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

Gustav Häger[1(✉)], Goutam Bhat[1], Martin Danelljan[1], Fahad Shahbaz Khan[1], Michael Felsberg[1], Piotr Rudl[2], and Patrick Doherty[2]

[1] Computer Vision Laboratory, Linköping University, Linköping, Sweden
gustav.hager@liu.se

[2] Artificial Intelligence and Integrated Computer Systems, Linköping University, Linköping, Sweden

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 object's position. Methods based on a detector on the other hand typically have good long-term robustness, but reduced accuracy compared to online methods.

Despite the complementarity of the aforementioned approaches, the problem of fusing them into a single framework is largely unexplored. In this paper, we propose a novel fusion between an online tracker and a pre-trained detector for tracking humans from a UAV. The system operates at real-time on a UAV platform. In addition we present a novel dataset for long-term tracking in a UAV setting, that includes scenarios that are typically not well represented in standard visual tracking datasets.

1 Introduction

Visual tracking is one of the classic computer vision problems, with 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 objects of interest, for example to make 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 [1–5]. A particularly interesting class of trackers is the model-free tracker. Here *model-free* refers to the fact the tracker does not require any information beyond an initial bounding box. These methods are typically evaluated on datasets such as OTB [6], or VOT [1–3]. These datasets are composed of a large number of short videos, typically recorded using a high quality camera.

A popular robotics platform is the Unmanned Aerial Vehicle (UAV), these robots are usually equipped with a wide range of sensors, including cameras. A typical situation is that the operator instructs the UAV to follow a designated

G. Bebis et al. (Eds.): ISVC 2016, Part I, LNCS 10072, pp. 557–568, 2016.
DOI: 10.1007/978-3-319-50835-1_50

Fig. 1. Visualization of fusion system, the detector output is blue, tracker output green and the fused result red. The combination of both tracker and detector produces a more accurate bounding box estimate than the respective inputs, as seen in the first two frames. The last two demonstrate drift correction.

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 [3]. Such scenarios are problematic for the current model-free trackers, as they are prone to model drift, and will eventually lose 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 convolutional neural networks (CNN) have increased the state of the art performance significantly in detection tasks [7]. Unfortunately this increase in performance demands significantly more computations. 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. A disadvantage of this type of tracker is that a single object will give a large number of detections of high confidence. For these reasons detector based methods typically give a more noisy estimate of the target bounding box.

In order for a UAV to accurately follow a designated person the tracking system must fulfill certain requirements. The object tracker should output position and size estimates that are accurate at all times, or notify the system that the estimate is not sufficiently precise to act on. The system should be robust in difficult situations 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. A visualization of the output from our detector, tracker and combined position estimate is present in Fig. 1.

1.1 Contribution

We propose an approach for fusing the output of an online model-free tracker and a pre-trained person detector for UAV based tracking of humans. The system is capable of real-time operation on a UAV platform.

Additionally we present a challenging dataset for long term tracking from a UAV. All sequences are recorded with a flying UAV, and are significantly longer than the typical tracking benchmark videos. The sequences contain long term occlusions of the entire tracked person, and background of varying complexity. Further challenging situations are long term partial occlusions, and significant change of pose of the tracked person. One sequence also includes a number of distracting events where other humans walk past the tracked person and temporarily occlude him.

2 Related Work

There are two common approaches to visual tracking, model free tracking using on-line learning to create a robust appearance model of the specific tracked target, or using a pre-trained detector and associating detections with a tracked target. Model free trackers such as those evaluated in the VOT challenge [1–3] require no prior information about the target, except an initial bounding box. An appearance model is created on-line 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 on-line tracking of humans on a micro UAV platform.

2.1 Visual Object Tracking

In the last few years a significant progress has been made in visual object tracking. In particular methods based on Discriminative Correlation Filters (DCF) have shown a great deal of promise, in the 2014 visual object tracking (VOT) challenge [2] the top 3 methods where 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 [8], but rather than using the HOG features we use the lower dimensional color names suggested in [9]. The lower dimensionality of the color names descriptor allows our implementation to run at high frame rates while maintaining comparable accuracy.

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 [10]. 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 Felsenzwalb [11] or a number of deep learning based methods [12,13]. In practice these more complex models require an order of magnitude or more of computational power beyond Dalal's method, as such they are impractical to use on a UAV with limited computational capacity, particularly when real-time operation is desired.

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 combining the tracker and detector in way that maintains the accuracy of the on-line tracker while keeping the long term robustness of the detector is not trivial. A previous attempt was made in [14] were the output of both the tracker and detector where used as inputs into a Probability Hypothesis Density (PHD) filter, this approach disregards that the on-line component contains valuable appearance information from the tracked object.

Other approaches include the PN learning proposed by Kalal [15] that utilizes binary classifiers and the structural constraints of the labels. This approach is purely on-line learning based, unlike our combination of pre-trained detector and on-line learning tracker.

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

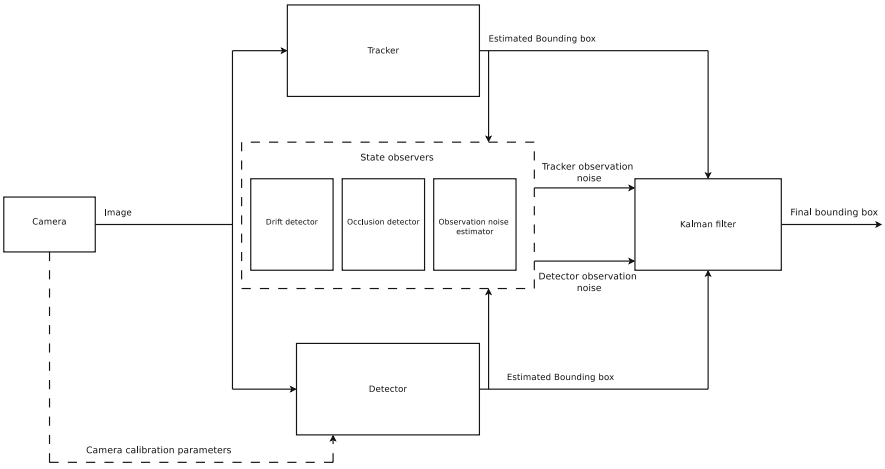

Fig. 2. An overview of the tracking system, the details of the tracker are described in Sect. 3.1, the detector in Sect. 3.2 and the observer components in Sect. 3.4

system is presented in Fig. 2. The complete system is composed of three main parts: an online model-free tracked 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 small region; 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 [8] and the ACT [16]. Both of these methods are based on the framework of Discriminative Correlation Filters. We use the color names representation proposed in [16], and the separate scale filter suggested in [8], where we use a gray scale feature instead of the HOG used by Danelljan et al. This is done in order to reduce the dimensionality of the translation and scale estimation filters. This reduced dimensionality (from 31 to 11 and 31 to 1 respectively) significantly reduces the computational burden while maintaining the performance.

Discriminative Correlation Filters create a classifier h by specifying a desired output y at a given input x and minimizing the error for the classifier h for the input x. With the commonly used approximation [8,16,17] for multidimensional features the error function becomes:

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

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

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

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 [8,16,17].

In a new frame a position estimate is computed from the filter response over a patch. The new position P_{trk} corresponds to the pixel in the patch with the highest value.

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 tracker's position estimate P_{trk} could be modeled as:

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

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

3.2 Person Detection

Our system uses an SVM with HOG features as image representation, as proposed by [10]. 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 [10], augmented with a few example frames collected by our UAV. In order to reduce the number of scales for the detection prior information of the targets size is taken into account. When no prior information about target size and position is available, the detector is run over a full scale-space pyramid of the image.

The detector outputs a large number of detections for each target, spread over a range of scales and positions. While each detection has a confidence, it is not guaranteed that the detection with the highest confidence is the correct one.

Once the detector has been evaluated over a new frame, all detections with confidence below a certain threshold are removed. For the remaining detections a weighted average is computed using a Gaussian centered on the current position estimate. This gives the detector estimate P_{det} of the targets position as:

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

Here, unlike in 3 the detector does not have a time-varying bias, as the model is not updated online. However the variance for the detector σ_{det} is typically much larger than σ_{trk}.

3.3 Our Fusion Framework

We combine information from the tracker and the detector in two ways. First the position and size estimated by both the tracker and the detector is combined by a Kalman filter in order to produce a more robust measure than either one individually.

Secondly the reliability of both the model-free tracker and the detector is monitored in order to correct for tracker drift. Additionally the reliability estimates are used to update the observation noise for the Kalman filter continuously. Finally, when the tracker proves reliable for a longer period of time, a snapshot of the appearance model is stored in order to re-detect the target if it is lost.

The state vector for the Kalman filter is:

$$K_{\text{state}} = [P_x, P_y, P_w] \tag{5}$$

where P_x, P_y correspond to the top-left corner of the tracked bounding box, and P_w to the width of the box. As the bounding box has a fixed ratio between width and height only the width is needed to represent the bounding box size.

3.4 State Monitoring

The current reliability of both the detector and the tracker is estimated continuously. This is done in order to detect corruption in the on-line learning component, and to set the observation noise for both inputs into the Kalman filter.

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, the time varying bias b_t caused by the tracker drifting can be detected by comparing the position estimates over time. If the tracker maintains high confidence, but with a consistent offset in the estimated position relative to the detector, it is likely that the appearance model used by the tracker has begun to drift away from the center of the target. Due to the noisy position estimate provided by the detector it is difficult to obtain an accurate estimate of the tracker bias. Instead of producing a correct estimate of the bias, the tracker model is restarted on the current best estimated position.

A rough estimate of the tracker's confidence in the current prediction can be obtained from the height of the correlation peak. In order to correctly re-identify a lost target snapshots of the tracker model is stored periodically. The current model is considered reliable only if the score peak has been higher than some threshold t_r for more than 100 consecutive frames.

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 these 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 this mode the detector scans the full image, until a reliable detection is made. Previously stored models are evaluated on the detection, if one matches sufficiently well tracking will resume.

Kalman Filter Observation Noise. The observation noise for the Kalman filter for both the detector and tracker is updated in each new frame. For the tracker the noise is set relative to the height of the peak. In practice only two settings for the observation noise are used: If the peak is above some threshold T_{low} the observation noise is set to a low value, otherwise it is set to a high one.

For the detector it is possible to use the spread of detections to estimate the variance σ_{det}. While it is possible to use the variance directly to set the observation noise, this would discard important information gathered over time. Instead the observation noise is weighted based on how well the detections have matched with the combined estimate over a short time window.

The final detector observation noise is set according to:

$$d_n = e^{(\frac{W_{\text{match}}}{d_c})^2} \tag{6}$$

where W_{match} is set high if the combined output has matched well during a short time window, and d_c is the distance from the current position estimate to the current detection. The d_c parameter mainly reduces the weight if the detector has very strong responses on some background object.

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. Each sequence features a different person to track. The main goal of our dataset is to capture longer sequences than is typically used in visual tracking, while 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. We call this new dataset 'Terra' after the lab where it was recorded. A description of the difficulties in each sequence is in Table 1.

4.1 Data Acquisition System

The 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 cm tip-to-tip) the platform is suitable for both indoor and outdoor use.

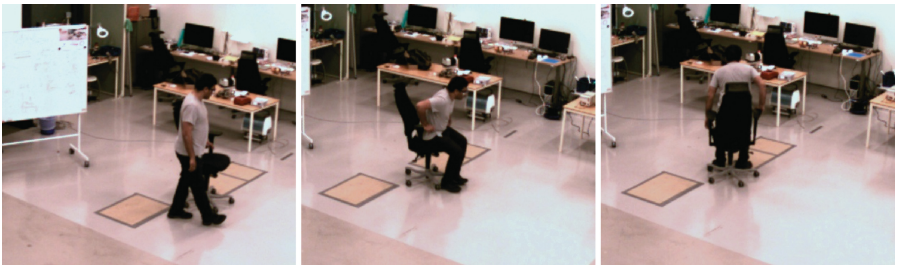

Fig. 3. Example frames from some challenging situations in our dataset. From the Sitting sequence. When the tracked person sits down the deformations are severe enough that most object detectors will fail.

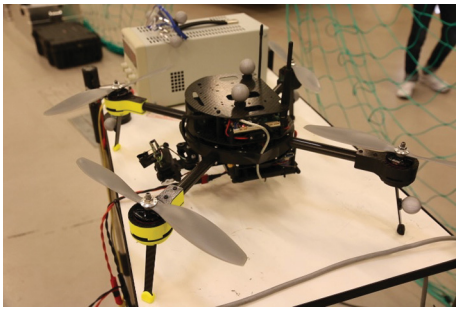

Fig. 4. The Linkquad UAV used to record our dataset. This configuration features a PointGrey camera and an Intel-NUC motherboard for running the vision system on board.

Table 1. An overview of the sequences included in our dataset. The first column has the sequence name, the following four columns the degree of some difficulties in each sequence. The final column the number of frames in each sequence.

	Occlusions	Scale changes	Viewpoint changes	Pose change	length
Occlusion1	Short full occlusion	Significant	Significant	Always upright	3610
Occlusion2	Full and partial	Limited	None	Sits in chair	3156
Sitting	Long full occlusion	Significant	Minor	Sits in chair	3177
Walking	Long partial occlusion	Limited	Significant	Always upright	4854

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 g, has very low power consumption and has a footprint smaller than a credit card. The system is based on two ARM-Cortex micro controllers running at 72 MHz 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.

The configuration used during the recording of the dataset was a Firefly MV FMVU-03MTC camera from Point Grey Research connected to an on board Intel NUC i5 computer. A photo of the configured LinkQuad is in Fig. 4.

4.2 Challenges

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. Finally two of the sequences contain significant pose changes for the humans. A summary of each sequence is in Table 1.

Some example frames of challenging situations are presented in Fig. 3.

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 [1–3] method of evaluating trackers provides an unbiased estimate of tracker accuracy for short term trackers, the automatic restarting present in the toolkit makes it unsuited for evaluation of long-term trackers with an automatic recovery mechanism. Instead we use a simpler metric of computing the bounding box overlap with the ground

We also include two short term tracker variants, the KCF [17] and the ACT [16] tracker. For both short term trackers the implementations used are those from the VOT 2014 challenge.

5.2 Results

We compare the performance of our proposed system using the tracker-detector fusion, with two baseline variants. The results for all methods is presented in Fig. 5. The first baseline is based only on the pre-trained detector run over the image as described in Sect. 3.2. The best detection is used as input into the Kalman filter in each frame. The output from the Kalman filter becomes the bounding box estimate for each frame. The tracker only method uses the same online learning visual tracker as the full system, but without the detector component. Resets of the tracker model are handled by observing the tracker confidence score only. Should the confidence drop to a low enough level the tracker is restarted.

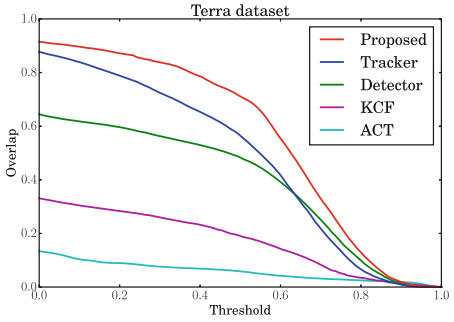

Fig. 5. The overlap for all frames with ground truth in the Terra dataset. The tracker-detector fusion clearly outperforms compared methods.

Interestingly using only the online tracker with a restart heuristic in case to low confidence yields better performance than the detection based method. This can likely be attributed to the purely detector based method having a not-insignificant possibility to get stuck on background objects. The KCF and ACT methods are based only on model-free trackers, and as such are very prone to drifting off the target and getting stuck on the background.

6 Conclusions and Future Work

Combining the output of a model-free tracker and a pre-trained object detector provides a significant increase in robustness for long-term tracking on UAVs. The proposed fusion method successfully combines the long-term reliability of a pre-trained detector with the precision of an online learned tracker, while maintaining real-time performance on a UAV platform. Possible future work include extending the dataset to a wider range of situations and humans, and making the system capable of tracking multiple targets at once.

Acknowlegdements. This work has been supported by SSF (CUAS, SymbiCloud), Wallenberg Autonomy and Software Programme (WASP) and ELLIIT.

References

1. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., Čehovin, L., Nebehay, G., Fernandez, G., Vojir, T.: The visual object tracking VOT 2013 challenge results (2013)
2. Kristan, M.: The visual object tracking VOT2014 challenge results. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8926, pp. 191–217. Springer, Heidelberg (2015). doi:10.1007/978-3-319-16181-5_14
3. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernandez, G., Vojir, T., Hager, G., Nebehay, G., Pflugfelder, R.: The visual object tracking VOT2015 challenge results. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (2015)
4. Patino, L., Ferryman, J.: Pets 2014: dataset and challenge. In: 2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 355–360. IEEE (2014)
5. Nawaz, T., Boyle, J., Li, L., Ferryman, J.: Tracking performance evaluation on pets 2015 challenge datasets. In: 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6. IEEE (2015)
6. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: CVPR (2013)
7. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**, 211–252 (2015)
8. Danelljan, M., Häger, G., Shahbaz Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: BMVC (2014)
9. Van De Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: Learning color names for real-world applications. IEEE Trans. Image Process. **18**, 1512–1523 (2009)
10. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
11. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**, 1627–1645 (2010)
12. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)

13. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
14. Danelljan, M., Khan, F.S., Felsberg, M., Granström, K., Heintz, F., Rudol, P., Wzorek, M., Kvarnström, J., Doherty, P.: A low-level active vision framework for collaborative unmanned aircraft systems. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8925, pp. 223–237. Springer, Heidelberg (2015). doi:10.1007/978-3-319-16178-5_15
15. Kalal, Z., Matas, J., Mikolajczyk, K.: P-N learning: Bootstrapping binary classifiers by structural constraints. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 49–56. IEEE (2010)
16. Danelljan, M., Shahbaz Khan, F., Felsberg, M., van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: CVPR (2014)
17. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)