FAST: Fast and Accurate Scale Estimation for Tracking

Haoyi Ma, *Student Member, IEEE*, Zongli Lin, *Fellow, IEEE*, and Scott T. Acton, *Fellow, IEEE*

***Abstract*—In visual object tracking, robust and accurate scale estimation of a target is a challenging task. Despite the associated computational expense, existing tracking methods cannot accommodate large scale variations. Here, we propose a scale searching scheme that obtains robust and accurate scale estimation by incorporating a novel and robust criterion, the average peak-to-correlation energy, into a multi-resolution translation filter framework. To address the problem of computational expense, we introduce an expeditious search strategy. The resulting system is named FAST: Fast and Accurate Scale estimation for Tracking. Comprehensive evaluation using the publicly available tracking benchmark datasets demonstrates that the proposed scale searching framework can accommodate large scale variation while also yielding computational efficiency.**

***Index Terms*—Average peak-to-correlation energy, correlation filter, principle component analysis, scale estimation, visual object tracking.**

I. Introduction

VISUAL object tracking (VOT) is a classical problem in image processing with numerous applications ranging from video compression to medical imaging [1]–[6]. The generic goal of tracking is to estimate the state of a target in an image sequence given only the state of the target in an initial frame. The tracking problem is challenging due to numerous complicating factors, such as illumination variation, partial and full occlusion. Since VOT is a basic unit of many time-critical computing systems, a given visual tracker should meet strict time and computational requirements. These requirements are particularly acute in embedded and mobile systems where resources are limited.

Recently, the discriminative correlation filter (DCF) has been successfully applied to VOT yielding both high accuracy and high efficiency [7]–[9], [34]–[36]. By exploiting the circular correlation, the correlation filter can be computed efficiently in the Fourier domain using only fast Fourier transforms (FFTs) and pointwise operations. However, the standard DCF based tracker employs a fixed size template and is therefore performance-limited when scale variation is encountered.

Several approaches [10]–[15] have been proposed to generalizing the standard DCF based tracker for scale adaptability. Danelljan *et al.* proposed the discriminative scale space tracker (DSST) to tackle the scale estimation problem by training two separate correlation filters for explicit translation and scale estimation. Furthermore, the fast discriminative scale space tracker (fDSST) was proposed in [10] to increase the speed and accuracy of the DSST. However, with both the DSST and the fDSST, unreliable translation estimation leads to inferior scale estimation, which results in a poor generalization. A kernelized correlation filter with detection proposals (KCFDP) was proposed in [12] to deal with the aspect ratio variation in scale estimation by incorporating a class-agnostic detection proposal method into the standard DCF based tracker. However, the detection proposal method EdgeBox [16] employed in KCFDP tends to miss targets in the presence of various factors, such as background clutter, since the EdgeBox generates proposals that favor bounding boxes with many internal contour segments. Another straightforward approach, the scale adaptive tracker with multiple features (SAMF), was proposed in [14] to accommodate scale variations by extending a standard DCF based tracker to multiple resolutions. However, the SAMF suffers from high computational expense due to an exhaustive scale searching strategy.

The main contributions of this letter can be summarized as follows. First, a scale searching framework is proposed to obtain robust and accurate scale estimation by exploiting a criterion termed the *average peak-to-correlation energy* (APCE). Second, a strategy is designed to reduce the computational cost of our scale searching framework. Third, extensive experiments are conducted using online tracking benchmark (OTB) datasets to validate the performance of our proposed FAST tracker.

As to the organization of the letter, Section II describes our proposed FAST tracker. In particular, the formulation of our baseline tracker multi-channel DCF is treated in Section II-A. Our scale searching scheme is described in Section II-B. The strategies investigated to reduce the computational cost are introduced in Section II-C. The experimental setup and comparison results are presented in Section III. Finally, Section IV concludes the letter. In the spirit of reproducible research, the results and code are available to the public at *https://github.com/haoyihaoyi/FAST*.

II. Proposed Algorithm

A. Multi-channel Discriminative Correlation Filters

FAST leverages the multi-channel DCF. The problem of learning an optimal correlation filter can be formulated as training a classifier using all cyclic shifts of a base sample. The feature channel $l \in \{1, 2, \ldots, c\}$ of the target sample x is denoted as x^l. The objective of learning the correlation filter w consisting of one filter w^l per feature channel can be achieved by minimizing

Manuscript received October 14, 2019; revised December 23, 2019; accepted December 24, 2019. Date of publication December 31, 2019; date of current version January 30, 2020. This work was supported in part by the Army Research Office under Grant W911NF-15-1-0275. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yap-Peng Tan. *(Corresponding author: Scott T. Acton.)*

The authors are with the Charles L. Brown Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904-4743 USA (e-mail: hm5au@virginia.edu; zl5y@virginia.edu; acton@virginia.edu).

Digital Object Identifier 10.1109/LSP.2019.2963147

the objective function

$$\varepsilon = \left\| y - \sum_{l=1}^{c} w^l \star x^l \right\|^2 + \lambda \sum_{l=1}^{c} \| w^l \|^2, \quad (1)$$

where $\star$ denotes the circular correlation and λ is a regularization parameter. The desired output y is a Gaussian function parameterized by the standard deviation. The objective function (1) is minimized by

$$W^l = \frac{A^l}{B + \lambda} = \frac{\overline{Y} X^l}{\sum_{l=1}^{c} \overline{X^l} X^l + \lambda}, \; l \in \{1, 2, \ldots, c\}, \quad (2)$$

where W^l, Y and X^l denote the discrete Fourier transform (DFT) of w^l, y and x^l, respectively, and $\overline{Y}$ and $\overline{X^l}$ denote the complex conjugate of Y and X^l, respectively. To adapt to the target appearance changes, the numerator A^l and the denominator B of (2) are updated as

$$A_t^l = (1 - \eta) A_{t-1}^l + \eta \overline{Y} X_t^l, \quad (3)$$

$$B_t = (1 - \eta) B_{t-1} + \eta \sum_{l=1}^{c} \overline{X_t^l} X_t^l, \quad (4)$$

where t is the frame number and η denotes the learning rate. The response map f_t corresponds to the test sample z_t can be computed as

$$f_t = \mathcal{F}^{-1} \left(\frac{\sum_{l=1}^{c} \overline{A_{t-1}^l} Z_t^l}{B_{t-1} + \lambda} \right), \quad (5)$$

where Z_t^l is the DFT of z_t^l. The translation estimation is obtained by localizing the maximum response value within f_t.

B. The Proposed Scale Searching Scheme of FAST

All the aforementioned methods (described in Section I) employ the naïve maximum response value as the criterion to obtain the scale estimation. However, we argue that the robustness of the approach using the maximum response value will be heavily degraded due to the presence of other factors, such as occlusion and illumination variation. Here, we propose to employ the APCE measure as the criterion to obtain the scale estimation.

The fluctuation and the peak value of the response map can reveal the confidence level of the tracking result [18], [23]. The ideal response map should have only one sharp peak at the true target location while being close to zero in all other areas, which will result in higher localization accuracy and associated APCE measure. On the contrary, a fierce fluctuation of the response map will result in a significant decrease of the APCE measure when encountering challenging scenarios, such as severe occlusion and illumination variation. Hence, the APCE measure, which takes both the peak value and the fluctuation of the response map into consideration, will be more robust than the naïve maximum response value when facing such challenges.

At the testing stage of frame t, k patches with their sizes in $\{h_{t-1} b | b \in \{s_1, s_2, \ldots, s_k\}\}$ are extracted and resized to a fixed template size through bilinear interpolation, where h_{t-1} denotes the size of the searching window of last frame. The response map corresponds to the resized patch is denoted as $f_{t,b}$, and the associated APCE measure can be computed as

$$APCE_{t,b} = \frac{|f_{t,b}^{\max} - f_{t,b}^{\min}|^2}{\text{mean}\left(\sum_{m,n} (f_{t,b}^{m,n} - f_{t,b}^{\min})^2 \right)}, \quad (6)$$

where $f_{t,b}^{\max}$, $f_{t,b}^{\min}$ and $f_{t,b}^{m,n}$ denote the maximum, the minimum and the m-th row n-th column elements of the response map $f_{t,b}$, respectively. The scale estimation is obtained by searching for the response map with the highest APCE measure as

$$scale = \arg\max_b APCE_{t,b}. \quad (7)$$

The translation estimation is obtained by localizing the maximum response value within the response map with the highest APCE measure.

C. Computational Cost Reduction

As shown in Section II-A, the computational cost is dominated by the FFT, and the number of the FFT computations scales linearly with the feature dimension c. We propose to carry out feature dimension reduction to reduce the required number of FFT computations. Our feature dimension reduction strategy is based on standard principle component analysis (PCA).

By the linearity of the Fourier transform, the numerator of (2), given by (3), can be rewritten as $A_t^l = \overline{Y} \mathcal{F}(g_t^l)$, where the target template is denoted as $g_t = (1 - \eta) g_{t-1} + \eta x_{t,scale}$. The target template is employed to construct the projection matrix $P \in \tilde{c} \times c$, which projects the feature onto a $\tilde{c}$-dimensional subspace while preserving the useful information. Under the orthogonality constraint $P_t P_t^T = I$, the projection matrix can be obtained by minimizing the reconstruction error of the target template as

$$\epsilon = \left\| g_t - P_t^T P_t g_t \right\|^2. \quad (8)$$

A solution to (8) can be achieved by performing an eigenvalue decomposition of $O_t = g_t g_t^T$, which denotes the autocorrelation matrix of the target template. The rows of P_t are set as the $\tilde{c}$ eigenvectors, which correspond to the largest eigenvalues. The correlation filter is updated as

$$\tilde{A}_t^l = \overline{Y} \tilde{G}_t^l, \quad (9)$$

$$\tilde{B}_t = (1 - \eta) \tilde{B}_{t-1} + \eta \sum_{l=1}^{\tilde{c}} \overline{\tilde{X}_{t,scale}^l} \tilde{X}_{t,scale}^l, \quad (10)$$

where the compressed training sample and the compressed target template are denoted as $\tilde{X}_{t,scale} = \mathcal{F}\{P_t x_{t,scale}\}$ and $\tilde{G}_t = \mathcal{F}\{P_t g_t\}$, respectively. The compressed test sample patch is computed as $\tilde{Z}_{t,b} = \mathcal{F}\{P_{t-1} z_{t,b}\}$. Equation (5) can then be rewritten as

$$f_{t,b} = \mathcal{F}^{-1} \left(\frac{\sum_{l=1}^{\tilde{c}} \overline{\tilde{A}_{t-1}^l} \tilde{Z}_{t,b}^l}{\tilde{B}_{t-1} + \lambda} \right), \quad (11)$$

where $\tilde{Z}_{t,b}^l$ denotes the feature channel l of the compressed test sample patch $\tilde{Z}_{t,b}$.

Benefiting from the robustness of the APCE criterion, the significant performance gain provides the flexibility to reduce the size of the performed FFTs. Here, the target sample is resized to one-half of the original size if the target sample size in the initial frame is higher than a defined threshold.

III. Experiments

A. Implementation Details

For the scaling pool, we employ the same setting $b \in \{0.985, 0.99, 0.995, 1.0, 1.005, 1.01, 1.015\}$ as in [14]. The feature

TABLE I
OVERALL COMPARISONS ON OTB DATASETS. THE FIRST AND THE SECOND BEST VALUES ARE HIGHLIGHTED IN BOLD AND UNDERLINED

Dataset	Metric	DSST	fDSST	KCFDP	SAMF	FAST
OTB2013	AUC	0.557	0.595	0.575	0.579	**0.621**
	Prec.	0.747	0.802	0.786	0.785	**0.824**
	Speed	29.53	90.80	36.21	27.44	**91.73**
OTB2015	AUC	0.517	0.549	0.543	0.554	**0.571**
	Prec.	0.689	0.722	0.736	0.754	**0.760**
	Speed	27.23	88.66	34.32	26.67	**90.08**

for image representation is obtained by augmenting HOG and grayscale pixel values. The augmented 32-dimensional feature will be projected onto an 18-dimensional subspace as in [22]. All tracking algorithms are implemented with Matlab on an Intel Core i7-7700 K 4.5 GHz CPU with 16 GB RAM. A GeForce GTX 1080 GPU and MatConvNet 1.0-beta25 are used to reproduce the results of the trackers that require GPU implementation.

B. Datasets and Metrics

All trackers are quantitatively evaluated on the online tracking benchmark (OTB) datasets OTB2013 and OTB2015, following the one-pass evaluation (OPE) protocol, which is described in [19], [20]. OTB2013 and OTB2015 contain 50 and 100 challenging image sequences, respectively. The sequences in OTB datasets are categorized by way of attributes, such as scale variation (SV).

To quantitatively evaluate the tracking results, three standard evaluation metrics are employed including the success rate, the precision and the tracking speed in frames per second (fps). Area-under-curve (AUC) scores of the success plots are utilized to indicate the performance of the trackers. The representative precision score for each tracker is reported at the threshold with 20 pixels as the setting in the standard OTB toolkit.

C. DCF-Based Scale Estimation Comparison

1) Overall Comparison: To compare the overall performance of FAST to other scale adaptive variants of the standard DCF based tracker, five trackers (SAMF, KCFDP, DSST, fDSST and our tracker FAST) are applied to the OTB datasets. The overall performance of the trackers is summarized in Table I in terms of the AUC score, the precision score for a threshold of 20 pixels and the average speed (fps). As shown in Table I, FAST obtains the best performance while maintaining the fastest running speed.

In comparison with fDSST, which runs at a comparable speed to FAST and obtains the second best overall performance on OTB2013, FAST improves the AUC score by 2.6% and 2.2% and the precision score by 2.2% and 3.8% on OTB2013 and OTB2015, respectively. Similar to fDSST, standard PCA is employed in FAST to carry out feature dimension reduction to reduce the required number of FFT computations. In contrast to fDSST, to enable robust and accurate scale estimation, FAST exploits the robustness of average peak-to-correlation energy criterion within a multi-resolution translation filter framework. fDSST tackles the scale estimation problem by learning two separate correlation filters for explicit translation and scale estimation. Besides, fDSST only employs a naïve maximum response value as the criterion, with which the robustness will be heavily degraded when scale variation presents with other challenging factors. The comparison between FAST and fDSST validates the effectiveness of our tracking framework and the robustness of the APCE criterion.

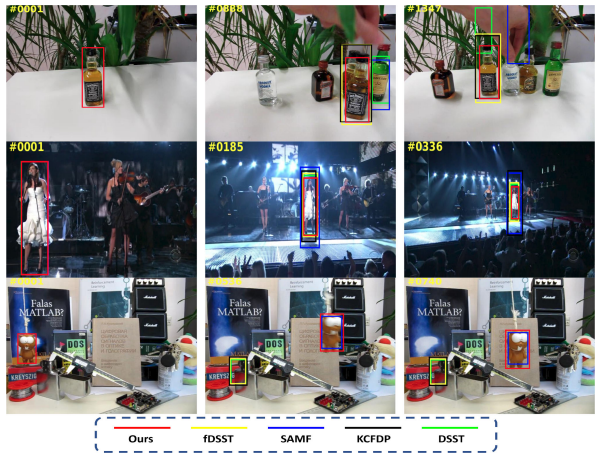

Fig. 1. Comparison of the FAST tracker with other scale adaptive variants of the standard DCF based trackers. Example frames shown are from *liquor* (top row), *singer1* (middle row) and *lemming* (bottom row) sequences that are selected from online tracking benchmark datasets involving scale variation.

TABLE II
ATTRIBUTED-BASED COMPARISONS ON THE SCALE VARIATION (SV) ATTRIBUTE OF OTB DATASETS

Dataset	Metric	DSST	fDSST	KCFDP	SAMF	FAST
OTB2013 (SV)	AUC	0.544	0.564	0.540	0.507	**0.579**
	Prec.	0.736	0.765	0.744	0.723	**0.772**
OTB2015 (SV)	AUC	0.482	0.507	0.510	0.507	**0.533**
	Prec.	0.654	0.673	0.701	0.719	**0.721**

In comparison with SAMF, which also utilizes a multi-resolution translation filter framework, FAST improves the AUC score by 4.2% and 1.7% and the precision score by 3.9% and 0.6% on OTB2013 and OTB2015, respectively. The performance improvement demonstrates the robustness of APCE as SAMF employs other discriminative features (grayscale pixel values, HOG and color names).

2) Attributed-Based Comparison: To validate the scale adaptability of our proposed method, we compare FAST with other scale adaptive DCF variants on the scale variation attribute of OTB datasets. OTB2013 and OTB2015 contain 28 and 66 sequences that involve scale variation challenges, respectively. As shown in Fig. 1 in Section I, our approach significantly increases the robustness and accuracy by estimating the target size accurately. The performance on the scale variation attribute is summarized in Table III.

As shown in Table II, FAST obtains the best performance on the scale variation attribute. In comparison with fDSST, our method improves the AUC score by 1.5% and 2.6% and the precision score by 0.7% and 4.8% on OTB2013 and OTB2015, respectively. The remarkable performance improvement of FAST shows that our method is superior in scale estimation despite the lowest computational cost.

In comparison with SAMF, which employs the multi-resolution translation filter framework, our method improves the AUC score by 7.2% and 2.6% and the precision score by 4.9% and 0.2% on OTB2013 and OTB2015, respectively. The comparison between FAST and SAMF exhibits the effectiveness and robustness of APCE in the scale variation scenario.

TABLE III
ATTRIBUTED-BASED COMPARISON WITH REPRESENTATIVE STATE-OF-THE-ART TRACKERS ON 11 ATTRIBUTES OF OTB2013 AND OTB2015 BENCHMARK SEQUENCES IN TERMS OF THE AUC SCORE OF THE SUCCESS PLOT. THE FIRST AND THE SECOND BEST VALUES ARE HIGHLIGHTED IN BOLD AND UNDERLINED

	OTB2013												OTB2015											
Method	Overall	SV	MB	IPR	OPR	FM	DEF	LR	OV	BC	IV	OCC	Overall	SV	MB	IPR	OPR	FM	DEF	LR	OV	BC	IV	OCC
CFNet_conv2	0.605	**0.600**	0.546	0.560	0.574	0.532	0.556	0.486	0.481	0.603	0.541	0.563	0.564	**0.544**	0.557	0.567	0.531	0.545	0.463	**0.590**	0.414	0.538	0.518	0.534
CF2	0.606	0.531	**0.621**	**0.584**	0.588	**0.581**	0.629	**0.557**	0.575	**0.623**	0.562	0.608	0.566	0.493	**0.575**	**0.572**	0.538	**0.553**	0.526	0.424	0.474	**0.582**	0.541	0.553
HDT	0.590	0.513	0.572	0.563	0.573	0.544	0.615	0.509	0.555	0.589	0.533	0.585	0.555	0.487	0.554	0.545	0.523	0.528	**0.527**	0.407	0.440	0.544	0.506	0.544
HCFT	0.597	0.531	0.583	0.569	0.577	0.555	0.605	**0.557**	0.575	**0.623**	0.544	0.593	0.558	0.493	0.560	0.556	0.524	0.543	0.516	0.424	0.474	**0.582**	0.518	0.537
LCT+	0.619	0.553	0.488	0.578	**0.613**	0.509	**0.645**	0.286	0.594	0.587	0.570	0.612	0.557	0.497	0.518	0.553	0.523	0.512	0.480	0.330	0.452	0.542	0.541	0.528
TLD	0.434	0.421	0.390	0.411	0.415	0.407	0.369	0.309	0.457	0.345	0.393	0.396	0.424	0.396	0.428	0.432	0.388	0.420	0.342	0.372	0.351	0.346	0.392	0.375
MEEM	0.558	0.498	0.505	0.522	0.547	0.528	0.538	0.360	0.606	0.569	0.516	0.537	0.525	0.479	0.532	0.526	0.514	0.515	0.474	0.355	**0.488**	0.514	0.492	0.518
STRUCK	0.472	0.425	0.426	0.441	0.430	0.457	0.389	0.372	0.459	0.458	0.425	0.410	0.460	0.414	0.456	0.457	0.414	0.453	0.377	0.347	0.374	0.425	0.398	0.423
RPT	0.571	0.538	0.547	0.558	0.546	0.541	0.519	0.363	0.576	0.614	0.546	0.521	0.532	0.489	0.499	0.522	0.504	0.517	0.481	0.358	0.475	0.572	0.516	0.484
TGPR	0.501	0.418	0.426	0.476	0.482	0.390	0.505	0.370	0.442	0.522	0.480	0.481	0.457	0.401	0.406	0.461	0.439	0.395	0.447	0.378	0.373	0.440	0.458	0.453
FAST	**0.621**	0.579	0.511	0.566	0.608	0.516	0.637	0.407	**0.663**	0.588	**0.603**	**0.627**	**0.571**	0.533	0.520	0.538	**0.544**	0.513	0.509	0.497	0.478	0.579	**0.582**	**0.571**

D. State-of-the-art Comparison

To demonstrate the superior performance of our approach, we compare our tracker FAST with 10 state-of-the-art methods, which can be broadly categorized as follows:

- Deep learning based trackers, such as HDT [28] and HCFT [32], which employs deep features in correlation filter framework, and CFNet_Conv2 [24], which trains the correlation filter end-to-end with ILSVRC15-VID containing almost 4500 videos with a total of more than one million annotated frames.
- Trackers which are designed for long-term tracking by employing both short-term tracker and online classifier, including LCT+ [25] and TLD [30].
- Representative trackers that employ single or multiple online classifiers, including MEEM [29], TGPR [26], and STRUCK [33] methods.
- Representative part-based tracking (RPT) methods [27] which exploit reliable patches.

Note that we employ publicly available code of compared trackers or copy the results from the original paper for fair comparison.

All tracking algorithms are evaluated on the OTB2013 and OTB2015 datasets, and only the overall and attributed-based AUC scores of all trackers are reported in Table III for the sake of simplicity.

1) Overall Performance: As shown in Table III, the FAST tracker achieves the best performance in terms of the AUC score. Compared to LCT+, which obtains the second best AUC score on OTB2013, FAST improves the AUC score by 0.2% and 1.4% on OTB2013 and OTB2015, respectively. In comparison with CF2, which achieves the second best AUC score on OTB2015, FAST improves the AUC score by 1.5% and 0.4% on OTB2013 and OTB2015, respectively. Although LCT+ and CF2 obtain favorable performance compared to our tracker, the speed of LCT+ and CF2 is only around 20 fps and 15 fps, respectively, FAST yields computational efficiency and operates at over 90 fps.

2) Attributed-Based Comparison: In terms of the scale variation attribute, FAST obtains the second-best AUC score, which is 2.1% and 1.1% lower than the AUC score of CFNet_conv2 on OTB2013 and OTB2015, respectively. In 10 out of 11 attributes, FAST obtains favorable performance with the exception being low resolution challenge. These promising results suggests the robustness of the APCE measure and the effectiveness of our tracking framework when facing different challenging scenarios.

E. Internal Analysis of the Proposed Approach

To investigate the impact of our proposed scale estimation module and computational cost reduction module, we propose three different variants of our proposed tracker FAST: the baseline tracker (DCF), DCF with only the scale estimation module denoted by DCF_se, and DCF with only the computational cost reduction module denoted by DCF_ccr. The performance of the proposed FAST tracker and its three variants are summarized in Table IV.

TABLE IV
OVERALL COMPARISONS OF FAST TRACKER AND ITS THREE VARIANTS ON OTB DATASETS. THE FIRST AND THE SECOND BEST VALUES ARE HIGHLIGHTED IN BOLD AND UNDERLINED

Dataset	Metric	DCF	DCF_se	DCF_ccr	FAST
OTB2013	AUC	0.522	0.592	0.524	**0.621**
	Prec.	0.727	0.777	0.731	**0.824**
	Speed	175.08	50.24	**279.94**	91.73
OTB2015	AUC	0.486	0.558	0.485	**0.571**
	Prec.	0.692	0.729	0.696	**0.760**
	Speed	160.5	48.86	**266.13**	90.08

As shown in the comparison between DCF_se and DCF along with the comparison between FAST and DCF_ccr, our proposed scale estimation module improves both the AUC score and the precision score significantly, which evidences the effectiveness of our proposed scale estimation module. The comparison between DCF_ccr and DCF together with the comparison between FAST and DCF_se show that our proposed computational cost reduction module can improve the tracking speed remarkably without sacrificing the accuracy. Hence, both the scale estimation module and the computational cost reduction module are key parts of the proposed FAST tracker.

IV. CONCLUSION

In this letter, an efficient and effective scale searching framework is proposed to obtain a robust and accurate scale estimation for visual object tracking by incorporating a novel and robust criterion named average peak-to-correlation energy into the multi-resolution translation filter framework. A dimensionality reduction strategy is proposed and validated to reduce the computational cost of the FAST tracker. Comprehensive evaluations on the online tracking benchmark datasets validate the effectiveness and efficiency of FAST and the robustness of the APCE criterion.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

REFERENCES

[1] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 1304–1318, Oct. 2004.

[2] N. Ray, S. T. Acton, and K. Ley, "Tracking leukocytes in vivo with shape and size constrained active contours," *IEEE Trans. Med. Imaging*, vol. 21, no. 10, pp. 1222–1235, Oct. 2002.

[3] J. Cui, S. T. Acton, and Z. Lin, "A Monte Carlo approach to rolling leukocyte tracking in vivo," *Med. Image Anal.*, vol. 10, no. 4, pp. 598–610, 2006.

[4] X. Liu, Z. Lin, and S. T. Acton, "A grid-based Bayesian approach to robust visual tracking," *Digit. Signal Process.*, vol. 22, no. 1, pp. 54–65, 2012.

[5] D. P. Mukherjee, N. Ray, and S. T. Acton, "Level set analysis for leukocyte detection and tracking," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 562–572, Apr. 2004.

[6] H. Ma, S. T. Acton, and Z. Lin, "OSLO: Automatic cell counting and segmentation for oligodendrocyte progenitor cells," in *Proc. Int. Conf. Image Process.*, 2018.

[7] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Comput. Vis. Pattern Recognit.*, 2010.

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, 2012.

[9] W. Zuo, X. Wu, L. Lin, L. Zhang, and M. Yang, "Learning support correlation filters for visual tracking," in *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 41, no. 5, pp. 1158–1172, 1 May 2019.

[10] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2017.

[11] M. Zhang, J. Xing, J. Gao, and W. Hu, "Robust visual tracking using joint scale-spatial correlation filters," in *Proc. Int. Conf. Image Process.*, 2015, pp. 1468–1472.

[12] D. Huang, L. Luo, M. Wen, Z. Chen, and C. Zhang, "Enable scale and aspect ratio adaptability in visual tracking with detection proposals," in *Proc Brit. Mach. Vis. Conf.*, 2015.

[13] A. S. Montero, J. Lang, and R. Laganiere, "Scalable kernel correlation filter with sparse feature integration," in *Proc. Int. Conf. Comput. Vis. Workshop*, 2015, pp. 587–594.

[14] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Eur. Conf. Comput. Vis.* 2014.

[15] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2014.

[16] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014.

[17] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2014.

[18] H. Ma, S. T. Acton, and Z. Lin, "SITUP: Scale invariant tracking using average peak-to-correlation energy," 2018, *arXiv:1812.03111v1 [eess.IV].*

[19] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. Comput. Vis. Pattern Recognit.*, 2013.

[20] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, 2015.

[21] P. Dollár, "Piotrs image and video matlab toolbox (PMT)," [Online]. Available: http://vision.ucsd.edu/pdollar/toolbox

[22] P. F. Felzenszwalb *et al.*, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[23] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proc. Comput. Vis. Pattern Recognit.*, 2017.

[24] J. Valmadre *et al.*, "End-to-end representation learning for correlation filter based tracking," in *Comput. Vis. Pattern Recognit.*, 2017.

[25] C. Ma *et al.*, "Adaptive correlation filters with long-term and short-term memory for object tracking," *Int. J. Comput. Vis.*, pp. 1–26, 2018.

[26] J. Gao *et al.*, "Transfer learning based visual tracking with gaussian processes regression," in *Proc. Eur. Conf. Comput. Vis.*, 2014.

[27] Y. Li, J. Zhu, and S. C. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proc. Comput. Vis. Pattern Recognit.*, 2015.

[28] Y. Qi *et al.*, "Hedged deep tracking," in *Proc. Comput. Vis. Pattern Recognit.*, 2016.

[29] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.*, 2014.

[30] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-leaning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[31] C. Ma *et al.*, "Hierarchical convolutional features for visual tracking," in *Proc. Int. Conf. Comput. Vis.*, 2015.

[32] C. Ma *et al.*, "Robust visual tracking via hierarchical convolutional features," in *Proc. Int. Conf. Comput. Vis.*, 2015.

[33] S. Hare *et al.*, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.

[34] C. Peng *et al.*, "Robust visual tracking via dirac-weighted cascading correlation filters," *IEEE Signal Process. Lett.*, vol. 25, no. 11, pp. 1700–1704, Nov. 2018.

[35] H. Wang *et al.*, "Robust visual tracking via semiadaptive weighted convolutional features," *IEEE Signal Process. Lett.*, vol. 25, no. 5 pp. 670–674, May 2018.

[36] Y. Han *et al.*, "Spatial-temporal context-aware tracking," *IEEE Signal Process. Lett.*, vol. 26, no. 3, pp. 500–504, Mar. 2019.