Vehicle Speed Estimation with Non-stationary Camera

Gaofeng Su

Abstract

Vehicle speed estimation plays a vital role in traffic management, data analysis, and Intelligent Transportation Systems (ITS). While traditional speed sensing systems, such as radar and laser-based technologies, offer high accuracy, their hardware costs remain a significant barrier. In contrast, vision-based approaches are gaining attention due to their relatively simple and cost-effective hardware requirements. However, most existing vision-based methods rely on stationary cameras, limiting their flexibility and the scalability of vehicle speed monitoring networks.

This paper proposes a simple, but effective homography-based approach for vehicle speed estimation using a non-stationary platform. The method utilizes vehicle keypoints detection and computes the homography matrix. By warping the 2D velocity vector obtained from the optical flow method, the actual vehicle speed is estimated with high accuracy. This approach offers a flexible and cost-efficient solution for vehicle speed monitoring.

1. Introduction

Vehicle speed monitoring plays a crucial role in traffic management, as it can improve road safety. According to the San Francisco Municipal Transportation Agency (SFMTA), San Francisco is introducing the Speed Safety Cameras Program to enhance road safety [1]. Additionally, with the growing interest in Intelligent Transportation Systems (ITS), on-road vehicle speed estimation can provide valuable traffic speed data for further analysis by ITS.

Traditional speed camera monitoring systems typically rely on expensive hardware such as radar, lidar, and vision cameras, which makes scaling a speed safety camera network challenging. With modern CPUs becoming more powerful and the advancement of computer vision technology, lots of research has been conducted on vision-based vehicle speed estimation methods [2]. However, most of these vision-based method are designed to works on stationary speed cameras. Relying on a stationary camera reduces the flexibility and scalability of the vehicle speed monitoring networks since the cost of the speed camera system is still relatively high.

One idea is to use non-stationary platforms such as phone cameras, which are low-cost and easy to set up to capture videos, to perform vehicle speed estimation. This not only introduces a new form of speed-sensing modality but also makes this sensing network more flexible and scalable.

Traditionally, use the camera to perform vehicle speed estimation, one key component is the camera parameters (i.e., the intrinsic and extrinsic matrix)[2]. For stationary cameras, this parameter can be calibrated offline, but this is tricky for non-stationary cameras as the extrinsic matrix changes over time and is hard to calibrate in real-time. Research has been done using sophisticated pipelines such as reconstructing 3D bounding boxes of vehicles to estimate speed. This paper proposed a simple method based on homography and warping technique but could yield relatively high accuracy. The paper is arranged as follows: Section 2 explains the methods and processing pipeline, section 3 will briefly describe the experiment setup section 4 is about results discussion, and Section 5 is the conclusion and future work.

2. Methodology

2.1. Relating Vehicle Speed with Homography

The key to the proposed method is that two images can be related by homography if the object/scene being captured lies on a single plane, even if the camera is not at the same center of projection (COP). Note that this observation still holds even if the extrinsic matrix of the camera changes.

This approach is commonly used in many stationary camera calibration methods. Typically, people either use a known planar object to calibrate the camera or assume that the road is a plane and utilize known road features, such as traffic lines, for calibration. The challenge arises in scenarios involving a hand-held camera, where obtaining a clear view of the road surface can be quite difficult. Furthermore, this method relies on well-maintained roads to ensure the visibility of these features, which might not always be realistic.

However, we can approach this problem from a different

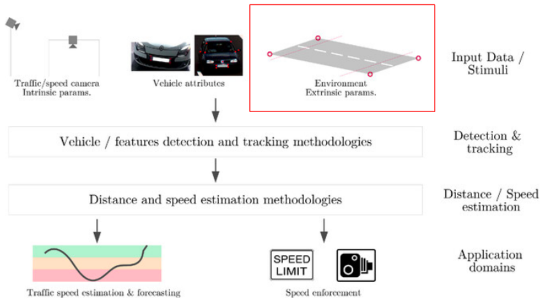


Figure 1. Common speed estimation pipeline, image adapted from [2]

.

Figure 2. Extract road features to find parallel lines, horizon etc. image adapted from [6]

.

perspective. Instead of relying on the road surface, we can approximate the side of the vehicle as a single plane. Since we are estimating the vehicle's speed, we must observe it from our point of view, which makes it more likely that we will see the side of the vehicle.

Using homography and warping, we can compute the velocity derived from optical flow. However, this approach relies on one key assumption: the vehicle's speed must be aligned with the plane of its side. Below is a less formal explanation:

Denoted a vehicle keypoint on image k as s_k, the same keypoint on image $k+1$ as s_{k+1}, let the matching keypoint be s, and the computed homography matrix to be H_k and H_{k+1} respectively. We have:

$$w_k s = H_k s_k \tag{1}$$

$$w_{k+1} s = H_{k+1} s_{k+1} \tag{2}$$

The speed vector from the optical flow is:

$$v = \frac{1}{dt}(s_{k+1} - s_k) = \frac{1}{dt}(H_{k+1}^{-1} w_{k+1} s - H_k^{-1} w_k s) \tag{3}$$

Since the speed vector and the homography matrix are computed independently, there is a constraint on the resulting scaling factors w_{k+1} and w_k. As the scaling factor controls the depth of the points, and the optical flow tracks the surface of the vehicle, this constraint implies that the actual speed must be aligned with the plane of the vehicle's side to correctly use the homography for warping the speed vector derived from optical flow.

2.2. System Pipeline

Based on this idea, the following pipeline is proposed (see Figure 3). The input to the system consists of a video stream and a predefined vehicle keypoint template with known dimensions.

The first step is to compute the dense optical flow map. We used OpenCV's implementation of Gunnar Farneback's dense optical flow algorithm [5]. Simultaneously, a keypoint detector identifies vehicles and detects keypoints in the current image. These keypoints are then matched with the predefined template to extract only the side plane keypoints. If more than four keypoints are available, the homography matrix can be computed and used to transform the speed vector at these keypoints. Ideally, the transformed speed vectors should all point horizontally (i.e., zero on the vertical axis). After obtaining the speeds of all observed keypoints, the final vehicle speed is calculated as the average of these keypoint speeds.

2.3. About Vehicle Template and Detection Model

Machine-learning-based keypoint detection models have proven to be robust in detecting vehicle keypoints with "semantic meaning," such as the "center of front/rear wheels" or the "corners of windows." However, developing a universal model that accommodates all vehicle types with realistic mechanical dimensions remains challenging. In this paper, we use the OpenPifPaf car keypoint detection model [3], which is based on a 66-keypoint sedan template (see Figure 4).

3. Experiment

To evaluate the proposed method and pipeline, we collected four hand-held camera recordings and performed speed estimation offline. The videos were recorded at 60 fps with a resolution of 2160 × 3840 pixels. To accelerate

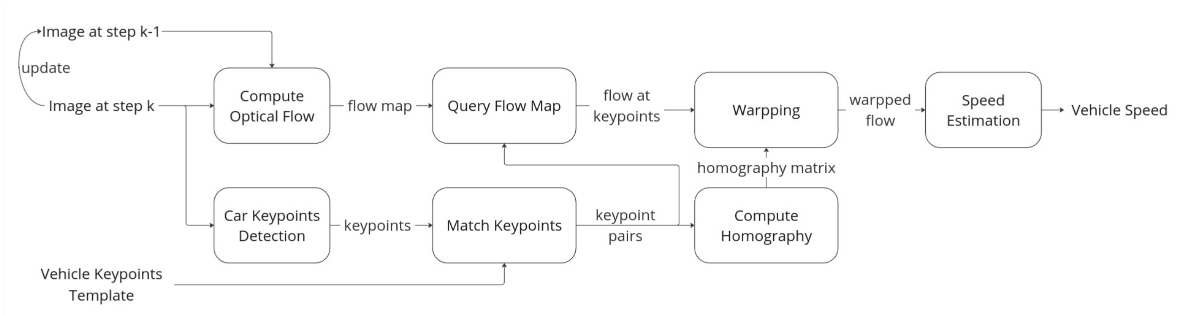


Figure 3. Proposed system pipeline

.

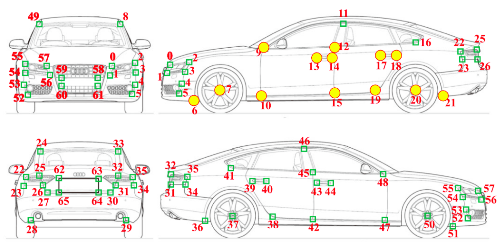


Figure 4. 66-keypoints sedan model, image adapted from [4] . The target keypoint is highlighted in yellow.

processing, the resolution was reduced to 540 × 960 pixels. Each video is approximately 3 seconds long. A speed gun was also used to measure the actual vehicle speed, providing ground truth data. Figure 5 illustrates the estimated speed and the transformed speed vector for Vehicle 1.

Below is a comparison between the estimated vehicle speed from hand-held camera footage and the speed gun measurements.

4. Result and Disccusion

The experiments of estimating speed of vehicles 1-3 demonstrate that the proposed method has relatively high accuracy. While the estimation for Vehicle 3 may seem inaccurate at first glance, a careful analysis of the footage revealed that Vehicle 3 was slowing down due to the traffic signal turning red. Therefore, although we could not provide a continuous ground truth, this result is reasonable in this context.

Another observation is that, after some time, the estimated speed dramatically changed, yielding incorrect results. This occurred because, as the car drove further from the camera, the viewing angle to the side of the vehicle became very narrow, making it difficult to detect many features. The features that were detected became nearly collinear, leading to singularity or near-singularity issues when solving the homography matrix, as illustrated in Figure ??. This limitation of the proposed method could potentially be mitigated by adding more keypoints (e.g., additional keypoints on the wheels).

5. Conclusion and Future Work

This paper proposes a simple and effective method to estimate vehicle speed on a non-stationary platform by treating the side of the vehicle as a single plane, matching keypoints to a predefined template, and transforming the speed vector from optical flow. The hand-held camera experiments demonstrate that the proposed method can accurately measure speed when ”good features” of the vehicle are detected. One limitation is that, as the vehicle moves away from the camera and the viewing angle to the side of the vehicle narrows, the estimation becomes unreliable. This could potentially be mitigated by adding more keypoints to the vehicle model.

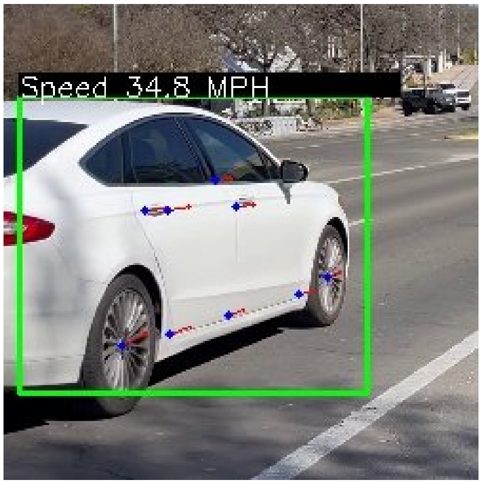
Currently, we are using a pre-trained sedan keypoint model. However, this can be improved by training a keypoint detection model tailored to specific types of vehicles, enabling us to detect and estimate the speed of not only sedans but also SUVs, trucks, and buses. Additionally, customizing the keypoint model by adding more keypoints, such as those on the wheels, might make the method more robust against poor viewing angles.

References

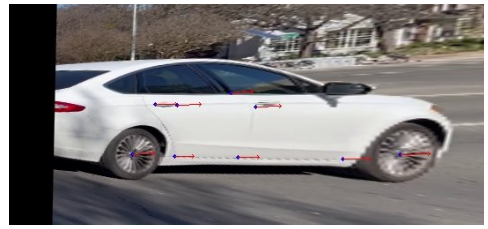
[1] San Francisco Municipal Transportation Agency: ”Why we’re introducing speed safety cameras, a first for California,” accessed Nov 21, 2024. [Online]. Available: https://www.sfmta.com/blog/why-were-introducing-speed-safety-cameras- first-california

[2] David Fernández Llorca et al.: ”Vision-based vehicle speed estimation: A survey,” 2021.

[3] OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association, Sven Kreiss et al. (2021)



(a) Estimated speed.



(b) Transformed image and the speed vector.

Figure 5. Speed estimation example of vehicle 1. Speed vectors are marked by red arrow, and keypoints are marked in blue circles.

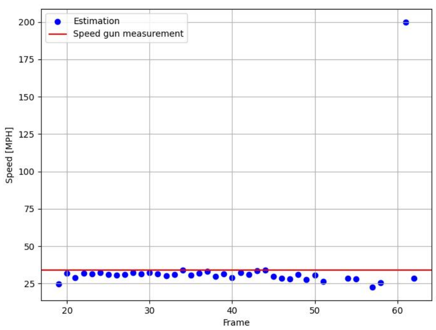


Figure 6. Estimated speed of vehicle 1 in footage recorded by a hand-held camera.

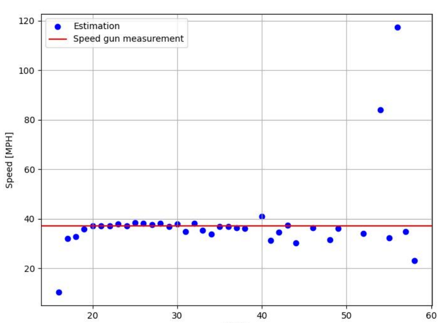


Figure 7. Estimated speed of vehicle 2 in footage recorded by a hand-held camera.

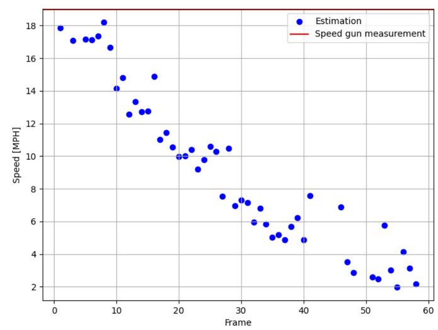


Figure 8. Estimated speed of vehicle 3 in footage recorded by a hand-held camera.

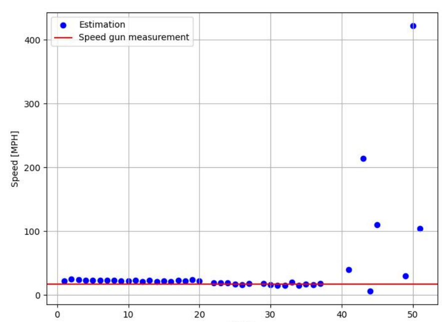


Figure 9. Estimated speed of vehicle 4 in footage recorded by a hand-held camera.

[4] ApolloCar3D: A Large 3D Car Instance Understanding Benchmark for Autonomous Driving, Xinbin Song et al. (2019)

[5] OpenCV, https://docs.opencv.org/3.4/dc/d6b/group$_{video_{track.html}}$ga5d10

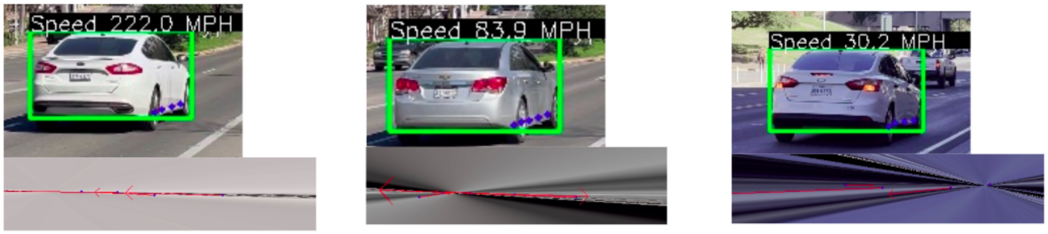


Figure 10. Poor viewing angle makes the detector only detect keypoints that are co-linear and introduce singularity issues of the homography matrix. The top row of images, shows the detected keypoints, as you can see they are forming a straight line. The bottom row shows the transformed image, and you can see the speed vector is randomly oriented.

[6] Automatic Roadside Camera Calibration with Transformers, Yong Li et al., 2023