Multi-label learning based target detecting from multi-frame data

Mengqing Mei | Fazhi He

School of Computer Science, Wuhan University, Wuhan, P.R. China

Correspondence
Fazhi He, School of Computer Science, Wuhan University, Wuhan, P.R. China.
Email: fzhe@whu.edu.cn

Abstract
In the field of target detecting, lots of progress have been made in recent years. Owing to the progress of multiple frames time series data, or video satellites, target detecting from space-borne satellite videos has been available. However, detecting a slightly moving target from space-borne videos is still a difficult task, because of the low resolution and illumination variation influence. This paper considers target detecting from time series data as multi-label problem as there are several different kinds of background objects and targets of interest. To some extent the background of time series data is comparative invariant, using background analysis method to extract the target from the background is promising. This paper proposes a novel target detecting algorithm based on multi-label learning and Gaussian background description model aiming at extracting slowly moving target. To further enhance performances, multi-frame fusion and post processing method was utilized to catch the slight difference due to movement. Experimental results on real world datasets indicate that the proposed method outperforms some state-of-the-art algorithms.

1 | INTRODUCTION

Object tracking is one of the most important research issues in computer vision [1]. Generic object tracking is the task of estimating the target states in an image sequence, given only its initial state [2]. Although object tracking has been studied over the past several decades, it remains a challenging problem. Lots of solutions have been proposed for addressing this fundamental challenge, including using compressed sensing [3], support vector machine (SVM) [4], color name (CN) [5] etc. Recent years, discriminative correlation filters (DCF) [2, 6–8] and convolution neural network (CNN) [9, 10] also show great potentials in object tracking.

With the development of satellite-borne imaging technology, the latest remote sensing satellite technology has obtained very high resolution (VHR) satellite-borne video, which has attracted a lot of attention in computer vision filed. Satellite video can acquire a period of continuous observation over a certain area. In 2013, the International Space Station (ISS) released a 1024X1024 video with a 1m spatial resolution for cars and trains tracking [11]. China launched the first self research and development (R&D) commercial satellite “Jilin-1” in 2015. These advancements indicate the great potentials for VHR satellite video tracking to be used in motion analysis, automated surveillance and suspicious object monitoring [12].

However, compared with traditional object tracking, satellite video tracking faces more challenges, including: 1)Larger scene size and image size; 2)Smaller target size; 3)Less features and more background clutter; 4)Bigger illumination variation influence [12]. All of these characteristics will significantly lead to the tracking drift and even tracking target loss [13].

Because the satellite videos are taken while the satellite gazes fixedly at the earth, the background information of the tracking object mostly keep unchanged. Owing to the special features of satellite videos, it is a good solution to employ background subtraction. When the objects are moving and the background is relatively unchanged, background subtraction method can identify moving objects and achieve satisfactory performance for separating the target from background. Since the satellite videos are taken from high altitude, the movement of interested target are slightly, which may lead to poor performance with background subtraction. This problem can be addressed by using multi-frame difference method. The multi-frame difference method can enlarge the displacement distance between consecutive frame, and make the object-background separation achieve better performance.
In order to solve the above problems, in this letter, we propose a background subtraction satellite tracker (BSST) to track the slightly moving object in satellite videos which based on Gaussian mixture model background subtraction and integral image. Experimental results on two satellite video datasets demonstrate the superiority of our proposed method over state-of-the-art tracking algorithms.

2 | METHODOLOGY

In this section, we elaborate on the proposed background subtraction satellite tracker. The whole procedure of our proposed method is shown in Figure 1.

The main steps are as follows:

1) Input the first frame and the kth frame of satellite video images into improved Gaussian mixture model background subtraction method to establish the background model. The tracking object will be segmented as foreground target using the background subtraction method. And update background model in subsequent video frames; 2) an integral matrix is obtained by integrating the segmented image with a rectangular integral box which has the same size as the ground truth bounding box; 3) the location of the maximum value in the integral matrix is the center position of the interested object; 4) using the central position of the interested object to get the tracking bounding box’s range area.

2.1 | Background subtraction

Background subtraction is a practical method in object detection, and has the ability to segment moving objects from background. Among most of classical background modeling methods, Gaussian mixture model (GMM) has a good performance for precise calculation [14]. Owing to the statistical characteristics of Gaussian mixture model, the background modeling is more objective and scientific than just using threshold to segment background and foreground[15].

If $X_i$ represents the intensity of a pixel $(x, y)$ at time $t$, the probability of intensity value in $X_i$ is

$$p(X_i) = \sum_{k=1}^{K} w_{k,t} \eta(X_i, \mu_{k,t}, \Sigma_{k,t}),$$  \hspace{1cm} (1)

where $K$ is the number of distributions, from 3 to 5 are often used. $w_{k,t}$ is an estimate of the weight of the $k$th Gaussian distribution function in the mixture model at time $t$, $\mu_{k,t}$ is the mean value of the $k$th Gaussian distribution function in the mixture model at time $t$, $\Sigma_{k,t}$ is the covariance matrix of the $k$th Gaussian distribution function in the mixture model at time $t$, and $\eta$ is a Gaussian probability density function

$$\eta(X_i, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_i-\mu)^2}{2\sigma^2}}.$$  \hspace{1cm} (2)

To facilitate the calculation, we assume the covariance matrix is:

$$\Sigma_{k,t} = \sigma_{k,t}^2 I.$$  \hspace{1cm} (3)

This equation assumes that the pixel values of three bands of RGB images are independent and have the same variances. This assumption can avoid costly matrix inversion, but it will sacrifice some accuracy.

For each new pixels $X_i$, $X_j$ is compared with the existing $K$ Gaussian distribution models according to:

$$|X_i - \mu_{k,t}| \leq 2.5 \sigma_{k,t}.$$  \hspace{1cm} (4)

If one of the value of pixels $X_j$ in the $k$th Gaussian distribution model meets Equation (4), it can be considered to match...
the $k$th Gaussian distribution function. Otherwise, this point does not match the distribution model.

The weights of the $K$ distributions at time $t$ are adapted according to:

$$w_{k,t} = (1 - \alpha)w_{k,t-1} + \alpha X_t,$$  

(5)

where $\alpha$ is the learning rate, and $M_{k,t}$ is 1 for the pixel that matched the distribution model, and 0 for the remaining pixels.

For the unmatched pixels, the $\mu$ and $\sigma$ parameters of the distributions model function remain the same. The parameters of the distribution function of the matched pixels are updated as following:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t,$$  

(6)

$$\sigma^2_t = (1 - \rho)\sigma^2_{t-1} + \rho (X_t - \mu_t)^T (X_t - \mu_t),$$  

(7)

where $\rho$ the second learning rate parameter is

$$\rho = \frac{\alpha}{w_{k,t}}.$$  

(8)

In addition, in order to ensure the sum of the weights is 1, the weights should be normalized after each updating:

$$w_{k,t} = \frac{w_{k,t}}{\sum_{k=1}^{K} w_{k,t}}.$$  

(9)

When $X_t$ does not match any Gaussian distribution function at time $t$, a new Gaussian distribution function is created, which will replace the original Gaussian distribution function with minimum weight. The value of pixel $X_t$ at time $t$ will be set as the new mean value $\mu_t$, and the covariance matrix $\sigma^2_{k,t}$ and the weight $w_{k,t}$ will be set to the initial value.

When the target is moving, the boundary of it may match none of the existing distributions model which will result in either the creation of a new distribution model or the increase in the variance of an existing distribution. Therefore, which portion of the mixture model best represents background processes need to be decided. The Gaussian are ordered by

$$rank = \frac{w_t}{\sigma_t}.$$  

(10)

The rank will increase when the distribution gains more weight or the variance decreases. It will make the most likely background distribution models remain on the top and the less probable drift background distribution model on the bottom to eventually replaced by new distribution models.

Then the first $N$ distribution model are chosen as the background model:

$$N = \text{argmin}_n \left( \sum_{k=1}^{N} w_k > T \right),$$  

(11)

where $T$ is the threshold for finding the minimum portion of the pixels that should be classified to the background.

The first $N$ Gaussian distribution models are selected as the background model and the rest was selected as the foreground model. Thus, if the pixel $X_t$ matches the first $N$ models, it will be classified as the background pixel; otherwise, this pixel is a foreground pixel. Since the pixels in interested target area do not fit Equation (4), it not match the first $N$ background models. As a result, the interested target will be regarded as the foreground.

### 2.2 Integral image

The integral image is an method for efficiently and quickly generating the sum of values in a rectangular subset of a grid. In integral image, the value of integral image at point $(x, y)$ is the sum of all the pixels above and to the left. Just using a few operations per pixel, the integral image can be computed. Once the integral image was computed, value of any rectangular subection area in the image can be computed at any scale or location in constant time [16].

When the pixel moves quickly, the pixel will be classified as foreground pixel, in this way, the corresponding color in the image is white (RGB value = 255), and the color of background pixels is black (RGB value = 0). So in the integral image, while the rectangle area is the same, the integral value is bigger, means this rectangle area includes more foreground pixels, this area is more probable to be the location of the tracking object. Therefore, the area that has the biggest value inside integral image is the location of the tracking object.

For less computation consumption, the integral area can be confined in a rectangular searching area which expands $r$ pixels outside the previous frame tracking bounding box, where $r$ is the searching radius.

### 2.3 Multi-frame difference

Generally, the target in the satellite videos move very slowly. Multi-frame difference is adopted for getting more obvious movement difference between two images.

While the interval is increasing, the foreground place will be more obvious, and the integral image value of tracking bounding box area is increasing. Nevertheless, as the frame interval increased, some background pixels may become noise points, which may disturb the accuracy of tracking result. Therefore, in order to get the best tracking result, it is important to find a best interval value $i$ between two frames. More discussion about the interval $i$ are shown in Sections 3 and 4.

### 3 EXPERIMENTS

Two videos are used in the experiments. The Canada Vancouver harbor dataset was provided by the UrtheCast Corp., the IEEE Geoscience and Remote Sensing Society(GRSS) Image Analysis and Data Fusion Technical Committee[17]. And the India
New Delhi dataset was provided by the Chang Guang Satellite Technology Co., Ltd. We compare our algorithm with 7 state-of-the-art tracking algorithms. The trackers used for comparison are: KCF [6], SAMF[18], fDSST[7], Staple[19], MOFT[16], C-COT[8], ECO[2]. The code of our algorithm is implemented by the Mixture of Matlab and C/C++, and the computer configuration is Intel Core i7-3770 CPU(3.4 GHz), 32 GB memory.

In the experiment, we cropped the video frames to get a smaller experimental area relatively close to the moving target. This is because the target in the remote sensing image is very small and it is difficult to see the tracking target and the tracking bounding box.

For assessment, the OTB standard success plot and precision plot were used in our experiments [20]. The performance ranking of the algorithms mainly refers to the success plots. Besides, the precision plots is adopted as auxiliary assessment.

### Canada Vancouver harbor

The first dataset is cited from the 2016 IEEE GRSS Data Fusion Contest[17], which is an Ultra High Definition (UHD) video. The resulting frame was fully rectified and resampled to 1m. The last 34 s of experiment dataset have 418 frames. The cropped area is from the coordinate (0,1650) of the original image, and the size of the image in the experimental area is 200×500. The size of the ground truth bounding box for the target is 30×80.

For the parameter interval \(i\) and the searching radius \(r\), we use experiments to find the best pair.

For the searching radius parameter \(r\), the AUC score of the success plots is shown in Table 1. It can be seen that when the search radius \(r = 10\), the proposed method has the best accuracy.

For the interval parameter \(i\), the AUC score are shown in Table 2. Table 2 illustrates that when \(i = 5\), the proposed method show the best performance.

Since we have found the best parameter pair of searching radius and interval (where \(r = 10, i = 5\)), Figure 2 shows parts of the tracking trajectory. As shown in Figure 2(b), most of the tracking algorithms cannot track the target correctly from the 150th frame, only our algorithm, MOFT and the 2016 VOT Challenge Champion C-COT can keep tracking the target. From Figure 2(b), the tracking box of C-COT tracker resized to a larger rectangle than the ground truth; however, in Figures 2(c) and 2(d), the tracking box of C-COT tracker get smaller and smaller; both will lead to the drop of the overlap score.

The success plots of the proposed method and comparison algorithms of Canada Vancouver harbor dataset are shown in Figure 4(a). The proposed algorithm outperforms the best tracker in the comparison algorithms, MOFT (scores 0.861), with the ratio of 4.1%.

From the precision plots shown in Figure 4(b), only the proposed algorithm and MOFT can get the score of 1.000, which means only these two trackers can track the target precisely in a CLE value at 10 throughout the tracking process.

### India New Delhi

The second dataset in this experiment is provided by Chang Guang Satellite Technology Co., Ltd. The last 28 s of the dataset video contain 700 frames. The image resolution of each frame is 3600 × 2700, covering a part of urban area in New Delhi city, India. The cropped area is from the coordinate (400, 850) of the original image, and the size of the image in the experimental area is 650 × 300. The size of the ground truth bounding box for the target is 72 × 26.

The video’s resolution is not very high, and there are many noises point and camera shaking. Affected by the inconsistency between the satellite revolving speed and the rotation speed of earth, the light deflected from the ground was changed. As a result, as shown in Figure 3, the white saturation of the target train become lower in the latter half of the video, leading to more difficulty to separate the target from the background.

For the searching radius \(r\), the AUC score in success plots are shown in Table 3. It can be seen that \(r = 3\) can get the best accuracy. This indicates that the proposed method can also obtain similar and stable performance under the other radii, and further verifies the robustness of our proposed method.

For the interval parameter \(i\), the AUC score are shown in Table 4, which illustrates that the proposed method show the best performance when \(i = 9\). Owing to the slower movement of the target in this dataset, smaller search radius pair with larger frame difference can get a better experiment result.

As we have determined the best searching radius and interval (where \(r = 3, i = 9\)), Figure 3 shows parts of the tracking trajectory. It can be seen that only our proposed algorithm and the MOFT tracker can correctly acquire the target location throughout the whole task, while other algorithms all lost the target since the 150th frame. From Figures 3(c) and 3(d), the proposed algorithm shows more precise performance than the MOFT tracker.

The success plots and the precision plots of the proposed method and comparison algorithms of India New Delhi dataset are shown in Figures 4(c) and 4(d).

From the success plots, except the MOFT algorithm, all of the other comparison algorithms have much lower scores than the proposed algorithm. The proposed algorithm, which gets an

| TABLE 1 | The AUC score of the searching radius of Vancouver dataset |
|---|---|---|---|---|---|---|
| \(r\) | 6 | 7 | 8 | 9 | 10 | 11 |
| AUC Score | 0.795 | 0.815 | 0.836 | 0.846 | 0.852 | 0.850 |

| TABLE 2 | The AUC score of the interval of Vancouver dataset |
|---|---|---|---|---|---|---|
| \(i\) | 3 | 4 | 5 | 6 | 7 | 8 |
| AUC Score | 0.891 | 0.895 | 0.896 | 0.895 | 0.894 | 0.888 |
**FIGURE 2** Parts of the tracking results for Canada Vancouver harbor data set

**FIGURE 3** Parts of the tracking results for India New Delhi data set
FIGURE 4  The success plots and the precision plots of Vancouver and India dataset

TABLE 3  The AUC score of the searching radius of India dataset

\[
\begin{array}{ccccccc}
\text{r} & 2 & 3 & 4 & 5 & 6 & 7 \\
\text{AUC Score} & 0.524 & 0.739 & 0.736 & 0.733 & 0.716 & 0.712 \\
\end{array}
\]

TABLE 4  The AUC score of the interval of India dataset

\[
\begin{array}{cccccccc}
\text{i} & 5 & 6 & 7 & 8 & 9 & 10 \\
\text{AUC Score} & 0.862 & 0.856 & 0.873 & 0.873 & 0.884 & 0.883 \\
\end{array}
\]

AUC score of 0.884, outperforms the top ranked tracker MOFT with the ratio of 8.1%.

From the precision plots, similar to the success plots, except the MOFT algorithm, all of the other comparison algorithms have much lower scores than the proposed algorithm. The proposed algorithm gets the precision score of 0.996, which is almost equal to 100%, and outperforms the top ranked tracker of the comparison algorithms, MOFT, with the ratio of 5.5%.

4  |  CONCLUSION

In this letter, we proposed an efficient and robust tracking algorithm that aimed at tracking object in satellite video datasets. In our algorithm, we first disposed frames with GMM background subtraction to get a background model, and the background model will be updated frame by frame afterward. Then, the most probable position of the target is obtained by using the integral image. Besides, the multi-frame difference method is used to obtain more accurate target position.

The qualitative and quantitative experimental results on remote sensing satellite video datasets show that our proposed method has the ability to track slightly moving targets more accurately than other state-of-the-art algorithms, and also shows that the proposed method is a better satellite video target tracking method.

In the future, our work will focus on the following two aspects: (1) employ correlation filter framework to get a more accuracy tracking result; (2) try to use machine learning deep networks to extract target feature for a better tracking performance.
ACKNOWLEDGEMENTS
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 study, and the IEEE GRSS Image Analysis and Data Fusion Technical Committee.

ORCID
Fazhi He https://orcid.org/0000-0001-7016-3698

REFERENCES
1. Wu, Y., et al.: Online object tracking: A benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Piscataway (2013)
2. Danelljan, M., et al.: ECO: efficient convolution operators for tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 6931–6939. IEEE, Piscataway (2017)
3. Zhang, K., et al.: Real-time compressive tracking. In: European Conference on Computer Vision, pp. 864–877. Springer, Berlin (2012)
4. Hare, S., et al.: Struck: Structured output tracking with kernels. In: IEEE International Conference on Computer Vision, ICCV 2011, pp. 263–270. IEEE, Piscataway (2011)
5. Danelljan, M., et al.: Adaptive color attributes for real-time visual tracking. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1090–1097. IEEE, Piscataway (2014)
6. Henriques, J. F., et al.: High-speed tracking with kernelized correlation filters. IEEE Trans. on Pattern Anal. and Mach. Intell. 37(3), 583–596 (2015)
7. Danelljan, M., et al.: Discriminative scale space tracking. IEEE Trans. Pattern Anal. Mach. Intell. 39(8), 1561–1575 (2017)
8. Danelljan, M., et al.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: 14th European Conference on Computer Vision, pp. 472–488. Springer, Berlin (2016)
9. Bhat, G., et al.: Unveiling the power of deep tracking. In: 15th European Conference on Computer Vision, pp. 493–509. Springer, Berlin (2018)
10. Li, B., et al.: High performance visual tracking with siamese region proposal network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Piscataway (2018)
11. Lu, Y., et al.: A dense connection based network for real-time object tracking. Neurocomputing 410(14), 229–236 (2020)
12. Kopsiaftis, G., Karantzalos, K., Vehicle detection and traffic density monitoring from very high resolution satellite video data. In: 2015 IEEE Int. Geoosci. and Remote Sens. Symp., IGARSS 2015, Milan, Italy, July 26-31, 2015 1881-1884 (2015)
13. Danelljan, M., et al.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: 14th European Conference on Computer Vision, pp. 472–488. Springer, Berlin (2016)
14. Zheng, Y., Zheng, P.: Hand segmentation based on improved gaussian mixture model. In: 2015 International Conference on Computer Science and Applications (CSA), pp. 168–171. ACM, New York (2015)
15. Xie, X., et al.: Image de-noising algorithm based on gaussian mixture model and adaptive threshold modeling. In: 2017 International Conference on Inventive Computing and Informatics (ICICI), pp. 226–229. IEEE, Piscataway (2017)
16. Wang, Q., et al.: Looking closer at the scene: Multiscale representation learning for remote sensing image scene classification. IEEE Trans. Neural Netw. Learn. Syst. 1(1), 1–15 (2020)
17. Mou, L., et al.: Multitemporal very high resolution from space: Outcome of the 2016 IEEE GRSS data fusion contest. IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 10(8), 3435–3447 (2017)
18. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: European Conference on Computer Vision, pp. 254–265. Springer, Berlin (2014)
19. Bertinetto, L., et al.: Staple: Complementary learners for real-time tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition CVPR 2016, pp. 1401–1409. IEEE, Piscataway (2016)
20. Wu, Y., et al.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. 37(9), 1834–1848 (2015)

How to cite this article: Mei M, He F. Multi-label learning based target detecting from multi-frame data. IET Image Process. 2021;1–7.
https://doi.org/10.1049/ipr2.12271