Joint Exploitation of Features and Optical Flow for Real-Time Moving Object Detection on Drones

Hazal Lezki[1,2], I. Ahu Ozturk[1], M. Akif Akpinar[1,4], M. Kerim Yucel[1,3], K. Berker Logoglu[1], Aykut Erdem[3] and Erkut Erdem[3]

[1] STM Defense Technologies and Trade Inc., Ankara, Turkey
{hlezki,iaozturk,makif.akpinar,myucel}@stm.com.tr, berkerlogoglu@gmail.com
[2] Deparment of Electrical and Electronics Engineering, TOBB University of Economics and Technology, Ankara, Turkey
[3] Computer Vision Lab, Department of Computer Engineering, Hacettepe University, Ankara, Turkey
[4] Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
{aykut,erkut}@cs.hacettepe.edu.tr

Abstract. Moving object detection is an imperative task in computer vision, where it is primarily used for surveillance applications. With the increasing availability of low-altitude aerial vehicles, new challenges for moving object detection have surfaced, both for academia and industry. In this paper, we propose a new approach that can detect moving objects efficiently and handle parallax cases. By introducing sparse flow based parallax handling and downscale processing, we push the boundaries of real-time performance with 16 FPS on limited embedded resources (a five-fold improvement over existing baselines), while managing to perform comparably or even improve the state-of-the-art in two different datasets. We also present a roadmap for extending our approach to exploit multi-modal data in order to mitigate the need for parameter tuning.

Keywords: Moving object detection, optical flow, UAV, drones, embedded vision, real-time vision

1 Introduction

Ranging from high-altitude Unmanned Aerial Vehicles (UAV) capable of flying at 65,000 feet [5] to low-altitude miniature drones, long-endurance variants to micro air vehicles weighing just a few grams [6], UAV industry has gone through a meteoric rise. Owing to their ever increasing availability in civilian and military sectors alike, UAV variants have been disruptive in the last decade and

[5] http://www.boeing.com/defense/phantom-eye/
[6] https://aerixdrones.com/products/vidius-the-worlds-smallest-fpv-drone

consequently found use in several applications, such as disaster relief, precision agriculture, cinematography, cargo delivery, industrial inspection, mapping, military surveillance and air support [1].

Following the industrial attention, academic community also contributed to the transformation of UAVs in various aspects, such as aerodynamics, avionics and various sensory data acquired by said platforms. Slightly different than remote sensing domain, drone-mounted imagery has paved the way for new research in computer vision (CV). There has been a large quantity of studies reported in object detection [2–6], action detection [7], visual object tracking [8–10], object counting [11] and road extraction [12]. In recent years, new datasets [7, 13–17], challenges and dedicated workshops [18, 19] have surfaced to bridge the gap between drone-specific vision problems and their generic versions.

From a practical perspective, low-altitude drones introduce several new problems for CV algorithms. Proneness to sudden platform movements and exposure to environmental conditions arguably affect low-altitude drones in a more pronounced manner compared to their high-altitude counterparts. Moreover, fast-changing operating altitudes and camera viewpoints result into the generation of data with a large diversity, which inherently furthers the complexity of virtually any vision problem. Their small-sized nature also impose severe limits on the availability of computational resources installed on-board, which calls for non-trivial engineering solutions [20, 21].

Moving object detection (MOD), primarily used for surveillance purposes, is a long-standing problem in CV and has been the subject of many studies [22–24]. Due to the presence of platform motion in drone vision, it becomes a notorious problem, where platform motion can easily be confused with moving regions/objects. Several solutions addressing platform motion issue have been reported [25, 26]. Moreover, low-altitude drone cases also suffer from severe motion parallax which causes objects closer to camera move faster than objects further away. Solutions provided for motion parallax issue is considered computationally expensive [27–29, 17], which makes the solutions even harder especially when on-board processing with (near) real-time performance is a hard constraint.

In this paper, we propose a new approach for moving object detection, primarily optimized for embedded resources for on-board functionality. We make two main contributions; first, we show that performing a large portion of our pipeline in lower resolutions significantly improve the runtime performance while keeping our accuracy high. Second, we design the matching part of the parallax handling scheme using a simple sparse-flow based technique which avoids the bottlenecks such as failing to extract features from candidate objects or inferior feature matching. Its sparse nature also contributes to further speed-ups, pushing further to real-time performance on embedded platforms.

The paper is organized as follows. In Section 2, related work in the literature is reviewed. The proposed approach is explained thoroughly in Section 3. Experimental results and their analysis are reported in Section 4. We conclude our work by drawing insights and making future recommendations in Section 5.

2 Related Work

The research community has contributed to moving object detection literature considerably over the last few decades. Earlier studies aimed to solve this problem for static cameras, where background subtraction [22] and temporal differencing [30] based solutions slowly transformed into more sophisticated approaches such as background learning via Mixture of Gaussians, Eigen backgrounds and motion layers [31, 32]. As mobile platforms started to emerge, a new layer of complexity was introduced; ego-motion. The presence of ego-motion renders obsolete the approaches devised for static cameras, as the platform motion is likely to produce quite a few false positives. Moreover, this problem becomes more pronounced when platform motion is sudden.

A simple method to tackle platform-motion induced false positives is to perform image alignment as a preprocessing step. By finding the affine/perspective transformation between two consecutive images, one can warp an image onto another and then perform temporal differencing. Primarily named as "feature-based" methods, such methods depend on accurate image alignment where accurate feature keypoint/descriptor computation is imperative [33]. Another approach to solve ego-motion in such cases can be referred as "motion-based", where motion layers [32] and optical flow [26] techniques are utilized. In cases where planar surface assumption (if any) does not hold, the perspective transformation based warping fails to handle motion parallax induced false positives. Unlike high-altitude scenarios, motion parallax becomes a severe problem in imagery taken from the ground as well as low-altitude UAV imagery. There are studies in the literature using various geometric constraints and flow-based solutions which claim to mitigate the effects of motion parallax [27, 34].

Building on the simple solutions reported above, several high impact studies have been reported in recent years. Based on their previous study [34], in [35] authors propose a new method that is related with the projective structure between consecutive image planes, which is used in conjunction with epipolar constraint. This new constraint is useful to detect the moving objects which move along the same direction with the camera, which is a configuration epipolar constraint misses to detect. Assessed using airborne videos, authors state abrupt motion or medium-level parallax might be detrimental to the efficacy of their algorithm. Authors of [36] tackle moving object detection for ground robots, where they use epipolar constraint along with a motion estimation mechanism to handle degenerate cases (camera and platform move to the same direction) in a Bayesian framework. Work reported in [27] handles moving object detection by using epipolar and flow-vector bound constraints, which facilitates parallax handling as well as degenerate cases. Authors estimate the camera pose by using Parallel Tracking and Mapping technique. Similar methods have been reported in [37] and [17], where both algorithms target low altitude imagery but the latter handles parallax in an optimized manner.

In addition to feature based methods mentioned above, motion-based methods have also emerged. In [28], authors fuse the sensory data with imagery to facilitate moving object detection in the presence of ego-motion and motion par-

allax. By using optical flow in conjunction with the epipolar constraint, authors show they can eliminate parallax effects in videos taken from ground vehicles. In work reported in [38], authors use a dense flow based method where optical flow and artificial flow are assessed for their orientation and magnitude to find moving objects in aerial imagery. Another study using flow-based approaches is [39], where authors use optical flow information along with a reduced Singular Value Decomposition and image inpainting stages to handle parallax and ego-motion. They present their results using sequences taken from aerial and ground vehicles. In [40], authors use artificial flow and background subtraction together. They formulate two scores; anomaly and motion scores where the former facilitates good precision and the latter helps achieve improved recall values.

3 Our Approach

In this work, we propose a hybrid moving object detection pipeline which fuses feature based and optical flow based approaches in an efficient manner for near real time performance. In addition, we propose many minor improvements in the pipeline for increasing processing speed as well as detection accuracy. Our proposed pipeline is given in Figure 1. It is based on well studied ego-motion compensation and plane-parallax decomposition approaches [17, 28, 34, 35, 41] and divided into different process lines for ease of understanding.

3.1 Preprocessing and Ego-Motion Compensation

One of the most challenging parts of moving object detection from a drone is to be able to detect varying size of objects from varying altitudes. In a background subtraction and ego-motion compensation based system, such as ours, the easiest way to cope with this variation is to be able to use varying length of time difference between frames that are compared. Thus, as the very first stage of our pipeline, we have implemented a dynamic frame buffer that changes its size according to the height measurements read (when available) from the IMU (Inertial Measurement Unit) as well as the users' desire of detection sensitivity. The size of the buffer, thus the time Δ between frames that will be processed, increases as the required sensitivity to detect smaller objects (and/or smaller movements) increase. In our system, before pushing the frames into our buffer, if the used camera is known and calibration is possible, we correct the lens distortion (radial and tangential) as well.

Typical to the majority of computer vision systems, feature extraction and matching take a significant time of our pipeline and form the bottleneck. Additionally, we claim that calculating the homography between frames in high resolution is not worth the loss in runtime. Therefore, we downscale the input images for feature extraction and matching (using SURF [42]), and then calculate the homographies between frames t, $t - \Delta$ and $t - \Delta$, $t - 2\Delta$. However, to detect smaller objects, the rest of the pipeline runs on original resolution.

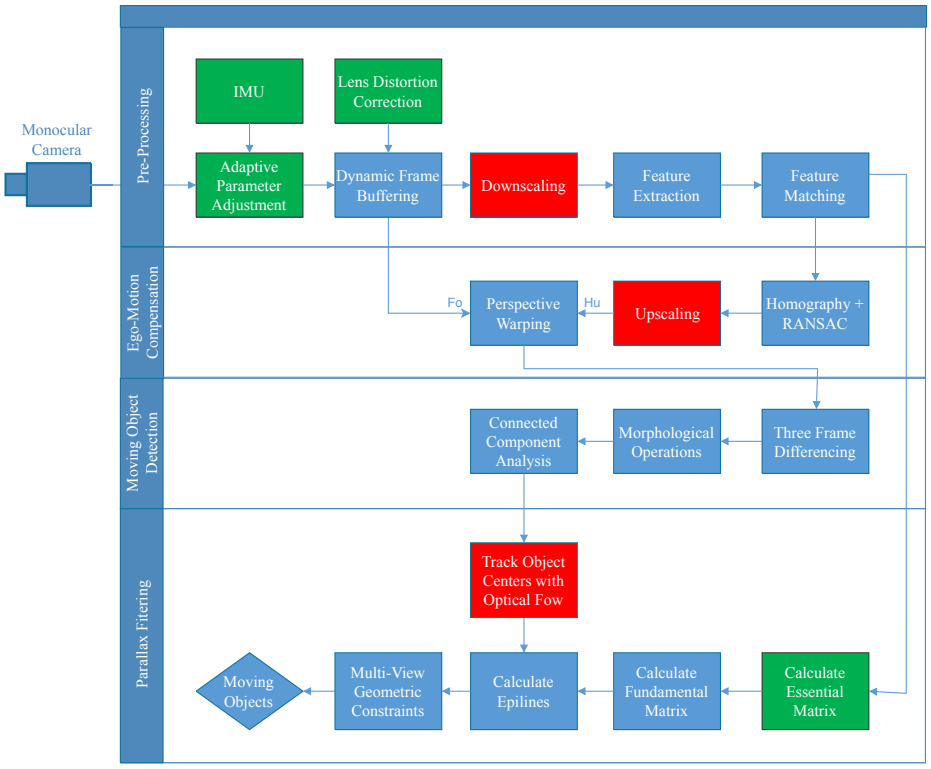


Fig. 1: Our proposed moving object detection pipeline. Red boxes represent the steps we build on other baselines. Green boxes represent steps that can be applied where IMU and camera calibration parameters are available. F_o represents the frame in original resolution and H_u represents upscaled homography.

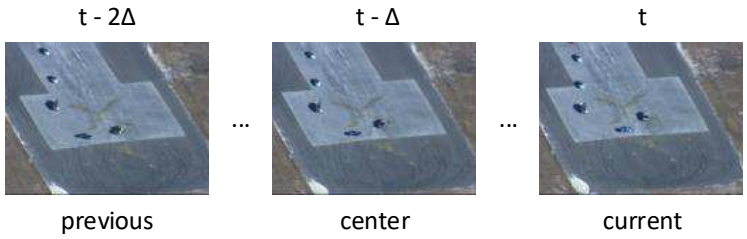


Fig. 2: Dynamic frame buffer. Δ changes depending on required sensitivity.

To achieve this, the homographies calculated in lower resolution H_d are used to calculate/estimate the original resolution homographies H_u using Equation 1.

$$H_u = H_d * P_{do} \tag{1}$$

where P_{do} is the perspective transformation between the downscaled image and original image.

3.2 Moving Object Detection

The calculated upscale homographies (H_u) are used for perspective warping (of original image F_o) and three-frame differencing. As can be seen in Figure 2, *current* and *previous* frames are warped on the *center* frame separately, and two separate two-frame differences are calculated. These two-frame difference results are then processed with an empirical threshold value, which produces a binary image for each. Morphological operations are used to cancel noise and associate pixels belonging to the same object. These two-frame differences (after thresholding and morphological operations) are joined with a logical AND operation to facilitate three-frame differencing. Resulting three-frame difference is then subjected to a connected component analysis to create the object bounding boxes.

3.3 Parallax Filtering

Especially for mini UAVs that operate typically under 150 metres, parallax can be a significant problem. Without a dedicated algorithm, there might be many false positives due to trees, buildings, etc. In the literature, using geometric constraints has proven to be an effective solution for eliminating parallax regions [28, 17, 35, 27]. In these studies, either features that are extracted on candidate moving objects are tracked/matched [17, 27] or each candidate pixel is densely tracked/matched [28, 35] to be able to apply geometric constraints. Instead of these, we propose a fast and efficient hybrid method that only tracks the center locations of the candidate objects using sparse optical flow (via [43]). As can be seen from Table 1, this method facilitates significant performance improvement compared to feature tracking based methods. After tracking only the center locations of the candidate objects, we apply epipolar constraint on tracked locations. As can be seen in Figure 3 and Figure 4, the benefits of tracking only object centers are two fold; epipolar constraint calculations are significantly reduced and the requirement of having keypoints/features on a candidate object is removed.

In order to understand the epipolar constraint [44], assume that $I_{t-\Delta}$ and I_t denote two images of a scene (taken by the same camera at different positions in space) at times $t - \Delta$ and t, and P denote a 3D point in the scene. In addition, let $p_{t-\Delta}$ be the projection of P on $I_{t-\Delta}$, and p_t be the projection of P on I_t.

In light of these, a unique fundamental matrix, represented by $F_t^{t-\Delta}$, that relates images I_t to $I_{t-\Delta}$ can be found, which satisfies

$$p_t^{i\,T} F_t^{t-\Delta} p_{t-\Delta}^i = 0, \tag{2}$$

for all corresponding points $p_{t-\Delta}^i$ and p_t^i where i represents each unique image point. In the case where P is a static point, it satisfies

$$el_t = F^t_{t-\Delta} p^i_{t-\Delta}, \tag{3}$$

$$el_{t-\Delta} = F^{t-\Delta}_t p^i_t \tag{4}$$

where $el_{t-\Delta}$ and el_t are epipolar lines corresponding to p_t and $p_{t-\Delta}$, respectively. If P is a 3D static point, p_t should be located on the epiline el_t (see Figure 5a). Otherwise, P will not satisfy the epipolar constraint (see Figure 5b). One exceptional case can occasionally rise, where the point of interest moves along the epilines themselves. This occurs when the camera and the point of interest move along the same direction (i.e. degenerate case).

If camera information required for camera calibration is available, essential matrix instead of fundamental matrix can be used for more accurate results as follows,

$$F \equiv K^{-T} \widehat{T} R K^{-1} = K^{-T} E K^{-1} \tag{5}$$

where K denotes the camera calibration matrix, $\widehat{T}$ denotes the skew symmetric translation matrix and R denotes the rotation matrix between corresponding frames.

4 Experiments

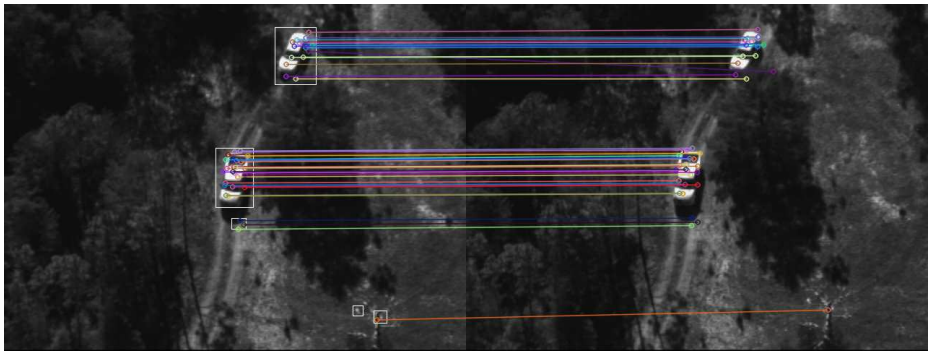
4.1 Datasets

We evaluate our technique in a rigorous manner using two different configurations. In the first one, we use the well-known VIVID [45] dataset. VIVID consists of nine sequences, where three are thermal IR data and the rest are RGB. VIVID annotations are available for every tenth frame and it contains annotations for only one object in the scene. We use a select number of VIVID sequences (egtest01-02-04-05) solely to compare our results with other algorithms. VIVID is the most commonly used dataset for evaluating moving object detection algorithms although it is intended for object tracking. Since VIVID is developed for benchmarking tracking algorithms, only single object (even though multiple moving objects exists) is annotated for each 10th frame.

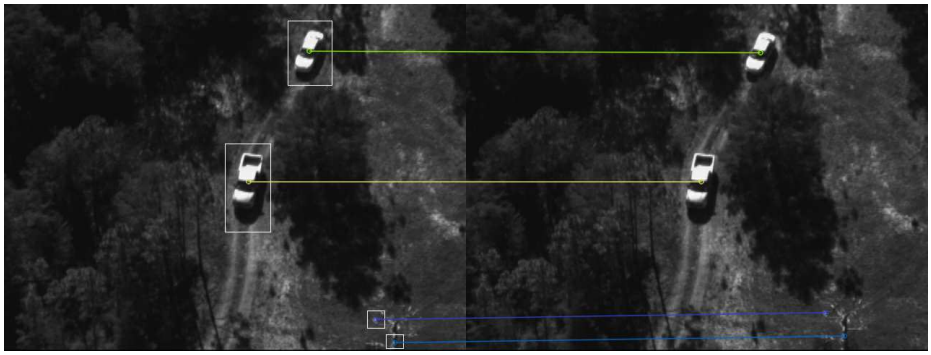
Our second set of evaluation is performed using the publicly available LAMOD dataset [17]. LAMOD consists of various sequences taken from two publicly available datasets, VIVID and UAV123 [16]. These sequences are hand-annotated from scratch for each moving object present in the scene. Annotations are available for each frame and the dataset provides a large set of adverse effects, such as motion parallax, occlusion, out-of-focus and altitude/viewpoint variation [17].

4.2 Results

Execution time. Improvements introduced in run-time performance by our approach is primarily two folds; calculation of the features and homography



(a) Example result for feature tracking on EgTest05.



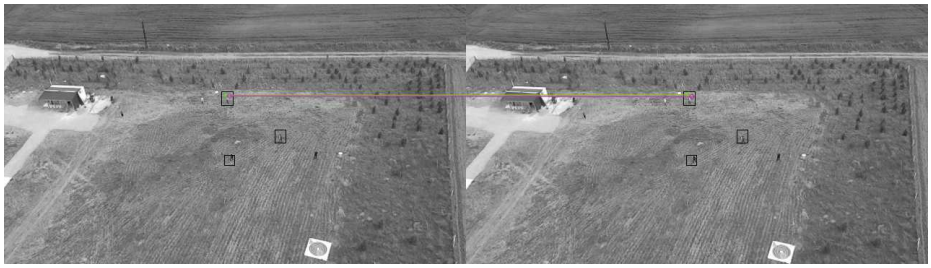
(b) Example result for object center tracking with sparse optical flow on EgTest05.

Fig. 3: Visual comparison of feature tracking and object center tracking with sparse optical flow in EgTest05. Note that there are multiple matches on some of the objects which results on multiple epipolar constraint calculations.

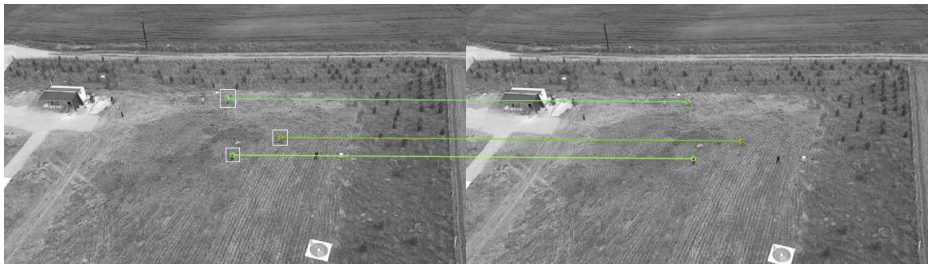
at downscale and sparse optical flow based parallax filtering. We perform our execution time analysis on NVIDIA Jetson TX1 and TX2 modules [7].

As expected, feature extraction in downscaled versions introduce significant speed-ups. We observe that from 1280x720 resolution to 640x360, downscale processing improves runtimes from 148 to 42 ms and 113 to 30 ms for TX1 and TX2, respectively. As downscale processing effectively reduces the number of extracted features, this also reflects on speed of feature matching. Comparing 1280x720 to 640x360 versions, speed of matching improves by the square of input size ratios due to brute-force matching. We see matching speeds change from 146 to 8 ms and 106 to 6 ms (approximately 1700% improvement) for TX1 and TX2, respectively. Sparse optical flow based parallax handling, compared to feature based parallax handling, also introduces considerable execution time gains, as

[7] https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/



(a) Example result for feature tracking on one of our in-house captured videos.

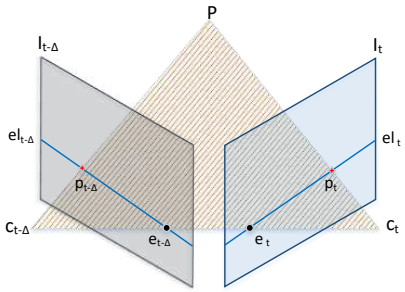


(b) Example result for object center tracking with sparse optical flow on our in-house captured video.

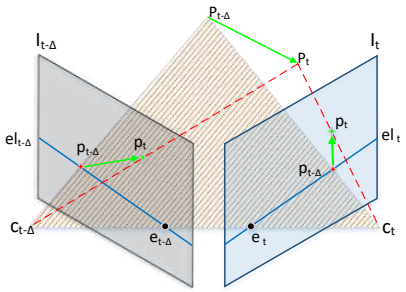
Fig. 4: Visual comparison of feature tracking and object center tracking with sparse optical flow on our in-house captured video. Note that some objects may not have features associated with them, therefore feature tracking (hence parallax handling) may fail. This problem is mitigated by using optical flow.

shown in Table 1. TX1 results show an improvement of 20% to 25% whereas TX2 results show improvements in between 18% to 20%.

Table 2 shows a detailed comparison of a recent technique [17] and our approach. A significant improvement up to 40% is observed for low resolution



(a) Static point



(b) Dynamic point

Fig. 5: Epipolar constraint. Image courtesy of [17].

inputs, both with and without parallax filtering. For larger input resolutions, improvements are in between 200% to 400%.

To support our claim that downscale processing does not lead to significant degradation in accuracy, we also assess our pipeline with full high resolution operation. We present results for original and downscaled operations for LAMOD ground truths in Table 4. Results show a slight decrease in accuracy when compared to high resolution. Except a maximum of 6% decrease in recall for *egtest02*, we do not see any other significant decrease in accuracies. In fact, precision and recall values do not even change in many cases, such as *egtest04* recall and *egtest04* precision values.

Accuracy. We first evaluate our proposed approach using single object ground truths of VIVID dataset to compare our performance with other baseline algorithms. We use precision/recall as our metric and take a minimum of 50% overlap to be a correct detection. As all the baseline algorithms have reported their results in terms of correct detection ratio and miss detection ratio, we convert these results to precision and recall for a better comparison (miss detection ratio is effectively $1 - precision$, whereas correct detection is ratio is equal to precision). We do not report results for parallax handling for sequences *EgTest01* and *EgTest02* as they do not have parallax effects. Results are shown in Table 3.

Our proposed algorithm performs comparably to other baselines, even surpassing them in several sequences; *EgTest01* and *egtest02* results outperform all others in precision, whereas our precision or recall values are the second best in

Table 1: Execution time of our proposed approach for different input resolutions. *Feat.* indicates the version where features are extracted from candidate objects for parallax filtering. *O.F.* indicates the version where objects centres are tracked with sparse optical flow for parallax filtering.

	640 x 480		1280 x 720	
	Feat.	O.F.	Feat.	O.F.
TX1	115 ms	**93 ms**	176 ms	**132 ms**
TX2	76 ms	**62 ms**	108 ms	**77 ms**

Table 2: Execution time of our proposed approach for different input resolutions. *NF* represents no parallax filtering, *PF* represents parallax filtering and *ours* refer to our proposed approach.

		640 x 480		1280 x 720	
		[17]	Ours	[17]	Ours
TX1	NF	115 ms	**70 ms**	350 ms	**102 ms**
	PF	175 ms	**93 ms**	450 ms	**132 ms**
TX2	NF	85 ms	**52 ms**	250 ms	**64 ms**
	PF	140 ms	**62 ms**	350 ms	**77 ms**

Table 3: Precision and recall values for 4 sequences in VIVID dataset with the original single object tracking ground truth. We extrapolate the results of baselines as they do not provide numerical results directly. *NF* and *PF* represent results without and with parallax filtering. Results in each row are precision and recall (in percentage), respectively.

	EgTest01	EgTest02	EgTest04	EgTest05
GMAC [46]	97.0 / 84.0	93.0 / 67.0	83.0 / 75.0	**87.0** / 77.0
MIL [47]	92.0 / 79.0	98.0 / **76.0**	33.0 / 20.0	35.0 / 16.0
OAB1 [48]	88.0 / 77.0	87.0 / 68.0	42.0 / 28.0	35.0 / 18.0
Castelli et al. [38]	86.0 / 84.0	—	**93.0** / **90.0**	85.0 / **82.0**
Ours (NF)	**99.4** / **97.2**	**98.6** / 56.9	76.0 / 77.0	73.0 / 80.0
Ours (PF)	—	—	78.0 / 62.0	83.0 / 66.0

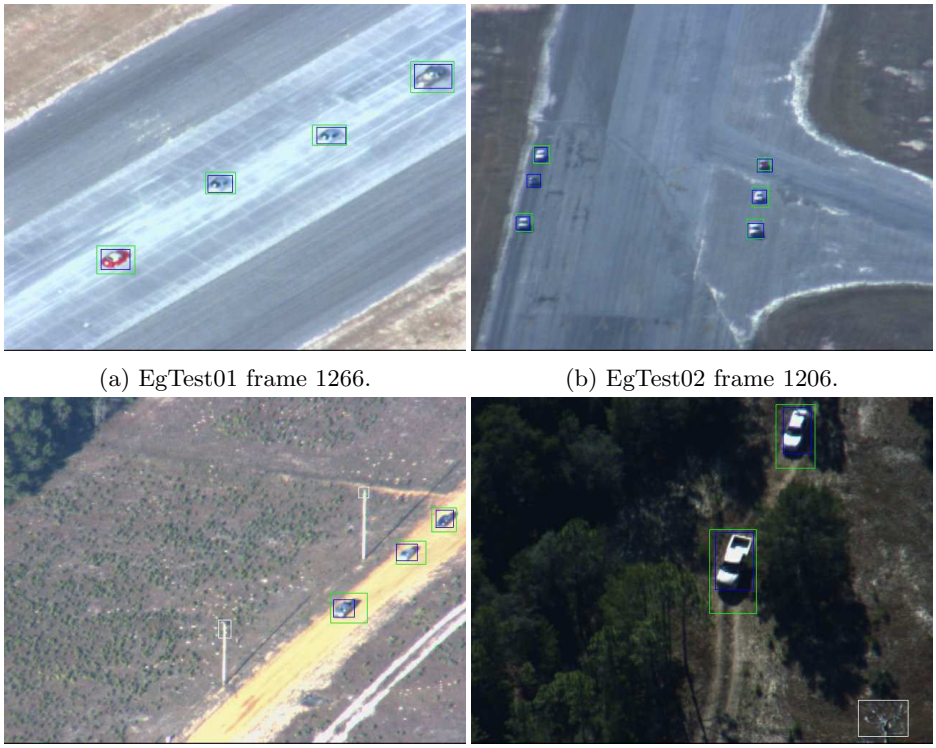
Table 4: Precision and recall values for 4 sequences in VIVID dataset with multi object moving object detection ground truth provided in LAMOD dataset. *NF* and *PF* represent results without and with parallax filtering. Results in each row are precision and recall (in percentage), respectively. Results indicated with $*$ calculate precision/recall for each frame and then average for entire sequence. Results indicated with † represent the results of our technique when it operates on original resolution images (no downscaling).

	EgTest01	EgTest02	EgTest04	EgTest05
Logoglu et al. [17] $*$	93.0 / 82.0	85.0 / 53.0	72.0 / 72.0	71.0 / **68.0**
Ours (NF) $*$	**97.4** / **93.0**	**92.4** / **61.0**	86.0 / **75.0**	70.0 / 66.0
Ours (PF) $*$	—	—	**91.0** / 60.0	**77.0** / 55.0
Ours (NF) †	**96.8** / **92.2**	**92.6** / **59.5**	85.0 / **72.0**	66.0 / **63.0**
Ours (PF) †	—	—	**89.0** / 57.0	**71.0** / 52.0
Ours (NF)	96.7 / 91.2	92.2 / 53.2	86.0 / 69.0	66.0 / 62.0
Ours (PF)	—	—	85.0 / 57.0	70.0 / 50.0

other sequences. Our method shines as it has close precision and recall values. When we perform parallax handling, an expected reduction in recall is compensated with an increase in precision, practically evening out or improving the final F-score. It must be noted that nearly all baselines are effectively object trackers, which means our algorithm performs quite accurately as we do not support our detection with a sophisticated tracker.

We then assess our pipeline for multiple moving objects using LAMOD dataset. We use precision/recall and per-frame precision/recall [8] (i.e. where precision and recall is calculated for every frame and then averaged) as our evaluation metric where 50% overlap is considered a detection. Similar to previous section of our evaluation, we do not report parallax filtering results for *EgTest01* and *EgTest02*. Exemplary results are visualized in Figure 6. Results are shown in Table 4.

[8] Authors of [17] have reported their results with this metric, therefore we give these results to compare our work.



(a) EgTest01 frame 1266.

(b) EgTest02 frame 1206.

(c) EgTest03 frame 285.

(d) EgTest04 frame 1035.

Fig. 6: Detection results on 4 sequences of VIVID dataset. Green boxes are detection results, blue boxes are ground truth data that are taken from LAMOD dataset, grey boxes are candidate objects that are filtered by our parallax filtering algorithm.

Results indicate our proposed algorithm significantly outperforms an existing baseline [17] in all sequences except *EgTest05*. Parallax filtering introduces considerable gains in precision and modest reductions in recall, as reported before. This is expected as *EgTest04* and *EgTest05* have degenerate cases (i.e. objects and the platform move along the same direction) and our approach currently does not handle such cases. This leads to the elimination of true positives by parallax filtering, thus the reduction in recall.

4.3 Multi-Modal Extension

In previous section, as we use public datasets where no IMU or camera information is available, we can not fully utilise the adaptive algorithm we show in Figure 1. This means we can not use lens distortion correction at all and we can only use a fixed set of parameters (i.e. dynamic buffer size) for all sequences.

In order to show how our pipeline works while utilising external sensory data, we present some qualitative results with our in-house captured videos, where we were able to acquire the relevant IMU and camera parameter information.

Lens Distortion Correction. Lens distortion distorts certain pixels to other locations, radially or tangentially in our case, which directly effects our results (see Figure 7 b)). This occurs as pixels are distorted to some other location and during image registration, they are erroneously detected as moving objects. By using radial and tangential coefficients specific to the camera lens, this effect can be corrected. Such correction leads to visible improvements in our performance (see Figure 7 d)).

Dynamic Frame Buffer. It can be hard to detect slowly moving objects in high altitudes as their relative displacement in the image is not large. This can be alleviated by using the height measurements provided by IMU; we dynamically change the size of the buffer (namely the distance between the frames to be differenced) linearly with the altitude. By doing so, we effectively amplify the perceived movement of slow moving objects, thus making them highly detectable. Exemplary results shown in Figure 8 c) and d) verify the said phenomena and shows a visible improvement in recall.

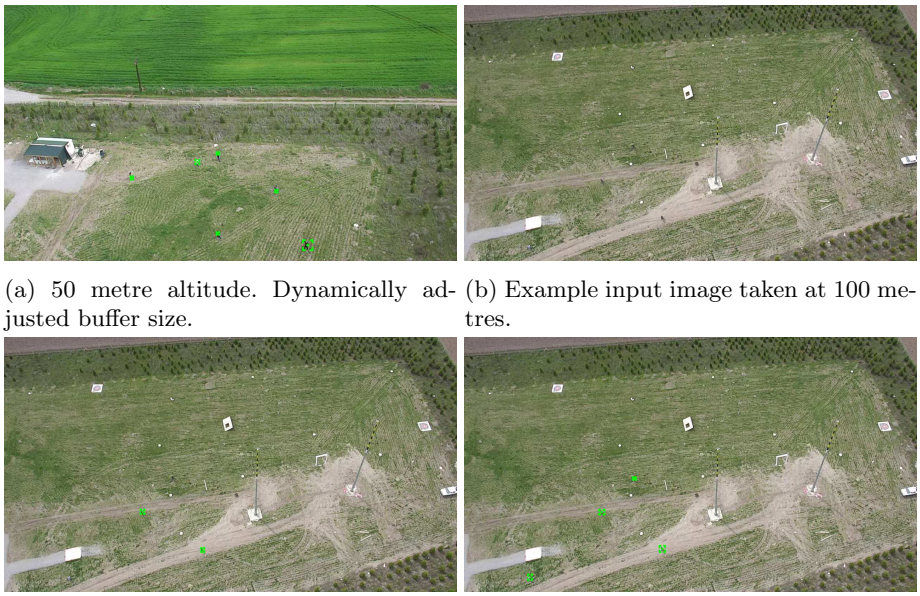
(a) Thresholded three-frame difference with no lens distortion correction.

(b) Detected moving objects without lens distortion correction.

(c) Thresholded three-frame difference with no lens distortion correction.

(d) Detected moving objects with lens distortion correction.

Fig. 7: The effect of lens distortion correction. Note that although the effects of lens correction on input images may be almost imperceptible, it gives rise to many pixel level errors.



(a) 50 metre altitude. Dynamically adjusted buffer size.

(b) Example input image taken at 100 metres.

(c) Image b) processed with a static buffer size parameter used in subfigure a).

(d) Image b) processed with dynamically adjusted buffer size.

Fig. 8: The effect of dynamic frame buffering. Note that dynamically adjusted buffer size for 50 metre altitude works accurately for 50 metres, but fails at 100 metre altitude. Adaptively changing the buffer size for 100 metres significantly improves our detection performance.

5 Conclusions

In this paper, we propose a new approach aimed at tackling moving object detection problem for imagery taken from low-altitude aerial platforms. Capable of handling the motion of the platform as well as the detrimental effects of motion parallax, our approach performs parallax handling by sparse optical flow based tracking along with epipolar constraint and performs a large portion of the pipeline in lower resolutions. These two changes introduce significant runtime improvements, reaching up to 16 FPS on embedded resources. Moreover, we analyze our approach in two different datasets for single and multiple moving object detection tasks. We observe that our results perform either comparably or better than existing state-of-the-art algorithms. We also outline an advanced pipeline capable of exploiting multi-modal data that might alleviate the need of laborious parameter tuning. As future work, we aim to integrate a light-weight scheme to alleviate the effect of degenerate motion cases. Should a dataset with IMU and camera information become publicly available, we aim to assess our approach in a multi-modal setting.

References

1. Clarke, R.: Understanding the drone epidemic. Computer Law & Security Review **30**(3) (2014) 230–246
2. Zhong, J., Lei, T., Yao, G.: Robust vehicle detection in aerial images based on cascaded convolutional neural networks. Sensors **17**(12) (2017) 2720
3. Li, F., Li, S., Zhu, C., Lan, X., Chang, H.: Cost-effective class-imbalance aware cnn for vehicle localization and categorization in high resolution aerial images. Remote Sensing **9**(5) (2017) 494
4. Tijtgat, N., Van Ranst, W., Volckaert, B., Goedemé, T., De Turck, F.: Embedded real-time object detection for a uav warning system. In: ICCV2017, the International Conference on Computer Vision. (2017) 2110–2118
5. Sommer, L.W., Schuchert, T., Beyerer, J.: Fast deep vehicle detection in aerial images. In: Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, IEEE (2017) 311–319
6. Stek, T.D.: Drones over mediterranean landscapes. the potential of small uav's (drones) for site detection and heritage management in archaeological survey projects: A case study from le pianelle in the tappino valley, molise (italy). Journal of Cultural Heritage **22** (2016) 1066–1071
7. Barekatain, M., Martí, M., Shih, H.F., Murray, S., Nakayama, K., Matsuo, Y., Prendinger, H.: Okutama-action: An aerial view video dataset for concurrent human action detection. In: 1st Joint BMTT-PETS Workshop on Tracking and Surveillance, CVPR. (2017) 1–8
8. Pestana, J., Sanchez-Lopez, J.L., Campoy, P., Saripalli, S.: Vision based gps-denied object tracking and following for unmanned aerial vehicles. In: Safety, security, and rescue robotics (ssrr), 2013 ieee international symposium on, IEEE (2013) 1–6
9. Dang, C.T., Pham, T.B., Truong, N.V., et al.: Vision based ground object tracking using ar drone quadrotor. In: Control, Automation and Information Sciences (ICCAIS), 2013 International Conference on, IEEE (2013) 146–151
10. Chen, P., Dang, Y., Liang, R., Zhu, W., He, X.: Real-time object tracking on a drone with multi-inertial sensing data. IEEE Transactions on Intelligent Transportation Systems **19**(1) (2018) 131–139
11. Hsieh, M.R., Lin, Y.L., Hsu, W.H.: Drone-based object counting by spatially regularized regional proposal network. In: The IEEE International Conference on Computer Vision (ICCV). Volume 1. (2017)
12. Kanistras, K., Martins, G., Rutherford, M.J., Valavanis, K.P.: A survey of unmanned aerial vehicles (uavs) for traffic monitoring. In: Unmanned Aircraft Systems (ICUAS), 2013 International Conference on, IEEE (2013) 221–234
13. Du, D., Qi, Y., Yu, H., Yang, Y., Duan, K., Li, G., Zhang, W., Huang, Q., Tian, Q.: The unmanned aerial vehicle benchmark: Object detection and tracking. arXiv preprint arXiv:1804.00518 (2018)
14. Wang, S., Bai, M., Mattyus, G., Chu, H., Luo, W., Yang, B., Liang, J., Cheverie, J., Fidler, S., Urtasun, R.: Torontocity: Seeing the world with a million eyes. In: Computer Vision (ICCV), 2017 IEEE International Conference on, IEEE (2017) 3028–3036
15. Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: Dota: A large-scale dataset for object detection in aerial images. In: Proc. CVPR. (2018)
16. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for uav tracking. In: European conference on computer vision, Springer (2016) 445–461

17. Berker Logoglu, K., Lezki, H., Kerim Yucel, M., Ozturk, A., Kucukkomurler, A., Karagoz, B., Erdem, E., Erdem, A.: Feature-based efficient moving object detection for low-altitude aerial platforms. In: The IEEE International Conference on Computer Vision (ICCV) Workshops. (Oct 2017)
18. Lam, D., Kuzma, R., McGee, K., Dooley, S., Laielli, M., Klaric, M., Bulatov, Y., McCord, B.: Xview: Objects in context in overhead imagery. arXiv preprint arXiv:1802.07856 (2018)
19. Zhu, P., Wen, L., Bian, X., Ling, H., Hu, Q.: Vision meets drones: A challenge. arXiv preprint arXiv:1804.07437 (2018)
20. Yu, Q., Medioni, G.: A gpu-based implementation of motion detection from a moving platform. (2008)
21. Kryjak, T., Komorkiewicz, M., Gorgon, M.: Real-time moving object detection for video surveillance system in fpga. In: Design and Architectures for Signal and Image Processing (DASIP), 2011 Conference on, IEEE (2011) 1–8
22. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proceedings of the IEEE **90**(7) (2002) 1151–1163
23. Eveland, C., Konolige, K., Bolles, R.C.: Background modeling for segmentation of video-rate stereo sequences. In: Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, IEEE (1998) 266–271
24. Zhou, X., Yang, C., Yu, W.: Moving object detection by detecting contiguous outliers in the low-rank representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(3) (2013) 597–610
25. Suganuma, N., Kubo, T.: Fast dynamic object extraction using stereovision based on occupancy grid maps and optical flow. In: Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on, IEEE (2011) 978–983
26. Rodríguez-Canosa, G.R., Thomas, S., Del Cerro, J., Barrientos, A., MacDonald, B.: A real-time method to detect and track moving objects (datmo) from unmanned aerial vehicles (uavs) using a single camera. Remote Sensing **4**(4) (2012) 1090–1111
27. Kimura, M., Shibasaki, R., Shao, X., Nagai, M.: Automatic extraction of moving objects from uav-borne monocular images using multi-view geometric constraints. In: IMAV 2014: International Micro Air Vehicle Conference and Competition 2014, Delft, The Netherlands, August 12-15, 2014, Delft University of Technology (2014)
28. Salgian, G., Bergen, J., Samarasekera, S., Kumar, R.: Moving target indication from a moving camera in the presence of strong parallax. Technical report, DTIC Document (2006)
29. Dey, S., Reilly, V., Saleemi, I., Shah, M.: Detection of independently moving objects in non-planar scenes via multi-frame monocular epipolar constraint. In: European Conference on Computer Vision, Springer (2012) 860–873
30. Paragios, N., Deriche, R.: Geodesic active contours and level sets for the detection and tracking of moving objects. IEEE Transactions on pattern analysis and machine intelligence **22**(3) (2000) 266–280
31. Joshi, K.A., Thakore, D.G.: A survey on moving object detection and tracking in video surveillance system. International Journal of Soft Computing and Engineering **2**(3) (2012) 44–48
32. Cao, X., Lan, J., Yan, P., Li, X.: Vehicle detection and tracking in airborne videos by multi-motion layer analysis. Machine Vision and Applications **23**(5) (2012) 921–935
33. Irani, M., Anandan, P.: A unified approach to moving object detection in 2d and 3d scenes. IEEE transactions on pattern analysis and machine intelligence **20**(6) (1998) 577–589

34. Kang, J., Cohen, I., Medioni, G., Yuan, C.: Detection and tracking of moving objects from a moving platform in presence of strong parallax. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. Volume 1., IEEE (2005) 10–17
35. Yuan, C., Medioni, G., Kang, J., Cohen, I.: Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(9) (Sept 2007) 1627–1641
36. Kundu, A., Krishna, K.M., Sivaswamy, J.: Moving object detection by multi-view geometric techniques from a single camera mounted robot. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. (Oct 2009) 4306–4312
37. Minaeian, S., Liu, J., Son, Y.J.: Effective and efficient detection of moving targets from a uavs camera. IEEE Transactions on Intelligent Transportation Systems **19** (2018) 497–506
38. Castelli, T., Trémeau, A., Konik, H., Dinet, E.: Moving object detection for unconstrained low-altitude aerial videos, a pose-independant detector based on artificial flow. In: Image and Signal Processing and Analysis (ISPA), 2015 9th International Symposium on, IEEE (2015) 42–47
39. Wu, Y., He, X., Nguyen, T.Q.: Moving object detection with a freely moving camera via background motion subtraction. IEEE Transactions on Circuits and Systems for Video Technology **27**(2) (Feb 2017) 236–248
40. Makino, K., Shibata, T., Yachida, S., Ogawa, T., Takahashi, K.: Moving-object detection method for moving cameras by merging background subtraction and optical flow methods. In: 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP). (Nov 2017) 383–387
41. Ali, S., Shah, M.: Cocoa: tracking in aerial imagery. In: Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications III. Volume 6209., International Society for Optics and Photonics (2006) 62090D
42. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European conference on computer vision, Springer (2006) 404–417
43. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision. (1981)
44. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
45. Collins, R., Zhou, X., Teh, S.K.: An open source tracking testbed and evaluation web site. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005). Volume 2. (2005) 35
46. Hasan, M.: Integrating geometric, motion and appearance constraints for robust tracking in aerial videos. (2013)
47. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 983–990
48. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: British Machine Vision Conference. Volume 1. (2006) 6