Visual Object Tracking for Unmanned Aerial Vehicles: A Benchmark and New Motion Models

Siyi Li, Dit-Yan Yeung
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
sliay@cse.ust.hk, dyyeung@cse.ust.hk

Abstract

Despite recent advances in the visual tracking community, most studies so far have focused on the observation model. As another important component in the tracking system, the motion model is much less well-explored especially for some extreme scenarios. In this paper, we consider one such scenario in which the camera is mounted on an unmanned aerial vehicle (UAV) or drone. We build a benchmark dataset of high diversity, consisting of 70 videos captured by drone cameras. To address the challenging issue of severe camera motion, we devise simple baselines to model the camera motion by geometric transformation based on background feature points. An extensive comparison of recent state-of-the-art trackers and their motion model variants on our drone tracking dataset validates both the necessity of the dataset and the effectiveness of the proposed methods. Our aim for this work is to lay the foundation for further research in the UAV tracking area.

Introduction

Visual tracking is a fundamental problem pertinent to many real-world applications including video surveillance, autonomous vehicle navigation, human-computer interaction, and many more. Given the initial state (e.g., position and size) of the target object in a video frame, the goal of tracking is to automatically estimate the states of the moving object in subsequent frames. Although visual tracking has been studied for decades, it remains a challenging problem due to various factors such as partial occlusion, fast and abrupt object motion, illumination changes, and large variations in viewpoint and pose.

In recent years, we have witnessed the advent of a new type of robot, unmanned aerial vehicles (UAVs) or drones (Floreano and Wood 2015). Although drones were mostly used for military applications in the past, the recent commercial drone revolution has seen an increasing number of research laboratories working on small, affordable, human-friendly drones. The rapid development of commercial drones could have a major impact on many civilian applications, including transportation and communication. Meanwhile, a number of foreseeable applications on this new platform will need visual tracking as a core enabling technology. To name a few, visual tracking can make drones useful for tracking animals, finding people, monitoring real-time traffic situations, and so on.

In this paper, we study visual tracking on the drone platform. Besides the research issues common to visual tracking in general, a major new challenge that we have to face is the abrupt camera motion frequently encountered when using drones to capture video. Specifically, a small perturbation such as a slight rotation of the camera often leads to large displacement of the target position in the image scene. Also, since a drone flies, its motion typically has a higher degree of freedom than that of many conventional tracking applications. Therefore a more sophisticated motion model is needed. As a result, conventional motion models used for tracking applications with stationary or slow-moving cameras are no longer applicable. One focus of this paper is in conducting a benchmark evaluation and proposing baseline algorithms to explicitly estimate the ego-motion.

The goals of this paper are three-fold: 1. Construct a unified drone tracking benchmark dataset with detailed analysis of statistics; 2. Design general baseline algorithms for camera motion estimation and integrate them into various tracking systems; 3. Conduct an extensive experimental comparison and provide basic insights into the motion model in tracking, with the aim of opening up a new research direction for the visual tracking community.

Related Work

Many methods have been proposed for single object tracking in the last decade. For a comprehensive review and comparison of the trackers proposed, readers are referred to (Wu, Lim, and Yang 2013; Smeulders et al. 2014). In this section, we review some recent algorithms for object tracking in terms of the target representation scheme and the motion model. We also review the existing tracking benchmarks.

The target representation scheme determines how the appearance of the target is represented. Most trackers can be categorized into the generative or discriminative approaches. Generative approaches typically assume a generative process of the appearance model and locate an object by searching for the region most similar to the reference model. Such methods are typically based on templates (Alt, Hinterstoisser, and Navab 2010; Matthews, Ishikawa, and Baker 2004; Arandjelović 2015), principal component analysis (Ross et al. 2008), sparse coding (Mei

and Ling 2009), and dictionary learning (Wang, Wang, and Yeung 2013). On the other hand, discriminative approaches usually learn classifiers capable of separating the target from the background. Many advanced machine learning techniques have been applied, including boosting (Grabner, Grabner, and Bischof 2006; Grabner, Leistner, and Bischof 2008), multiple-instance learning (Babenko, Yang, and Belongie 2011), structured output support vector machine (Hare, Saffari, and Torr 2011), and Gaussian process regression (Gao et al. 2014). Recently, deep convolutional neural networks (CNNs) have also demonstrated great success in tracking applications. Some exploit the representation power of CNNs by pre-training them on an external large-scale image dataset (Hong et al. 2015; Wang et al. 2015a) while others (Nam and Han 2016) utilize existing tracking videos to capture more domain-specific information. Besides, some correlation filter based tracking methods (Henriques et al. 2015; Danelljan et al. 2014; 2015) have been shown to achieve real-time speed and robust tracking performance.

The motion model generates a set of candidates for the target based on an estimation obtained from the previous frame. Both deterministic and stochastic methods are used. Deterministic methods such as tracking-by-detection (Avidan 2004; Danelljan et al. 2014; Ma et al. 2015) usually adopt the sliding window approach to exhaustively search for the best candidate within a region. On the other hand, stochastic methods such as particle filters (Arulampalam et al. 2002) recursively infer the hidden state of the target and are thus relatively insensitive to local minima. Also, particle filters can easily incorporate changes in scale, aspect ratio and even rotation and skewness due to their efficiency. Additionally, recent work (Ma et al. 2015) found it useful to add a re-detection module to the tracking system, especially in the case of fast motion and out-of-view targets in long-term tracking tasks. While many studies focus on stationary camera setting only, very few consider moving cameras. The authors in (Mei and Porikli 2008) formulated a factorial hidden Markov model for joint tracking and video registration. However, they only performed simple evaluation on toy sequences. A probabilistic framework was proposed in (Choi, Pantofaru, and Savarese 2013) for tracking multiple persons in a 3D coordinate system with a moving camera and (Liu 2016) presented a method for multi-view 3D human tracking. In all of these methods, initial camera information is required for 3D localization.

In recent years, many datasets and corresponding benchmarks have been developed for visual tracking. One milestone is the work by (Wu, Lim, and Yang 2013) which set up a unified benchmark consisting of 50 videos with full annotations. The authors also proposed a novel performance measure and a strict protocol for tracker evaluation. Recently, the benchmark has been extended (Wu, Lim, and Yang 2015). Another representative work is the Visual Object Tracking (VOT) challenge (Kristan et al. 2014). The major difference is their evaluation metric. These recent efforts have substantially advanced the development of visual tracking especially with respect to the appearance model. Another interesting work (Song and Xiao 2013) has built an RGB-D dataset so that the depth information can also be utilized for tracking. Other benchmark datasets include NUS-PRO (Li et al. 2015) and ALOV++ (Smeulders et al. 2014). To summarize, most existing benchmarks have been designed for studying the challenging appearance changes in tracking, but our dataset makes an attempt to open up a new direction that focuses more on the motion model.

Drone Tracking Benchmark

To construct the drone tracking dataset, we have collected 70 video sequences with RGB data and manually annotated the ground-truth bounding boxes in all video frames. Some of the videos are recorded on a university campus by a DJI Phantom 2 Vision+ drone. These sequences mostly focus on tracking people and cars with specially designed camera motion. The other videos are collected from YouTube to add more diversity to both the target appearance and the scene itself. The original resolution of each video frame is 1280×720.

Figure 1: First frames of some video sequences in our dataset. A bounding box around the target is shown for each video.

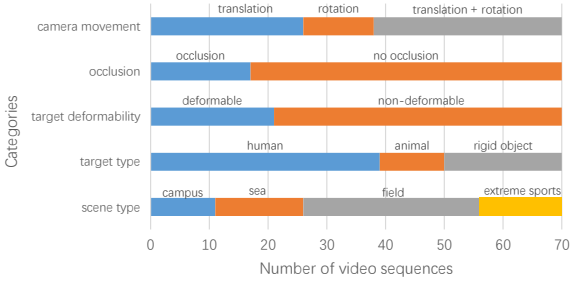


Figure 2: Statistics of our drone tracking dataset.

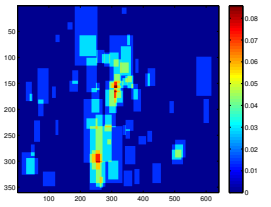
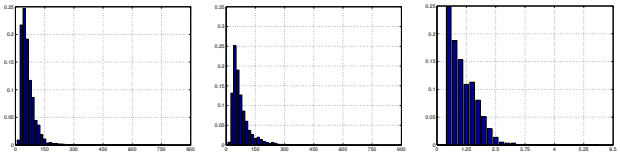


Figure 3: Distribution of ground-truth bounding box location over all sequences.



(a) Width. (b) Height. (c) Aspect ratio.

Figure 4: Distribution of ground-truth bounding box size over all sequences.

Dataset Statistics

A good benchmark dataset should exhibit high diversity and low bias. Fig. 1 shows the first frames of some examples from our drone tracking benchmark, with a bounding box around the target shown for each video. Fig. 2 summarizes the statistics of our dataset. We further analyze the dataset according to the following aspects:

Motion type: The biggest distinction from the existing tracking datasets such as VOT (Kristan et al. 2014) and VTB50 (Wu, Lim, and Yang 2013) is that our dataset covers different types of camera motion including both translation and rotation. This distinction poses great challenges to the conventional motion models used in the existing trackers.

Occlusion: Our dataset covers both short-term and long-term occlusion. It also contains some highly challenging cases in which there is high similarity between the occluder and the target in addition to having fast camera motion.

Target type: We divide the targets into three types: humans, animals, and rigid objects. Rigid objects such as cars and boats can only translate or rotate. As for animals and humans, their movements have a higher degree of freedom and usually consist of deformation which makes tracking more difficult.

Scene type: Different from the PTB (Song and Xiao 2013) dataset, all the videos in our dataset capture outdoor scenes. The background differs considerably from sequence to sequence. For example, while the grass background is simple, the one of extreme water sports is more complex.

Bounding box distribution: Fig. 3 and 4 show the location and size distributions of the ground-truth bounding boxes over all sequences. The box location distribution is computed as a normalized histogram in a 640×360 image space where the value in each pixel denotes the probability for a bounding box to cover that pixel. The box size distribution is computed with respect to the width, height, and aspect ratio over all sequences. The dataset covers different objects with large variations in size and aspect ratios.

For better evaluation and analysis of the strengths and weaknesses of different tracking methods, we follow the VTB50 (Wu, Lim, and Yang 2013) to categorize the sequences by annotating them with different attributes. We remove the attributes with too few videos and add the following three attributes: "aspect ratio variation (ARV)", "fast camera motion (FCM)", and "similar objects around (SOA)". The first one measures the ability of a tracker to handle deformation and rotation while the other two measure the performance of the motion model in extreme cases. The attribute distribution of the entire dataset is shown in Fig. 5.

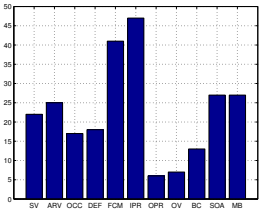


Figure 5: Attribute distribution of the entire dataset. Each subset of sequences corresponds to one of the attributes, namely: scale variation (SV), aspect ratio variation (ARV), occlusion (OCC), deformation (DEF), fast camera motion (FCM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background cluttered (BC), similar objects around (SOA), and motion blur (MB).

Evaluation

For the evaluation of tracking methods, we follow (Wu, Lim, and Yang 2013) to use the success and precision plots for quantitative analysis. Both plots show the percentage of successfully tracked frames with respect to the threshold. The success plot thresholds the intersection over the union (IOU) metric and the precision plot thresholds the center location error. To rank the trackers, two types of ranking scores are used: the Area Under the Curve (AUC) metric for the success plot and the representative precision score at the threshold of 20 for the precision plot. Different from (Wu, Lim, and Yang 2013), here we only keep the one-pass evaluation (OPE) setting since it is the most common setting in practical applications.

Methods

In this section, we consider the design of special motion models to address the issue of camera motion in online tracking. We first present our approximate camera model from the viewpoint of multiple view geometry. Based on the camera model, we then propose a simple yet effective baseline method for camera motion correction during tracking. We note that it is general enough for all existing trackers.

Camera Model

A camera model describes a mapping between the 3D world (object space) and a 2D image plane. Various camera models

have been well studied in the topic of multiple view geometry (Hartley and Zisserman 2003). The most widely used one is the general pinhole camera model. (Hoiem, Efros, and Hebert 2008) proposed a simplified camera model which assumes that all the objects of interest rest on the ground plane. This simplified camera model has been used for tracking ground-plane objects such as cars and pedestrians (Choi, Pantofaru, and Savarese 2013). However, all the camera models need initial information about the camera to infer the 3D location of the object and for camera calibration. Unfortunately, such information is not usually available in many object tracking applications.

Here we take a different approach by parameterizing the camera directly in the 2D image plane. Note that since the camera on a drone is usually far away from the target, we may simply ignore any differences in depth between the target and the background clues and thus assume that the captured frames can be regarded as different planar objects. Then, from the viewpoint of two view geometry, these planes are related by a projective transformation which is also known as 2D homography. Mathematically, let $\mathbf{g}_t$ and $\mathbf{g}_{t-1}$ denote the homogeneous coordinates of a static feature point in frame t and $t-1$, respectively. We can then parameterize the camera model by a transformation matrix $\mathbf{H}$:

$$\mathbf{g}_t = \mathbf{H}\,\mathbf{g}_{t-1}, \quad \mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & 1 \end{bmatrix}. \qquad (1)$$

Note that since our main motivation for introducing the camera model is to roughly guide the search area for tracking instead of determining the exact location, the above homography approximation works well in practice. Consequently, we only need to estimate the transformation matrix $\mathbf{H}$. Since initial camera information is no longer needed, this approach is more general in its applicability.

Baseline Method

In conventional tracking methods, only the target motion is modeled. Let $\mathbf{z}_t$ and $\mathbf{z}_{t-1}$ denote the target coordinates in frame t and $t-1$, respectively. The motion model is simply expressed as:

$$\mathbf{z}_t = \mathbf{z}_{t-1} + \Delta\mathbf{z}_t, \qquad (2)$$

Particle filter based methods model $\Delta\mathbf{z}_t$ by a Gaussian distribution while sliding window based methods model $\Delta\mathbf{z}_t$ by a uniform distribution in a local area. This simple motion model works well under normal scenarios. However, in extreme cases such as drone tracking, modeling $\Delta\mathbf{z}_t$ alone is inadequate. Specifically, assuming a small $\Delta\mathbf{z}_t$ will simply miss the target in the next frame while assuming a large $\Delta\mathbf{z}_t$ will increase the risk of drifting.

Based on the camera assumption above, we can express the new motion model as a combination of the camera projection and the target motion:

$$\mathbf{z}_t = \mathbf{H}_t\,\mathbf{z}_{t-1} + \Delta\mathbf{z}_t, \qquad (3)$$

where $\mathbf{H}_t$ denotes the camera motion and $\Delta\mathbf{z}_t$ is the location displacement due to motion of the target. Once we get a reasonable estimate of the camera motion $\mathbf{H}_t$, the target motion displacement can be estimated more accurately in a local area.

Note that this baseline method can easily be incorporated into all existing tracking methods. Specifically, we first estimate the homography $\mathbf{H}_t$ by feature point matching as in (Fischler and Bolles 1981). Then the previous target location estimation is projected to the current image plane by $\mathbf{H}_t$. For sliding window based trackers, a local area centered on the transformed target coordinates will be searched. For particle filter based trackers, all the sample candidates which are maintained will be transformed to the current image plane. Other than these changes, each tracker still works in the same way.

Experiments

In this section, we present an extensive evaluation of the recent state-of-the-art trackers and their motion model variants on the drone tracking dataset.

Dataset Validation

To understand the performance gap between the traditional tracking setting and the drone tracking setting, we first validate the proposed drone tracking benchmark (DTB) by conventional state-of-the-art tracking approaches. Specifically, we choose the following representative trackers:

- three correlation filter based approaches: KCF (Henriques et al. 2015), DSST (Danelljan et al. 2014), SRDCF (Danelljan et al. 2015)
- a color-based discriminative tracker: DAT (Possegger, Mauthner, and Bischof 2015)
- a competitive particle filter based approach: HOG-LR (Wang et al. 2015b)
- an expert ensemble based approach: MEEM (Zhang, Ma, and Sclaroff 2014)
- two deep learning based approaches: SO-DLT (Wang et al. 2015a), MDNet (Nam and Han 2016)

All of these approaches have shown excellent performance in the VTB50 dataset. Their overall performance for our DTB dataset is shown in Fig. 6. We can see that the performance of these state-of-the-art trackers is significantly worse when compared to their performance in the VTB50 dataset, showing that DTB is quite a challenging dataset even for the top performing trackers.

To further identify the most challenging factors of our new dataset, we also show the attribute-based performance of each tracker in Fig. 7 where each group corresponds to a different attribute. In general, deep learning based trackers outperform other trackers by a large margin, especially for MDNet, which utilizes additional tracking sequences for training. In terms of the appearance challenge, we find that all the trackers get the lowest score in the attribute "out-of-plane rotation (OPR)", which means that significant target deformations and rotations are still the most challenging part of the appearance model. We note that deep learning feature representation is superior to traditional handcrafted features. Some of the trackers (MEEM, KCF, DSST) do not account

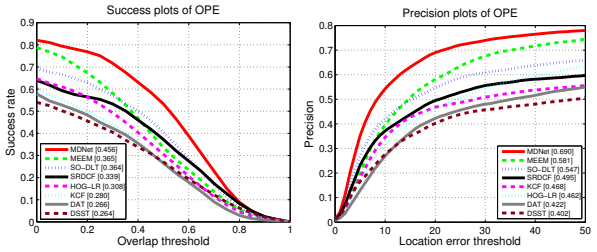


Figure 6: Success and precision plots of conventional tracking approaches for DTB. The performance score of each tracker is shown in the legend.

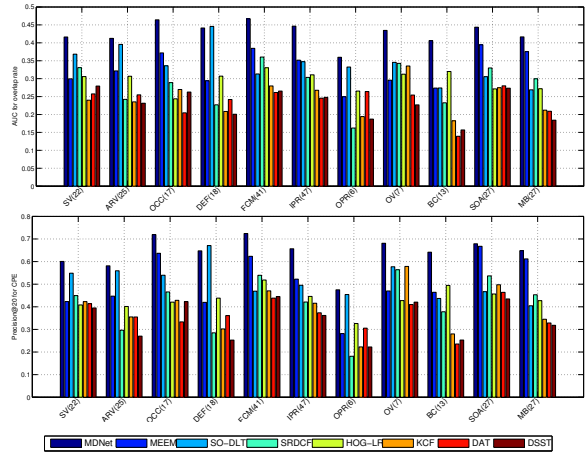


Figure 7: Average performance ranking scores of the trackers on different subsets of the test sequences in DTB.

for scale variation or aspect ratio variation, so they perform poorly in the "scale variation (SV)" and "aspect ratio variation (ARV)" attributes. As for the motion model related attribute, we can see that particle filter based methods (MDNet, HOG-LR) are better than local sliding window based methods (SO-DLT, DAT, DSST, KCF). One reason is that the particle filter framework can explore the state space in a more efficient way. Nevertheless, none of them is satisfactory in the drone tracking setting. We believe that a more sophisticated motion model is an indispensable part of the drone tracking system.

Motion Model Evaluation

In this part, we empirically evaluate the proposed motion model by integrating it into the above state-of-the-art trackers. Note that the focus of this paper is on demonstrating the benefit of the proposed motion model in a tracking system. Thus comparison is done by fixing the other parts of the trackers.

Implementation Details The static feature points are detected using the SURF detector (Bay et al. 2008). After that, we use feature matching and then apply a RANSAC (Fischler and Bolles 1981) step to estimate the homography $\mathbf{H}$. For comparison, we also consider a more restricted form of $\mathbf{H}$, the affine transformation, in the following experiments. In the feature point matching process, we discard feature points which are near the current target location estimation. The maximum number of feature points is set to 100. The tracker implementation is based on the code from the original authors with only modification made to the motion model.

Quantitative Evaluation The overall performance scores of the trackers and their motion model variants in DTB are shown in Tab. 1. We can see that the homography generally gives more robust and accurate estimation than the affine transformation, especially for local sliding window based approaches (DSST, DAT, KCF, SO-DLT). We notice that for SRDCF and MDNet, the affine estimation performs comparably or even better than the homography motion model. This seems to imply that with a very strong observation model, both forms are reasonable approximations of the camera motion in practice. Under the homography camera motion assumption, all the trackers with the new motion model outperform the original ones by about 8% on average. The improvement is more significant for local sliding window based methods such as SO-DLT and DSST, where the homography motion model can achieve more than a 15% performance gain. This demonstrates both the importance of the motion model and the effectiveness of the baseline method in the drone tracking setting.

	Overlap AUC			Precision@20		
	original	affine	homography	original	affine	homography
MDNet	0.456	**0.511**	0.499	0.690	**0.766**	0.749
MEEM	0.365	0.323	**0.379**	0.581	0.500	**0.616**
SO-DLT	0.364	0.350	**0.416**	0.547	0.533	**0.632**
SRDCF	0.339	0.362	**0.367**	0.495	0.544	**0.544**
HOG-LR	0.308	0.310	**0.343**	0.462	0.478	**0.524**
KCF	0.280	0.263	**0.302**	0.468	0.433	**0.498**
DAT	0.266	0.231	**0.279**	0.422	0.372	**0.447**
DSST	0.264	0.273	**0.330**	0.402	0.420	**0.512**

Table 1: Overall performance scores of different trackers and their motion model variants in DTB. The best one is in bold.

To gain more insights, we further compare the performance of the homography motion model with conventional models under different individual sequence attributes in DTB. Due to space limitations, we only show the percentage improvement of the proposed methods in terms of the AUC score in Tab. 2. We first observe that the proposed baseline method is very effective for handling fast camera motion. In such circumstances, the conventional approach will simply miss the right search area. It might appear that the conventional model can tackle it by adjusting the search step size. However, searching for a larger candidate region without considering the camera motion also increases the chance of drifting. On the contrary, with the camera motion taken into consideration, the proposed method can guide the tracker to the right local search area and thus perform better. This can be verified in the "similar objects around (SOA)" attribute. Besides, the performance gain in the "motion blur (MB)" attribute also demonstrates that our method is robust to cam-

	SV(22)	ARV(25)	OCC(17)	DEF(18)	FCM(41)	IPR(47)	OPR(6)	OV(7)	BC(13)	SOA(27)	MB(27)
MDNet	6.97%	2.35%	2.56%	0.39%	16.23%	7.52%	-2.61%	-1.36%	21.55%	21.30%	29.50%
MEEM	-1.23%	-1.74%	10.16%	2.72%	4.32%	0.85%	-7.41%	2.74%	-1.10%	3.37%	4.53%
SO-DLT	5.67%	2.93%	11.19%	-9.13%	36.49%	6.99%	-23.92%	-11.11%	11.64%	39.28%	48.42%
SRDCF	0.27%	-2.52%	23.19%	-0.18%	14.29%	8.00%	-2.71%	-1.05%	19.00%	23.70%	25.65%
HOG-LR	14.36%	-1.34%	32.69%	0.00%	12.11%	8.08%	-10.40%	-25.66%	-5.41%	30.61%	20.23%
KCF	6.29%	-2.59%	-0.48%	-12.52%	27.11%	3.96%	-22.82%	-30.59%	-8.28%	32.61%	34.70%
DAT	-6.64%	-4.28%	15.69%	-8.56%	15.30%	1.34%	-7.50%	-10.77%	11.93%	15.72%	18.65%
DSST	12.46%	0.60%	16.29%	7.03%	44.82%	16.50%	-5.56%	5.24%	22.05%	36.69%	72.24%

Table 2: Per-attribute comparison between trackers and their homography motion model variants. The number shows the percentage improvement in the AUC score of the homography motion model over the original ones. A positive value indicates that the proposed motion model is better.

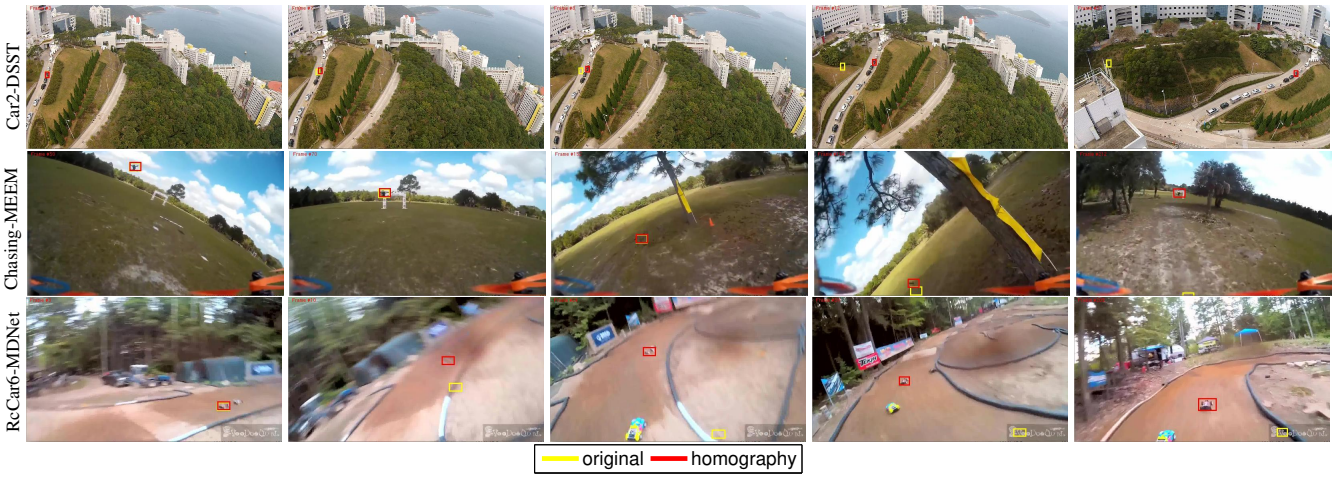


Figure 8: Qualitative comparison of trackers with different motion models on some challenging sequences in DTB.

era shakes. Note that the proposed motion model might fail when both the estimation of homography is too noisy and the observation clue in the target tracker is also distracted by noise. The compounding error in these cases will lead to incorrect tracking results. For the newly introduced attribute "aspect ratio variation (ARV)", we notice a small performance drop in a few extreme cases with serious deformation. For the "deformation (DEF)" and "out-of-plane rotation (OPR)" attributes, object deformation and out-of-plane rotation pose significant challenges to the observation model of the tracker. For the "out-of-view (OV)" attributes, simply estimating the camera motion cannot predict where the target will reappear. As a result, the proposed motion model might hurt the actual performance in those cases.

Qualitative Results To have a more comprehensive understanding of both the dataset and the proposed approach, we show a qualitative comparison of different methods on some challenging videos in Fig. 8.

The first row shows an example of tracking a car with very similar objects around, where the drone camera is rotating sharply. DSST fails immediately when the camera starts to rotate while the homography motion model can successfully track the target to the very end. The second row demonstrates a more challenging case where the goal is to track a flying drone. Both the target and the camera are moving fast and randomly. Although the MEEM tracker itself contains a re-detection module, it still loses the target when the pursuing drone makes a sharp turn in frame #164. On the other hand, the modified model can handle it quite well. In the last row, we show a more difficult example where the camera motion causes significant blur of the target object. In this case, only the deep learning based tracker with additional homography estimation succeeds to track the target.

Conclusion and Future Work

In this paper, we have explored the potential of conducting visual tracking on the drone platform. We propose a unified drone tracking benchmark which covers a variety of videos captured by drone cameras. To address the challenging issue of abrupt camera motion, we design simple baselines to model the camera motion by projective transformation based on background feature clues. We present an extensive comparison of recent state-of-the-art trackers and their motion model variants on the drone tracking benchmark. The result demonstrates that by explicitly modeling camera motion, trackers can achieve substantial performance improvement under the proposed motion model.

Although our proposed baseline methods are effective,

some cases of failure do exist. For example, the camera estimation is based on traditional low-level feature point detection which is noisy and even wrong in some circumstances. How to design convolutional neural networks to learn more accurate camera motion on video data is an interesting problem. Currently in the baseline method, camera estimation works in a standalone way. Integrating camera estimation and target tracking in a coherent learning framework is expected to help. We will pursue research in these directions in our future work.

Acknowledgments

This research has been supported by General Research Fund 16207316 from the Research Grants Council of Hong Kong.

References

Alt, N.; Hinterstoisser, S.; and Navab, N. 2010. Rapid selection of reliable templates for visual tracking. In *CVPR*, 1355–1362. IEEE.

Arandjelović, O. 2015. Automatic vehicle tracking and recognition from aerial image sequences. In *AVSS*, 1–6.

Arulampalam, M.; Maskell, S.; Gordon, N.; and Clapp, T. 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing* 50(2):174–188.

Avidan, S. 2004. Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(8):1064–1072.

Babenko, B.; Yang, M.; and Belongie, S. 2011. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(8):1619–1632.

Bay, H.; Ess, A.; Tuytelaars, T.; and Van Gool, L. 2008. Speeded-up robust features (SURF). *Computer Vision and Image Understanding* 110(3):346–359.

Choi, W.; Pantofaru, C.; and Savarese, S. 2013. A general framework for tracking multiple people from a moving camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(7):1577–1591.

Danelljan, M.; Häger, G.; Khan, F. S.; and Felsberg, M. 2014. Accurate scale estimation for robust visual tracking. In *BMVC*.

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

Fischler, M. A., and Bolles, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6):381–395.

Floreano, D., and Wood, R. J. 2015. Science, technology and the future of small autonomous drones. *Nature* 521(7553):460–466.

Gao, J.; Ling, H.; Hu, W.; and Xing, J. 2014. Transfer learning based visual tracking with Gaussian processes regression. In *ECCV*, 188–203.

Grabner, H.; Grabner, M.; and Bischof, H. 2006. Real-time tracking via on-line boosting. In *BMVC*, 47–56.

Grabner, H.; Leistner, C.; and Bischof, H. 2008. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 234–247.

Hare, S.; Saffari, A.; and Torr, P. H. 2011. Struck: Structured output tracking with kernels. In *ICCV*, 263–270.

Hartley, R., and Zisserman, A. 2003. *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Henriques, J. F.; Caseiro, R.; Martins, P.; and Batista, J. 2015. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(3):583–596.

Hoiem, D.; Efros, A. A.; and Hebert, M. 2008. Putting objects in perspective. *International Journal of Computer Vision* 80(1):3–15.

Hong, S.; You, T.; Kwak, S.; and Han, B. 2015. Online tracking by learning discriminative saliency map with convolutional neural network. In *ICML*, 597–606.

Kristan, M.; Pflugfelder, R.; Leonardis, A.; Matas, J.; Čehovin, L.; Nebehay, G.; Vojíř, T.; Fernandez, G.; Lukežič, A.; Dimitriev, A.; et al. 2014. The visual object tracking VOT2014 challenge results. In *ECCV Workshops*, 191–217. Springer.

Li, A.; Lin, M.; Wu, Y.; Yang, M.; and Yan, S. 2015. NUS-PRO: A new visual tracking challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* PP(99):1–1.

Liu, X. 2016. Multi-view 3d human tracking in crowded scenes. In *AAAI*.

Ma, C.; Yang, X.; Zhang, C.; and Yang, M.-H. 2015. Long-term correlation tracking. In *CVPR*, 5388–5396.

Matthews, I.; Ishikawa, T.; and Baker, S. 2004. The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(6):810–815.

Mei, X., and Ling, H. 2009. Robust visual tracking using l_1 minimization. In *ICCV*, 1436–1443.

Mei, X., and Porikli, F. 2008. Joint tracking and video registration by factorial hidden Markov models. In *ICASSP*, 973–976. IEEE.

Nam, H., and Han, B. 2016. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*.

Possegger, H.; Mauthner, T.; and Bischof, H. 2015. In defense of color-based model-free tracking. In *CVPR*, 2113–2120.

Ross, D.; Lim, J.; Lin, R.; and Yang, M. 2008. Incremental learning for robust visual tracking. *International Journal of Computer Vision* 77(1):125–141.

Smeulders, A. W.; Chu, D. M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; and Shah, M. 2014. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(7):1442–1468.

Song, S., and Xiao, J. 2013. Tracking revisited using RGBD camera: Unified benchmark and baselines. In *ICCV*, 233–240. IEEE.

Wang, N.; Li, S.; Gupta, A.; and Yeung, D.-Y. 2015a. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587*.

Wang, N.; Shi, J.; Yeung, D.-Y.; and Jia, J. 2015b. Understanding and diagnosing visual tracking systems. In *ICCV*, 3101–3109.

Wang, N.; Wang, J.; and Yeung, D.-Y. 2013. Online robust non-negative dictionary learning for visual tracking. In *ICCV*, 657–664.

Wu, Y.; Lim, J.; and Yang, M.-H. 2013. Online object tracking: A benchmark. In *CVPR*, 2411–2418. IEEE.

Wu, Y.; Lim, J.; and Yang, M. 2015. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(9):1834–1848.

Zhang, J.; Ma, S.; and Sclaroff, S. 2014. MEEM: Robust tracking via multiple experts using entropy minimization. In *ECCV*, 188–203. Springer.