ATG-PVD: Ticketing Parking Violations on a Drone

Hengli Wang[1], Yuxuan Liu[1], Huaiyang Huang[1], Yuheng Pan[2], Wenbin Yu[2], Jialin Jiang[2], Dianbin Lyu[2], Mohammud J. Bocus[3], Ming Liu[1], Ioannis Pitas[4], and Rui Fan[2,5](✉)

[1] HKUST Robotics Institute, Hong Kong, China
{hwangdf,yliuhb,hhuangat,eelium}@ust.hk
[2] ATG Robotics, Hangzhou, China
{panyuheng,yuwenbin,jiangjialin,lvdianbin}@atg-itech.com
[3] University of Bristol, Bristol, UK
junaid.bocus@bristol.ac.uk
[4] Aristotle University of Thessaloniki, Thessaloniki, Greece
pitas@csd.auth.gr
[5] UC San Diego, La Jolla, USA
rui.fan@ieee.org

Abstract. In this paper, we introduce a novel suspect-and-investigate framework, which can be easily embedded in a drone for automated parking violation detection (PVD). Our proposed framework consists of: 1) SwiftFlow, an efficient and accurate convolutional neural network (CNN) for unsupervised optical flow estimation; 2) Flow-RCNN, a flow-guided CNN for car detection and classification; and 3) an illegally parked car (IPC) candidate investigation module developed based on visual SLAM. The proposed framework was successfully embedded in a drone from ATG Robotics. The experimental results demonstrate that, firstly, our proposed SwiftFlow outperforms all other state-of-the-art unsupervised optical flow estimation approaches in terms of both speed and accuracy; secondly, IPC candidates can be effectively and efficiently detected by our proposed Flow-RCNN, with a better performance than our baseline network, Faster-RCNN; finally, the actual IPCs can be successfully verified by our investigation module after drone re-localization.

Dataset and Demo Video:
sites.google.com/view/atg-pvd

1 Introduction

We are currently experiencing an unprecedented crisis due to the ongoing Coronavirus Disease 2019 (COVID-19) pandemic. Its worldwide escalation has taken us by surprise, causing major disruptions to global health, economic and social

H. Wang, Y. Liu, H. Huang, Y. Pan—Equal contributions.

A. Bartoli and A. Fusiello (Eds.): ECCV 2020 Workshops, LNCS 12538, pp. 1–17, 2020.
https://doi.org/10.1007/978-3-030-66823-5_32

systems. Indeed, our lives have changed overnight – businesses and schools are closed, most employees are working from home, and many have found themselves without a job. Millions of people across the globe are confined to their homes, while healthcare workers are at the frontline of the COVID-19 response [1]. With the increase in COVID-19 cases, public transport use has plummeted, as commuters shun buses, trams, and trains in favor of private cars and taxis. For instance, USA Today reported that the transit ridership demand in April 2020 was down by about 75% nationwide, compared to normal, with figures of 85% in San Francisco, 67% in Detroit and 60% in Philadelphia [2].

With the increasing number of vehicles on the roads, parking spaces have become scarce and many vehicles are parked just by the roadside, which in turn results in a significant increase in parking violations. In late March 2020, the Department of Transportation in Los Angeles [3] announced relaxed parking enforcement regulations as part of the emergency response to COVID-19, so that their citizens could practice safe social distancing without being concerned about a ticket. As the Return-to-Work Plan progresses, the relaxed parking enforcement regulations are no longer in force, consequently increasing the workload of the local traffic law enforcement officers. The demand for automated and intelligent parking violation detection (PVD) systems has thus become greater than ever.

The existing automated PVD systems typically recognize illegally parked cars (IPCs) by analyzing the videos acquired by closed-circuit televisions (CCTVs) through 2D/3D object detection algorithms [4] or video surveillance analysis algorithms [5]. However, the efficiency of such methods relies on CCTV camera positions, as IPCs cannot always be detected, especially if they are at a distant location. Deploying more CCTVs can definitely minimize misdetections, but this will also incur a high cost, and/or may not be practical. Therefore, many researchers have turned their focus towards mobile PVD systems, which can be mounted on any vehicle type. For example, the Birmingham City Council in England utilizes surveillance cars to detect IPCs and record their plate numbers [6]. However, such surveillance cars are expensive and typically require drivers. Therefore, autonomous machines, especially drones, have emerged as more efficient and cheaper alternatives.

The cars in the street can be grouped into three categories: 1) moving cars (MCs), 2) legally parked cars (LPCs) and 3) IPCs. MCs can be distinguished from LPCs and IPCs using dynamic object detection techniques, such as optical flow analysis, while IPCs can be distinguished from LPCs using object detection networks, such as Faster-RCNN [7], with the assistance of parking spot information. In this paper, we introduce a novel *suspect-and-investigate* PVD system (see Fig. 1) embedded in a drone. In the suspicion phase, we first employ a novel unsupervised optical flow estimation network, referred to as *SwiftFlow*, to estimate the optical flow F_t between I_t and I_{t+1}. F_t is then incorporated into a novel object detection and classification network, referred to as *Flow-RCNN*, to detect cars and classify them into MCs, LPCs and IPC candidates. A visual simultaneous localization and mapping (VSLAM) module then builds a localizable map

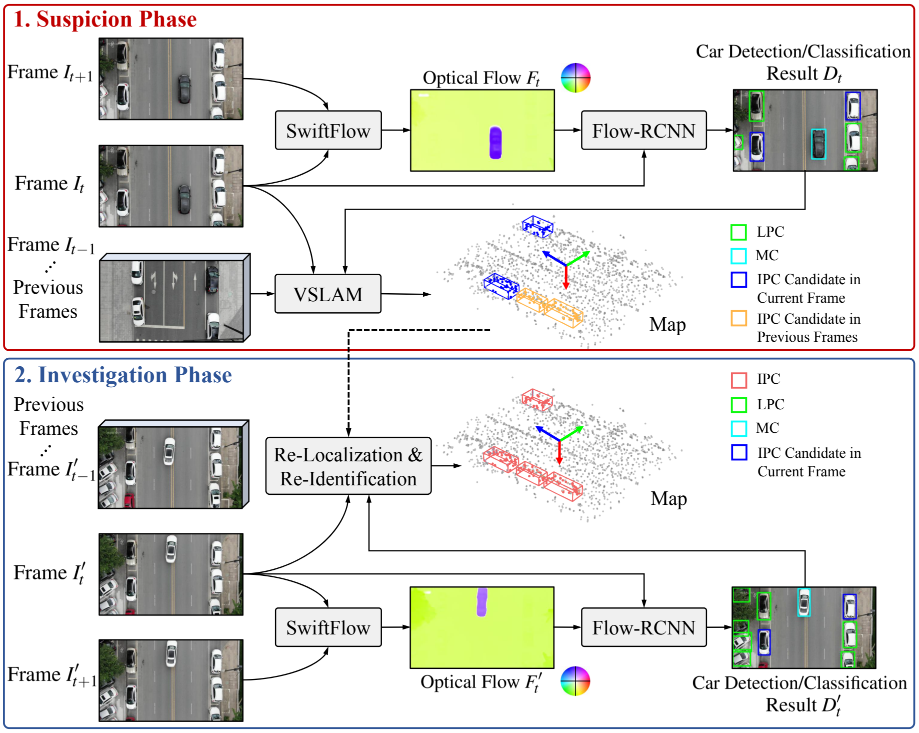


Fig. 1. The framework of our proposed suspect-and-investigate PVD system: the first phase identifies suspected IPC candidates, and the second phase investigates the suspected IPC candidates and issues tickets to the actual IPCs. The frame I_t in the suspicion phase corresponds to the frame I'_t in the investigation phase.

containing the suspected IPC candidates. After a parking grace period (which is typically five minutes) has elapsed, the drone flies back to the same location. The VSLAM module in the investigation phase subsequently detects loop closure and re-localizes the drone in the pre-built map. Finally, the suspected IPC candidates are re-identified, and the actual IPCs are marked in the map. Our main contributions are summarized as follows:

– A novel suspect-and-investigate PVD framework;
– SwiftFlow, a novel unsupervised optical flow estimation network;
– Flow-RCNN, a novel car detection and classification network;
– A large-scale PVD dataset, published for research purposes.

2 Related Work

2.1 Optical Flow Estimation

Traditional approaches generally formulate optical flow estimation as a global energy minimization problem [8–11]. Recently, convolutional neural networks

(CNNs) have achieved impressive performance in optical flow estimation. FlowNet [12] was the pioneering work in end-to-end deep optical flow estimation. Its key component is a so-called correlation layer, which can provide explicit matching capabilities. Later methods, PWC-Net [13] and LiteFlowNet [14] introduced the popular coarse-to-fine architecture, which provides a good trade-off between optical flow accuracy and computation efficiency. Meanwhile, IRR-PWCNet [15] demonstrates that occlusion prediction integrated into optical flow estimation can effectively enhance the optical flow estimation accuracy.

Although the aforementioned supervised optical flow estimation methods perform impressively, they generally require a large amount of optical flow ground truth to learn the best solution. Acquiring such ground truth, especially for real-world datasets, is extremely time-consuming and labor-intensive, making these supervised approaches difficult to apply in real-world applications. For these reasons, unsupervised learning has recently become the preferred technique for such applications. For instance, DSTFlow [16] employs a photometric loss and a smooth loss in CNN training, which are similar to the global energy used in traditional methods. Additionally, some methods, such as UnFlow [17], DDFlow [18] and SelFlow [19] integrate occlusion reasoning into unsupervised optical flow estimation frameworks to further improve their accuracy. However, such approaches are typically computationally intensive, and they are difficult to embed in a drone.

2.2 Object Detection

Discovering objects and their locations in images is still a challenging problem in computer vision. Due to their promising results, CNNs have emerged as a powerful tool for object detection. The modern deep object detection algorithms can be grouped into two main types: a) anchor-based and b) anchor-free.

Anchor-based methods predict bounding boxes based on initial guesses. According to the pipelines and primary proposal sources, they can be further categorized as either one-stage or two-stage methods. The former make predictions directly from hand-crafted anchors. For example, RetinaNet [20] employs a feature pyramid network (FPN) to produce dense predictions at multiple scales. On the other hand, the two-stage methods make predictions using the proposals produced by a one-stage detector. For instance, Fast-RCNN [21] and Faster-RCNN [7] perform cropping and resizing on images or feature maps, according to the bounding box proposals. The RCNN branch in Faster-RCNN utilizes a field of view (FOV), that is larger than the bounding box proposals, so as to extract regions of interest (RoIs) directly from the feature maps.

Anchor-free methods usually do not rely on human-designed region proposals to bootstrap the detection process. For example, CornerNet [22] translates the object detection problem into a keypoint detection and matching problem, where specially-designed pooling layers construct biased receptive fields for corner point detection. CenterNet [23], which is based on CornerNet [22], utilizes two customized modules: a) cascade corner pooling and b) center pooling, to enrich information collected by both the top-left and bottom-right corners. It detects each object as a triplet, rather than a pair, of keypoints.

In recent years, incorporating additional visual information, such as semantic predictions, into object classification is becoming an increasingly ubiquitous part of object detection. Since MCs can be easily distinguished from optical flow images, we incorporate the latter into our framework to improve IPC candidate detection.

2.3 VSLAM

Traditional VSLAM approaches leverage visual features and the geometric relations between multiple views of a 3D scene (typically known as multi-view geometry) to estimate camera poses and construct/update a map of the 3D scene. The state-of-the-art VSLAM approaches are classified as either indirect [24–26] or direct [27–29]. Both types extract visual features from images and associate them with descriptors. However, the indirect methods sample corners and associate them with higher dimensional descriptors, while the direct methods typically sample pixels with a relatively large local intensity gradient and associate them with a patch of pixels surrounding their sampled location. Furthermore, these two types of methods typically minimize different objective functions: the indirect methods resort to geometric residuals, whereas the direct methods resort to photometric residuals.

In order to combine the advantages of these two types of methods, Froster *et al.* [30] proposed semi-direct visual odometry (SVO), which tracks camera poses via sparse image alignment and utilizes hierarchical bundle adjustment (BA) as the back-end to optimize the geometry structure and camera motion. Furthermore, many researchers have integrated other computer vision tasks, such as 2D object detection [31–33], instance segmentation [34,35] and flow/depth prediction [36,37], into their SLAM systems, so as to address the problem of the existence of dynamic objects, by exploiting high-level semantic information. For example, Huang *et al.* [32] proposed ClusterVO, which uses a multi-level probabilistic association scheme to both track low-level visual features and realize high-level object detection. Moreover, Yang *et al.* [31] introduced CubeSLAM, which performs single image 3D cuboid object detection, together with multi-view object SLAM.

3 ATG-PVD Framework

3.1 SwiftFlow

Since our proposed SwiftFlow network is based on the pipeline of PWC-Net [13], we first provide readers with some preliminaries about the latter. In PWC-Net [13], feature maps are first extracted from video frames using a Siamese pyramid network. Then, the feature map x_{t+1}^l of the $(t+1)$-th video frame at level l is aligned with the feature map x_t^l of the t-th video frame at level l

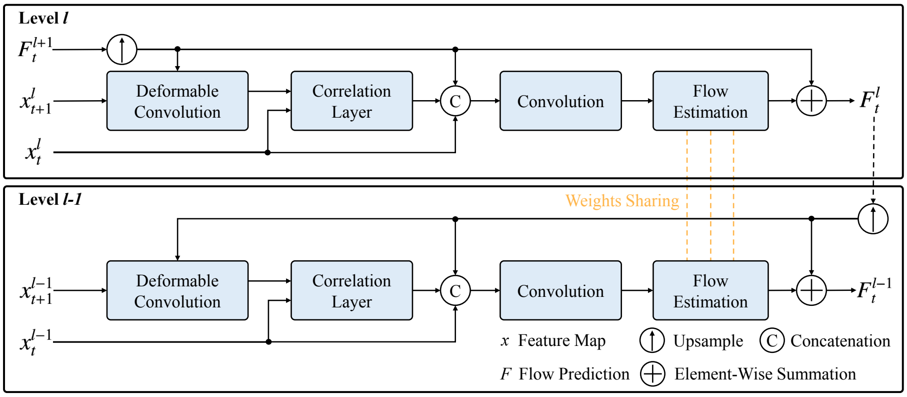


Fig. 2. The decoder architecture of our proposed SwiftFlow. The pipeline of two adjacent levels in the decoder are displayed for simplicity.

via a warping operation based on the upsampled flow prediction F_t^{l+1} at level $l+1$. A correlation layer is then employed to compute the cost volume, which is subsequently concatenated with x_t^l as well as the upsampled flow prediction F_t^{l+1} at level $l+1$. Finally, the flow residual, predicted by the flow estimation module, is combined with the upsampled flow prediction F_t^{l+1} at level $l+1$ using an element-wise summation to generate the flow prediction F_t^l at level l. We iterate this process and obtain the flow predictions at different scales.

SwiftFlow improves on PWC-Net [13] in terms of computational efficiency, so that it can perform in real time on a drone. The decoder in PWC-Net [13] has too many parameters, so we make three major modifications to the decoder architecture (see Fig. 2) to minimize the model size and improve accuracy. As the decoder in PWC-Net [13] employs a dense connection scheme in each pyramid level, making the network computationally intensive, SwiftFlow establishes connections only between two adjacent levels, which can reduce the number of network parameters by 50%. Furthermore, the optical flow estimation modules at different pyramid levels of PWC-Net [13] have different learnable weights to estimate optical flow residuals. Considering that the optical flow estimation modules at different levels have the same functionality and the optical flow residuals at different levels have similar value ranges, we believe sharing the weights of optical flow estimation modules at all pyramid levels can be a more effective and efficient strategy. We also add an additional convolutional layer before the optical flow estimation module at each level for feature map alignment. Moreover, we notice that the warping operation can induce ambiguity to occluded areas, which breaks correlation layer symmetry. We propose to add an asymmetric layer before the correlation layer to alleviate this problem and improve optical flow estimation accuracy. Therefore, we replace the warping operation with a deformable convolutional layer [38], as shown in Fig. 2.

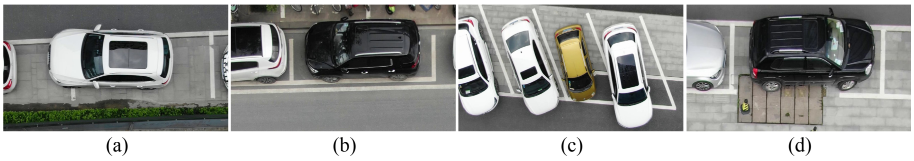


Fig. 3. Challenging cases for parked car detection and classification.

Referring to the commonly applied unsupervised training strategy, we train SwiftFlow by minimizing the following weighted sum of losses:

$$L = \lambda_{\text{photo}} \cdot L_{\text{photo}} + \lambda_{\text{smooth}} \cdot L_{\text{smooth}} + \lambda_{\text{self}} \cdot L_{\text{self}}, \tag{1}$$

where L_{photo} is the photometric loss that considers an occlusion-aware mask [39], L_{smooth} is the smoothness regularization [40], and L_{self} is the self-supervision Charbonnier loss [18]. Following the instructions in [41], we set $\lambda_{\text{photo}} = 1$ and $\lambda_{\text{smooth}} = 2$ in our experiments. Moreover, we use $\lambda_{\text{self}} = 0$ for the first 50% of training steps, and increase it to 0.3 linearly for the next 10% of training steps, after which it stays at a constant value.

3.2 Flow-RCNN

Given an RGB video frame and its corresponding estimated optical flow, the proposed Flow-RCNN detects cars in the video frame and classifies them into MCs, LPCs, and IPC candidates.

Judging whether a car is legally parked is very challenging. Intuitively, we can resort to the parking spot delimitation lines, which are typically painted in white. However, in real-world environments, methods that rely solely on the parking spot information may fail. For instance, in Fig. 3(a), the white car is not parked entirely within the designated parking spot; in Fig. 3(b), only parts of the white car and parking spot appear; and in Fig. 3(c) and Fig. 3(d), the parking spots are not enclosed. Moreover, parking spots are not always bounded by rectangular line markings, as illustrated in Fig. 3(c). It is challenging to design a rule-guided method to solve for these cases, even with perfectly labeled cars and parking spots. Furthermore, various tall objects, such as light poles and trees, often present salient optical flow estimations. In this case, the methods that rely entirely on optical flow information can wrongly characterize an IPC/LPC as an MC. Therefore, an end-to-end, optical flow-guided, and detect-and-classify architecture for IPC candidate detection provides a better alternative.

The architecture of our proposed Flow-RCNN is illustrated in Fig. 4. It incorporates the optical flow information, obtained by SwiftFlow in Sect. 3.1, into the conventional Faster-RCNN [7] architecture for IPC candidate detection, and it outputs the position and category (MC, LPC or IPC candidate) of each car in the video frame in an end-to-end manner. The RGB video frame is first passed through a backbone CNN to produce multi-scale feature maps y^i. The features

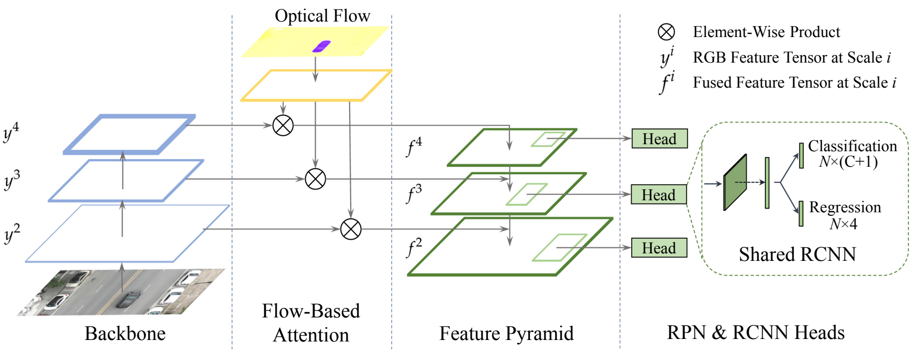


Fig. 4. Flow-RCNN architecture. The optical flow image, obtained from SwiftFlow, is fed into multiple convolutional layers. The optical flow features then dynamically weigh each element in the multi-scale feature maps extracted from the RGB image.

extracted from the optical flow image then dynamically weigh the activation of each element in the multi-scale feature maps y^i, which enables the detector to focus more on MCs. We then fuse the multi-scale feature maps to produce a feature pyramid for the subsequent region proposal network (RPN) and RCNN heads [7]. Since our dataset is highly imbalanced (see Fig. 7), *i.e.*, most vehicles are regarded as IPC candidates or IPCs, we apply focal loss [20] to mitigate the class imbalance problem in the classification stage.

3.3 Mapping, Re-localization and Re-identification

Given RGB images and the corresponding detected IPC candidates, our next target is to build a 3D map, investigate each IPC candidate and mark it in the map. To this end, we develop a mapping, re-localization and re-identification module, as illustrated in Fig. 5, on top of ORB-SLAM2 [42].

Our proposed system applies a suspect-and-investigate scheme to mark IPCs in 3D. In the suspicion phase, we leverage ORB-SLAM2 [42] to build a 3D localizable map and mark the detected IPC candidates in the map. Given an RGB image containing detected IPC candidates, the system first extracts ORB [43] features $\{\mathbf{u}_0, \dots, \mathbf{u}_t\}$ and associates them with 2D bounding boxes $\{\mathcal{B}_0^{2D}, \dots, \mathcal{B}_h^{2D}\}$. We explicitly exclude the ORB features extracted from MCs in the subsequent procedures, *i.e.*, tracking and mapping. The rest of the features are then matched with the 3D keypoints $\{\mathbf{x}_0, \dots, \mathbf{x}_m\}$ in the map. With these 3D-2D correspondences $\mathcal{K} \doteq \{(i_k, j_k)\}_{k=1:N}$, the current camera pose $\mathbf{T} = [\mathbf{R}, \mathbf{t}]$ is estimated in a perspective-n-point (PnP) scheme by minimizing the reprojection error as follows [42]:

$$\mathbf{R}^*, \mathbf{t}^* = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{(i,j)\in\mathcal{K}} \|\mathbf{u}_i - \pi(\mathbf{R}\mathbf{x}_j - \mathbf{t})\|, \tag{2}$$

where $\pi(\cdot)$ is the camera projection function. After solving the camera pose, the inlier correspondences $\mathcal{K}^* \doteq \{(i_k, j_k)\}_{k=1:N'}$ can be determined via their

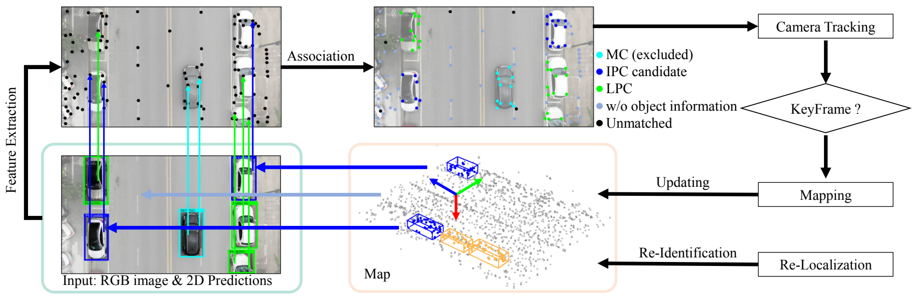


Fig. 5. The pipeline of our mapping, re-localization and re-identification module.

reprojection errors. Then we attempt to associate 2D bounding boxes in the current frame with candidates in the map. A pair of 3D and 2D bounding boxes $(\mathcal{B}_i^{3D}, \mathcal{B}_j^{2D})$ is associated if $|\mathcal{K}_{ij}| > \delta_{\text{obj}}$, where $\mathcal{K}_{ij}$ is a subset of $\mathcal{K}$, $(\mathbf{u}_i, \mathbf{x}_j)$ with $(i, j) \in \mathcal{K}^*$ is a pair of 2D/3D keypoints belonging to a pair of 2D/3D bounding boxes respectively and δ_{obj} is the threshold. In the mapping module, the system triangulates 2D feature correspondences into 3D keypoints, which are assigned with their corresponding 3D bounding box information. Then, it jointly optimizes the camera poses of keyframes $\{\mathbf{T}_0, \ldots, \mathbf{T}_n\}$ and the 3D keypoint positions $\{\mathbf{x}_0, \ldots, \mathbf{x}_m\}$. We consider the 3D bounding boxes in the suspicion phase as IPC candidates and mark them in the map. In the investigation phase, the system detects loop closure to re-localize the drone in the pre-built map. After the drone is successfully re-localized, we further verify existing IPC candidates. In the re-localization stage, if sufficient semantic keypoints belonging to a candidate $\mathcal{B}_i^{3D}$ are associated with a detected vehicle $\mathcal{B}_j^{2D}$ in the current frame, we re-identify the candidate as an IPC and mark it in the map. The proposed solution does not take into account that the local traffic law enforcement officers already have 2D street maps with labeled parking spots, but the drone map can be registered with such 2D street maps to greatly improve IPC detection.

4 Experiments

4.1 Experimental Setup

Our proposed PVD system is embedded in an ATG-R680 drone[1] (see Fig. 6), controlled by a Pixhawk 4[2] advanced autopilot. The maximum take-off weight of the drone is 5.6 kg. We utilize an Argus zoom pot[3] microminiature tri-axis gimbal camera to capture images with a resolution of 2160×3840 pixels at 25 fps. The captured images are then processed by an NVIDIA Jetson TX2

[1] atg-itech.com.
[2] docs.px4.io/v1.9.0/en/flight_controller/pixhawk4.html.
[3] topxgun.com/en/product-argus.html.

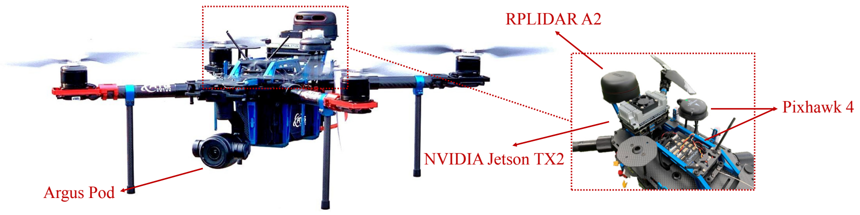


Fig. 6. Experimental setup.

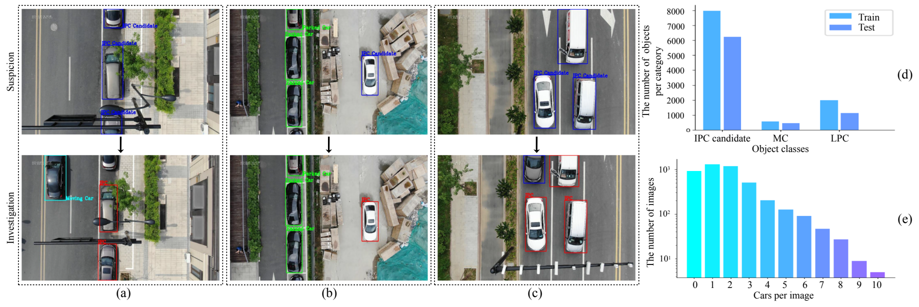


Fig. 7. Our created ATG-PVD dataset: (a)–(c) the images on the first row are used in the suspicion phase, while the images on the second row are used in the investigation phase; (d) and (e) the statistical analysis of the dataset.

GPU[4], which has an 8 GB LPDDR4 memory and 256 CUDA cores, for IPC detection. Furthermore, we also equip our drone with an RPLIDAR A2[5], which can perform 360° omnidirectional laser range scanning.

4.2 ATG-PVD Dataset

Using the aforementioned experimental setup, we created a large-scale real-world dataset, named the ATG-PVD dataset, for parking violation detection. Our dataset is publicly available at sites.google.com/view/atg-pvd for research purposes. The ATG-PVD dataset contains seven sequences (resolution: 2160×3840 pixels) and the corresponding 2D bounding box annotations for car detection and classification. The ground truth used in the suspicion phase has three classes: a) IPC candidates, b) MCs and c) LPCs, while in the investigation phase, the IPC ground truth is also provided. Examples of the images used in the suspicion and investigation phases are shown in Fig. 7(a)–(c).

In our experiments, we divide our ATG-PVD dataset into a training set and a testing set, which respectively contains 4924 and 4398 images. The statistics for these two sets are shown in Fig. 7(d) and (e), where it can be observed that there are more IPC candidates or IPCs than MCs and LPCs. Additionally, most

[4] developer.nvidia.com/embedded/jetson-tx2.
[5] slamtec.com/en/Lidar/A2.

Table 1. Ablation study of our SwiftFlow on the KITTI flow 2015 [44] training dataset. Best results are shown in bold font.

Backbone	Reduce Dense	Shared Weights	Deformable Convolution	F1-all (%)	# Params (M)
PWC-Net	–	–	–	8.37	8.75
	✓	–	–	7.22	5.26
	✓	✓	–	6.95	**2.18**
	✓	✓	✓	**6.51**	2.51

Table 2. The evaluation results on the KITTI flow benchmarks, where DDFlow [18], UnFlow [17], Flow2Stereo [45] and SelFlow [19] are the state-of-the-art self-supervised approaches. Best results are shown in bold font.

Approach	KITTI 2012		KITTI 2015		Runtime (s)
	Out-Noc (%)	Rank	F1-all (%)	Rank	
DDFlow [18]	4.57	60	14.29	91	0.06
UnFlow [17]	4.28	53	11.11	66	0.12
Flow2Stereo [45]	4.02	48	11.10	65	0.05
SelFlow [19]	3.32	34	8.42	51	0.09
SwiftFlow (Ours)	**2.64**	**24**	**7.23**	**35**	**0.03**

images contain fewer than five cars. Furthermore, our experiments are conducted on downsampled images with a resolution of 540×960 pixels. Sections 4.3, 4.4, and 4.5 respectively discuss the performances of SwiftFlow, Flow-RCNN and our PVD system in terms of both qualitative and quantitative experimental results.

4.3 Evaluation of SwiftFlow

Ablation Study. We conduct an ablation study to validate the effectiveness of SwiftFLow. The experimental results are presented in Table 1. We can see that, by removing dense connections between different levels, our approach can reduce many parameters, but still retain a similar optical flow estimation performance, compared with the PWC-Net [13] baseline. Moreover, sharing weights of flow estimation modules can yield a performance improvement with fewer parameters. Furthermore, thanks to deformable convolution, our proposed SwiftFlow achieves the best performance with only a few additional parameters.

Evaluation. Since our ATG-PVD dataset does not contain optical flow ground truth, we evaluate our proposed SwiftFlow on the KITTI flow 2012 [46] and 2015 [44] benchmarks. According to the online leaderboard of the KITTI flow benchmarks, as shown in Table 2, our SwiftFlow ranks 24th on the KITTI flow 2012 benchmark[6] and 35th on the KITTI flow 2015 benchmark[7], outperforming

[6] cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=flow.
[7] cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=flow.

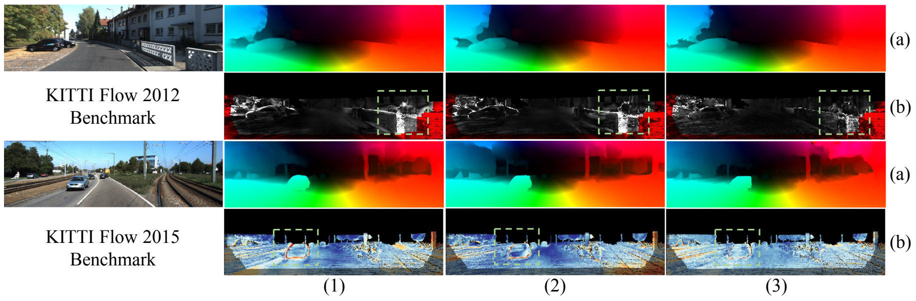


Fig. 8. Examples from the KITTI flow benchmarks, where rows (a) and (b) on columns (1)–(3) show the optical flow estimations and the corresponding error maps of (1) UnFlow [17], (2) SelFlow [19] and (3) our SwiftFlow, respectively. Significantly improved regions are highlighted with green dashed boxes. (Color figure online)

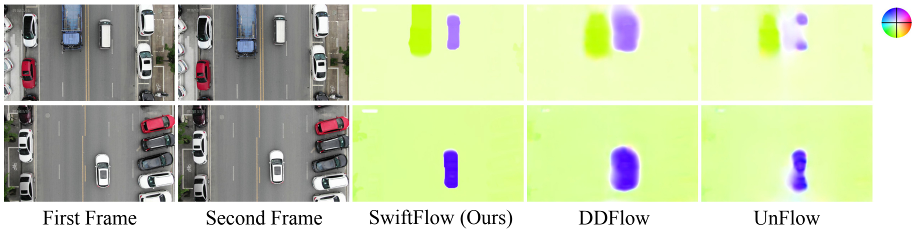


Fig. 9. Examples of the optical flow estimation results on our ATG-PVD dataset. Our proposed SwiftFlow is compared with DDFlow [18] and UnFlow [17].

all other state-of-the-art unsupervised optical flow estimation approaches, with a faster running speed (in real time) achieved in the mean time. Figure 8 presents examples from the KITTI flow benchmarks, where we can see that SwiftFlow yields more robust results than others. Furthermore, Fig. 9 shows optical flow estimation results on our ATG-PVD dataset, indicating that our proposed SwiftFlow performs much more accurately than both DDFlow [18] and UnFlow [17], another two well-known unsupervised optical flow estimation approaches, especially on the boundary of the MCs.

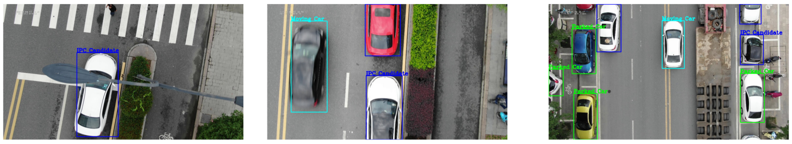
4.4 Evaluation of Flow-RCNN

In our experiments, we compute the mean average precision (mAP) over ten IoU thresholds between 0.50 and 0.95 (refer to [47] for more details) to quantitatively evaluate the performance of our proposed Flow-RCNN. It should be noted that IPCs are regarded as IPC candidates in both training and testing experiments, due to the fact that IPCs are re-identified as IPC candidates.

We compute mAP for all three categories (IPC candidate, MC and LPC) so as to comprehensively evaluate the performance of our proposed Flow-RCNN. The quantitative results are provided in Table 3, where it can be observed that

Table 3. Car detection mAP, where the best results are shown in bold font.

Method	Total	IPC candidate	MC	LPC
Faster-RCNN [7]	0.770	0.844	0.672	0.789
Flow-RCNN (ours)	**0.789**	**0.845**	**0.733**	**0.796**



(a) Occlusion (b) Motion blur (c) Complex environment

Fig. 10. Examples of our Flow-RCNN results.

Flow-RCNN outperforms the baseline network Faster-RCNN [7] (especially for MC detection) in terms of both car detection and classification. It is rather astonishing that Faster-RCNN can still successfully detect many MCs from only RGB images, even without using optical flow information. We speculate that the baseline network might also consider the road textures around a car when inferring its category. For instance, an MC is typically at the center of a lane, and the road textures around it are similar, which can weaken the influence caused by motion blur problem.

Experimental results of our Flow-RCNN are given in Fig. 10, showing the robustness of our proposed approach. For example, in Fig. 10(a), the light pole, that occludes part of an IPC candidate, can produce a similar optical flow estimation to an MC. Fortunately, our Flow-RCNN which fuses both RGB and flow information can still detect the IPC candidate correctly. Furthermore, although it is hard to extract features from a blurred car image, it can be seen in Fig. 10(b) that our proposed approach can avoid such misdetections by leveraging additional optical flow information. Moreover, in complex environments, such as the case shown in Fig. 10(c), car with different categories can still be successfully detected and classified.

4.5 Evaluation of Parking Violation Detection

We also comprehensively evaluate the performance of the entire system for parking violation detection using our ATG-PVD dataset, and a precision of 91.7%, a recall of 94.9% and an F1-Score of 93.3% are achieved. An example of the detected IPCs in the map is illustrated in Fig. 11, where readers can observe that our proposed suspect-and-investigate system can detect parking violations effectively and efficiently.

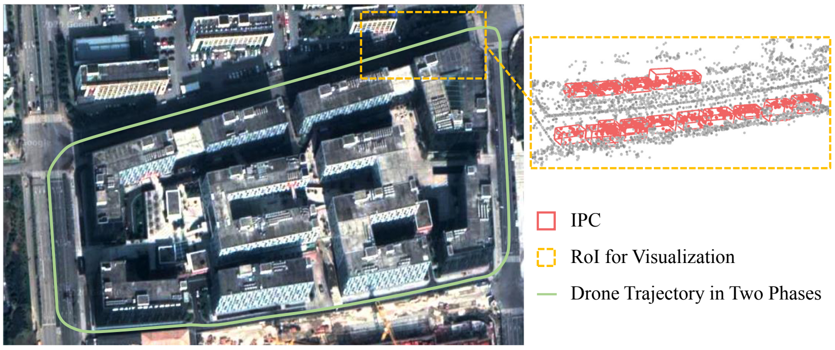


Fig. 11. An example of the detected IPCs in the map.

5 Conclusion

In this paper, we proposed a novel, robust and cost-effective parking violation detection system embedded in an ATG-R680 drone equipped with a TX2 GPU. Our system utilizes a so-called suspect-and-investigate framework, which consists of: 1) an unsupervised optical flow estimation network named SwiftFlow, 2) a novel flow-guided object detection network named Flow-RCNN, and 3) a drone re-localization and IPC re-identification module based on VSLAM. On the KITTI flow 2012 and 2015 benchmarks, our proposed SwiftFlow outperforms all other state-of-the-art unsupervised optical flow estimation approaches in terms of both speed (real-time performance was achieved) and accuracy. By incorporating the inferred optical flow information into our object detection framework, IPC candidates, MCs and LPCs can be effectively detected and classified, even in many challenging cases. In the investigation phase, our VSLAM module detects loop closure to re-localize the drone in the pre-built map. After the drone is successfully re-localized, we further re-identify whether an existing IPC candidate is an actual IPC. The experimental results both qualitatively and quantitatively demonstrate the effectiveness and robustness of our proposed parking violation detection system.

Acknowledgements. This work was supported by the National Natural Science Foundation of China, under grant No. U1713211, Collaborative Research Fund by the Research Grants Council Hong Kong, under Project No. C4063-18G, and the Research Grant Council of Hong Kong SAR Government, China, under Project No. 11210017, awarded to Prof. Ming Liu.

References

1. McKee, M., Stuckler, D.: If the world fails to protect the economy, covid-19 will damage health not just now but also in the future. Nat. Med. **26**(5), 640–642 (2020)
2. Hughes, T.: Poor, essential and on the bus: coronavirus is putting public transportation riders at risk, April 2020

3. Garcetti, E.: Mayor garcetti relaxes parking enforcement, March 2020
4. Zhou, Y., Liu, L., Shao, L., Mellor, M.: DAVE: a unified framework for fast vehicle detection and annotation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 278–293. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_18
5. Regazzoni, C.S., Cavallaro, A., Wu, Y., Konrad, J., Hampapur, A.: Video analytics for surveillance: theory and practice [from the guest editors]. IEEE Signal Process. Mag. **27**(5), 16–17 (2010)
6. Council, B.C.: Codes of practice for operation of CCTV Enforcement Cameras. Birmingham City Council (2013)
7. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
8. Horn, B.K., Schunck, B.G.: Determining optical flow. In: Techniques and Applications of Image Understanding, vol. 281, pp. 319–331. International Society for Optics and Photonics (1981)
9. Mémin, E., Pérez, P.: Dense estimation and object-based segmentation of the optical flow with robust techniques. IEEE Trans. Image Process. **7**(5), 703–719 (1998)
10. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24673-2_3
11. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV-L^1 optical flow. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 214–223. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74936-3_22
12. Dosovitskiy, A., et al.: Flownet: learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2758–2766 (2015)
13. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-NET: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8934–8943 (2018)
14. Hui, T.W., Tang, X., Change Loy, C.: Liteflownet: a lightweight convolutional neural network for optical flow estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8981–8989 (2018)
15. Hur, J., Roth, S.: Iterative residual refinement for joint optical flow and occlusion estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5754–5763 (2019)
16. Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X., Zha, H.: Unsupervised deep learning for optical flow estimation. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
17. Meister, S., Hur, J., Roth, S.: Unflow: unsupervised learning of optical flow with a bidirectional census loss. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
18. Liu, P., King, I., Lyu, M.R., Xu, J.: DDFlow: learning optical flow with unlabeled data distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 8770–8777 (2019)
19. Liu, P., Lyu, M., King, I., Xu, J.: SelFlow: self-supervised learning of optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4571–4580 (2019)

20. Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell. **42**(2), 318–327 (2020)
21. Girshick, R.: Fast R-CNN. In: International Conference on Computer Vision (ICCV) (2015)
22. Law, H., Deng, J.: Cornernet: detecting objects as paired keypoints. In: The European Conference on Computer Vision (ECCV), September 2018
23. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. In: arXiv preprint arXiv:1904.07850. (2019)
24. Klein, G., Murray, D.: Parallel tracking and mapping for small ar engel2018dso. In: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 1–10. IEEE Computer Society (2007)
25. Strasdat, H., Davison, A.J., Montiel, J.M., Konolige, K.: Double window optimisation for constant time visual slam. In: 2011 International Conference on Computer Vision, IEEE (2011) 2352–2359
26. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular slam system. IEEE Trans. Rob. **31**(5), 1147–1163 (2015)
27. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: dense tracking and mapping in real-time. In: IEEE 2011 International Conference on Computer Vision, pp. 2320–2327 (2011)
28. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_54
29. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 611–625 (2018)
30. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: fast semi-direct monocular visual odometry. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 15–22. IEEE (2014)
31. Yang, S., Scherer, S.: CubeSLAM: monocular 3-D object SLAM. IEEE Trans. Rob. **35**(4), 925–938 (2019)
32. Huang, J., Yang, S., Mu, T.J., Hu, S.M.: ClusterVO: clustering moving instances and estimating visual odometry for self and surroundings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2168–2177 (2020)
33. Nicholson, L., Milford, M., Sünderhauf, N.: QuadricSLAM: dual quadrics from object detections as landmarks in object-oriented SLAM. IEEE Robot. Autom. Lett. **4**(1), 1–8 (2018)
34. Runz, M., Buffier, M., Agapito, L.: MaskFusion: real-time recognition, tracking and reconstruction of multiple moving objects. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE 2018, pp. 10–20 (2018)
35. McCormac, J., Clark, R., Bloesch, M., Davison, A., Leutenegger, S.: Fusion++: Volumetric object-level slam. In, : international conference on 3D vision (3DV). IEEE **2018**, 32–41 (2018)
36. Zhang, T., Zhang, H., Li, Y., Nakamura, Y., Zhang, L.: FlowFusion: dynamic dense RGB-D slam based on optical flow. arXiv preprint arXiv:2003.05102 (2020)
37. Tateno, K., Tombari, F., Laina, I., Navab, N.: CNN-SLAM: real-time dense monocular slam with learned depth prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6243–6252 (2017)
38. Dai, J., et al.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 764–773 (2017)

39. Wang, Y., Yang, Y., Yang, Z., Zhao, L., Wang, P., Xu, W.: Occlusion aware unsupervised learning of optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4884–4893 (2018)
40. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271), pp. 839–846. IEEE (1998)
41. Jonschkowski, R., Stone, A., Barron, J.T., Gordon, A., Konolige, K., Angelova, A.: What matters in unsupervised optical flow. arXiv preprint arXiv:2006.04902 (2020)
42. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source slam system for monocular, stereo, and RGB-D cameras. IEEE Trans. Rob. **33**(5), 1255–1262 (2017)
43. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: an efficient alternative to sift or surf. In: International Conference on Computer Vision. IEEE **2011**, 2564–2571 (2011)
44. Menze, M., Heipke, C., Geiger, A.: Joint 3d estimation of vehicles and scene flow. In: ISPRS Workshop on Image Sequence Analysis (ISA) (2015)
45. Liu, P., King, I., Lyu, M.R., Xu, J.: Flow2Stereo: effective self-supervised learning of optical flow and stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6648–6657 (2020)
46. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
47. Lin, T.: Microsoft COCO: common objects in context. CoRR abs/1405.0312 (2014)