Journal of Big Data

METHODOLOGY Open Access

Remote sensing detection enhancement

Tian J. Ma[1,2*]

*Correspondence:
tma@sandia.gov
[1] Sandia National Laboratories, Albuquerque, NM 87185, USA
Full list of author information is available at the end of the article

Abstract

Big Data in the area of Remote Sensing has been growing rapidly. Remote sensors are used in surveillance, security, traffic, environmental monitoring, and autonomous sensing. Real-time detection of small moving targets using a remote sensor is an ongoing, challenging problem. Since the object is located far away from the sensor, the object often appears too small. The object's signal-to-noise-ratio (SNR) is often very low. Occurrences such as camera motion, moving backgrounds (e.g., rustling leaves), low contrast and resolution of foreground objects makes it difficult to segment out the targeted moving objects of interest. Due to the limited appearance of the target, it is tough to obtain the target's characteristics such as its shape and texture. Without these characteristics, filtering out false detections can be a difficult task. Detecting these targets, would often require the detector to operate under a low detection threshold. However, lowering the detection threshold could lead to an increase of false alarms. In this paper, the author will introduce a new method that improves the probability to detect low SNR objects, while decreasing the number of false alarms as compared to using the traditional baseline detection technique.

Keywords: Remote sensing, Low SNR object detection, Small object detection, Constrained velocity matched filter, Velocity matched filter, Track-Before-Detect

Introduction

Big Data in the area of Remote Sensing has been growing rapidly. Remote sensors are used in surveillance, security, traffic, environmental monitoring, and autonomous sensing. Data from remote sensors can be fused with other data sources to generate actionable intelligence for decision makers [1]. However, real-time detection of low signal-to-noise-ratio (SNR) small moving targets using a remote sensor has been an ongoing, challenging problem. Since the object is located far away, the object often appears too small on the sensor. The object's SNR is often very low. Occurrences such as camera motion, moving backgrounds (e.g., rustling leaves), low contrast and resolution of foreground objects makes it difficult to segment out the targeted moving objects of interest. Due to the limited appearance of the target, it is difficult to obtain the target's characteristics such as its shape and texture. Without these characteristics, filtering out false detections can be a challenging task. Detecting these targets, would often require the detector to operate under a low detection threshold. However, lowering the detection threshold could lead to an increase of false alarms. In this paper, the author will introduce a new method that improves the probability to detect low SNR objects, while decreasing the number of false alarms as compared to using the traditional baseline detection technique. The paper is organized into the following sections: Section "Related works" provides a discussion on related research work published in open literature; Section "Experiment setup" focuses on the data used for our experimental results; Section "Methods" is a detailed discussion on our methods used; Section "Results and discussion" goes over our results; and Section "Conclusion" is a summary conclusion of the paper.

Related works

Recent artificial intelligence (AI) based object recognition methods uses a deep learning approach to segment objects in motion [2–4]. Deep learning algorithms can now achieve human-level performances on a variety of difficult tasks. Success on these methods generally require large training datasets [5] with quality features [5, 6]. However, deep learning methods have several drawbacks; including understanding its reasoning in making decisions. These methods are unclear to a human observer [7, 8] and requires large training data sets [5]. This can often lead to unexpected results when real data does not resemble those in the training datasets [9]. Moreover, studies have shown that these methods are vulnerable to adversarial attacks [10], fooling deep learning models to misclassify objects [10, 11]. Resolving these challenges are especially important in time critical security surveillance application, where failure to detect a target could result in high consequences.

Traditional change detection methods rely on background subtraction [12]. The background is continuously updated as each new frame is received. For each new frame that comes in, the estimated background will be subtracted from the new frame to produce a "Difference Image". Thresholding can then be applied on the Difference Image to identify changes in the scene [13–16]. Though these methods do not require pre-trained labels, they generally have a higher false alarm rate. Traditional background estimation can typically be categorized into two categories: pixel-based approaches and dimension reduction approaches. The popular pixel-based approach—Gaussian Mixture Model (GMM) [17], is known for its simplicity in implementation. Dimensional reduction approaches such as Principle Components of Pursuit (PCP) [18] and other subspace estimation approaches [19] are known for their robustness on handling camera jitters and light illumination.

Despite research advances in background modeling and AI object recognition approaches, these methods alone are inadequate to detect low SNR targets. Low SNR targets with limited features and training examples are not ideal for machine learning problems. Lowering the threshold using traditional background subtraction detection, could cause an increased number of false alarms.

Thus far, Velocity Matched Filter (VMF) techniques have been introduced [20, 21] and it appears to be extremely effective in improving the detection and tracking of low SNR targets. These methods enhance the target's signal and noise ratio (SNR) by integrating the target's energy over a period of time. It also matches the target's true velocity and direction of where it is traveling. Numerous Track-Before-Detect (TBD) algorithms [22–26] have evolved using VMF. TBD algorithms have been studied in a variety of remote sensors such as Radar [23, 24, 26, 27], Sonar [28], and Infrared (IR) [29]. While

Table 1 Camera specifications

Video camera	Frame rates	Image resolution	Lens focal length
Mysterium X	24 frames per second	3072 × 1620	72 mm

Table 2 Experiment location

Camera location	Peak of Sandia Mountain, Albuquerque, New Mexico
Camera location elevation	10,379 feet
Ground target elevation	6060 feet

earlier publications have demonstrated the effectiveness of using TBD algorithms in single target scenarios, it has also been used for tracking multiple targets, [25, 30] generally requiring the number of targets to be known ahead of time. Dynamic programming algorithms for performing VMF without a known target velocity has also been developed [26, 31]. While dynamic programming techniques seem to overcome run-time performance limitation, it tends to degrade on maneuvering targets due to the lack of motion modeling. More recently [32], motion models have been incorporated in TBD techniques to better handle target maneuver but the challenge remains on how to initialize tracking. Many of the published techniques are only demonstrated with simulated data.

Though VMF or TBD type techniques have shown to be theoretically appealing in low SNR target tracking scenarios, there are several challenges that make these approaches difficult to apply in the real world. First, the algorithm assumes a priori knowledge in which the number of targets are known or a priori knowledge to initialize the track. This knowledge is often not provided in real-time surveillance application. Recent advances made to [22, 33] eliminate these problems are by inserting an additional detection process before the TBD processor. However, initiating a low SNR target track is a challenge for this framework. If the target is not detectable by the threshold set by the initial detector process, then TBD will not start. On the other hand, setting a low detection threshold to force TBD to initiate could lead to many false alarms. Secondly, to gain the maximum benefit of the TBD approach, it assumes the noise is white [21]. Nonstationary changes such as jitters or sensor noises which are not removed, can introduce correlated noises that can potentially degrade VMF's performance. Finally, VMF assumes that the objects can be seen and that finding the best matched hypothesis is always available. However, in a real-world situation, objects may not always be observable by the sensors. For example, a person's movement can be observed by the sensor, but not when the person is walking behind a big tree. In this situation, finding the best matched filter becomes difficult because all the match hypotheses could be invalid.

Experiment setup

To evaluate our technique in a real-world scenario, a video camera (Table 1) was placed at the top of Sandia Mountain (Table 2) to collect live traffic data. The distance of the camera to the target on the ground was approximately 4000 feet. Since the image is

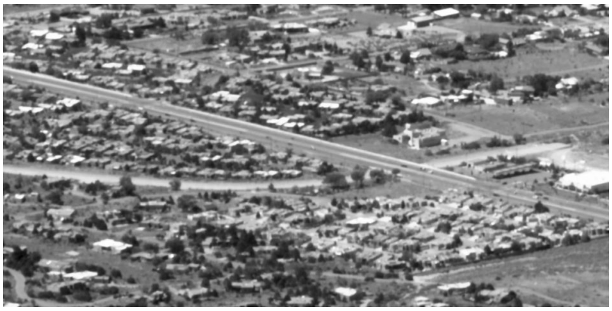

Fig. 1 A cropped image captured by the video camera

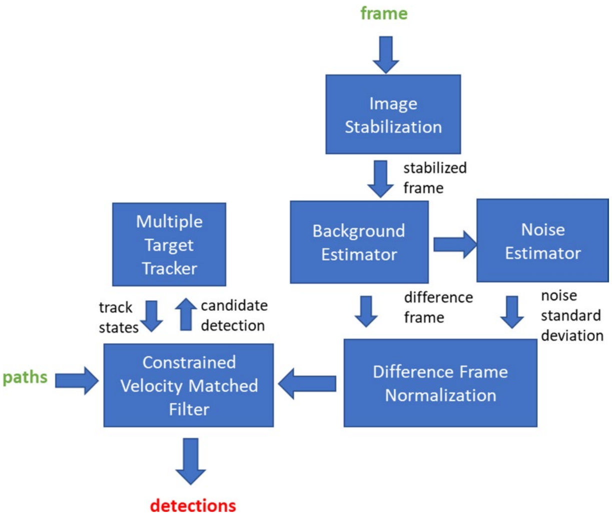

Fig. 2 Processing workflow

large, (Fig. 1) only shows a cropped off portion of the image. The targets were barely visible, and they appeared like small dots. The size of the target(s) in the image ranged from 4 to 20 pixels. The camera video is subjected to jitter motion naturally induced by the wind. This setup allowed us to evaluate the challenges of remote target detection under a real-world natural setting.

Methods

To overcome the limitations of existing TBD methods, this paper provides the following key contributions: (1) An ideal “Normalized Difference Frame” calculation to perform VMF enhancement; and (2) A novel Constrained Velocity Matched Filter (CVMF) that combines known physical constraints with the target’s dynamic motion constraints to enhance its SNR. Our processing workflow is summarized as shown in (Fig. 2).

Image stabilization

To eliminate motion jitters on the camera induced by wind, we used the first frame of the video as a reference frame and registered subsequent frames from the video onto the reference frame. This was accomplished by using a frame-to-frame registration technique as described in [34] to create a stabilized frame.

Background estimation

The stabilized frame was then fed to a temporal background estimator so that the background subtraction could be performed. The process of background subtraction can be expressed mathematically using the following equation:

$$D(t) = F_s(t) - B(t-1) \tag{1}$$

where $D(t)$, corresponds to the Difference Frame at time t, $F_s(t)$ corresponds to the stabilized frame at time t, and $B(t-1)$ corresponds to the background computed in the previous time step. For simplicity in implementation, the popular Gaussian Mixture Modeling (GMM) background estimation method [17] was used in our processing. However, it is important to note that our method can also be applied to other temporal background estimation methods such as the Principal Component of Pursuit [18], and Subspace Tracking techniques [19].

Noise estimator

In general, a background cannot be perfectly estimated regardless of which background estimation method used. Hence, it is important to model a deviation of background models. To model the estimated background deviation, we estimated the temporal variance v of frame pixel location (i,j) at each time step t using an Infinite Impulse Response (IIR) filter with the following equation:

$$v\left(i,j,t\right) = (1-\gamma)D\left(i,j,t\right)^2 + \gamma v\left(i,j,t-1\right) \tag{2}$$

where γ is the variance update rate [0,1].

The temporal standard deviation for pixel (i,j) at time t, is obtained using the following equation:

$$\sigma\left(i,j,t\right) = \sqrt{v\left(i,j,t\right)} \tag{3}$$

Difference frame normalization

Pixels in different parts of an image can have different temporal standard deviation, depending on factors such as the environment and the scene structure. For example, the temporal standard deviation of the pixels in the waterfall region with constant running water is much higher than the pixels of an empty field. Hence, it is important to normalize the Difference Frame with respect to its temporal noise estimation before any thresholding is applied. The Normalized Difference Frame N_d for frame pixel location $\left(i,j\right)$ in time t is expressed as follow:

$$N_d = \frac{D\left(i,j,t\right)}{\sigma\left(i,j,t-1\right)} \tag{4}$$

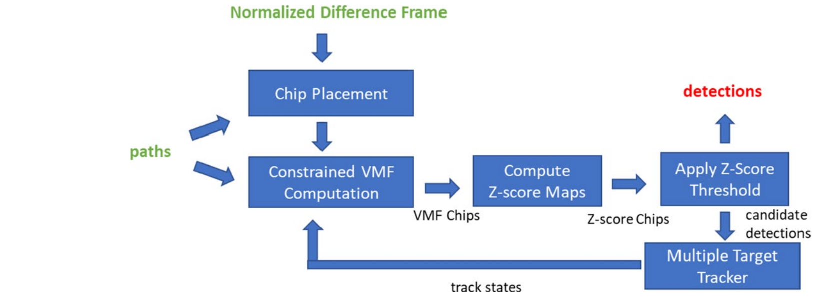

Fig. 3 Constrained velocity matched filter process

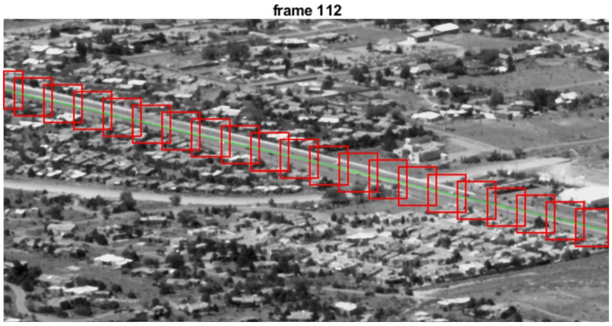

Fig. 4 Constraint processing illustration (processing region is denoted by red box, road path is denoted by green line)

While numerous existing methods attempt to detect objects on Difference Frames [13–16], our method attempts to find objects on the Normalized Difference Frame.

Constrained velocity matched filter

The Constrained Velocity Matched Filter (CVMF) uses a combination of physical constraint and motion estimation constraint to find, match, and integrate target signals along a motion path to enhance the target's SNR. Performing operation on the Normalized Difference Frame is more ideal because it reduces the risk of enhancing the noise on high noise region areas (e.g., high scene contrast region, waterfalls, etc.). For detecting vehicles in this video, a physical road constraint is imposed in the CVMF processing. However, for other applications, other constraints can be used, such as railroads for trains, pathways inside a building. A summary of the CVMF method is depicted in (Fig. 3).

Given the road constraints, we divided the path into different numbers of processing region (called "chips") along the road in the Normalized Difference Frame. An illustration is shown in (Fig. 4). The size of each "chip" used was 65 × 65 pixels. In general, the size of the "chip" should be selected based on the knowledge of the target's size and the path. For example, the region selected should be big enough to cover the width of the path with enough margin to account for path uncertainties. In addition, the region should be large enough to include non-targeted areas.

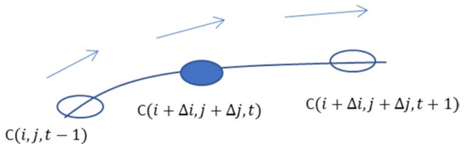
$C(i+\Delta i, j+\Delta j, t)$
$C(i+\Delta i, j+\Delta j, t+1)$
$C(i, j, t-1)$

Fig. 5 Example of an object's movement multiple time steps

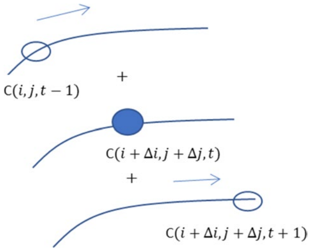
$C(i, j, t-1)$
$+$
$C(i+\Delta i, j+\Delta j, t)$
$+$
$C(i+\Delta i, j+\Delta j, t+1)$

Fig. 6 Shifting and adding operation

The continuous VMF process [20, 21] can be implemented in a discrete form, by shift-and-add operation with different velocity hypotheses along the path region in both forward and backward direction. For instance, suppose an object's movement is within the camera's view over a sequence time step as illustrated in (Fig. 5). For each processing chip, we can perform a range of shift-and-add operation for a range of velocities in attempt to match the target's movement over a period of time (Fig. 6).

The "sum chip" is the summation of individual chips over the temporal window. Mathematically, this can be expressed as the following:

$$S_k(i,j,t) = C(i+\Delta i, j+\Delta j, t-w) + \ldots C(i,j,t) + \ldots C(i+\Delta i, j+\Delta j, t+w) \tag{5}$$

where S, is the summation of the pixel (i,j) across multiple frames. $(\Delta i,\ \Delta j)$ corresponds to the shift positions, and w, represents the frame window for the summation, and k corresponds to the index of the matched hypothesis. The total number of matched hypothesis K can be expressed as:

$$K = M * N \tag{6}$$

where M is the number of directional hypotheses and N is the number of velocity hypotheses. Since the movement of the individual targets are constrained in a predetermined path, M is 2 in most cases (either forward or backward direction). M can be greater than 2 when the chip is at the intersection. The number of velocities depends on the target's speed. The units of the velocity in the target's movement can generally be described in fractions of pixels per frame. We started with an initial set of velocities and allowed for further refinement once a track has been established.

To find the detection in the sum chip S for a given hypothesis k, we first normalized the sum chip to form a Z-score chip. We can do this by computing the mean μ_s and standard deviation σ_s of the sum chip S. For dense target scenarios, it is recommended that a trim mean is used instead, to avoid high SNR targets inflating the mean estimates.

$$\mu_s = \frac{1}{p}\sum_{p=1}^{P} S(p) \tag{7}$$

$$\sigma_s = \sqrt{\frac{1}{P}\sum_{p=1}^{P}\left(S(p) - \mu_s\right)^2} \tag{8}$$

Then, we compute the Z score of the sum chip Z_s for each pixel $\left(i,j\right)$ using the following equation:

$$Z_s\left(i,j\right) = \frac{S\left(i,j\right) - \mu_s}{\sigma_s} \tag{9}$$

The following thresholding logic is applied to perform detection.

If $(|Z_s\left(i,j\right)| \geq T)$, then pixel $\left(i,j\right)$ is a candidate detection.

Pixel detected locations are generated from all hypotheses. They are consolidated to eliminate redundant detections from each chip. Adjacent pixel detections are clustered to represent a single target.

The centroid of the target's cluster is then fed to the Multiple Target Tracker (MTT) for association and tracking. To simplify, MTT is implemented using a simple 4-state constant velocity model [35]. An object's dynamic movement can be expressed mathematically using the following equations:

$$\begin{aligned} \boldsymbol{x}(t) =& \boldsymbol{A}\boldsymbol{x}(t-1) + \boldsymbol{q}(t-1), \quad \boldsymbol{q}(t) \sim N(0,\boldsymbol{Q}) \\ \boldsymbol{y}(t) =& \boldsymbol{H}\boldsymbol{x}(t) + \boldsymbol{r}(t), \quad \boldsymbol{r}(t) \sim N(0,\boldsymbol{R}) \end{aligned} \tag{10}$$

where $\boldsymbol{x}$ corresponds to the state vector, $\boldsymbol{y}$ corresponds to the output vector, $\boldsymbol{A}$ corresponds to the system matrix, and $\boldsymbol{H}$ corresponds to the output matrix. The system includes additive process noise q and measurement noise r, which are modeled as white noise gaussian with zero mean**.** The constant velocity model can be expressed in the following form:

$$\begin{aligned} x_1(t) &= x_1(t-1) + \Delta T x_3(\mathrm{t}-1) + \mathrm{q}_1 \\ x_2(t) &= x_2(\mathrm{t}-1) + \Delta T x_4(t-1) + q_2 \\ x_3(t) &= x_3(t-1) + q_3 \\ x_4(t) &= x_4(t-1) + q_4 \end{aligned} \tag{11}$$

where x_1, x_2 represents the positions of the object, and x_3, x_4 corresponds to the velocity state of each position component, and ΔT, corresponds to delta time changes between the state update.

In matrix form, this can be expressed as:

$$\begin{aligned} \boldsymbol{x}(t) &= \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}(t-1) + \boldsymbol{Q}, \\ \boldsymbol{y}(t) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{x}(t) + \boldsymbol{R} \end{aligned} \tag{12}$$

where $\boldsymbol{Q}$, is the process noise matrix, and $\boldsymbol{R}$, is the measurement noise matrix. Kalman Filtering can be used to predict and update the state estimates and its covariance estimate $\boldsymbol{P}$ at each time step.

Prediction steps:

$$\begin{aligned} \hat{\boldsymbol{x}}(k|k-1) &\boldsymbol{=} \hat{\boldsymbol{x}}(k-1|k-1) \\ \boldsymbol{P}(k|k-1) &\boldsymbol{=} \boldsymbol{AP}(k-1|k-1)\boldsymbol{A}^{\boldsymbol{T}} + \boldsymbol{Q} \end{aligned} \tag{13}$$

Update steps:

$$\begin{aligned} \boldsymbol{K}(k) &= \boldsymbol{P}(k|k-1)\boldsymbol{H}^{\boldsymbol{T}}(\boldsymbol{HP}(k|k-1)\boldsymbol{H}^{\boldsymbol{T}} + \boldsymbol{R})^{-1} \\ \hat{\boldsymbol{x}}(k|k) &= \hat{\boldsymbol{x}}(k|k-1) + \boldsymbol{K}(\boldsymbol{y}(k) - \boldsymbol{H})\hat{\boldsymbol{x}}(k|k-1) \\ \boldsymbol{P}(k|k) &= (\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{H})\boldsymbol{P}(k|k-1) \end{aligned} \tag{14}$$

As the target(s) are being tracked, the state vectors $\hat{\boldsymbol{x}}$ associated with covariance $\boldsymbol{P}$ (motion constraint) are fed back to the CVMF process to fine tune the pre-defined velocity bins and improve the accuracy of matching. Having feedback from a tracker to CVMF also adds robustness to maintain the tracking of moving objects in a temporary occlusion (e.g., a car temporarily obscured by a tree). The tracker's state is capable of propagating to the next time step, assuming the target is traveling in a similar speed without the need to re-initialize VMF filters. Different applications might require a more sophiscated modeling of dynamic behavior such as the target's acceleration [35].

Results and discussion

We compared our method with our baseline processing method as depicted in (Fig. 7). The baseline processing workflow is the same as the one in (Fig. 2) but without the additional CVMF component. The same video was used as an input to both processing methods. Both methods were evaluated over the same detection areas in the same regions of the image. Valid detections in the frames were manually labeled to provide assessment for the probability of detection and false alarm measures.

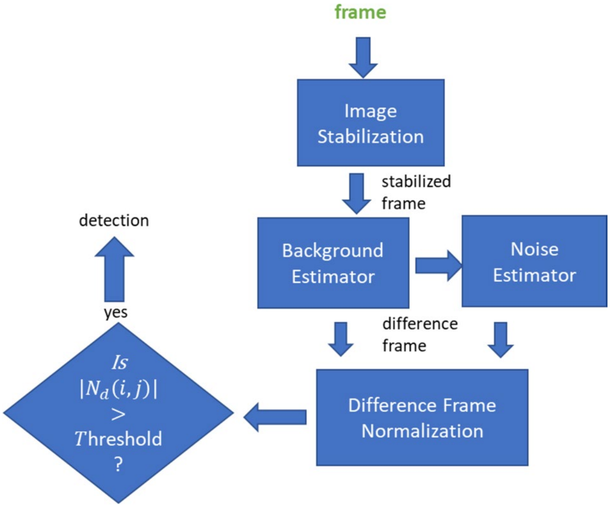

Fig. 7 Baseline detection processing

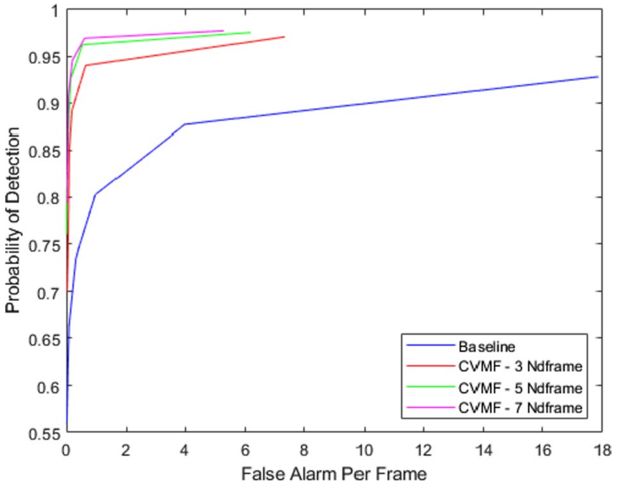

Fig. 8 ROC curves comparison—normalized difference frame

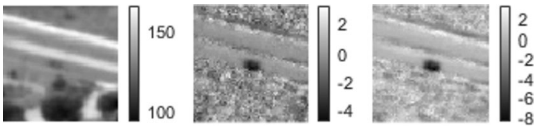

Fig. 9 Target enhancement (left—original, center- baseline normalized difference, right—CVMF 5-frame Z-scores)

Figure 8 shows a comparison of the Receiver Operating Characteristics (ROC) data curve as a baseline, along with different window of frames used in the CVMF calculation. The ROC curve improves as the window of frames increases; however, it reaches an asymptotic state at seven frames. A more sophisticated motion model is probably needed to integrate the target's motion over a longer framed window. An example of a qualitative comparison is depicted in (Fig. 9). The baseline Normalized Difference Frame shows a very low SNR target at around 4.0. After incorporating the CVMF enhancement, the target's SNR increased to 8.0.

To support our claim of performing VMF operations on Normalized Difference Frames instead of operating on Difference Frames, (Fig. 10) shows the ROC curve comparison of a baseline single Difference Frame thresholding versus using the CVMF method operating on Difference Frames. An example of a qualitative comparison is depicted in (Fig. 11). Though CVMF is also effective in boosting SNR on the Difference Frame domain, it does not perform as well on the Normalized Difference Frame. A direct ROC curve comparison of CVMF operating on Difference Frame versus CVMF operating Normalized Difference Frame is shown in (Fig. 12). As shown, operation on Normalized Difference Frames significantly outperforms operation on Difference Frames.

Conclusion

In this paper, we have provided a new method to enhance detection of low SNR targets. The innovation of this work is evident by the issuance of a U.S. patent [36]. Our CVMF method incorporates physical constraints from known roads and dynamic motion

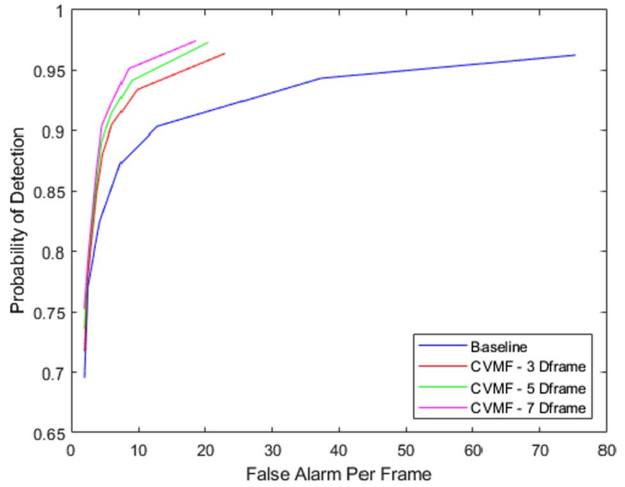

Fig. 10 ROC curve comparison—difference frame

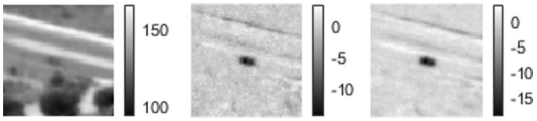

Fig. 11 Target enhancement (left—original frame, center—baseline difference, right—CVMF 5-frame difference)

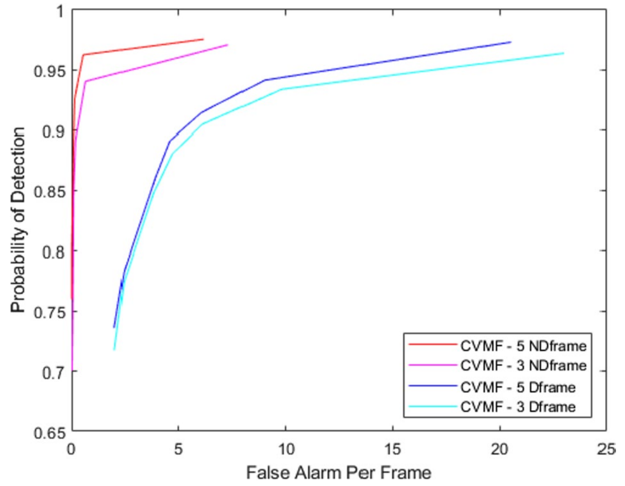

Fig. 12 CVMF normalized difference vs CVMF difference

constraints obtained from a Kalman Tracker to accurately find, match, and integrate target signals over multiple frames to improve the target's SNR. We have demonstrated our results using real data collected by an actual sensor. Our method has established a significant improvement over baseline traditional detection techniques. In addition, we have proven that our technique can achieve better performance if the CVMF operations were performed under the Normalized Difference Frames. Currently, our CVMF performance converges to a steady state at seven frames. In the future, we plan to incorporate a more sophisticated motion model (e.g., constant acceleration model) to support a longer frame integration window. A more sophisticated motion model can potentially provide a more accurate estimation of the target's motion over a longer period of time.

Abbreviations

SNL: Sandia National Laboratories; CVMF: Constrained velocity matched filter; VMF: Velocity matched filter; SNR: Signal-to-noise-ratio; TBD: Track-Before-Detect; ROC: Receiver operating characteristics.

Acknowledgements

We would like to thank Gary Whitlow (Electronics Technologist) at Sandia National Laboratories for collecting the video data.

Authors' contributions

TJM is the author of the paper. He contributed to the original conceptual design of the algorithm and software prototype. The author read and approved the final manuscript.

Authors' information

Tian Ma is a Distinguished R&D Computer Scientist at Sandia National Laboratories. He has over 18 years of experience in data analysis, data processing, and data exploitation of remote sensing systems, especially in the nuclear nonproliferation arena. He is a nationally recognized expert in detection algorithms and a pioneer in the field of tracking systems where he has innovated and delivered state-of-the-art real-time detection and tracking algorithms to operational systems that have solved U.S. government technical challenges and provided new, needed mission enabling capabilities. He has numerous patents and is honored with multiple national awards. He received a B.S. in Computer Engineering and an M.S. in Electrical and Computer Engineering from the University of Illinois at Chicago and an M.B.A. in Management of Technology from the University of New Mexico.

Funding

The work of this paper is funded by Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Availability of data and materials

Data is currently resided in Sandia National Laboratories. Additional approval may be needed for release of collected data.

Declarations

Ethics approval and consent to partcipate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competition interests.

Author details

[1]Sandia National Laboratories, Albuquerque, NM 87185, USA. [2]Sandia National Laboratories, Livermore, CA 94550, USA.

Received: 17 May 2021 Accepted: 12 September 2021
Published online: 02 October 2021

References

1. Ma TJ, Garcia RJ, Danford F, Patrizi L, Galasso J, Loyd J. Big data actionable intelligence architecture. J Big Data. 2020. https://doi.org/10.1186/s40537-020-00378-7.
2. Wang Y, Luo Z, Jodoin P-M. Interactive deep learning method for segmenting moving objects. Pattern Recognit Lett. 2016. https://doi.org/10.1016/j.patrec.2016.09.014.
3. Bideau P, Chowdhury AR, Menon RR, Learned-Miller E. The best of both worlds: combining CNNs and geometric constraints for hierarchical motion segmentation. IEEE Conf Comput Vis Pattern Recognit (CVPR). 2018.
4. Lim LA, Keles HY. Learning multi-scale features for foreground segmentation. Pattern Anal Applic. 2019. https://doi.org/10.1007/s10044-019-00845-9.
5. Barbedo JGA. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. Comput Electron Agric. 2018;153:46–53. https://doi.org/10.1016/j.compag.2018.08.013.

6. Vakalopoulou M, Karantzalos K, Komodakis N, Paragios N. Building detection in very high resolution multispectral data with deep learning features. IEEE Int Geosci Remote Sens Symp. 2015;2015:1873–6. https://doi.org/10.1109/IGARSS.2015.7326158.
7. Rudin C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell. 2019;1:206–15. https://doi.org/10.1038/s42256-019-0048-xD.
8. Sun C, Shrivastava A, Singh S, Gupta A. Revisiting unreasonable effectiveness of data in deep learning era. IEEE Int Conf Comput Vis. 2017. https://doi.org/10.1109/ICCV.2017.97.
9. Deahl D. Volvo's self-driving cars are having trouble recognizing kangaroos. The verge, July 3. 2017. https://www.theverge.com/2017/7/3/15916076/volvo-self-driving-cars-trouble-recognizing-kangaroos. Accessed 10 May 2021.
10. Akhtar N, Mian A. Threat of adversarial attacks on deep learning in computer vision: a survey. IEEE Access. 2018;6:14410–30. https://doi.org/10.1109/ACCESS.2018.2807385.
11. Yuan X, He P, Zhu Q, Li X. Adversarial examples: attacks and defenses for deep learning. IEEE Trans Neural Netw Learning Syst. 2019;30(9):2805–24. https://doi.org/10.1109/TNNLS.2018.2886017.
12. Piccard M. Background subtraction techniques: a review. 2004 IEEE Int Conf Syst Man Cybern. 2004;4:3099–104. https://doi.org/10.1109/ICSMC.2004.1400815.
13. Jacques J, Claudio RJ, Soraia RM. Background subtraction and shadow detection in grayscale video sequences. In 18th Brazilian symposium on computer graphics and image processing. IEEE. 2005; pp. 189–196.
14. Yang J, Yang W, Li M. An efficient moving object detection algorithm based on improved GMM and cropped frame technique. In: IEEE Int Conf Mechatron Autom. IEEE. 2012; pp. 658–663.
15. Yin J, Liu L, Li H, Liu Q. The infrared moving object detection and security detection related algorithms based on w4 and frame difference. Infrared Phys Technol. 2016;77:302–15.
16. Sengar SS, Mukhopadhyay S. Moving object detection based on frame difference and W4. SIViP. 2017;11:1357–64. https://doi.org/10.1007/s11760-017-1093-8.
17. Zivkovic Z. Improved adaptive Gaussian mixture model for background subtraction. In proceedings of the 17th international conference on pattern recognition, 2004. ICPR 2004. 2004. Vol 2, pp. 28–31. https://doi.org/10.1109/ICPR.2004.1333992.
18. Zhou Z, Li X, Wright J, Candès E, Ma Y. Stable principal component pursuit. In 2010 IEEE international symposium on information theory. 2010; pp. 1518–22. https://doi.org/10.1109/ISIT.2010.5513535.
19. Vaswani N, Bouwmans T, Javed S, Narayanamurthy P. Robust subspace learning: robust PCA, robust subspace tracking, and robust subspace recovery. IEEE Signal Process Mag. 2018;35(4):32–55. https://doi.org/10.1109/MSP.2018.2826566.
20. Reed IS, Gagliardi RM, Shao HM. Application of three-dimensional filtering to moving target detection. IEEE Trans Aerosp Electron Syst. 1983;AES-19(6):898–905.
21. Reed IS, Gagliardi RM, Stotts LB. A recursive moving-target-indication algorithm for optical image sequences. IEEE Trans Aerosp Electron Syst. 1990;26(3):434–40.
22. Yi W, Morelande MR, Kong L, Yang J. An efficient multi-frame track-before-detect algorithm for multi-target tracking. IEEE J Sel Top Signal Process. 2013;7(3):421–34.
23. Orlando D, Venturino L, Lops M, Ricci G. Track-before-detect strategies for STAP radars. IEEE Trans Signal Process. 2010;58(2):933–8.
24. Buzzi S, Lops M, Venturino L. Track-before-detect procedures for early detection of moving target from airborne radars. IEEE Trans Aerosp Electron Syst. 2005;41(3):937–54.
25. Buzzi S, Lops M, Venturino L, Ferri M. Track-before-detect procedures in a multi-target environment. IEEE Trans Aerosp Electron Syst. 2008;44(3):1135–50.
26. Grossi E, Lops M, Venturino L. A novel dynamic programming algorithm for track-before-detect in radar systems. IEEE Trans Signal Process. 2013;61(10):2608–19.
27. Grossi E, Lops M, Venturino L. A heuristic algorithm for track-before-detect with thresholded observations in radar systems. IEEE Signal Process Lett. 2013;20(8):811–4.
28. Chan YT, Niezgoda GH, Morton SP. Passive sonar detection and localization by matched velocity filtering. IEEE J Ocean Eng. 1995;20(3):179–89.
29. Wu B, Yan H. A novel track-before-detect algorithm for small dim infrared target. In 2011 international conference on multimedia and signal processing (CMSP), 14–15 May 2011; 1:103–6.
30. Úbeda-Medina L, García-Fernández ÁF, Grajal J. Adaptive auxiliary particle filter for track-before-detect with multiple targets. IEEE Trans Aerosp Electron Syst. 2017;53(5):2317–30. https://doi.org/10.1109/TAES.2017.2691958.
31. Wang J, Yi W, Kirubarajan T, Kong L. An efficient recursive multiframe track-before-detect algorithm. IEEE Trans Aerosp Electron Syst. 2018;54(1):190–204. https://doi.org/10.1109/TAES.2017.2741898.
32. Yi W, Fang Z, Li W, Hoseinnezhad R, Kong L. Multi-frame track-before-detect algorithm for maneuvering target tracking. IEEE Trans Veh Technol. 2020;69(4):4104–18. https://doi.org/10.1109/TVT.2020.2976095.
33. Grossi E, Lops M, Venturino L. A track-before-detect algorithm with thresholded observations and closely-spaced targets. IEEE Signal Process Lett. 2013;20(12):1171–4.
34. Guizar-Sicairos M, Thurman ST, Fienup JR. Efficient subpixel image registration algorithms. Opt Lett. 2008;33:156–8.
35. Rong Li X, Jilkov VP. Survey of maneuvering target tracking. Part I. Dynamic models. IEEE Trans Aerosp Electron Syst. 2003;39(4):1333–64. https://doi.org/10.1109/TAES.2003.1261132.
36. Ma TJ. Object detection and tracking system. U.S. Patent No. 9,665,942. Issued 30 May 2017.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.