Optical Flow Based Motion Detection for Autonomous Driving

Ka Man Lo
kamanphoebe@gmail.com

https://github.com/kamanphoebe/MotionDetection.git

Abstract

Motion detection is a fundamental but challenging task for autonomous driving. In particular scenes like highway, remote objects have to be paid extra attention for better controlling decision. Aiming at distant vehicles, we train a neural network model to classify the motion status using optical flow field information as the input. The experiments result in high accuracy, showing that our idea is viable and promising. The trained model also obtain an acceptable performance for nearby vehicles. Our work is implemented in PyTorch and open tools including nuScenes, FastFlowNet and RAFT are used. Visualization videos are available at https://www.youtube.com/playlist?list=PLVVrWgq4OrlBnRebmkGZO1iDHEksMHKGk.

I. Introduction

Motion detection, or moving object detection, is a computer vision related technique for detecting the physical movement of an object relative to its background. It is widely used in various areas like smart homes, surveillance and security, and also plays a crucial role in autonomous driving. To make better future plan on controlling during driving, vehicles need to monitor the road condition well. Careful inspection for faraway environment is required for scenes that allow high-speed driving like highways or quiet roads. However, the perception range of lidar and radar sensors are not always far enough to cover distant objects and thus computer vision based methods should be applied under these circumstances. Traditional methods of motion detection rely on the difference of pixels between frames. Therefore, detecting motion in the distance, especially those in radial direction, is a challenging issue since they are usually just a few pixels changes.

Optical flow estimation is a commonly used technique in motion detection tasks for providing velocity information. It is calculated based on the brightness constancy constraint, supposing the timestamps of two consecutive frames are close enough that the brightness of the same positions in real world will remain unchanged. In this paper, we use different algorithms to obtain optical flow field information of vehicles in between 30 to 70 meters from the nuScenes [2] dataset, and feed them into neural network ResNet18 [4] as inputs. The model then outputs the binary prediction of motion status, i.e., still or moving. Our experiments show that the moving targets are successfully detected with a high correct rate. We also use the trained model to infer nearby vehicles and obtain a reasonable accuracy.

The rest of the paper is organized as follows: Section II gives a brief review of relevant works. Section III demonstrates the framework of our work, followed by the experimental details and results in Section IV. Finally, conclusions and possible future work are presented in Section V.

II. Related Work

In this section, basic approaches regarding motion detection are firstly reviewed, together with specific topics relevant to our work, namely small object motion detection and autonomous driving. After that, we dive deeper into optical flow algorithms.

A. Motion detection

Traditional methods used in detecting motion can be mainly divided into four categories: background subtraction, frame difference, temporal difference and optical flow estimation. A previous review done by Manchanda and Sharma [7] includes works using these approaches to detect motion for general purposes. The works they list were published from 2009 to 2015. There is also some afterwards improvement based on the basic methods in recent years, such as [15][14][1][5].

For recent works about motion detection in autonomous driving, both [8] and [13] make use of CRF related model. The former targets at a specific range while the latter jointly feeds disparity map and optical flow field as model input. Our work is inspired by [9]. Yet we concentrate solely on motion classification for bounding boxes so far while their work combines object detection and motion segmentation. Besides, we exploit the original optical flow information instead of converting it to RGB images so as to prevent normalization in this process and preserve numerical precision. More details about our implementation are stated in Section III.

Focusing on the topic of small object motion detection, existing works usually focus on insects, like [12][10], which entirely differ from autonomous driving in appearance and background.

B. Optical flow

FastFlowNet [6] and RAFT [11] achieve state-of-the-art speed and accuracy respectively for estimating optical flow field. FastFlowNet is $10\times$ faster than RAFT while RAFT obtains a F1 error of 5.10% on the KITTI [3] dataset, which is only half of the value of FastFlowNet. The two algorithms are used and are compared with each other in our work. Example inferences of FastFlowNet and RAFT using the same raw image pair are depicted in Figure 1.

III. Method

The framework of our work is presented in this section, starting with the pipeline and followed by details of labeling and data prepoccessing.

Figure 1: Examples of optical flow predictions on the nuScenes dataset. From left to right: the preceding raw image of a keyframe pairs, flow visualization of FastFlowNet and RAFT.

A. Pipeline

The overview of our work is outlined below:

1. Select keyframe pairs that contain target vehicles from nuScenes
2. Generate optical flow field for all keyframe pairs via FastFlowNet or RAFT
3. Label the objects as still or moving by estimating their velocity
4. Extract optical flow information of objects within the modified 2D bounding boxes after some preprocessing and feed them into the neural network
5. Train a binary classifier from scratch using ResNet18 along with some necessary adjustments of layers

B. Labeling

Data of 2D bounding box and binary motion ground truth are recorded in every label. The former is marked by the coordinates `xmin`, `xmax`, `ymin` and `ymax`, which are simply deduced from the eight corners of the original 3D bounding box by picking the minimum and the maximum of `x` and `y`.

The motion ground truth is decided based on the velocity calculated as

$$velocity = \frac{position_2 - position_1}{timestamp_2 - timestamp_1} \tag{1}$$

where $position$ is given with respect to the global coordinate system. If the absolute value of velocity $\mid velocity \mid \geq 2$ m/s, then the object is marked as moving and vice versa.

C. Data preprocessing

To determine whether an object is moving, we need the optical flow information not only of the object itself, but also of the background around. Therefore, some preprocessing has to be done on the 2D bounding box before inputting to the network, as mentioned in the 4th step of the pipeline. First, the box is reshaped to be a square with side length $max(width, height)$. Then triple the length of sides and if needed, pad the box with edge values. Finally, cut off data outside the box and resize it to 224×224 using bilinear interpolation.

IV. Experiments

In this section, we first detail the dataset used and experimental setup. Then the results are discussed. The construction and evaluation of a generalized dataset are presented at last.

Table 1: Settings for filtering the nuScenes dataset.

Setting	Description
Target categories	vehicle.car vehicle.emergency.ambulance vehicle.emergency.police vehicle.truck vehicle.bus.bendy vehicle.bus.rigid vehicle.construction
Distance	30m - 70m
Visibility	80% - 100%
Sensor	CAM_FRONT

A. Dataset

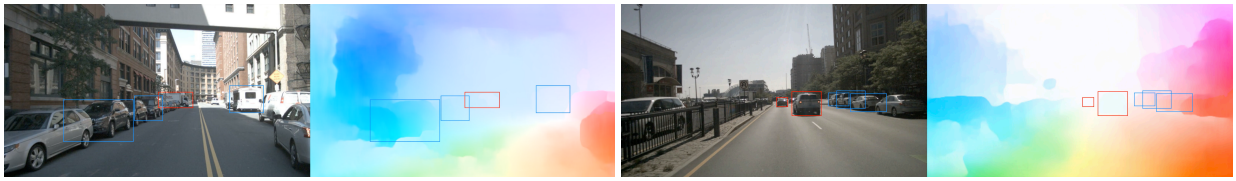
Our model is trained and evaluated on the filtered nuScenes [2] dataset. nuScenes comprises 1000 diverse scenes, covering different locations, time and weather conditions. For simplicity, we exclude scenes of "night", "rain" and "lightning", thereby 604 scenes are remained. We then collect keyframe pairs that contain any of the seven types of vehicles within the specific distance and visibility range as shown in Table 1. After that, optical flow field of the frame pairs is calculated through FastFlowNet or RAFT, and is saved as `.npy` file. As a result, we obtain 18460 objects in total, while 16467 of them used as training set and 1993 for evaluation. Considering the amount of data is rather small, random horizontal flip with probability 0.5 is performed for data augmentation.

B. Experimental setup

The model architecture we used is chosen to be ResNet18 [4]. However, since the input is not in the form of RGB images, we have to train the model from scratch instead of applying a pretrained model. Therefore, the number of output channels of the first convolutional layer is modified to be 64 to make the network adapt to our input. The size of final output is also changed to 1, which should be a number within [0, 1]. If the output value greater than 0.5, the object will be classified as moving and vice versa. The rest of the model structure stay unchanged. Table 2 lists the settings of hyperparameters.

C. Experimental result

The accuracy of our model is greater than 90%, as shown in Table 3. The model trained in FastFlowNet input unexpectedly



(a) Correct predictions

(b) Incorrect predictions due to unclear flow

(c) Incorrect predictions due to misleading surround

Figure 2: Visualization of inference using the model trained with FastFlowNet input. Blue boxes and red boxes represent still and moving objects respectively. These are the frames of the generalized dataset so nearby objects are also detected.

Table 2: Hyperparameter setting for ResNet18 model training.

Function	Hyperparameter	Setting
Dataloader	Batch size	128
Optimizer	Algorithm	SGD
	Learning rate	0.01
	Weight decay	0.01
	Momentum	0.9
Scheduler	Schedule	StepLR
	Step size	10
	Gamma	0.5

Table 3: Quantitative performance of models with optical flow input generated by various algorithms. The pretrained model of both optical flow algorithms is trained on the KITTI [3] dataset.

Optical flow algorithm	F1 (%)	Precision (%)	Recall (%)
FastFlowNet	92.9	94.3	91.7
RAFT	89.5	89.7	89.9

have better performance than the one with input genernated by RAFT, which achieves state-of-the-art accuracy for optical flow estimation. Note that it is not fair to compare our results directly with other motion detection methods, since we simplify the data a lot. Nonetheless, the pretty values still reflect the feasibility of the idea of our work.

Predictions are visualized for intuitive understanding of the model performance. Several correct and incorrect inferences are depicted in Figure 2. We sum up two main reasons for the wrong classifications:

- **Unclear optical flow for remote or slow objects**. These kind of objects are always tough to deal with because of the tiny difference of distance in visual world, hence the unclear flow which confuses the network.
- **Being affected by the background or foreground**. As mentioned in Section III, the surrounding flow information is also included as part of the input. Therefore, it could be misleading when there are some other objects got involved.

D. Generalization

We extend our filtered dataset for inference by adding non-keyframes and nearby objects from nuScenes. To start with, optical flow is calculated for frame pairs at 4-frame intervals. Since there is no annotation for non-keyframes in nuScenes, bounding box positions and ground truths of objects are estimated based on the information of corresponding keyframes. The former are calculated by linear interpolation while the latter remain the same as the closest previous keyframes with respect to the non-keyframes. Vehicles near than 30m meters, but with visibility requirement as before, are also taken into account. Evaluating on this generalized dataset, the F-score of our model significantly drops to 60%, as shown in Table 4. Our visualization videos contain inference of generalization.

Table 4: Quantitative performance of evaluation on the generalized dataset. The pretrained model of optical flow algorithm is trained on KITTI.

Optical flow algorithm	F1 (%)	Precision (%)	Recall (%)
FastFlowNet	60.4	63.0	62.8

V. Conclusion and future work

In this paper, we have investigated the effect of binary motion classification for annotated remote vehicles by inputting optical flow information to neural network. The experimental result reports that our model can successfully detect the mo-

tion and the high accuracy illustrates the great potential of our idea. Cases failed to be correctly inferred are mainly caused by unclear optical flow and misleading surrounding flow information. Our trained model is not so applicable for nearby objects yet the performance might dramatically enhance if the model is trained with them. There is still much room for improvement for our work:

- Remove the strict rules for filtering data to adapt to concrete use
- Train the optical flow model from scratch (by self-supervised learning), rather than apply pretrained models
- Construct an end-to-end classification network architecture, leaving the middle stages regarding generating optical flow field to be implicit

Acknowledgement

The code for generating optical flow field information is based on the corresponding original projects, FastFlowNet and RAFT.

References

[1] Oussama Boufares, Mohamed Boussif, and Noureddine Aloui. Moving object detection system based on the modified temporal difference and otsu algorithm. In *2021 18th International Multi-Conference on Systems, Signals Devices (SSD)*, pages 1378–1382, 2021.

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[3] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[5] Junjie Huang, Wei Zou, Zheng Zhu, and Jiagang Zhu. An efficient optical flow based motion detection method for non-stationary scenes. In *2019 Chinese Control And Decision Conference (CCDC)*, pages 5272–5277, 2019.

[6] Lingtong Kong, Chunhua Shen, and Jie Yang. Fastflownet: A lightweight network for fast optical flow estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[7] Sumati Manchanda and Shanu Sharma. Analysis of computer vision based techniques for motion detection. In *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, pages 445–450, 2016.

[8] Fahimeh Nezhadalinaei, Lei Zhang, Mohammad Mahdizadeh, and Faezeh Jamshidi. Motion object detection and tracking optimization in autonomous vehicles in specific range with optimized deep neural network. In *2021 7th International Conference on Web Research (ICWR)*, pages 53–63, 2021.

[9] Mennatullah Siam, Heba Mahgoub, Mohamed Zahran, Senthil Yogamani, Martin Jagersand, and Ahmad El-Sallab. Modnet: Motion and appearance based moving object detection network for autonomous driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2859–2864, 2018.

[10] Vladan Stojnić, Vladimir Risojević, Mario Muštra, Vedran Jovanović, Janja Filipi, Nikola Kezić, and Zdenka Babić. A method for detection of small moving objects in uav videos. *Remote Sensing*, 13(4), 2021.

[11] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. *CoRR*, abs/2003.12039, 2020.

[12] Hongxin Wang, Jigen Peng, and Shigang Yue. A feedback neural network for small target motion detection in cluttered backgrounds. *CoRR*, abs/1805.00342, 2018.

[13] Zhipeng Xiao, Bin Dai, Tao Wu, Liang Xiao, and Tongtong Chen. Dense scene flow based coarse-to-fine rigid moving object detection for autonomous vehicle. *IEEE Access*, 5:23492–23501, 2017.

[14] Zhenxiong Xu, Danhong Zhang, and Lin Du. Moving object detection based on improved three frame difference and background subtraction. In *2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, pages 79–82, 2017.

[15] Junhui Zuo, Zhenhong Jia, Jie Yang, and Nikola Kasabov. Moving target detection based on improved gaussian mixture background subtraction in video images. *IEEE Access*, 7:152612–152623, 2019.