object. Online learning is required in for the tracker to adapt these appearance changes, adjust and update to new situations from time to time. There are 2 types of appearances variations which are intrinsic (pose changing, shape deformation) and extrinsic (occlusion, camera motion, camera viewpoint and illumination variation). Thus, these appearance variations must be handled by the online learning algorithm which is divided into 2 categories: generative method and discriminative method.

Appearance model: Generally, generative online learning method will learn the appearance of the object, then it will update online on the object model in order to adapt the appearance changes. Adam *et al*. (2006) represented the target using integral histogram and robust in target with partial occlusions or pose changes. Ross *et al*. (2008) presented an appearance-based tracker to gradually learn a low dimensional Eigen basis representation for tracking the target that with changing pose, illumination and appearance from time to time. Ross *et al*. (2008) Model is satisfying but it will encounter drifting problem. Jia *et al*. (2012) implemented a template update strategy which incremental subspace learning and sparse representation are combined together. The adaption of the template reduces possibility of drifting and the effect of the occluded target template.

Bao *et al*. (2012) proposed by adding an l2 norm regularization on the coefficients associated with the trivial templates into a new l1 norm related minimization model, it can achieved a better tracking accuracy than other l1 tracker, (Mei and Ling, 2009; Mei *et al*., 2011). Mei and Ling (2009) casted the tracking as a sparse approximation problem in a particle filter framework and achieved a very promising tracking result. Mei *et al*. (2011) presented a new approach known as Bounded Particle Resampling (BPR)-L1 tracker to enhance the template updates by detect occlusions and lessen the drifting problem.

Liu *et al*. (2013) proposed a new selection-based dictionary learning method known as K-selection and modelled the target appearance by using a sparse coding histogram based on a learned dictionary. By this way, it can adapt to appearance changes and drifting problem is reduced. Liu *et al*. (2010) proposed a two stage sparse optimization to minimize the reconstruction error of the target and select a sparse set of features to maximize the discriminative power. Tian *et al*. (2015) gathered the sparse coefficients of all patches in an object into a histogram based on their spatial distribution. The candidates are predicted for object verification during tracking by using particle filter methodology. Sparse coding is implemented to evaluate degree of changes of the appearance model and thus reduced the drifting problem.

Cheng *et al*. (2015) had conducted both generative and discriminative trackers under the particle filter framework. Common method implemented by Cheng *et al*. (2015), Jia *et al*. (2012), Liu *et al*. (2010, 2013), Mei and Ling (2009) and Tian *et al*. (2015) is utilizing sparse representation to represent the target and their research prove that sparse representation is more powerful tool to handle and analysis appearance representation during online tracking where it had overcame many challenging attribute such as heavy occlusions, illumination changes and pose variation. Li *et al*. (2016a) embedded "Online Reconstruction Error Prediction (OREP)" into the IVT (Ross *et al*., 2008) framework to predict appearance reconstruction error and proven that OREP greatly improved the performance of some video sequences as compared with Bao *et al*. (2012) and Ross *et al*. (2008).

Meanwhile, discriminative learning method required a classifier to be trained and updated online to differentiate the object from the background. It is also known as tracking-by-detection because it requires the user to manually identify the target in the first frame to generate a set of features of target. Then, another separate set of features is generated automatically to describe the background. Next, the target will be separated from the background in the subsequent frames. Similarly, it must be updated continuously to handle the appearance changes.

Support vector tracking (Avidan, 2004) proposed SVM to optimize the classification score by generating a Gaussian pyramid from every support vector, known as "Support Vector Pyramid" to account large motions in the image plane. The experiment shows that it performs better in long period of vehicles tracking. Babenko *et al*. (2009) proposed online MIL algorithm for object tracking and achieves promising performance with real-time tracking. Henriques *et al*. (2012) proposed Fourier analysis that capable for extremely fast learning and detection with the fast fourier transform. Closed-form solutions for training and detection with several types of kernels including the popular Gaussian and polynomial kernels are derived and the algorithm achieved competitive performance.

Yang *et al*. (2014) proposed superpixels in an appearance model that gives flexible and effective mid-level cues to distinguish the background and the foreground target. This model is more capable to handle the situations with big changes of pose and scale, shape deformation, occlusion and camera shake. Zhang and Song (2013) presented online Weighted Multiple Instance Learning (WMIL) to integrate the sample importance into the learning procedure, compute a new bag probability function that combines the weighted instance probability. Patras and Hancock (2010) presented a discriminative framework that coupled the predictor to a probabilistic classifier to predict the target accurately.

Yuan *et al*. (2014) proposed a robust superpixel-based tracker via depth fusion, developed sufficient structural information and high flexibility of mid-level features, depth-map's discriminative ability for the target and background separation, thus generated stronger discriminative ability. Fan *et al*. (2014) presented a supervised approach to learn and update a structured, sparse and discriminative representation that alternating between robust sparse coding and dictionary updating. Zhuang *et al*. (2014) presented Discriminative Sparse Similarity map (DSS map) to find the candidate with highest score in the evaluation model based upon a matrix and thus, obtain the best tracking results effectively.

Chen *et al*. (2016a) presented a robust Discriminative Local Collaborative (DLC). DLC encodes the candidates by an efficient local regularized least square solver with the l2 norm minimization by using the local image patches of both the target templates and the ones on the background cooperatively. Yang *et al*. (2016) applied Laplacian Regularized Least Squares (LapRLS) to learn a robust classifier for exploiting unlabelled data and preserving the local geometrical structure of the feature space adequately.

Qian and Xu (2016) presented Subclass Discriminant Constraint (SDC) for visual tracking. The SDC searches for a discriminative subspace to allow linear separation of image blocks that connected with the object and the background. Two dictionaries are constructed and learned in such subspaces for tracking and detection. A transformation matrix and sparse coefficient codes are being found out during dictionary learning. The similarity between the target candidate and the template is determined over sparse coefficients according to the histogram intersection.

Correlation filter: Correlation filter based tracking (Bolme *et al*., 2010) utilized filters trained on example images to model the appearance of objects. The object is initially selected by a tracking window that centred on the object in the first frame. By correlating the filter over a search window in next frame tracking and filter training collaborate to track the target. Next the new position of the target is indicated from the location respective to the maximum value in the correlation output. Based on this new location, appearance variation is updated online. Fourier domain Fast Fourier Transform (FFT) is applied to compute correlation to generate a fast tracker.

Zhang *et al*. (2016) presented spatial kernel phase correlation based tracker (SPC) that only implements phase correlation filter on adoption of the phase spectrum to estimate the object's translation. SPC achieves favourable tracking performance as it is more robust to noise and cluster. Liu *et al*. (2016b) presented a part-based representation tracker via kernelized correlation filter for visual tracking and Spatial-Temporal Angle Matrix (STAM) that used to select reliable patches from parts via multiple correlation filters to obtain stable patches effectively. Combination of this framework increases the diversity of affine matrices and related candidates.

Chen *et al*. (2016b) proposed a patch based tracker which adaptively integrates the kernel correlation filters with multiple effective features to handle occlusion challenges. The effective patches are selected by using an adaptive weight selection scheme and thus, improves the robustness of algorithm. Li *et al.* (2016a, b) presented a multi-view correlation tracker where multi-view model fuses various features and more discriminative features is selected. Fast training and efficient target locating provided by correlation filter framework had enhanced stability of scale variation tracking.

Others state of art method: Sevilla-Lara and Learned-Miller (2012) proposed Distribution Fields (DFs) to build an image that allows smoothing the objective function and the information about pixel values is keep intact. DFs descriptor has the advantage on slow changes in appearance and pose and minor occlusions. Zhang *et al.* (2012) proposed Compressive Tracking (CT) to preserve the structure of original image space based on non-adaptive random projections. By adopting a very sparse measurement matrix, features from the foreground and background targets are compressed efficiently. Generative and discriminative appearance models are combined in CT algorithm to encounter for scene variations.

Grabner *et al.* (2006) presented an on-line AdaBoost feature selection algorithm that has an advantages on its capability of on-line training, allowing the adaption of the classifier while tracking the object. Thereby appearance changes of the object such as out of plane rotations, illumination changes are handled effectively. Oron *et al.* (2015) proposed Locally Orderless Tracking (LOT) that will estimate the amount of local (dis) order in the target automatically, allows the tracker specific in both rigid and deformable objects on-line without prior assumptions.

Dinh *et al.* (2011) proposed auto exploration on the context information in two semantic terms which are distracters and supporters by using a sequential randomized forest an online template-based appearance model and local features. The tracker able to handle some challenges in tracking in uncontrolled environments with abrupt motion, occlusion, motion blur and frame-cut. Grabner *et al.* (2008) proposed a novel on-line semi-supervised boosting method to reduce drifting problem in tracking applications. The update process is formulated in a semi-supervised fashion as combined decision of a given prior and an on-line classier without adjusting any parameters.

Kwon and Lee (2010) proposed visual tracking decomposition scheme that efficiently highlights the object with drastic changes of motion and appearance or both. Zhuang *et al.* (2016) proposed a shallow and deep collaborative model that collaborates generative model to construct a local binary mask for handling occlusion tracking and a discriminative classifier to learn generic features. Cooperation between of these two models is more favourable to overcome occlusion and target appearance change.

Hu *et al.* (2016) proposed a Deep Metric Learning (DML) approach for under the particle filter framework that utilizes a feed-forward neural network architecture to classify the target object and background regions. A set of hierarchical nonlinear transformations in the feed-forward neural network is learned in order to project both the template and particles into the same feature space. The marginal between objects and backgrounds are maximized and thus, that objects are separated from the background regions easily.

MATERIALS AND METHODS

In this study, we will discuss on the basic concept of high speed tracking with Kernelized Correlation Filters (KCF) (Henriques *et al.* 2015). The parameters and idea of optimization technique used to optimize the performance of KCF in our research is presented with initial setting and procedures required. The research flow is presented in a flow chart to highlight the crucial procedures in our framework. The proposed method will be evaluated by its tacking performances which is OR and CLE where the calculation for both evaluator is discussed in study below.

Proposed method: The proposed method is basically a simple modification on the KCF. Instead of the default values, two parameters are selected to be varied in the optimization process which are p (padding) and s (spatial bandwidth). Padding is the extra area surrounding the target while spatial bandwidth is used to predict the response of an imaging system to very small objects which directly related to the size of the image and its object.

The range of values are defined at first in order to determine the global best values that are corresponding to the best result obtained. At first, the original setting for padding and spatial bandwidth is 1.5 and 0.1, respectively. By altering the values of padding from 1-2 with increment of 0.1 while 0.05-0.5 with increment of 0.05 for spatial bandwidth, the OR and CLE will be computed at each combination. Then, the global best tracking performance is obtained from all the combinations and recorded for all different sequences.

However, for some video sequences, the results obtained cannot achieved any improvement within the combination of these values. Thus, we increase the range of values to be optimized where padding is increased from

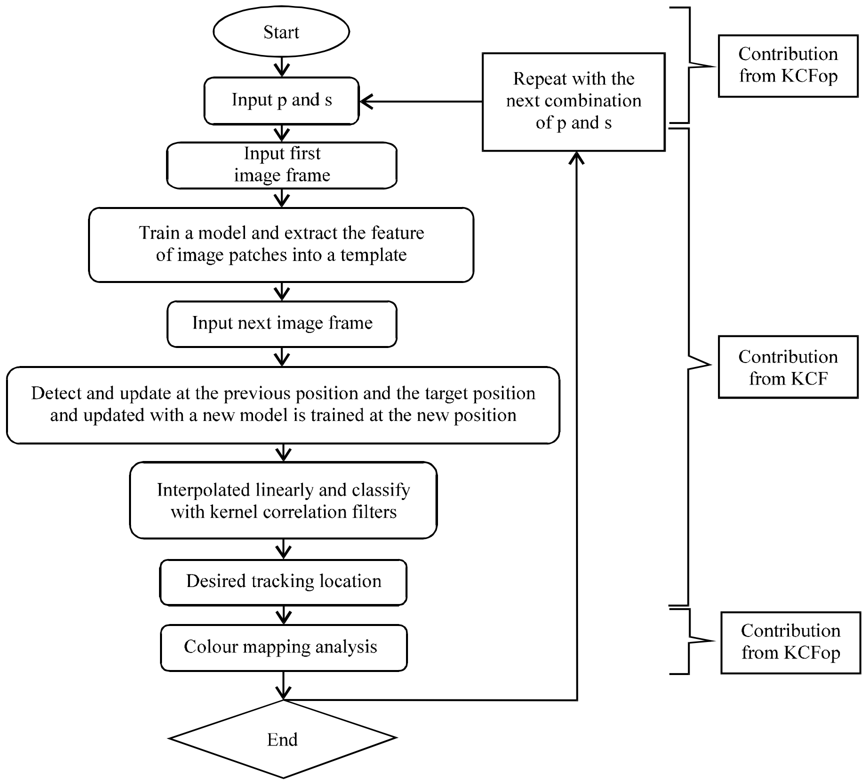

Fig. 1: The flow chart of the KCFop tracking algorithm

0.1-4.0 with increment of 0.1 while spatial bandwidth is increased from 0.01-0.4 with increment of 0.05. The reasons of choosing 0.1 and 0.05 as the increment value for padding and spatial bandwidth, respectively are because of some limitations which are time constraint and the value of the OR and CLE are not affected even when the number of decimal is increased further for the increment value. This ensure the time to compute the tracking results to be as fast as possible.

The algorithm is run by using MATLAB 2012a with the computer's specification of Intel (R) Core (TM) i7-3520M CPU @ 2.90 GHz. The global best tracking results are obtained when the highest OR and the lowest CLE achieved. The analysis will be presented using colour mapping technique where the colour will provide the information each performance. In addition, the global best results are selected and compared with the others tracking methods which the OR and CLE can be obtained from the website of visual tracking benchmark under the category of TB-100 sequences.

Flow chart: The tracking pipeline is presented in this study with a flow chart shown in Fig. 1 where the main procedures conducted to perform the parameters enhancement of the visual tracking is listed out here step by step. The performance evaluator or and CLE are used to measure the tracking performance of all video sequences because OR and CLE are the most used and common performance evaluators in visual tracking and thus provide an easier way to compare with other tracking method. The techniques to calculate OR and CLE will be discussed in the following study.

Initially the padding and spatial bandwidth is set to be 1.5 and 0.1. However, this default setting will limit the performance of the tracking. Thus, in our proposed method, padding and spatial bandwidth will be varied corresponding to frame size and the target size in the first frame for every different sequence. First, the algorithm of KCF is modified, so that, it can run through the different set of combinations of padding and spatial bandwidth. As the first set of combination is finished, the second combination is looped and proceeded the same as before and continue until the last combination. The tracking begins with the first sequence until the last sequences one by one.

Next, it will undergo several processes that are similar with the original KCF tracking algorithm where a model is trained with the image patch at the initial position of the target with the feature descriptor. Thus, a feature template is created to extract the feature of image patches. Then,

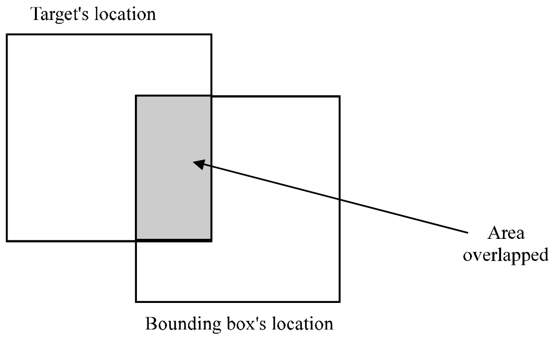

Fig. 2: Area overlapped between target's location and bounding box's location

the tracking process is started from the first frame until the last frame. The patch is detected at the previous position and the target position with maximum value of tracking performance is created and updated with a new model is trained at the new position. Based on the obtained value, it is interpolated linearly and classified with kernel correlation filters which is similar with the default algorithm by Henriques *et al*. (2015).

Performance evaluation: Both are qualitative evaluation and OR and CLE of our proposed method will be compared with other algorithms proposed in this visual tracking field. OR is defined as the percentage of the overlapping area between the region of ground truth and target as shown in Fig. 2. OR is calculated based on Eq. 1. Based on this concept or is calculated by using a MATLAB command which is "rectin" to calculate the intersecting area of these two boxes:

$$AOR = \frac{\text{area}\{ROI_t \cap ROI_g\}}{\text{area}\{ROI_t \cup ROI_g\}} \quad (1)$$

CLE is defined as the euclidean distance between the centre location of a target size and the ground truth as shown in Fig. 3 where it is measured in pixel as shown in Eq. 2:

$$CLE = \sqrt{(x_t - x_g)^2 + (y_t - y_g)^2} \quad (2)$$

This concept is basically the same with calculating the distance between 2 points using the x and y coordinate where in our case is the coordinates at the centre of the box. The higher the OR indicated that the target is tracked more accurately. OR score with more than 0.5 will only be considered as a successful tracking. While for the CLE, the lower the score it is, the better the tracking is.

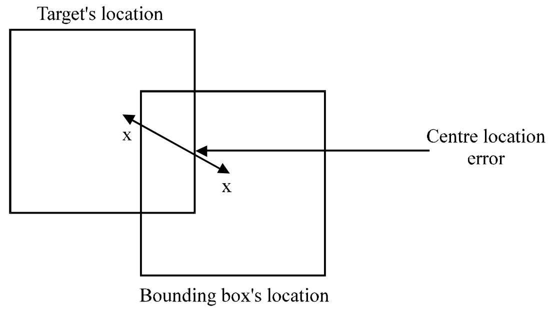

Fig. 3: CLE between target's location and bonding box's location

RESULTS AND DISCUSSION

The results obtained from the optimization and tracking performance of the proposed algorithm will be discussed in this study. A total of 86 sequences have been run through the experiment and the results of the global best performance corresponding with its own padding, p and spatial bandwidth, s are recorded. Meanwhile, the global best values of p and s obtained for each sequence are also discussed and tabulated with respect to their best tracking performance in term of OR and CLE. The results of the proposed algorithm are compared with other algorithms. In this study, 15 out of the 86 sequences results are illustrated in Fig. 4.

They are Basketball, Bolt2, Boy, Car4, Coke, Crowds, David3, Deer, DragonBaby, FaceOcc1, Freeman1, Human6, Ironman, MotorRolling and Skiing in ascending order. A bounding box with red outline is plotted on the desired target location and the target will be tracked continuously with the newly updated desired tracking location until the end. Only 5 out of 12 state-of-the-art methods are depicted to be compared in these sequences where, red, green, blue, yellow, white and black represents the ground truth (desired target location), KCFop, KCF, DFT, CT and CXT, respectively.

Optimization analysis: Once, the tracking performances of one sequence is obtained, a map data study where colour scale is applied in order to determine the best tracking results. Table 2 and 3 display the colour mapping of OR and CLE, respectively for the walking sequence. In this case, there are 3 colours used as the scale in this colour mapping analysis. The best value which is corresponding to the highest OR and the lowest CLE will be indicated with light green while the worst value which is corresponding to the lowest OR and highest CLE will be indicated with dark red. The mean value between the best and the worst value will be indicated with yellow.

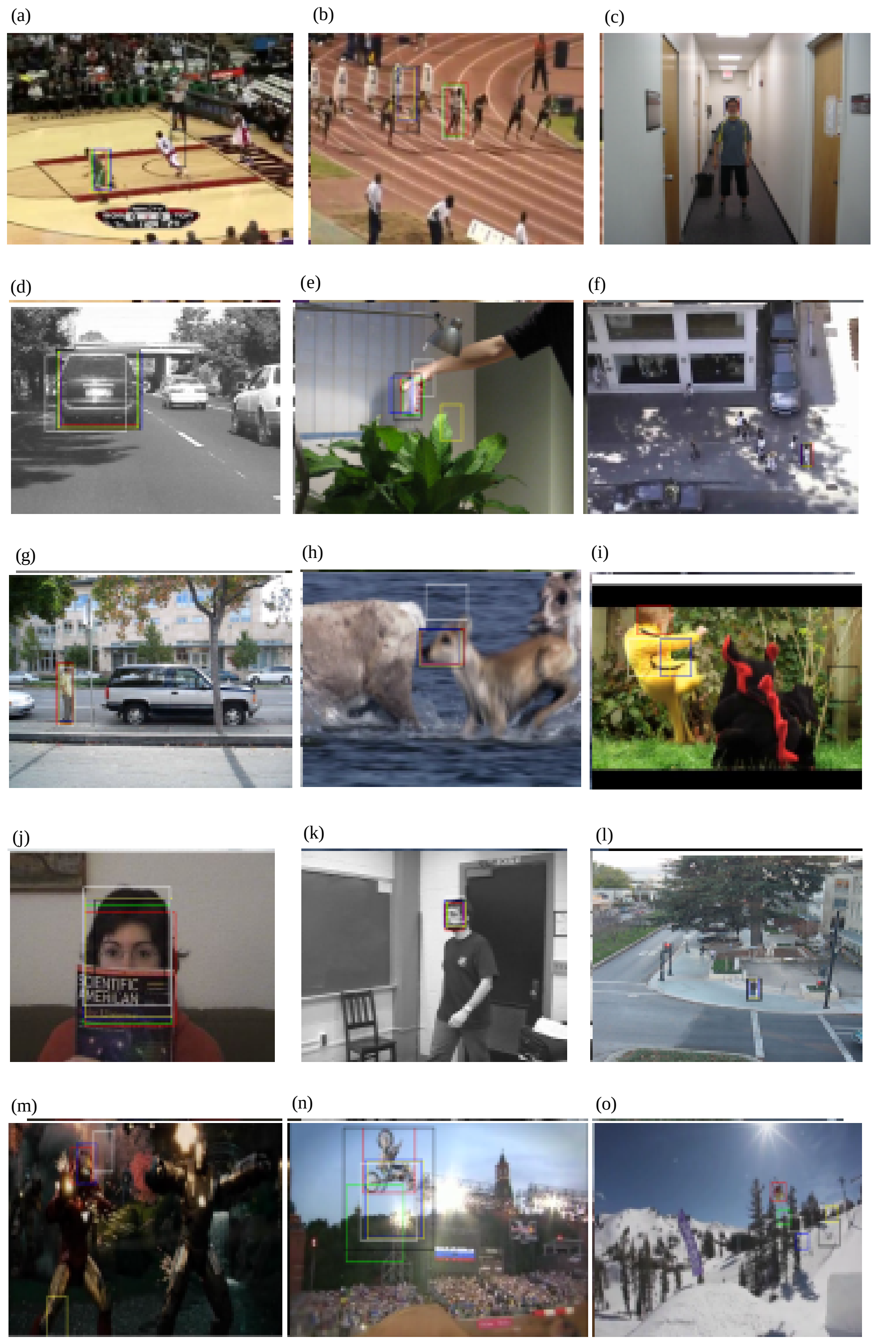

Fig. 4: a-o)Tracking performance comparisons

The density of the colour is varied through all according to the colour scale used. In walking sequence, the best tracking result of 0.486 OR and 4.280 CLE acquired at 1.0 padding and 0.05 spatial bandwidth. Thus, after obtaining these values of padding and spatial bandwidth, the process of analysis for the global best OR and CLE with respect with its own padding and spatial bandwidth for all 86 video sequences. Once, optimization complete, the proposed algorithm, KCFop, (i.e., Kenelized Correlation Filters Optimized) results are presented in Table 4.

Tracking result: The OR and CLE comparison with the KCFop and the other 12 tracking algorithms as tabulated

Table 2: Color map for deciding the padding and spatial bandwith for walking sequence in term of OR

84. Walking - OR										
p \ s	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
1.0	**0.486**	0.440	0.250	0.055	0.017	0.010	0.010	0.010	0.010	0.010
1.1	0.466	0.435	0.252	0.086	0.017	0.010	0.010	0.010	0.010	0.010
1.2	0.472	0.457	0.319	0.101	0.020	0.012	0.010	0.010	0.010	0.010
1.3	0.462	0.441	0.265	0.135	0.022	0.012	0.010	0.010	0.010	0.010
1.4	0.475	0.455	0.327	0.183	0.048	0.018	0.011	0.011	0.011	0.010
1.5	0.471	0.456	0.347	0.211	0.090	0.036	0.011	0.011	0.011	0.011
1.6	0.466	0.445	0.342	0.220	0.101	0.037	0.011	0.011	0.011	0.011
1.7	0.472	0.454	0.356	0.236	0.107	0.054	0.026	0.011	0.011	0.011
1.8	0.459	0.448	0.350	0.260	0.105	0.074	0.027	0.011	0.011	0.011
1.9	0.471	0.453	0.353	0.264	0.115	0.077	0.037	0.012	0.012	0.011
2.0	0.467	0.456	0.355	0.288	0.118	0.086	0.056	0.011	0.011	0.011

Table 3: Color map for deciding the padding and spatial bandwith for walking sequence in term of CLE

84. Walking - CLE										
p \ s	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
1.0	**4.28**	4.37	14.57	302.72	382.68	415.30	415.58	416.35	416.45	417.23
1.1	5.17	4.45	13.96	219.35	377.06	415.26	415.26	415.42	416.12	416.38
1.2	5.04	3.90	8.32	211.19	362.70	411.25	411.49	411.61	414.55	414.68
1.3	5.14	4.34	13.54	123.77	362.65	411.25	415.17	415.14	415.24	411.71
1.4	5.08	3.99	8.12	68.31	306.25	366.27	415.05	415.10	415.15	415.17
1.5	5.12	3.97	7.25	40.28	215.55	310.03	407.21	414.90	414.17	414.85
1.6	5.51	4.39	7.39	41.41	211.67	310.04	407.99	413.72	414.23	414.13
1.7	5.15	4.09	6.96	15.65	207.98	295.81	327.96	410.06	412.45	413.03
1.8	5.79	4.32	7.15	10.84	205.02	222.63	328.54	408.13	411.91	412.21
1.9	5.16	4.11	7.04	10.87	204.76	219.01	309.96	403.37	407.82	409.08
2.0	5.51	4.05	7.03	9.58	204.70	212.10	280.56	393.86	407.98	408.92

Table 4: All 86 video sequences tracking performance of KCFop

No.	Sequence	p	s	OR	CLE	No.	Sequence	p	s	OR	CLE
1	Basketball	1.8	0.08	0.761	6.1390	44	Human4	1.7	0.07	0.428	64.9430
2	Biker	1.5	0.10	0.243	77.177	45	Human5	2.0	0.04	0.330	6.35900
3	Bird1	1.8	0.10	0.257	73.363	46	Human6	1.2	0.05	0.210	94.6240
4	Bird2	0.9	0.05	0.782	8.2460	47	Human7	1.6	0.08	0.416	5.89700
5	BlurBody	1.8	0.05	0.696	6.9420	48	Human8	1.0	0.15	0.492	3.02000
6	BlurCar3	2.4	0.03	0.814	3.3340	49	Human9	1.3	0.10	0.316	11.4320
7	BlurFace	1.4	0.04	0.758	7.2230	50	Ironman	1.5	0.10	0.145	194.943
8	Board	1.2	0.10	0.715	14.377	51	Jogging	1.6	0.15	0.710	4.14200
9	Bolt	1.4	0.10	0.682	6.3250	52	Jump	1.9	0.15	0.114	47.2740
10	Bolt2	2.1	0.25	0.358	41.030	53	Jumping	2.6	0.15	0.691	3.49000
11	Boy	1.4	0.06	0.763	2.4440	54	KiteSurf	1.3	0.15	0.509	14.4270
12	Car1	0.9	0.25	0.121	39.410	55	Lemming	1.5	0.10	0.397	77.8680
13	Car2	1.4	0.03	0.685	3.8760	56	Liquor	1.5	0.10	0.843	5.26900
14	Car4	1.4	0.14	0.491	9.6850	57	Man	1.4	0.06	0.832	2.24300
15	CarDark	0.4	0.05	0.777	2.3380	58	Matrix	1.8	0.25	0.282	52.2740
16	CarScale	0.6	0.06	0.396	15.427	59	Mhyang	1.0	0.15	0.800	2.93600

Table 4: Continue

No.	Sequence	p	s	OR	CLE	No.	Sequence	p	s	OR	CLE
17	ClifBar	1.8	0.20	0.456	10.284	60	MotorRolling	0.1	0.40	0.203	102.511
18	Coke	1.7	0.20	0.703	9.8050	61	MountainBike	1.4	0.15	0.663	6.57200
19	Couple	3.8	0.07	0.631	3.3860	62	Panda	1.2	0.07	0.179	41.0710
20	Coupon	1.2	0.05	0.946	1.5320	63	RedTeam	1.2	0.10	0.478	3.52900
21	Crossing	1.4	0.07	0.673	2.1070	64	Shaking	2.1	0.03	0.714	6.75700
22	Crowds	1.5	0.10	0.794	3.0650	65	Singer1	0.7	0.15	0.346	12.63500
23	Dancer2	1.6	0.01	0.742	5.9930	66	Singer2	1.2	0.01	0.736	8.58500
24	David	1.1	0.35	0.464	7.4480	67	Skater	1.2	0.15	0.572	10.6670
25	David2	1.7	0.11	0.831	1.9800	68	Skater2	2.1	0.15	0.598	13.8200
26	David3	1.2	0.09	0.749	4.1180	69	Skating1	2.4	0.15	0.487	7.44800
27	Deer	1.3	0.05	0.660	8.1010	70	Skating2	1.7	0.11	0.419	23.3740
28	Diving	1.3	0.12	0.330	23.051	71	Skiing	1.4	0.30	0.093	180.515
29	Dog1	1.7	0.15	0.528	3.4340	72	Soccer	1.7	0.30	0.395	13.5160
30	Doll	1.7	0.05	0.565	6.8320	73	Subway	1.7	0.15	0.783	2.46100
31	DragonBaby	2.1	0.30	0.426	39.201	74	Surfer	1.8	0.20	0.439	3.96400
32	Dudek	1.5	0.20	0.715	10.922	75	Suv	1.5	0.13	0.879	3.42500
33	FaceOcc1	2.3	0.03	0.789	13.406	76	Sylvester	1.8	0.15	0.623	12.5490
34	FaceOcc2	1.8	0.20	0.760	6.6390	77	Tiger1	2.0	0.10	0.793	7.65400
35	Fish	1.4	0.02	0.814	3.8670	78	Tiger2	2.2	0.04	0.707	10.9740
36	FleetFace	2.5	0.30	0.615	17.146	79	Toy	1.4	0.05	0.461	7.65300
37	Football	1.7	0.40	0.611	6.6710	80	Trans	1.2	0.05	0.523	27.2720
38	Football1	2.5	0.10	0.824	2.1570	81	Trellis	1.9	0.30	0.595	7.37900
39	Freeman1	1.3	0.25	0.408	7.9080	82	Twinnings	1.4	0.10	0.590	4.41400
40	Freeman3	1.6	0.10	0.317	19.254	83	Vase	1.3	0.09	0.279	10.5670
41	Freeman4	1.6	0.10	0.391	4.9940	84	Walking	1.2	0.07	0.486	4.28000
42	Girl	1.2	0.15	0.579	8.4330	85	Walking2	0.6	0.15	0.374	8.52300
43	Gym	1.3	0.10	0.448	11.844	86	Woman	1.7	0.10	0.662	9.66300

in Table A1 and A2, respectively for all 86 video sequences. The colour of the words represents the ranking of the performance where green, blue and red colour indicate the best (highest OR), second and third place. Some of the sequences achieved significant improvement on OR and CLE but there are 6 video sequences which unable to be improved such as Biker, Crowds, Freeman3, Ironman, Lemming and Liquor.

KCFop has the best tracking performance in OR evaluation for 27 sequences, 15 sequences obtained the second highest OR and 13 sequences obtained the third highest OR. In addition, 39 sequences obtained the lowest CLE, 20 sequences obtained the second lowest CLE and 13 sequences obtained the third lowest CLE. In general, the proposed KCFop method obtained the most outstanding results in term of both average OR and CLE of all 86 video sequences which are 0.554 and 19.803, respectively. The average OR and CLE of all 86 vedio sequences for all 13 methods are also, tabulated in Table 4 and 5.

The improvement on average OR is 18.30%, from 0.468 increase to 0.554, there is significant improvement on average CLE, from 38.789 pixels reduce to 19.803 pixels which is reduced by 48.95%. The result of tracking performance shows that KCFop has achieved better tracking performance followed by KCF and struck with both having 0.468 and then by CXT with 0.422. KCFop has surpassed semiB the most by 125.41%, followed by CT with 92.54% and lastly Frag with 67.35%.

Table 5: Comparison of average OR and CLE between KCFop and the other methods

	OR		CLE	
Methods	Average	Different in (%)	Average	Different in (%)
KCFop	0.554	-	19.8030	-
KCF	0.468	18.30	38.7890	48.95
Struck	0.468	18.48	44.2110	55.21
DFT	0.343	61.45	78.6000	74.81
CSK	0.399	38.70	91.5060	78.36
MIL	0.350	58.51	65.2640	69.66
Frag	0.331	67.35	78.8090	74.87
CT	0.288	92.54	77.4310	74.42
OAB	0.357	55.16	71.9730	72.49
LOT	0.335	65.20	63.3740	68.75
CXT	0.422	31.26	66.8110	70.36
semiB	0.246	125.41	175.495	88.72
VTD	0.387	43.20	53.6630	63.10

KCFop also, acquired the lowest average CLE among the other methods, followed by KCF with 38.789 pixels and later by struck with 44.211 pixels. Furthermore, based on the CLE percentage difference, KCFop has also, outperformed semiB by 88.72%, followed by CSK with 78.36% and lastly Frag with 74.87%. Thus, KCFop had achieved a better improved and able to obtain a promising tracking performance in our proposed method (Appendex).

CONCLUSION

The proposed method, KCFop has promising tracking performance among the 12 tracking methods as compared in here. There are still having many challenging issues due to the destabilizing attributes that lead to the performance drift. Overall, the proposed concept of simple optimization that is implemented into the tracking algorithm by enhancing the global best padding and spatial bandwidth obtained. Through this parameters optimization, the average OR of the proposed method is increased to 0.554 with 18.30% of improvement and the average CLE of the proposed method is decreased to 19.803 pixels with 48.95% of improvement as compared to the KCF when tested on 86 video sequences. Thus, KCFop is commended to enhance visual tracking.

ACKNOWLEDGEMENT

This research and publication were supported by Universiti Sains Malaysia Grant No. 1001/PAERO/814276.

NOMENCLATURE

λ = Regularization parameter
X^H = Hermitian transpose
X^* = Complex-conjugate of X
I = Identity matrix
F = DFT matrix
K = Kernel matrix
α = Vector of coefficient α_i
k^{XZ} = Kernel correlation
σ = Kernel sigma
p = Padding
s = Spatial bandwidth
w_1, w_2 = Frame width and target width
h_1, h_2 = Frame height and target height
x_t, y_t = Centre location of the target
x_g, y_g = Ground truth location of the target
ROI_t = Area of Target's Region
ROI_g = Area of ground Truth's Region

Appendix:

Table A1a: Comparison of OR between KCFop with other algorithms

		KCFop	KCF	Struck	DFT	CSK	MIL	Frag	CT	OAB	LOT	CXT	semiB	VTD
1	Basketball	**0.761**	0.676	0.203	0.607	**0.706**	0.220	0.619	0.256	0.029	0.681	0.023	0.022	**0.727**
2	Biker	0.243	0.243	0.266	0.251	0.257	0.259	0.226	0.013	0.260	**0.285**	**0.409**	0.234	**0.288**
3	Bird1	**0.257**	0.054	0.092	0.186	0.018	**0.291**	0.042	**0.363**	0.055	0.035	0.013	0.045	0.023
4	Bird2	**0.782**	0.575	0.567	**0.598**	0.578	0.557	0.435	0.096	**0.648**	0.084	0.239	0.363	0.183
5	BlurBody	0.696	0.435	**0.724**	0.096	0.443	0.039	0.543	0.030	**0.713**	0.263	**0.729**	0.101	0.236
6	BlurCar3	**0.814**	**0.788**	**0.773**	0.121	0.452	0.273	0.458	0.219	0.730	0.240	0.601	0.726	0.187
7	BlurFace	**0.758**	0.737	0.469	0.328	**0.778**	0.275	0.642	0.151	0.481	0.198	**0.833**	0.469	0.416
8	Board	**0.715**	**0.639**	**0.652**	0.325	0.444	0.407	0.512	0.447	0.265	0.209	0.290	0.200	0.266
9	Bolt	**0.682**	**0.679**	0.014	0.034	0.019	0.011	0.113	0.009	0.042	**0.521**	0.016	0.059	0.366
10	Bolt2	0.358	0.011	0.222	0.010	0.481	**0.683**	0.326	0.367	0.011	**0.514**	0.012	0.031	**0.502**
11	Boy	**0.763**	0.750	**0.760**	0.401	0.657	0.491	0.389	0.590	**0.791**	0.533	0.542	0.323	0.626
12	Car1	**0.121**	0.100	0.111	0.096	0.098	0.093	0.082	0.066	0.083	**0.141**	**0.397**	0.056	0.118
13	Car2	0.685	0.658	0.688	0.157	**0.690**	0.166	0.259	0.234	0.569	0.104	**0.875**	0.493	**0.801**
14	Car4	**0.491**	**0.480**	**0.489**	0.237	0.466	0.258	0.188	0.213	0.213	0.042	0.307	0.225	0.360
15	CarDark	0.777	0.599	**0.892**	0.380	0.755	0.195	0.301	0.003	**0.785**	0.421	0.566	**0.832**	0.541
16	CarScale	0.396	0.380	0.412	0.414	0.415	0.410	0.425	**0.431**	0.395	0.346	**0.684**	0.323	**0.430**
17	ClifBar	0.456	0.244	0.478	0.199	**0.495**	0.489	0.174	0.447	**0.548**	0.247	**0.520**	0.163	0.435
18	Coke	**0.703**	0.549	**0.673**	0.109	**0.570**	0.202	0.039	0.226	0.330	0.122	0.423	0.046	0.142
19	Couple	**0.631**	0.190	**0.536**	0.077	0.075	0.498	**0.569**	0.469	0.358	0.448	0.483	0.342	0.064
20	Coupon	**0.946**	**0.944**	0.876	**0.915**	0.898	0.623	0.251	0.574	0.505	0.218	0.833	0.148	0.645
21	Crossing	0.673	0.664	0.677	0.496	0.479	**0.727**	0.311	**0.683**	0.664	0.303	0.361	**0.684**	0.317
22	Crowds	**0.794**	**0.794**	0.613	**0.793**	**0.761**	0.055	0.023	0.004	0.074	0.01	0.093	0.071	0.025
23	Dancer2	0.742	0.716	**0.762**	**0.792**	**0.788**	0.734	0.741	0.736	0.752	0.691	0.724	0.540	0.722
24	David	0.464	0.431	0.239	0.297	0.402	0.428	0.169	**0.496**	0.387	0.264	**0.649**	0.243	**0.558**
25	David2	**0.831**	0.827	**0.871**	0.541	0.820	0.451	0.240	0.003	0.318	0.602	**0.883**	0.536	0.686
26	David3	**0.749**	**0.740**	0.292	0.559	0.498	0.537	**0.670**	0.306	0.326	0.665	0.120	0.149	0.403
27	Deer	0.660	0.595	**0.740**	0.256	**0.745**	0.124	0.173	0.040	**0.721**	0.198	0.697	0.639	0.058
28	Diving	**0.330**	**0.264**	**0.334**	0.215	0.231	0.233	0.163	0.017	0.226	0.181	0.183	0.136	0.204

Table A1b: Comparison of OR between KCFop with other algorithms

29	Dog1	0.528	0.523	0.546	0.435	0.548	0.532	0.542	0.535	0.544	**0.823**	**0.798**	0.442	**0.594**
30	Doll	0.565	0.550	0.545	0.295	0.316	0.466	0.534	0.453	0.536	**0.666**	**0.746**	0.122	**0.648**
31	DragonBaby	**0.426**	0.315	0.201	0.141	0.203	0.254	**0.398**	0.268	0.190	**0.528**	0.350	0.232	0.324
32	Dudek	0.715	0.698	**0.730**	0.690	0.716	0.707	0.535	0.647	0.657	0.536	**0.731**	0.382	**0.799**
33	FaceOcc1	**0.789**	0.772	0.729	0.687	**0.795**	0.596	**0.816**	0.637	0.660	0.409	0.634	0.569	0.684
34	FaceOcc2	0.760	0.722	**0.785**	**0.768**	**0.780**	0.672	0.648	0.608	0.598	0.458	0.742	0.475	0.736
35	Fish	**0.814**	**0.801**	**0.859**	0.757	0.208	0.450	0.545	0.716	0.037	0.229	0.783	0.210	0.556
36	FleetFace	**0.615**	0.546	**0.608**	0.482	0.587	0.491	0.465	0.521	0.525	0.569	0.564	0.259	**0.629**
37	Football	0.611	0.502	0.534	**0.658**	0.553	0.589	**0.701**	0.610	0.335	**0.655**	0.542	0.153	0.564
38	Football1	**0.824**	0.676	0.668	**0.867**	0.465	0.660	0.364	0.227	0.271	0.536	**0.763**	0.126	0.570
39	Freeman1	**0.408**	0.191	0.338	**0.386**	0.235	0.341	**0.374**	0.144	0.354	0.196	0.339	0.133	0.348
40	Freeman3	**0.317**	**0.317**	0.255	**0.314**	0.301	0.010	**0.317**	0.002	0.258	0.118	**0.710**	0.146	0.300
41	Freeman4	**0.391**	**0.173**	0.170	0.169	0.131	0.054	0.141	0.005	0.109	0.159	0.166	**0.193**	0.156
42	Girl	**0.579**	0.480	**0.745**	0.283	0.367	0.398	0.454	0.306	**0.721**	0.418	0.554	0.368	0.551
43	Gym	0.448	0.353	0.428	0.111	0.336	**0.463**	**0.482**	0.279	0.036	0.277	**0.452**	0.048	0.410
44	Human4	**0.428**	**0.354**	0.142	0.135	0.138	0.122	0.050	0.128	0.135	0.143	0.065	0.078	**0.246**
45	Human5	**0.330**	0.191	**0.350**	0.063	0.199	0.213	0.030	0.178	0.218	**0.361**	0.237	0.210	0.289
46	Human6	0.210	0.203	0.214	0.217	0.210	**0.242**	**0.222**	0.219	0.204	**0.363**	0.145	0.189	0.182
47	Human7	0.416	0.256	**0.484**	0.178	0.341	**0.484**	0.275	0.188	**0.417**	0.346	**0.429**	0.208	0.283
48	Human8	**0.492**	**0.457**	0.132	0.119	0.184	0.123	0.097	0.035	0.097	**0.695**	0.114	0.021	0.288
49	Human9	**0.316**	0.301	0.101	0.132	0.248	**0.356**	**0.334**	0.263	0.201	0.127	0.080	0.154	0.212
50	Ironman	**0.145**	**0.145**	0.087	0.054	**0.118**	0.070	0.027	0.071	0.028	0.088	0.045	0.037	**0.203**
51	Jogging	**0.710**	0.181	0.171	0.187	0.180	0.181	0.518	0.178	**0.600**	0.092	**0.766**	0.530	0.154
52	Jump	**0.114**	0.064	0.110	0.070	0.092	0.054	**0.175**	0.100	0.083	**0.195**	0.060	0.040	0.060
53	Jumping	**0.691**	0.279	**0.617**	0.110	0.050	0.521	**0.672**	0.043	0.069	0.578	0.524	0.058	0.123
54	KiteSurf	**0.509**	**0.442**	**0.639**	0.389	0.260	0.312	0.126	0.267	0.314	0.029	0.323	0.296	0.129
55	Lemming	0.397	0.397	0.481	0.405	0.331	**0.649**	0.305	**0.549**	**0.594**	**0.594**	0.455	0.115	0.434
56	Liquor	**0.843**	**0.843**	0.412	0.222	0.252	0.216	0.330	0.203	0.449	**0.828**	0.257	0.396	**0.496**
57	Man	0.832	0.831	**0.898**	0.202	**0.874**	0.194	0.201	0.227	0.859	0.185	0.850	**0.884**	0.303
58	Matrix	**0.282**	0.119	0.097	0.058	0.031	0.122	0.057	0.019	**0.266**	0.111	0.058	**0.135**	0.121
59	MHyang	**0.800**	0.772	**0.818**	0.709	0.794	0.507	0.651	0.600	0.748	0.215	**0.850**	0.664	0.731

Table A1c: Comparison of OR between KCFop with other algorithms

60	MotorRolling	**0.203**	0.093	**0.154**	0.084	0.090	0.122	0.118	0.103	0.110	0.121	**0.129**	0.031	0.114
61	MountainBike	0.663	0.627	**0.709**	0.298	**0.712**	0.452	0.128	0.143	0.631	0.578	0.226	0.229	**0.698**
62	Panda	0.179	0.150	**0.516**	0.279	0.123	**0.542**	0.329	**0.418**	0.152	0.041	0.185	0.237	0.276
63	RedTeam	0.478	0.467	**0.497**	0.272	**0.508**	0.399	0.138	0.474	**0.512**	0.100	0.380	0.388	0.204
64	Shaking	**0.714**	0.039	0.348	**0.637**	0.575	0.427	0.082	0.101	0.013	0.125	0.119	0.007	**0.711**
65	Singer1	0.346	0.331	0.358	0.355	**0.359**	0.356	0.205	0.348	0.336	0.194	**0.492**	0.117	**0.488**
66	Singer2	**0.736**	**0.708**	0.041	**0.626**	0.043	0.514	0.196	0.083	0.045	0.256	0.066	0.030	0.417
67	Skater	0.572	0.562	**0.632**	0.359	0.548	**0.613**	0.509	0.380	0.532	0.445	**0.623**	0.359	0.541
68	Skater2	**0.598**	0.580	**0.589**	0.068	**0.592**	0.470	0.545	0.145	0.532	0.554	0.403	0.128	0.533
69	Skating1	**0.487**	0.456	0.308	0.135	**0.497**	0.127	0.125	0.091	0.394	0.258	0.140	0.084	**0.525**
70	Skating2	**0.419**	0.341	0.244	0.149	0.076	0.175	0.205	0.214	0.240	**0.414**	0.129	0.055	**0.440**
71	Skiing	**0.093**	0.047	0.034	0.051	0.059	0.055	0.033	**0.068**	**0.078**	0.021	**0.093**	0.043	0.066
72	Soccer	**0.395**	**0.362**	0.185	0.176	0.114	0.171	0.174	0.169	0.086	0.203	0.126	0.066	**0.325**
73	Subway	**0.783**	**0.747**	0.654	**0.725**	0.193	0.648	0.462	0.575	0.164	0.562	0.175	0.286	0.157
74	Surfer	**0.439**	**0.380**	0.368	0.042	0.005	0.259	0.221	0.088	0.039	0.322	**0.730**	0.103	0.318
75	Suv	**0.879**	**0.878**	0.511	0.072	0.524	0.194	0.628	0.235	0.628	0.654	**0.736**	0.378	0.453
76	Sylvester	0.623	0.607	**0.723**	0.377	**0.631**	0.529	0.584	**0.666**	0.560	0.565	0.600	0.341	0.619
77	Tiger1	**0.793**	**0.782**	0.153	**0.532**	0.261	0.120	0.265	0.410	0.109	0.139	0.322	0.400	0.118
78	Tiger2	**0.707**	0.351	**0.540**	**0.677**	0.170	0.455	0.118	0.442	0.151	0.135	0.339	0.153	0.303
79	Toy	0.461	0.426	**0.508**	0.431	0.359	0.239	0.380	0.362	0.297	0.299	**0.730**	0.143	**0.527**
80	Trans	**0.523**	0.389	0.514	0.488	**0.536**	0.34	0.519	0.511	0.503	0.461	0.510	0.137	**0.529**
81	Trellis	**0.595**	0.522	**0.615**	0.361	0.480	0.248	0.285	0.336	0.140	0.306	**0.656**	0.198	0.453
82	Twinnings	**0.590**	0.550	**0.579**	0.485	0.522	0.512	0.471	0.444	0.420	0.504	0.436	0.272	**0.650**
83	Vase	0.279	0.277	0.307	**0.329**	0.328	0.327	0.327	0.311	0.262	**0.366**	**0.562**	0.117	0.282
84	Walking	0.486	0.456	**0.571**	0.562	0.537	0.545	0.537	0.520	0.543	**0.704**	0.171	0.194	**0.607**
85	Walking2	0.374	0.279	**0.510**	0.405	**0.456**	0.288	0.274	0.266	0.363	0.335	0.369	**0.463**	0.326
86	Woman	**0.662**	0.658	**0.732**	**0.756**	0.196	0.157	0.147	0.131	0.479	0.089	0.199	0.109	0.144
	Average	**0.554**	**0.468**	**0.468**	**0.343**	**0.399**	**0.350**	**0.331**	**0.288**	**0.357**	**0.335**	**0.422**	**0.246**	**0.387**

Table A2a: Comparison of CLE between KCFop with other algorithms

		KCFop	KCF	Struck	DFT	CSK	MIL	Frag	CT	OAB	LOT	CXT	semiB	VTD
1	Basketball	**6.139**	7.889	118.26	18.028	**6.527**	91.918	13.020	89.106	205.477	6.678	214.569	327.981	**5.617**
2	Biker	77.177	77.177	**28.507**	81.680	79.321	**27.240**	93.456	41.900	148.642	74.261	**22.938**	222.002	78.690
3	Bird1	**73.363**	152.340	155.371	131.371	777.302	**41.765**	325.305	**23.536**	179.635	122.513	148.250	426.400	117.450
4	Bird2	**8.246**	21.370	19.747	47.779	**18.297**	18.311	28.109	110.540	**15.635**	108.688	42.898	161.342	67.468
5	BlurBody	**6.942**	64.115	**13.973**	259.156	60.666	206.746	37.794	176.548	15.454	101.450	**9.9590**	354.142	146.900
6	BlurCar3	**3.334**	**4.138**	6.523	154.691	166.813	138.155	58.218	68.566	15.110	120.645	**5.154**	44.29	107.796
7	BlurFace	**7.223**	**8.364**	42.353	75.636	9.681	71.976	23.618	119.952	54.388	156.877	**6.442**	189.599	40.689
8	Board	**14.377**	**36.137**	**34.002**	99.862	65.934	91.819	91.842	63.061	113.606	182.389	126.425	350.475	125.599
9	Bolt	**6.325**	**6.365**	398.839	367.276	429.395	393.543	183.383	363.799	256.281	**12.512**	385.488	363.179	**25.158**
10	Bolt2	41.030	329.808	86.408	276.700	**11.634**	**7.392**	43.208	115.196	136.575	**9.399**	266.152	277.067	17.115
11	Boy	**2.444**	**2.867**	3.845	106.310	20.145	12.835	40.516	9.031	**2.917**	66.002	7.392	211.792	7.574
12	Car1	**39.410**	**42.435**	51.729	91.455	570.435	112.169	115.714	118.741	95.249	114.463	**20.748**	134.812	110.068
13	Car2	3.876	3.968	**2.418**	87.688	**2.531**	55.809	89.399	42.643	30.787	104.339	**2.883**	44.926	3.945
14	Car4	**9.685**	**9.877**	**8.690**	61.944	19.134	50.777	131.549	86.027	95.326	167.285	58.118	152.262	36.995
15	CarDark	**2.338**	6.047	**0.954**	58.846	3.233	43.478	36.466	119.219	2.838	30.853	16.490	**2.084**	16.479
16	CarScale	**15.427**	**16.142**	36.431	75.752	83.015	33.471	**19.743**	25.953	30.739	101.224	24.519	132.049	38.461
17	ClifBar	**10.284**	36.721	23.507	53.984	**10.877**	11.367	39.653	12.520	**7.820**	33.526	19.706	108.729	24.669
18	Coke	**9.805**	18.653	**12.085**	70.698	**13.644**	46.724	124.808	40.491	35.856	69.420	25.740	365.276	68.653
19	Couple	**3.386**	47.556	**11.333**	108.596	144.577	34.526	**8.801**	36.385	57.621	37.092	41.758	64.738	104.252
20	Coupon	**1.532**	**1.568**	4.413	**2.518**	3.240	18.666	71.569	18.98	54.615	27.684	4.693	118.818	10.657
21	Crossing	**2.107**	**2.250**	**2.808**	22.280	8.959	3.177	38.589	3.560	4.540	36.713	23.414	3.645	26.125
22	Crowds	**3.065**	**3.065**	7.328	**3.331**	**3.691**	428.345	371.335	379.588	248.458	394.946	277.540	349.954	447.363
23	Dancer2	**5.993**	**6.412**	8.911	**5.511**	6.857	9.732	11.235	8.597	10.204	15.643	7.442	74.124	7.870
24	David	**7.448**	**8.062**	42.801	42.876	17.690	16.860	82.071	10.494	21.733	23.854	**6.048**	47.092	11.386
25	David2	**1.980**	2.083	**1.499**	17.288	2.329	10.927	56.868	76.704	33.807	4.092	**1.321**	11.849	2.856
26	David3	**4.118**	**4.302**	106.501	50.931	56.102	29.681	13.552	88.664	83.435	**9.842**	222.212	236.465	66.722
27	Deer	8.101	21.165	**5.270**	98.749	**4.973**	100.729	105.088	246.423	6.989	65.223	**6.749**	61.531	134.849
28	Diving	**23.051**	42.571	**19.222**	50.572	81.773	62.677	84.745	98.051	55.586	**37.272**	64.275	177.577	73.819
29	Dog1	**3.434**	**4.234**	5.663	41.237	**3.811**	7.832	11.921	6.986	5.897	4.635	4.886	75.419	11.038
30	Doll	**6.832**	8.359	8.919	59.547	44.721	16.677	13.738	21.824	12.380	**6.322**	**4.654**	113.941	7.294

Table A2b: Comparison of CLE between KCFop with other algorithms

31	DragonBaby	**39.201**	50.399	69.255	75.583	87.909	43.611	46.387	59.040	80.062	**26.821**	**40.737**	85.482	41.426
32	Dudek	**10.922**	**11.382**	11.449	18.719	13.394	17.697	82.693	26.534	31.381	85.126	12.818	218.485	**10.296**
33	FaceOcc1	**13.406**	15.983	18.778	23.588	**11.932**	29.858	**10.972**	25.817	24.658	34.748	25.345	74.495	20.203
34	FaceOcc2	6.639	7.666	**5.955**	7.875	**5.923**	13.599	15.952	18.948	19.582	14.891	**6.269**	50.763	8.286
35	Fish	**3.867**	**4.077**	**3.401**	8.838	41.189	24.141	21.560	10.684	87.030	33.619	6.252	62.852	16.793
36	FleetFace	**17.146**	26.370	**23.005**	68.016	**25.601**	63.119	67.659	58.435	52.097	33.741	45.052	262.377	46.134
37	Football	**6.671**	14.602	17.312	9.293	16.187	12.087	**5.357**	11.910	72.371	**6.553**	12.833	244.361	13.640
38	Football1	**2.157**	5.474	5.465	**1.975**	16.51	5.618	15.697	20.712	85.677	6.850	**2.611**	162.035	7.437
39	Freeman1	**7.908**	94.883	14.275	10.432	125.464	11.216	**10.097**	118.716	35.722	86.911	26.847	174.354	**10.317**
40	Freeman3	**19.254**	**19.254**	**16.834**	32.602	53.903	87.565	40.467	65.319	40.661	40.527	**3.593**	152.661	23.957
41	Freeman4	**4.994**	**27.116**	48.696	57.478	78.870	62.079	72.268	132.586	135.931	**38.629**	65.642	75.983	61.685
42	Girl	**8.433**	11.916	**2.573**	23.981	19.342	13.666	20.670	18.852	**3.696**	22.783	10.965	40.752	8.598
43	Gym	11.844	16.258	18.463	104.466	27.073	11.798	**10.039**	43.743	110.962	**9.171**	15.763	230.366	**10.842**
44	Human4	**64.943**	**131.776**	253.189	245.913	218.251	294.103	310.205	295.679	174.139	182.591	327.373	306.524	**162.145**
45	Human5	**6.359**	175.496	**6.873**	259.063	285.489	221.522	286.204	240.823	189.130	90.302	200.861	402.705	**17.244**
46	Human6	94.624	107.652	87.038	131.106	179.052	**35.374**	84.193	**60.175**	159.887	**35.866**	96.366	341.235	125.894
47	Human7	**5.897**	48.198	**4.557**	48.021	18.187	**4.047**	44.123	31.971	14.365	57.329	7.979	132.994	26.880
48	Human8	**3.020**	**3.843**	63.786	73.795	50.017	74.947	74.834	92.137	74.153	**4.005**	67.324	224.014	18.999
49	Human9	**11.432**	**14.769**	42.454	56.440	44.698	23.397	**21.637**	29.670	24.179	42.473	48.394	103.631	33.453
50	Ironman	194.943	194.943	**127.670**	239.665	185.277	193.374	270.874	180.342	213.801	**98.711**	162.672	175.996	**63.242**
51	Jogging	**4.142**	88.267	62.055	31.443	134.982	96.345	21.457	92.494	**7.164**	90.876	**5.590**	51.985	83.290
52	Jump	**47.274**	84.114	84.828	92.602	88.790	135.725	**78.120**	91.835	110.600	**41.289**	132.196	227.149	146.933
53	Jumping	**3.490**	26.117	6.547	67.082	85.972	9.992	**5.602**	47.731	46.345	**5.575**	9.991	94.258	41.387
54	KiteSurf	**14.427**	**17.269**	**6.135**	32.397	36.474	22.601	141.139	90.083	64.598	84.386	36.773	32.856	76.258
55	Lemming	77.868	77.868	37.753	77.754	114.233	**12.065**	126.877	32.252	**18.051**	**19.116**	61.393	286.245	79.224
56	Liquor	**5.269**	**5.269**	90.995	221.128	160.559	141.882	99.647	185.888	71.074	**8.530**	131.805	231.344	**60.172**
57	Man	2.243	2.259	**1.369**	39.939	**1.776**	36.704	57.023	35.067	2.319	33.117	2.235	**1.647**	22.483
58	Matrix	**52.274**	76.423	194.545	105.784	113.692	**55.029**	181.533	129.105	108.855	**73.544**	151.587	185.492	76.735
59	Mhyang	**2.936**	3.922	**2.592**	9.063	**3.610**	20.402	12.512	13.285	7.403	113.066	3.975	24.272	4.355
60	MotorRolling	**102.511**	228.644	145.713	174.172	621.137	160.991	146.233	162.792	172.081	**135.445**	**131.760**	292.706	164.724

Table A2c: Comparison of CLE between KCFop with other algorithms

61	MountainBike	6.572	7.661	8.626	155.083	6.506	73.023	206.739	214.278	12.036	24.913	178.763	240.303	9.784
62	Panda	41.071	42.055	7.181	48.457	77.039	5.948	61.655	9.000	143.362	78.297	64.022	78.602	54.600
63	RedTeam	3.529	3.807	4.275	50.261	3.033	6.651	54.900	4.045	5.803	71.862	16.806	82.192	52.935
64	Shaking	6.757	112.502	30.698	26.292	17.165	24.027	192.146	80.017	191.559	82.595	129.206	274.379	9.046
65	Singer1	12.635	12.825	14.514	18.784	14.013	16.365	88.869	15.534	12.928	141.389	11.361	318.416	4.191
66	Singer2	8.585	10.283	174.335	21.848	185.47	22.530	88.634	127.308	187.607	76.837	163.593	355.743	43.690
67	Skater	10.667	10.691	8.204	46.034	13.751	10.202	24.253	27.406	20.669	9.499	7.622	84.168	9.737
68	Skater2	13.820	17.901	15.702	125.037	16.786	35.372	20.016	63.156	24.419	16.036	26.465	174.289	22.351
69	Skating1	7.448	7.668	82.963	174.235	7.785	139.400	149.345	150.76	43.211	110.504	129.768	287.636	9.347
70	Skating2	23.374	30.762	57.883	139.994	221.975	108.805	125.839	73.728	54.814	15.054	170.658	336.207	23.110
71	Skiing	180.515	260.053	251.928	276.200	247.592	266.975	270.012	256.897	195.489	247.865	153.129	251.708	263.270
72	Soccer	13.516	15.375	71.360	139.532	70.510	77.853	95.355	92.397	127.533	42.237	89.216	220.885	23.557
73	Subway	2.461	2.969	4.469	3.312	164.372	7.595	15.792	11.095	112.994	4.713	139.508	105.362	141.319
74	Surfer	3.964	8.737	9.043	150.88	161.750	16.969	51.593	35.104	72.064	26.080	3.072	218.530	11.272
75	Suv	3.425	3.491	50.622	112.122	573.235	82.965	42.001	72.669	31.138	29.478	9.891	120.230	57.247
76	Sylvester	12.549	12.918	6.299	44.883	9.916	15.204	14.996	8.556	14.807	11.350	14.782	101.486	19.601
77	Tiger1	7.654	8.053	128.697	41.298	69.863	108.927	74.345	29.928	94.881	111.375	45.361	105.258	107.270
78	Tiger2	10.974	47.437	21.878	12.428	59.558	27.313	113.542	28.374	252.663	150.377	41.437	193.463	40.875
79	Toy	7.653	7.796	11.371	31.882	43.005	58.059	27.428	22.431	43.391	53.031	6.120	153.222	10.628
80	Trans	27.272	54.675	43.186	56.155	39.305	89.033	38.962	36.768	55.209	59.961	67.228	333.400	45.329
81	Trellis	7.379	7.764	6.917	44.872	18.824	71.469	59.508	41.695	98.330	47.738	7.006	69.042	32.250
82	Twinnings	4.414	6.767	5.927	12.777	9.973	9.853	17.361	16.388	17.765	20.959	21.861	111.25	6.253
83	Vase	10.567	12.428	24.276	13.260	12.879	18.953	18.168	15.827	34.694	17.500	14.118	172.288	19.635
84	Walking	4.280	3.971	4.632	5.870	7.169	3.397	9.306	6.950	5.339	2.352	205.672	327.096	5.783
85	Walking2	8.523	28.983	11.163	29.086	17.931	60.647	57.525	58.526	29.108	64.866	34.689	12.288	46.245
86	Woman	9.663	10.063	4.168	8.498	207.336	125.293	111.924	114.482	32.277	130.903	72.492	211.548	118.863
	Average	**19.803**	**38.789**	**44.211**	**78.600**	**91.506**	**65.264**	**78.809**	**77.431**	**71.973**	**63.374**	**66.811**	**175.495**	**53.663**

REFERENCES

Adam, A., E. Rivlin and I. Shimshoni, 2006. Robust fragments-based tracking using the integral histogram. Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), June 17-22, 2006, IEEE, New York, USA., ISBN:0-7695-2597-0, pp: 798-805.

Avidan, S., 2004. Support vector tracking. IEEE Trans. Patt. Anal. Mach. Intel., 26: 1064-1072.

Babenko, B., M.H. Yang and S. Belongie, 2009. Visual tracking with online multiple instance learning. Proceedings of the International Conference on Computer Vision and Pattern Recognition, June 20-25, 2009, Miami, FL., pp: 983-990.

Bao, C., Y. Wu, H. Ling and H. Ji, 2012. Real time robust L1 tracker using accelerated proximal gradient approach. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 16-21, 2012, IEEE, Providence, Rhode Island, ISBN:978-1-4673-1226-4, pp: 1830-1837.

Bolme, D.S., J.R. Beveridge, B.A. Draper and Y.M. Lui, 2010. Visual object tracking using adaptive correlation filters. Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern RecognitionIn (CVPR), June 13-18, 2010, IEEE, San Francisco, California, ISBN:978-1-4244-6984-0, pp: 2544-2550.

Chen, S., S. Li, R. Ji, Y. Yan and S. Zhu, 2016a. Discriminative local collaborative representation for online object tracking. Knowl. Based Syst., 100: 13-24.

Chen, W., K. Zhang and Q. Liu, 2016b. Robust visual tracking via patch based kernel correlation filters with adaptive multiple feature ensemble. Neurocomput., 214: 607-617.

Cheng, X., N. Li, T. Zhou, L. Zhou and Z. Wu, 2015. Object tracking via collaborative multi-task learning and appearance model updating. Appl. Soft Comput., 31: 81-90.

Dalal, N. and B. Triggs, 2005. Histograms of oriented gradients for human detection. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), June 20-25, 2005, IEEE, San Diego, California, ISBN:0-7695-2372-2, pp: 886-893.

Dinh, T.B., N. Vo and G. Medioni, 2011. Context tracker: Exploring supporters and distracters in unconstrained environments. Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 20-25, 2011, IEEE, Colorado Springs, Colorado, ISBN:978-1-4577-0394-2, pp: 1177-1184.

Fan, B., Y. Du and Y. Cong, 2014. Online learning discriminative dictionary with label information for robust object tracking. Abstract Appl. Anal., 2014: 1-12.

Fergus, R., P. Perona and A. Zisserman, 2003. Object class recognition by unsupervised scale-invariant learning. Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Vol. 2, June 18-20, 2003, IEEE, Madison, Wisconsin, ISBN:0-7695-1900-8, pp: II-264-II-271.

Grabner, H., C. Leistner and H. Bischof, 2008. Semi-supervised on-line boosting for robust tracking. Proceedings of the 10th European Conference on Computer Vision, October 12-18, 2008, Marseille, France, pp: 234-247.

Grabner, H., M. Grabner and H. Bischof, 2006. Real-time tracking via online boosting. BMVA, 1: 47-56.

Henriques, J.F., R. Caseiro, P. Martins and J. Batista, 2012. Exploiting the circulant structure of tracking-by-detection with kernels. Proceedings of the European Conference on Computer Vision, October 7-13, 2012, Springer, Berlin, Germany, ISBN:978-3-642-33764-2, pp: 702-715.

Henriques, J.F., R. Caseiro, P. Martins and J. Batista, 2015. High-speed tracking with kernelized correlation filters. IEEE. Trans. Pattern Anal. Mach. Intell., 37: 583-596.

Hu, J., J. Lu and Y.P. Tan, 2016. Deep metric learning for visual tracking. IEEE. Trans. Circuits Syst. Video Technol., 26: 2056-2068.

Jia, X., H. Lu and M.H. Yang, 2012. Visual tracking via adaptive structural local sparse appearance model. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 16-21, 2012, Rhode Island, USA., pp: 1822-1829.

Kwon, J. and K.M. Lee, 2010. Visual tracking decomposition. Proceedings of the Conference on Computer Vision and Pattern Recognition, June 13-18, 2010, San Francisco, CA., USA., pp: 1269-1276.

Li, G., B. Ma, J. Huang, Q. Huang and W. Zhang, 2016b. Beyond appearance model: Learning appearance variations for object tracking. Neurocomputing, 214: 796-804.

Li, X., Q. Liu, Z. He, H. Wang and C. Zhang *et al*., 2016a. A multi-view model for visual tracking via. correlation filters. Knowl. Based Syst., 113: 88-99.

Liu, B., J. Huang, C. Kulikowski and L. Yang, 2013. Robust visual tracking using local sparse appearance model and K-selection. IEEE. Trans. Pattern Anal. Mach. Intell., 35: 2968-2981.

Liu, B., L. Yang, J. Huang, P. Meer and L. Gong *et al*., 2010. Robust and fast collaborative tracking with two stage sparse optimization. Proceedings of the European Conference on Computer Vision, September 5-11, 2010, Springer, Berlin, Germany, ISBN:978-3-642-15560-4, pp: 624-637.

Liu, F., T. Zhou and J. Yang, 2016. Geometric affine transformation estimation via correlation filter for visual tracking. Neurocomputing, 214: 109-120.

Mei, X. and H. Ling, 2009. Robust visual tracking using l1 minimization. Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, September 29-October 2, 2009, IEEE, Kyoto, Japan, ISBN:978-1-4244-4420-5, pp: 1436-1443.

Mei, X., H. Ling, Y. Wu, E. Blasch and L. Bai, 2011. Minimum error bounded efficient $/ell_1$ tracker with occlusion detection. Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), June 20-25, 2011, IEEE, Washington, USA., ISBN:978-1-4577-0394-2, Pages: 1257-1264.

Oron, S., A. Bar-Hillel, D. Levi and S. Avidan, 2015. Locally orderless tracking. Intl. J. Comput. Vision, 111: 213-228.

Patras, I. and E.R. Hancock, 2010. Coupled prediction classification for robust visual tracking. IEEE. Trans. Pattern Anal. Mach. Intell., 32: 1553-1567.

Qian, C. and Z. Xu, 2016. Robust visual tracking via sparse representation under subclass discriminant constraint. IEEE. Trans. Circuits Syst. Video Techn., 26: 1293-1307.

Ross, D.A., J. Lim, R.S. Lin and M.H. Yang, 2008. Incremental learning for robust visual tracking. Int. J. Comput. Vision, 7: 125-141.

Sabzmeydani, P. and G. Mori, 2007. Detecting pedestrians by learning shapelet features. Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07), June 17-22, 2007, IEEE, Minneapolis, Minnesota, ISBN:1-4244-1179-3, pp: 1-8.

Sevilla-Lara, L. and E. Learned-Miller, 2012. Distribution fields for tracking. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 16-21, 2012, IEEE, Providence, Rhode Island, ISBN:978-1-4673-1226-4, pp: 1910-1917.

Shotton, J., J. Winn, C. Rother and A. Criminisi, 2009. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout and context. Intl. J. Comput. Vision, 81: 2-23.

Tian, C., X. Gao, W. Wei and H. Zheng, 2015. Visual tracking based on the adaptive color attention tuned sparse generative object model. IEEE. Trans. Image Process., 24: 5236-5248.

Winn, J., A. Criminisi and T. Minka, 2005. Object categorization by learned universal visual dictionary. Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV'05) Vol. 1, October 17-21, 2005, IEEE, Beijing, China, ISBN:0-7695-2334-X, pp: 1800-1807.

Wu, Y., J. Lim and M.H. Yang, 2013. Online object tracking: A benchmark. Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2013, IEEE, Portland, Oregon, USA., ISBN:978-0-7695-4989-7, pp: 2411-2418.

Yang, F., H. Lu and M.H. Yang, 2014. Robust superpixel tracking. IEEE. Trans. Image Process., 23: 1639-1651.

Yang, H., L. Shao, F. Zheng, L. Wang and Z. Song, 2011. Recent advances and trends in visual tracking: A review. Neurocomputing, 74: 3823-3831.

Yang, M., Y. Wu, M. Pei, B. Ma and Y. Jia, 2016. Online discriminative tracking with active example selection. IEEE. Trans. Circuits Syst. Video Technol., 26: 1279-1292.

Yuan, Y., J. Fang and Q. Wang, 2014. Robust superpixel tracking via depth fusion. IEEE. Trans. Circuits Syst. Video Technol., 24: 15-26.

Zhang, K. and H. Song, 2013. Real-time visual tracking via online weighted multiple instance learning. Pattern Recognit., 46: 397-411.

Zhang, K., L. Zhang and M.H. Yang, 2012. Real-time compressive tracking. Proceedings of the European Conference on Computer Vision, October 7-13, 2012, Springer, Berlin, Germany, ISBN:978-3-642-33711-6, pp: 864-877.

Zhang, L., D. Bi, Y. Zha, S. Gao and H. Wang *et al*., 2016. Robust and fast visual tracking via spatial kernel phase correlation filter. Neurocomputing, 204: 77-86.

Zhuang, B., H. Lu, Z. Xiao and D. Wang, 2014. Visual tracking via discriminative sparse similarity map. IEEE. Trans. Image Process., 23: 1872-1881.

Zhuang, B., L. Wang and H. Lu, 2016. Visual tracking via shallow and deep collaborative model. Neurocomputing, 218: 61-71.