Chapter

Surveillance with UAV Videos

İbrahim Delibaşoğlu

Abstract

Unmanned aerial vehicles (UAVs) and drones are now accessible to everyone and are widely used in civilian and military fields. In military applications, UAVs can be used in border surveillance to detect or track any moving object/target. The challenge of processing UAV images is the unpredictable background motions due to camera movement and small target sizes. In this chapter, a short literature brief will be discussed for moving object detection and long-term object tracking. Publicly available datasets in the literature are introduced. General approaches and success rates in the proposed methods are evaluated and approach to how deep learning-based solutions can be used together with classical methods are discussed. In addition to the methods in the literature for moving object detection problems, possible solution approaches for the challenges are also shared.

Keywords: surveillance, moving object, motion detection, foreground detection, object tracking, long-term tracking, UAV video, drones

1. Introduction

Unmanned aerial vehicles (UAV) and drones are now accessible to everyone and are widely used in civilian and military fields. Considering security applications, drones could be used in applications such as surveillance, target detection and tracking. Drone surveillance allows us to continuously gather information about a tracked target from a distance. So drones with the capabilities of features such as object tracking, autonomous navigation, and event analysis are a hot topic in computer vision society. The challenge of processing drone videos is the unpredictable background motion due to camera movement. In this chapter, a short literature brief, potential approaches to improve the moving object detection performance, will be discussed and publicly available datasets in the literature will be introduced. In addition, the current situation of deep learning-based solutions, which give good results in many research areas, in motion detection and potential solutions will be discussed. General approaches and success rates in the proposed methods will be shared, and approaches on how deep learning-based solutions can be used together with classical methods will be proposed. In brief, we propose some post-processing techniques to improve the performance of background modeling-based methods, and software architecture to speed up operations by dividing them into small parts.

Section 2 represents moving target detection issues from UAV videos, while Section 2.1 represents how to build a simple background model. Section 2.2 introduces sample datasets for moving target detection and Section 2.3 gives potential approaches to enhance the background modeling approach for moving target detection. Some object tracking methods that can be used together with moving object detection and Convolutional Neural Network (CNN) based methods are emphasized in Sections 3 and 4, respectively. Finally, the conclusion is discussed in Section 5.

2. Moving object detection

The problem of detecting moving objects is a computer vision issue that is needed in areas such as real-time object tracking, event analysis and security applications. Based on the computer vision literature carried out in recent years, it is a problem that has been studied extensively [1]. The purpose of moving object detection is to classify the image as foreground and background. The classification could be challenging according to factors such as the motion state of the camera, ambient lighting, background cluttering, and dynamic changes in the background. Images obtained from cameras mounted on drones have a free motion, and it causes much background motion (also called global motion in the literature). Another important issue is that these images could be taken in very different regions such as mountains, nature, forests, cities, rural areas, and they can contain very small targets according to the altitude of the UAV.

In moving object detection applications, the aim is to have high accuracy as well as real-time operation of the application. When the studies carried out in the literature are examined, it is seen that subtraction of consecutive frames, background modeling and optical flow-based methods are used. Although the subtraction of consecutive frames method works fast and can adapt quickly to background changes, the success rate is very low [2]. In the background modeling approach, a background model (an image formed as a result of the average of the previous *n* frames) is extracted using frames history [3]. Classical image processing techniques [4], statistical methods [5–7] and neural networks [8] have been used for background modeling in the literature. Gaussian mixture model (GMM) [9] builds a Gaussian distribution model for each pixel and adaptive GMM [7] improves it for dynamic background. Kim et al. [10] propose a spatio-temporal Gaussian model minimizing image registration errors. Zhong Z. et al. [11] propose a background updating strategy performing at both pixel and object levels and apply a pixel-based adaptive segmentation method. Dual-target non-parametric background modeling method [12] proposes a dual-target updating strategy to eliminate false detection caused by background movements and illumination changes. Scene conditional background update method [13], named SCBU, builds a statistical background model without contamination of the foreground pixels. Background subtraction is applied between current frame and updated background model while calculated foreground likelihood map is used to extract initial foreground regions by applying high and low threshold values. In MCD method [6], a dual-mode single Gaussian model is proposed with age, mean and variance of each pixel, and it compensates for the camera motion by mixing neighboring approaches. Simple threshold with respect to the variance is applied in MCD method for foreground detection. Yu et al. [14] use a candidate background model similar to MCD and they propose a method to update candidate or main background model pixels in each frame. In background subtraction step, they apply the neighborhood subtraction approach, which takes into account the neighbors of each pixel. BSDOF [15] method extracts candidate foreground masks with background subtraction and applies threshold for variance value of each pixel. In background subtraction process, also uses dense optical flow to weigh the difference for each pixel. Then, it obtains a final mask with the combination of candidate masks and region growing strategy using candidate masks. Thus, false detection is largely eliminated.

For background modeling approach in moving cameras (such as cameras mounted to UAVs), global motion is generally eliminated by using homography matrix obtained by Lucas Kanade [16] (KLT) and RANSAC [17] methods. Selected points in the previous frame are tracked in the current frame with KLT and homography matrix representing global (camera) motion is calculated with RANSAC method. Then, previous frame or background model is warped to the current frame to eliminate the global motion. Sample grid-based selected points and estimated positions are visualized as flow vectors in **Figure 1**.

One of the biggest problems in using only pixel intensity values is that these kinds of methods are so sensitive to illumination changes and registration errors caused by homography errors. As a solution to these issues, different features such as texture [18], edge [19] and haar-like [20] are proposed in the literature. Edge and texture features can better address the illumination change issue and also eliminate the ghosting effect left by foreground objects. Local Binary Pattern (LBP) and its variants [21, 22] are other types of texture feature used for foreground detection in the literature. In addition to such additional features, deep learning methods that offer effective solutions to many problems have also been used in the foreground detection problem. For this purpose, FlowNet2 [23] architecture estimating optical flow vectors are used in foreground detection problems [24]. Optical flow means the displacement of each pixel in consecutive frames. KLT method is also an optical flow method that tracks the given points in the consecutive frame and it is categorized as sparse optical flow. On other hand, estimating pixel displacement of each pixel is called dense optical flow. FlowNet2 is one of the most known architectures which also has publicly available pre-trained weights. The disadvantage of deep learning methods is that they require much computational cost, especially for high-dimensional images, and may not perform well for so small targets due to the training image dimensions and

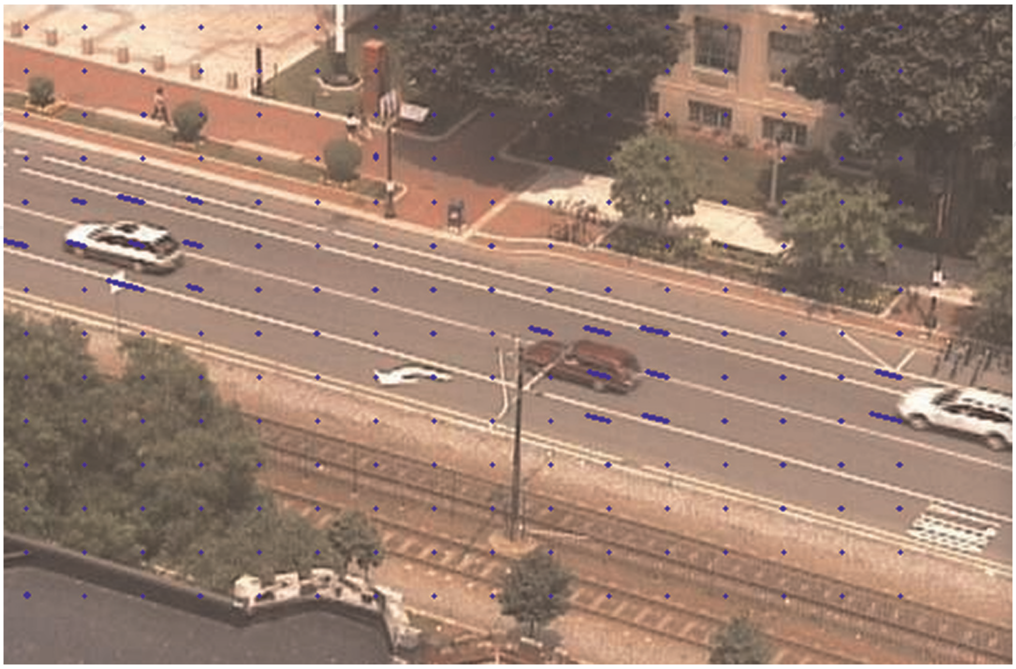

Figure 1.
Visualization of flow vectors for grid points.

contents. Considering that UAV images may contain a lot of small targets, it can be thought that the optical flow model to be trained with small moving object images could perform better. On the other hand processing, high-dimensional input images require much RAM in the GPU. **Figure 2** shows sample visualization of optical flows for FlowNetCSS (which is pre-trained model that mostly detects the small changes and more lightweight according to FlowNet2), Farneback and Nvidia Optical Flow (NVOF). FlowNetCSS is a sub-network of Flownet2.

In this work, we have used FlowNet pre-trained weights which have been trained on MPI-Sintel dataset [25] containing images with the resolution of 1024 × 436. **Figure 3** shows the FlowNetCSS output on 1920 × 1080 resolution images from PESMOD dataset [26]. In **Figure 4**, the model is runned for a patch of the frames instead of full resolution and it performs better for the small targets (two people hiking in the mountain). Simple thresholding could be applied for optical flow matrices and the foreground mask showing the moving pixels could be obtained directly. But for small targets, it may be useful to process the small regions as shown in **Figure 4**. Global motion compensation with homography matrix may also be used

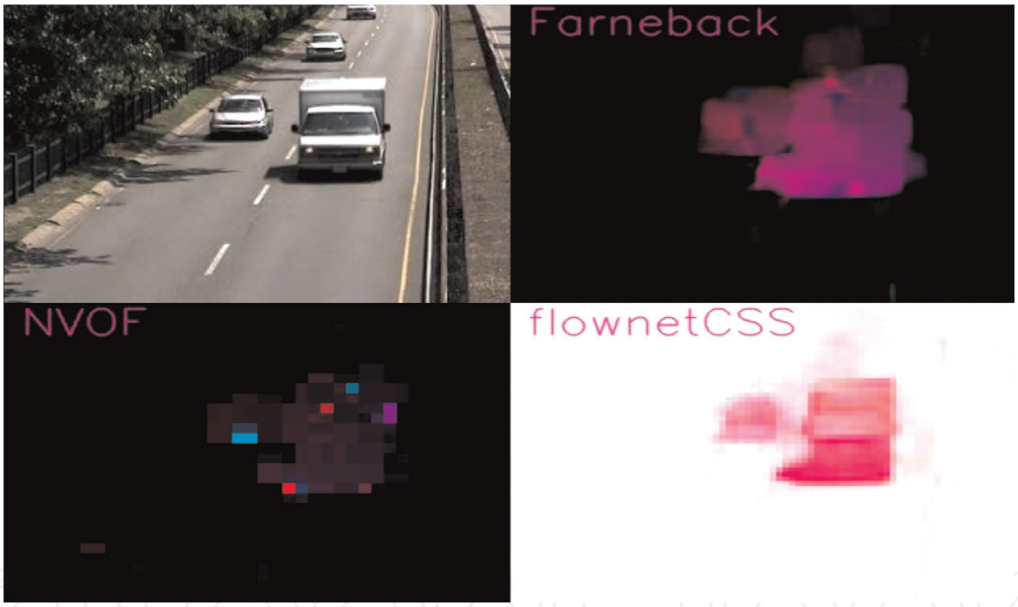

Figure 2.
Visualization of optical flow vectors of FlowNetCSS, Farneback and NVOF.

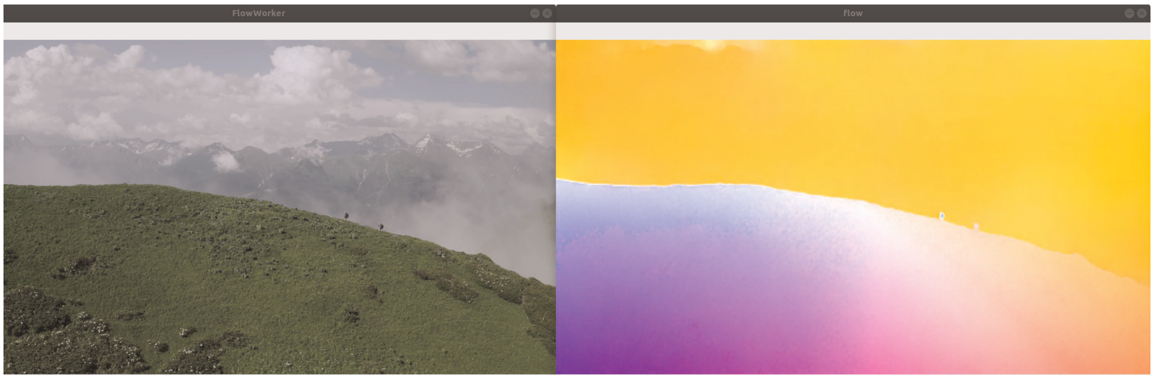

Figure 3.
FlowNet visualization on PESMOD [26] sample frames.

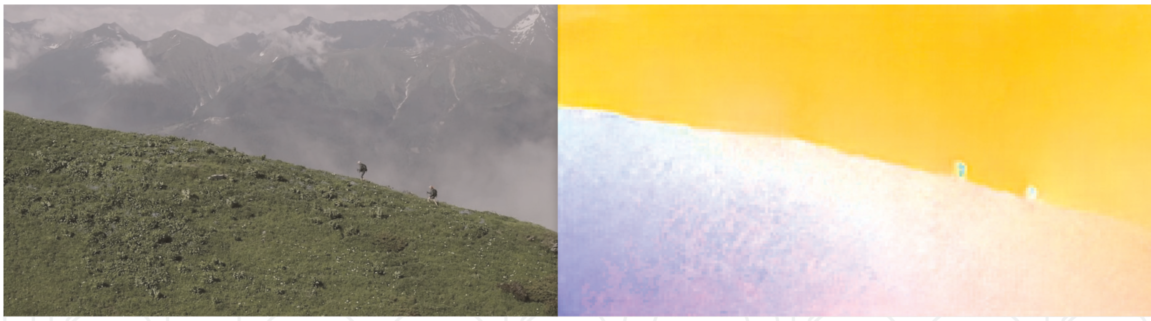

Figure 4.
FlowNet visualization on a patch of PESMOD [26] sample frames.

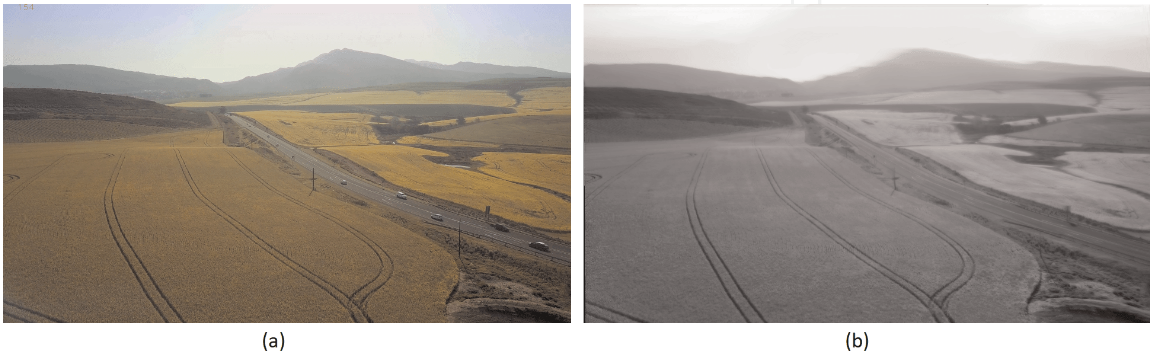

Figure 5.
(a) Sample frame (b) Background model μ image.

before estimating dense optical flow, so simple thresholding can give the moving pixels with better accuracy.

2.1 Building a background model

Consider that H represents homography matrix between frames in time $t-1$ and t. The background model B at time of *t-1* is warped to the current frame by using Eq. (1). Thus, the pixels in the background and current frame are aligned to handle global motion. $\alpha(i)$ represents the learning rate of each pixel while $\mu_t(i)$ represents average pixel values. The background model B consists of mean and learning values as shown in Eq. (2) and (3).

$$B_t = H_{t-1} B_{t-1} \tag{1}$$

$$\alpha(i) = \frac{1}{age(i)} \tag{2}$$

$$\mu_t(i) = (1 - \alpha_t(i))\,\mu_{t-1}(i) + \alpha(x)\ \ I_t(i) \tag{3}$$

In the equations, I represents a frame while i represents a pixel in a frame. Learning rates (α) is determined with the *age* value of each pixel. Sample frame and background image is shown in the **Figure 5** for maximum age value 30.It is also important to set pixels whose age is less than a fixed threshold value to zero. Because the pixels that have just entered the frame need to wait for a while to be evaluated. After building the background model, current frame is subtracted from μ image to obtain a foreground mask. But using a simple model with only RGB coloir features is so sensitive to errors like shadow, ghost effect, illumination changes and background motion. Thus, it is important to use extra texture features for background modeling as mentioned in Section 2. In Chapter 2.3 we discuss some approaches to improve the performance of BSDOF method using color features effectively.

2.2 Datasets

Changedetection.net (CDNET) [27] dataset is a large-scale video dataset consisting of 11 different categories, but only PTZ subsequence consists of images taken by moving camera. PTZ sequence does not include free motion so it is not so appropriate to evaluate motion detection problem for UAV images. SCBU dataset [13] includes images of walking pedestrians with a free motion camera. The VIVID [28] dataset consisting of aerial images is a good candidate to evaluate moving object detection methods. It consists of moving vehicle images and has a resolution of 640x480. PESMOD [15] dataset represents a new challenging high-resolution dataset for evaluation of small moving object detection methods. It includes eight different sequences with a resolution of 1920x1080 and consists of small moving targets (vehicles and humans). PESMOD dataset contains totally of 4107 frames and 13,834 labeled bounding boxes for moving targets. The details of each sequence is given in **Table 1**.

Average precision (Pr), recall (R) and f1 (F_1) score values of MCD, SCBU and BSDOF methods for PESMOD dataset are given in **Table 2**. In the Eq. (4), FP refers to wrongly detected boxes, TP refers to the number of true detections, and FN refers to ground truth boxes that is missed by the method. Pr indicates the accuracy of positive predictions (estimated as motion) while R (also named sensitivity) represents the ratio of the number of pixels correctly classified as foreground (motion) to the actual

Sequence name	Number of frames	Number of moving objects
Pexels-Elliot-road	664	3416
Pexels-Miksanskiy	729	189
Pexels-Shuraev-trekking	400	800
Pexels-Welton	470	1129
Pexels-Marian	622	2791
Pexels-Grisha-snow	115	1150
Pexels-zaborski	582	3290
Pexels-Wolfgang	525	1069

Table 1.
The details of PESMOD dataset.

Metrics	MCD [6]	SCBU [13]	BSDOF [15]
Precision	0.3928	0.3248	**0.4890**
Recall	**0.4163**	0.3127	0.4061
F1 score	0.2856	0.3072	**0.3898**

Bold values in the table represents the best score for each row.

Table 2.
Comparison of average precision, recall and f1 score values of MCD, SCBU and BSDOF methods on PESMOD dataset.

number of foreground pixels. F_1 score is the combination of Pr and R, and is equal to 1 for perfect classification

$$Pr = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F_1 = \frac{2^* Pr^* R}{Pr + R} \tag{4}$$

The BSDOF method is suitable to implement in the GPU. It runs at about 26 fps for 1920x1080 on a PC with Ubuntu 18.04 operation system, AMD Ryzen 53,600 processor with 16 GB RAM, and Nvidia GeForce RTX2070 graphic card. MCD runs at about 8 fps on the same machine. SCBU is also implemented for CPU and we have used the binary files. So that we could not measure the processing time of the SCBU method.

2.3 Prospective solutions for challenges

As mentioned in the detailed review article [29], we can say that the main challenges are still dynamic backgrounds, registration errors and small targets. Using extra features like LBP for better performance also increases the computational cost, it is not suitable for real-time requirements of high dimensional videos. Therefore, an alternative solution might be to create a background model by only using color features and process the texture features only for the extracted candidate target regions. This allows to avoid extracting texture features for each pixel. In addition to texture features, classical methods and/or Deep Neural Networks (DNN) can be used to find a similarity score between background image and current frame for candidate target regions. Structural Similarity (SSIM) score [30] can be used to measure the similarity between image patches. As an alternative, any pre-trained CNN model could be used for feature extraction. But using a lightweight sub-network is important since it will be applied to many candidate regions. **Figure 6** shows sample detected bounding boxes with BSDOF method on PESMOD dataset. **Table 3** represents average SSIM scores between current frame and background image patches for ground truth and false positives (FP).

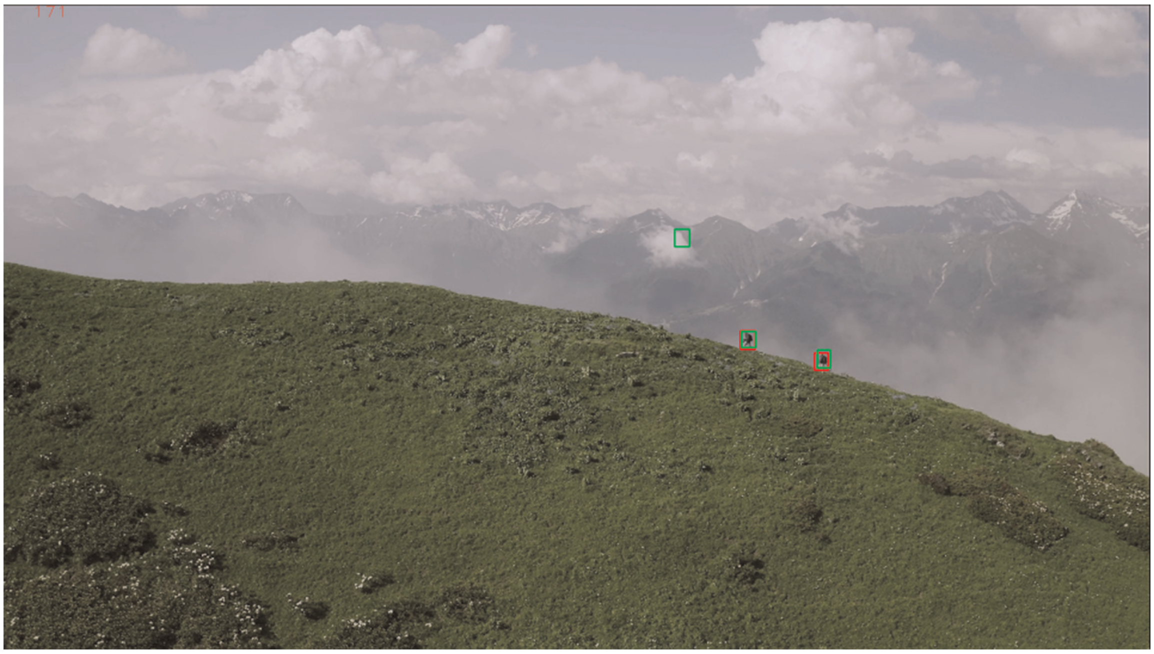

Figure 6.
Moving object detection output of BSDOF for Pexels-Shuraev-trekking sequence.

Sequence name	SSIM (GT)	SSIM (FP)
Pexels-Elliot-road	0.2569	0.3930
Pexels-Miksanskiy	0.3525	0.7599
Pexels-Shuraev-trekking	0.3511	0.6493
Pexels-Welton	0.4164	0.4671
Pexels-Marian	0.3797	0.3934
Pexels-Grisha-snow	0.4164	0.3875
Pexels-zaborski	0.4290	0.3691
Pexels-Wolfgang	0.3410	0.6077

Table 3.
SSIM scores for ground truth (GT) and false positives (PF) of BSDOF method.

Experiments with similarity comparison results show that it can be useful to eliminate some false detections caused by registration errors and illumination changes. Similarity score is expected high for false detection (no moving objects) and low for moving object regions. But, as a result of our observations, it has been observed that the similarity measure can be low in very small areas such as 5x5 pixels and in regions with no moving object. The background model can be blurred for some pixels due to registration error and/or moving background. It results low similarity score for these cases. In general, extreme wrong detections could be eliminated with a high threshold value not to lose the true detections.

Image registration errors cause possible false detection, especially for objects with sharp edges. Even if similarity comparison can help to eliminate false detection, simple tracking approaches could also be used for this issue. Historical points of each detection are stored in a *tracker list*, and detection for each frame is compared to the *tracker list*. So, tracked regions can be classified by hit count (number of detection consecutive frames) and total pixel displacements. However, it should be noted that coordinate values in the *tracker list* must be adjusted for each frame to eliminate global motion. This approach will work well if the moving target region can be extracted successfully in consecutive frames and the bounding boxes overlap with high intersection of union (IOU) value for good matching. As an alternative approach, a robust tracking method can be used but probably requires more computational cost. Targets detected with the moving object detection algorithm can be tracked with a robust tracker to obtain more precise results, and thus the tracking process continues in case the target stops.

As another approach, classical background modeling and deep learning-based methods can be used in collaboration with different processes. Our experiments show that classical methods suffer more from image registration errors, especially for fast camera movements. Therefore, the classical method and deep learning results can be combined using different strategies according to camera movement speed. Alternatively, dense optical flow with deep learning could be applied only for small patches detected by classical background modeling. In order to implement such an approach a software infrastructure in which background modeling and deep learning methods working in different processes communicate with each other and share data is essential in terms of speed. It allows us to run the processes in a pipeline logic to speed up the algorithm as shown in **Figure 7**. In the proposed architecture, process-1 applies classical background modeling approach and informs process-2 to start via zeroMQ.

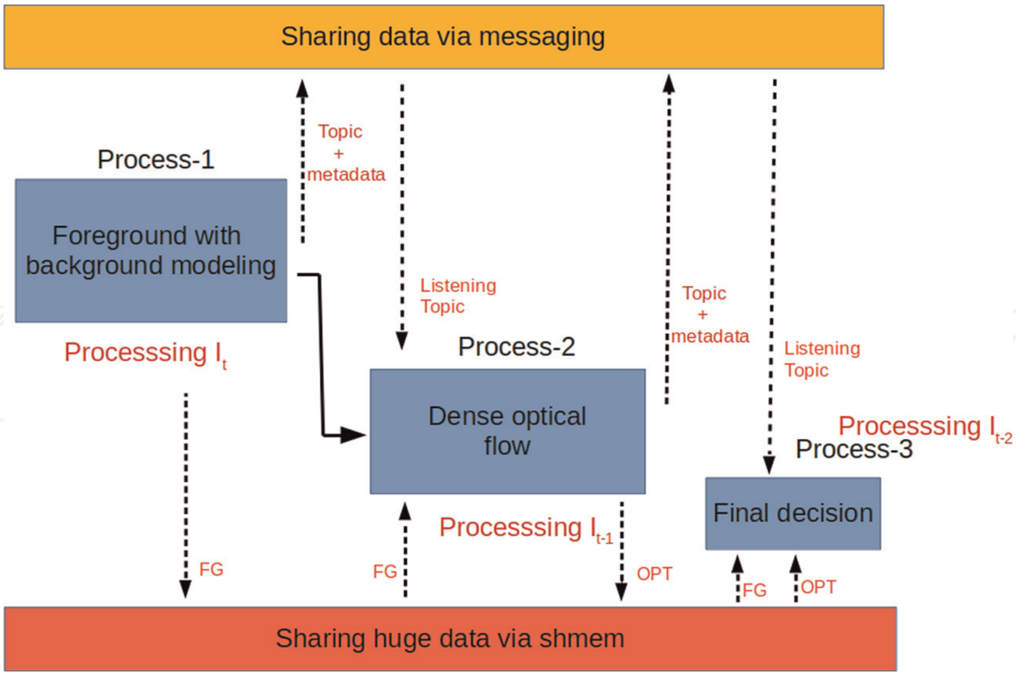

Figure 7.
Software architecture to run processes in pipeline logic.

ZeroMQ messaging library is used to transfer meta-data and inform the other processes to begin to process the frame that is ready. The foreground mask cannot be shared via messaging protocols in real-time, so that shared memory (shmem) is used to transfer this huge data between processes. Accordingly, the foreground mask is transferred to process-2 with shared memory and process-2 applies deep learning based dense optical flow only for patches extracted from input foreground mask. Finally, process-3 estimates moving target bounding boxes by processing dense optical flow output. Process-1 processes I_t while process-2 processes I_{t-1} with such a parallel working structure created in the pipeline logic.

3. Object tracking with UAV images

Object tracking is the re-detection of a target in consecutive frames after the tracker is initialized with the first bounding box as an input. It is a challenging problem for situations such as fast camera movement, occlusion, background movements, cluttering, illumination and scale changes. Tracking methods can be evaluated in different categories such as detection-based tracking, detection-free tracking, 3D object tracking, short-term tracking and long-term tracking. Detection-based tracking requires an object detector and tracking indicates assigning ID for each object. Detection-free tracking can be preferred for UAV images to handle any kind of targets and small-sized objects which is hard to detect with an object detector. As a simple approach, we can consider that we can eliminate the wrong detections after following each candidate moving object region and confirming the movement of the object with the tracker. Then we can decide for moving object with the output of the tracker. Thus, target tracking can be used in cooperation with motion detection to increase accuracy and provide better tracking.

The software architecture suggested in the previous section also seems reasonable to implement the tracking method applied after the motion detector. In this section,

	Precision	Recall	F1
KCF [33]	0.4456	0.1214	0.1908
CSRT [34]	0.5006	0.5573	0.5275
ECO [35]	0.4965	0.5241	0.5099
TLD [32]	0.2460	0.4523	0.3186
Re3(S) [36]	0.4680	0.8030	0.5913

Table 4.
Performance comparison of tracker methods on UAV123 long-term tracking sequences.

we compare the performances of some tracker methods on UAV123 dataset [31].The dataset consists of a total of 123 video sequences obtained from low-altitude UAVs. The 20 subset images in the dataset are evaluated separately for long-term object tracking, in which targets sometimes occludes, appear and disappear, providing a better benchmark for long-term tracking. We compare performances of classical methods such as TLD [32], KCF [33], CSRT [34], ECO [35] and deep learning based method Re3 [36]. In classical methods, only TLD can handle disappeared targets in long-term tracking. Even if ECO and CSRT trackers are successful for tracking non-occluded objects, they do not have a mechanism to re-detect the object after failed. TLD can recover from full occlusion but produces frequent false positives. KCF is faster than TLD, CSRT and ECO but has lower performance. ECO and CSRT has reasonable performances except oclusion and recovering case specially important in long-term tracking. On the other hand, lightweight Re3 model can track objects at higher FPS (about 100–150 according to the GPU specifications). It allows us to track multiple objects in real-time. Average tracker performances are represented in **Table 4** for UAV123 long-term subset sequences.

Re3(S) indicates the small (lightweight) re3 model in the **Table 4** and average score shows that Re3 has the best recall score by far. In the performance comparison, the moving target detection is considered true (TP) if the intersection of union (IOU) between predicted and ground truth bounding box is greater than 0.5. Experiments show us that a moving object algorithm with tracking method support will provide significant advantages both in eliminating wrong detection and in continuous tracking.

4. Training CNN for moving object detection

Deep learning based solutions are an important alternative to eliminate the disadvantage of classical methods for moving object detection problem, because background modeling based methods suffer from high number of false detections. We have mentioned the deep learning based optical flow studies at the beginning of the chapter. This section summarizes the situation for supervised deep learning methods performed in the problem of moving object detection.

Deep learning based methods outperform the classical image processing based methods in CDNET dataset, but CDNET does not contain free motion images/videos. CDNET ground truths are pixel-wise masks of moving objects. FgSegNetV2 [37] is a encoder-decoder type deep neural network, and performs well on the CDNET dataset. MotionRec [38] is a single-stage deep learning framework proposed for moving object detection problem. It firstly estimates the background representation from past history frames with a temporal depth reduction block. The temporal and spatial features are used to generate multi-level feature pyramids with a backbone model. Finally, multi-level feature pyramid is used in the regressing and classification layers. MotionRec runs in the range of 2 to 5 fps depending on the selected temporal history depth from 10 to 30, over Nvidia Titan Xp GPU. JanusNet [39] is another deep network trained for moving object detection problem from UAV images. It tries to extract and combine dense optical flow and generates a coarse foreground attention map. Experiments show that it efficiently detects small moving targets. JanusNet is trained with a simulated dataset, which is generated using Unreal Engine 4. It runs at 25fps on Nvidia GTX1070 GPU and 3.1 fps on Nvidia Jetson Nano for 640 × 640 resolution images. JanusNet has also a performance comparison with the FgSegNetV2, and it shows that FgSegNetV2 cannot perform well for UAV videos due to requiring to be trained on a specific scene to work well on that scene. Considering the deep learning studies in the literature and the datasets used for training the model, it can be said that there is still a long way to go for a general-purpose supervised moving object detection method. On the other hand, classical methods can achieve reasonable results with the additional post-processing techniques and most importantly, they can work in real-time even at Nvidia modules at the edge.

5. Conclusions

This chapter discusses the moving object detection problem for UAV videos. We represent datasets, the performance of some methods in the literature, the challenges, and prospective solutions. For motion detection, especially background modeling-based methods are emphasized, and some post-processing methods are proposed to improve the performance as a solution to the challenges. We propose dense optical flow and simple tracking as a post-processing step with specific software architecture. Moreover, we evaluate selected trackers on a long-term object tracking dataset to analyze the performances of the trackers. Finally, we introduce some deep learning architectures and compare traditional methods in terms of general-purpose and real-life use.

Author details

İbrahim Delibaşoğlu
Faculty of Computer and Information Sciences, Department of Software Engineering,
Sakarya University, Sakarya, Turkey

*Address all correspondence to: ibrahimdelibasoglu@sakarya.edu.tr

References

[1] Chapel M, Bouwmans T. Moving objects detection with a moving camera: A comprehensive review. Computer Science Review. 2020;**38**:100310

[2] Collins R, Lipton A, Kanade T, Fujiyoshi H, Duggins D, Tsin Y, et al. A system for video surveillance and monitoring. VSAM Final Report. 2000; **2000**:1

[3] Bouwmans T, Hofer-lin B, Porikli F, Vacavant A. Traditional approaches in background modeling for video surveillance. Handbook Background Modeling And Foreground Detection For Video Surveillance. Taylor & Francis Group; 2014

[4] Allebosch G, Deboeverie F, Veelaert P, Philips W. EFIC: Edge based foreground background segmentation and interior classification for dynamic camera viewpoints. International Conference On Advanced Concepts For Intelligent Vision Systems. 2015. pp. 130-141

[5] Zivkovic Z, Van Der Heijden F. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters. 2006;**27**:773-780

[6] Moo Yi K, Yun K, Wan Kim S, Jin Chang H, Young Choi J. Detection of moving objects with non-stationary cameras in 5.8 ms: Bringing motion detection to your mobile device. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2013. pp. 27-34

[7] Zivkovic Z. Improved adaptive Gaussian mixture model for background subtraction. Proceedings of the 17th International Conference on Pattern Recognition. 2004. pp. 28-31

[8] De Gregorio M, Giordano M. WiSARDrp for Change Detection in Video Sequences. ESANN; 2017

[9] Stauffer C, Grimson W. Adaptive background mixture models for real-time tracking. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149). 1999. pp. 246-252

[10] Kim S, Yun K, Yi K, Kim S, Choi J. Detection of moving objects with a moving camera using non-panoramic background model. Machine Vision and Applications. 2013;**24**:1015-1028

[11] Zhong Z, Zhang B, Lu G, Zhao Y, Xu Y. An adaptive background modeling method for foreground segmentation. IEEE Transactions on Intelligent Transportation Systems. 2016;**18**: 1109-1121

[12] Zhong Z, Wen J, Zhang B, Xu Y. A general moving detection method using dual-target nonparametric background model. Knowledge-Based Systems. 2019; **164**:85-95

[13] Yun K, Lim J, Choi J. Scene conditional background update for moving object detection in a moving camera. Pattern Recognition Letters. 2017;**88**:57-63

[14] Yu Y, Kurnianggoro L, Jo K. Moving object detection for a moving camera based on global motion compensation and adaptive background model. International Journal of Control, Automation and Systems. 2019;**17**: 1866-1874

[15] Delibasoglu I. Real-time motion detection with candidate masks and region growing for moving cameras.

Journal of Electronic Imaging. 2021;**30**: 063027

[16] Tomasi C, Kanade T. Detection and tracking of point. International Journal of Computer Vision. 1991;**9**:137-154

[17] Fischler M, Bolles R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM. 1981;**24**: 381-395

[18] Heikkilä M, Pietikäinen M, Heikkilä J. A texture-based method for detecting moving objects. BMVC. 2004; **401**:1-10

[19] Huerta I, Rowe D, Viñas M, Mozerov M, Gonzàlez J. Background Subtraction Fusing Colour, Intensity and Edge Cues. Proceedings of the Conference on AMDO. 2007. pp. 279-288

[20] Zhao P, Zhao Y, Cai A. Hierarchical codebook background model using haar-like features. IEEE International Conference on Network Infrastructure and Digital Content. 2012. pp. 438-442

[21] Bilodeau G, Jodoin J, Saunier N. Change detection in feature space using local binary similarity patterns. International Conference on Computer and Robot Vision. 2013. pp. 106-112

[22] Wang T, Liang J, Wang X, Wang S. Background modeling using local binary patterns of motion vector. Visual Communications and Image Processing. 2012. pp. 1-5

[23] Ilg E, Mayer N, Saikia T, Keuper M, Dosovitskiy A, Brox T. Flownet 2.0: Evolution of optical flow estimation with deep networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. pp. 2462-2470

[24] Huang J, Zou W, Zhu J, Zhu Z. Optical flow based real-time moving object detection in unconstrained scenes 2018

[25] Butler D, Wulff J, Stanley G, Black M. A naturalistic open source movie for optical flow evaluation. European Conference on Computer Vision (ECCV). 2012. pp. 611-625

[26] Delibasoglu I. UAV images dataset for moving object detection from moving cameras. 2021

[27] Wang Y, Jodoin P, Porikli F, Konrad J, Benezeth Y, Ishwar P. CDnet 2014: An expanded change detection benchmark dataset. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2014. pp. 387-394

[28] Collins R, Zhou X, Teh S. An open source tracking testbed and evaluation web site. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. 2005. p. 35

[29] Garcia-Garcia B, Bouwmans T, Silva A. Background subtraction in real applications: Challenges, current models and future directions. Computer Science Review. 2020;**35**:100204

[30] Wang Z, Bovik A, Sheikh H, Simoncelli E. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing. 2004;**13**:600-612

[31] Mueller M, Smith N, Ghanem B. A benchmark and simulator for uav tracking. European Conference on Computer Vision. 2016;**2016**:445-461

[32] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011;**34**:1409-1422

[33] Henriques J, Caseiro R, Martins P, Batista J. High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2014;**37**:583-596

[34] Luke A, Voji T, Zajc L, Matas J, Kristan M. Discriminative correlation filter tracker with channel and spatial reliability. International Journal of Computer Vision. 2018;**126**(7):671-688

[35] Danelljan M, Bhat G, Shahbaz Khan F, Felsberg M. Eco: Efficient convolution operators for tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. pp. 6638-6646

[36] Farhadi D, Fox D. Re 3: Real-time recurrent regression networks for visual tracking of generic objects. IEEE Robotics and Automotive Letters. 2018; **3**:788-795

[37] Lim L, Keles H. Learning multi-scale features for foreground segmentation. Pattern Analysis and Applications. 2020; **23**:1369-1380

[38] Mandal M, Kumar L, Saran M. MotionRec: A unified deep framework for moving object recognition. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2020. pp. 2734-2743

[39] Zhao Y, Shafique K, Rasheed Z, Li M. JanusNet: Detection of moving objects from UAV platforms. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021. pp. 3899-3908