IJCSI International Journal of Computer Science Issues, Volume 13, Issue 5, September 2016
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.20943/01201605.108119

A Comparative Study of Tracking Moving Objects in videos

Walaa Omar El-Farouk Badr [1], Hossam. El-Din Mostafa [2], and Rasheed Mokhtar El-Awady [3]

[1] Assistant Lecturer of Electronics and Communication department, Mansoura Higher Institute for Engineering and Technology.

[2] Lecturer of Electronics and Communication of Engineering, Mansoura University.

[3] Prof. of Electronics and Communication Engineering, Mansoura University.

ABSTRACT

Visual tracking is considered to be one of the most important challenges in computer vision with numerous applications such as object recognition and detection. In the present paper, five tracking techniques will be introduced circulant structure with kernels (CSK), Kernelized correlation filters (KCF), Adaptive color attributes (ACT), distractor – awareness tracker (DAT), and Multi-Template Scale KCF (MTSc-KCF) for the visual object tracking (VOT14), and VOT15 challenge datasets. Performance evaluation for each method was calculated using four measures; center location error (CLE), overlap precision (OP), distance precision (DP), and speed in frames per second (FPS). Results have shown that KCF tracker is the fastest technique in VOT14 but the CSK tracker is the fastest in VOT15. They are used in time-critical application with satisfactory performance. MTSc-KCF, and KCF achieve the best results in most sequences and the highest precision at lower threshold. Each tracker performs favorable and competitive results in some sequence and fails in others. So it is noted that the choice of the tracker is application dependent.

Keywords: *Visual tracking; correlation filter; distractor; distance precision; precision plot.*

1. INTRODUCTION

One of the main goals of computer vision is to enable computers to replicate the basic functions of human vision such as motion perception and scene understanding. To achieve the goal of intelligent motion perception, much effort has been spent on visual object tracking. Essentially, the core of visual object tracking is to robustly estimate the motion state (i.e., location, orientation, size, etc.) of a target object in each frame of an input image sequence

Visual object tracking is a classical and very popular problem in computer vision with a lot of applications such as vehicle navigation, human computer interface, human motion analysis, surveillance, image understanding, human-computer interaction, , and robotics and many more. It is concerned with low-level visual processing and high-level image analysis. Despite numerous object tracking methods that have been proposed in recent year, most of these trackers suffer a degradation in performance mainly because of several challenges that include illumination changes, partial or full occlusion, motion blur, complex motion, out of plane rotation, and camera motion [1].

Visual object tracking methods include image input, appearance feature description, context information integration, decision and model update as shown in Fig.1 [2]. Most trackers either depend on intensity or texture information [3, 4], while others depend on color information that is limited to simple color space transformation [5]. In contrast to visual tracking, color features are providing excellent performance for different application such as object recognition and detection. Using color information for visual tracking is a very difficult problem due to variation in illumination, shadows, shading, camera, and object geometry. So, it is a must to choose the suitable color transformation for visual object tracking.

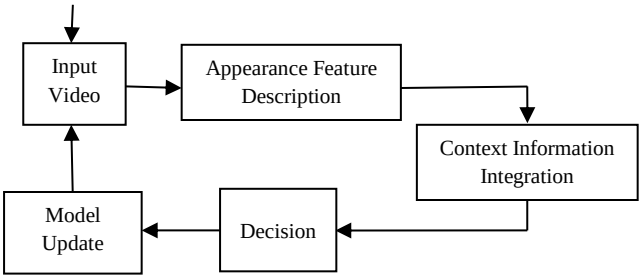

Fig. (1) The flowchart of visual tracking

To tackle these challenges; firstly, the objective is to determine the position of the object in the first frame either manually or by using reference model (ground truth). Secondly, it is a must to detect the locations of object in an image sequence with separating the target object from the background. The object is tracked in each frame of the video by several approaches. In this paper, there is a comparison between several methods of visual tracking that provide high

performance among the top visual trackers such as: tracking by detection approach circulant structures with kernels (CSK), KCF approach, Adaptive color attributes (ACT), Multi-Template Scale MTSc-KCF, and distractor awareness (DAT). The aim of these methods is to track the object in each frame by trying to find out the region in the frame whose interior generates a sample distribution over the target object model which has the best match with the reference model distribution.

2. PREVIOUS WORK

Due to the importance of visual tracking, many approaches have been proposed to handle its problems. There exist two main approaches namely discriminative and generative methods that are used to handle the different problems of visual tracking. The generative methods handle the problem by searching for regions that are most similar to the target model [4, 6]. Recent benchmark evaluation shows that the generative models are outperformed by discriminative approaches which incorporate binary classifier to distinguish the object from its surrounding background [7, 8]. The models in these methods are based on templates, subspace models, HOG features [9] and Haar-like features [4, 3]. However, rectangular initialization bounding boxes include background information. Another method uses segmentation methods in order to improve the generative methods, but these methods still suffer from missing the advantages of discriminative methods to distinguish the object from its surrounding background [10, 7]. Another problem with using template-based methods is that the objective function is not enough to achieve the optimum solution [11]. So, an alternative is the use of histogram to describe the object. Histogram-based (kernel-based) descriptors integrate information over a large patch of the image. So they are not sensitive to spatial structure and give best results due to they are very fast. Another approach that uses multiple kernel to overcome the problem of losing of spatial information, which happens when building the histogram and improve the results of single histogram descriptor [12, 11], but it requires other mechanism to determine the number and shape of the kernels. If the number of kernel is too small, other additions are statistics analysis, and using feature selection [13, 14]. Distribution fields (DFs) uses a histogram that contains robust information and preserves the spatial information of the object by using a distribution at every pixel. It can be shown that it is a combination of histogram-based descriptors and template-based descriptors [11].

On the other side, the discriminative approaches pose the problem by differentiating the target from the background by using tracking as a binary classification problem. It has also been exploited to handle appearance changes during visual tracking, where a classifier is trained and updated online to distinguish the object from the background. This method is also termed as tracking by detection, in which a target object identified by the user in the first frame is described by a set of features. A separate set of features describes the background, and a binary classifier separates target from background in successive frames. To handle appearance changes, the classifier is updated incrementally over time. They also exploit visual information from the target and the background. Due to the success of discriminative approaches, many classifiers can be explored such as: SVMs, RVM [3] and several methods which depend on boosting [15] in order to distinguish the foreground from the background by an ensemble of classifiers. Some trackers use a tracking method that recognizes the object representation by partial least squares analysis and using more than one appearance model which is initialized in the first frame [14].

3. TRACKING APPROACHES

3.1 The CSK Tracker:

Tracking by detection has been proved to be a successful method. This stems directly from the development of discriminative methods in machine analysis, and their application to detect with offline training. It provides the highest speed among the visual trackers due to the circulant structure of the kernel. This method explores a dense sampling strategy by training a Gaussian kernel classifier with all subwindows (samples). It allows more efficient training. The reason is that the kernel matrix in this case becomes highly structured and circulant. This algorithm could be operated directly on the pixel values and without using feature extraction due to the using of fast Fourier transform.

Steps of Tracking:

- Initializing the target object in the first frame manually or using the first position of the object from the ground truth.
- Training images must be pre-processed with cosine windows
- Calculating the response of the classifier at all locations(subwindows) by using dense gauss kernel and FFT using equation of detection as follows,

$$response = real\left(ifft2\left(alphaf\ .*\ fft2(k)\right)\right) \quad (1)$$

where $alphaf$ is a classifier coefficient in Fourier transform and $(.*)$ called product–wise operation in matlab and $fft2(\,)$, $ifft2(\,)$ are fast Fourier transform, inverse fast Fourier transform respectively in matlab.

- Finding the maximum response.
- Getting subwindow at the current position of the target so as to train the classifier
- Training new models in order to determine new alpha and new position where alpha is kernel regularized least square solution (KRLS) according to the equation,

$$alphaf_{new} = yf\ ./fft2(k) + \lambda \quad (2)$$

where yf is a classifier response in Fourier transform and λ is regularization parameters, $\lambda = 10^{-2}$ in the present work.

- Finding Gaussian kernel (k) by using dense gauss kernel function as follows,

$$k = \exp(-\frac{1}{\sigma^2}\left(\|x^2\| + \|z^2\| - 2F^{-1}(F(x) \odot F^*(z))\right) \quad (3)$$

where $\sigma = 0.2$ is a gaussian kernel bandwidth, x is training image at current frame and z is test image at next frame. $F(.)$, $F^{-1}(.)$ denote fourier transform and inverse fouruer transform respectively [4].

3.2 The KCF Tracker:

It is the new version of CSK tracker but it deals with multi-channel HOG features for best performance. It depends on Gaussian Kernel correlation. The input patches are weighted by a cosine window that smoothly removes discontinuities at the image boundaries caused by the cyclic shift. The region of tracking has three times the size of the target to provide additional negative samples and some context.
Due to the training samples which consist of shifts of a base sample, it is a must to specify a regression target for each one in y [9].
The tracker implements three functions as follows:

- Training function: it trains the image patch at the initial position of the target

$$Alpha = train(x, y, sigma, lambda),$$

where x is the train image patch, sigma is feature bandwidth, and lambda is regularization factor [9].

- Detecting function: it detects over the patch at the previous position and the target position is updated to the one that has the maximum value. So, train has a new model at the new position.

$$response = detect(alphaf, x, z, sigma),$$

where z is the test image patch.

- Kernel correlation function, as it is called by the two previous functions, will compute Gaussian kernel correlation between x, z

$$K = kernal\ correlation(x, z, sigma),$$

where k can be written as k^{xz},

$$k^{xz} = \exp(-\frac{1}{\sigma^2}(\|x^2\| + \|z^2\| - 2F^{-1}(\sum_c \hat{x}_c^* \odot \hat{z}_c))) \quad (4)$$

where $\sigma = 0.5$, F^{-1} is the Fourier inverse, $\hat{x}_c^*$ is the conjugate of Fourier train patch for channel c and $\hat{z}_c$ Fourier test patch for channel c. The operations are only element-wise operations in Fourier domain due to the diagonalization, which result in the tracker to be the faster one. One challenge for the system is that happens due to the absence of a failure recovery mechanism. For more details of the tracker [9].

3.3 The ACT Tracker:

It is the extension of the CSK tracker with color attributes, which have shown excellent performance and results for object recognition. Color attributes, or color names (CN), are linguistic color labels assigned by humans to represent colors in the world. In a linguistic study performed by Berlin and Kay [16], it was concluded that the English language contains eleven basic color terms: black, blue, brown, grey, green, orange, pink, purple, red, white and yellow. In the field of computer vision, color naming is an operation that associates RGB observations with linguistic color labels. The mapping provided by [17] is used, this mapping is automatically recognized from images retrieved with Google-image search. This maps the RGB values to a probabilistic 11 dimensional color representation which sums up to 1.

Nevertheless, the high dimensionality of color attributes results in an increasing in the time performance and computational overhead, which could limit its application in real time surveillance. So, in order to overcome this problem, the ACT tracker proposes an adaptive dimensionality reduction technique which reduces the eleven dimension of the color attributes to two only [18].

The ACT updates the MOSSE tracker from linear kernels classifier and one dimensional feature to Gaussian classifier and multi-channel color features to be sub-optimal. Since the visual tracking is sensitive to appearance changes, so it is necessary for the target model to be updated over time through equation,

$$\dot{x}^p = (1 - \gamma)\dot{x}^{p-1} + \gamma x^p \quad (5)$$

where $\dot{x}^p$ is the updated learned target appearance, γ is the learning rate for the appearance model update, and p is the index of the current frame. The advantage of this model that it is not needed to store all the previous appearance but only the current model in each new frame can be saved.

Finally, the tracker proposes an adaptive dimensionality reduction technique that preserves useful information while reducing the number of color dimensions.

3.4 The DAT Tracker*:*

The tracker presents a discriminative object model to differentiate the object of interest from the background. Also, it relies on standard color histograms. In contrast, it extends this model to identify and suppress distracting region in advance to improve the tracking performance. It proposes an efficient scale estimation scheme which gives the chance and allows obtaining accurate tracking results.

There is a difference between supporting and distracting regions. Supporting regions have different appearance than the target but co-occur with it, providing valuable cues to overcome occlusions. Distractors, on the other hand, exhibit similar appearance and may therefore get confused with the target. So, it needs to track these distracters in addition to the target in order to prevent drifting. DAT tracker adapts the object representation such as that potentially distracting region which is suppressed in advance with the background. So it combines object background model with distractor aware model to give the final object model as follow,

$$P(x \in O/b_x) = \lambda_p P(x \in O/O, D, b_x) + (1-\lambda_p)\, P(x \in O/O, S, b_x) \quad (6)$$

where λ_p=0.5, is a predefined weighting parameter [19].
Thus, applying this model causes high likelihood scores while decreasing the effect of distracting region. So no explicit tracking of distractors is required
It uses tracking-by-detection principle to localize the object of interest in a new frame and obtain the new location as follow,

$$\dot{O}_t = \arg_{O_{t,i}} \max(s_v(O_{t,i}) s_d(O_{t,i})) \quad (7)$$

where $s_v(.)$, $s_d(.)$ denote vote score and distance score respectively. After localizing the object, it performs scale estimation models to adapt the scale of the current object hypothesis $\dot{O}_t$ according to,

$$O_t = \lambda_s O_t^S + (1-\lambda_s)\ \dot{O}_t \quad (8)$$

where λ_s=0.2 scale update parameter [19].

3.5 The MTSc-KCF Tracker

It is the update of the KCF tracker by addressing two drawbacks that causes failure of it. Although the efficiency of the KCF tracker but it uses a fixed target scale in detection and a filter update rule that makes the use of only one template at time [20].

MTSc-KCF proposes two component to solve these two drawbacks. First, it associate multiple multi-dimensional templates in computing the optimal filter taps. Second, it addresses the problem of fixed scale tracking by using scale scheme [20].

3.5.1 Multiple Template

It allows for training with multiple templates so more than one circulant matrix X. X is data matrix containing all templates and all their cyclic shift where X=[X_1, X_2,….X_n], where each X_i is a ciculant matrix generated from the i^{th} template.
The multiple template kernelized correlation problem can be formulated for two training examples as [20]:

$$\min_{w_1} \sum_i (w_1^T\ \emptyset(x_i) - y_i) + \lambda \|w_1\|_2^2 + \mu \left\|w_1 - w_2^j\right\|_2^2$$

$$\min_{w_1} \textstyle\sum_i (w_1^T\ \emptyset(x_i) - y_i) + \lambda \|w_1\|_2^2 + \mu \left\|w_1 - w_2^j\right\|_2^2 \quad (9)$$

where w_1, w_2 are filter taps that must be trained, j is the number of iteration j=10, the regularization parameter set to $\lambda = 10^{-4}$, we set $\mu = 10^{-5}$ as an initial value and it doubles in each iteration. Then we solve for w_1, w_2 via alternating fixed point optimization.

First, a solution for w_2 is initialized and uses it to update w_1. Then we use the updated w_1 to solve for w_2 and so on, until we achieve the stopping criterion.

3.5.2 Scale scheme

It is used to address the target scale issue to solve the problem of scale target variation. It selects the scale that maximizes the posterior probability instead of likelihood as follow [20],

$$\max_i p(s_i/y) = P(y\ /\ s_i)\ P(s_i) \quad (10)$$

Where s_i is i^{th} scale, and $p(y\ /\ s_i)$ is the likelihood that is defined by the maximum detection response at the i^{th} scale. The prior term $P(s_i)$ is assumed to follow a Gaussian distribution that it centered around the previous scale and σ is standard deviation. This scale scheme makes the tracker more sensitive to gradual scale changes.

4. EXPEREMENTAL RESULTS

4.1 Datasets

The presented approaches were implemented using Matlab version R2015a (8.5) on Intel Core(TM) i5-3230M CPU 2.60 GHz with 4GB RAM. The VOT14, VOT15 datasets had been utilized. These dataset contains videos which have been collected from well-known tracking evaluations: such as the Amsterdam Library of Ordinary Videos (ALOV) [21]. The VOT committee proposed a sequence selection methodology in order to compile datasets which cover various real-life visual phenomena. As a result, the VOT14 datasets consist of 25 sequences [22], VOT15 datasets contain 60 video sequences [23]. These sequences pose challenging situations such as illumination changes, object deformations and appearance changes, abrupt motion changes, significant scale variations, camera motion, and occlusions. These challenging datasets are considered to be the largest model free tracking benchmarks till now. Both datasets have been annotated with ground-truth to account for non-standard rectangles that can be rotated or scaled.

4.2 Evaluation methodology

To compare between the presented algorithms, results were compared using three evaluation metrics, center location error (CLE), overlap precision (OP), and distance precision (DP). CLE is computed as the average Euclidean distance between the estimated center location of the target and the ground-truth. OP or in other words success rate is defined as relative number of frames where the overlap between the tracking and ground-truth bounding box exceeds a certain threshold c. We estimate OP at threshold of 0.5. DP is the relative number of frames in the sequence where the center location error is smaller than a certain threshold [18]. The DP values were reported at a threshold of 20 pixels [6, 18]. We must know that the more OP, DP, and FPS scores

increase, the more the success of the tracker increases. However, the more decrease in CLE scores, means the more success of the tracker

A precision plot shows the ratio of successful frames whose tracker output is within the given threshold (x-axis of the plot, in pixels) from the ground-truth, measured by the center distance between bounding boxes.

In the precision plots, the distance precision is plotted over a range of thresholds as shown in Figures (2), (3). The trackers were ranked using the DP scores at 20 pixels. A higher precision at low thresholds means that the tracker is more accurate, while a lost target will prevent it from achieving perfect precision for a very large threshold range. When a representative precision score is needed, the chosen threshold is 20 pixels, as done in previous works.

4.3 Results

VOT14 Experiments: From the results in Table (4), KCF tracker is the best in case of speed (runs at hundreds of frames per seconds) so it can be used in time critical applications such as visual surveillance or robotics. The results were summarized in Tables (1), (2), and (3) using the OP, DP, and CLE values respectively over all sequences. Also, the speed of the trackers was taken into consideration in median frames per second (FPS) as shown in Table (4). Figures (2.a), (2.b) show the precision plots for 16 different sequences taken from VOT14 datasets.
KCF achieves the best DP equally with ACT if it is compared with the other trackers. The ACT is the best in case of mean CLE at the cost of lower frames rates. It is also found that each tracker is the best in some sequences only. This is due to the attributes of the sequence such as illumination variations, pose angle, partial occlusion, background clutter, shape deformation, motion blur, scale variation, and out -of -plane rotation. But the ACT and KCF are more stable and reach the value 1 of the precision value at low threshold than the two others in most sequences. ACT tracker provides significant performance due to the using of color attributes but with lower speed. KCF is the fastest tracker due to its circulant structure and its diagonalization by the DFT.

VOT15 Experiments: in this dataset, we compare MTSc-KCF tracker with CSK, KCF, and ACT trackers. From Tables (5), (6), (7), we found that MTSc-KCF is the best in terms of mean DP, CLE. It achieves the best mean OP equally with KCF if it is compared with the other trackers. We must know that MTSc-KCF is the best in case of the majority of videos that include examples of target object subject to substantial variations in scale.
While the tracker performs very well, it is computationally quite complex, resulting in a very slow tracking, which limits its practical applicability. It will be interesting to see in future whether certain steps could be simplified to achieve a faster tracking with high performance.
The performance of ACT tracker in VOT15 is not as efficient as in vot14 because the sequences in this dataset do not include a lot of color sequences but rather include examples of target objects distinguished by scale variations.
We noted from the results in Table (8) that CSK tracker is the fastest one so it can be used in critical time application [6, 9]. We show the most accurate results of all trackers at different threshold as shown in precision plots in Figures (3.a), (3.b).

5. CONCLUSION

A comparison between tracking algorithms have been presented. It is noted that the CSK, KCF tracker runs at hundreds of FPS so they are suitable for real time-critical applications and can be implemented with only a few lines of code. It achieves competitive DP in the most of sequences of the dataset VOT14. It is noted that, if we use MTSc-KCF tracker in dataset VOT15, it achieve the best DP, and CLE but it is very slow. It is noted that all trackers fail in at least one sequence. Hence, the choice of a tracker depends on attributes of the video and the application under consideration.

REFERENCES

[1] Q. Wang and F. Chen, "Object Tracking via Partial Least Squares Analysis", IEEE Trans. Image Processing, vol. 21, no. 10, pp. 4454-4465, Oct 2012.
[2] H. Yang, L. Shao, F.Zheng, L.wang, and Z. Song, "Recent advances and trends in visual Tracking: A review", Neurocomputing, vol. 74, no. 18, pp. 3823-3831, Nov 2011.
[3] S. Hare, A. Saffari, and P. Torr, "Struck: Structured output tracking with kernels", In ICCV, 2011. IEEE International Conference on, pages 263-270.
[4] K. Zhang, L. Zhang, and M. Yang, "Real-time compressive tracking", In ECCV, 2012.
[5] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking", In CVPR, 2012.
[6] J. Henriques, R. Caseiro, P. Martins, and J. Batista," Exploiting the circulant structure of tracking-by-detection with kernels", In ECCV, 2012. Pages 702-715. Springer, 2012.
[7] M. Godec, P. M. Roth, and H. Bischof, "Hough-based Tracking of Non-Rigid Objects", In Proc. ICCV, 2011.
[8] Y. Wu and M.-H. Yang, "Online Object Tracking: A Benchmark", In Proc. CVPR, 2013.
[9] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High Speed Tracking with krnelized Correlation Filters", pattern Analysis and Machine Intelligence, IEEE Transaction on, 37(3): 583-596, 2015.
[10] V. Belagiannis, F. Schubert, N. Navab, and S. Ilic, "Segmentation Based Particle Filtering for Real-Time 2D Object Tracking", In Proc. ECCV, 2012.
[11] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking", In CVPR, 2012.
[12] G. Hager, G. D. Hager, M. Dewan, and C. V. Stewart, "Multiple kernel tracking with SSD", In CVPR, 2004.
[13] A. P. Leung and S. Gong, "Mean shift tracking with random sampling", In BMVC, 2006.

[14] M.Rutharesh, R.Naveenkumar, and S. Krishnakumar, "Object Tracking Using Partial Least Squares Analysis", In IJARCSSE, vol. 4, no.2, pp.235-239, Feb 2014.

[15] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervise On-line Boosting for Robust Tracking", In Proc. ECCV, 2008.

[16] B. Berlin and P. Kay," Basic Color Terms: Their Universality and Evolution", UC Press, Berkeley, CA, 1969.

[17] J. van de Weijer, C. Schmid, J. J. Verbeek, and D. Larlus, "Learning color names for real-world applications", TIP, 18(7):1512–1524, 2009.

[18] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive Color attributes for Real-Time Visual Tracking", In Proc. CVPR, 2014.

[19] H. Possegger, T.Manuthner, and H. Bischof, "In Defense of Color-based Model-free Tracking", In CVPR, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[20] A. Bibi, and B. Ghanem, "Multi Template Scale-Adaptive Correlation Filters", ICCV workshop, IEEE international Conference on, pages 50-57, 2015.

[21] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual Tracking: an Experimenta Survey", PAMI, 36(7):1442–1468, 2014.

[22] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L.Cehovin, G. Nebehay, G. Fernandez, et al, "The Visual Object Tracking VOT2014 challenge results", In Proc. VOT (ECCV Workshop), 2014.

[23] The vot 2015 evaluation kit. http://www.votchallenge.net.

Table 1. OP results for four trackers using VOT14 dataset

sequence	Overlap Precision (OP)			
	CSK	KCF	ACT	DAT
ball	0.37	0.72	0.68	**0.99**
basketball	0.88	0.89	0.8	**0.99**
bolt	**1**	**1**	**1**	0.97
bicycle	**0.47**	0.35	0.4	0.44
car	0.46	**0.48**	0.46	0.46
david	0.89	**0.95**	0.68	0.6
drunk	0.5	0.53	**0.55**	0.49
Fish1	0.16	0.11	**0.26**	**0.26**
Fish2	0.2	**0.27**	0.22	0.2
Hand1	0.25	0.3	**0.34**	0.26
Hand2	0.15	0.3	0.22	**0.68**
polarbear	0.66	**0.72**	0.58	0.29
skating	**0.38**	0.36	**0.38**	0.25
sphere	**1**	**1**	**1**	0.95
sunshade	**1**	**1**	**1**	0.82
surfing	**1**	**1**	**1**	**1**
torus	0.1	0.96	**0.97**	0.93
trellis	0.58	**0.63**	0.6	0.4
tunnel	**0.1**	**0.1**	**0.1**	0.09
woman	0.93	**0.94**	0.93	0.78
mean	0.55	**0.63**	0.61	0.59

Table 2. DP results for four trackers using VOT14 dataset

sequence	Distance Precision (DP)			
	CSK	KCF	ACT	DAT
ball	0.66	0.93	0.82	**0.97**
basketball	**1**	0.92	0.99	0.97
bolt	**1**	0.99	**1**	0.98
bicycle	**1**	**1**	**1**	0.96
car	0.68	**0.83**	0.71	0.75
david	0.86	**1**	0.41	0.49
drunk	0.38	0.4	**0.61**	0.37
Fish1	0.16	0.2	0.68	**0.90**
Fish2	0.23	0.28	**0.58**	0.44
Hand1	0.31	0.21	**0.43**	0.39
Hand2	0.2	0.23	0.26	**0.85**
polarbear	0.68	**0.96**	0.65	0.26
skating	**0.8**	0.73	0.73	0.34
sphere	**1**	0.99	**1**	0.84
sunshade	**1**	**1**	**1**	0.99
surfing	**1**	**1**	**1**	**1**
torus	0.3	1	**1**	**1**
trellis	0.99	**0.99**	0.77	0.73
tunnel	0.99	**1**	**1**	0.28
woman	0.94	**0.985**	**0.94**	0.76
mean	0.7	**0.78**	**0.78**	0.71

Table 3. CLE results for four trackers using VOT14 dataset

sequence	Center Location Error (CLE)			
	CSK	KCF	ACT	DAT
ball	15.9	10	13.7	**8.16**
basketball	**6.55**	8.35	9.45	8.8
bolt	4.46	6.63	**4.13**	7.7
bicycle	**4.22**	5.5	4.65	9.33
car	29.7	**12.7**	27.4	21.6
david	12	**7.17**	26.5	25.7
drunk	37.8	**24.6**	28.8	43.6
Fish1	111	103	20.2	**9.47**
Fish2	307	142	80.9	**76.5**
Hand1	**57.1**	74	62.7	72.9
Hand2	69	49.5	61.6	**15.8**
polarbear	15.4	**11.2**	19.3	24.2
skating	**14.8**	16.2	15.9	139
sphere	8.9	9.47	**6.6**	13.3
sunshade	**3.3**	4.3	**3.3**	11.1
surfing	**1.67**	2.29	2.03	2.46
torus	52.6	**3.7**	3.94	6.62
trellis	11.7	**10.1**	33.8	16.4
tunnel	10.8	**6.5**	10.2	27.2
woman	11.4	**7.67**	8.34	14.9
mean	39.15	25.74	**22.17**	27.74

Table 4. FPS results for four trackers using VOT14 dataset

sequence	Frame per Seconds (FPS)			
	CSK	KCF	ACT	DAT
ball	**117.39**	61.63	58.7	34.77
basketball	127.3	**154**	67.4	25.5
bolt	171.6	**211.47**	86.5	25.6
bicycle	205.6	**234**	84.4	50.2
car	262.98	**390.7**	103	51.2
david	40.76	**51.84**	31	42.04
drunk	18.37	**45.69**	31.9	29.68
Fish1	**199.15**	176.22	149	59.16
Fish2	**111.07**	96.05	58.3	26.31
Hand1	142.6	**183.58**	130	8.74
Hand2	442.3	**487.89**	152	31.68
polarbear	60.23	**115.57**	80.2	28.12
skating	70.78	**145.63**	68.7	28.6
sphere	**91.75**	87.29	33.4	40.09
sunshade	92.31	**93.79**	95.3	40.71
surfing	**271.84**	158.50	141	57.35
torus	**196.09**	140	80.2	40.55
trellis	**83.34**	50.43	17.5	57.96
tunnel	**92.9**	77.2	45.3	27.5
woman	**231.22**	75.8	87.5	39.68
mean	151.46	**151.85**	80.1	38.65

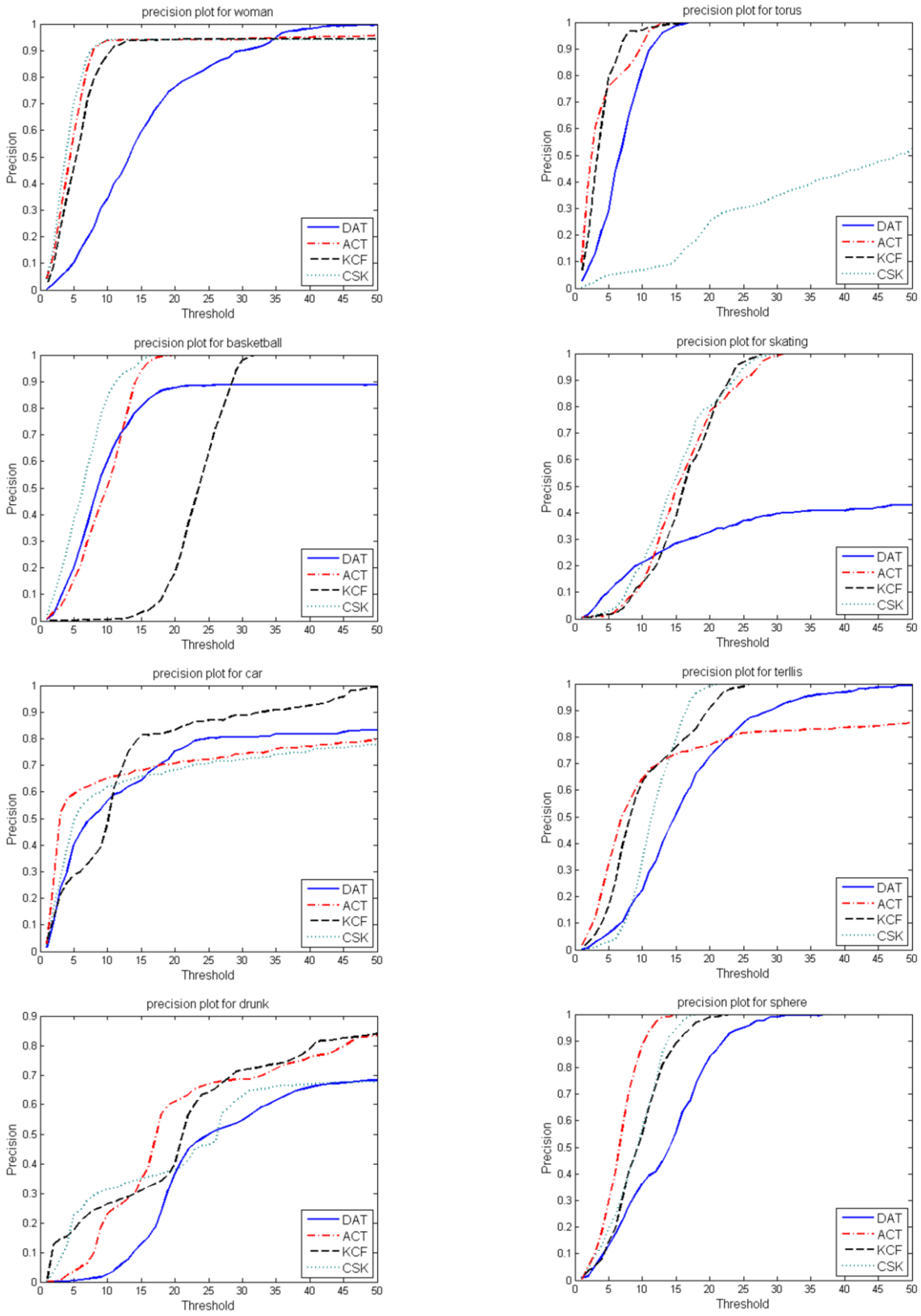

Fig. (2.a) Precision plots of eight different sequence of the VOT14 dataset

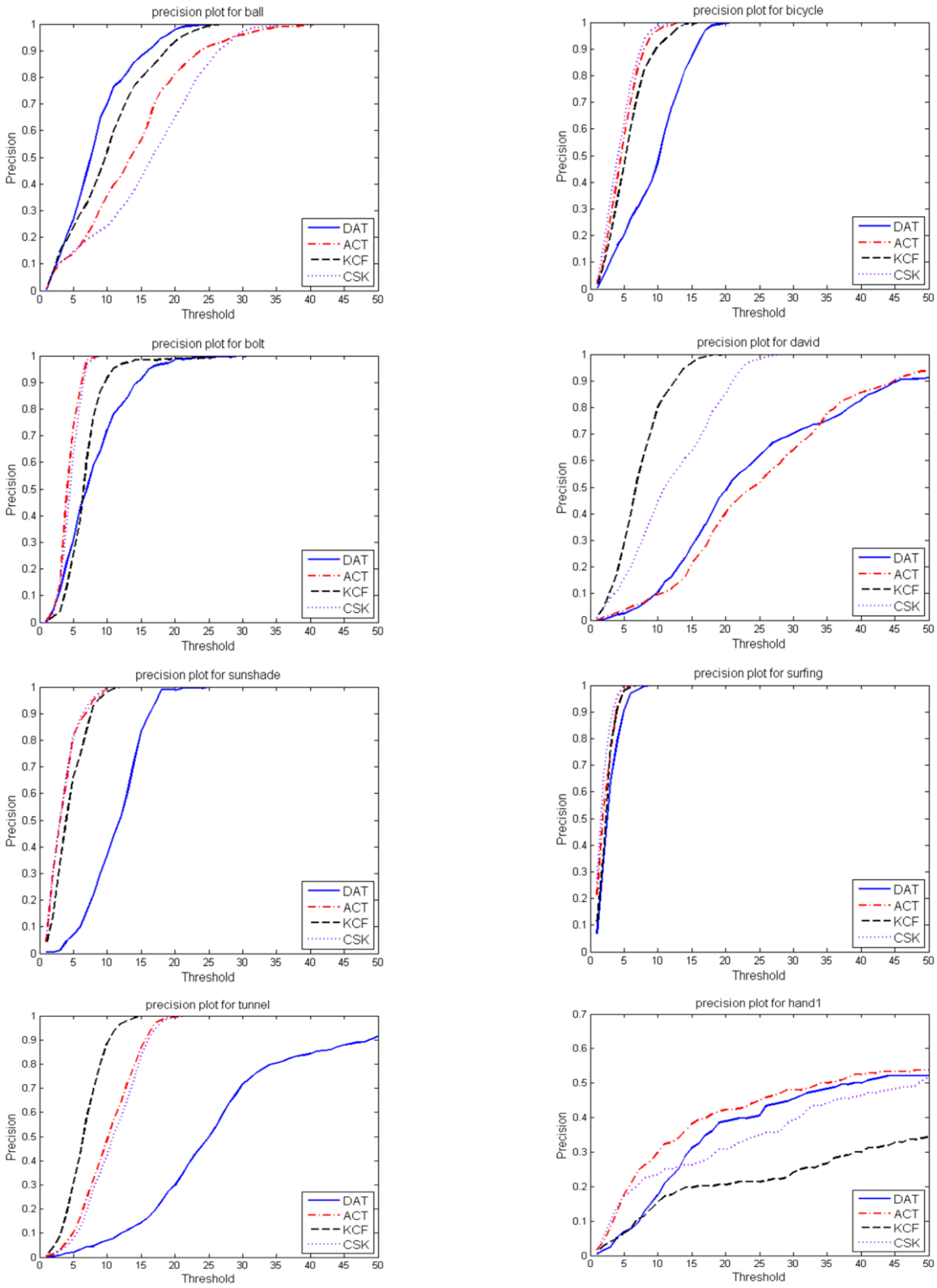

Fig. (2.b) Precision plots of eight different sequence of the VOT14 dataset

Table 5. OP results for four trackers using VOT15 dataset

sequence	Overlap Precision (OP)			
	CSK	KCF	ACT	AKCF
bag	**0.41**	0.18	0.33	0.29
birds2	0.58	0.549	0.625	**0.68**
crossing	0.50	**0.565**	0.504	0.55
car1	0.79	**0.795**	0.651	0.767
car2	**1**	0.6	**1**	0.52
blanket	0.116	**0.76**	0.156	0.326
dinosaur	0.46	0.347	**0.515**	0.38
godfather	**0.47**	0.39	0.28	0.385
helicopter	0.4	0.4	0.4	**0.42**
marching	1	1	1	1
racing	0.199	0.19	0.218	**0.263**
road	0.679	0.73	0.751	**0.81**
sheep	**0.546**	0.5	**0.546**	0.5
sphere	0.54	0.97	0.2	**0.985**
tunnel	0.298	0.29	0.304	**0.4**
traffic	**1**	**1**	**1**	**1**
wiper	0.75	**1**	0.998	**1**
mean	0.57	**0.6**	0.56	**0.6**

Table 6. DP results for four trackers using VOT15 dataset

sequence	Distance Precision (DP)			
	CSK	KCF	ACT	AKCF
bag	0.589	0.393	**0.617**	0.36
birds2	0.79	0.657	0.827	**0.874**
crossing	0.83	0.954	0.832	**0.985**
car1	**0.996**	**0.996**	0.698	0.956
car2	**1**	**1**	**1**	0.682
blanket	0.667	**0.987**	0.156	0.907
dinosaur	0.47	0.405	**0.684**	0.451
godfather	**1**	0.94	**1**	**1**
helicopter	0.41	0.347	0.513	**0.8**
marching	**1**	0.99	0.995	0.995
racing	**0.69**	0.391	0.474	0.635
road	0.919	**0.996**	O.99	**0.996**
sheep	0.924	**1**	0.99	0.98
sphere	0.502	0.667	0.1	**0.82**
tunnel	**1**	**1**	**1**	**1**
traffic	**1**	**1**	**1**	**1**
wiper	0.96	**1**	**1**	**1**
mean	0.81	0.81	0.7	**0.85**

Table 7. CLE results for four trackers using VOT15 dataset

sequence	Center Location Error (CLE)			
	CSK	KCF	ACT	AKCF
bag	**18.6**	33.3	24.1	28.4
birds2	15.3	16	11	**10.4**
crossing	19	14.4	18.3	**5.94**
car1	6.93	**4.9**	101	10.88
car2	2.93	6.12	**2.83**	19.5
blanket	19.5	**6.1**	15.4	13.6
dinosaur	37	40.2	**23.3**	46.8
godfather	**6.36**	9.2	8.45	8.24
helicopter	28.5	23.8	28.9	**13.9**
marching	**5.55**	6.32	6.73	6.58
racing	18	22.7	21.4	**16.1**
road	12.2	9.37	8.25	**6.6**
sheep	10.6	**7.25**	7.91	7.7
sphere	65.9	17.2	43.5	**16.4**
tunnel	8.62	7.54	8.17	**3.55**
traffic	3.79	3.3	4.15	**3.04**
wiper	8.67	5.2	4.78	**4.72**
mean	22.11	13.7	19.89	**13.08**

Table 8. FPS results for four trackers using VOT15 dataset

sequence	Frames per seconds (FPS)			
	CSK	KCF	ACT	AKCF
bag	**163.66**	79.4	24.8	13
birds2	**101.7**	46.1	52.4	7.63
crossing	36.92	**44.8**	20	7.2
car1	58.42	**88**	11.6	13.6
car2	**357.0**	333	176	50.2
blanket	124.5	**157**	117	18.8
dinosaur	121.58	**239**	86.7	9.11
godfather	288.81	**368**	159	43.4
helicopter	44.73	**62.9**	86.7	4.52
marching	52.16	**81.7**	12.9	12.7
racing	**99.39**	97.2	98.8	16.63
road	**182.51**	182	65.5	27.7
sheep	**468.25**	267	171	59.6
sphere	**126.38**	61.4	38.5	8.78
tunnel	**365.39**	226	230	41.2
traffic	**156.9**	103	82.8	19.6
wiper	**147.2**	108	89.4	15.3
mean	**170.3**	149.67	89.6	21.7

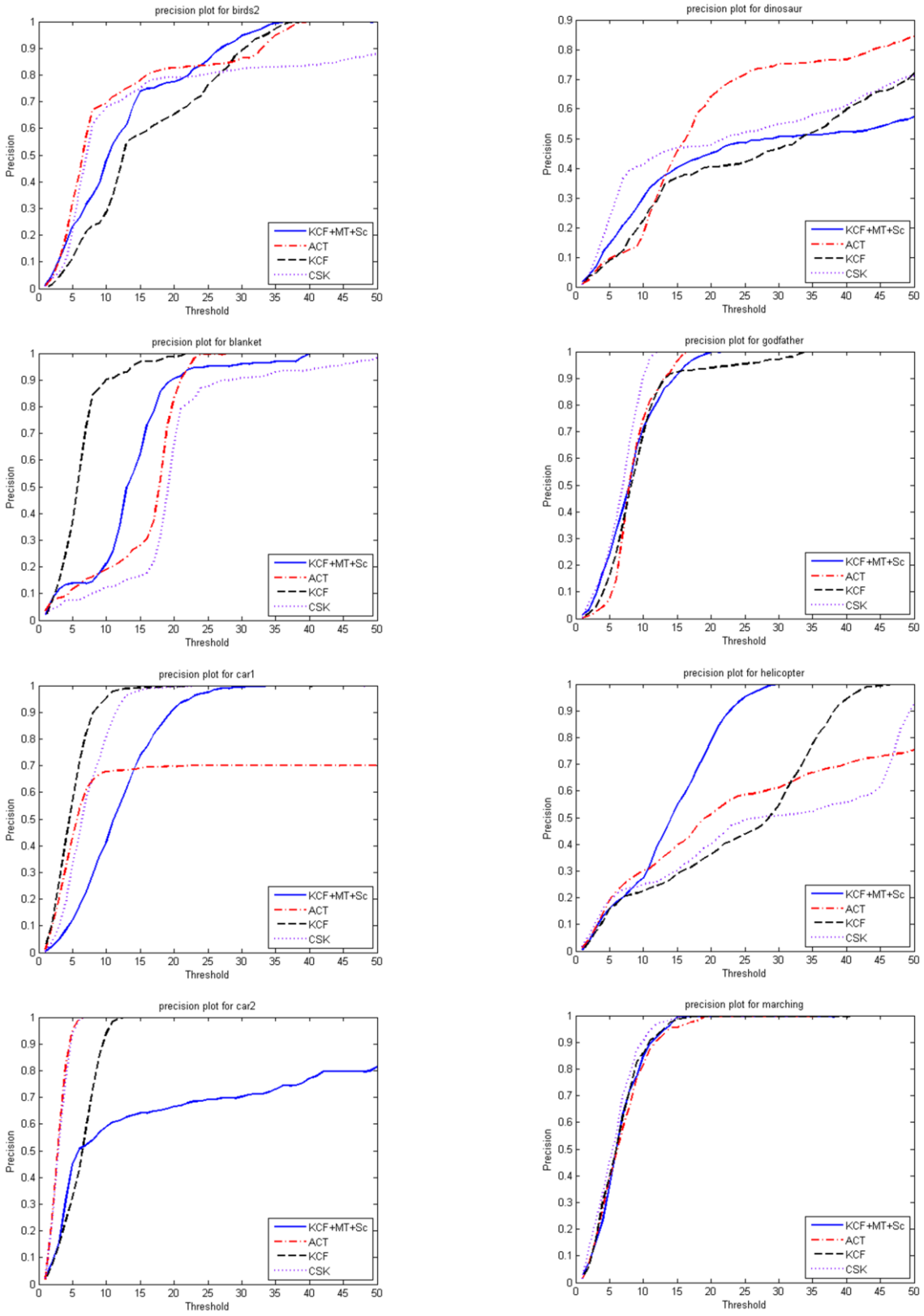

Fig. (3.a) Precision plots of eight different sequence of the VOT15 dataset

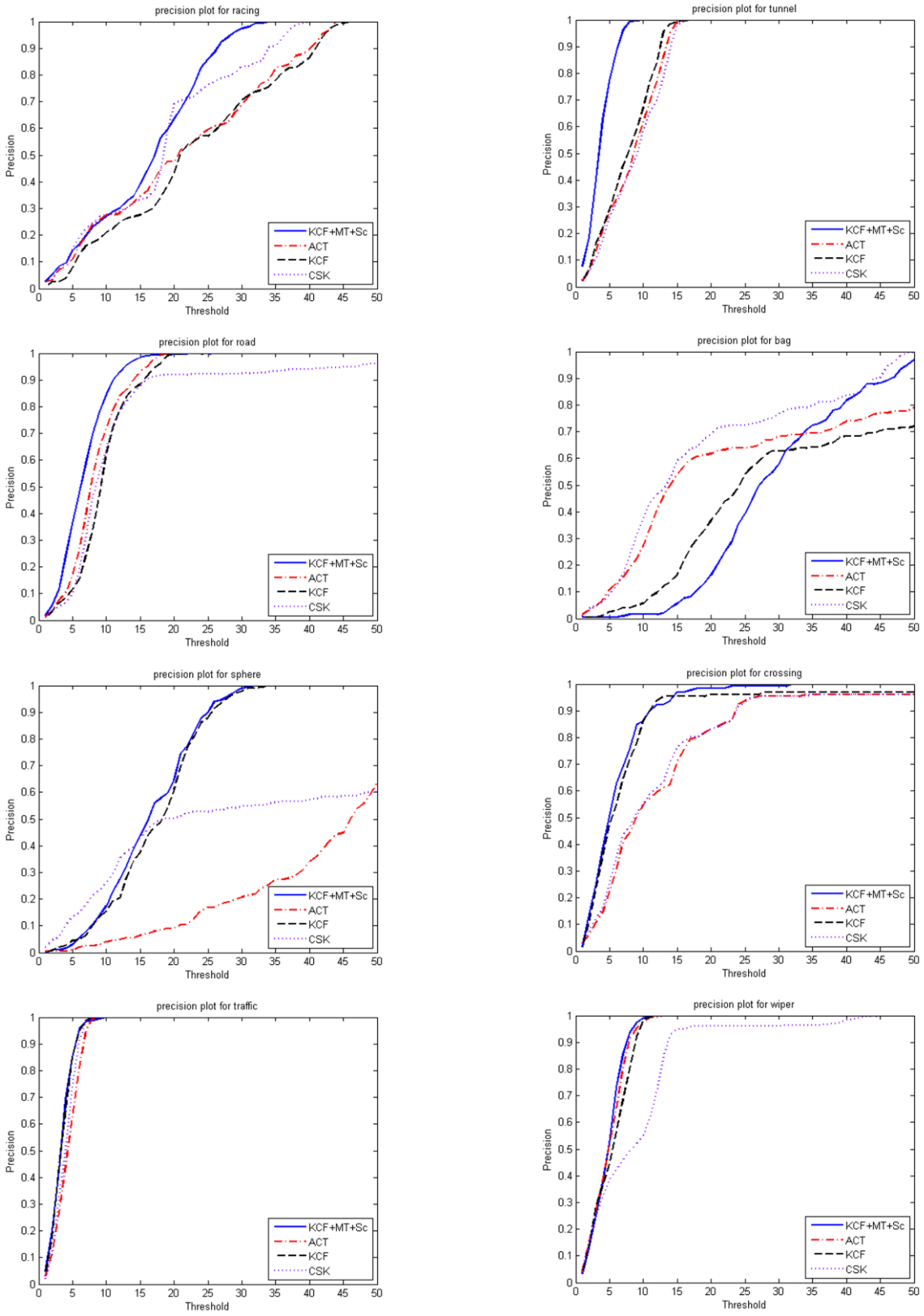

Fig. (3.b) Precision plots of eight different sequence of the VOT15 dataset