A Real-Time Method to Estimate Speed of Object Based on Object Detection and Optical Flow Calculation

Kaizhan Liu¹, Yunming Ye¹*, Xutao Li¹ and Yan Li²

¹Department of Computer Science, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China
²Shenzhen Polytechnic, Shenzhen, China

* yeyunming@hit.edu.cn

Abstract. In recent years Convolutional Neural Network (CNN) has been widely used in computer vision field and makes great progress in lots of contents like object detection and classification. Even so, combining Convolutional Neural Network, which means making multiple CNN frameworks working synchronously and sharing their output information, could figure out useful message that each of them cannot provide singly. Here we introduce a method to real-time estimate speed of object by combining two CNN: YOLOv2 and FlowNet. In every frame, YOLOv2 provides object size; object location and object type while FlowNet providing the optical flow of whole image. On one hand, object size and object location help to select out the object part of optical flow image thus calculating out the average optical flow of every object. On the other hand, object type and object size help to figure out the relationship between optical flow and true speed by means of optics theory and priori knowledge. Therefore, with these two key information, speed of object can be estimated. This method manages to estimate multiple objects at real-time speed by only using a normal camera even in moving status, whose error is acceptable in most application fields like manless driving or robot vision.

1. Introduction

Speed, as a physical quantity to describe one object’s motion referring to another one, is objective and quantizable. We human cannot directly measure the value of speed by our sensors, so normally we can just estimate it through other indirect methods. Speed estimation plays an important role in many application fields, such as vehicle speed detection in transportation industry, object motion analysis in robot field, pedestrian behavior analysis in manless driving field. To practically apply speed estimation method to above application fields, three basic conditions are needed to be reached: 1) the running speed of method should be or be closed to real-time, 2) the ability to simultaneously measure multiple objects’ speed, 3) keeping valid even the observer itself is moving.

On the other hand, thanks to the flourish of deep learning on computer vision field, some traditional difficult tasks are provided with new solutions. In object detection field, a series of CNN-based algorithm such as RCNN [1], fast RCNN [2], SSD [3], YOLO [4] were proposed, where SSD and YOLO is able to reach real-time speed. These algorithms are able to output bounding boxes of multiple objects from each frame image. In optical flow calculation field, Philipp Fischer et al. [5] designed out a CNN to calculate optical flow, which shows approximate performance compared to the traditional algorithms and also reaches the real-time speed.
Though above CNN-based algorithms are in specific niche which has indirect relation with speed estimation, neither of them is able to achieve speed estimation singly. With the idea of sharing output information of multiple networks so as to figure out useful message that each of them cannot provide singly, this paper proposes a general speed estimation method based on combination of object detection network and optical flow calculation network (ODFSE). It uses the bounding box from object detection network to select out optical flow of object from the whole flow image provided by FlowNet, and uses the label and image size of object, by means of optics theory, priori knowledge and camera calibration. Shown by experiment result, this method is able to estimate speed of different type of object in real-time speed even the camera itself is in moving status.

2. Related Work

Normally we estimate the speed of object by two main types of method: physical method (e.g. Radar and Laser) or image processing method [6]. These methods have respectively advantages but also have respectively disadvantages.

About the physical method, the principle is: an ejector of radar or laser ejects the wave or light signal and a receiver receives the returned signal blocked by object. Since the speed of object will change the signal’s frequency, by comparing the frequency of received signal with ejecting signal we can calculate out the speed. This method has been widely used in market. Company like VST, Bushnell, Lai Lai and so on develop a number of embedded devices of vehicle speed estimation, running at real-time speed, 1 km/h error and measuring range is 18-300km/m. Apart from the benefits, this kind of method shows some drawbacks: 1) can only measure speed of vehicle and ball, 2) can only measure one object at the same time, 3) measuring distance is limited, 4) device itself must be still.

About the image processing method, there are multiple solutions. Yu et al. [7] uses camera calibration method to get the relationship between camera and real world, then using image matching method to calculate the change of location of vehicle to estimate the speed of vehicle. Wang [8] used priori knowledge of vehicle’s size to achieve object tracking, then use frame difference to calculate out the speed of vehicle. Yan et al. [9] used the information of fixed point in camera to calculate the speed of camera itself. A.L. Harvey et al. [10] considers the correlation of two neighbouring frames to estimate the distance vehicle moves each frame thus transforms it to its speed. Similarly, S. Pumrin et al. [11] uses frame difference method and camera calibration method to achieve speed estimation. The methods mentioned above are all used to measure the speed of vehicle. As for other objects, Huei-Yung Lin and Chia-Hong Chang [12] found that the motion blur’s parameter has relationship with speed of spherical object and use it to calculate out the speed of it, this method is only suitable for spherical object. Above methods are well-behaved in some specific scenes but have a number of drawbacks: 1) most of them can only be used to measure speed of vehicle, some might be used to spherical object, 2) the precision and robustness are limited by the precision and robustness of image processing algorithm itself 3) most of these methods require camera to be still because most of them use frame difference method to detect object, which needs the background to be unchanged.

3. Speed estimation method ODFSE

The process of ODFSE is: Firstly, using the bounding boxes from object detection network to select out the object parts of the whole optical flow image from optical flow calculation networks. Secondly, figuring out the transformation relationship between camera and real world thanks to the information of object type and object width and height in pixels provided by object detection network, where the former one helps to get the real size of object with prior knowledge and the latter one helps to get the image size of object. Lastly, transform the optical flow to real speed with above information together with optics theory and camera calibration. The framework of the method is shown in Figure 1.
3.1. Object detection

To achieve object detection in real-time, we choose YOLOv2 [13] as the basic network. The reasons are: 1) the same as YOLOv1 [4], YOLOv2 is an end-to-end CNN which takes object detection problem as a regression problem, makes it possible to run in real-time speed, 2) YOLOv2 use 98 anchor boxes with pre-selected size to represent object, whose width and height are approximately equal to the width and height of object, 3) YOLOv2’s detecting object relies on the whole image information instead of partial information where CNN like SSD and Faster-RCNN rely, so it might perform better in the situation when objects are moving, which is the case in our work.

We at the beginning used the original network trained with COCO dataset which contains 80 types of object to detect a horizontal moving smart car in camera. Basically network is able to catch the smart car in still or slow-moving state. But as the speed of smart car increased, network began to lose the detection because the image of object became blurring and network did not get used to its features. To solve this problem, we make a tiny dataset where objects move in serious of level of speed to retrain the network. The dataset is divided in 10 groups, whose speed ranges from 50 cm/s to 150 cm/s with a 10 cm/s step. Each of them contains 5 videos representing 5 different types of object (e.g. bottle, motor car and so on). The learning rate is set to 0.001, epoch is 50. A compare of retrained network with original network is shown in figure 2, where (a) is the output of original network and (b) is the output of retrained network. We can see the retrained one is more capable to catch object with high speed than the original one.

![Figure 1. The framework of ODFSE.](image)

![Figure 2(a). (a) Output of original network. It is not able to catch object in fast speed. (b) Output of retrained network. It is able to catch object in fast speed.](images)
prior knowledge of object real size, that is: if the object type is “Person”, we then consider the object size is the size of an ordinary grown-up man with height of 170cm and weight of 30cm. About the object size we use the ratio of pixels of object and pixels of whole image to represent. The object location does not take part in calculation of speed but it helps the FlowNet select out the object part, then we are able to get the optical flow of each object.

3.2. Optical flow calculation
FlowNet [5] is a network to calculate optical flow. It is an end-to-end CNN which takes the front and back frame as input then output optical flow value of whole image. The principle of the network is using convolutional layer to extract features of the difference between the front one and back one, then using upconvolutional layer to put the features back to image size, thus the optical flow of whole image is calculated out. As the information is for the whole image, it actually cannot give out the respective optical flow value of each object. To achieve this, as described in 3.1, the object bounding box helps to select out object part as shown in figure 2(a), and the selected part is shown in figure 3(b).

![Figure 3](image)

**Figure 3.** (a) Using bound box to select out the object part of optical flow. (b) The image contains both optical flow of object and background.

To calculate the optical flow value of object in figure 2(b), we need to classify the object with background. Here we use k-means algorithm to do so. As most pixels of the bounding box belong to object, the cluster containing the most elements should belong to the object, and the rest should belong to background. By taking the average value of both of them we manage to get optical flow of object and background. By the way, the ability to calculate out optical flow of background enables the method working even in a moving status.

3.3. Calculation of true speed
To calculate the real speed of object from above data, we need to find out the correspondence between camera optics parameter and real world.

The model of this correspondence is: assuming one point $m_0$ owns real speed $v_0$ (to testify in convinience here we assume the camera is still, and the speed direction is parallel to camera plane), its projection on image $m'_0$, then owns speed $v'_0$. In unit time $dt$, the journey of them are $v_0 dt$ and $v'_0 dt$. Following the theory of camera projection, there should be:

$$\frac{v_0 dt}{z} = \frac{v'_0 dt}{f}$$

(1)

Where $f$ stands for focal length of camera, $z$ stands for distance between camera and object. Besides, the relationship of image speed $v'_0$ and optical flow speed $v_{opt}$ should be:

$$v_{opt} = v'_0 * k_0$$

(2)
Where \( k_0 \) stands for a constant value.
From (1), (2) we got:

\[
v_0: v_{opt} = z: (f \cdot k_0) = z: k
\]  

(3)

Where \( k = f \cdot k_0 \) is specific value as \( f \) is specific with specific camera.
Thus we find out the correspondence of \( v_0 \) and \( v_{opt} \). Figure 4 shows the vector diagram.

**Figure 4.** The vector diagram of camera focusing principle.

**Figure 5.** The vector diagram to show the relationship among \( z, \theta, h_{sum} \) and \( h_{prior} \).

The remaining problem is to find the specific value \( k \) and the formulation to get value \( z \). As a specific value, we use calibration to get value \( k \). The process is: make an object moving in real speed \( v_1 \) in distance of \( z_1 \) with camera, get the optical flow speed \( v_{opt1} \) from FlowNet, then we get:

\[
k = v_{opt1} \cdot z_1 \cdot v_1^{-1}
\]

(4)

About the value \( z \), it is a physical quantity which can be measured by manpower or depth sensor, but as mentioned in chapter 3.1, we can use prior knowledge of object size to figure out the value even without above conditions. Figure 5 shows the principle.

\( h_{prior} \) is the prior height of object mentioned in chapter 3.1, \( h_{sum} \) is the height that camera can at most shoot in distance of \( z, \theta \) is angle of view. So we can get the relationship:

\[
z = \frac{h_{sum}}{\tan\theta} = \frac{h_{prior}}{r_{yolo} \cdot \tan\theta}
\]

(5)

Where \( r_{yolo} \) is the ratio of \( h_{prior} \) and \( h_{sum} \) that YOLO can offer.
\( \tan\theta \) is the specific value of camera, which can also be accessed by calibration, the process is: take an object with known height \( h_2 \) (e.g. a ruler), adjust its distance with camera until the object can filling the screen of camera, record the distance \( z_2 \), then:

\[
\tan\theta = h_2 \cdot z_2^{-1}
\]

(6)

From (4), (5) and (6):

\[
v_0 = \frac{h_{prior} \cdot z_2}{r_{yolo} \cdot h_2} \cdot \frac{v_1}{v_{opt1} \cdot z_1} \cdot v_{opt}
\]

(7)

Since \( z_2, h_2, v_1, v_{opt1}, z_1 \) are known parameters, \( h_{prior} \) is the prior value, \( r_{yolo} \) are output of YOLO network, \( v_{opt} \) is output of FlowNet, we can calculate out the true speed \( v_0 \). Note that \( v_0 \) and \( v_{opt} \) are vector type that is we can not only calculate out the value of speed but also the direction of it.
Besides, for convenience we assume the camera is still, but as described in chapter 3.2, the optical flow can still be calculate correctly even the camera is moving.

4. Experiment

4.1. Measurement Platform
Since there is no public dataset about speed, we need to make a dataset with accurate and actual ground truth of speed. Here we set up a smart car which can load some tiny objects and move together with a constant speed. Also, two hall elements are installed beside its tyres so the actual speed of car and object can be captured in real-time. Thus, we are able to simulate the moving scene of an object and do the experiment to evaluate our method.

4.2. Series of experiments
To prove the accuracy, robustness and stability of this method, we set up 3 experiments referring to different topics: 1) the accuracy and stability of our method; 2) speed value influence on method’s effectiveness; 3) camera movement’s influence on method’s effectiveness. Here are some relative variables: 1) estimated object value \( v_o \); 2) ground truth of object speed value \( v_g \); 3) distance between object and camera \( z \); 4) camera speed value \( v_c \); 5) object type class \( T_o \).

4.2.1. Accuracy and stability of estimated speed. Here we test the accuracy and stability of our estimated method and calculate the error and variance with ground truth of speed. We respectively use a bottle and a telephone as experiment object, and also set the camera speed \( v_c \) to 0 m/s, ground truth speed \( v_g \) to approximate 0.3cm/s, table 1 shows the experiment result in 10 groups, where the first 5 ones belongs to object ‘Bottle’ and the last 5 ones belong to object ‘Telephone’.

| \( T_o \) | \( v_c \) (ms\(^{-1}\)) | \( v_g \) (ms\(^{-1}\)) | \( v_o \) (ms\(^{-1}\)) |
|---|---|---|---|
| 1 | Bottle | 0.000 | 0.305 | 0.295 |
| 2 | Bottle | 0.000 | 0.278 | 0.256 |
| 3 | Bottle | 0.000 | 0.310 | 0.278 |
| 4 | Bottle | 0.000 | 0.331 | 0.315 |
| 5 | Bottle | 0.000 | 0.341 | 0.385 |
| 6 | Telephone | 0.000 | 0.357 | 0.328 |
| 7 | Telephone | 0.000 | 0.383 | 0.351 |
| 8 | Telephone | 0.000 | 0.362 | 0.308 |
| 9 | Telephone | 0.000 | 0.352 | 0.332 |
| 10 | Telephone | 0.000 | 0.337 | 0.317 |

The average error between \( v_o \) and \( v_g \) is: 8% and the variance is 0.115. The error is mostly from: 1) the quarter view of camera causes the precision loss in distant view, 2) error of bounding box, 3) error of measurement quantities. And the variance is mostly from: 1) the float up or down of size of bounding box when object is moving, 2) the error between neighbour frames caused by the sample interval of camera.

4.2.2. Speed of object influence on method’s effectiveness Here we test if the method’s effectiveness will be disturbed by the value of speed of object or not. We take a bottle as experiment object, and set the camera to be still, and then set the ground truth \( v_g \) ranges from 0.0m/s to 2.000m/s, table 2 shows the experiment result.
### Table 2. Experiment on speed’s influence on method’s effectiveness.

| \( T_o \) | \( v_c (\text{ms}^{-1}) \) | \( v_d (\text{ms}^{-1}) \) | \( v_o (\text{ms}^{-1}) \) |
|---|---|---|---|
| 1 | Bottle | 0.000 | 0.000 | 0.000 |
| 2 | Bottle | 0.000 | 0.115 | 0.102 |
| 3 | Bottle | 0.000 | 0.363 | 0.382 |
| 4 | Bottle | 0.000 | 0.583 | 0.630 |
| 5 | Bottle | 0.000 | 0.703 | 0.615 |
| 6 | Bottle | 0.000 | 0.897 | 0.831 |
| 7 | Bottle | 0.000 | 1.512 | - |
| 8 | Bottle | 0.000 | 2.103 | - |

The estimated speed of first 6 groups basically match the ground truth, this shows the effectiveness of method is kept in slow moving status. But the last 2 groups cannot get the estimated speed, this is because the blur of image caused by high moving speed disable the YOLOv2 to catch object. Figure 6 shows the rendering of experiment where a smart car is moving in a slash.

![Figure 6. Rendering of experiment](image)

4.2.3. **Camera movement’s influence on method’s effectiveness.** Here we test if the method’s effectiveness will be disturbed by the movement of camera or not. When the smart car is moving, we also move the camera with a horizontal direction, and we compare the estimated speed and ground truth speed that labeled in image of figure 6.

![Figure 7. Three states of experiment with camera moving.](image)

- (a) State1, smart car is moving, camera does not move yet.
- (b) State2, both smart car and camera are moving.
- (c) State 3, camera is still moving, smart car stops.
From figure 7 we know that the estimated speed of all three states basically match the ground truth speed, which illustrates the method is still valid when camera is also moving.

5. Conclusion
Based on above analysis and experiments, the method mentioned in this paper is able to estimate the true speed of different types of object with a normal camera or video at a real-time speed, even when the observer itself is moving. This method can be used in fields like road accident analysis, players’ motion analysis in count and so on.

Acknowledgments
The research was supported in part by Shenzhen Science and Technology Program under Grant Nos. JCYJ20160330163900579 and JCYJ20160413163534712.

References
[1] Girshick, Ross, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]// IEEE conference on computer vision and pattern recognition, Columbus, OH, USA, 2014: 580-587.
[2] Girshick R. Fast r-cnn[C]// IEEE international conference on computer vision, Santiago, Chile, 2015: 1440-1448.
[3] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]// European conference on computer vision. Springer, Cham, 2016: 21-37.
[4] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]// IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016: 779-788.
[5] Dosovitskiy A, Fischery P, Ilg E, et al. FlowNet: Learning Optical Flow with Convolutional Networks[J]. 2015:2758-2766.
[6] Lin H Y and Chang C H. Automatic speed measurements of spherical objects using an off-the-shelf digital camera[C]// IEEE International Conference on Mechatronics, Beijing, China, 2005:66-71.
[7] Yu Y Y, Li X Y and Zhao Y T. A method to calculate the macroscopic optical flow speed of vehicle[C]// IGTA, Beijing, China, 2009:20-23.
[8] Wang J. A method to recognize vehicle and to measure its speed based on video[D]. Northern nationalities university, 2009.
[9] Yan L P, Sun C X, Wang X L, et al. A method to real-time measure speed based on vehicle-mounted camera. CN, CN103686083A[P]. 2014.
[10] Harvey L and Cohen A. Vehicle speed measurement using an imaging method[C]// International Conference on Industrial Electronics, Control and Instrumentation, Kobe, Japan, 1991: 1730-1733.
[11] Pumrin S and Dailey D J. Roadside camera motion detection for automated speed measurement[C]// IEEE Conference on Intelligent Transportation Systems, Singapore, 2002: 147-151.
[12] Lin H Y and Chang C H. Automatic speed measurements of spherical objects using an off-the-shelf digital camera[C]// IEEE International Conference on Mechatronics, IEEE, 2005:66-71.
[13] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[M]. ArXiv preprint ArXiv:2016.612.08242.