Supplementary Material
Need for Speed: A Benchmark for Higher Frame Rate Object Tracking

Hamed Kiani Galoogahi[1*], Ashton Fagg[2*], Chen Huang[1], Deva Ramanan[1] and Simon Lucey[1]
[1]Robotics Institute, Carnegie Mellon University
[2]SAIVT Lab, Queensland University of Technology
{hamedk,chenh2}@andrew.cmu.edu ashton@fagg.id.au {deva,slucey}@cs.cmu.edu

1. Introduction

This supplementary material provides additional details of the Need for Speed (NfS) dataset. The full dataset and benchmark, including all the videos, frames, annotations, Gyro and IMU raw data, evaluation codes and results (in mat files and video demos) are publicly available at http://ci2cv.net/nfs/index.html.

Frame samples of NfS: The NfS dataset consists of 100 videos. The first frame of each video is shown in Fig. 1 and Fig. 2. The target of interest is highlighted by a blue bounding box.

Per tracker evaluation: Fig. 3 compares tracking higher versus lower frame rate videos (success plots) for each evaluated method. These results are summarized by Fig. 3 (success rate at IoU > 0.50) in the main manuscript. Here, we illustrate success plots of all tracker over all overlapping thresholds. For lower frame rate tracking (30 FPS) results are reported for both with and without motion blur. AUCs are reported in the legend. This more detailed evaluation shows that all trackers achieve a significant improvement on tracking higher frame rate videos, compared to lower frame rate videos. Moreover, this evaluation shows that all trackers are fairly robust to the presence of motion blur in lower frame rate videos.

Attribute description: All 9 attributes annotated in NfS are described in Table. 1. For attribute based evaluation, please see the main manuscript.

Evaluated methods: All methods evaluated in the main manuscript are summarized in Table. 2.

Updated learning rates: Here, we mathematically show why we selected to update learning rate as $LR_{new} = \frac{1}{8}LR_{old}$ for higher frame tracking (240 FPS).

Online adaptation in all CF tracker are, generally, performed by updating current visual model of the target at frame $f+1$ as $x^{f+1}_{model} = x^{f}_{model} + \eta x^{f+1}$, where η is the learning rate, x^{f+1}_{model} is the updated appearance model,

Table 1. Attributes and their detailed description.

Attr	Description
IV	Illumination Variation - the illumination in the target region changes significantly.
SV	Scale Variation - the ratio of the bounding boxes of the first frame and the current frame is out of the range [1/ts, ts], ts >1 (ts=2).
OCC	Occlusion - the target is partially or fully occluded.
DEF	Deformation - non-rigid object deformation.
FM	Fast Motion - the motion of the ground truth is larger than tm pixels (tm=20)[1].
VC	Viewpoint Change - viewpoint change caused by in-plane rotation, out-plane rotation and camera movement changes target appearance significantly.
OV	Out-of-View - the target is partially or fully out of the view.
BC	Background Clutters - the target and its surrounding background share similar color or texture.
LR	Low Resolution - the number of pixels inside the ground-truth bounding box is less than 400.

x^{f}_{model} is the appearance model at the previous frame, x^{f+1} is the object appearance (*e.g.* HOG features) in current frame (i+1), and $f = 0, ..., F-1$ [3, 11, 6]. F is the number of frames. x^0_{model} for the first frame is initialized as $x^0_{model} = x^0$. The adaptation formulation at frame $f+1$ can be expended as:

$$x^{f+1}_{model} = x^0 + \eta \sum_{i=1}^{f+1} x^i \quad (1)$$

The number of frames over a fixed period of time (*e.g.* 1 second) at 240 FPS videos is 8 times more than that at 30 FPS videos. Thus, to retain the amount of visual information used to update the model in higher frame rate videos to be same as that in lower frame rate videos, we (approximately) need to divide the learning rate by 8, meaning that $\eta_{new} = \frac{1}{8}\eta_{old}$.

Figure 1. Sample frames of NfS videos

References

[1] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, June 2016. 5

[2] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, pages 850–865, 2016. 5

[3] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, pages 2544–2550. IEEE, 2010. 1

[4] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014. 5

[5] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, pages 4310–4318, 2015. 5

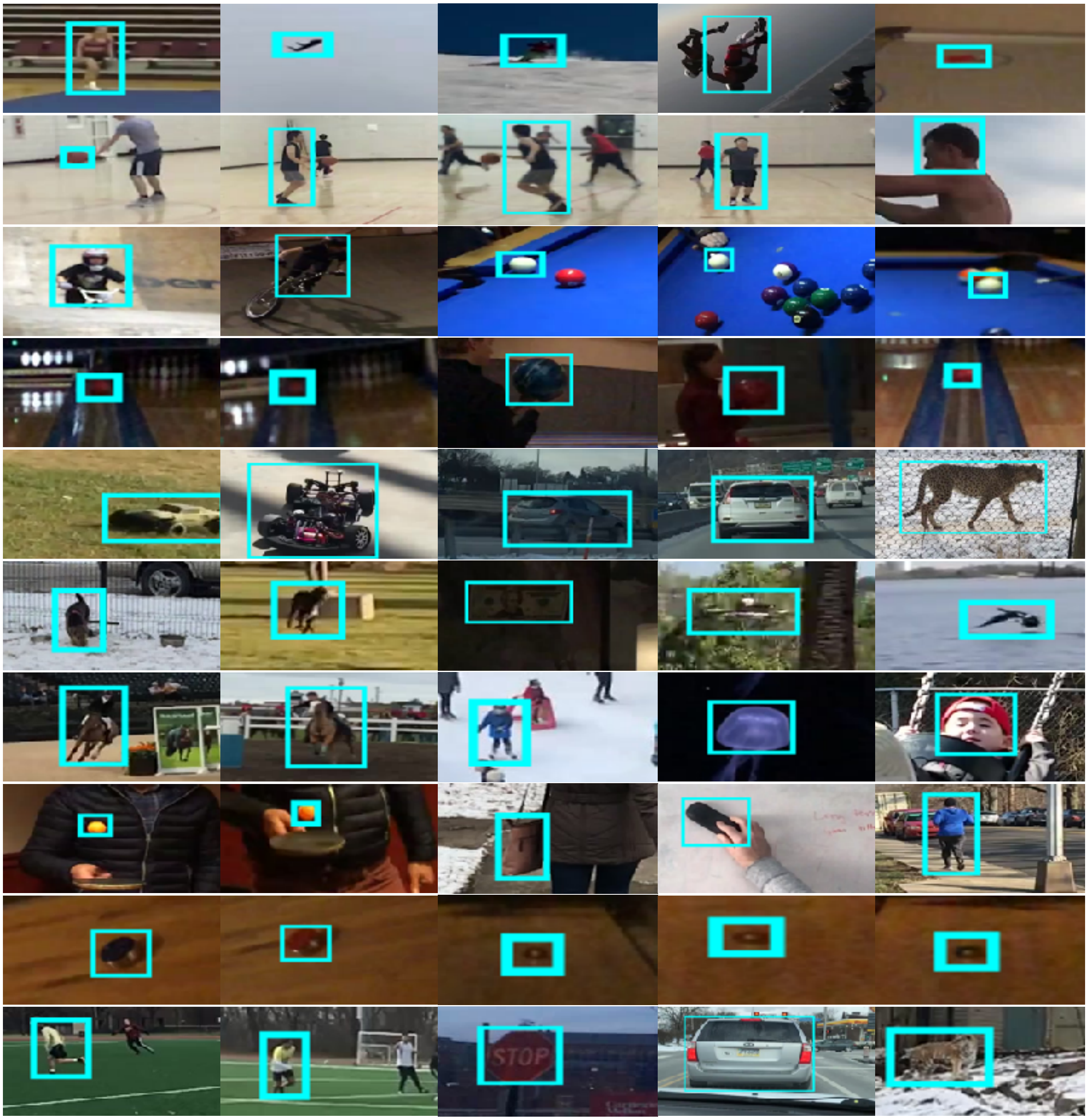

Figure 2. Sample frames of NfS videos - cont.

[6] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, pages 472–488, 2016. 1

[7] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, 2016. 5

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *PAMI*, 37(3):583–596, 2015. 5

[9] H. Kiani Galoogahi, A. Fagg, and S. Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, 2017. 5

[10] H. Kiani Galoogahi, T. Sim, and S. Lucey. Multi-channel correlation filters. In *ICCV*, pages 3072–3079, 2013. 5

[11] H. Kiani Galoogahi, T. Sim, and S. Lucey. Correlation filters with limited boundaries. In *CVPR*, pages 4630–4638, 2015.

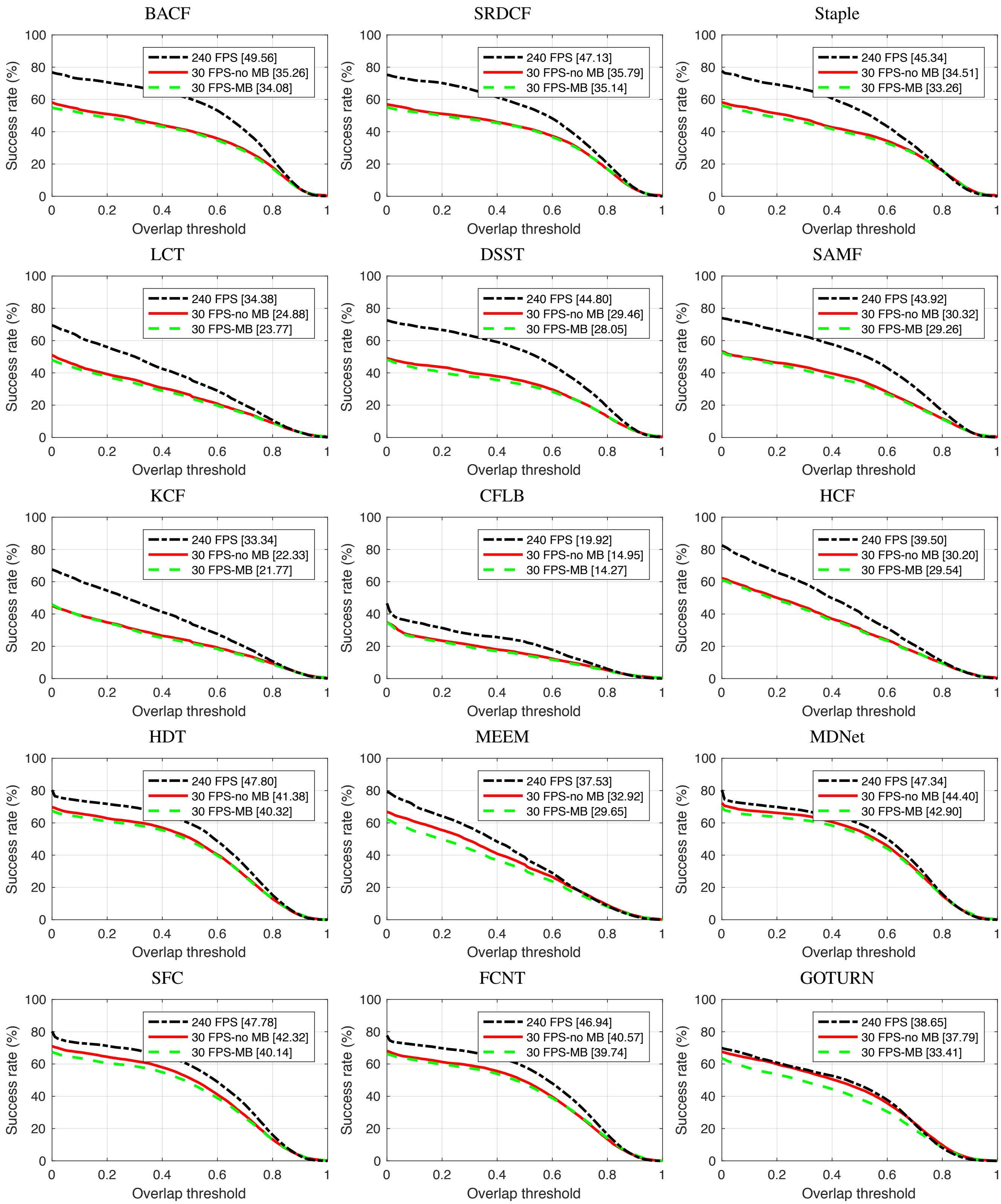

Figure 3. Comparing higher frame rate tracking (240 FPS) versus lower frame rate tracking (30 FPS) for each tracker. For higher frame rate tracking CF trackers are performed by updated learning rate. The results of lower frame rate tracking are plotted for both videos with and without motion blur (30 FPS-MB and 30 FPS- no MB). AUCs are reported in brackets.

Table 2. Evaluated methods.

Tracker	**Learning**	**Feature**
BACF [9]	CF	HOG
SRDCF [5]	CF	HOG
Staple [1]	CF + colo scores	HOG + color
LCT [14]	CF + random ferns	HOG
DSST [4]	CF	HOG
SAMF [12]	CF	HOG + Color Names
KCF [8]	CF	HOG
CFLB [10]	CF	pixel values
HCF [13]	CF	deep feature
HDT [16]	CF + Hedge Algo.	deep feature
MEEM [18]	SVM	color
MDNet [15]	CNN	deep feature
SiameseFc [2]	CNN	deep feature
FCNT [17]	CNN	deep feature
GOTURN [7]	CNN	deep feature

1

[12] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV*, pages 254–265, 2014. 5

[13] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *CVPR*, pages 3074–3082, 2015. 5

[14] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. Long-term correlation tracking. In *CVPR*, pages 5388–5396, 2015. 5

[15] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. 2016. 5

[16] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang. Hedged deep tracking. In *CVPR*, pages 4303–4311, 2016. 5

[17] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *ICCV*, pages 3119–3127, 2015. 5

[18] J. Zhang, S. Ma, and S. Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *ECCV*, pages 188–203, 2014. 5