

# Combining Visual Tracking and Person Detection for Long Term Tracking on a UAV

Gustav Häger, Goutam Bhat, Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, Piotr Rudol and Patrick Doherty

## Conference article

Cite this conference article as:

Häger, G., Bhat, G., Danelljan, M., Shahbaz, F., Felsberg, M., Rudol, P., Doherty, P. Combining Visual Tracking and Person Detection for Long Term Tracking on a UAV, In Proceedings of the 12th International Symposium on Advances in Visual Computing, ; 2016, pp. 557-568. ISBN: 978-3-319-50834-4

DOI: [https://doi.org/10.1007/978-3-319-50835-1\\_50](https://doi.org/10.1007/978-3-319-50835-1_50)

Copyright: <https://www.springer.com/>

The self-archived postprint version of this conference article is available at Linköping University Institutional Repository (DiVA):

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-137897>



# Combining visual tracking and person detection for long term tracking on a UAV

Gustav Häger, Goutam Bhat, Martin Danelljan, Piotr Rudol,  
Fahad Khan, Michael Felsberg, Patrick Doherty

October 31, 2019

## Abstract

Visual object tracking performance has improved significantly in recent years. Most trackers are based on either of two paradigms: online learning of an appearance model or the use of a pre-trained object detector. Methods based on online learning provide high accuracy, but are prone to model drift. The model drift occurs when the tracker fails to correctly estimate the tracked objects position. Methods based on a detector on the other hand typically have good long-term robustness, but reduced accuracy compared to online methods.

Since few attempts have been made to combine these approaches in a principled manner, we propose a novel fusion between of an online tracker and a pre-trained detector for tracking humans from a UAV. The system runs in real-time on a UAV platform. In addition we present a novel dataset for long-term tracking in a UAV setting, including scenarios that are typically not well represented in visual tracking datasets.

## 1 Introduction

Visual tracking is one of the classic computer vision problems, as it has a wide range of applications in surveillance and robotics. In a surveillance scenario a tracking system could be used to detect when a person is moving into a prohibited area. In robotics a real-time tracking system can be used to track the positions of potentially dangerous objects, or to make

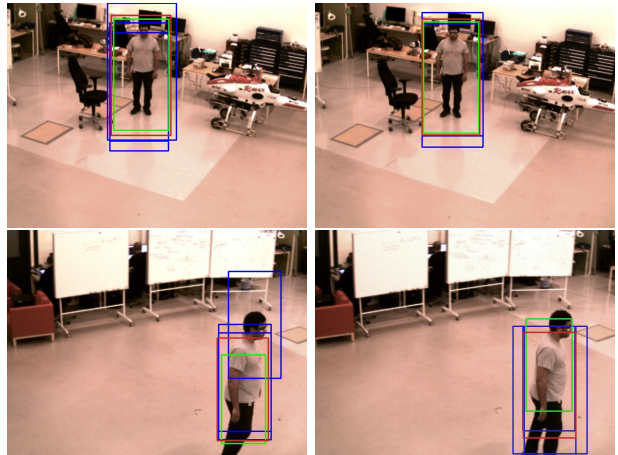


Figure 1: Visualization of fusion system, the detector output is blue, tracker output green and the final combined output red. Top row shows how the combination of both tracker and detector produces a more accurate bounding box estimate than the input. Bottom row shows how drift in the tracker is corrected by using the detector measurements

the robot follow a specific person at a set distance. Recently a number of challenges in the visual tracking area have triggered a high pace of improvement in the area of online-tracking [10, 9, 8, 12, 11]. A particularly interesting class of trackers is the model-free tracker. Here model-free refers to the fact the tracker does not require any prior information about the tracked target, only an initializing bounding box is required. These methods are typically evaluated on

datasets such as OTB [13], PETS [12, 11], or VOT [10, 9, 8]. These datasets are composed of a large number of short videos, typically taken from youtube or recorded explicitly for purpose of benchmarking, usually with a stationary or smoothly moving camera.

A popular type of robotics platform is the Unmanned Aerial Vehicle (UAV), these robots are usually equipped with a wide range of sensors, including a camera. A typical situation is that the operator instructs the UAV to follow a designated person at a fixed distance without manual intervention. This requires the UAV to have the ability to track the designated target, and act on the information produced by the tracker. As the camera is fixed on the UAV the view might suddenly change when the UAV is repositioning or is impacted by wind. It is usually desired that the system can follow the designated person for an extended period of time, likely for thousands of frames rather than the few hundred common in most benchmark videos [8]. Such scenarios are problematic for the current model-free trackers, as they are prone to model drift, and will eventually loose the tracked object.

The drift problem is not present in methods based on a pre-trained object detector, as they do not update the appearance model online. The most recent methods such as deformable parts models (DPM), and methods based on convolutional neural networks (CNN) have increase the state of the art performance significantly in detection tasks. However the high computational complexity in evaluating such models make them unsuited to real-time operation on systems with limited hardware such as a UAV. A tracking system based on general object detectors will attempt to associate each detection with a tracked object, or when no known object matches initialize a new track. An additional disadvantage of this type of tracker is a single object will give a large number of detections, the positioning is typically less exact then in a model free tracker. The output from the detector and tracker, as well as the final combined output for our system is visualized in figure 1.

In order for a UAV to accurately follow a designated person the tracking system must fulfill certain requirements. The object tracker should output posi-

tion and size estimates that are accurate at all times, as well as notice when the estimate is not sufficiently certain. The system should be robust against temporary difficulties such as occlusions and unstable camera movement. Finally in order to be practically useful it should be capable of real-time operation on the limited hardware present on a UAV.

## 1.1 Contribution

We propose a framework for combining the output of an online learning visual tracker and an offline human detector. The framework is capable of real time operation on a UAV platform while being robust over large numbers of tracked frames. Our method is compared with two baseline methods on a dataset gathered using our UAV platform. We also evaluate on the PETS2016 low-level (tracking) data, where the system initializes tracks automatically rather than by an operator.

We also present a dataset for long term tracking. All sequences are recorded with a flying UAV, and are significantly longer than the typical short term tracking video. The sequences contain long term occlusions of the entire tracked person, and background of varying complexity. Further challenging situations are long term partial occlusions, significant changes in viewpoint and deformations of the tracked person as he is sitting down. One sequence also includes a number of distracting events where other humans walk past the tracked person and temporarily occludes him.

## 2 Related work

There are two common approaches to visual tracking, model free tracker using online learning to create a robust appearance model of the specific tracked target, or using a pre-trained detector and associating detection with a tracked target. Model free trackers require no prior information about the target, except an initializing bounding box. An appearance model is then created online by gathering additional samples while tracking. Detection based trackers on the other hand use a detector for the object or class to

track, this detector is applied on each new frame. The tracking problem then becomes a matter of associating each detection with an already tracked object or initialing new objects to track. However few attempts have been made to combine the strengths of both approaches into a single system. In this paper we present such a system, for online tracking of humans on a micro UAV platform.

## 2.1 Visual object tracking

In the last few years a great deal of progress has been made in visual object tracking. In particular methods based on discriminative correlation filters have shown a great deal of promise, in the 2014 challenge the top 3 methods were DCF based. Trackers based on the DCF framework exploit the circulant structure of images and the Fourier transform to efficiently create a linear classifier. Our method is based on a combination of the winning entry in the VOT 2014 challenge [2], but rather than using the HOG features we use the lower dimensional color names suggested in [4]. This allows our implementation to run at very high frame-rates while maintaining good accuracy in both translation and scale estimation.

## 2.2 Visual object detection

Methods for visual object detection, using a wide range of classifiers and feature representations exist in literature. Of particular interest is the method utilizing Histogram of Oriented gradient features proposed by Dalal [1]. Using this feature representation in a sliding window support vector machine (SVM) an efficient and robust classifier is obtained. This provides a fast detector that is suitable for real-time operation.

Other popular methods include Deformable Parts models such as the one proposed by Felsenwalb [5] or a number of deep learning based methods. In practice these more complex models require an order of magnitude or more of computational power above Dalals method, as such they are impractical to use on a UAV with limited computational capacity, particularly for real-time operation.

## 2.3 Detector and tracker fusion

The combination of a model-free tracker and a static detector is a conceptually simple way to improve the long term robustness of a tracking system. However how to combine the outputs in way that maintains the accuracy of the online tracker while maintaining the robustness of the detector approach over longer term is not trivial. A previous attempt was made in [3] where the output of both the tracker and detector were used as inputs into a Probability Hypothesis Density (PHD) filter. However this approach disregards that the online component contains valuable appearance information from the tracked object. Other approaches include the PN learning proposed by Kalal [7] that utilizes binary classifiers and the structural constraints of the labels.

## 3 Active vision framework

Our vision framework combines the output of a pre-trained human detector with that of a model-free correlation filter based tracker. An overview of the system is present in figure 3. The complete system is composed of three main parts, an online model-free tracker based on the Discriminative correlation filter framework. A human detector trained off-line, with a static model that runs over the image in a sliding window, or is evaluated at a particular point. A system that observes the performance of each subsystem in order to estimate the current reliability of each one.

### 3.1 DCF based online tracker

The online tracker used is based partly on the DSST [2] and the ACT [4]. Both of these methods and ours are based on the framework of discriminative correlation filters. We use the color names representation proposed in [4], and the separate scale filter suggested in [2], where we use a gray scale feature instead of the HOG used by Danelljan. These changes allows the tracker to run at high frame-rates, while maintaining similar performance. Further it gives a degree of complementarity with the detector as it uses HoG features.

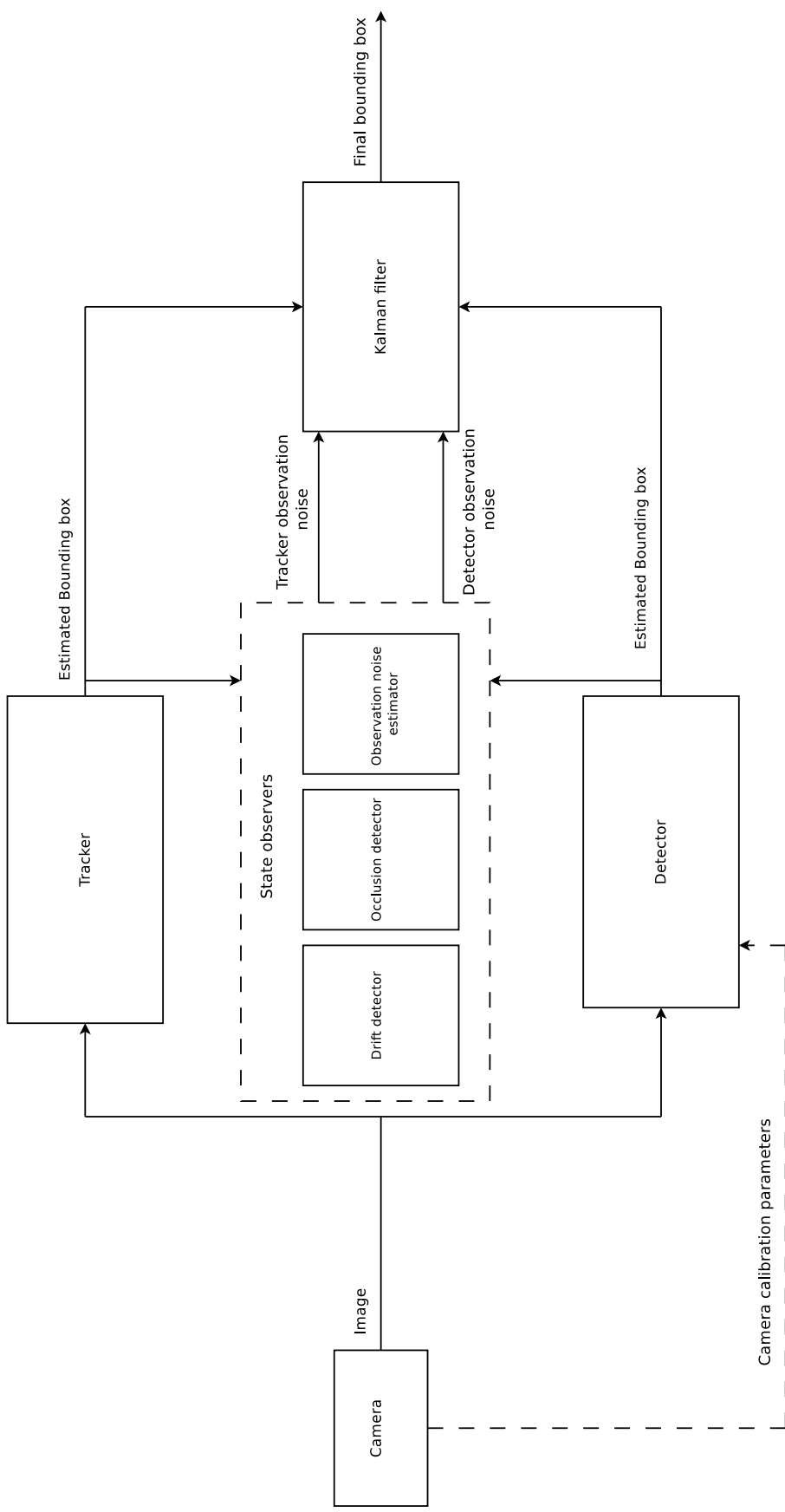


Figure 2: An overview of the tracking system, the details of the tracker are described in 3.1, the detector in 3.2 and the observer components in 3.4

Discriminative correlation filters create a classifier  $h$  by specifying a desired output  $y$  at a given input  $x$  input and minimizing the error for the classifier  $h$  for the input  $x$ . With the commonly used approximation [2, 4, 6] for multidimensional features the error function becomes:

$$\epsilon = || \sum_{l=1}^d h^l \star x^l - y || + \lambda \sum_{l=1}^d ||h^l||^2 \quad (1)$$

The  $\star$  denotes circular correlation, while the  $\lambda$  is a small regularization factor. This optimization can be efficiently solved in the Fourier domain with the closed form solution:

$$H^l = \frac{\bar{Y} X^l}{\sum_{k=1}^d \bar{X}^k X^k + \lambda} \quad (2)$$

Where  $H, Y, X$  denotes the Fourier transform of the respective variables, and  $\bar{X}$  the complex conjugate. The classifier is updated using linear interpolation for each frame yielding a compact and efficient appearance representation. Further details and derivations can be found in [4, 2, 6].

In a new frame a position estimate is computed by computing the filter response over a patch. The new position  $P_{trk}$  is then taken as the point in the output with the highest value. The new target position is taken as the maximum of this score function.

In cases of tracker drift the model will typically be corrupted by gradually adapting to the background instead of the target. Initially this will give an offset from the true target position that gradually moves away from the correct position over time. When taking the possibility of drift into account the trackers position estimate  $P_{trk}$  could be modeled as:

$$P_{trk} = P + \mathcal{N}(b_t, \sigma_{trk}) \quad (3)$$

Where the true position  $P$  is perturbed by noise from  $\mathcal{N}(b_t, \sigma_{trk})$  that represents the current tracker drift as a time-varying bias  $b_t$ , and the variance of the position estimate  $\sigma_{trk}$  is approximately constant over time.

### 3.2 Person detection

Our system uses a SVM with HoG features as image representation, as proposed by [1]. The classifier is

evaluated in a sliding window manner over a scale pyramid. The scale pyramid is computed with the current target size estimate in the center. The SVM model is trained on the INRIA dataset, augmented with a few example frames collected by our UAV. In order to reduce the number of scales detection is run at some prior information about the rough size of the detected humans is needed, otherwise the scale-search becomes prohibitively slow.

The detector outputs a large number of detections for each target, spread over a range of scales and positions. While there are confidences assigned by the detector, it is not guaranteed that the detection with the highest confidence is the most accurate one.

Once the detector has been evaluated over a new frame all detections with confidence below a certain threshold is removed, the remaining detections confidences are re-weighted by multiplying with a Gaussian centered on the current target position. After weighting the detection with highest confidence is used output from the detection system. The final estimate of the detector position can be modeled as:

$$P_{det} = P + \mathcal{N}(0, \sigma_{det}) \quad (4)$$

Where unlike in 3 the detector does not have a time-varying bias. However the variance for the detector  $\sigma_{det}$  is typically much larger than  $\sigma_{trk}$ .

### 3.3 Our fusion framework

We combine information from the tracker and detector in two ways. First the position and size estimated by both the tracker and detector is combined as inputs in a Kalman filter to generate a more reliable estimate than each one individually

Secondly the reliability of both the model-free tracker and the detector is monitored in order to correct for tracker drift, detect target loss and associate recently detected new objects with old tracks. Additionally the reliability estimates are used to update the observation noise for the Kalman filter continuously. In addition when the tracker proves reliable for longer periods of time, snapshots of the current appearance model is stored for use in re detection should the target be lost in the future.

### 3.4 State monitoring

The current reliability of both the detector and tracker is computed continuously, in order to set the observation noise for the Kalman filter, and in order to detect corruptions in the online learning component. The variance for the detector can be computed from the spread of detections compared to the current best estimated position. While it this estimate could be done momentarily, we accumulate data over a short time window in order to produce a more reliable estimate for  $\sigma_{det}$ . In practice the observation noise is set to either a high level when the  $\sigma_{det}$  value is large, or a lower one when it is smaller.

From the proposed observation models 3 and 4 a principled approach for detecting model drift can be derived. Since the detector is unbiased but noisy, drift in the tracker can be detected by comparing the respective estimates over time. If the online tracker maintains high confidence, but with a consistent offset in position estimate compared to the detector for a number of frames, it is likely that the appearance model used by the tracker has begun to drift away from the center of the target. In these case the tracker model is re-initialized at the current best estimate of the targets position.

A rough estimate of the trackers confidence can be computed as the ratio of energy in the correlation peak over the total energy of the response. Should this ratio be insufficiently large in a frame the tracker model is not updated. If it is consistently high for a large number of frames ( 100), while the detector confirms the accuracy a snapshot of the current model is stored for use in re detection.

Using this confidence information it is possible to detect situations when the tracked person is no longer in view for the tracker, such as occlusions. In such situations the confidence of the tracker will typically drop very low, but begin to increase as the model adapts to the occluding object. After sufficient time the confidence will be higher than typical when tracking an articulated human. At the same time the detector will consistently fail to give any detections. In such cases the system will flag for loss of target and switch into re detection mode. When in re detection mode the detector scans the full image, until a reli-

able detection is made, previously stored models are evaluated on the new detection. Should one of the stored models match sufficiently well the tracking resumes.

## 4 Dataset

We provide a dataset of four sequences for long-term UAV tracking. The sequences are recorded with the UAV flown manually, with the pilot instructed to keep the target in view. The sequences feature a range of different persons walking around the lab. The main goal in recording the dataset was to capture longer sequences than is typically used in visual tracking, and representing UAV specific difficulties well. Since all sequences are recorded using a flying UAV the camera is continuously moving, with some sudden jerks as the UAV repositions.

The sequences feature some difficulties well represented in visual tracking datasets, such as very long term partial occlusions, periodic full occlusions and jerky camera movement. One sequence has the tracked person sitting down for a period, one has multiple humans crossing each other in the image. In all sequences additional humans are present in the background. An additional difficulty is that the sequences are far longer than the ones commonly used, at 3400-4900 frames, while in most datasets sequences with more than 1000 frames are rare, and most are approximately 300-400 frames.

### 4.1 Data acquisition system

LinkQuad is a versatile autonomous Micro Aerial Vehicle. The platform's airframe is characterized by a modular design which allows for easy reconfiguration to adopt to a variety of applications. Thanks to a compact design (below 70 centimeters tip-to-tip) the platform is suitable for both indoor and outdoor use. It is equipped with custom designed optimized propellers which contribute to an endurance of up to 30 minutes. The maximum take-off weight of the LinkQuad is 1.6 kilograms with up to 300 grams of payload.

LinkQuad is equipped with in-house designed flight

control board - the LinkBoard. The LinkBoard has a modular design and this allows for adjusting the required computational power depending on mission requirements. In the full configuration, the LinkBoard weighs 30 grams, has very low power consumption and has a footprint smaller than a credit card. The system is based on two ARM-Cortex microcontrollers running at 72MHz which implement the core flight functionalities.

The LinkBoard includes a three-axis accelerometer, three rate gyroscopes, and absolute and differential pressure sensors for estimation of the altitude and the air speed, respectively. The LinkBoard features a number of interfaces which allow for easy extension and integration of additional equipment.

## 5 Experiments

We evaluate our proposed tracker and detector fusion on our own dataset. The results are reported as overlap and precision plots.

### 5.1 Evaluation methodology

While the VOT [10, 9, 8] method of evaluating trackers provides the least biased estimate of tracker accuracy for short term trackers, the automatic restarting present in the toolkit makes it unsuited for evaluation of long-term trackers on sequences with significant occlusion. Results of two short term DCF based trackers are also included, due to In both cases the implementation used is the one used in the VOT2014 challenge.

Instead we use a less sophisticated measure of total number of correctly tracked frames, where a frame is considered correctly tracked if the estimated bounding box overlap is greater than some threshold. The results is presented as a success curve with the threshold on the x-axis and the ratio of total successfully tracked frames for the threshold on the y-axis. When computing the scores the frames with undefined ground truth, for example due to the tracked person being occluded is not counted.

### 5.2 Results

We compare our proposed system with two variants, we also compare with some state of the art methods from the VOT challenge.

We compare the performance of our proposed system using the tracker-detector fusion, with two baseline variants. The first baseline uses only the detector component, and a Kalman filter. The second baseline uses only our tracker component, here restarts of the tracker model is managed by watching only the confidence of the tracker.

### 5.3 On the PETS dataset

To be included...

## References

- [1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. [3](#), [5](#)
- [2] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014. [3](#), [5](#)
- [3] M. Danelljan, F. S. Khan, M. Felsberg, K. Granström, F. Heintz, P. Rudol, M. Wzorek, J. Kvarnström, and P. Doherty. A low-level active vision framework for collaborative unmanned aircraft systems. In *Computer Vision-ECCV 2014 Workshops*, pages 223–237. Springer International Publishing, 2014. [3](#)
- [4] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. van de Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*, 2014. [3](#), [5](#)
- [5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010. [3](#)
- [6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized

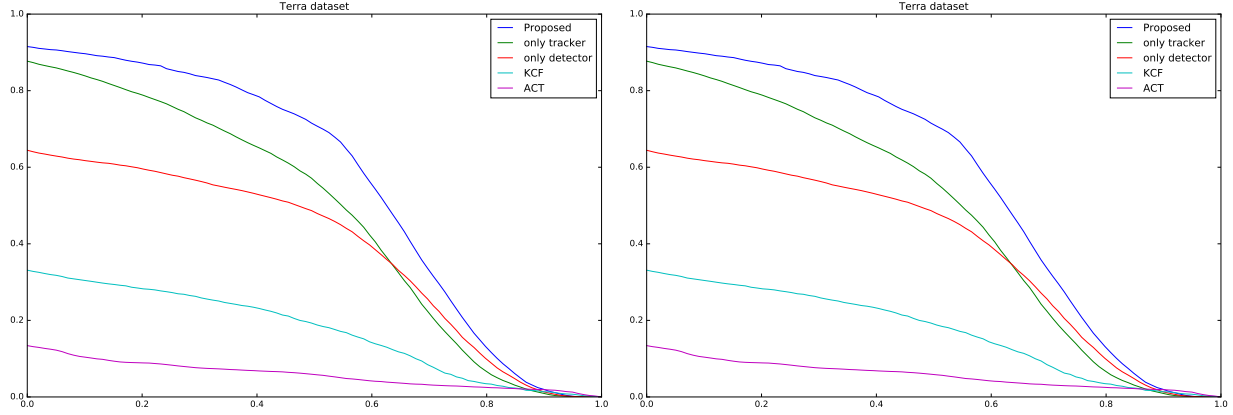


Figure 3: Total overlap per threshold for all sequences. Left plot is missing

- correlation filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2015. [5](#)
- [7] Z. Kalal, J. Matas, and K. Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 49–56. IEEE, 2010. [3](#)
- [8] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder. The visual object tracking vot2015 challenge results. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015. [1](#), [2](#), [7](#)
- [9] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Cehovin, G. Nebehay, T. Vojir, G. Fernandez, et al. The visual object tracking vot2014 challenge results. In *ECCV Workshops*, 2014. [1](#), [2](#), [7](#)
- [10] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Čehovin, G. Nebehay, G. Fernandez, and T. Vojir. The visual object tracking vot2013 challenge results. Dec 2013. [1](#), [2](#), [7](#)
- [11] T. Nawaz, J. Boyle, L. Li, and J. Ferryman. Tracking performance evaluation on pets 2015 challenge datasets. In *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*, pages 1–6. IEEE, 2015. [1](#), [2](#)
- [12] L. Patino and J. Ferryman. Pets 2014: dataset and challenge. In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*, pages 355–360. IEEE, 2014. [1](#), [2](#)
- [13] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013. [2](#)