A Benchmark and Simulator for UAV Tracking -Supplementary Material-

Matthias Mueller, Neil Smith, and Bernard Ghanem

King Abdullah University of Science and Technology (KAUST), Saudi Arabia
{matthias.mueller.2, neil.smith, bernard.ghanem}@kaust.edu.sa

1 Benchmark

Table 1 summarizes the selected trackers along with their representation scheme, search method, runtime, and a generic description.

Table 1. Evaluated Algorithms. Representation: PI - Pixel Intensity, HOG - Histogram of Oriented Gradients, CN - Color Names, GK - Gaussian Kernel, K - Keypoints, BP - Binary Pattern, SVM - Support Vector Machine. Search: PF - Particle Filter, RS - Random Sampling, DS - Dense Sampling.

	Representation	Search	FPS	Description
ASLA [1]	Sparse	PF	1.4	Adaptive structural local sparse appearance model, CVPR'12
CSK [2]	PI, GK	DS	400	Single-channel Kernelized Correlation Filter, ECCV'12
DCF [3]	HOG	DS	457	Dual Correlation Filter PAMI'15
DSST [4]	PCA-HOG, I	DS	35.3	Accurate Scale Estimation (Winner VOT2014)
IVT [5]	PCA	PF	11.0	Incremental Learning, IJCVIP'08
KCF [3]	HOG, GK	DS	296	Multi-channel Kernelized Correlation Filter, PAMI'15
MEEM [6]	SVM	RS	7.7	Multiple Experts using Entropy Minimization, ECCV'14
MOSSE [7]	PI	DS	512	Minimum Output Sum of Squared Error Correlation Filter, CVPR'10
MUSTER [8]	HOG, CN, GK, KP	DS	0.9	MUlti-Store Tracker, CVPR2015
OAB [9]	BP, Haar, HOG	RS	4.8	On-line Boosting, BMVA'06
SAMF [10]	PI, HOG, CN, GK	DS	5.3	Scale Adaptive Kernel Correlation Filter, ECCV'14 Workshop
SRDCF [11]	HOG	DS	5.2	Spatially Regularized Correlation Filter, ICCV'15 (Winner VOT-TIR2015)
Struck [12]	SVM, Haar	RS	14.7	Structured output tracking with kernels, ICCV'11
TLD [13]	BP	RS	9.1	Tracking-Learning-Detection, PAMI'11

1.1 Per-video performance.

Figures 1 and 2 show the performance in terms of precision and overlap per video for UAV123 and OTB100 respectively. The scores are displayed as a color gradient map where red corresponds to 0 and dark green to 1. The score of the best performing tracker per video is shown in the last row and the average across all videos per tracker is shown in the last column. As can be clearly seen, in OTB100 most videos have at least one tracker that performs well there exist many sequences in UAV123, where none of the trackers are successful.

1.2 Robustness performance.

Figure 3 shows the spatial robustness evaluation (SRE) which tests the sensitivity to shift/scale changes in the target. Each tracker processes approximately 1.3 million frames. The results are consistent with the OPE plots (refer to paper) and trackers rank similarly with overall lower scores. This indicates that most trackers have some robustness to noise in the initialization and degrade in performance in a similar fashion.

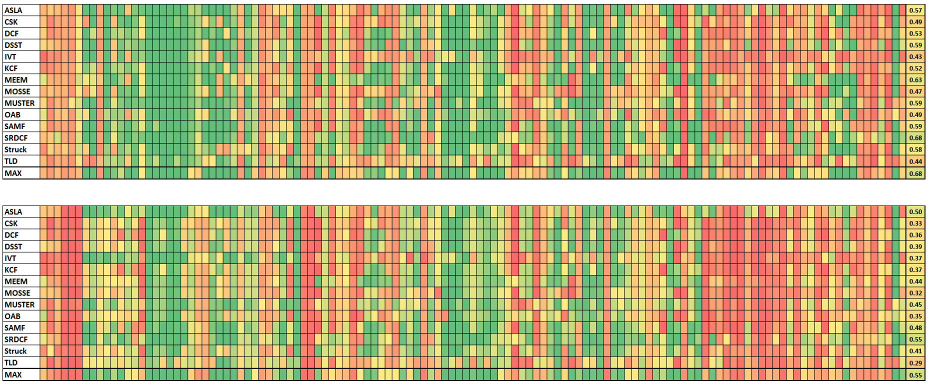

Fig. 1. OPE Precision (top) and Success (bottom) per video for UAV123.

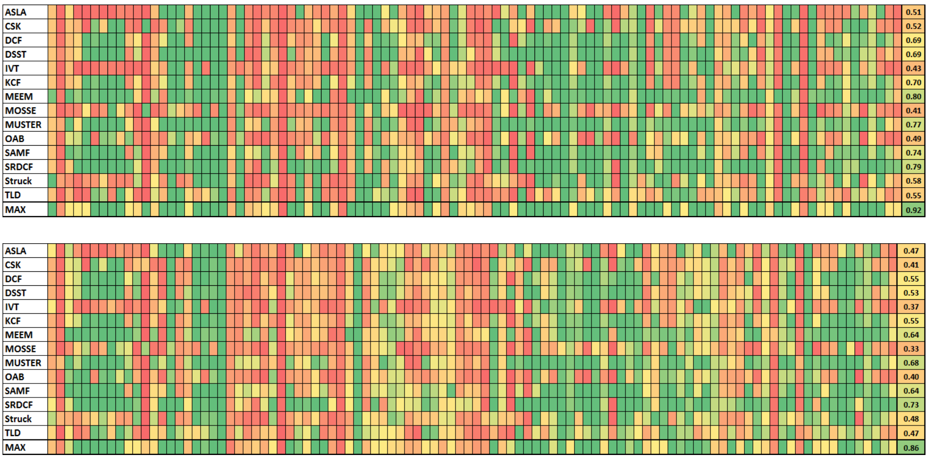

Fig. 2. OPE Precision (top) and Success (bottom) per video for OTB100.

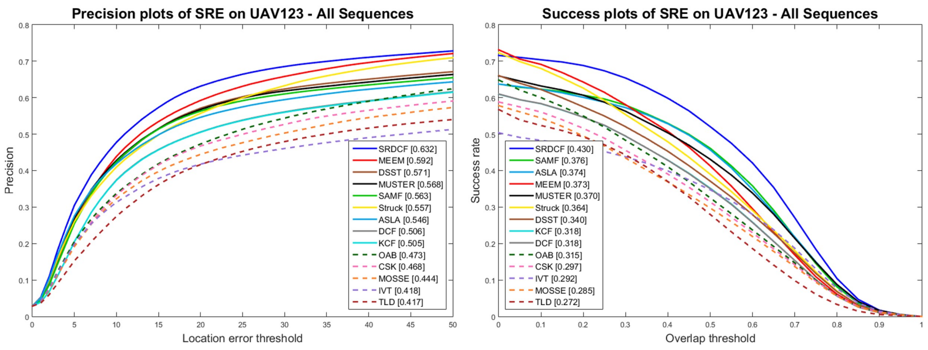

Fig. 3. Precision and success plots for SRE on UAV123.

1.3 Attribute performance.

Figures 4 and 5 show the performance of trackers on UAV123 per attribute.

Scale Variation. In the UAV123 dataset, 89% of videos have scale variation so results are similar to overall performance. Trackers that are scale adaptive perform better than trackers that are not.

Aspect Ratio Change. In the proposed benchmark, aspect ratio change (ARC) represents 55% of the video while OTB100. This is a dramatic increase as expected since the videos are captured from UAVs that must constantly change camera pitch and yaw to follow an object. The ranking of the trackers remains similar but performance degrades significantly in comparison to the overall OPE plots (e.g. by 15% for SRDCF). In particular, videos with rapid ARC lead to trackers losing the object completely.

Low Resolution. The performance in the success plots for low resolution performance significantly degrades and rankings of the trackers changes slightly (e.g. ASLA becomes the top performer). SRDCF degrades significantly by 33%. Note that trackers (e.g. STRUCK) that operate on pixel-wise features perform better on low resolution sequences than trackers that operate on histograms of gradient or color with a cell size of more than one pixel.

Fast Motion. There are a lot less sequences in the aerial dataset because FM is generally due to camera motion. We have less continuous fast motion because the UAV is keeping up with the object. However, we do have more abrupt fast motion when the UAV makes sharp turns or stops. Again when fast motion is isolated as an attribute the ranking remains fairly similar with a percentage decrease (e.g. 27% decrease by SRDCF) since current benchmarks are pre-recorded scenes.

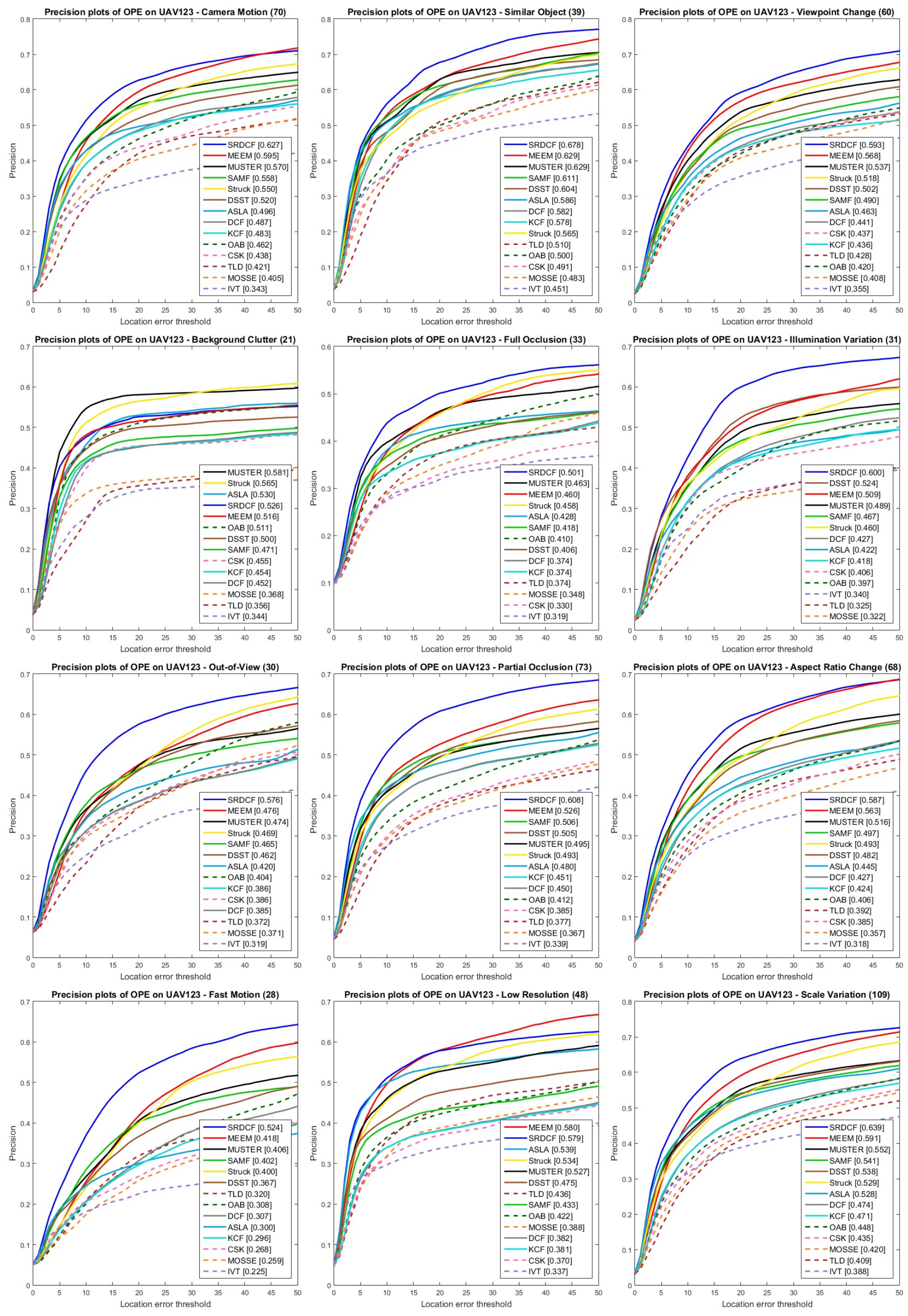

Fig. 4. OPE precision plots by attribute.

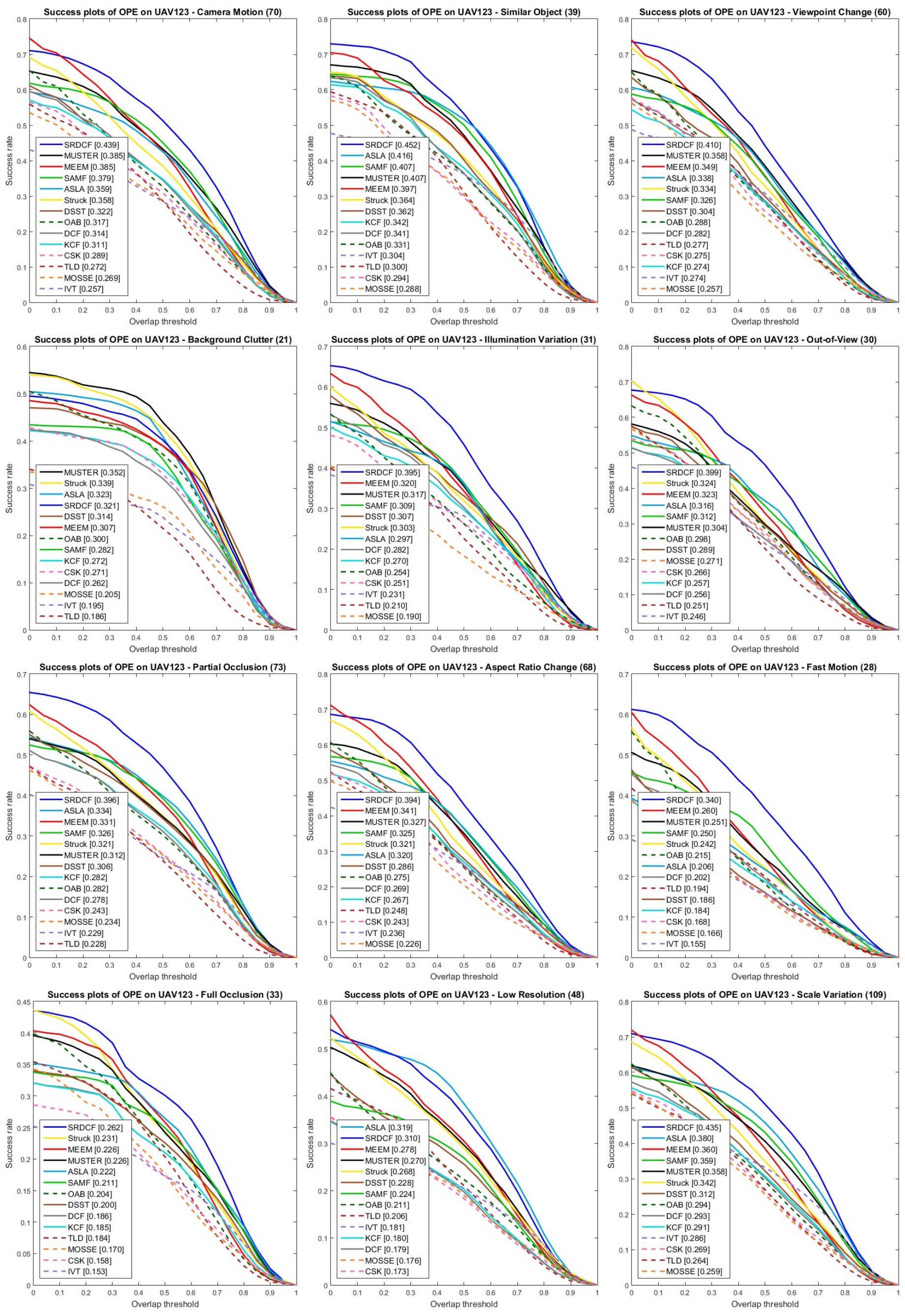

Fig. 5. OPE success plots by attribute.

2 Simulator

2.1 UAV physics simulation and control

Inside UE4, the UAV is represented as an octo-copter with attached camera gimbal and low-level flight controller that maintains the UAV position and altitude within a physics based environment. Movement of the copter is updated per frame by UE4's physics simulator that accounts for Newtonian gravitational forces, input mass, size of the UAV, and linear/angular damping. Rotating the normal of the thrust vector along the central x and y axis of the copter and varying thrust enable the copter to move within the physics environment similar to real-world flight. Similar to hardware in the loop (HITL) approaches, the UAV control in UE4 utilizes several tuned PID controllers (written in C++ and accessed as a UE4 BluePrint function). Since we want to model a UAV in position hold, movement in the X and Y directions is simulated through the update of required roll and pitch by a PID controller for each that minimizes the error of desired velocity in the x and y established by the setpoint. This simulates the stick inputs from a flight transmitter or more specifically desired velocity vectors updates by the tracker under evaluation. Similar to real-world tuning we adjust experimentally within the simulator the Kp, Ki and Kd weights until we achieve a smooth response. Altitude of the UAV is maintained by an additional PID controller that variably adjusts thrust based on desired error between current altitude and desired altitude. Through this system we are able to simulate closely real-world flight of multirotors and control it within the simulator through either a joystick or direct input. The actual position of the UAV remains unknown to the controller and the trackers.

2.2 Frame capture, object segmentation, flight logging

Attached to the UAV is a camera set at a 45 degree angle and located below the frame of the copter. At every frame of the copter, the camera's viewport is stored in two textures (full rendered frame and custom depth mask) then written to a RAM disk for fast retrieval in Matlab. UE4 supports custom depth maps that can be assigned to any object in the engine. Finally, at each frame, we log the current bounding box, position, orientation, and velocity of the tracked target and the UAV.

2.3 MATLAB/C++ integration

Since the majority of trackers in the evaluation run only in MATLAB direct C++ integration in the UE4 simulator is not possible for all. These trackers are setup to read from the RAM disk the latest image outputted by the simulator. The outputted frames are output at 720p or 360p. For tracker evaluation we use 360p, to reduce latency. A script runs in Matlab to initialize the current tracker with a bounding box sent by the simulator. Once initialized a bool callback is sent to the UE4 simulator and the simulation of the vehicle or ai is triggered to start. The Matlab script continues to feed the latest read frame to the current tracker. For slow trackers they will get the most recent frame but drop/miss frames while they are in the middle of processing. Out of the four trackers evaluated (SRDCF, SAMF, MEEM, STRUCK) the only tracker that maintains 30fps at 360p is STRUCK. The output of the tracker after processing a frame is a bounding box. The updated bounding box is read by UE4 at every frame and used to calculate error between the center of the camera frame and the bounding box center.

2.4 Visual Servoing

The objective of the onboard flight control simulator is to update the position of the UAV to bring the tracked object back to the center of the camera FOV. This is efficiently accomplished in real-time by the calculation of the error from the tracker's bounding box and its integration with two PID controllers. Since the camera system is mounted on a gimbal and the camera angle and altitude are held constant, only the vertical and horizontal offsets need to be calculated in the current video frame to properly reorient the UAV. The translational error in the camera frame is obtained by finding the difference between the current target's bounding box center (computed by the OFC tracker) and center of the video frame. A fully-tuned PID loop for both the X and Y dimensions receives the offset vector and calculates the proportional response of the copter movement to recenter the tracked object. Please see the attached video for demonstration of various trackers performance in following the targets. The visual servoing technique employed in the simulator is robust and with top performing trackers is able to follow targets across large and diverse environments. In the online evaluation experiments the UAV is able to follow autonomously the vehicle across a 1km winding off-road track with varying elevation changes.

References

1. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. (June 2012) 1822–1829
2. Henriques, J., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C., eds.: Computer Vision ECCV 2012. Volume 7575 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 702–715
3. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. Pattern Analysis and Machine Intelligence, IEEE Transactions on (2015)
4. Danelljan, M., Hger, G., Shahbaz Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: Proceedings of the British Machine Vision Conference, BMVA Press (2014)
5. Ross, D., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. International Journal of Computer Vision **77**(1-3) (2008) 125–141
6. Zhang, J., Ma, S., Sclaroff, S.: MEEM: robust tracking via multiple experts using entropy minimization. In: Proc. of the European Conference on Computer Vision (ECCV). (2014)
7. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. (June 2010) 2544–2550
8. Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D., Tao, D.: Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. (June 2015) 749–758
9. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: Proceedings of the British Machine Vision Conference, BMVA Press (2006) 6.1–6.10 doi:10.5244/C.20.6.
10. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Čehovin, L., Nebehay, G., Vojíř, T., Fernandez, G., Lukežič, A., Dimitriev, A., et al.: The visual object tracking vot2014 challenge results. In: Computer Vision-ECCV 2014 Workshops, Springer (2014) 191–217
11. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: The IEEE International Conference on Computer Vision (ICCV). (Dec 2015)
12. Hare, S., Saffari, A., Torr, P.H.S.: Struck: Structured output tracking with kernels. In: 2011 International Conference on Computer Vision, IEEE (Nov 2011) 263–270
13. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-Learning-Detection. IEEE transactions on pattern analysis and machine intelligence **34**(7) (Dec 2011) 1409–1422