ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

SCHOOL OF COMPUTER AND COMMUNICATION SCIENCES

Environment Modeling for Unmanned Aerial Vehicles

Author
Vidit Vidit
vidit.vidit@epfl.ch

Supervisor
Rozantsev Artem
artem.rozantsev@epfl.ch

June 9, 2016

Abstract

The project presents an approach for Unmanned Aerial Vehicles(UAVs) detection. The method discussed in the report uses background subtraction techniques and cascade classifier to detect the UAVs. The proposed method has faster and improved detection rate compared to the direct usage of cascade classifier for the indoor scenes. However, improved results couldn't be achieved for the outdoor scenes.

Contents

1 Introduction

Over the last few years the usage of UAVs like drones and other aircrafts has seen a rise in both commercial and public domains. Their usage varies from the surveillance, aerial mapping to recreational purposes. Due to this wide scale acceptance, many useful applications can be developed where swarm of these drones can be employed to perform certain tasks or a drone has to navigate through an environment avoiding other UAVs. In order to perform these tasks, the detection of other UAVs plays an important role.

As the cameras are already present on the most of the UAVs, we can exploit them and using computer vision algorithms perform the detection task. This is the motivation behind the project. The real life scenarios present few key challenges in performing this task. The first and foremost is the that lighting conditions are never constant, so the algorithm should be robust to these changes. The UAVs will have sharp movement hence motion compensation is essential. Another important constraint is to make detection as fast as possible, ideally near real time with frame rate around 25-30 frames per second(fps).

1.1 Objectives

With overall goals and constraints known for this interesting problem, the scope for the Semester Project is limited to study and test the performance of background subtraction and classifier for drone detection in the videos of indoor and outdoor scenes.

1.2 Dataset

For the project, all the performance and timing profiling is done on the three indoor scenes and three outdoor scene made available by CVLAB. The frame resolution is 1080×1920 for indoor scenes and 480×752 for outdoor scenes.

The remainder of the report is organized as follows, chapter 2 discusses the methodology used and explains the details of the different algorithm used. The results of the algorithm are detailed in chapter 3. And finally chapter 4 concludes the report.

2 Methodology

The proposed final method has three key components i.e. Motion Compensation, Background Subtraction and Classifier. In the following sections these components are discussed more concretely. The dense optical flow approaches were also experimented but due to slow performance were not explored further.

2.1 Motion Compensation

The drone can move quite rapidly, hence the resulting sequence of images captured from drone's camera will have objects displaced by a large margin even if they are static. This motion has to be compensated in order to properly apply background subtraction techniques as they work best with static camera scenes.

The motion compensation starts with extracting feature points(FPs) in the frames and tracking these feature point in the subsequent frames. The Good Features to Track [1] is used to get the feature points. In order to track these feature points in subsequent frames, optical flow method is employed. The corresponding feature points in two frames(f_i,f_{i-1}) are then used to estimate the perspective transformation(H_i). The current frame is projected back to the plane of first frame using inverse of the product of the predecessor frames' perspective transform T_i^{-1}.

$$T_i = T_{i-1} \times H_i \tag{2.1}$$

$$T_1 = H_1 = I \tag{2.2}$$

where, I is 3×3 identity matrix. For the first frame f_1, T_1 and H_1 are initialized as in eq. (2.2).

The drawback of the above method as such is that when scene changes drastically (for eg. change in the camera viewpoint by 90 degrees or more) the feature points cannot be tracked. There can be situations where the feature points are occluded or due to lighting change their flow can not be calculated. In order to mitigate these situations, feature points are recalculated after the number of tracked feature points decreases below a threshold(θ_{th}). Moreover, the current frame is not projected back to its predecessor' plane and transforms are initialized as eq. (2.3).

$$T_i = H_i = I \tag{2.3}$$

The number of FPs(N_{FP}) and θ_{th} are the parameters to be optimized for the motion compensation stage. Their values are presented in the section 3.1. fig. 2.1 presents the flowchart for the motion compensation algorithm for a sequence of images. An alternative approach is also implemented, where predecessor frames in background model(section 2.2) are projected forward to current frame keeping rest of the pipeline same.

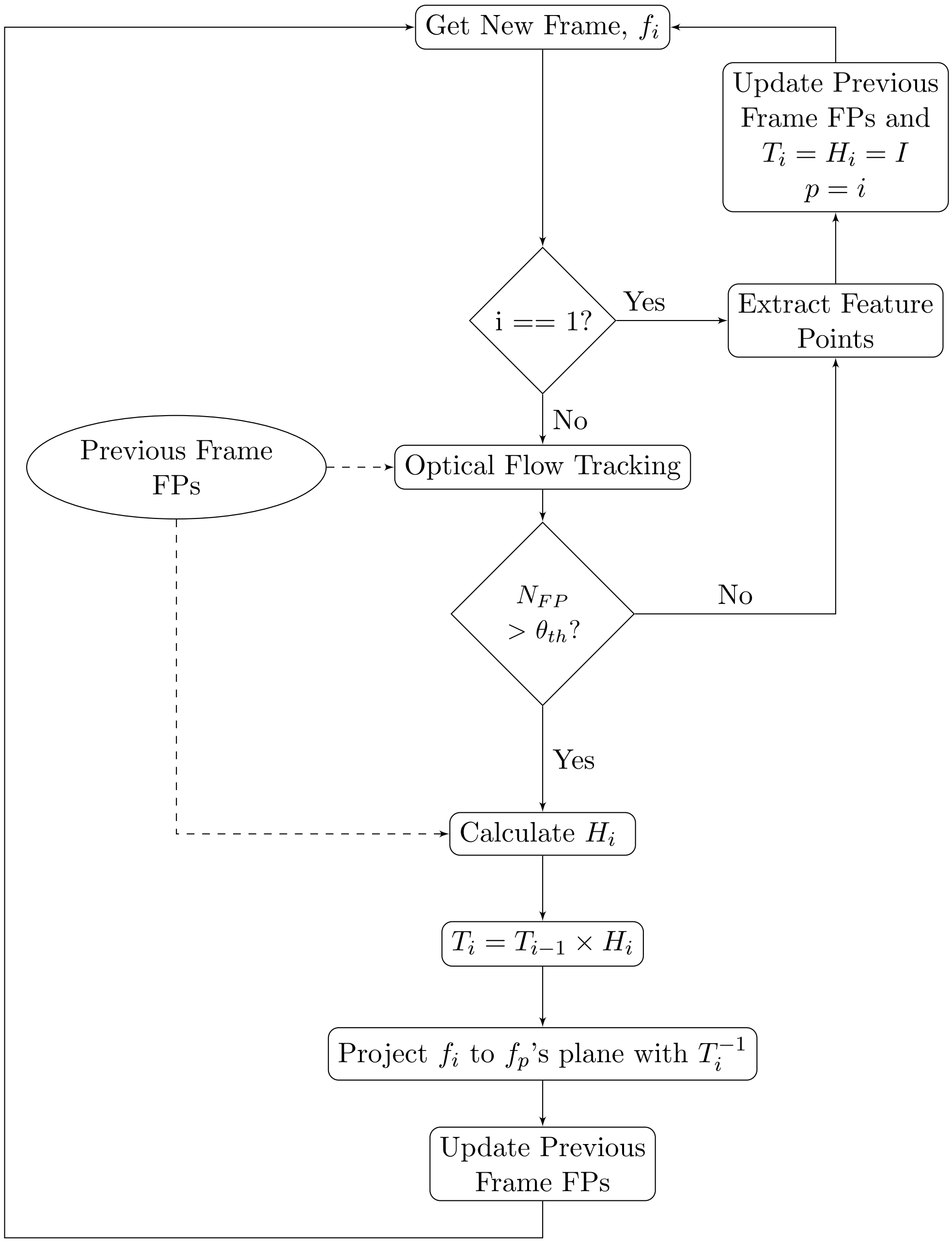


Figure 2.1: Motion Compensation Algorithm

2.2 Background Subtraction

The background subtraction techniques are used to identify the foreground objects in the scene. The different implementations of the background subtraction are tested using OpenCV[1], BGSLibrary [2], and Vibe [3].

In section 3.2, the comparison between the results obtained is discussed. The best results are obtained with OpenCV's Mixture of Gaussian(MOG2) Method [4], [5] and Vibe. Both of them maintain a background model and update it frame by frame. The MOG2 method assigns a probability to each pixel for belonging into foreground and background, whereas Vibe maintains a history of pixel values on which simple thresholding is done to classify the pixel to background or foreground. Furthermore, motion compensation for Vibe is done with projection of predecessor frames to current frame's plane. It is observed that MOG2 method is more robust to scene changes and identifies foreground quicker than Vibe.

2.3 Cascade Classifier

The Cascade Classifier consists of several simple classifiers that are applied to the frames in stages to detect the drones. The candidate region in the frame has to pass all the stages in order to be classified as region consisting the drone. The classifier is trained using positive images(frames having drones) and negative images(frames without drone) from the dataset mentioned in section 1.2. The OpenCV tool[2] [3] is used to train the classifier and classification.

2.4 Dense Optical Flow

The dense optical flow methods estimate the flow value for each pixel. Once the flow value is obtained for each pixel, a dominant motion can be estimated which inturn will be the motion of the drone with the on-board camera. The other non dominant motion will be candidate for the drones in the scene. The implementation provided in [6], [7] is used for the experimentation. This method was considered for the experimentation because motion compensation using FPs would not be essential, as the dominant motion would be known. The section 3.3 presents the performance of the method.

[1] OpenCV Background Subtraction `http://docs.opencv.org/master/d1/dc5/tutorial_background_subtraction.html`

[2] Cascade Classifier Training `http://docs.opencv.org/2.4/doc/user_guide/ug_traincascade.html`

[3] Cascade Classification `http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html`

2.5 Process Flow

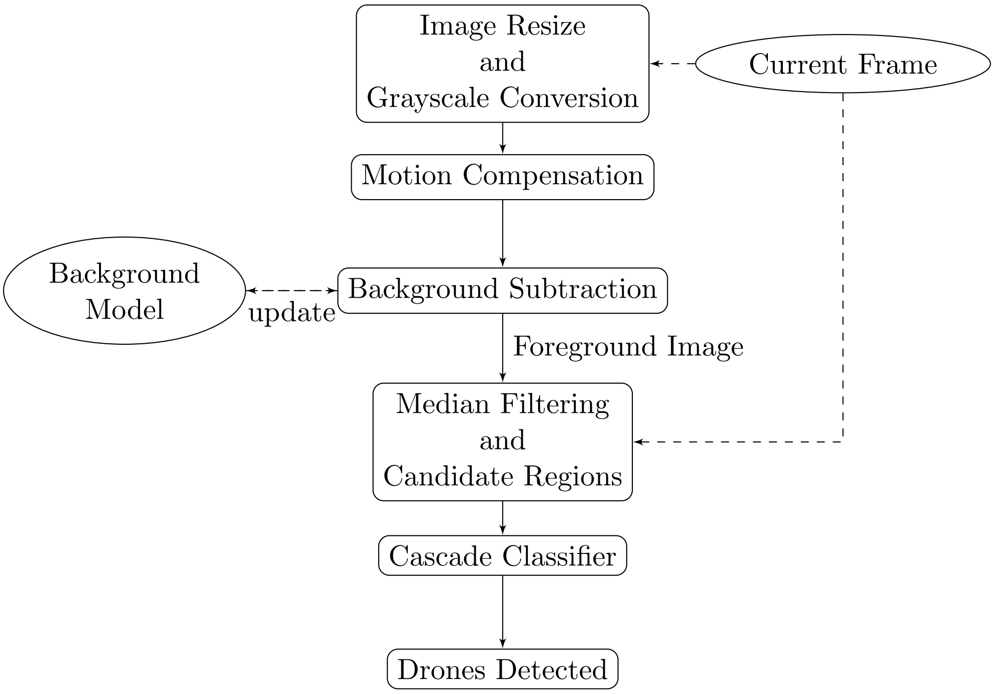


Figure 2.2: Flowchart of the proposed algorithm

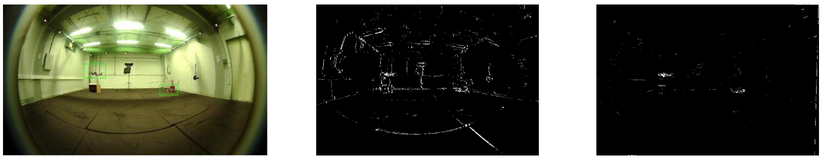
The flowchart 2.2 summarizes the overall algorithm. The image resize and colorspace conversion are Pre-Processing steps. The algorithm proceeds with motion compensation and background subtraction. Background Subtraction requires a background model for foreground detection and updates it with each frame. The foreground generated is post processed with median filter and candidate regions are obtained by multiplying it with original frame. Finally the cascade classifier is used to detect drones in the frame.

3 Results and Interpretations

The section discusses the results obtained for the different stages of the algorithm. MacBook Pro with 2.7 GHz Intel Core i5 processor and 8GB RAM is used to run and profile the algorithm.

3.1 Motion Compensation

The fig. 3.1 and fig. 3.2 illustrate the effect of motion compensation in improving the results of background subtraction. The background subtraction method used for these images is OpenCV's Mixture of Gaussian. Though, the motion compensation method described in section 2.1 is very naive approach but produced decent results and performance. The different values for the parameters i.e. N_{FP} and θ_{th} described in section 2.1, were tested and optimal result is obtained with $N_{FP} = 300$ and $\theta_{th} = 100$.



(a) Indoor Scene: Green box bounds the drone

(b) Background Subtraction without Motion Compensation

(c) Background Subtraction with Motion Compensation

Figure 3.1: Motion Compensation in Indoor Scene

(a) Outdoor Scene: Green box bounds the drone

(b) Background Subtraction without Motion Compensation

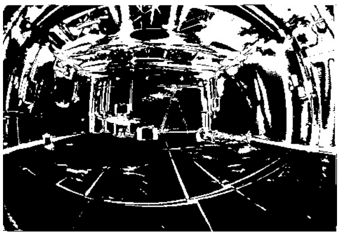
(c) Background Subtraction with Motion Compensation

Figure 3.2: Motion Compensation in Outdoor Scene

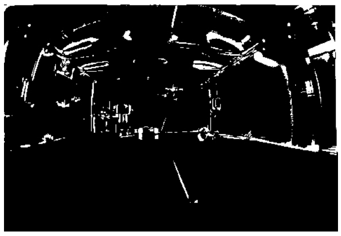
3.2 Comparison of Background Subtraction Methods

The different methods for background subtraction are experimented using the libraries mentioned in section 2.2. The BGSLibrary [2] provides many such algorithms but in [8] authors have illustrated that only selected algorithms work best with dynamic background. These methods are DPWrenGABGS [9], LBAdaptiveSOM [10], T2FGMM_UM [11], FuzzyChoquetIntegral [12], and DPEigenbackgroundBGS [13]. Additionally, OpenCV Mixture of Gaussian(MOG2) and Vibe [3] are tested. The results for both indoor and outdoor scenes are depicted in fig. 3.3 and fig. 3.4 respectively.

(a) Indoor Scene: Green Box bounds drone

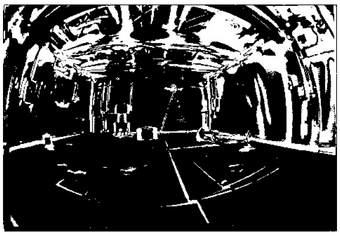


(b) DPEigenbackgroundBGS



(c) DPWrenGABGS

(d) FuzzyChoquetIntegral



(e) LBAdaptiveSOM

(f) T2FGMM_UM

(g) OpenCVMOG2

(h) Vibe

Figure 3.3: Comparison between different Background Subtraction Algorithms for Indoor Scene

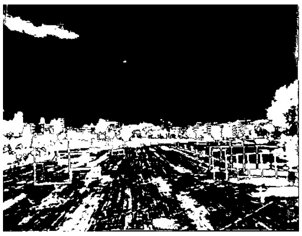
The decision of the quality of the foreground generated is by visual inspection. The results illustrate the OpenCV MOG2 and Vibe are the best among the others. Even after tuning the parameters for BGSLibrary methods result did not improve. Hence the only these two methods were taken for further analysis.

(a) Outdoor Scene: Green Box bounds drone

(b) DPEigenbackground-BGS

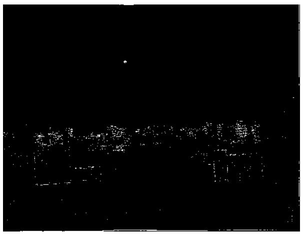
(c) DPWrenGABGS

(d) FuzzyChoquetIntegral



(e) LBAdaptiveSOM

(f) T2FGMM_UM



(g) OpenCVMOG2

(h) Vibe

Figure 3.4: Comparison between different Background Subtraction Algorithms for Outdoor Scene

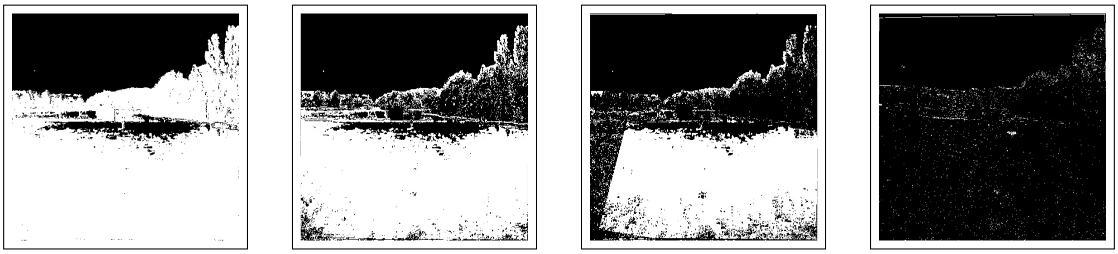
In the indoor scenes, both the methods are equally good. For the outdoor scenes, foregrounds generated by Vibe are mostly better than the OpenCV MOG2. The reason being that the OpenCV MOG2 output contains many noisy pixels. This can be solved using a median filter for on the foreground generated, but it could lead to another issue of filtering out drone when size of the drone blob is smaller filter window size.

With reasoning above Vibe stands out to be a better solution but it has another drawback. It recovers much slower from the drastic lighting change compared to the OpenCV MOG2. The fig. 3.5, fig. 3.6, and fig. 3.7 illustrate this problem. For this particular situation, it took 2 extra frames for OpenCV to produce proper foreground but Vibe couldn't produce decent foreground even after multiple frames.

Both methods have drawbacks but depending on the conditions choice between either two can be made.

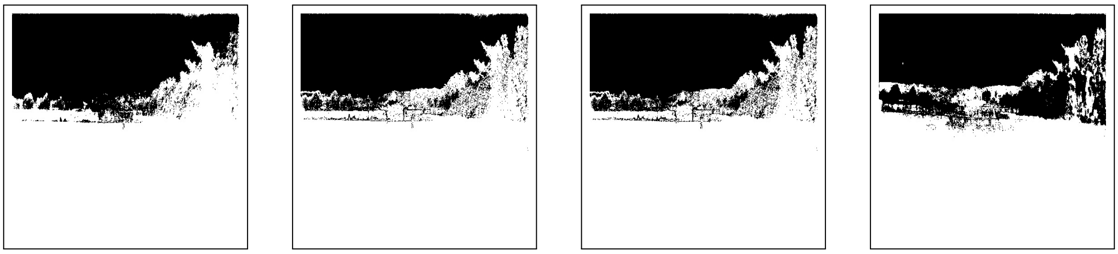
(a) at t (b) at t+1 (c) at t+2 (d) at t+3

Figure 3.5: Frame depicting change in the lighting condition



(a) at t+1 (b) at t+2 (c) at t+3 (d) at t+5, took 2 extra frames to recover from lighting change

Figure 3.6: Background Subtraction with OpenCVMOG2 under lighting change



(a) at t+1 (b) at t+2 (c) at t+3 (d) at t+15, even after 12 more frames foreground detected is improper

Figure 3.7: Background Subtraction with Vibe under lighting change

3.3 Comparison with Dense Optical Flow

The dense optical flow techniques [6] and [7] were tested to obtain the dominant motion in the images instead of feature based motion compensation. These methods did not work for the outdoor scenes as they where not able to handle changes in lighting condition,whereas worked for the indoor where lighting is much more constant. However, the major shortcoming of these methods is that they are computationally intensive so they ran slower than the background subtraction methods, depicted in the table 3.1. These methods were not explored further due this reason.

Method	**Average Time per Frame(in ms)**
T.Brox[6]	12000
Liu[7]	5000
OpenCV MOG2	0.8
Vibe	1.3

Table 3.1: Time Performance Comparison between Dense Optical Flow and Background Subtraction

3.4 Detection with Cascade Classifier

The foreground detected using Background Subtraction methods may still have few blobs which are not drones. This may occurs when motion compensation is not accurate or when the number of FPs goes down below a threshold leading to re-initialization of the background model. The Cascade Classifier is trained to detect the drones in the images and thus, reduce the false positives in the result.

Method	**Average Detection Time per Frame(in ms)**
Classifier without Background Subtraction scale = 1.05	73
Classifier with Background Subtraction scale = 1.05	61
Classifier with Background Subtraction scale = 1.15	25

Table 3.2: Comparison of Detection timing of Cascade Classifier with/without Background Subtraction

The Cascade Classifier is trained separately for both indoor and outdoor scenes. Histogram of Oriented Gradients(HOG) features are used in these classifiers. The detection is proper in the indoor scenes than the outdoor. The outdoor scene classifier produced more false positive detection. It is trained with 2052 positive images and 3000 negative images. The poorer detection could be because of the small number of negative images seen during training stage. For larger number of negative images training couldn't be performed due to time constraint of the project. Hence the further analysis is only presented for the indoor scenes.

The detected foreground image is multiplied with original image frames to get an image of candidate regions containing drones. The Cascade Classifier could be used without Background Subtraction but since candidate regions is reduced with Background Subtraction, detection time is also reduced. The Cascade Classifier has a scale levels parameter which enables it to detect same object irrespective of the size for which it was trained with. It is observed that even with coarser level of scale, drones are detected when background subtraction is used. This brought down the timing significantly. The table 3.2 list the timing details with different configuration.

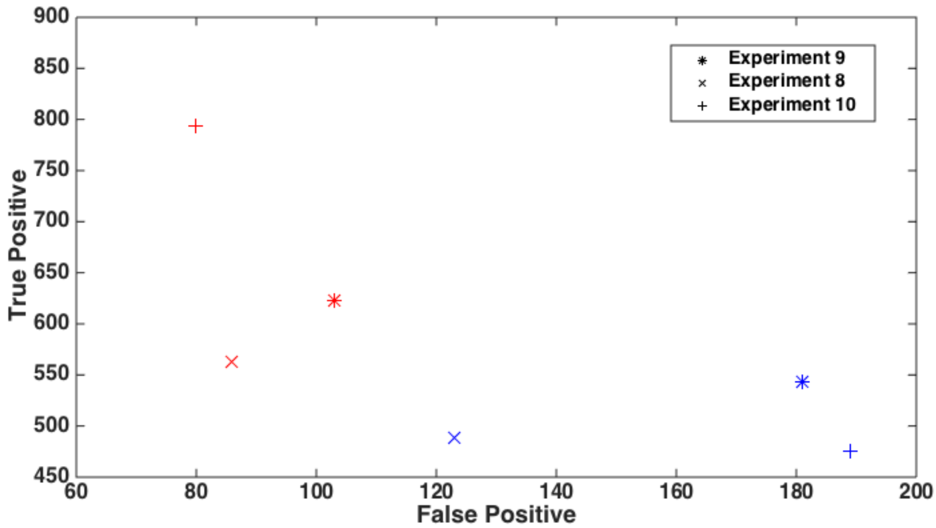


Figure 3.8: True Positive vs False Positive comparison between Cascade Classifier with Background Subtraction(in red) and without it(in blue).

The true positive vs false positive for classifier with(red)/without(blue) background subtraction is shown in fig. 3.8. The values are obtained for three indoor scenes. It can be seen that the number of the false positives reduced with the with background subtraction. This was known as the candidate regions were few foreground blobs instead of entire frame. There is an increase in the true positives with background subtraction and reason could that blobs are just big enough to bound the drones hence, HOG features do not include gradients of other objects occluded by drone in the environment.

The ground truth for the scenes are Experiment 8 - 1798,Experiment 9 - 1202, and Experiment 10 - 2002. On comparison with ground truth the true positives obtained are fairly low. It can be partially due to the scenes in which drones are hovering at place for a long period hence, leading to its inclusion in background model. Also sometimes foreground blobs are smaller parts of drone instead of entire drone, which leads to their non identification. The classifier itself requires better training and parameter tuning as its performance without background subtraction is not good.

Even with these shortcoming it is shown that augmentation of background subtraction method with classifier improves the detection.

3.5 Time Profiles

The breakdown of the time taken by different components described in of the algorithm (section 2.5) is presented for indoor(table 3.3) and outdoor(table 3.4) scene. The timings are the average value over dataset. It is interesting to see that the background subtraction takes a small fraction of overall pipeline's time. The classifier and pre-processing take most of the time. The timing for Opencv MOG2 and Vibe are not similar for the motion compensation part because of the different approach(section 2.2) followed in each. Since, the classifier couldn't be trained for outdoor scenes its timing is not presented in table 3.4.

	Average Time per Frame(in ms)	
Components	**OpenCV MOG2**	**Vibe**
Pre Processing	15.8	16.1
Motion Compensation with Perspective transform	6.3	7.6
Background Subtraction	0.85	1.2
Post Processing	0.31	0.28
Cascade Classifier	25	25
Total	48.26	50.18

Table 3.3: Time Profile For Indoor Scene

	Average Time per Frame(in ms)	
Components	**OpenCV MOG2**	**Vibe**
Pre Processing	10.1	9.8
Motion Compensation with Perspective transform	10.3	12.9
Background Subtraction	0.98	1.5
Post Processing	0.48	0.19
Total	21.86	24.39

Table 3.4: Time Profile For Outdoor Scene

4 Conclusion

The two best performing background subtraction methods i.e. Opencv MOG2 and Vibe can be used depending on the environment condition. Opencv MOG2 is best option if the lighting condition change quite often and Vibe is able to detect smaller moving objects in the scene. The latter is helpful when drones to be detected are quite far from the camera. For the motion compensation stage more sophisticated methods can be tested to find corresponding feature points in two frames.

As discussed in the previous chapter the proposed method has few drawbacks, nevertheless it performed decently in the drone detection task for the indoor scenes. The algorithm can be extended for the outdoor scenes with a trained classifier. As the results suggest the usage of background subtraction along with classifier detection is faster and more reliable. The proposed method can also be extended to run on the drones using their on-board processors.

Bibliography

[1] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, IEEE, 1994, pp. 593–600.

[2] A. Sobral, "Bgslibrary: An opencv c++ background subtraction library," in *IX Workshop de Visao Computacional (WVC'2013)*, Rio de Janeiro, Brazil, Jun. 2013. [Online]. Available: `https://github.com/andrewssobral/bgslibrary`.

[3] M. Van Droogenbroeck and O. Barnich, "Vibe: A disruptive method for background subtraction," *Background Modeling and Foreground Detection for Video Surveillance*, pp. 7–1, 2014.

[4] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, IEEE, vol. 2, 2004, pp. 28–31.

[5] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern recognition letters*, vol. 27, no. 7, pp. 773–780, 2006.

[6] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 3, pp. 500–513, 2011.

[7] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," PhD thesis, Citeseer, 2009.

[8] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Computer Vision and Image Understanding*, vol. 122, pp. 4–21, 2014.

[9] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 780–785, 1997.

[10] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *Image Processing, IEEE Transactions on*, vol. 17, no. 7, pp. 1168–1177, 2008.

[11] F. El Baf, T. Bouwmans, and B. Vachon, "Type-2 fuzzy mixture of gaussians model: Application to background modeling," in *Advances in Visual Computing*, Springer, 2008, pp. 772–781.

[12] ——, "Fuzzy integral for moving object detection," in *Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE World Congress on Computational Intelligence). IEEE International Conference on*, IEEE, 2008, pp. 1729–1736.

[13] N. M. Oliver, B. Rosario, and A. P. Pentland, "A bayesian computer vision system for modeling human interactions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 831–843, 2000.