Event-Guided Structured Output Tracking of Fast-Moving Objects Using a CeleX Sensor

Jing Huang, Shizheng Wang, Menghan Guo, *Member, IEEE*, and Shoushun Chen, *Senior Member, IEEE*

***Abstract*— In this paper, we propose an event-guided support vector machine (ESVM) for tracking high-speed moving objects. Tracking fast-moving objects with low frame rate cameras is always difficult due to motion blur and large displacements. The accuracy problem can be solved by using high frame rate cameras at the expense of tremendous computational cost. For this issue, our ESVM incorporates event-based guiding methods into the traditional structured support vector machine to improve the tracking accuracy at a relatively low-complexity level. The event-based guiding methods include two models, event position guided search localization and event intensity guided sample supplement, which are based on the event features of the CeleX motion sensor. The motion sensor continuously responds to intensity change, which is generally related to object motion. Once it has detected intensity change, the motion sensor outputs event packages, and each of them contains the pixel location, time stamp, and pixel illumination. The generated events are continuous in the temporal domain and thus record the motion trajectory of fast-moving objects, which cannot be fully captured by frame-based cameras. In this paper, we convert high-speed test sequences into sequences of spiking events recorded by the CeleX motion sensor. Our approach presents fairly high computational efficiency, and experiments over sequences from multiple tracking benchmarks demonstrate the superior accuracy and real-time performance of our method, compared to the state-of-the-art trackers.**

***Index Terms*— Dynamic motion sensor, event guided, support vector machine (SVM).**

I. Introduction

OBJECT tracking is a key technology for many technical applications in areas such as robotics, surveillance, medicine, autonomous driving, automation and sensor networks. Over decades of extensive study, fast object tracking is still a challenging and computationally expensive problem.

Currently, most of the released tracking sequences in public benchmarks such as VOT2016 [1], UVA123 [2], OTB100 [3] and ALOV300 [4] are captured at 30fps, which results in motion blur for high speed moving objects. Traditional object tracking is implemented based on the concept of 'frames'. The tracking accuracy has achieved a fairly high level (60%) for moving objects with normal speed [5]; however, the accuracy drops to around 30% when tracking high speed moving objects [6]. This is to be expected because motion blur is introduced when capturing fast motion at standard frame rate. Moreover, the search area in traditional tracking methods is commonly set to be a constant, which could be too large for slow motion or too small for fast motion.

Tracking accuracy decreases if an object's displacement distance outreaches the search area as this makes it much more difficult to track fast moving objects. Nevertheless, such problems can be solved by increasing the capture rate. Galoogahi, *et al.* [6] have released a new benchmark containing sequences captured at 240fps from real world scenarios. When tracking high framerate sequences, most trackers improve the accuracy by more than 50%, and the state-of-the-art trackers are able to achieve a success rate of 60% [6]. Nonetheless, the accuracy improves at an expense of large computational cost. Compared with 30fps sequences, the input data of 240fps sequences is eight times larger. For sub-step tasks like sample searching and feature extraction, series of computationally demanding operations need to be performed on each acquired frame. Therefore, tracking high framerate sequences increases the computational cost and thus makes it a challenge to implement real-time tracking.

In this work, we propose an event-guided structured output tracking method, an Event-guided Support Vector Machine (ESVM), by using an event-based motion sensor, namely CeleX sensor [7]. The input data of ESVM consist of image sequences at a low framerate (30fps) and their corresponding high-frequency event flows captured by CeleX under the same scenarios. As shown in Fig.1, the CeleX motion sensor is not limited by the concept of 'frame' and dynamically outputs event packages, each of which contains information of pixel location, illumination and time-stamps representing when the pixels are triggered by an illumination change. Making use of the pixel location in event packages, we first propose an Event Position guided Search Localization (EPSL) method to guide the moving object localization. The EPSL is able to provide adaptive search areas and guide potential motion direction, such that it increases the search accuracy and reduces the computational cost. Furthermore, using the intensity information in event packages, an Event Intensity guided Sample Supplement (EISS) method is proposed to supplement more effective samples for online tracker learning and updating. The innovation and contribution of this paper is in proposing an event-guided structured output tracking method for real-time tracking of fast moving objects with high accuracy, while maintaining the computational cost at a low level.

The rest of the paper is structured as follows: Section II describes the related work on the event-based motion sensor and classical tracking-by-detection framework based on the Structured Support Vector Machine (SSVM). Section III details the proposed ESVM tracking algorithm, while Section IV discusses the results of experiments. Finally, conclusions are drawn in Section V.

II. Related Work

A. Event-Based Motion Sensor

传感器输出异步地址事件包，指示光强变化的检测。

The motion sensor used in this work is a CeleX sensor with 320 × 384 pixels [7], as shown in Fig.1. The sensor outputs asynchronous address event packages indicating detection of light intensity variation. The event packages are continuous in the temporal domain and correlated with motions since intensity variation is commonly caused by moving objects or moving cameras. Each pixel works independently and detects intensity in a logarithmic manner. It directly outputs logarithmic photo-receptor voltage as the pixel illumination when a pixel event is generated. Therefore, each event package contains the pixel location, the absolute intensity value and a time-stamp tagged off-chip when being read out. In addition, the sensor is able to generate full-frame pictures controlled

Manuscript received November 21, 2017; revised March 22, 2018 and May 12, 2018; accepted May 22, 2018. Date of publication May 28, 2018; date of current version September 13, 2018. This paper was recommended by Associate Editor A. Prati. *(Corresponding author: Jing Huang.)*

J. Huang, S. Wang, and M. Guo are with Nanyang Technological University, Singapore 639798 (e-mail: jhuang@ntu.edu.sg).

S. Chen is with the Department of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2018.2841516

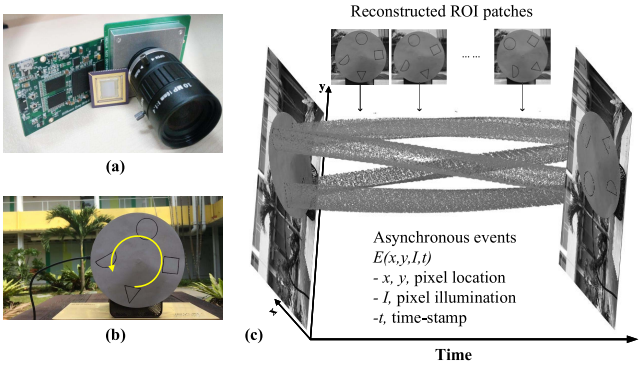

Fig. 1. Overview of CeleX motion sensor. (a) CeleX sensor with 384 × 320 pixels. (b) Rotating plate with four different geometrical patterns. (c) The spatio-temporal spacing of events generated in response to the rotating plate: pixels independently generate asynchronous event packages. Static background is captured initially and then updated by enclosing the intensities of dynamic events to it, such that small patches of region-of-interest (ROI) can be reconstructed when required.

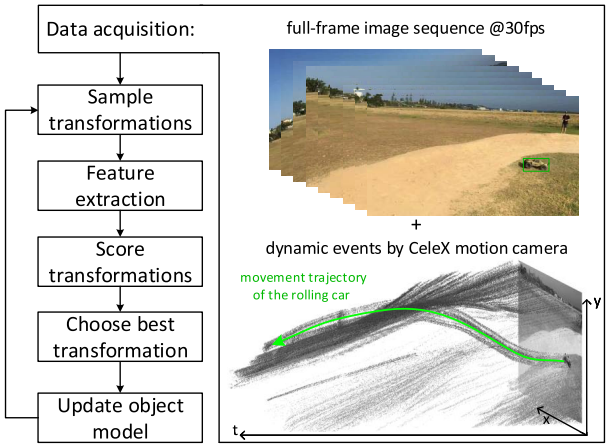

Fig. 2. General structure of the proposed ESVM, whose input data include 30fps image frames and dynamic events captured by the CeleX, taking the *car_rc_rolling* sequence as an example.

by an external signal. Considering that the generated events are sparse in spatial domain, the static background is captured initially and then updated by enclosing the intensities of dynamic events. The derived up-to-date full image guarantees that all pixel values in each reconstructed sample patch are valid, as shown in Fig.1, which illustrates an example of the spatial-temporal asynchronous events with intensities and the possible reconstructed sample patches. The temporal resolution of the events can achieve a value as high as 200Meps [8], which is equivalent to a 1000fps high speed camera.

B. Classical Tracking-by-Detection Framework

In the traditional adaptive tracking-by-detection algorithms, a classifier is learned online to distinguish a target object from its background shown in Fig.3 (a). During tracking, it is assumed that a change in position of the target can be estimated by maximizing the scoring function $F(X_t^{P_t})$, which represents the similarity between candidate patches on the feature map X of frame t, $X_t^{P_{t-1}\circ y}$, and the target tracking object in position P_{t-1} in frame $t-1$, $X_{t-1}^{P_{t-1}}$. The objective for the tracker is to estimate a transformation (e.g. translation, rotation) $y_t \in Y$ such that the new position of the object is approximated by the transformation $P_t = P_{t-1} \circ y_t$. Y denotes our

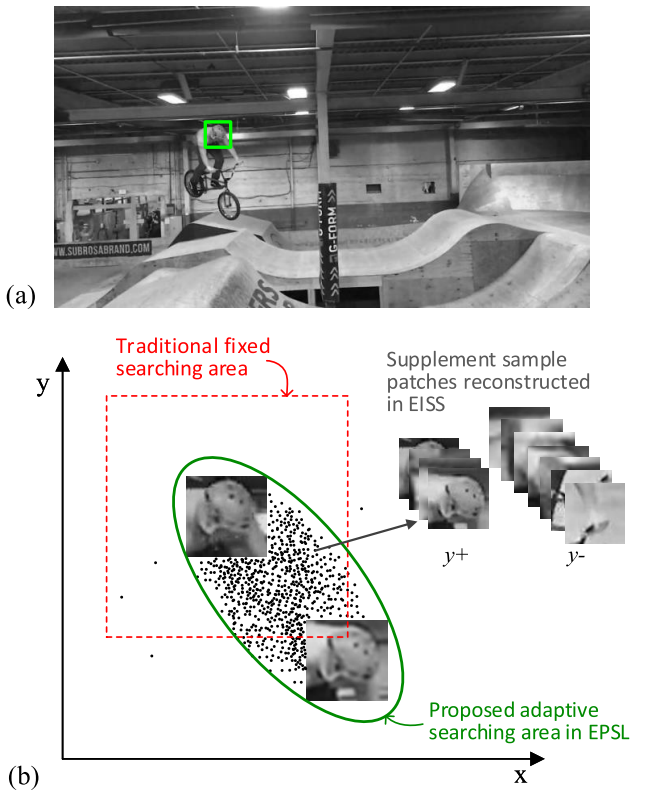

Fig. 3. Demonstration of the proposed tracking method using ESVM: (a) Sample image from biker head sequence with the bonding box of ROI. (b) Illustration of a fast moving object tracking process, including demonstration of EPSL and EISS.

search space and its form depends on the type of transformation to be tracked. For most tracking-by-detection approaches, this transformation is a 2D translation, in which case $Y = (u, v) \mid (u^2 + v^2 < r^2)$, where r is a search radius.

Taking a classical structured output tracker, Struck [10], as an example, it directly estimates the transformation y_t between consecutive images by a prediction function f, which is based on the scoring function F. In this way, the transformation y_t from P_{t-1} to P_t can be predicted according to:

$$y_t = f(X_t^{P_{t-1}}) = \arg\max_{y \in Y} F(X_t^{P_{t-1}\circ y}), \tag{1}$$

The SSVM solver [11], [12] has been widely used in tracking due to its outstanding performance. However, from the problem formulation, there are at least two limitations in the traditional sample transformation module. Firstly, the search radius of Y is generally set to a relatively large constant resulting in unnecessary computational cost and inaccurate tracking results. Secondly, sample updating is insufficient because it loses the object feature between X_{t-1} and X_t due to fast motion, which is the reason why trackers show better performance on video sequences with higher frame rate, e.g. 240fps [6].

III. Proposed Algorithm

In order to solve the above-mentioned problems, we propose ESVM to improve the sample transformation module for high speed moving object tracking. The CeleX and the proposed event-guiding methods support and guide a classical tracking-by-detection approach. This paper only focuses on the ideal sample transformation that the position parameter vector P is considered as a 2D bounding

box with fixed size, and the transformation y only includes the 2D translation without additional components, like scale, rotation or shape. In this work, the SSVM tracker is selected because it, as a single layer Neural network algorithm, is able to achieve a processing speed of 150fps after optimization, which is faster than most of the deep learning based object trackers. What is more, the SSVM algorithm is much easier for future hardware implementation than multi-layer neural network algorithms.

Fig. 2 shows the general structure of the proposed ESVM method, whose input data is a combination of image sequences at a low framerate (30fps) and their corresponding dynamic event flows captured by the CeleX sensor under the same scenarios. The general architecture of ESVM follows the framework of traditional SSVM [11]. Note that, the event flows used in the tracking process are to support a standard approach tracking the 30fps image sequences through two event-guiding methods, EPSL and EISS, as shown in Fig. 3.

The first event-guiding method, EPSL, is proposed to generate an adaptively guided search space of transformations Y_G by using the location information of the dynamic events. The main purpose of this proposal is to provide additional information of the object's potential movement region. Traditional frame-based sensors cannot capture the object motion between two consecutive frames, while the dynamic detection feature of our motion sensor enables recording of object motion by continuously generating events in the temporal domain, as shown in Fig. 1. It is notable that the trigger condition of the pixel events is the over-threshold illumination changes regardless of foreground object or background. In practice, especially when camera is moving, events are generated due to the movements of both object and background. To identify the events that relate to the object, given the previous object location P_{t-1}, the standard SSVM is firstly applied using Eq.(1) to get the predicted location $P_t^{Pre} = P_{t-1} \circ y_t^{Pre}$, where y_t^{Pre} is the predicted transformation. Making use of the hypothesis of movement continuity in the temporal domain, the potential object motion range is determined through connected component labeling based on P_{t-1}, P_t^{Pre} and the events generated during time interval from $t-1$ to t. The events contained in the connected components are marked as $E_{[t-1,t]}$ and their location set is $L_{[t-1,t]}$.

Therefore, as defined in Eq.(2), the location P_{t-1} after a transformation from the guided search region, $y \in Y_G$, locates within an adaptive region that is defined as the union of three location regions: the previous object location P_{t-1}, the predicted location P_t^{Pre} in frame t calculated by the traditional SSVM, and the event location region $L_{[t-1,t]}$.

$$P_{t-1} \circ y \in \{P_{t-1} \cup P_t^{Pre} \cup L_{[t-1,t]}\}, \ \ y \in Y_G, \tag{2}$$

The second event-guided method, EISS, is proposed to produce supplementary moving object sample patches making use of the intensity information of events. The supplementary samples are reconstructed from the events $E_{[t-1,t]}$, as shown in Fig. 3(b), and contribute to the supplement of additional support vectors to handle the fast updating of object representation. As shown in Fig. 1, the event packages generated by the motion sensor contain not only pixel locations, but also their intensities. Benefitting from the motion sensor's high temporal resolution, users can acquire an intermediate frame at a temporal precision Δt as accurate as 50ns. That means because events are continuous with 50ns accuracy in the temporal domain, any intermediate frames t' can be reconstructed between frame $t-1$ and t using the event intensities to acquire a higher sampling rate. Once Δt has been set, $(T_t - T_{t-1})/\Delta t$ frames could be reconstructed and injected into the original sequence of frames according to the required temporal precision.

The injected frames are used to generate the (event-) guided supplementary samples (GSS) $X_{t'}^{P_{t'}}$ and update the tracker model by selecting more support vectors with respect to the appearance change of tracked moving object from time T_{t-1} to T_t. Following the design of traditional SSVM [11], the candidate positive and negative support vectors are selected using Eq.(3) and Eq.(4) from the feature vectors $X_{t'}$, which are extracted from frame $I_{t'}$.

$$\mathrm{y}_+ = \arg\min_{y \in Y} \{\Delta(y, \overline{y}) + F(X_{t'}^{P_{t-1} \circ y})\}, \tag{3}$$

$$\mathrm{y}_- = \arg\max_{y \in Y} \{\Delta(y, \overline{y}) + F(X_{t'}^{P_{t-1} \circ y})\}. \tag{4}$$

Here, $\Delta(y, \overline{y})$ is the loss function regulated to [0, 1] for punishing the position shift from the real object position $p^{\overline{y}}$ to the sample position p^y. It will be set to 0, if the two positions are totally the same; and set to 1 if there is no overlap between the two bonding boxes at these positions. $F(X_t, y)$ is the scoring function used in Eq.(1), which calculates the score related to the similarity between the candidate sample $X_{t'}^{P_{t-1} \circ y}$ and the tracked sample in the previous frame $t-1$.

Combining both EPSL and EISS, the proposed method ESVM, calculates the ultimate tracking result P_t^{fin} based on the proposed adaptive search space and supplement support vectors. The detailed algorithm is shown as Algorithm 1.

Algorithm 1

input: P_{t-1}, object location in previous frame $t-1$
$(X_{t-1}, y) \in S_{t-1}$, samples within the maintained set of support vectors after processing frame $t-1$
X_t, training sample in current frame t

Event Position guided Search Localization
1. $y_t^{Pre} = \arg\max_{y \in Y} F(X_t^{P_{t-1} \circ y})$
2. $P_t^{Pre} = P_{t-1} \circ y_t^{Pre}$
3. $Y_G = \{P_{t-1} \cup P_t^{Pre} \cup L_{[t-1,t]}\}$

Event Intensity guided Sample Supplement
4. for $j = 1$ to $\frac{t-(t-1)}{\Delta t}$
5. $\quad t' = t - 1 + \Delta t \cdot j$
6. $\quad y_{t'} = \arg\max_{y \in Y_G} F(X_{t'}^{P_{t-1} \circ y})$
7. $\quad P_{t'} = P_{t-1} \circ y_{t'}$
8. $\quad (t, y_+, y_-) \leftarrow GSS(X_{t'}^{P_{t'}}, y_{t'})$
9. $\quad$ if $(t' = t)$ $P_t^{fin} = P_{t'}$
10. end for
11. Optimize()
12. Return P_t^{fin}, S_t

IV. Experimental Results

The experiments are designed to validate that the proposed ESVM algorithm considerably improves the comprehensive performance of tracking accuracy and efficiency with respect to fast moving objects. The test sequences are selected from the high framerate (240fps) NFS benchmark [6] and 30fps OTB [3]. High framerate sequences can closely approximate the continuous motions in the real world and enable the motion sensor to generate supplementary events between frames at 30Hz. Among the sequences in NFS, the *biker_head* and *car_rc_rolling* sequences contain fast moving objects and complex background. The *bolt* sequence from OTB [3], which challenges traditional tracking algorithms, is selected to analyze the contributions of the event-guiding methods on the standard SSVM tracker.

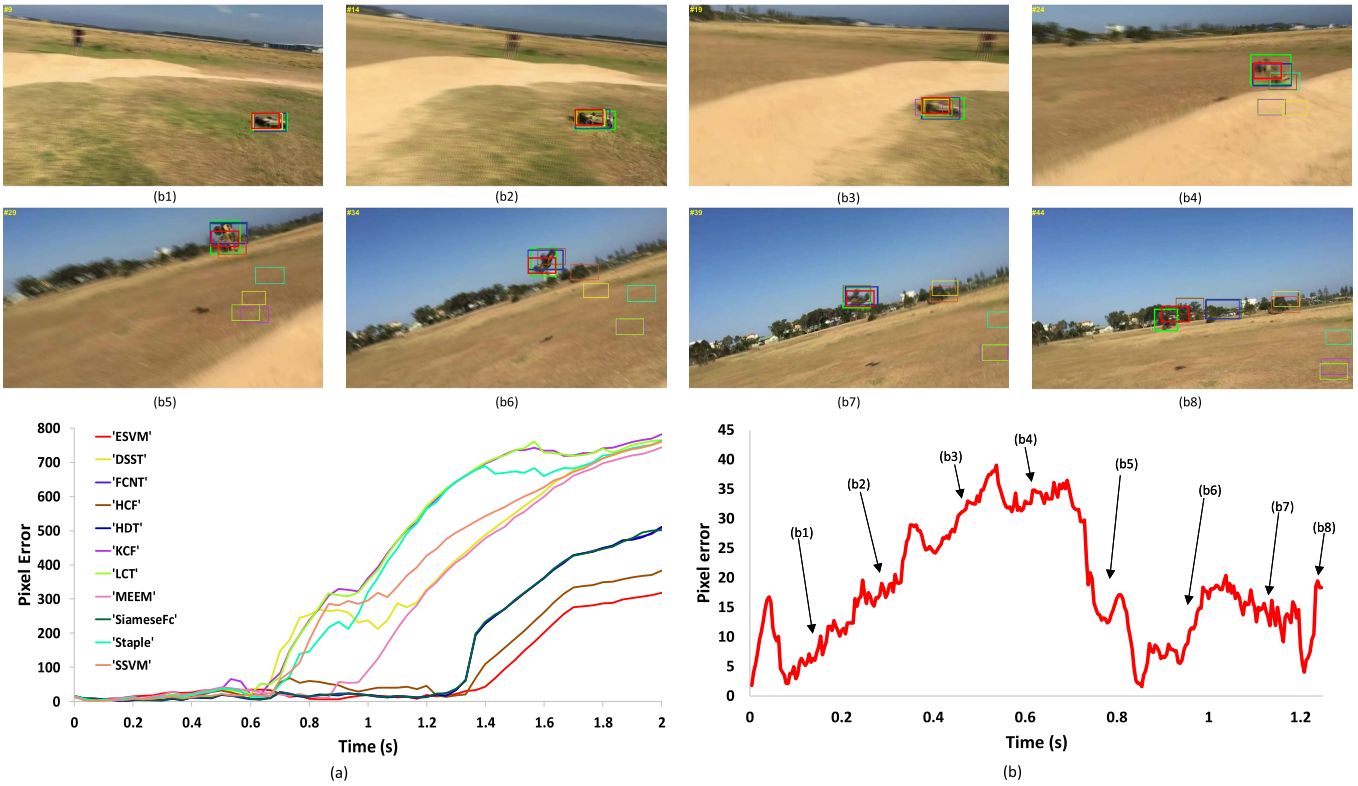

Fig. 4. Tracking performance. (a) Tracking errors for both ESVM and state-of-the-art algorithms on *car_rc_rolling* sequence at 30fps. (b) Tracking errors of ESVM for the first 1.2s and (b1-b8) are sample frames showing tracking performance of all listed trackers, the color of bonding boxes are consistent with the legend in (a).

We compare the tracking results of ESVM with those of nine state-of-the-art baseline trackers, i.e. HDT [13], SiameseFc [14], Staple [15], FCNT [16], HCF [17], KCF [18], LCT [19], MEEM [20] and DSST [21]. Additionally, in order to research on the effectiveness of EPSL and EISS models in the proposed ESVM method, we also conducted experiments using standard SSVM. The method SSVM with only EPSL is referred to as ESVM- and ESVM refers to SSVM with both EPSL and EISS.

Events for *bolt* are from the DVS VOT benchmark [9], while the events for NFS sequences are generated using our motion sensor [7]. The event recording setup employs the method in [9], where original 240fps video sequences are displayed using a consumer-grade DELL U2414H LED monitor with a refresh rate of 60 Hz and the native resolution of 1920 × 1080. To track NFS sequences at 30fps, event packages generated by every seven frames in eight of 240fps sequences are used to reconstruct supplemental samples. However, the frame rate of *bolt* is originally 30fps, such that no event package can be generated between two consecutive frames. Therefore, there is only the ESVM- result for *bolt*.

A. Location Precision

Table I lists the tracking results of object location precision for the proposed ESVM tracker and the baseline trackers mentioned in the NFS benchmark. For the *car_rc_rolling* 30fps sequence, ESVM ranks the first with respect to location accuracy followed by the ESVM- method. The average location error (ALE) using ESVM is 83.321 that is 25 pixels smaller than the best baseline tracker HCF, whose ALE is 108.525. For the *biker_head* 30fps sequence, ESVM performs better than most baseline trackers and is comparable to the first with a minor difference of 0.12 pixel. With the help of EISS, ESVM tracking results for 240fps sequences are used as reference. By comparing the ALE of trackers on 30fps and 240fps sequences, the accuracy of ESVM tracker on 30fps is as good as other trackers' performance on 240fps. Therefore, the proposed tracking algorithm ESVM, which employs event-based motion sensor to guide object tracking in low frame rate videos, demonstrates the state-of-the-art performance. Moreover, the results show that ESVM- remarkably improves the tracking accuracy compared with traditional SSVM, while ESVM performs even better. The location precision plot of the *car_rc_rolling* sequence is demonstrated in Fig. 4(a), which indicates that the proposed ESVM tracker outperforms the other trackers in tracking fast moving objects. For the *car_rc_rolling* sequence, traditional

TABLE I

AVERAGE LOCATION ERROR (ALE) COMPARISON ON TRACKING SCENARIOS INCLUDING HIGHER FRAMERATE TRACKING (240fps) AND LOWER FRAMERATE TRACKING (30fps)

	biker_head		*car_rc_rolling*		*bolt*
	30fps	**240fps**	**30fps**	**240fps**	**30fps**
HDT	442.8563	25.9101	135.5448	66.9254	5.0181
SiameseFc	431.3484	20.0044	136.4567	67.3311	125.17
Staple	**3.7215**	3.8445	351.9668	143.129	**4.136**
FCNT	24.9628	18.1085	136.0955	67.6556	7.2892
HCF	5.2031	3.9494	108.5254	**20.2911**	6.8574
KCF	500.3417	4.9028	379.3422	375.525	6.7622
LCT	357.4497	4.9020	377.4301	103.915	5.2233
MEEM	4.2633	**3.4865**	266.3398	102.105	15.5379
DSST	62.0206	3.5718	307.9312	119.366	4.5785
ESVM	3.8400	3.8559	**83.3213**	80.9876	-
ESVM-	4.3060	-	104.9961	-	5.4555
SSVM	428.5034	-	320.9378	-	370.206

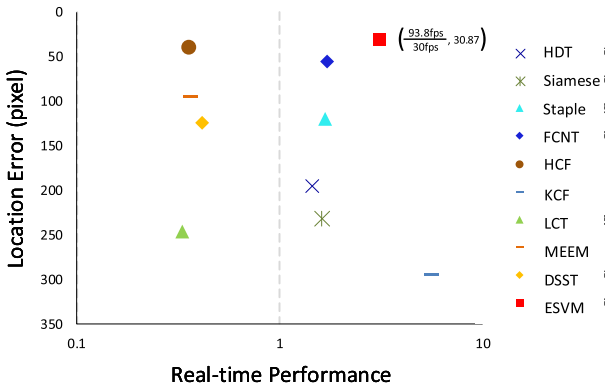

Fig. 5. Plot of location error and real-time performance.

trackers lose the object due to large displacement, object deformation or background complexity. As observed, half of the anchor trackers lose the target during 0.6-0.7s, where large displacements occur. In contrast, with the increasing motion of the car, EISS method supplements more additional samples, so that more support vectors would be generated and contribute to distinguishing foreground from background and tracking target object. Fig. 4(b) shows the location error on each frame using ESVM and Fig.4(b1-b8) are eight sample frames with bonding boxes of all trackers.

B. Computational Cost

For object tracking, real-time performance is always a big challenge. In this work, the machine we used runs at 3.6 GHz with an NVIDIA GeForce GTX Titan Black GPU, and the system is under Ubuntu 14.04. For the traditional SSVM tracker, we use the default parameters including Haar features, a Gaussian kernel with $\sigma = 0.2$ and a budget of 100 support vectors. Compared to the speed of other baseline trackers, the processing speed of ESVM, which reaches 93.8fps, is much faster than most of the others [6], as shown in Fig.5.

C. Accuracy and Real-Time Performance

When taking both accuracy and processing speed into consideration, ESVM ranks first as shown in Fig.5. The location error for each tracker is the average value of ALE on all 30fps sequences. Performance value is computed by dividing the processing speed by 30fps. That is, value larger than 1 indicates that trackers have potential to process low frame-rate sequences in real-time. Obviously, ESVM ranks first because it performs the best at balancing accuracy and real-time performance. In contrast, KCF is the fastest tracker; however, its accuracy is not satisfactory.

V. Conclusion

In this work, we propose two event-based guiding methods, EPSL and EISS, incorporated with a traditional tracking-by-detection algorithm, SSVM, for the task of visual tracking. Different to the classical trackers, the proposed ESVM exploits an online adaptive search area to achieve more accurate localization. Moreover, we utilize the gray-level intensities of event packages generated by our motion sensor to reconstruct supplement samples and update the tracker model with respect to the new appearance of the target over time. Compared to the state-of-the-art trackers, ours has demonstrated the competitive accuracy and superior real-time performance over a comprehensive evaluation. Applications on real-word scenarios with event data recorded in real time and additional tracking conditions, such as rotation and scale transformations, will be explored in our future work.

References

[1] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 639–651.
[2] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 445–461.
[3] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.
[4] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.
[5] M. Kristan *et al.*, "The visual object tracking VOT2016 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Jul. 2016, pp. 777–823.
[6] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, "Need for speed: A benchmark for higher frame rate object tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1134–1143.
[7] J. Huang, M. Guo, and S. Chen, "A dynamic vision sensor with direct logarithmic output and full-frame picture-on-demand," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2017, pp. 1–4.
[8] M. Guo, J. Huang, and S. Chen, "Live demonstration: A 768×640 pixels 200 Meps dynamic vision sensor," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2017, p. 1.
[9] Y. Hu, H. Liu, M. Pfeiffer, and T. Delbruck, "DVS benchmark datasets for object tracking, action recognition, and object recognition," *Frontiers Neurosci.*, vol. 10, p. 405, Aug. 2016.
[10] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 263–270.
[11] S. Hare *et al.*, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.
[12] S. Zhang, Y. Sui, S. Zhao, and L. Zhang, "Graph-regularized structured support vector machine for object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 6, pp. 1249–1262, Jun. 2017.
[13] Y. Qi *et al.*, "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4303–4311.
[14] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 850–865.
[15] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1401–1409.
[16] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3119–3127.
[17] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3074–3082.
[18] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
[19] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5388–5396.
[20] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.
[21] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, BMVA Press, 2014.