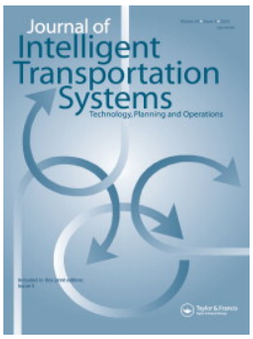

Taylor & Francis
Taylor & Francis Group

Journal of Intelligent Transportation Systems
Technology, Planning, and Operations

ISSN: 1547-2450 (Print) 1547-2442 (Online) Journal homepage: www.tandfonline.com/journals/gits20

Capturing the true bounding boxes: vehicle kinematic data extraction using unmanned aerial vehicles

Tian Mi, Dénes Takács, Henry Liu & Gábor Orosz

To cite this article: Tian Mi, Dénes Takács, Henry Liu & Gábor Orosz (2025) Capturing the true bounding boxes: vehicle kinematic data extraction using unmanned aerial vehicles, Journal of Intelligent Transportation Systems, 29:5, 566-578, DOI: 10.1080/15472450.2024.2341395

To link to this article: https://doi.org/10.1080/15472450.2024.2341395

JOURNAL OF INTELLIGENT TRANSPORTATION SYSTEMS
2025, VOL. 29, NO. 5, 566–578
https://doi.org/10.1080/15472450.2024.2341395

Capturing the true bounding boxes: vehicle kinematic data extraction using unmanned aerial vehicles

Tian Mi[a,b], Dénes Takács[c], Henry Liu[a], and Gábor Orosz[a,b]

[a]Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI, USA; [b]Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, USA; [c]Department of Applied Mechanics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics, Budapest, Hungary

ABSTRACT

This paper presents a methodology by which kinematic variables of road vehicles can be extracted from unmanned aerial vehicle (UAV) footage. The oriented bounding boxes of the vehicles are identified based on the aerial view of the intersection, and the kinematic variables, such as position, longitudinal velocity, lateral velocity, yaw angle and yaw rate, are determined. The bounding boxes are converted to the perspective of a roadside camera using homography, to generate labeled data sets for training the machine learning-based perception systems of smart intersections. Compared to ordinary GPS data-based technology, the proposed method provides smoother data and more information about the dynamics of the vehicles. In the meantime, it does not require any additional instrumentation on the vehicles. The extracted kinematic variables can be used for motion prediction of road traffic participants and for control of connected automated vehicles (CAVs) in intelligent transportation systems.

ARTICLE HISTORY
Received 21 July 2023
Revised 15 March 2024
Accepted 6 April 2024

KEYWORDS
data sets for machine learning algorithms; kinematic variables; unmanned aerial vehicles; vehicle tracking; video processing

1. Introduction

Unmanned aerial vehicles (UAVs), also referred as drones, have attracted increasing attention of researchers in traffic monitoring and management due to their mobility and low cost. Instead of fixed-location sensors, such as fixed cameras, radars, and loop detectors, which can only collect data from specific perspectives at specific locations, UAVs can serve as mobile sensors in modern traffic networks. UAVs can be coordinated to collect large scale traffic data and the extracted vehicle trajectories can be used to investigate safety (Zheng et al., 2022) and to study traffic congestion (E. Barmpounakis & Geroliminis, 2020). Drones can carry different types of sensors, such as video cameras, thermal cameras, infrared cameras, LIDAR, and radar (Pajares, 2015). Among these, high-resolution video cameras are the most popular sensors for traffic monitoring (Datondji et al., 2016; Husain et al., 2020). Probe vehicles equipped with high-precision GPS can also be used to collect traffic data, but such instrumentation is expensive and only a small fraction of the vehicles can be instrumented. Besides traffic surveillance, UAVs can also be used in other applications, for example, freight delivery and road construction, but this is beyond the scope of this paper. The survey by Barmpounakis et al. (2016) is listed for the interest of the readers.

Puri (2005) gives a comprehensive survey of early research activities on using UAVs for traffic surveillance. Back then most of the work was still in the design stage, and mainly focused on the control and operation of UAVs. The related research between 2005 and 2012 is summarized in (Kanistras et al., 2013), with real-time algorithms developed for vehicle detection, classification and tracking in intelligent transportation systems. Liu et al. (2013) review computer vision techniques used for vehicle detection and tracking. Algorithms are categorized as motion-based methods (which compare frames and search for differences or trace the movements of pixels) and feature-based methods (which use inherent vehicle features such as colors and edges). Tracking algorithms mostly follow the motion-based method; namely, vehicles that are spatially close in consecutive frames can be identified as the same vehicle. Contrarily, machine

CONTACT Tian Mi tianm@umich.edu Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI 48109, USA

learning (ML) algorithms use feature-based vehicle detection.

Recent advances in ML technologies bring traffic monitoring to a new stage. Won (2020) provides a survey on vehicle classification with different sensors where ML techniques play an important role. Bouguettaya et al. (2022) discuss vehicle detection from UAV images using different deep learning methods. Compared to hand-crafted image processing or shallow learning methods, deep learning enhances the ability of real-time, accurate vehicle detection. Especially, convolutional neural networks (CNNs), such as You Only Look Once (YOLO) (Redmon et al., 2016) and its variants, exhibit a lot of potential in this sense, see e.g., Liu et al. (2023). CNNs are often combined with other neural network structures, for example, recurrent neural networks (RNNs), long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997), and generative adversarial networks (GANs) (Goodfellow et al., 2020). RNNs feed the output from the previous state as an input of the current state, and thus, can be used for vehicle tracking. LSTMs fix some problems (such as vanishing gradient) of classical RNN methods. GANs can be used to generate new images which are similar to the ones in the existing data sets. A major challenge for learning-based methods is however the lack of labeled data sets. Namely, these methods rely on classical (hand-crafted) image-processing techniques in the training stage.

Meanwhile, classical computer vision methods are also applied to UAV images in order to study traffic. Braut et al. (2012) use a hovering drone at an intersection to recover origin-destination (OD) matrices. In order to reduce the large vibration of the drone, they use homography to transfer each video frame to a reference image. Guido et al. (2016) present a method to extract vehicle positions and speeds from UAV videos, and the results are compared with GPS data. To ensure accuracy, it uses ground control points to match the image with the ground coordinates. Khan et al. (2017) propose an automated framework to extract vehicle trajectories from UAV-obtained video footage more efficiently. Kaufmann et al. (2018) use UAV observations to study the moving synchronized flow patterns in downtown areas. Chen et al. (2021) develop an ensemble detector for vehicle detection, use a kernelized correlation filter for vehicle tracking, and convert the vehicle positions from Cartesian coordinates to Frenet coordinates along the roadway.

Regardless of the applied methodologies, the main goal of the above-mentioned studies is the use of UAV videos for vehicle detection, classification, and tracking, in order to monitor traffic, study traffic flow, and calibrate traffic simulations. The extracted information typically contain vehicle types and counts, vehicle positions and speeds, and traffic flow speeds. For these purposes, vehicles are treated as points and their orientations are ignored. There are a few data sets and algorithms (Bock et al., 2020; Liu & Mattyus, 2015; Razakarivony & Jurie, 2016; Xia et al., 2018; Zheng et al., 2022) which consider the orientations (the bounding boxes) of vehicles. However, the purpose of these research is vehicle (or object) detection and classification, and not the extraction of vehicle kinematic variables. For example, the inD data set (Bock et al., 2020) provides bounding boxes of different types of road users with orientation information, but each object is treated as a single point (centroid of the bounding box) instead of a rigid body. While such trajectory data may be sufficient for many transportation applications, the details of vehicle kinematics are ignored in prior research.

In the coming era of connected and automated vehicles (CAVs) and intelligent transportation systems (ITSs), knowing the kinematic variables of both the automated vehicles and the surrounding human-driven vehicles are crucial for motion planning and vehicle control. Instead of on-board sensors of limited range, and fixed-location sensors aimed for aggregated data, in this paper, a UAV is used to extract kinematic variables. As illustrated in Figure 1, the main contributions of this research are:

1. it develops an algorithm to determine true bounding boxes with high precision from UAV footage, and it generates vehicle kinematic data including position, velocity, yaw angle and yaw rate;
2. it establishes a methodology by which labeled data sets can be generated for training the machine learning-based visual perception systems of roadside-view cameras; for which the auto-generated roadside-view bounding boxes can serve as ground truth data.

The rest of the paper is organized as follows. Section 2 introduces the experimental setup. Section 3 explains the image processing procedure, the data extraction from the processed video, and compares the results with data obtained from a GPS-equipped vehicle. In Section 4, the bounding boxes obtained from the drone view camera are converted to the roadside-view camera image using homography. Section 5 concludes the paper and lays out future research directions.

2. Experiment setup

In order to show the capabilities of our methodology, an experimental demonstration is shown in our paper.

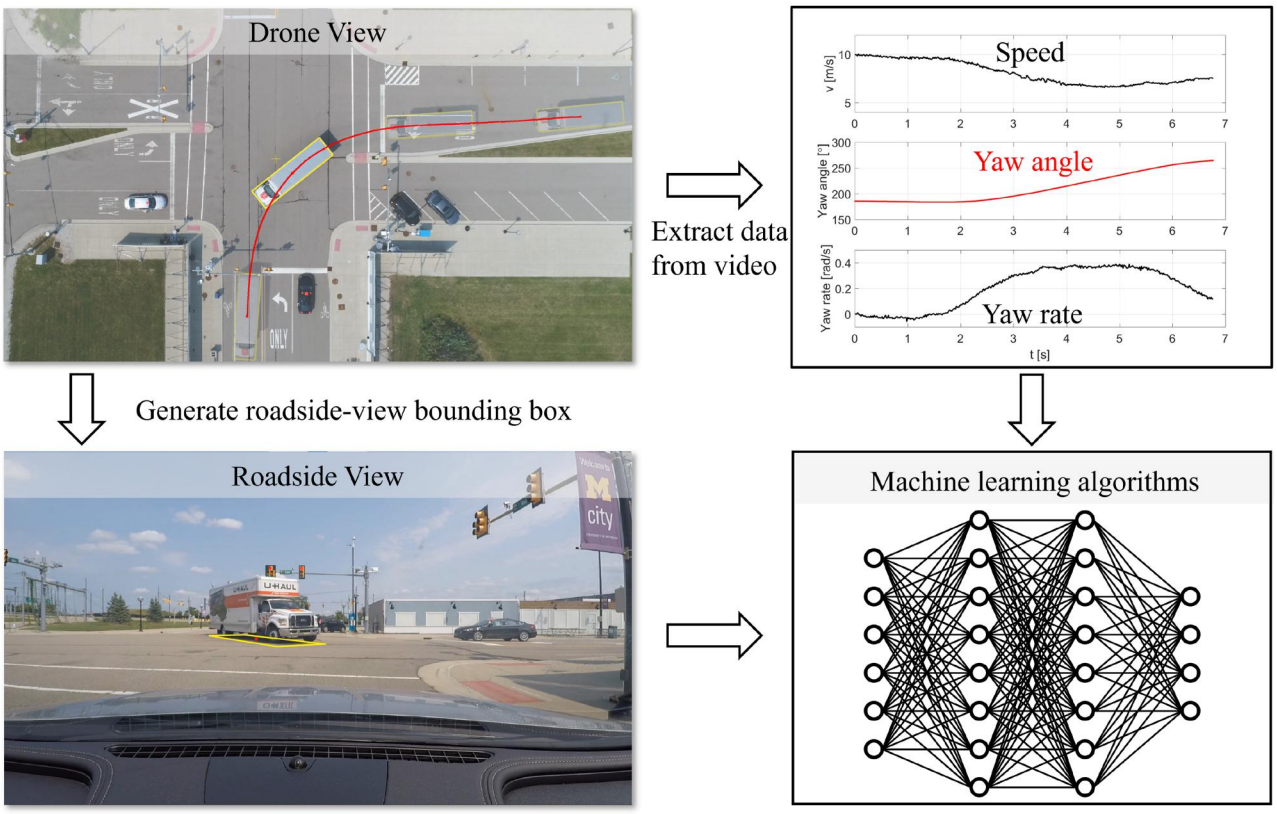

Figure 1. Using UAVs to track vehicles and to facilitate machine learning algorithms.

The experiments were carried out at the Mcity Test Facility at the University of Michigan, Ann Arbor. A classical traffic scenario was emulated in the intersection of State Street (north-south) and Main Street (east-west), as shown in Figure 2.

The experiment is designed as follows. A truck of total length 10.5 m moves toward the intersection along the westbound of Main Street, makes a left turn to State Street, and leaves the intersection. At the same time, a car is standing at the eastbound of intersection with a fixed camera facing forward. This camera plays the role of a roadside camera in the experiments. The truck has an onboard GPS of 10 Hz sampling frequency installed on top of the cabin. There are three other vehicles parked at the intersection to imitate a real urban environment. A DJI Phantom 4 Pro drone equipped with a video camera with a 3-axis gimbal, capable of recording 60 frames per second (fps) in 4K resolution, is sent above the intersection and hovers about 75 m (about 250 ft) high. The camera is facing down to the intersection in order to record the movements of vehicles. The movement of the truck is captured by both the drone camera (drone view) and the car camera (roadside view).

3. Data generation

As detailed below, the data extraction process is divided into two parts: image processing and data extraction. The data extracted from images are also compared with GPS data.

3.1. Image processing

The image processing procedure is illustrated in Figure 3. Prior to detection, the first step is the stabilization of the image to compensate for the motion of the drone during the experiments. A background image (drone view) of the intersection is selected when the target vehicles are not present, and it is defined as the region of interest (ROI); see the red frame in Figure 3. During the image processing, the ROI is identified in each frame and the unnecessary part of the image is cropped. This not only improves the computation time by reducing the size of the image, but also stabilizes the video obtained from the hovering drone.

The second step is to detect moving objects by comparing each frame of the video with the background. Setting a threshold, the comparison results in a binary image, and the Matlab morphological operations are

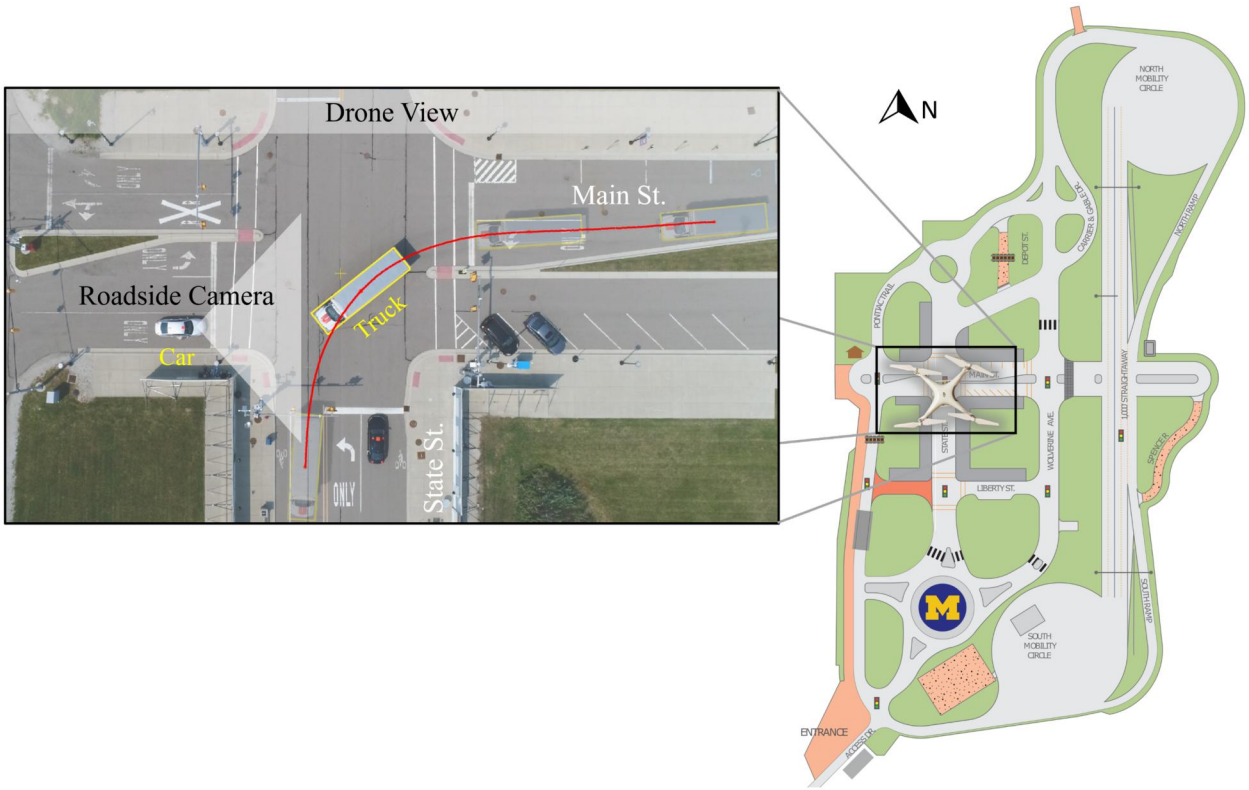

Figure 2. Intersection at mcity Test facility. A DJI phantom 4 pro drone with high precision video camera is hovering about 250 feet above the intersection to record the movements of ground vehicles. A truck arrives from the east, makes a left turn, and leaves the intersection to the South; while a car is standing at the west side of the intersection with a camera facing to the east.

applied to eliminate the small changes of the image (due to shadows, leaves moving in the wind, and changes of camera perspective) and to merge the neighboring patches which belong to the same object. We use the Matlab function "bwmorph" to apply morphological operations on the binary images. Other functions, such as "regionprops" which measures the properties of image regions, are also used to facilitate the image processing procedure. The outcomes of this step are the detection boxes (marked as green squares in Figure 3), and the number of objects. Note that the detection boxes are square-shaped as they do not consider the orientations of the vehicles, and they can be in different sizes according to the actual vehicle sizes. The goal of this step is to find the rough locations of the vehicles and to prepare for precise vehicle detection.

Here, we remark that the parked vehicles are not detected since they are also located in our background image. Namely, they are not in the interest of our experiment. Of course, as long as a vehicle is not in the background image, even if it is parked, it can still be detected by our algorithm. The background image can be updated to accommodate larger changes during the day (or different times of the year), but we omit these since the time of interest is less than a minute in this research.

With the detection boxes obtained from the previous step, precise vehicle detection is applied to obtain the bounding boxes (marked as yellow rectangles in Figure 3) containing the precise location and orientation (yaw angle) of each vehicle. Given the drone-view image of each vehicle, the area within the detection box is compared with the vehicle images rotated by different angles, and the most correlated one is selected to draw the yellow bounding box. We use the Matlab functions "imrotate" and "normxcorr2" to rotate the images and to compute the correlations between images. This way, the position and yaw angle of each vehicle can be determined. The last step is to convert the image coordinates to the ground coordinates, and to calculate velocity and yaw rate. The details of this procedure are given in the next subsection.

3.2. Data extraction

In this part, the data obtained from the video processing is discussed and analyzed. The vehicle is considered as a rigid body, and the position, velocity and heading angle at different points of the vehicle are calculated. Furthermore, the bicycle model is introduced to facilitate the analysis.

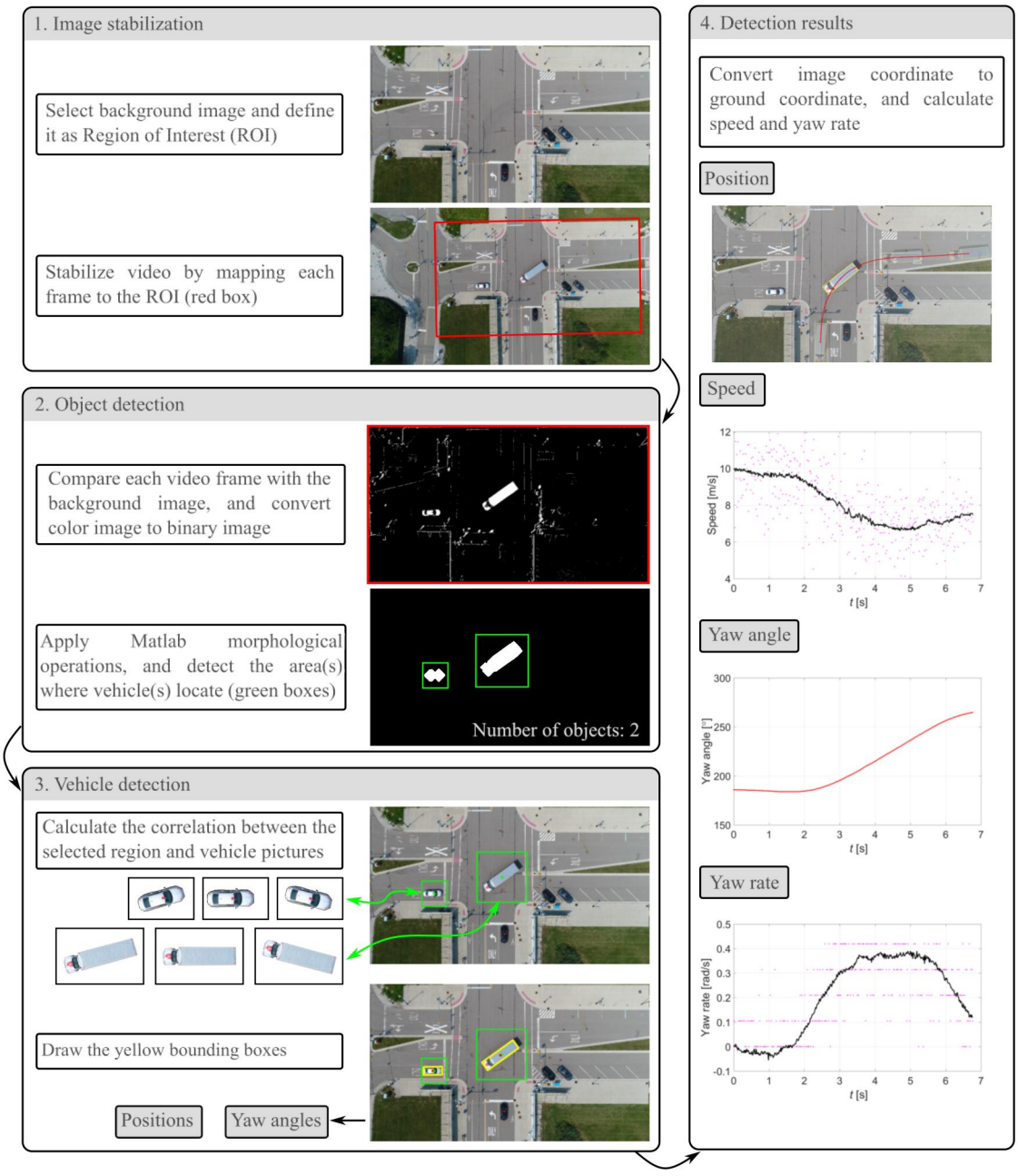

Figure 3. Vehicle detection process.

Taking the truck as an example (see Figure 4), Point C is the geometric center of the bounding box, point T marks the location on the cabin where the GPS antenna is installed, and point R is the center of the real axle. The position vector

$$\mathbf{r}_{\mathrm{C}} = \begin{bmatrix} x_{\mathrm{C}} \\ y_{\mathrm{C}} \end{bmatrix}, \tag{1}$$

of center point C and the yaw angle ψ of the vehicle can be directly obtained from the bounding box after converting the image coordinates to the ground coordinates. The distance between point T and center point C is $d_{\mathrm{CT}} = 3.49$ m, while the distance between center point C and rear axle center point R is $d_{\mathrm{CR}} = 2.21$ m. Thus, the positions of point T and point R can be calculated as:

$$\mathbf{r}_{\mathrm{T}} = \begin{bmatrix} x_{\mathrm{T}} \\ y_{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} x_{\mathrm{C}} \\ y_{\mathrm{C}} \end{bmatrix} + d_{\mathrm{CT}} \begin{bmatrix} \cos\psi \\ \sin\psi \end{bmatrix},$$
$$\mathbf{r}_{\mathrm{R}} = \begin{bmatrix} x_{\mathrm{R}} \\ y_{\mathrm{R}} \end{bmatrix} = \begin{bmatrix} x_{\mathrm{C}} \\ y_{\mathrm{C}} \end{bmatrix} - d_{\mathrm{CR}} \begin{bmatrix} \cos\psi \\ \sin\psi \end{bmatrix}. \tag{2}$$

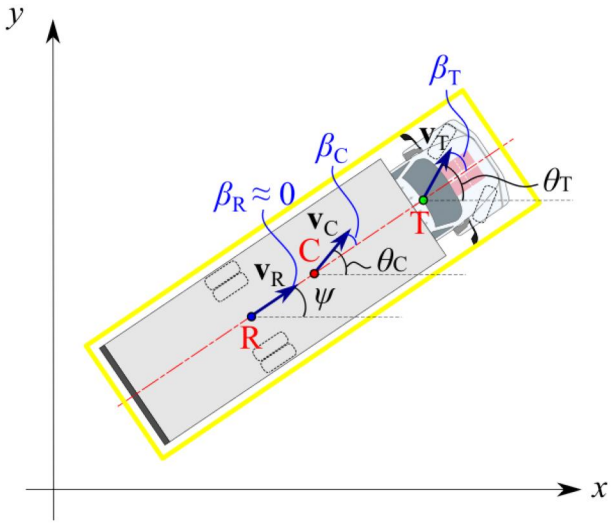

Figure 4. Bicycle model at the truck.

Knowing the position information at each time step, we can estimate the velocity of point C as:

$$\mathbf{v}_C = \begin{bmatrix} \frac{\Delta x_C}{\Delta t} \\ \frac{\Delta y_C}{\Delta t} \end{bmatrix}, \tag{3}$$

where Δx_C and Δy_C are the changes of the x and y coordinates between frames and $1/\Delta t$ is the frame rate. Then the speed v_C, which is the magnitude of the velocity, is given by:

$$v_C = \frac{\Delta r_C}{\Delta t}, \tag{4}$$

where $\Delta r_C = \sqrt{\Delta x_C^2 + \Delta y_C^2}$. The velocity and speed can be defined similarly for points R and T. Figure 4 shows the velocity vectors $\mathbf{v}_T$, $\mathbf{v}_C$, and $\mathbf{v}_R$ of points T, C, and R, respectively.

However, a direct division by Δt in Eqs. (3) and (4) can lead to large errors. One pixel from the image corresponds to about 0.0275 m in reality, while the time difference is $\Delta t = 1/60$ seconds (with a frame rate of 60 fps). With these, even an error of one pixel can lead to $0.0275 \times 60 = 1.65$ m/s when estimating the speed. Thus, rather than computing the speed from the raw position data, we first smooth the position using a moving average. Figure 5(a) shows the speed obtained when using a moving average over 21 data points, which corresponds to a 1/3 s window (1/6 s ahead and 1/6 s behind). Note that this parameter can be tuned based on the camera speed (fps). One-sided smoothing, which only uses past information, can also be adapted when applying the algorithm online. When the vehicle is moving straight toward the intersection at the beginning of the trip, for about 2 s, the speed at the three points (T, C and R) are very close. However, when the vehicle starts to turn, the differences become obvious; point T at the front of the vehicle has the largest speed.

The longitudinal and lateral velocities are the velocity components along the vehicle's symmetry axis and perpendicular to it. For example, for point C, these can be calculated as

$$\begin{aligned} v_C^{\text{lon}} &= \frac{\Delta x_C}{\Delta t}\cos\psi + \frac{\Delta y_C}{\Delta t}\sin\psi, \\ v_C^{\text{lat}} &= -\frac{\Delta x_C}{\Delta t}\sin\psi + \frac{\Delta y_C}{\Delta t}\cos\psi. \end{aligned} \tag{5}$$

These are also computed after smoothing the position and the yaw angle using a 21-point moving average. In Figure 5(b), the black curve is the longitudinal velocity and the green, red, blue curves are the lateral velocities for the points T, C, and R, respectively. All three points have the same longitudinal velocity since the vehicle is considered as a rigid body (i.e., the length of the vehicle does not change in time) and the bounding box also has a fixed length.

The heading angle θ of a point is given by the direction of the velocity at that point. For example, at the center C of the vehicle, the heading angle θ_C can be defined as the angle between the velocity $\mathbf{v}_C$ given in (3) and the horizontal axis:

$$\theta_C = \arctan\frac{\Delta y_C}{\Delta x_C}, \tag{6}$$

see Figure 4. One may observe that heading angle θ is given by the sum of yaw angle ψ and slip angle β, where the latter one characterizes the direction of the velocity with respect to the symmetry axis of the vehicle. For example, for point C we have $\theta_C = \psi + \beta_C$. Substituting this into Eq. (6) and utilizing trigonometric identities we obtain

$$\beta_C = \arctan\frac{-\Delta x_C \sin\psi + \Delta y_C \cos\psi}{\Delta x_C \cos\psi + \Delta y_C \sin\psi} = \arctan\frac{v_C^{\text{lat}}}{v_C^{\text{lon}}}, \tag{7}$$

where in the last step we exploited the expressions Eq. (5) for the longitudinal and lateral velocities. That is, zero lateral velocity corresponds to zero slip angle.

Figure 4 demonstrates the heading and slip angles in a kinematic bicycle model, where the lateral velocity (and consequently the slip angle) at the rear axle center R is considered to be zero. In other words, the yaw angle can be approximated by the heading angle at point R. Figure 5(c) depicts the heading angles calculated by the velocity directions at points T, C, and R. For comparison, the yaw angle ψ obtained from the bounding box is also plotted as a magenta curve.

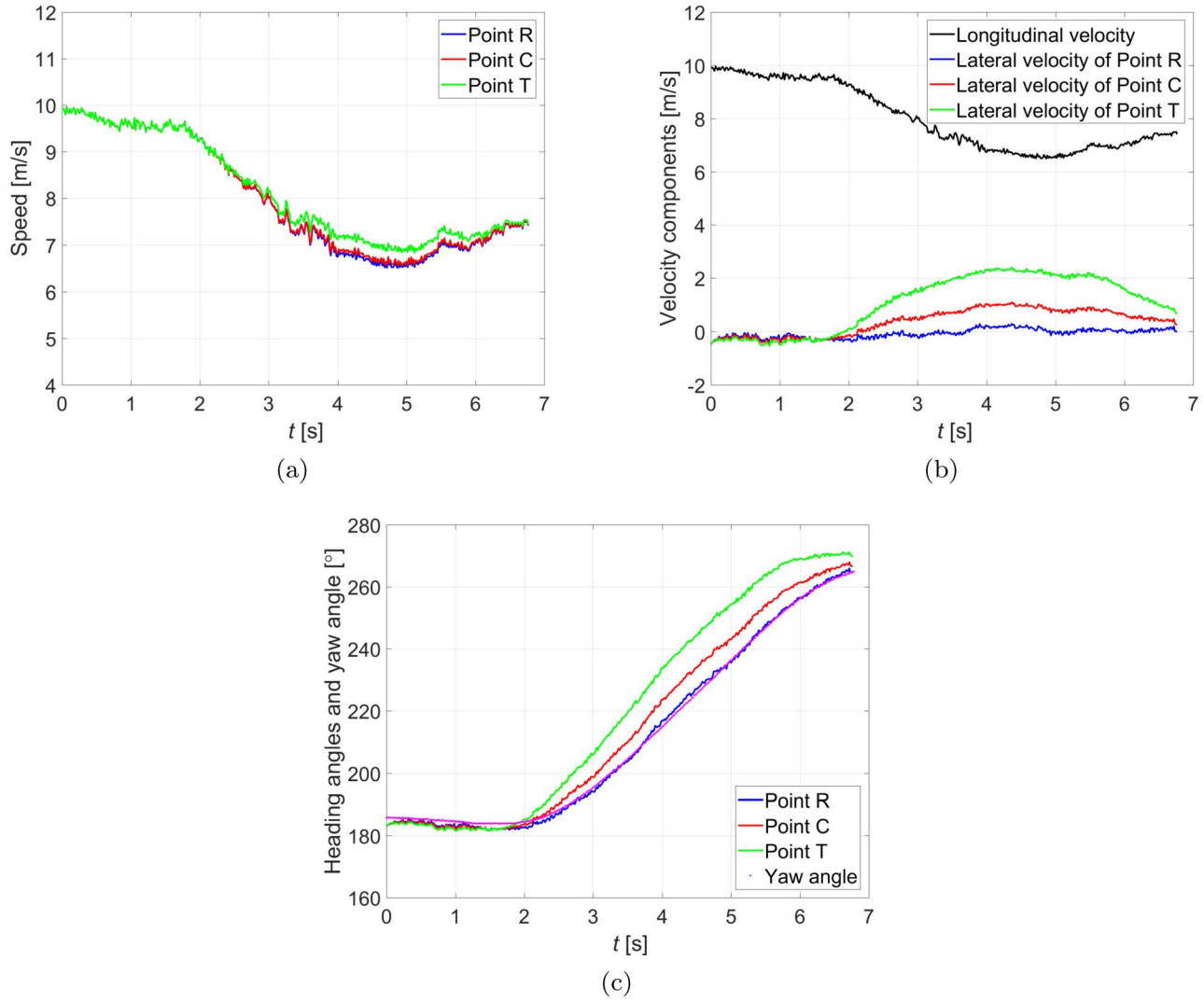

Figure 5. Data extraction from the truck bounding box at different points: (a) speeds; (b) longitudinal velocity and lateral velocities; (c) heading angles and yaw angle.

Point T has the largest heading angle, and the heading angle at point R is very close to the yaw angle ($\theta_R \approx \psi \iff \beta_R \approx 0$). Correspondingly, in Figure 5(b) the lateral velocities are approximately zero at all three points during the first 2 s. When the turn starts, the lateral velocities at point T and point C increase to different magnitudes while it remains close to zero at point R. This shows that there is only a small side slip at the rear wheels, and validates the assumption of the kinematic bicycle model.

The yaw rate can be calculated using the yaw angle of the vehicle as:

$$\omega = \frac{\Delta\psi}{\Delta t} \tag{8}$$

The smoothed curves of speed and yaw rate and their original data are plotted in Figure 3. We emphasize that the yaw rate is calculated as the derivative of the yaw angle and not as the derivative of a heading angle. The heading angle depends on which point of the rigid body we consider. Meanwhile, there is a unique yaw angle, and consequently, a unique yaw rate, defined for the rigid body.

We also analyze the obtained data of the standing vehicle (car), as the speed and yaw rate are known to be 0. The standard deviation of the position components at the center point in the x and y directions are within 0.03 m. After smoothing the position data, the mean values of the computed velocity components of the center point in the x and y directions are both below 0.02 m/s, while the standard deviations are within 0.09 m/s. The standard deviation of the yaw angle is about 0.002 rad. The smoothed yaw rate has a mean value very close to 0 (e-17) rad/s, with a standard deviation of 0.01 rad/s.

3.3. Comparison to GPS data

The detection results from the drone view video (60 Hz) are compared with the GPS data (10 Hz). The GPS antenna is installed at point T of the truck (see Figure 4). Thus, the position, speed, and heading angle are all plotted at point T in Figure 6, where the solid green curves are obtained from the drone video and the black circles are the GPS coordinates (transferred to the local coordinate system of the intersection). Panels (a), (c) and (e) relate to the test run investigated above, while panels (b), (d) and (f) are representing another test run.

In Figure 6(a), the GPS and the video data have a good fit, while the trajectory from the video is smoother. The speed information obtained from the drone video in Figure 6(c) is the same as the green curve in Figure 5(a), which is smoothed using the window size of 1/3 s. The yaw angle cannot be generated from the GPS data directly (at point T), thus it is hard to compare with the video process result. Instead, the heading angle at point T is compared. From the drone view video, the heading angle θ_T at point T is calculated similarly as shown by Eq. (6) for point C.

While for the run shown in panels (a), (c) and (e), the video-based position and the GPS position match well (see panel (a)), for the other run shown in panels (b), (d) and (f), the GPS position has more than 4 meters difference due to the GPS drift (Kim et al., 2015) (see panel (b)). Panel (d) shows that the vehicle runs slower compared to panel (c), with a maximum speed of about 9 m/s. Thus, it takes longer to finish the turn. The GPS-based speed and heading angle show good agreement with the video-based results in panels (c,d) and (e,f), respectively.

In summary, the processed video results in smoother data in position and yaw angle (or heading angle) and provide richer information on the vehicle dynamics (heading angles and velocity components at different points, yaw rate) compared to the GPS measurements, which only give information about the motion of a single point.

4. Homography

In order to transform the bounding box from a drone view image to a roadside view image (see Figure 2), homography (Criminisi et al., 1997) is used in this section. This transformation generates bounding boxes for a roadside camera together with the ground truth data (position, speed, yaw angle, and yaw rate) extracted from the drone view image. The obtained data set can be utilized for training in machine learning algorithms.

Considering the high vertical position of the drone, we can assume that the bounding box in the drone view (x, y, z) is a rectangle, corresponding to the rectangular bounding box on the ground plane (X, Y, Z). On the other hand, as illustrated in Figure 7, a rectangle on the ground plane is deformed from the perspective of the roadside camera. Correspondingly, the bounding box forms a quadrilateral in the roadside camera view (x', y', z'). To convert the coordinates of an arbitrary point (x, y) on the drone image plane to a point (x', y') on the roadside image plane, we define the transformation:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \frac{1}{w} \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{=\mathbf{H}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{9}$$

where $\mathbf{H}$ is the transformation matrix. This matrix originates from geometric transformations: a spatial shift (3 coordinates) is defined between the camera positions; three sequential spatial rotation (3 angles) between the coordinate systems of the two cameras; center projections to the image planes (2 distances between the center points of the projections and image planes). All together, these transformations are defined by 8 scalar parameters, and w stands for the scaling factor related to the optical properties of the camera objectives. It is worth to note that homography does not consider the distortion of the camera lens, and it preserves the straight lines.

Consider a point (x_1, y_1) on the drone image plane, and the projected point (x_1', y_1') on the roadside image plane. Then Eq. (9) gives

$$\frac{1}{w}(h_{11}x_1 + h_{12}y_1 + h_{13}) = x_1', \tag{10a}$$

$$\frac{1}{w}(h_{21}x_1 + h_{22}y_1 + h_{23}) = y_1', \tag{10b}$$

$$\frac{1}{w}(h_{31}x_1 + h_{32}y_1 + h_{33}) = 1. \tag{10c}$$

and Eqs. (10a) and (10c) lead to

$$x_1h_{11} + y_1h_{12} + h_{13} - x_1x_1'h_{31} - y_1x_1'h_{32} - x_1'h_{33} = 0. \tag{11}$$

Similarly, Eqs. (10b) and (10c) provide

$$x_1h_{21} + y_1h_{22} + h_{23} - x_1y_1'h_{31} - y_1y_1'h_{32} - y_1'h_{33} = 0. \tag{12}$$

As it can be seen, the scaling factor w does not show up in these equations, and, for example, the assumption $h_{33} = 1$ can be used without the loss of

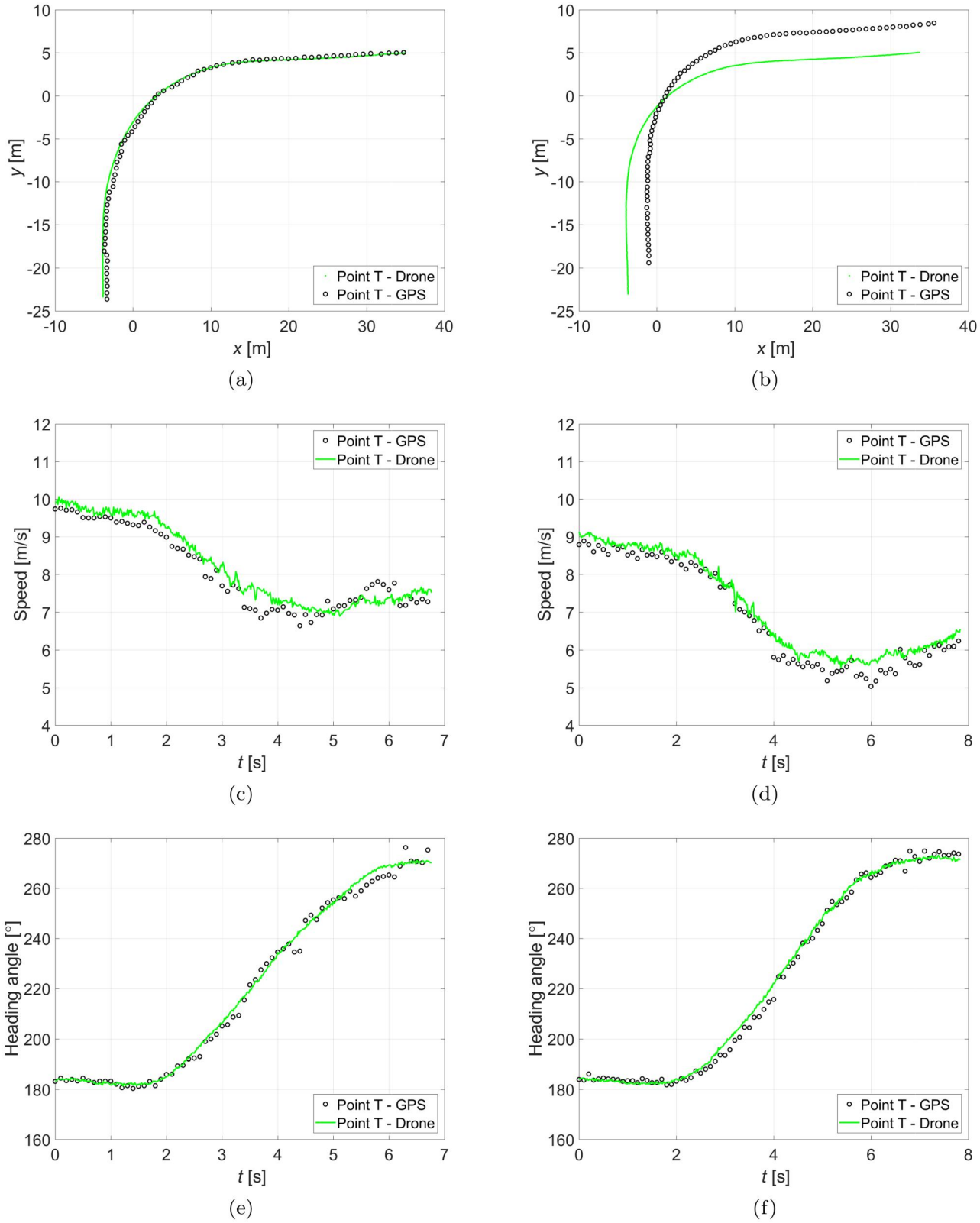

Figure 6. Comparison with GPS data: (a,b) position; (c,d) speed; (e,f) heading angle. (a,c,e) are from one run of the experiment, while (b,d,f) are from another run.

generality. As a consequence, eight unknowns remain in the matrix **H**, and four points are needed to identify the homography transformation matrix.

To map from the drone image plane to the roadside image plane, more (than four) points can be selected to enhance the accuracy of the transformation. The problem

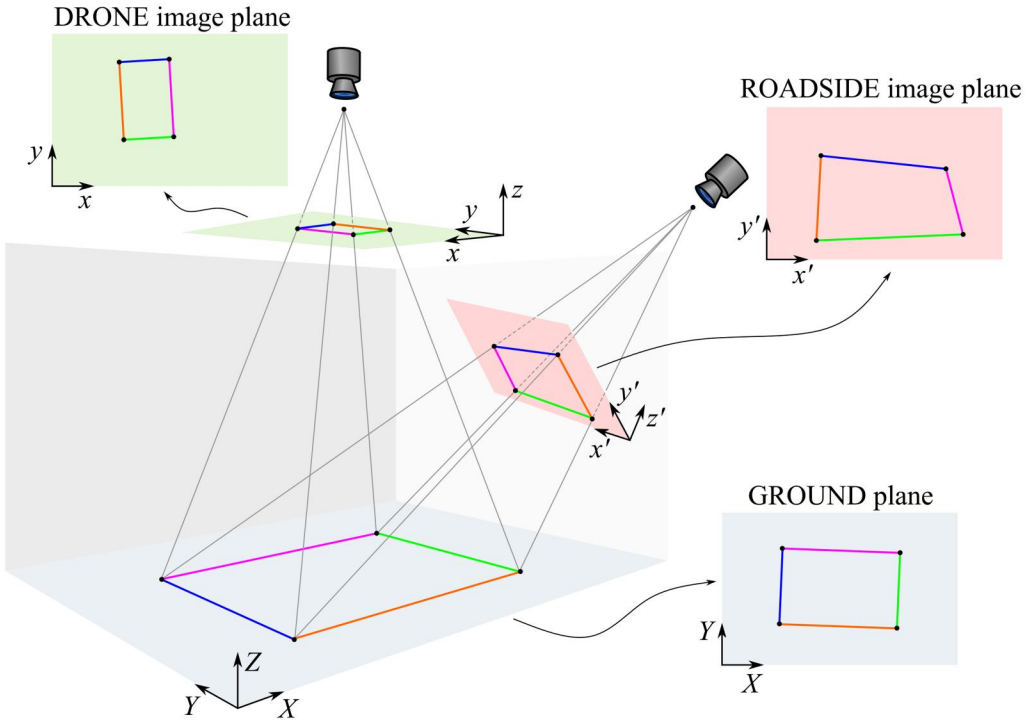

Figure 7. The rectangle on the ground plane is projected to the image plane of the drone camera, where its shape closely resembles a rectangle due to the high altitude of the drone. The rectangle on the ground plane is realized as a quadrilateral on the image plane of the roadside camera.

is converted to finding the matrix **H** for the best fit projected plane in the fixed-camera view, which can be solved using the singular value decomposition.

Rewrite the **H** matrix as

$$\mathbf{h} = \left[h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33} \right]^{\mathrm{T}}, \tag{13}$$

and consider n points. Then Eq. (9) becomes

$$\mathbf{Ah} = \mathbf{0}, \tag{14}$$

where

$$\mathbf{A} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1' & -y_1x_1' & -x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1' & -y_1y_1' & -y_1' \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x_2' & -y_2x_2' & -x_2' \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y_2' & -y_2y_2' & -y_2' \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx_n' & -y_nx_n' & -x_n' \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny_n' & -y_ny_n' & -y_n' \end{bmatrix}_{2n\times 9}. \tag{15}$$

Vector **h** can be obtained by the eigenvector of the least eigenvalue of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, which is the direct result of the singular value decomposition of a matrix **A**.

As shown in Figure 8, $n = 13$ points are selected from the drone view and from the roadside view, marked as yellow circles. The red crosses on the roadside view are the points mapped from the drone view using matrix **H**, and they are very close to the hand-selected points. The panels in the left of Figure 9 show the bounding boxes of the truck in the drone view at different positions. These are mapped to the roadside camera image on the right. Note that the image distortion caused by the roadside camera is corrected prior to the projection.

5. Conclusion

Using images from unmanned aerial vehicles (UAVs), an algorithm was developed to track ground vehicles in an intersection, extract their kinematic data, and generate labeled data sets for machine learning algorithms. Different from previous studies focusing on vehicle detection, classification, and tracking in traffic surveillance, we used recorded videos from UAVs to draw bounding boxes around the vehicles, and to extract and analyze the kinematic variables of vehicles. These variables included position, yaw angle, velocity (i.e., speed and heading angle), and yaw rate. The position and yaw angle can be read directly from the bounding boxes, while the velocity and yaw rate are calculated using numerical differentiation with respect to time. Part of the results including position, heading angle, and speed were compared with the data from the GPS installed on the vehicle; while the yaw angle and yaw rate can only be obtained from the UAV images. Furthermore, the bounding boxes from the

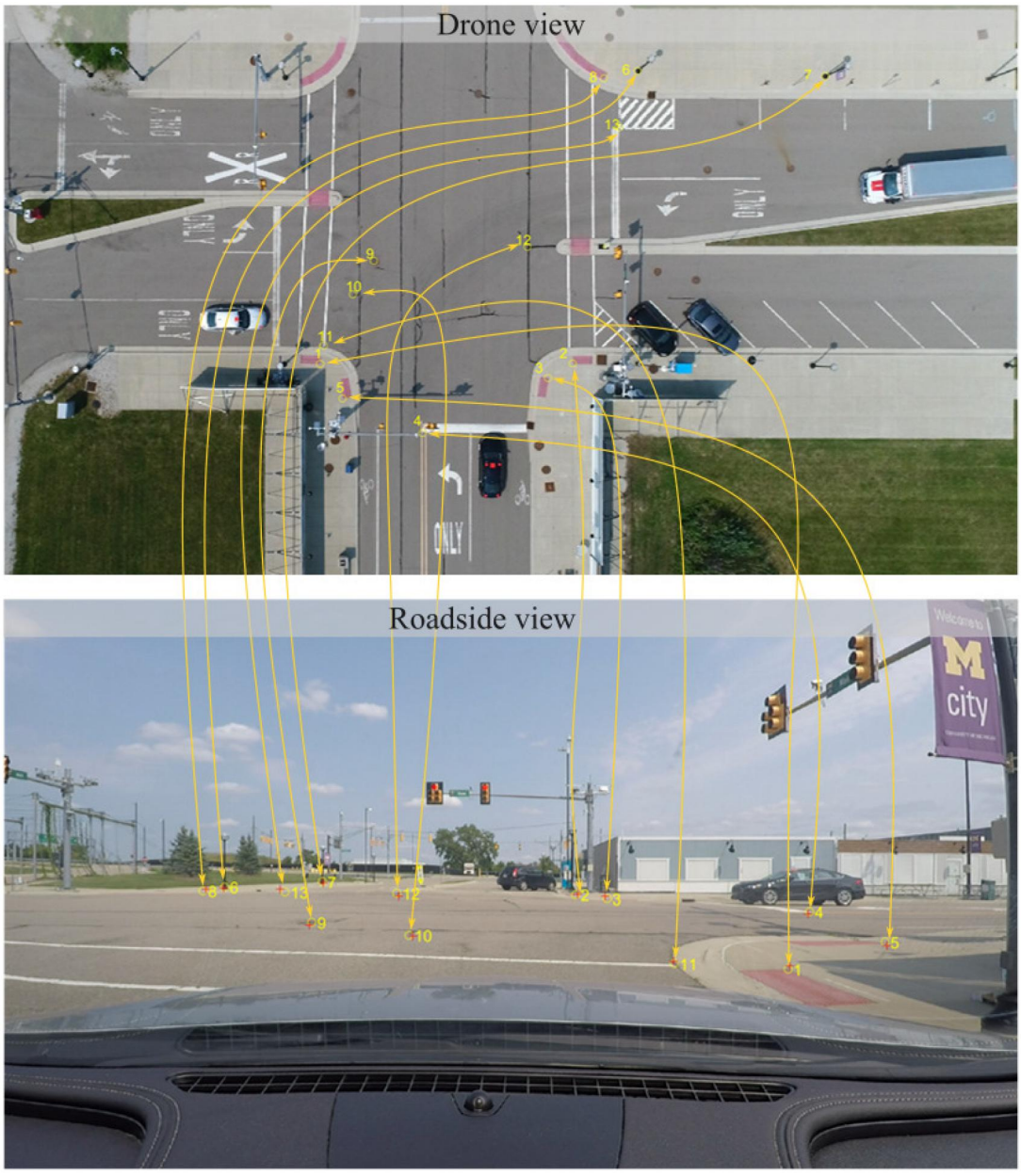

Figure 8. Points selected to map the drone view (upper figure) to the roadside view (lower figure). Yellow circles are hand selected points, and red crosses are calculated from the drone view points using matrix **H** for validation purpose.

drone view image were projected to a roadside view image using homography.

The extracted kinematic data can be used in the control and planning of smart intersections, especially in environments containing connected automated vehicles (CAVs). Knowing the dynamics of other vehicles can benefit the CAVs for their decision making and path planning. With the obtained data and bounding boxes (from roadside camera) as the" ground truth", this algorithm can be used to generate training data for machine learning algorithms that can perform more complex tasks, for example, online tracking. Note that the UAV images are only used for generating labeled data sets. After trained, the machine learning algorithms will only use images from the roadside camera as input.

In this study, the background image is one frame of the drone view video without the target vehicles, which may be difficult to obtain in real traffic. This can be resolved by taking an average of multiple frames. The bounding box size of a certain vehicle is fixed in this study, however, the vehicle may deform due to the perspective of the drone camera, especially close to the edge of the image. This deformation may be pre-processed before the vehicle detection. Future work will apply the proposed framework to train machine learning algorithms using the generated high-precision data.

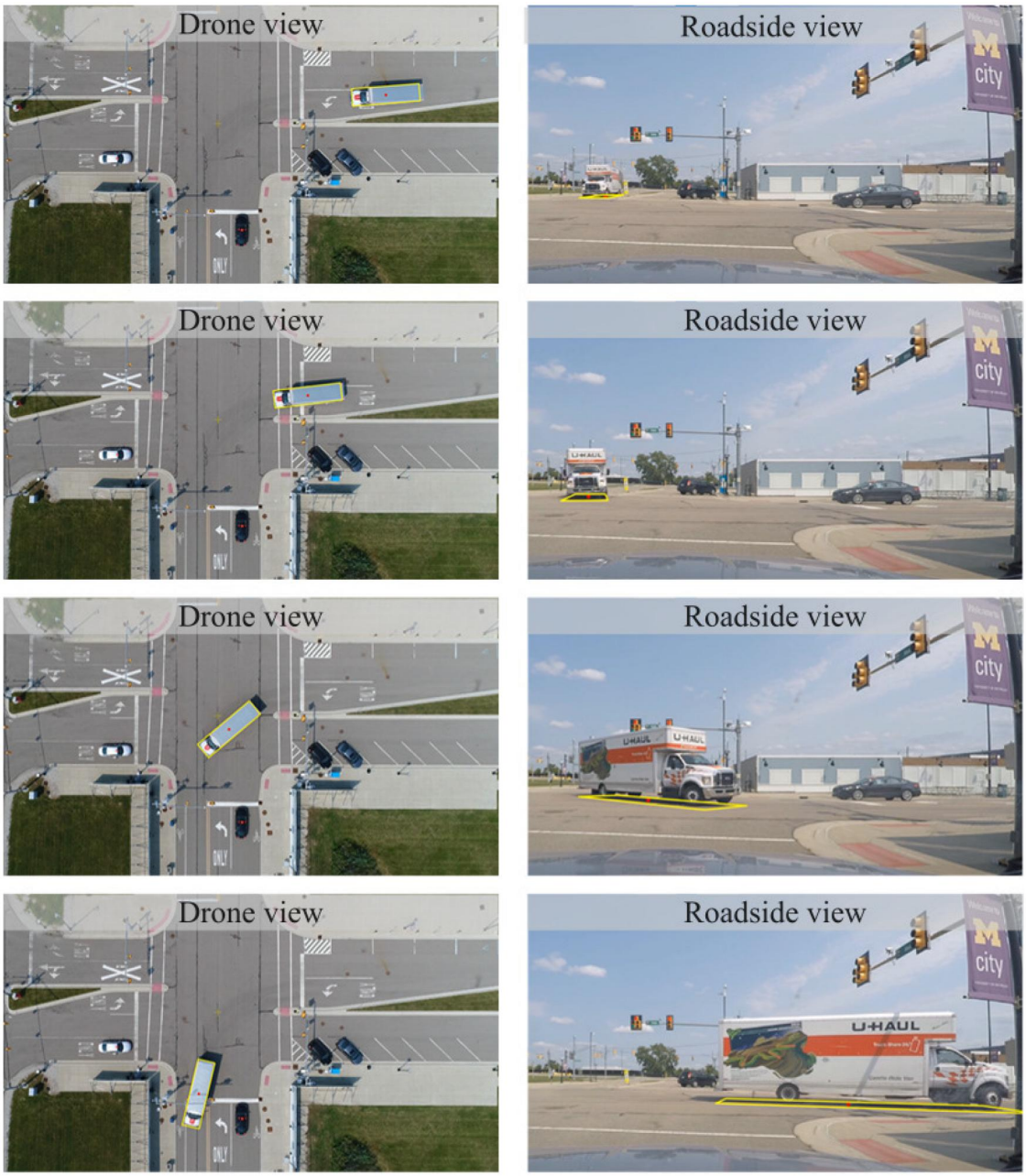

Figure 9. Using homography to transfer the bounding box from the drone view to the roadside view.

Acknowledgements

Dénes Takács was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, and he would also like to thank the Rosztoczy Foundation for their generous support. The authors would like to thank Anil Alan, Xunbi Ji, Sanghoon Oh, Minghao Shen and Hao Wang for their help in the experiments.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research was supported by the University of Michigan's Center for Connected and Automated Transportation through the US DOT grant [69A3552348305] and by the National Research, Development and Innovation Office of Hungary under grant no. [NKFI-146201].

ORCID

Dénes Takács http://orcid.org/0000-0003-1226-8613

References

Barmpounakis, E. N., Vlahogianni, E. I., & Golias, J. C. (2016). Unmanned aerial aircraft systems for transportation engineering: Current practice and future challenges. *International Journal of Transportation Science and Technology*, *5*(3), 111–122. https://doi.org/10.1016/j.ijtst.2017.02.001

Barmpounakis, E., & Geroliminis, N. (2020). On the new era of urban traffic monitoring with massive drone data: The pneuma large-scale field experiment. *Transportation Research C*, *111*, 50–71. https://doi.org/10.1016/j.trc.2019.11.023

Bock, J., Krajewski, R., Moers, T., Runde, S., Vater, L., Eckstein, L. (2020). The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections. In *IEEE Intelligent Vehicles Symposium (IV)* (p. 1929–1934). IEEE.

Bouguettaya, A., Zarzour, H., Kechida, A., & Taberkit, A. M. (2022). Vehicle detection from UAV imagery with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(11), 6047–6067. https://doi.org/10.1109/TNNLS.2021.3080276

Braut, V., Čuljak, M., Vukotić, V., Šegvić, S., Ševrović, M., & Gold, H. (2012). *Estimating OD matrices at intersections in airborne video-a pilot study* [Paper presentation]. In 35th International Convention MIPRO (pp. 977–982).

Chen, X., Li, Z., Yang, Y., Qi, L., & Ke, R. (2021). High-resolution vehicle trajectory extraction and denoising from aerial videos. *IEEE Transactions on Intelligent Transportation Systems*, *22*(5), 3190–3202. https://doi.org/10.1109/TITS.2020.3003782

Criminisi, A., Reid, I., Zisserman, A. (1997). Computing the plane to plane homography. www.robots.ox.ac.uk/∼vgg/presentations/bmvc97/criminispaper/node3.html

Datondji, S. R. E., Dupuis, Y., Subirats, P., & Vasseur, P. (2016). A survey of vision-based traffic monitoring of road intersections. *IEEE Transactions on Intelligent Transportation Systems*, *17*(10), 2681–2698. https://doi.org/10.1109/TITS.2016.2530146

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Guido, G., Gallelli, V., Rogano, D., & Vitale, A. (2016). Evaluating the accuracy of vehicle tracking data obtained from unmanned aerial vehicles. *International Journal of Transportation Science and Technology*, *5*(3), 136–151. https://doi.org/10.1016/j.ijtst.2016.12.001

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Husain, A. A., Maity, T., & Yadav, R. K. (2020). Vehicle detection in intelligent transport system under a hazy environment: A survey. *IET Image Processing*, *14*(1), 1–10. https://doi.org/10.1049/iet-ipr.2018.5351

Kanistras, K., Martins, G., Rutherford, M. J., & Valavanis, K. P. (2013). A survey of unmanned aerial vehicles (UAVs) for traffic monitoring. In *International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 221–234). IEEE.

Kaufmann, S., Kerner, B. S., Rehborn, H., Koller, M., & Klenov, S. L. (2018). Aerial observations of moving synchronized flow patterns in over-saturated city traffic. *Transportation Research Part C: Emerging Technologies*, *86*, 393–406. https://doi.org/10.1016/j.trc.2017.11.024

Khan, M. A., Ectors, W., Bellemans, T., Janssens, D., & Wets, G. (2017). Unmanned aerial vehicle–based traffic analysis: Methodological framework for automated multi-vehicle trajectory extraction. *Transportation Research Record*, *2626*(1), 25–33. https://doi.org/10.3141/2626-04

Kim, K., Kim, W., Choi, D., Myung, H. (2015). Calibration of the drift error in GPS using optical flow and fixed reference station. In *15th International Conference on Control, Automation and Systems (ICCAS)* (pp. 1370–1373). IEEE.

Liu, K., & Mattyus, G. (2015). Fast multiclass vehicle detection on aerial images. *IEEE Geoscience and Remote Sensing Letters*, *12*(9), 1938–1942. https://doi.org/10.1109/LGRS.2015.2439517

Liu, Y., Tian, B., Chen, S., Zhu, F., & Wang, K. (2013). A survey of vision-based vehicle detection and tracking techniques in its. In *IEEE International Conference on Vehicular Electronics and Safety* (pp. 72–77). IEEE.

Liu, Z., He, J., Zhang, C., Yan, X., Wang, C., & Qiao, B. (2023). Vehicle trajectory extraction at the exit areas of urban freeways based on a novel composite algorithms framework. *Journal of Intelligent Transportation Systems*, *27*(3), 295–313. https://doi.org/10.1080/15472450.2021.2021079

Pajares, G. (2015). Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs). *Photogrammetric Engineering & Remote Sensing*, *81*(4), 281–330. https://doi.org/10.14358/PERS.81.4.281

Puri, A. (2005). *A survey of unmanned aerial vehicles (uav) for traffic surveillance* (pp. 1–29). Department of Computer Science and Engineering, University of South Florida.

Razakarivony, S., & Jurie, F. (2016). Vehicle detection in aerial imagery: A small target detection benchmark. *Journal of Visual Communication and Image Representation*, *34*, 187–203. https://doi.org/10.1016/j.jvcir.2015.11.002

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 779–788). IEEE.

Won, M. (2020). Intelligent traffic monitoring systems for vehicle classification: A survey. *IEEE Access*. *8*, 73340–73358. https://doi.org/10.1109/ACCESS.2020.2987634

Xia, G.-S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., & Zhang, L. (2018). *DOTA: A large-scale dataset for object detection in aerial images*. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3974–3983). IEEE.

Zheng, O., Abdel-Aty, M., Yue, L., Abdelraouf, A., Wang, Z., & Mahmoud, N. (2022). Citysim: A drone-based vehicle trajectory dataset for safety oriented research and digital twins. *arXiv preprint arXiv:2208.11036*.