Review of “High-Speed Tracking wit Kernelized Correlation Filters”

Heng Fan

1. Paper summary

This paper [1] proposes a kernelized correlation filter tracking algorithm training a robust classifier with thousands of dense samples or patches. To deal with the problems of storage and computation with these huge number of samples, this paper first shows these large number of samples can be formulated into a circulant matrix. Then the tracking task is formulated as a regression problem with the circulant matrix, which is actually proven to be equivalent to a linear correlation filter. To derive a more powerful regressor, kernel trick is introduced to transform the linear correlation to kernelized correlation filter. With the help of Discrete Fourier Transform (DFT), the kernelized correlation filter can be very efficiently solved. To represent object with stronger features such as HoG, the authors also extend the single correlation to the multi-channel correlation filters, result in significant improvement in performance. Experiments on popular tracking benchmark OTB2013 demonstrate the outstanding performance of this method.

2. Experimental results

The authors conduct experiments on the popular tracking benchmark OTB2013 [x], which contains 50 videos and each of them are manually annotated. Experimental results demonstrate that the proposed KCF tracking algorithm outperforms other state-of-the-art trackers (at that moment), as shown in Figure 1. Besides, the authors also report the speed of their KCF tracker and comparisons to different variants and other tracking methods, as shown in Table 1.

Tab. 1: Ablation experimental results and further comparison.

	Algorithm	Feature	Mean precision (20 px)	Mean FPS
Proposed	KCF	HOG	**73.2%**	172
	DCF		**72.8%**	**292**
	KCF	Raw pixels	56.0%	154
	DCF		45.1%	278
Other algorithms	Struck [7]		65.6%	20
	TLD [4]		60.8%	28
	MOSSE [9]		43.1%	**615**
	MIL [5]		47.5%	38
	ORIA [14]		45.7%	9
	CT [3]		40.6%	64

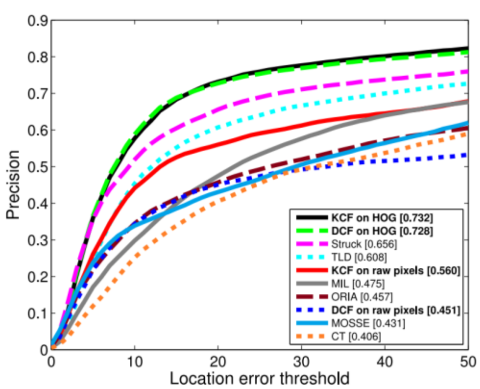

Fig. 1: Comparison with other trackers.

3. Contribution

3.1 A fast while accurate tracking framework.

This paper introduces the kernelized correlation filter tracking method by exploring the circulant structure of thousands of dense samples. With the help of DFT, the correlation filter can be very efficiently solved.

Compared to conventional tracking algorithms, the proposed KCF tracker runs at several hundred frames per second, and still obtains the top performance at that moment.

3.2 Detailed analysis of KCF tracking algorithm.

This paper gives the very detailed analysis of KCF tracking method. For example, the authors conduct various experiments on different variants of KCF, such as using different features (raw pixels or HoG) and detailed comparisons between linear correlation and kernelized correlation filter. All these make the method more easily understood. Thus, I think this is also a contribution of this work.

3.3 Excellent performance and code release.

On the large-scale benchmark, the KCF tracker achieves the best performance (at that moment) and runs very efficiently (hundreds of frames per second), which leaves great room for future research on real-time visual tracking. In addition, they also release their implementation, inspiring a lot of later works based on correlation filters.

4. Criticism

4.1 No scale estimation.

In visual tracking, scale variation is a common factor which results in tracking drift and even failure. Though the proposed KCF tracking algorithm works efficently, it cannot deal with the problem of scale changes because of the limitation of correlation filter. A solution to alleviate this issue is to add an extra scale correlation filter to estimate object scale, at the cost of scarifying efficiency.

4.2 Sensitive to large appearance because of simple hand-crafted feature.

In the KCF tracking, the object appearance is represented with HoG feature. Though using this simple feature allows the KCF tracker to obtain very fast running speed, it is sensitive large appearance changes such as rotation and deformation owing to the limitations of HoG itself. A better way is to replace the simple HoG feature with more robust deep features.

4.3 Too simple update strategy.

To adapt the tracker to object appearance changes during tracking, one needs to update the appearance model. In this paper, the authors just utilize a very simple linear update strategy for appearance model in each frame, which is prone to cause drift. For example, when the object target suffers from occlusion, the update method will update the background into appearance model, resulting in drift and even failure in subsequent tracking.

Reference

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