Object Tracking in Satellite Videos by Fusing the Kernel Correlation Filter and the Three-Frame-Difference Algorithm

Bo Du, *Member, IEEE*, Yujia Sun, Shihan Cai, Chen Wu, *Member, IEEE*, and Qian Du, *Senior Member, IEEE*

***Abstract*— Object tracking is a popular topic in the field of computer vision. The detailed spatial information provided by a very high resolution remote sensing sensor makes it possible to track targets of interest in satellite videos. In recent years, correlation filters have yielded promising results. However, in terms of dealing with object tracking in satellite videos, the kernel correlation filter (KCF) tracker achieves poor results due to the fact that the size of each target is too small compared with the entire image, and the target and the background are very similar. Therefore, in this letter, we propose a new object tracking method for satellite videos by fusing the KCF tracker and a three-frame-difference algorithm. A specific strategy is proposed herein for taking advantage of the KCF tracker and the three-frame-difference algorithm to build a strong tracker. We evaluate the proposed method in three satellite videos and show its superiority to other state-of-the-art tracking methods.**

***Index Terms*— Data fusion, kernel correlation filter (KCF), object tracking, satellite video, three-frame difference.**

I. Introduction

In terms of visual information processing for low-resolution images, most algorithms do not work well [1]. In order to obtain more information about an image, very high resolution (VHR) images are desirable [2]. Accordingly, VHR remote sensing imagery has become of great interest and is widely used in many applications [3]–[6]. Recently, commercial satellite technology has achieved significant progress in capturing VHR videos. Remote sensing videos have great potential for motion analysis [3], traffic monitoring [4], [5], suspicious object surveillance [6], etc. Currently, videos with a 1-m spatial resolution from the International Space Station were released [7] and have drawn much attention about object recognition and tracking for vehicles. The Jilin-1 commercial satellite produced by China can provide VHR satellite videos at a spatial resolution of 0.72 m [8]. These advancements show the possibility of tracking moving objects from satellite videos.

In general, a tracking algorithm can be categorized as either being a generative model [9]–[12] or a discriminative model [13]–[20] based on its representation schemes. In terms of the generative model, tracking is treated as a searching problem to find a region, which is most similar to the target object, within a neighborhood in the previous frame. A variety of search algorithms based on the generative model have been proposed. For instance, l_1-tracker [9] used a sparse linear combination of the target and the trivial fragmental templates. Adam *et al.* [10] designed an appearance model to deal with partial occlusion. Ross *et al.* [11] proposed incremental visual tracking algorithm to understand the incremental low-dimensional subspace [21]. Discriminative models have attracted wider attention than generative models. They regard object tracking as a binary classification problem. Hare *et al.* [13] utilized image features to train a classifier based on the support vector machine [17], [22]. Kalal *et al.* [15] used a set of structure constraints to guide the sampling process of a boosting classifier. In addition, a great deal of effort has been devoted to extract features [23] such as color features, the deformable part-based model [24], and the convolution neural network (CNN). Algorithms that use color features [25]–[29] and CNN [30]–[32] have obtained quite satisfying results.

Recently, tracking methods based on correlation filters have obtained excellent performance in object tracking [18], [20], [26], [27], [33]–[35]. Henriques *et al.* [18] proposed a kernel correlation filter (KCF) algorithm to conduct dense sampling in the area around the target. It can take advantage of the abundant information among negative samples by dense sampling. Besides, the KCF transforms the computation from the spatial domain into the Fourier domain by constructing a circulant matrix. As a result, the computational cost is substantially reduced.

For a satellite video, the total size of a frame can be up to 3840×2160 pixels or more, containing more than eight million pixels, more than 100 times the size of normal frames. Meanwhile, the target of interest takes up only about 30×80 pixels and sometimes even less. This is too small compared with the entire frame for analysis. Besides, the target is mostly very similar to the background and has a comparatively low resolution. These issues lead to a higher probability of

Manuscript received May 25, 2017; revised September 19, 2017 and November 17, 2017; accepted November 18, 2017. Date of publication December 18, 2017; date of current version January 23, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61601333 and Grant 61471274, in part by the China Postdoctoral Science Foundation under Grant 2016T90733, in part by the Natural Science Foundation of Hubei Province of China under Grant 2016CFB245, and in part by the Open Research Fund of Key Laboratory of Digital Earth Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences under Grant 2015LDE001. *(Corresponding author: Chen Wu.)*

B. Du is with the International School of Software and also with the School of Computer Science, Wuhan University, Wuhan 430072, China (e-mail: gunspace@163.com).

Y. Sun and S. Cai are with the School of Computer Science, Wuhan University, Wuhan 430072, China (e-mail: 2014202110102@whu.edu.cn; 736351411@qq.com).

C. Wu is with the International School of Software, Wuhan University, Wuhan 430079, China (e-mail: chen.wu@whu.edu.cn).

Q. Du is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762 USA (e-mail: du@ece.msstate.edu).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2017.2776899

tracking window drift. The satellite camera is relatively stable, and the surroundings of a target in a satellite-video experience have less change than common object tracking data.

Therefore, we utilize the three-frame-difference method to detect moving objects. The three-frame-difference method can detect slight and pixel-related changes among three frames and can highlight small moving targets from the background. With the assistance of the three-frame-difference method, the drift offset caused by the KCF can be reduced.

Thus, in this letter, we propose a new object tracking method by fusing the KCF tracker and the three-frame-difference method for a satellite video. The KCF tracker and the three-frame-difference method are included in the specific fusion framework. This fusion strategy can take advantage of both the KCF tracker and the three-frame-difference method. By combining the shape information from the KCF tracker and the change information from the three-frame-difference method into the final tracking results, our proposed method can address each target's position more accurately than do the many state-of-the-art tracking algorithms. An overview flowchart of the proposed method is shown in Fig. 1. Because the whole size of the satellite image is too large, we zoomed into a very small area of the image, which includes the moving target.

II. Method

A. Three-Frame Difference

For a given video sequence, the current frame is denoted by the kth frame and the previous frame is denoted by the $(k-1)$th frame. A binary image D is generated as

$$D(x,y)=\left\{\begin{array}{l}1,|f_k(x,y)-f_{k-1}(x,y)|\geq T\\0,|f_k(x,y)-f_{k-1}(x,y)|<T\end{array}\right\} \tag{1}$$

where $f_k(x,y)$ represents the grayscale value on point (x, y) at frame k, $D(x, y)$ indicates the corresponding binary value, and T is a threshold and has great importance for the final result of the three-frame difference. T is automatically calculated by Otsu [36]. Wojcik and Kaminski [37] proposed the three-frame-difference method. Given three sequential frames, e.g., $(k-1)$th, kth, and $(k+1)$th, the $D_1(x,y)$ is calculated by subtracting the $(k-1)$th frame from the kth frame as in (2) and $D_2(x,y)$ is calculated by subtracting the kth frame from the $(k+1)$th frame as in (3). Then, (4) can be used to generate $D(x,y)$ by $D_1(x,y)\cap D_1(x,y)$

$$D_1(x,y)=\begin{cases}1,|f_k(x,y)-f_{k-1}(x,y)|\geq T\\0,|f_k(x,y)-f_{k-1}(x,y)|<T\end{cases} \tag{2}$$

$$D_2(x,y)=\begin{cases}1,|f_{k+1}(x,y)-f_k(x,y)|\geq T\\0,|f_{k+1}(x,y)-f_k(x,y)|<T\end{cases} \tag{3}$$

$$D(x,y)=\begin{cases}1,D_1(i,j)\cap D_2(i,j)=1\\0,D_1(i,j)\cap D_2(i,j)=0.\end{cases} \tag{4}$$

The three-frame-difference method can deal with occlusion more effectively than does the intuitive two-frame difference and can reduce the irrelevant noise points.

B. Kernel Correlation Filter Tracking

The core component of most modern trackers is a discriminative classifier tasked with distinguishing between a target and its surrounding environment. To cope with natural image changes, the classifier is typically trained with translated and scaled sample patches. Such sample sets are riddled with redundancies—any overlapping pixels are constrained to be the same. Based on a simple observation, KCF [18] was proposed to take full advantage of the negative samples and reduce the redundancies. Besides, the KCF regards the tracking problems as regression rather than classification. For each sample, instead of labeling the positive samples as 1 and the negative samples as 0, the KCF gives a value ranging between 0 and 1.

A typical tracker based on a correlation filter trains the classifier with a target region sample image X of size to $I\times J$. By circularly shifting X, as shown in (5), the method obtains numerous training samples $x_{i,j}$, where $(i,j)\in\{0,1,\ldots,I-1\}\times\{0,1,\ldots,J-1\}$

$$X=C(x)=\begin{bmatrix}x_1 & x_2 & x_3 & \ldots & x_n\\ x_n & x_1 & x_2 & \ldots & x_{n-1}\\ x_{n-1} & x_n & x_1 & \ldots & x_{n-2}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ x_2 & x_3 & x_4 & \ldots & x_1\end{bmatrix} \tag{5}$$

where $\boldsymbol{x}=[x_1, x_2, \ldots, x_n]$ represents the base sample, and $\mathbf{X}$ represents the training samples by circularly shifting $\boldsymbol{x}$. For a 2-D image patch, x means the 1-D feature representation of the image patch, such as histogram of oriented gradient (HOG). The KCF utilizes a classifier to map them to the Gaussian function label $y_{i,j}$. Then, the algorithm models the target with the filter w. The ridge regression training result can be achieved by searching for the minimum value as

$$\min_w\sum_{i,j}|\varphi(x_{i,j}\cdot w-y_{i,j})|^2+\lambda||w||^2 \tag{6}$$

where φ denotes the kernel function mapping features into a kernel space, and λ is a regularization parameter. According to [18], it is known that $w=\sum_{i,j}\alpha_{i,j}\varphi(x_{i,j})$, where

$$F(\alpha)=\frac{F(y)}{F(k^{xx})+\lambda}. \tag{7}$$

In (7), k^{xx} refers to the kernel correlation [18], F is defined as the discrete operator, and λ is a constant to reduce the probability of overfitting. Here, we adopt the following Gaussian kernel function:

$$k^{xx'}=\exp\left(-\frac{1}{\sigma^2}(||x||^2+||x'||^2-2F^{-1}(F(x)\odot F^*(x')))\right) \tag{8}$$

where F^{-1} represents the inverse Fourier transform, $F^*(x')$ refers to the conjugate of $F(x')$, and $\odot$ is the Hadamard product of the matrix.

In the detection phase, we first take the target location in the former frame as the central position, clip an image patch z of size $I\times J$ in the new frame, and compute the response value of the classifier as

$$\hat{y}=F^{-1}(A\odot F(k^{\tilde{x}z})) \tag{9}$$

where $\tilde{x}$ is the learned target appearance model and the response value $\hat{y}$ refers to the similarity between the candidate target and the real target. Thus, the current position of the target can be detected by searching for the maximum value of $\hat{y}$, i.e.,

$$L=\max(\hat{y}). \tag{10}$$

Then, the target location in the current frame can be estimated, which is taken as the base sample for the next frame.

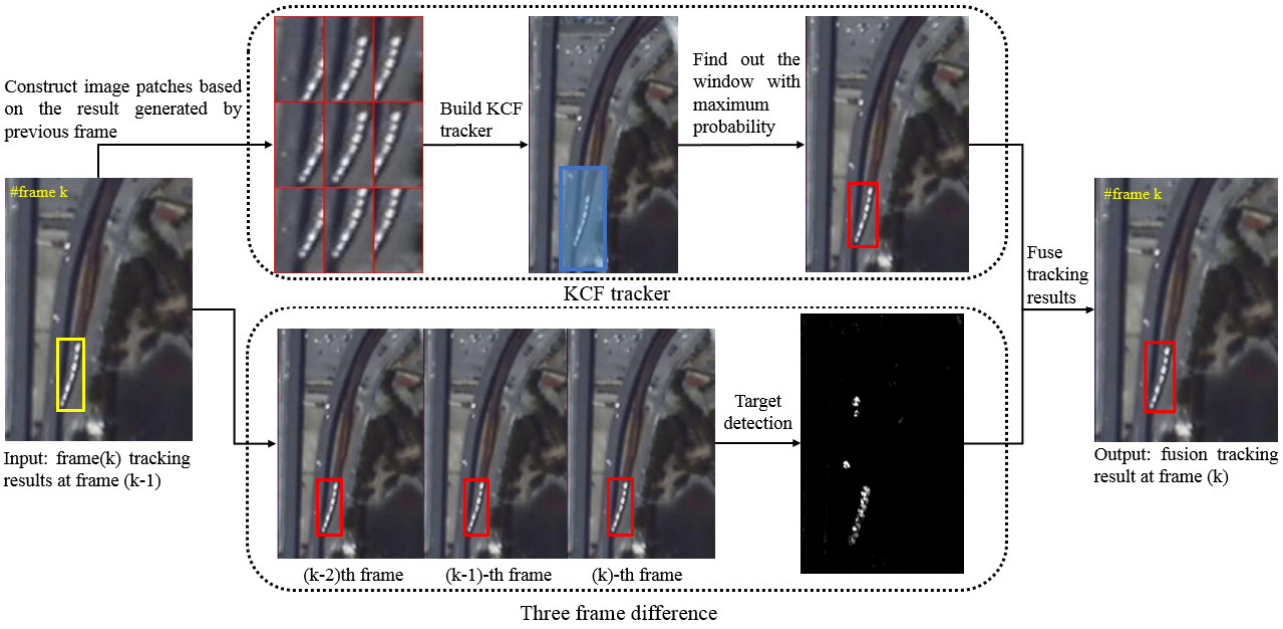

Fig. 1. System flowchart of the proposed algorithm. Many image patches can be generated by the circulant matrix based on the results from the prior frame. The KCF tracker is built by training the image patches and then searching the area around the base image sample. It can find the target location according to the maximum response value of the classifier. Besides, three-frame difference is utilized to detect the target in each given frame. Our fusion tracker fuses the results generated by the KCF tracker and the three-frame difference into the final results.

C. Tracker Fusion

In this section, we will introduce the fusion strategy for combining the results of the KCF tracker and the three-frame difference [38]. For each frame, the inputs are the results of the KCF tracker and the three-frame difference. We call the results T_j, $j \in [1,2]$. Each tracking result consists of N bounding candidate samples $b_{i,j} i \in [1 \ldots N]$—one for each frame $i$in the sequence. The fusion result T^* created by our method consists of one rectangular box for each frame.

Unlike the majority voting in data fusion, the proposed method sets a parameter for each candidate box and we call it the attraction a. The closer a fusion candidate is to a tracking result box, the stronger it is attracted. First, we will introduce how to compute the distance between two boxes. For boxes b and c, the distance can be calculated as follows:

$$d(b,c) = ||(d_x(b,c), d_y(b,c), d_w(b,c), d_h(b,c))^T||_2 = \left\| \left(\frac{c_x - b_x}{c_w + b_w}, \frac{c_y - b_y}{c_h + b_h}, \frac{c_w - b_w}{c_w + b_w}, \frac{c_h - b_h}{c_h + b_h} \right)^T \right\|_2 \quad (11)$$

where x, y, w, and h represent the horizontal ordinates of the top left corner, width, and height of the box. We assume that all the boxes have the same size and ignore scale changes, so the distance can be simplified as follows:

$$d(b,c) = \left\| \left(\frac{c_x - b_x}{c_w + b_w}, \frac{c_y - b_y}{c_h + b_h} \right)^T \right\|_2. \quad (12)$$

For the candidate e at frame i, its attraction $a_i(e)$ is

$$a_i(e) = \sum_{j \in M} \frac{1}{d(b_{i,j}, e)^2 + \sigma} \quad (13)$$

where σ is a constant which controls the distance's influence on the attraction. This parameter is crucial to the final tracking results. $b_{i,j}$ represents the candidate boxes for algorithm j at frame i. In order to find the final fusion box $e_i^* \in T^*$, we choose the one that has the greatest attraction among all the candidate boxes as the final result.

Algorithm 1 Fusion Strategy of the Proposed Algorithm

Input: tracking results $r_{t,1}$ and $r_{t,2}$ of the KCF tracking and the three-frame-difference method at frame (t)

Method:

a) Sample boxes around the tracking result e_{t-1}^* at frame (t - 1)
b) Search the candidate boxes $b_{t,1}$ and $b_{t,2}$ based on the tracking results $r_{t,1}$ and $r_{t,2}$. Compute the value $a_t(e)$ according to (11)–(13).
c) Calculate the final fusion tracking result e_t^*, which has the greatest attraction a among all the candidate boxes.
d) Calculate the tracking results $r_{t+1,1}$ and $r_{t+1,2}$ at frame $(t+1)$ based on the fusion tracking result e_t^*, the KCF tracking, and the three-frame-difference calculation.

Output: tracking location e_t^* at frame (t)

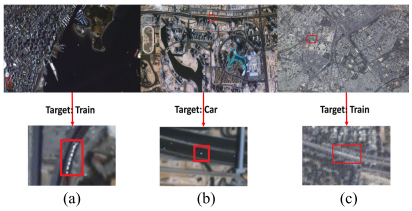

Fig. 2. Three data sets were evaluated in the experiments. Moving train is the target in (a) Canada and (c) New Delhi. Car is the target in (b) Dubai data set.

In this letter, we are building a fusion tracker based on the three-frame-difference method and the KCF tracker. The basic steps of our algorithm are presented in Algorithm 1.

III. Experiments

Three videos are used in the experiments. The first two videos have been provided by Deimos Imaging and UrtheCast,

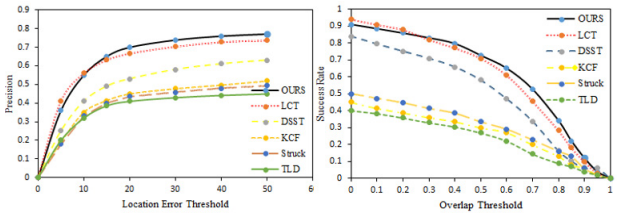

Fig. 3. (Left) Precision and (Right) success plots for three video sequences.

TABLE I
AVERAGE CLE (IN PIXEL)

	Tracker Fusion	TLD	STRUCK	KCF	LCT	DSST
Canada	**11**	18	15	15	**11**	13
Dubai	**12**	X	X	X	16	15
New Delhi	**9**	X	X	X	11	X
Average	**11**	X	X	X	13	X

and the third video has been provided by Chang Guang Satellite Technology Co., Ltd. These videos describe the traffic conditions of Canada, Dubai, and New Delhi, respectively. The image sizes of the first and second data sets are 3840 × 2160 pixels. The image size of the third data set is 3600 × 2700 pixels. For our experiments, moving trains and cars have been selected as targets. Fig. 2 shows the details of these three data sets. Furthermore, we initialize the position of the first frame and evaluate the proposed algorithm by comparing the output tracking bounding box with the ground truth bounding box. For the sake of comparison, five state-of-the-art tracking algorithms are employed: tracking-learning-detection (TLD) [15], struck [13], KCF [18], long-term correlation tracking (LCT) [39], and discriminative scale space tracking (DSST) [33].

The proposed algorithm is implemented in a C++ OpenCV library with 8-GB memory and an Intel Core i5 2.8-GHz CPU. The speed of the proposed algorithm is 9 frames/s. Since previous work on the KCF algorithm has proved that applying the HOG feature can achieve a higher tracking accuracy than applying the raw pixel feature, we employ the HOG feature in our fusion framework [18]. The size of the searching window is set as 1.5 times the target size. The σ used in the Gaussian function has been chosen as 0.5. The cell size of the HOG feature is 4×4, the block size is 16×16, the block stride is 8×8, and the orientation bin number of the HOG feature is 9. In addition, the regularization λ is set as 10^{-4}. α in (7) is set as 0.25. All these parameters are set the same as [18]. For the three-frame-difference calculation, the T value in (2) and (3) is acquired by self-adaption clustering image thresholding, namely OTSU [36]. For the fusion part of the experiments, σ, a constant to control the distance's influence on the attraction in (13), is set as the initial size of the ground truth bounding box.

In terms of assessment metrics, the precision plot and success plot have been adopted [40], [41]. A frame may be considered correctly tracked if the predicted target center is within a distance threshold to the ground truth. Precision curves simply show the percentage of the correctly tracked frames for a range of distance thresholds. As shown in Fig. 3, our method achieved excellent performance with respect to the other five competitor trackers for precision plots. We also calculated the area under curve (AUC) for each tracker. For precision plots, the AUC of the proposed method is 0.76, larger than LCT (0.74), DSST (0.52), KCF (0.38), struck (0.36), and TLD (0.33). A frame may be considered as correct if the overlap score (OS) of the predicted target and ground truth is larger than the threshold. In terms of success plots, the AUC of the proposed method is 0.56, larger than LCT (0.54), DSST (0.41), KCF (0.26), struck (0.23), and TLD (0.21).

TABLE II
AVERAGE BOUNDING BOX OS (IN PERCENTAGE)

	Tracker Fusion	TLD	STRUCK	KCF	LCT	DSST
Canada	71	48	57	58	**73**	66
Dubai	**69**	X	X	X	64	64
New Delhi	**63**	X	X	X	58	X
Average	**71**	X	X	X	65	X

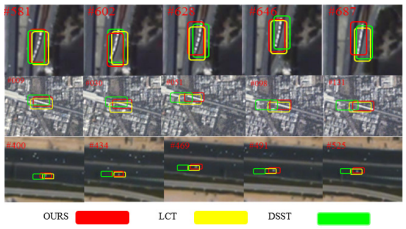

Fig. 4. Screenshots of some tracking results in the three videos.

Tables I and II list the accurate results according to the center location error (CLE) in pixel and OS in percentage, respectively. The best performance is marked with red bold digits. The proposed fusion tracker ranks the first for the Dubai and New Delhi data sets and the second for the Canada data set. The average CLE and OS of the proposed fusion tracker outperform those of the other five trackers. In fact, TLD, struck, and KCF completely failed in terms of the Dubai and New Delhi data sets, and DSST completely failed in terms of the New Delhi data set. Only the proposed fusion tracker and LCT were successful for all three data sets. The results show that the proposed fusion tracker is efficient for satellite-video tracking.

Fig. 4 shows screenshots of some tracking results of the three videos. Besides our proposed method, LCT and DSST have been selected as they perform better than the other methods in terms of comparative algorithms. It can be observed that our proposed method can track moving targets accurately. LCT can achieve a visually similar result. However, our proposed method is superior according to the quantitative assessments shown in Fig. 3 and Tables I and II.

The reasons why the performance of the proposed method is higher than those of other state-of-the-art algorithms are as follows. In satellite videos, the target moves only slightly and is extremely similar to the background due to the comparatively low resolution. To address these problems, we combined the KCF [18] tracker with the three-frame-difference method to form our proposed method. The three-frame-difference method can detect slight and pixel-related changes among three frames and can highlight small moving targets even among the background. While the three-frame-difference method captures the approximate area of the target, the KCF tracker can address the target's more accurate position. On the contrary, the other five state-of-the-art algorithms failed to detect the slight change in the target.

IV. CONCLUSION

In this letter, a new fusion tracker for object tracking in satellite videos has been proposed. The proposed new fusion

tracker fuses the KCF tracker and the three-frame-difference method. Since the satellite image has over eight million pixels and the target of interest is very small and the image has a low resolution, most traditional object tracking algorithms cannot yield satisfactory results. The proposed fusion tracker takes advantage of both the KCF tracker and the three-frame-difference method and fuses them with a specific strategy. We identified each candidate box's attraction value by evaluating its distance from the bounding boxes generated by the KCF tracker and the three-frame-difference calculation. Experiments on three satellite videos show that the fusion tracker outperforms five state-of-the-art trackers in precision plots and success plots. An interesting direction for future work would be to improve the time performance of the fusion tracker.

ACKNOWLEDGMENT

The authors would like to thank Deimos Imaging, UrtheCast, and Chang Guang Satellite Technology Co., Ltd., for acquiring and providing the data used in this letter, and the IEEE GRSS Image Analysis and Data Fusion Technical Committee.

REFERENCES

[1] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.

[2] X. Lu, H. Yuan, P. Yan, Y. Yuan, and X. Li, "Geometry constrained sparse coding for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 1648–1655.

[3] Z. He, S. Yi, Y.-M. Cheung, X. You, and Y. Y. Tang, "Robust object tracking via key patch sparse representation," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 354–364, Feb. 2017.

[4] J. Caro-Gutierrez, M. E. Bravo-Zanoguera, and F. F. González-Navarro, "Methodology for automatic collection of vehicle traffic data by object tracking," in *Proc. Mexican Int. Conf. Artif. Intell.*, Cancún, Mexico, 2016, pp. 482–493.

[5] S. K. Patel and A. Mishra, "Moving object tracking techniques: A critical review," *Indian J. Comput. Sci. Eng.*, vol. 4, no. 2, pp. 95–102, Apr./May 2013.

[6] A. S. Jalal and V. Singh, "The state-of-the-art in visual object tracking," *Informatica*, vol. 36, no. 3, pp. 227–248, Sep. 2012.

[7] IEEE GRSS. (Jan. 3, 2016). *IEEE GRSS Data Fusion Contest*. [Online]. Available: http://www.grss-ieee.org/community/technical-committees/data-fusion

[8] N. Chen. (May 13, 2016). *Jilin-1: China's First Commercial Remote Sensing Satellites Aim to Fill the Void*. [Online]. Available: http://english.cas.cn/newsroom/news/201605/t20160513_163009.shtml

[9] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2259–2272, Nov. 2011.

[10] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2006, pp. 798–805.

[11] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, Aug. 2008.

[12] X. Lu, Y. Yuan, and P. Yan, "Robust visual tracking with discriminative sparse learning," *Pattern Recognit.*, vol. 46, no. 7, pp. 1762–1771, Jul. 2013.

[13] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. Int. Conf. Comput. Vis.*, Colorado Springs, CO, USA, 2011, pp. 263–270.

[14] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. 12th Eur. Conf. Comput. Vis.*, Florence, Italy, 2012, pp. 864–877.

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

[16] D. Zhang, J. Han, L. Jiang, S. Ye, and X. Chang, "Revealing event saliency in unconstrained video collection," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1746–1758, Apr. 2017.

[17] S. Avidan, "Support vector tracking," in *Proc. IEEE Comput. Conf. Comput. Vis. Pattern Recognit.*, Kauai, HI, USA, Dec. 2001, pp. 184–191.

[18] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[19] Y. Wu, B. Shen, and H. Ling, "Online robust image alignment via iterative convex optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 1808–1814.

[20] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 2544–2550.

[21] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Geometric mean for subspace selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 260–274, Feb. 2009.

[22] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1088–1099, Jul. 2006.

[23] J. Han *et al.*, "Representing and retrieving video shots in human-centric brain imaging space," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2723–2736, Jul. 2013.

[24] Y. Yuan, Y. Lu, and Q. Wang, "Tracking as a whole: Multi-target tracking by modeling group behavior with sequential detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3339–3349, Dec. 2017.

[25] Q. Wang, J. Fang, and Y. Yuan, "Multi-cue based tracking," *Neurocomputing*, vol. 131, pp. 227–236, May 2014.

[26] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1090–1097.

[27] A. Lukežič, T. Vojíř, L. Čehovin, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 4847–4856.

[28] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 2113–2120.

[29] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 1401–1409.

[30] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 850–865.

[31] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 4293–4302.

[32] H. Nam, M. Baek, and B. Han. (2016). "Modeling and propagating cnns in a tree structure for visual tracking." [Online]. Available: https://arxiv.org/abs/1608.07242

[33] 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.

[34] 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.*, Florence, Italy, 2012, pp. 702–715.

[35] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Eur. Conf. Comput. Vis.*, Zürich, Switzerland, 2014, pp. 254–265.

[36] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[37] G. M. Wojcik and W. A. Kaminski, "Liquid state machine built of Hodgkin–Huxley neurons," *Informatica*, vol. 15, no. 1, pp. 39–44, 2004.

[38] C. Bailer, A. Pagani, and D. Stricker, "A superior tracking approach: Building a strong tracker through fusion," in *Proc. Eur. Conf. Comput. Vis.*, Zürich, Switzerland, 2014, pp. 170–185.

[39] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 5388–5396.

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

[41] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 2411–2418.