sensors

MDPI

Article

Vehicle Spatial Distribution and 3D Trajectory Extraction Algorithm in a Cross-Camera Traffic Scene

Xinyao Tang, Huansheng Song *, Wei Wang and Yanni Yang

School of Information Engineering, Chang'an University, Xi'an 710064, China; andy19966212@126.com (X.T.); wangwei_211@chd.edu.cn (W.W.); yanniyang@chd.edu.cn (Y.Y.)

* Correspondence: hshsong@chd.edu.cn

Received: 30 September 2020; Accepted: 13 November 2020; Published: 14 November 2020

Abstract: The three-dimensional trajectory data of vehicles have important practical meaning for traffic behavior analysis. To solve the problems of narrow visual angle in single-camera scenes and lack of continuous trajectories in 3D space by current cross-camera trajectory extraction methods, we propose an algorithm of vehicle spatial distribution and 3D trajectory extraction in this paper. First, a panoramic image of a road with spatial information is generated based on camera calibration, which is used to convert cross-camera perspectives into 3D physical space. Then, we choose YOLOv4 to obtain 2D bounding boxes of vehicles in cross-camera scenes. Based on the above information, 3D bounding boxes around vehicles are built with geometric constraints which are used to obtain projection centroids of vehicles. Finally, by calculating the spatial distribution of projection centroids in the panoramic image, 3D trajectories of vehicles are extracted. The experimental results indicate that our algorithm can effectively complete vehicle spatial distribution and 3D trajectory extraction in various traffic scenes, which outperforms other comparison algorithms.

Keywords: camera calibration; cross-camera traffic scene; road panoramic image; vehicle spatial distribution; 3D trajectory extraction

1. Introduction

Vehicle spatial distribution and 3D trajectory extraction is an important sub-task in the field of computer vision. With the development of intelligent transportation systems (ITS), a large amount of vehicle trajectory data reflecting movements is obtained through traffic surveillance videos, which can be used for traffic behavior analysis [1,2] such as speeding and lane change, traffic flow parameter (volume, density, etc.) calculation and prediction [3–5] and so on. Based on these data, traffic state estimation [6,7] and traffic management and control [8] can be conducted, which plays a key role in ensuring traffic efficiency and is of great research significance and practical value.

In current applications, trajectories mainly refer to two-dimensional trajectories in the image space, which do not contain spatial information of vehicles in the real world. Compared with 2D trajectories, 3D trajectories have one more dimension of spatial information, which has more obvious advantages in practical applications and can be further applied to traffic accident scene reconstruction and responsibility identification [9], as well as vehicle path planning [10] in autonomous driving and cooperative vehicle infrastructure system (CVIS) to avoid collision.

Currently, the most commonly used methods for obtaining 3D vehicle trajectories are based on object detection and feature point methods [11–13], which have been maturely applied in single-camera scenes. With the development of deep convolutional neural networks (DCNNs), several excellent object detection networks [14–18] have emerged, which greatly improve the accuracy and speed of object detection compared with the traditional feature extraction and classifier methods [19]. Based on object detection, feature points are extracted for vehicles to obtain 3D trajectories in the world space

combined with camera calibration. Although these methods have been widely and maturely used in single-camera scenes, the trajectory results are not accurate under the condition of low camera perspectives and vehicle occlusion. To solve the problem, 3D object detection is considered because only the presence, 2D location and rough type of vehicles in the image space can be obtained by 2D object detection method and it is difficult to achieve a fine-grained description of vehicles. Compared with 2D methods, perspective distortion can be eliminated by 3D object detection. Moreover, 3D bounding box fits the vehicle better and can describe vehicle size, pose, and other information on the physical scale. Therefore, 3D model is more suitable for 3D trajectory extraction in traffic scenes. At the same time, the visual angle of single-camera scene is usually narrow, which is unable to meet the needs of applications in wide range scenes, so it is necessary to solve the problem of 3D vehicle trajectory extraction in the whole space.

At present, full space fusion mainly relies on multi-scene stitching methods, which can be divided into two categories. (1) Image stitching based on image alignment [20,21]. The feature points of the overlapping areas in multiple images are detected and matched to construct homography matrixes between images. Then, the panoramic image is generated based on the matrixes. This kind of method is mature and widely used, especially in the panoramic photography of mobile phone applications [22]. However, camera calibration is not used in these methods, which means physical information cannot be reflected in the panoramic image. (2) Image stitching based on camera calibration [23,24]. The transformation between world coordinate systems of the scenes are determined based on overlapping areas in the image and camera calibration to generate the panoramic image which contains actual physical information and can be used to measure and locate the world coordinates in the image. However, this kind of method requires complicated manual calibration of each camera, and has rarely been used in large scope of road measurement.

Methods of vehicle trajectory extraction in the whole space is cross-camera vehicle tracking, which means obtaining continuous vehicle trajectory from images taken by multiple cameras with or without overlapping areas. These methods usually contain three essential steps: camera calibration, vehicle detection, and tracking in single-camera scenes and cross-camera vehicle matching. For cameras with overlapping areas, the spatial correlation can be calculated by overlapping areas to obtain continuous trajectories. However, in practical applications, "blind areas" are often existed in images taken by multiple cameras. In case of this condition, methods of re-identification are used to accurately and efficiently match vehicles in different perspectives through vehicle apparent features. Then, continuous vehicle trajectories in the whole space can be obtained by space-time information inference.

Currently, re-identification methods used in cross-camera vehicle tracking are mostly based on vehicle features, such as vehicle color, shape, and texture, among which SIFT feature is the most commonly used due to its invariance to light, rotation, and scale. However, robustness of SIFT to affine transformation is low. To improve this problem, Hsu et al. [25] proposed a method of cross-camera vehicle matching based on ASIFT feature and min-hash technique, which can overcome the influence of multi-camera perspectives to feature detection but cannot obtain 3D vehicle trajectory and solve the problem of vehicle occlusion. Castaneda et al. [26] proposed a method of multi-camera detection and tracking of vehicles in non-overlapping tunnel scene, which uses optical flow and Kalman filter for vehicle tracking and state estimation in single scenes. Due to the special light environment in tunnel scene, vehicle color cannot be used as matching criterion. Thus, vertical and horizontal signatures are proposed to describe the similarity between vehicles. Combined with cross-camera vehicle travel time and lane position constraints, continuous vehicle trajectory can be obtained. To some extent, the problem of vehicle occlusion can be solved in this method, but the physical location of vehicle trajectory in 3D space is still not available. To further obtain vehicle trajectory in 3D space, multi-scene cameras should be calibrated in advance and the topological relationship between cameras should be determined to convert multi-camera perspectives into point sets in 3D coordinate system. Straw et al. [27] proposed a method of cross-camera vehicle tracking which uses DLT and triangulation for camera calibration and Kalman filter for vehicle state estimation.

Although continuous trajectory in 3D space could be obtained, the accuracy is low, which cannot meet practical applications. Peng et al. [28] proposed a method of multi-camera vehicle detection and tracking in non-overlapping traffic surveillance, using convolutional neural network (CNN) for object detection and feature extraction and homography matrix for displaying vehicle trajectory to satellite map. This method can accurately show vehicle trajectory in panoramic map, but these trajectories do not contain physical location in 3D space. Byeon et al. [29] proposed an online method of cross-camera vehicle positioning and tracking, which uses Tsai two-step calibration method for camera calibration and represents vehicle matching as multi-dimensional assignment to solve the problem of vehicle matching in multi-camera scenes. Vehicle trajectory can be obtained in this method, but the road panoramic image with spatial information is not generated. Qian et al. [30] proposed a cross-camera vehicle tracking system for smart cities which uses object detection, segmentation, and multi-object tracking algorithms to extract vehicle trajectories in single-camera scenes. Then, a cross-camera multi-object tracking network is proposed to predict a matrix which measures the feature distance between trajectories in single-camera scenes. The system won the first place in AI City 2020 Challenge and can better solve the problem of vehicle matching in cross-camera scenes. However, continuous 3D trajectories of vehicles and the panoramic image of the scene cannot be obtained.

In view of the problems existing in current cross-camera vehicle tracking methods, such as the influence of visual angle, vehicle occlusion, and lack of continuous trajectories in 3D space, we propose an algorithm of vehicle spatial distribution and 3D trajectory extraction in cross-camera traffic scene. The main contributions of this paper are summarized as follows:

- A method of road space fusion in cross-camera scenes based on camera calibration is proposed to generate a road panoramic image with physical information, which is used to convert cross-camera perspectives into 3D physical space.
- A method of 3D vehicle detection based on geometric constraints is proposed to accurately obtain projection centroids of vehicles, which is used to describe vehicle spatial distribution in the panoramic image and 3D trajectory extraction of vehicles.

The rest of this paper is organized as follows. The proposed algorithm to complete vehicle spatial distribution and 3D trajectory extraction is illustrated in Section 2. Experiment results and some comparison experiments are presented in Section 3. Conclusions and future work are given out in Section 4.

2. Materials and Methods

2.1. Framework

The overall flow chart of the proposed algorithm is shown in Figure 1. First, a panoramic image of the road with spatial information is generated based on camera calibration, which is used to convert the cross-camera perspective into 3D physical space. Secondly, 3D bounding box is constructed by geometric constraints, which is used to obtain the projection centroid of the vehicle. Finally, 3D trajectory of the vehicle is extracted by calculating the spatial distribution of the projection centroid in the road panoramic image.

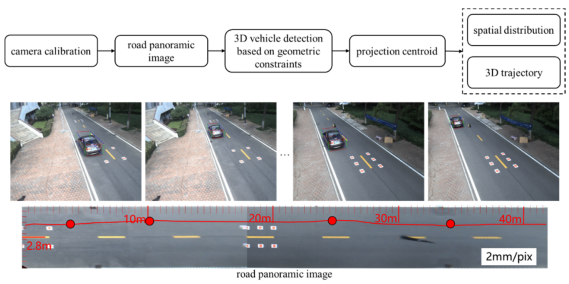


Figure 1. The overall flow chart of the proposed algorithm.

2.2. Road Space Fusion in Cross-Camera Scene

2.2.1. Camera Calibration Model and Parameter Calculation

To complete road space fusion in cross-camera scenes, the relationship between 2D image space and 3D world space must be derived through camera calibration. In this paper, we refer to the study [31] and our previous work [32,33] to define coordinate systems and camera calibration model, and choose the single vanishing point-based calibration method VWL (One Vanishing Point, Known Width and Length) to complete the calculation of calibration parameters.

Schematic diagram of coordinate system and camera calibration model is shown in Figure 2. In this paper, three coordinate systems are defined, all of which are right-handed. The world coordinate system is defined by the x, y, z axis, and the origin O_w is located at the projection point of the camera on the road plane, whereas z is perpendicular to the road plane upwards. The camera coordinate system is defined by the x_c, y_c, z_c axis, and the origin O_c is located at the camera optical center, and x_c is parallel to x, z_c pointing to the ground along the camera optical axis, y_c perpendicular to the plane $x_cO_cz_c$. The image coordinate system is defined by u, v axis, and the origin O_i is located at image center. In the image coordinate system, u is horizontal right and v is vertical downward. z_c intersects the road plane at $r = (c_x, c_y)$ in the image coordinate system, which is called the principal point and its default location is at the center of the image. c_x, c_y represent half of the image width and height, respectively.

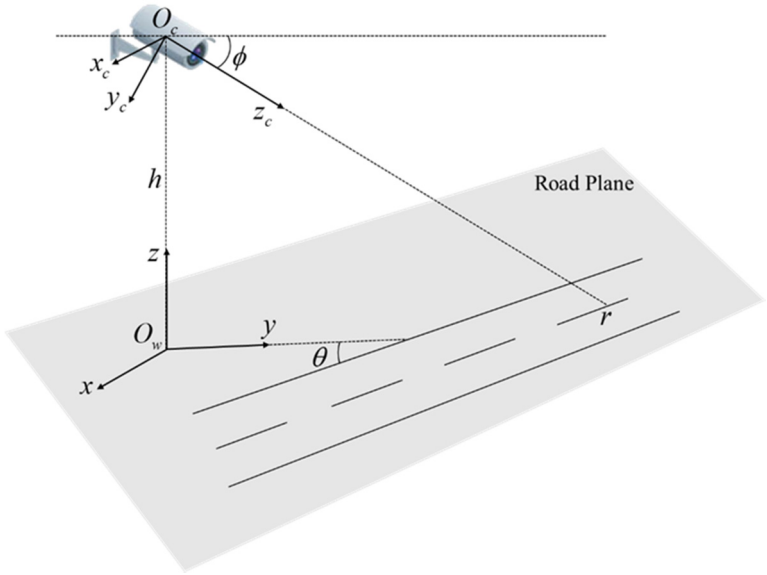


Figure 2. Schematic diagram of coordinate systems and camera calibration model.

In camera calibration, calibration parameters usually include camera focal length f, camera height h above the road plane, tilt angle ϕ and pan angle θ. In addition, roll angle can be represented by a simple image rotation, which has no effect on calibration results and is not considered in this paper. Through the camera model, the projection expression from the world coordinate system to the image coordinate system can be deduced as follows:

$$\alpha \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & -f\sin\phi & -f\cos\phi & fh\cos\phi \\ 0 & \cos\phi & -\sin\phi & h\sin\phi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \tag{1}$$

where $\alpha \neq 0$ is the scale factor, the homogeneous coordinates of the world point and its projection are $\begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ and $\begin{bmatrix} u & v & 1 \end{bmatrix}^T$.

In this paper, the single vanishing point-based calibration method VWL [31,32] is adopted to solve the calibration parameters f, h, ϕ, θ, and the vanishing point $VP = (u_0, v_0)$ along the direction of traffic flow is extracted by road edge lines.

As shown in Figure 3, a line segment in the world coordinate system and its projection in the image coordinate system are presented, respectively. In Figure 3a, due to the pan angle θ, the point at infinity along the road direction can be expressed as $\boldsymbol{x}_0 = \begin{bmatrix} -\tan\theta & 1 & 0 & 0 \end{bmatrix}^T$ in the world homogeneous coordinate. In Figure 3b, according to the vanishing point principle, $(u_0,\ v_0)$ is the projection of $\boldsymbol{x}_0$ in the image coordinate system. From Equation (1), the calibration parameters ϕ, θ can be solved as follows:

$$\phi = \arctan(-v_0/f), \tag{2}$$

$$\theta = \arctan(-u_0 \cos\phi / f), \tag{3}$$

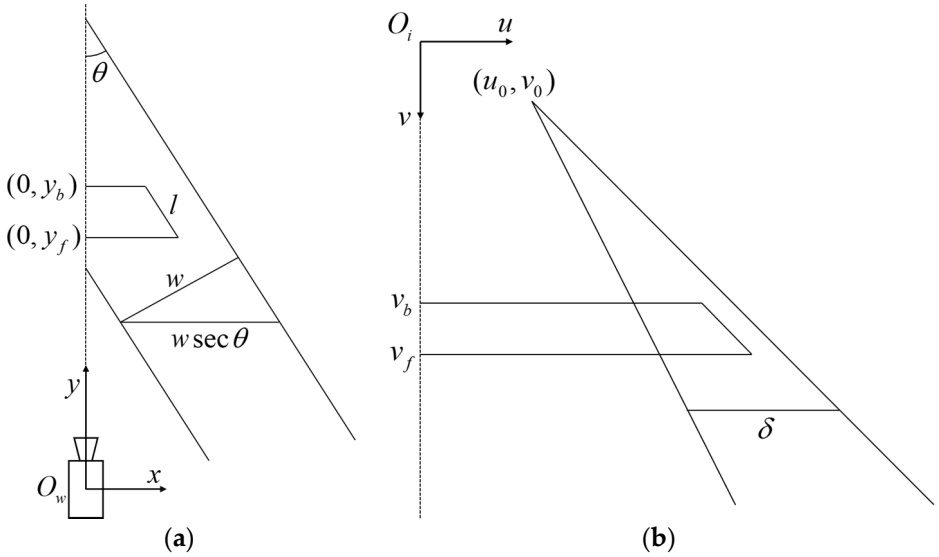


Figure 3. A line segment in the world coordinate system and its projection in the image. (**a**) World coordinate system; (**b**) Image coordinate system.

Besides vanishing points, markings on the road plane are also commonly used signs. In Figure 3a, the physical length of a line segment parallel to the road direction is l. The vertical coordinates of the front and back point are y_b, y_f and v_b, v_f, where y represents the world coordinate system while v the image. The physical width of the road is w with a pixel length δ in the corresponding image coordinate system. It can be obtained from literature [31] that h can be expressed by w or l indirectly as follows:

$$h = \frac{fw\sin\phi}{\delta\cos\theta}, \tag{4}$$

$$h = \frac{f\tau l\cos\phi}{f^2 + {v_0}^2}, \tag{5}$$

where $\tau = (v_f - v_0)(v_b - v_0)/(v_f - v_b)$, $\sin\phi, \cos\phi, \cos\theta$ can be solved from Equation (2) and (3). By equating Equations (4) and (5) and substituting into $\sin\phi, \cos\phi, \cos\theta$, a fourth-order equation in f can be derived as:

$$f^4 + [2({u_0}^2 + {v_0}^2) - {k_V}^2]f^2 + ({u_0}^2 + {v_0}^2)^2 - {k_V}^2{v_0}^2 = 0 \tag{6}$$

where $k_V = \delta\tau l/(wv_0)$.

From Equation (6), f can be solved first. When f is uniquely determined, ϕ, θ can be solved according to Equations (2) and (3), and h can be finally solved according to Equations (4) or (5).

Thus, all the calibration parameters are calculated and the mapping between world and image can be described according to Equation (1).

To illustrate the road space in a straightforward way, the origin of the image coordinate system O_i and the y axis of the world coordinate system are adjusted. First, the origin of the image coordinate system is moved to the upper left corner of the image, corresponding to the change of the internal parameter matrix K:

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Then, the y axis is adjusted to the direction along the traffic flow. Therefore, the rotation matrix R contains two parts, respectively representing a rotation of $\phi + \pi/2$ about the x axis and θ about the z axis, which can be specifically expressed as:

$$R = R_x(\phi + \pi/2)R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ -\sin\phi\sin\theta & -\sin\phi\cos\theta & -\cos\phi \\ \cos\phi\sin\theta & \cos\phi\cos\theta & -\sin\phi \end{bmatrix}$$

The translation matrix is:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -h \end{bmatrix}$$

Therefore, the adjusted mapping from world point (x, y, z) to image point (u, v) in homogeneous form can be expressed as:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KRT\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = H\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

where $H = \left[h_{ij}\right], i = 1, 2, 3; j = 1, 2, 3, 4$ is the 3×4 projection matrix from the world coordinate to the image coordinate, and s is the scale factor.

Finally, according to the derivation, the adjusted mapping between world and image can be described as follows:

$$\text{World} - \text{to} - \text{Image} \begin{cases} u = \frac{h_{11}x + h_{12}y + h_{13}z + h_{14}}{h_{31}x + h_{32}y + h_{33}z + h_{34}} \\ v = \frac{h_{21}x + h_{22}y + h_{23}z + h_{24}}{h_{31}x + h_{32}y + h_{33}z + h_{34}} \end{cases}, \tag{7}$$

$$\text{Image} - \text{to} - \text{World} \begin{cases} x = \frac{b_1(h_{22} - h_{32}v) - b_2(h_{12} - h_{32}u)}{(h_{11} - h_{31}u)(h_{22} - h_{32}v) - (h_{12} - h_{32}u)(h_{21} - h_{31}v)} \\ y = \frac{-b_1(h_{21} - h_{31}v) + b_2(h_{11} - h_{31}u)}{(h_{11} - h_{31}u)(h_{22} - h_{32}v) - (h_{12} - h_{32}u)(h_{21} - h_{31}v)} \end{cases}, \tag{8}$$

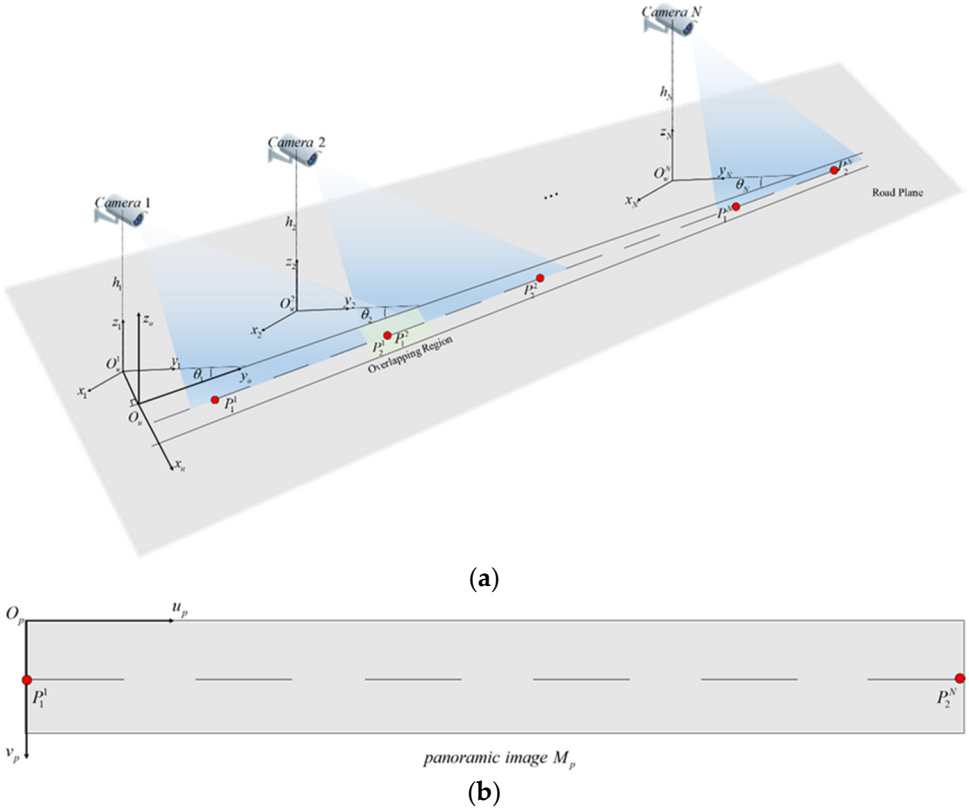
where $\begin{cases} b_1 = u(h_{33}z + h_{34}) - (h_{13}z + h_{14}) \\ b_2 = v(h_{33}z + h_{34}) - (h_{23}z + h_{24}) \end{cases}$.

2.2.2. Unified World Coordinate System and Road Panoramic Image Generation

The mapping between world and image in a single scene can be described through camera calibration. To complete 3D vehicle trajectory extraction in cross-camera scenes, the road space needs to be fused. At present, image stitching methods are often used, but most of them rely on overlapping areas to extract feature points for matching and obtaining transformation of scenes. However, feature extraction and matching are time-consuming. For multi-scene (more than two scenes) stitching, accumulated errors are existed in transformation of scenes, which will affect the quality of final image stitching result and the measurement accuracy of physical distance. Therefore, we propose a road space fusion algorithm in cross-camera scenes based on camera calibration which is

not completely dependent on overlapping areas between scenes. When there are no overlapping areas between scenes, only the distances between cameras are needed.

Schematic diagram of road space fusion in cross-camera scenes is shown in Figure 4. In Figure 4a, number of cameras in the scene is $N(N \geq 2)$, the set of sub-scene world coordinate systems is defined as $\left\{W_s^i : O_w^i - x_i y_i z_i; i = 1, 2, \cdots, N\right\}$, which is the same as the world coordinate system in the single scene described in the previous section. The unified world coordinate system is defined as $W_u : O_u - x_u y_u z_u$, and the origin O_u is located in the road edge close to the camera. $O_u O_w^1$ is perpendicular to the road edge. The mapping matrix between the world coordinate system and the image coordinate system of each scene is the adjusted result described in the previous section, which is defined as $H_i, i = 1, 2, \cdots, N$. The red dots in Figure 4 are the control points set to identify the road areas. Two control points are set for each scene. The sets of control points in image and world coordinate system are $\left\{P_{2d}^i : p_1^i, p_2^i; i = 1, 2, \cdots, N\right\}$ and $\left\{P_{3d}^i : P_1^i, P_2^i; i = 1, 2, \cdots, N\right\}$ respectively. In Figure 4b, the panoramic image coordinate system is defined as $O_p - u_p v_p$, and the origin O_p is located at the upper left corner of the panoramic image, which is similar to the image coordinate system.



(**a**)

(**b**)

Figure 4. Schematic diagram of road space fusion in cross-camera scenes. (**a**) The unified world coordinate system; (**b**) The panoramic image of road space.

Schematic diagram of road distribution in the panoramic image is shown in Figure 5. The proposed road space fusion algorithm in cross-camera scenes is specifically illustrated with this figure.

Step 1: Camera calibration. The calibration method proposed in this paper is used to calculate calibration parameters of each camera in the scene, including internal parameter matrix K_i, rotation matrix R_i, translation matrix T_i and projection matrix of each camera $H_i = K_i R_i T_i; i = 1, 2, \cdots, N$.

Step 2: Road area identification by setting control points. Harris corner extraction algorithm is used to obtain the image coordinate set of the nearest and furthest marking endpoints on the road plane in each scene, which is denoted as $\left\{P_{2d}^i : p_1^i = (x_1^i, y_1^i), p_2^i = (x_2^i, y_2^i); i = 1, 2, \cdots, N\right\}$. Equation (8) is used

to convert P^i_{2d} to the world coordinate set $\left\{P^i_{3d} : P^i_1 = (X^i_1, Y^i_1, 0), P^i_2 = (X^i_2, Y^i_2, 0); i = 1, 2, \cdots, N\right\}$. The range of road area is calculated from P^i_{3d} as $\left\{R^i_l : \left|Y^i_2 - Y^i_1\right|; i = 1, 2, \cdots, N\right\}$.

Step 3: Set control parameter groups and divide pixels of the panoramic image M_p into corresponding scenes. The width of the road is w (mm). The scale in the road space along the width direction is r_w (pixel/mm) and the length direction r_l. The height and width of M_p are wr_w and $r_l \sum_{i=1}^{N} \left|Y^i_2 - Y^i_1\right|$, where the corresponding length of each scene on the panoramic image M_p is $r_l\left|Y^i_2 - Y^i_1\right|; i = 1, 2, \cdots, N$.

Step 4: Generate the complete panoramic image M_p. The panoramic image coordinates are traversed from the origin at the upper left corner. A point (u, v) in the image coordinate system belongs to scene i and its corresponding world coordinate point is $(X^i_1 + v/r_w - w/2, Y^i_1 + (u - R_i)/r_l, 0)$, where $R_i = \begin{cases} 0 & i = 1 \\ \sum_{t=1}^{i-1} r_l\left|Y^t_2 - Y^t_1\right| & i = 2, 3, \cdots, N \end{cases}$.

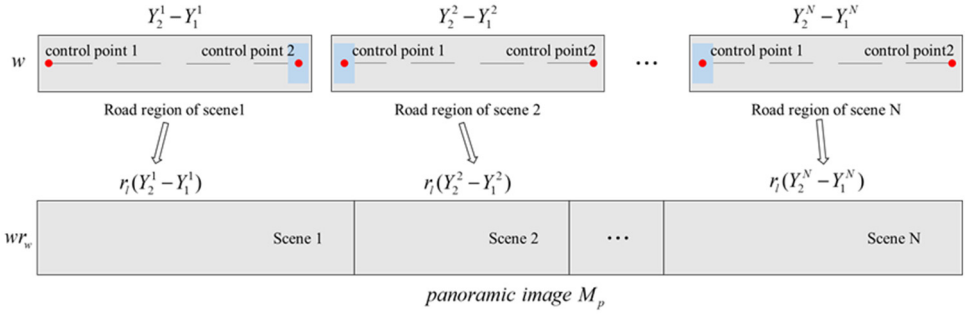


Figure 5. Schematic diagram of road distribution in the panoramic image.

The pixel in the road area I_{pixel} corresponding to the world coordinate point is taken out (if any) and put to the position of the panoramic image coordinate point. Repeat this process until all the pixels of the corresponding road areas in all scenes are taken out and put into the panoramic image correctly.

Since the generated panoramic image contains physical information of road space, the position in the sub-scene world coordinate system and the unified world coordinate system can be calculated directly from a point in the panoramic image. In addition, the position in the unified world coordinate system and the panoramic image coordinate system can also be analyzed from a point in the sub-scene world coordinate system. The specific mapping equation group is as follows:

- panoramic image-to-world

$$\begin{cases} Unified\ world\ coordinate & (v/r_w + X^i_1 - w/2, u/r_l + Y^1_1, 0) \\ Subscene\ world\ coordinates & (v/r_w + X^i_1 - w/2, (u - R_i)/r_l + Y^i_1, 0) \end{cases}, \tag{9}$$

where a point in the panoramic image is denoted as (u, v), i represents the number of the sub-scene, $R_i = \begin{cases} 0 & i = 1 \\ \sum_{t=1}^{i-1} r_l\left|Y^t_2 - Y^t_1\right| & i = 2, 3, \cdots, N \end{cases}$.

- Sub-scene world-to-panoramic image

$$panoramic\ image\ coordinate \quad (r_l(Y - Y^i_1 + U_i), r_w[X - (X^i_1 - w/2)]), \tag{10}$$

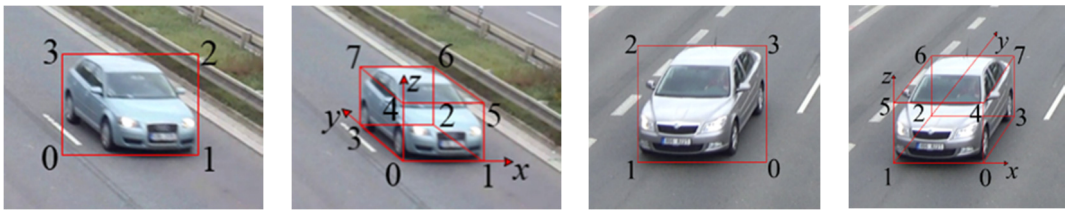
where a point in sub-scene i is denoted as $(X, Y, 0)$, $U_i = \begin{cases} 0 & i = 1 \\ \sum_{t=1}^{i-1}\left|Y^t_2 - Y^t_1\right| & i = 2, 3, \cdots, N \end{cases}$.

2.3. 3D Vehicle Detection for Distribution and Trajetory Extraction

2.3.1. 3D Bounding Boxes and Projection Centroids of Vehicles

Based on road space fusion in cross-camera scenes, to further obtain vehicle spatial distribution and 3D trajectory, vehicle detection in the scene is needed. Since the height of vehicle feature points is unknown, projection centroid is adopted in this paper instead, which depends on 3D vehicle detection. Considering actual application requirements, we choose YOLOv4 [34] for 2D vehicle detection. The detection results contain center point, width, and height of 2D bounding box in the image coordinate system, vehicle type (car, truck, bus) and its confidence. Then, the best 3D vehicle detection result and projection centroid are obtained by geometric constraints for vehicle spatial distribution and 3D trajectory extraction.

Figure 6 shows the vehicle model of 2D/3D bounding box from left and right perspectives. In each sub-figure, the left represents 2D model while the right 3D model. 2D model is in the image coordinate system. The axes in 3D model are the same direction as the world coordinate system, and the origin is the bottom left point of the 3D model. The vertices of 2D bounding box model are numbered from 0 to 3, and the corresponding image coordinates are denoted as $P_i^{2D} = (u_i^{2D}, v_i^{2D}), i = 0, 1, 2, 3$. In the same way, the vertices of 3D bounding box model are numbered from 0 to 7, and the corresponding world and image coordinates are denoted as $P_i^{3D} = (x_i^{3D}, y_i^{3D}, z_i^{3D}), i = 0, 1, \cdots, 7$ and $P_j^{3Di} = (u_j^{3Di}, v_j^{3Di}), j = 0, 1, \cdots, 7$. The world coordinates of eight vertices and projection centroid of the vehicle from different perspectives are presented in Table 1.



(**a**) (**b**)

Figure 6. Schematic diagram of vehicle model of 2D/3D bounding box from different perspectives. (**a**) Left perspective; (**b**) Right perspective.

Table 1. Eight vertices and projection centroid in 3D bounding box model from different perspectives.

Number	Perspective	
	Left	**Right**
0	$\left(x_1^{3D} - w_v, y_1^{3D}, z_1^{3D}\right)$	$\left(x_1^{3D} + w_v, y_1^{3D}, z_1^{3D}\right)$
1	$\left(x_1^{3D}, y_1^{3D}, z_1^{3D}\right)$	$\left(x_1^{3D}, y_1^{3D}, z_1^{3D}\right)$
2	$\left(x_1^{3D}, y_1^{3D} + l_v, z_1^{3D}\right)$	$\left(x_1^{3D}, y_1^{3D} + l_v, z_1^{3D}\right)$
3	$\left(x_1^{3D} - w_v, y_1^{3D} + l_v, z_1^{3D}\right)$	$\left(x_1^{3D} + w_v, y_1^{3D} + l_v, z_1^{3D}\right)$
4	$\left(x_1^{3D} - w_v, y_1^{3D}, z_1^{3D} + h_v\right)$	$\left(x_1^{3D} + w_v, y_1^{3D}, z_1^{3D} + h_v\right)$
5	$\left(x_1^{3D}, y_1^{3D}, z_1^{3D} + h_v\right)$	$\left(x_1^{3D}, y_1^{3D}, z_1^{3D} + h_v\right)$
6	$\left(x_1^{3D}, y_1^{3D} + l_v, z_1^{3D} + h_v\right)$	$\left(x_1^{3D}, y_1^{3D} + l_v, z_1^{3D} + h_v\right)$
7	$\left(x_1^{3D} - w_v, y_1^{3D} + l_v, z_1^{3D} + h_v\right)$	$\left(x_1^{3D} + w_v, y_1^{3D} + l_v, z_1^{3D} + h_v\right)$
Projection centroid	$\left(x_1^{3D} - w_v/2, y_1^{3D} + l_v/2, z_1^{3D}\right)$	$\left(x_1^{3D} + w_v/2, y_1^{3D} + l_v/2, z_1^{3D}\right)$

Schematic diagram of 2D/3D vehicle detection is shown in Figure 7 (the left represents 2D detection while the right 3D detection) and the algorithm is specifically described as follows:

Step 1: YOLOv4 is used to obtain the vertices in the image coordinate system $P_i^{2D} = (u_i^{2D}, v_i^{2D})$, i = 0,1, 2, 3 and vehicle type. The base point of 2D bounding box is set as $P_1^{2D} = (u_1^{2D}, v_1^{2D})$ in the image

coordinate system, which can be converted into $P_1^{3D} = (x_1^{3D}, y_1^{3D}, z_1^{3D})$ in the world coordinate system by Equation (8), where $z_1^{3D} = 0$.

Step 2: Suppose 3D vehicle physical size (l_v, w_v, h_v), l_v,w_v,h_v represent vehicle length, width, and height respectively. According to Table 1, the world coordinates of the eight vertices in 3D bounding box model are calculated as $P_i^{3D} = (x_i^{3D}, y_i^{3D}, z_i^{3D}), i = 0, 1, \cdots, 7$.

Step 3: The calculation results in Step 2 are converted to the image coordinates $P_j^{3Di} = (u_j^{3Di}, v_j^{3Di})$, j = 0,1, $\cdots$,7 through Equation (7) to complete 3D vehicle detection.

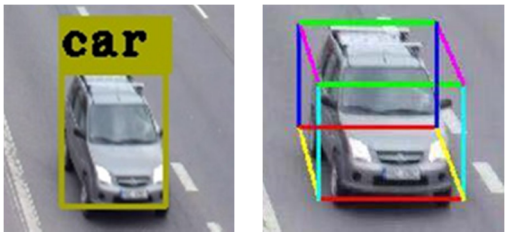


Figure 7. Schematic diagram of 2D (**left**)/3D (**right**) vehicle detection.

2.3.2. Geometric Constraints

According to the above 3D vehicle detection algorithm, obtaining accurate 3D vehicle physical size is the premise to complete precise 3D vehicle detection. Due to the factor of perspective distortion and lack of depth information in monocular image, accurate size cannot be obtained by vehicle type which is derived from YOLOv4. Therefore, geometric constraints are considered to accurately calculate 3D vehicle physical size, which includes diagonal constraint and vanishing point constraint.

3D vehicle detection is equivalent to obtaining 3D vehicle physical size $X = (l_v, w_v, h_v)$, and the diagonal pixel length of 2D bounding box is defined as:

$$l_{2D} = \left\| P_1^{2D} - P_3^{2D} \right\|_2, \tag{11}$$

where $\| \cdot \|_2$ denotes the Euclidean distance between two points.

According to 3D bounding box model, P_1^{3Di} and P_7^{3Di} are selected, and the diagonal pixel length of 3D bounding box can also be defined as:

$$l_{3D} = \left\| P_1^{3Di} - P_7^{3Di} \right\|_2, \tag{12}$$

The difference of Equation (11) and (12) consists of a set of diagonal constraint. Figure 8 shows the vehicle diagonal constraint. The red/yellow wireframe represents 2D/3D bounding box. When 2D bounding box and 3D bounding box are completely fitted, the blue line segment indicates that the 2D/3D diagonals completely coincide in the image coordinate system and the value of diagonal constraint is 0, which means 3D vehicle physical size is relatively accurate. The word relatively means the size is accurate in the case of diagonal constraint.

According to the principle of vanishing point, the straight line composed of 0–3, 1–2, 4–7, 5–6 point pairs in the 3D bounding box model must pass the vanishing point along the road direction in the image coordinate system. Therefore, it can be used as another set of constraints to accurately calculate 3D vehicle physical size.

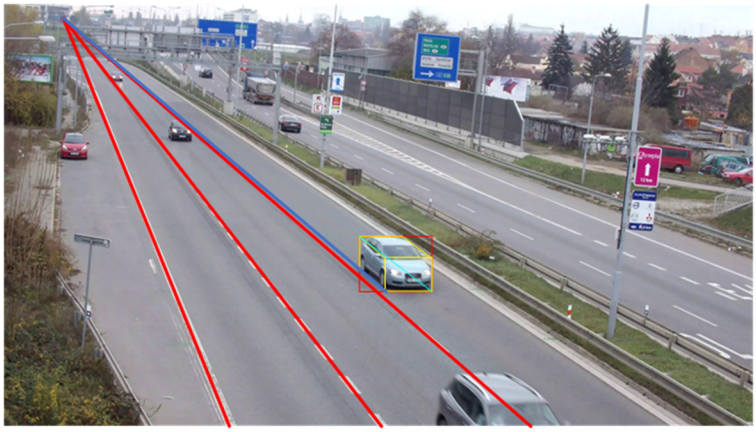


Figure 8. Schematic diagram of geometric constraints.

In the image coordinate system, the included angle between two lines (one formed by point pairs, the other formed by one point and the vanishing point along the road direction) can be denoted as θ.

For four point pairs, according to the cosine theorem, we can derive $\theta_1, \theta_2, \theta_3, \theta_4$ as follows:

$$\cos\theta_1 = \frac{\left\|P_0^{3\mathrm{Di}} - P_3^{3\mathrm{Di}}\right\|_2^2 + \left\|P_0^{3\mathrm{Di}} - \mathrm{VP}\right\|_2^2 - \left\|P_3^{3\mathrm{Di}} - \mathrm{VP}\right\|_2^2}{2 \cdot \left\|P_0^{3\mathrm{Di}} - P_3^{3\mathrm{Di}}\right\|_2^2 \cdot \left\|P_0^{3\mathrm{Di}} - \mathrm{VP}\right\|_2^2}, \tag{13}$$

$$\cos\theta_2 = \frac{\left\|P_1^{3\mathrm{Di}} - P_2^{3\mathrm{Di}}\right\|_2^2 + \left\|P_1^{3\mathrm{Di}} - \mathrm{VP}\right\|_2^2 - \left\|P_2^{3\mathrm{Di}} - \mathrm{VP}\right\|_2^2}{2 \cdot \left\|P_1^{3\mathrm{Di}} - P_2^{3\mathrm{Di}}\right\|_2^2 \cdot \left\|P_1^{3\mathrm{Di}} - \mathrm{VP}\right\|_2^2}, \tag{14}$$

$$\cos\theta_3 = \frac{\left\|P_4^{3\mathrm{Di}} - P_7^{3\mathrm{Di}}\right\|_2^2 + \left\|P_4^{3\mathrm{Di}} - \mathrm{VP}\right\|_2^2 - \left\|P_7^{3\mathrm{Di}} - \mathrm{VP}\right\|_2^2}{2 \cdot \left\|P_4^{3\mathrm{Di}} - P_7^{3\mathrm{Di}}\right\|_2^2 \cdot \left\|P_4^{3\mathrm{Di}} - \mathrm{VP}\right\|_2^2}, \tag{15}$$

$$\cos\theta_4 = \frac{\left\|P_5^{3\mathrm{Di}} - P_6^{3\mathrm{Di}}\right\|_2^2 + \left\|P_5^{3\mathrm{Di}} - \mathrm{VP}\right\|_2^2 - \left\|P_6^{3\mathrm{Di}} - \mathrm{VP}\right\|_2^2}{2 \cdot \left\|P_5^{3\mathrm{Di}} - P_6^{3\mathrm{Di}}\right\|_2^2 \cdot \left\|P_5^{3\mathrm{Di}} - \mathrm{VP}\right\|_2^2} \tag{16}$$

The sum of four equations above consists of a set of vanishing point constraint. As shown in Figure 8, the red line segment is used to extract the vanishing point. When 2D bounding box and 3D bounding box are completely fitted, the deep blue line shows that the line formed by point pairs and the vanishing point completely coincide in the image coordinate system and the value of vanishing point constraint is 0, which means 3D vehicle physical size is relatively accurate. The word relatively means the size is accurate in the case of vanishing point constraint.

In this paper, the steps to obtain the vehicle geometric constraints are as follows:

Step 1: YOLOv4 is used to obtain the vertices in the image coordinate system $P_i^{2\mathrm{D}} = (u_i^{2\mathrm{D}}, v_i^{2\mathrm{D}}), i = 0, 2, 3$, base point $P_1^{2\mathrm{D}} = (u_1^{2\mathrm{D}}, v_1^{2\mathrm{D}})$ and vehicle type.

Step 2: $(l_\mathrm{v}, w_\mathrm{v}, h_\mathrm{v})$ is considered to be a set of unknown parameters. The base point in the world coordinate system can be obtained by Equation (8) as $P_1^{3\mathrm{D}} = (x_1^{3\mathrm{D}}, y_1^{3\mathrm{D}}, z_1^{3\mathrm{D}})$, where $z_1^{3\mathrm{D}} = 0$. Then, According to Table 1, the world coordinates $P_0^{3\mathrm{D}}$, $P_2^{3\mathrm{D}}$ to $P_7^{3\mathrm{D}}$ can be calculated.

Step 3: According to Equation (11), the diagonal pixel length of 2D bounding box is calculated. Then, the world coordinates of vertex 1 and 7 are converted to the image coordinates according to Equation (7) as $P_1^{3\mathrm{Di}}, P_7^{3\mathrm{Di}}$. Finally, the diagonal pixel length of 3D vehicle bounding box is calculated according to Equation (12), and a set of diagonal constraints are formed.

Step 4: According to Equation group (8), the world coordinates of vertices from 0 to 7 are converted to image coordinates as P_0^{3Di} to P_7^{3Di}. The values of $\cos\theta_1$ to $\cos\theta_4$ can be calculated according to Equations (13) to (16), and a set of vanishing point constraints are formed.

According to the above algorithm, the diagonal constraint and vanishing point constraint are obtained to construct the constraint error as $l_{\text{cal}} - l_{\text{truth}}$. Where l_{cal} is the actual constraint value obtained by calculation, and l_{truth} is the ideal constraint value when 2D bounding box and 3D bounding box are completely fitted. By analyzing the above algorithm, it can be easily seen that the variables in the constraint error are composed of parameters $l_{\text{v}}, w_{\text{v}}, h_{\text{v}}$, which can constitute the nonlinear constraint space of parameter vectors.

To sum up, the nonlinear constraint function of the parameter $X = (l_{\text{v}}, w_{\text{v}}, h_{\text{v}})$ is:

$$\underset{X}{\text{argmin}} \frac{1}{2} \left(\sum_{i=1}^{N_f} \lambda_d (l_{2D} - l_{3D})^2 + \sum_{j=1}^{4} \lambda_v (\cos\theta_j - 4)^2 \right), \tag{17}$$

where N_f is the occurrence time of the same vehicle in video frames, λ_{d} and λ_{v} respectively represent the error coefficient of the diagonal constraint and vanishing point constraint which are usually set to 1 and can be adjusted in different conditions, and $\min_X$ represents the value of X when the constraint function reaches the minimum.

The constraint function is nonlinear. LM (Levenberg-Marquardt) method is adopted in this paper to solve the constraint function, which is easy to reach convergence. The initial value X_0 can be obtained by referring to the national road vehicle size standard [35] based on the vehicle type derived by YOLO.

After solving accurate 3D vehicle physical size, 3D vehicle detection can be completed. Then, the world coordinates of projection centroids can be calculated. According to Equations (9) and (10), coordinates of vehicles in the panoramic image and other scenes can be obtained. As shown in Figure 9, vehicle spatial distribution and 3D trajectory in cross-camera scenes can be obtained by vehicles in continuous motion.

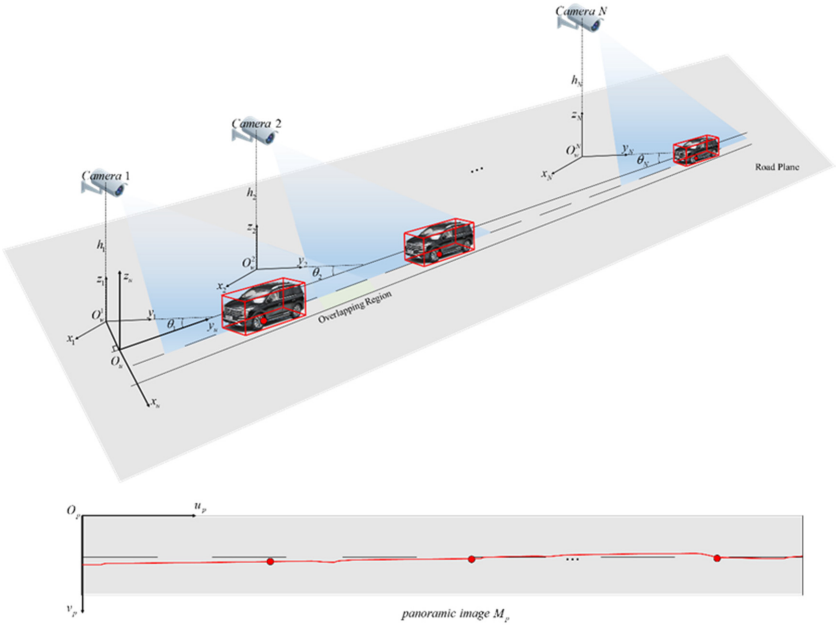


Figure 9. Schematic diagram of vehicle spatial distribution and 3D trajectory extraction in cross-camera scenes.

3. Results

In our experiments, we used the Intel Core i7-8700 CPU, NVIDIA 1080Ti GPU (Graphics Processing Unit), 32GB memory, and Windows 10 operating system. The open source framework Darknet is used for vehicle detection.

Experiments are carried out on the public dataset BrnoCompSpeed [36] and actual road scene respectively, and the algorithm illustrated in Section 2 is adopted in the experiments. First, road space fusion algorithm in cross-camera scenes is used to generate the panoramic image of road with spatial information. Secondly, YOLOv4 combined with geometric constraints is used for 3D vehicle detection to obtain projection centroids. Finally, the projection centroids are projected to the panoramic image to derive vehicle spatial distribution and 3D trajectories. The experiments can be divided into the following two aspects: (1) Verify the accuracy of projection centroids obtained by 3D vehicle detection algorithm for vehicle spatial distribution. (2) Compare the proposed 3D vehicle trajectory extraction algorithm with several 3D tracking methods in this paper.

3.1. BrnoCompSpeed Dataset Single-Camera Scene

Due to the lack of cross-camera datasets from road surveillance perspectives, we choose a public dataset of single-camera scenes from surveillance perspectives published by researchers of Brno University of Technology for our experiments. The cross-camera dataset made by ourselves and experiments carried out on this scene are described in detail in Section 3.2.

The public dataset BrnoCompSpeed contains six traffic scenes captured by roadside surveillance cameras. Each scene can be divided into left, middle, and right perspectives, with a total of 18 HD (High Definition) videos (about 200 GB). The resolution of all the videos is 1920 × 1080. The dataset contains various types of vehicles such as hatch-back, sedan, SUV, truck and bus, and the position and velocity of vehicles are accurately recorded by radar. Therefore, this dataset can be used to verify the accuracy of vehicle spatial distribution and 3D trajectories in single-camera scenes.

As shown in Figure 10, we select three scenes of different perspectives from six scenes for verification which do not contain winding roads. In all the three scenes, the width of a single lane is 3.5 m, the length of a single short white marking line is 1.5 m, the length of a single long white marking line is 3 m, and the length between the starting points of the long white marking lines is 9 m. First, the three scenes are calibrated separately. Calibration results are shown in Table 2. Based on calibration, the road space fusion algorithm described in Section 2.2.2 is adopted to generate the panoramic image with physical information. Since the scenes in the dataset are single-camera scenes, we generate a roadblock containing physical information for convenience which is shown in Figure 11. Each small square of the roadblock represents the actual road space size of 3.5 × 9 m.

(**a**) (**b**) (**c**)

Figure 10. Dataset road scene. (**a**) Scene 1; (**b**) Scene 2; (**c**) Scene 3.

Table 2. Camera calibration results of dataset road scene.

Parameter \ Scene	Scene 1	Scene 2	Scene 3
f	2878.13	3994.17	3384.25
ϕ/rad	0.17874	0.15717	0.26295
θ/rad	0.26604	0.03535	−0.24869
h/mm	10119.08	8071.00	8126.49
VP	(144.74, 34.78)	(812.62, −109.12)	(1855.68, −373.44)
H	$\begin{bmatrix} 3025.25 & 154.76 & -170.68 & 1.727\times10^6 \\ 5.17 & 18.99 & -2928.28 & 2.963\times10^7 \\ 0.26 & 0.95 & -0.18 & 1799.11 \end{bmatrix}$	$\begin{bmatrix} 4025.18 & 806.43 & -150.27 & 1.213\times10^6 \\ -3.25 & -91.80 & -4029.46 & 3.252\times10^7 \\ 0.03 & 0.99 & -0.16 & 1263.33 \end{bmatrix}$	$\begin{bmatrix} 3051.98 & 1731.45 & -249.54 & 2.028\times10^6 \\ 88.18 & -347.23 & -3408.29 & 2.770\times10^7 \\ -0.24 & 0.94 & -0.26 & 2112.36 \end{bmatrix}$

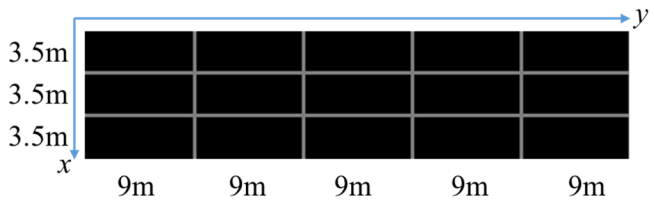


Figure 11. Schematic diagram of the roadblock.

The real position of the vehicle in the world coordinate system is defined as P_r and the measured position is P_m. The effective field of view of the scene is set to L_s(m). Then, the vehicle spatial distribution error can be defined as:

$$error = \frac{||P_r - P_m||_2}{L_s/2} \times 100\%, \tag{18}$$

Examples of the vehicle spatial distribution and 3D trajectories in dataset scenes are shown in Figure 12. In this experiment, L_s is set to 450 m, and the base point in scene 2 can be selected using either left or right perspective. Each scene contains multiple vehicles, and there are some cases of vehicle occlusion. For each instance, the top image contains 3D vehicle detection and 2D trajectory results, and the roadblock on the bottom side contains vehicle spatial distribution and 3D trajectory results. Each vehicle corresponds to one color without repetition. Tables 3–5 correspond to the 3D physical size, the image, and world coordinates and spatial distribution error of each vehicle in dataset scene 1 to scene 3. The value of y-axis in the world coordinate system is presented in an ascending order which indicates the distance between the vehicle and the camera is from near to far. To present the results in a straightforward way, the position and direction of the vehicle is marked in the roadblock with a white line segment and a white arrow respectively.

Table 3. Measurement of vehicle spatial distribution errors in dataset scene 1.

Instance		Size/m	Image Coordinate	World Coordinate/mm	Error/%
1	Car1	(4.30, 1.80, 1.40)	[963, 861]	[8956.79, 32,729.00, 0]	1.40%
	Car2	(4.30, 1.80, 1.35)	[676, 597]	[8340.02, 49,855.60, 0]	1.85%
	Car3	(4.30, 1.80, 1.35)	[778, 510]	[11,971.37, 58,457.92, 0]	2.54%
	Car4	(4.20, 1.60, 1.35)	[993, 456]	[18,405.90, 64,583.41, 0]	3.66%
2	Car1	(4.40, 1.80, 1.45)	[773, 692]	[8525.12, 42,167.95, 0]	0.86%
	Car2	(4.30, 1.70, 1.40)	[836, 588]	[11,274.48, 49,911.95, 0]	1.61%
	Truck1	(11.00, 2.70, 2.80)	[901, 358]	[21,181.46, 84,556.09, 0]	3.89%
3	Truck1	(20.00, 2.80, 3.80)	[817, 708]	[8950.18, 40,973.38, 0]	2.12%
	Car1	(4.20, 1.60, 1.35)	[1291, 603]	[18,714.95, 46,472.42, 0]	1.39%

Table 4. Measurement of vehicle spatial distribution errors in dataset scene 2.

Instance		Size/m	Image Coordinate	World Coordinate/mm	Error/%
1	Car1	(4.50, 1.80, 1.50)	[868, 881]	[382.40, 32,656.53, 0]	1.60%
	Car2	(4.50, 1.80, 1.50)	[1276, 859]	[3892.26, 33,316.97, 0]	1.44%
	Car3	(4.50, 1.90, 1.65)	[828, 380]	[144.69, 68,624.58, 0]	2.80%
	Car4	(4.50, 1.90, 1.55)	[1381, 310]	[11,383.98, 80,370.24, 0]	3.35%
2	Truck1	(11.00, 2.90, 2.80)	[539, 595]	[−3345.10, 46,901.24, 0]	2.03%
	Car1	(4.25, 1.70, 1.35)	[1455, 395]	[10,631.62, 66,104.90, 0]	2.34%
	Car2	(4.30, 1.80, 1.40)	[859, 380]	[679.93, 68,605.65, 0]	1.75%
	Car3	(4.25, 1.70, 1.35)	[1451, 285]	[13,652.22, 85,716.80, 0]	3.84%
3	Car1	(4.40, 1.80, 1.35)	[835, 793]	[120.70, 36,037.78, 0]	1.63%
	Car2	(4.30, 1.70, 1.40)	[798, 515]	[−300.43, 53,117.64, 0]	1.75%
	Car3	(4.30, 1.80, 1.35)	[1539, 453]	[10,753.86, 58,902.55, 0]	2.03%
	Car4	(4.40, 1.70, 1.30)	[1038, 418]	[3486.72, 63,307.65, 0]	3.14%
	Car5	(4.30, 1.60, 1.30)	[1582, 374]	[13,332.64, 69,056.44, 0]	3.16%
	Car6	(4.30, 1.70, 1.40)	[836, 307]	[342.68, 81,376.07, 0]	3.18%

Table 5. Measurement of vehicle spatial distribution errors in dataset scene 3.

Instance		Size/m	Image Coordinate	World Coordinate/mm	Error/%
1	Car1	(4.50, 1.70, 1.50)	[1227, 950]	[−3308.45, 19,939.27, 0]	0.31%
	Truck1	(8.90, 2.50, 2.40)	[1054, 479]	[−7100.23, 31,741.25, 0]	1.60%
	Car2	(4.40, 1.60, 1.40)	[1310, 287]	[−6155.65, 42,427.68, 0]	0.57%
	Car3	(4.20, 1.60, 1.40)	[1395, 145]	[−6654.15, 55,028.20, 0]	1.23%
2	Car1	(5.40, 1.90, 1.85)	[1390, 624]	[−3232.59, 27,506.14, 0]	0.72%
	Car2	(4.50, 1.70, 1.50)	[996, 449]	[−7956.65, 32,833.57, 0]	0.93%
	Car3	(5.15, 1.80, 1.70)	[1548, 243]	[−3473.58, 46,422.95, 0]	1.11%
	Car4	(4.30, 1.60, 1.50)	[1120, 202]	[−9853.38, 48,349.16, 0]	1.31%
3	Truck1	(8.90, 2.30, 2.70)	[1074, 545]	[−6371.75, 29,346.67, 0]	1.67%
	Car1	(4.30, 1.70, 1.40)	[1485, 195]	[−4721.74, 50,309.12, 0]	1.72%

From the experimental results, it can be seen that the average error of vehicle spatial distribution within the scope of hundred meters is less than 5%, which means the accuracy can reach the centimeter level. In the meanwhile, the proposed algorithm is also adaptable to the situation of part vehicle occlusion.

3.2. Actual Road Cross-Camera Scene

To further verify the application ability of the proposed algorithm, we choose the actual road with large traffic flow which is located on the Middle Section of South Second Ring Road in Xi'an, ShaanXi Province, China to make a small dataset of cross-camera scenes. The dataset consists of three groups of HD videos (a total of six videos), and each of which is about 0.5 h long. The resolution of all the videos is 1280 × 720. Figure 13 shows the image of the actual road scenes with no overlapping area which are taken by 2 cameras with a distance of 210 m. In the actual road scene, the road width is 7.5 m, the length of a single white marking line on the road plane is 6m, and the length between the starting points of the white marking lines is 11.80 m and 11.39 m in two scenes respectively. First, the scenes taken by two cameras are calibrated separately. Calibration results are shown in Table 6. Based on calibration, the panoramic image with physical information is generated by the road space fusion algorithm described in Section 2.2.2, which is shown in Figure 14. A degree scale in the image represents an actual distance of the starting points of four white marking lines and 3.75 m in the image width and height direction.

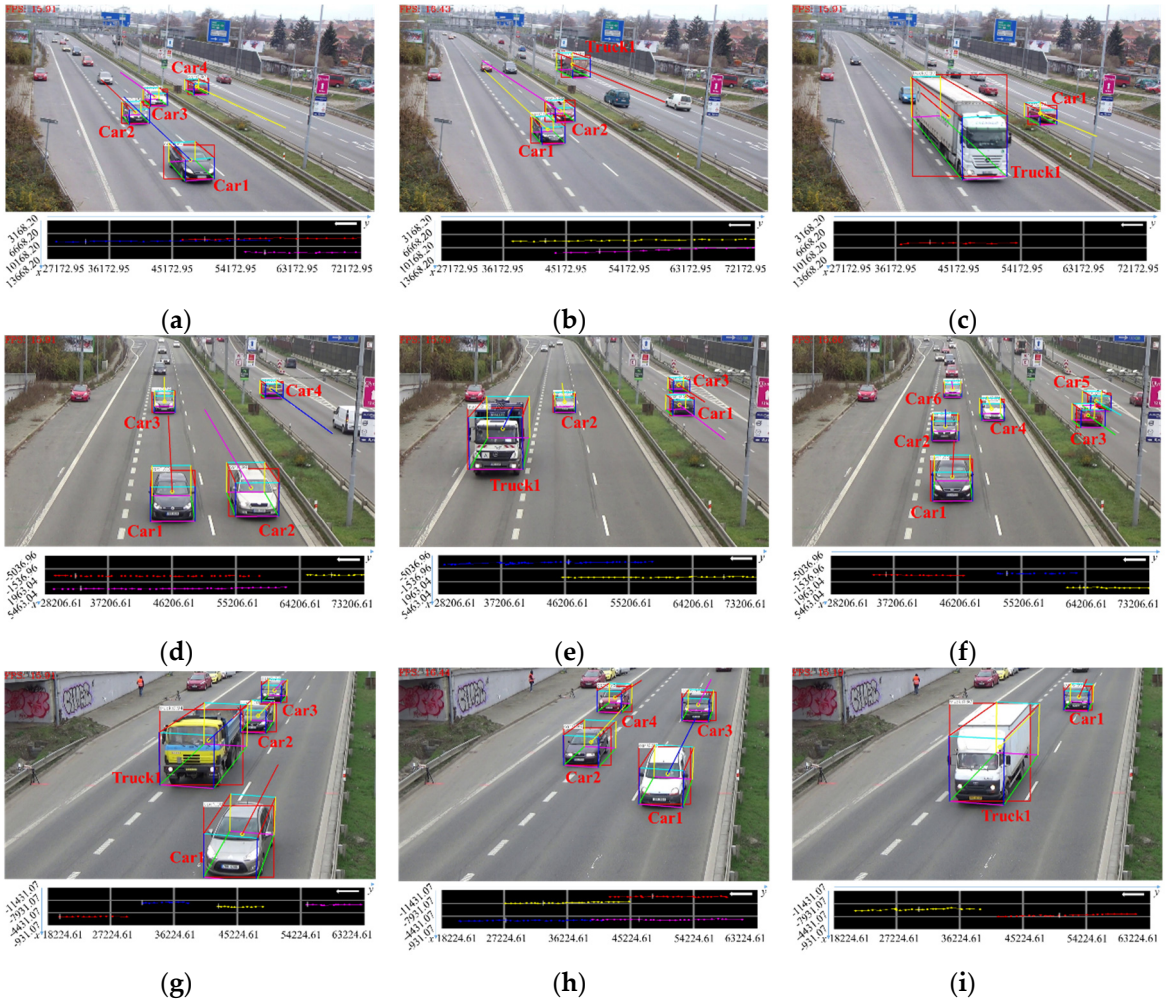


Figure 12. Examples of vehicle spatial distribution and 3D trajectory extraction in dataset road scene. (**a**) Scene1-example1; (**b**) Scene1-example2; (**c**) Scene1-example3; (**d**) Scene2-example1; (**e**) Scene2-example2; (**f**) Scene2-example3; (**g**) Scene3-example1; (**h**) Scene3-example2; (**i**) Scene3-example3.

Figure 13. Actual road scene. (**a**) Camera 1; (**b**) Camera 2.

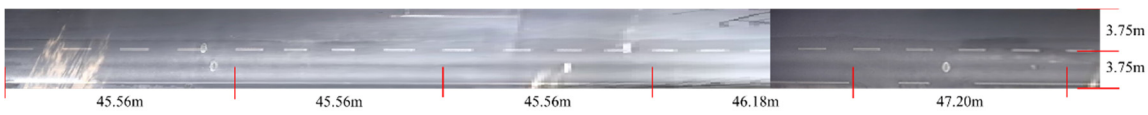


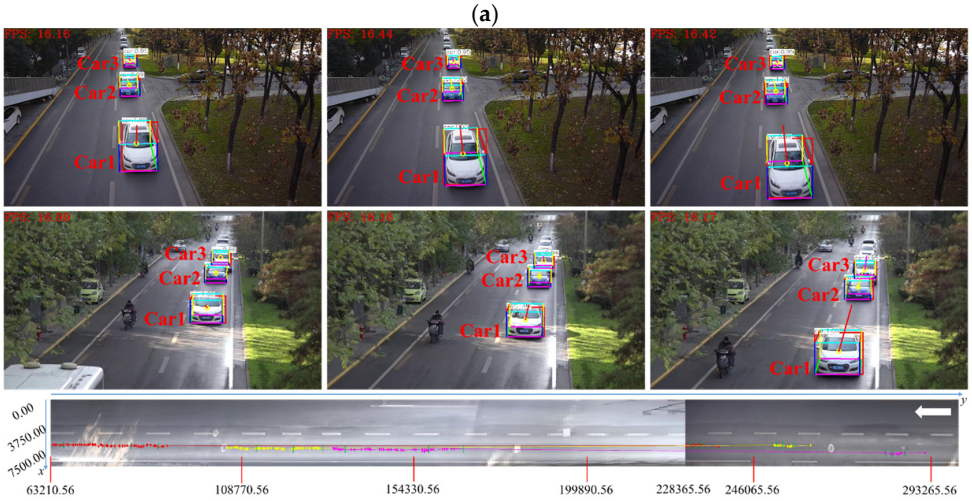
Figure 14. The panoramic image of actual road scene.

Table 6. Camera calibration results of actual road scene.

Parameter \ Camera	Camera 1	Camera 2
f	1853.22	5749.81
ϕ/rad	0.21361	0.07326
θ/rad	0.09411	−0.04820
h/mm	7950.72	7877.36
VP	(461, −42)	(921, −62)
H	$\begin{bmatrix} 1903.79 & 448.53 & -135.67 & 1.079\times10^{6} \\ -3.86 & -40.86 & -1887.41 & 1.501\times10^{7} \\ 0.092 & 0.97 & -0.21 & 1685.47 \end{bmatrix}$	$\begin{bmatrix} 5712.37 & 914.59 & -46.85 & 3.690\times10^{5} \\ 2.98 & -61.76 & -5760.74 & 4.538\times10^{7} \\ -0.048 & 0.996 & -0.732 & 576.60 \end{bmatrix}$

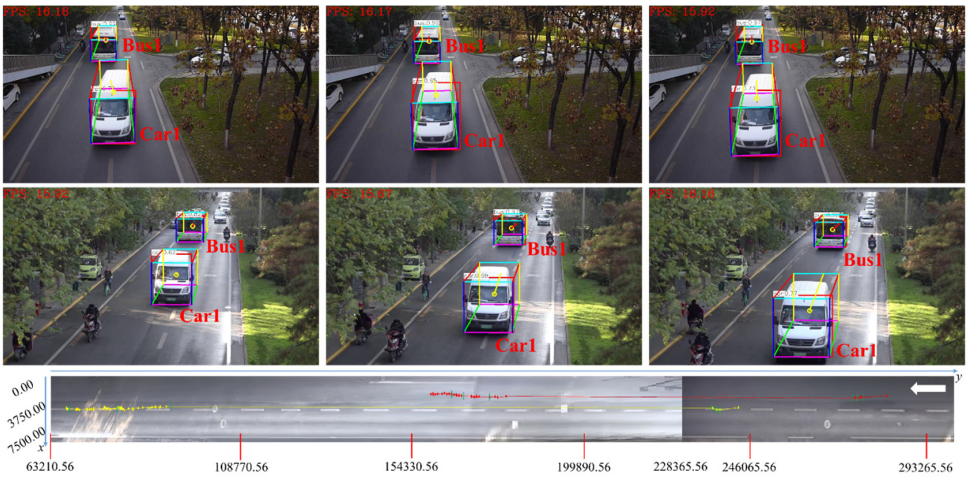
In our experiment, we choose three examples of vehicles, which are shown in Figure 15. For each example (similar to the dataset scene), 3D vehicle detection results in two cameras are shown in the first two lines respectively, and 3D vehicle trajectory extraction results are shown in the third line. Each vehicle corresponds to one color without repetition. Table 7 shows the results of vehicle spatial distribution in actual road scene. Similar to the single-camera scenes, we mark the position and direction of the vehicle in the panoramic image with a green line segment and a white arrow, respectively. From the experimental results, it can be seen that continuous 3D trajectories of vehicles in cross-camera scenes can be effectively extracted.

(**a**)



(**b**)

Figure 15. *Cont.*



(**c**)

Figure 15. Examples of vehicle spatial distribution and 3D trajectory extraction in actual road scene. (**a**) Example 1; (**b**) Example 2; (**c**) Example 3.

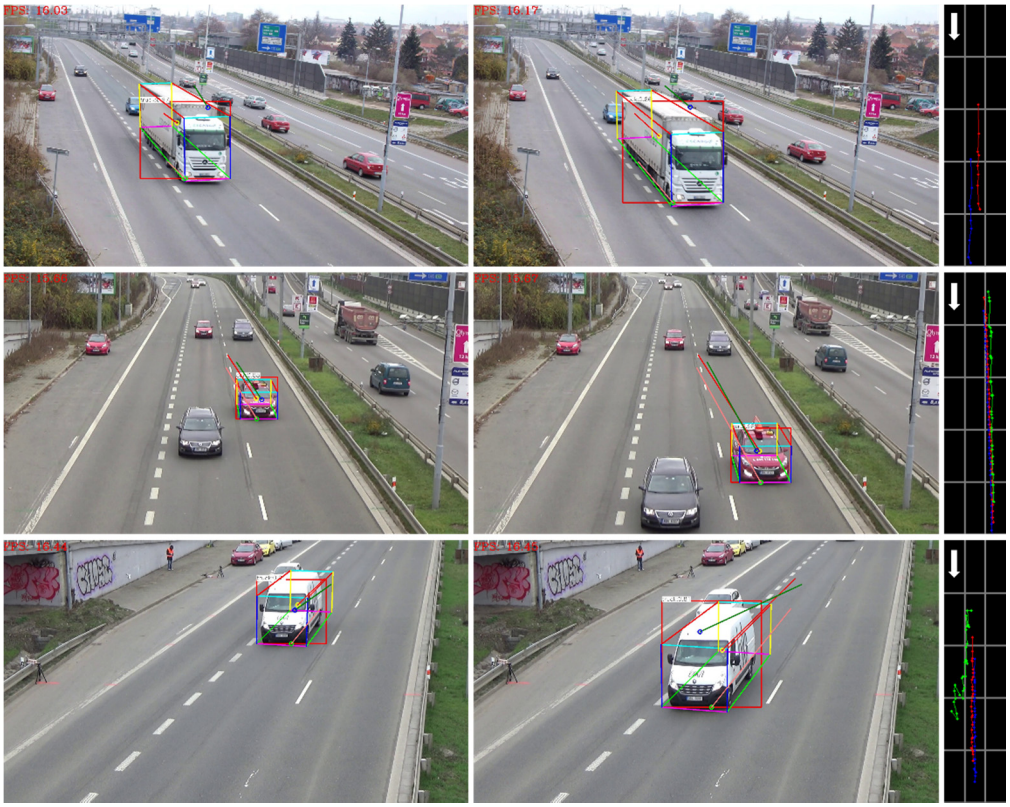
Table 7. Vehicle spatial distribution in actual road scene.

Example	Position		World Coordinate in Camera 1/mm	World Coordinate in Camera 2/mm	Unified World Coordinate/mm
1	1-4117	Car1	[182.59, 34,884.74, 0]	not appear in camera 2	[4388.75, 241,500.13, 0]
		Car2	[1214.55, 59,403.27, 0]	not appear in camera 2	[5420.77, 266,018.66, 0]
	1-4122	Car1	[276.22, 31,783.392, 0]	not appear in camera 2	[4482.38, 238,398.79, 0]
		Car2	[1343.44, 55,747.96, 0]	not appear in camera 2	[5549.60, 262,363.36, 0]
	1-4138	Car1	[433.30, 21,957.72, 0]	not appear in camera 2	[4639.462, 228,573.11, 0]
		Car2	[1372.33, 46,263.32, 0]	not appear in camera 2	[5578.49, 252,878.72, 0]
	2-4624	Car1	not appear in camera 1	[−2315.95, 134,591.43, 0]	[4583.15, 134,591.43, 0]
		Car2	not appear in camera 1	[−1860.88, 166,941.60, 0]	[5038.22, 166,941.60, 0]
	2-4660	Car1	not appear in camera 1	[−2921.01, 109,135.85, 0]	[3978.10, 109,135.85, 0]
		Car2	not appear in camera 1	[−1832.96, 146,397.29, 0]	[5066.15, 146,397.29, 0]
	2-4712	Car1	not appear in camera 1	[−4233.11, 70,563.03, 0]	[2666.00, 70,563.03, 0]
		Car2	not appear in camera 1	[−1901.24, 116,527.55, 0]	[4997.87, 116,527.55, 0]
2	1-2801	Car1	[899.46, 26,003.79, 0]	not appear in camera 2	[5105.62, 232,619.19, 0]
		Car2	[1025.87, 49,825.21, 0]	not appear in camera 2	[5232.03, 256,440.60, 0]
		Car3	[1820.65, 78,805.77, 0]	not appear in camera 2	[6026.81, 285,421.17, 0]
	1-2807	Car1	[993.10, 23,872.62, 0]	not appear in camera 2	[5199.26, 230,488.01, 0]
		Car2	[965.72, 47,215.47, 0]	not appear in camera 2	[5171.89, 253,830.87, 0]
		Car3	[1893.67, 76,750.40, 0]	not appear in camera 2	[6099.84, 283,365.80, 0]
	1-2811	Car1	[980.66, 22,200.39, 0]	not appear in camera 2	[5186.82, 228,815.78, 0]
		Car2	[973.85, 45,420.60, 0]	not appear in camera 2	[5180.01, 252,035.99, 0]
		Car3	[1903.34, 75,569.57, 0]	not appear in camera 2	[6109.50, 282,184.96, 0]
	2-3437	Car1	not appear in camera 1	[−1564.61, 91,261.22, 0]	[5334.49, 91,261.22, 0]
		Car2	not appear in camera 1	[−1469.92, 132,265.81, 0]	[5429.19, 132,265.81, 0]
		Car3	not appear in camera 1	[−1300.03, 165,745.23, 0]	[5599.08, 165,745.23, 0]
	2-3455	Car1	not appear in camera 1	[−1687.04, 82,533.43, 0]	[5212.07, 82,533.43, 0]
		Car2	not appear in camera 1	[−1337.24, 126,348.42, 0]	[5561.87, 126,348.42, 0]
		Car3	not appear in camera 1	[−1239.88, 158,211.49, 0]	[5659.23, 158,211.49, 0]
	2-3490	Car1	not appear in camera 1	[−1787.93, 65,696.12, 0]	[5111.17, 65,696.12, 0]
		Car2	not appear in camera 1	[−1548.42, 111,637.28, 0]	[5350.69, 11,1637.28, 0]
		Car3	not appear in camera 1	[−1253.69, 138,779.72, 0]	[5645.42, 138,779.72, 0]
3	1-2588	Car1	[−371.55, 31,699.965, 0]	not appear in camera 2	[3834.615, 238,315.36, 0]
		Bus1	[−1789.39, 69,195.99, 0]	not appear in camera 2	[2416.77, 275,811.39, 0]
	1-2591	Car1	[−399.30, 30,724.42, 0]	not appear in camera 2	[3806.86, 237,339.82, 0]
		Bus1	[−1716.91, 67,609.70, 0]	not appear in camera 2	[2489.25, 274,225.10, 0]
	1-2593	Car1	[−556.38, 29,752.87, 0]	not appear in camera 2	[3649.787, 236,368.26, 0]
		Bus1	[−1738.99, 66,999.57, 0]	not appear in camera 2	[2467.17, 273,614.97, 0]

Table 7. *Cont.*

Example	Position		World Coordinate in Camera 1/mm	World Coordinate in Camera 2/mm	Unified World Coordinate/mm
3	2-3151	Car1	not appear in camera 1	[−3437.65, 91,918.145, 0]	[3461.46, 91,918.14, 0]
		Bus1	not appear in camera 1	[−4509.80, 173,877.80, 0]	[2389.30, 173,877.80, 0]
	2-3172	Car1	not appear in camera 1	[−3164.47, 77,067.36, 0]	[3734.63, 77,067.36, 0]
		Bus1	not appear in camera 1	[−4884.50, 168,037.29, 0]	[2014.60, 168,037.29, 0]
	2-3187	Car1	not appear in camera 1	[−3153.01, 67,825.67, 0]	[3746.10, 67,825.67, 0]
		Bus1	not appear in camera 1	[−4941.79, 163,175.51, 0]	[1957.31, 163,175.51, 0]

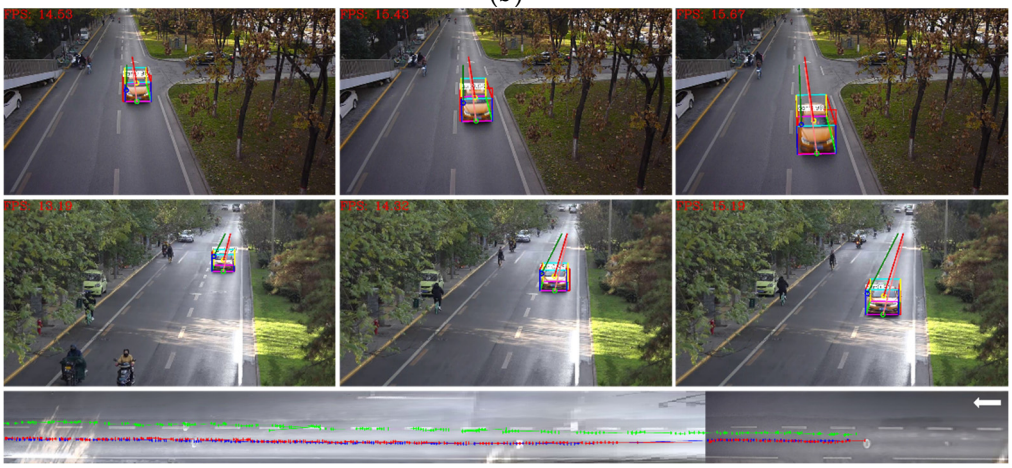
As shown in Figure 16, the proposed algorithm is compared with the 3D tracking methods based on feature point and 2D bounding box, which are represented by red, green, and orange respectively. It can be seen that the method based on feature point is greatly influenced by vehicle texture and surrounding environment, which cannot reflect true driving direction well, and may not be able to obtain continuous 3D trajectory under the condition of occlusion. The method based on 2D bounding box cannot accurately reflect the true driving position due to an unknown distance from bottom edge to the road plane. The proposed algorithm is superior to the existing methods because it can obtain accurate 3D vehicle bounding box, and is robust to vehicle occlusion and low visual angle of cameras. Comparison of the performance of several 3D tracking methods is summarized in Table 8.



(**a**)

Figure 16. *Cont.*

(**b**)



(**c**)

(**d**)

Figure 16. Comparison examples of different 3D trajectory extraction algorithms. (**a**) Comparison example 1; (**b**) Comparison example 2; (**c**) Comparison example 3; (**d**) Comparison example 4.

Table 8. Comparison of different 3D trajectory extraction algorithms.

Algorithm		Actual Driving Direction	Actual Driving Position	Continuous 3D Trajectory	Cross-Camera Scene	Panoramic Image
Single-Camera tracking methods	Gu et al. [11]	√	√	√	×	×
	Bullinger et al. [12]	√	√	√	×	×
	Cao et al. [13]	×	×	×	×	×
Cross-Camera tracking methods	Castaneda et al. [26]	×	×	×	√	×
	Peng et al. [28]	√	√	√	√	×
	Qian et al. [30]	×	×	×	√	×
	Ours	√	√	√	√	√

Since the proposed 3D vehicle detection algorithm is based on geometric constraints, the overall processing speed is fast. It can be seen from examples in Figure 15, the average processing speed of our algorithm on the GPU platform is 16 FPS with an average time of 600 ms, which can achieve real-time performance.

During the experiment, it can also be found that the accuracy of vehicle spatial distribution and 3D trajectory is related to the pan angle θ of the camera. Therefore, we count the accuracy under different camera pan angles, which is shown in Figure 17. When the pan angle is close to 0°, the information of the vehicle side surface is invisible, which leads to the decrease of 3D vehicle detection accuracy. In practical applications, the pan angle of the camera can be increased appropriately to retain most of the visual information of the vehicle.

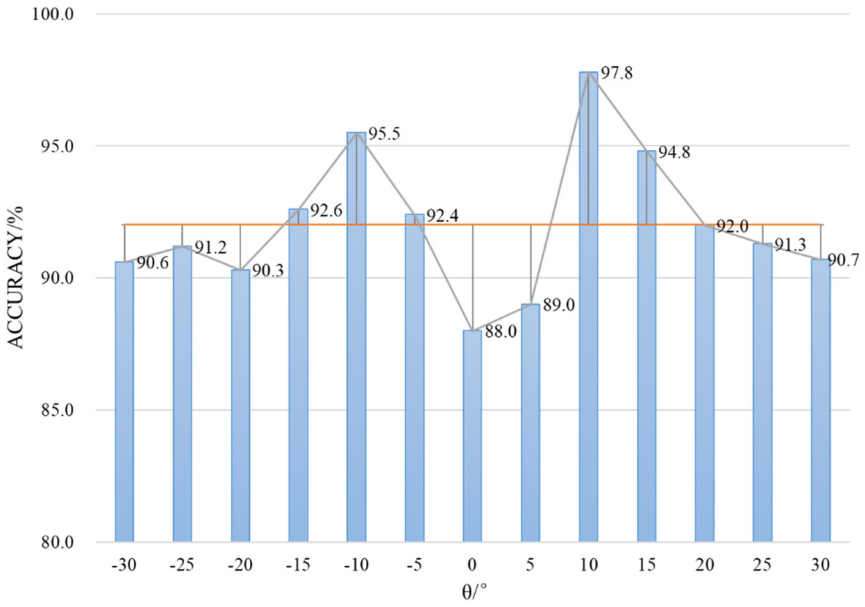


Figure 17. The accuracy of the proposed algorithm with different camera pan angles.

4. Conclusions

Through experimental verification, the proposed algorithm of vehicle spatial distribution and 3D trajectory extraction in cross-camera scenes in this paper has achieved good results in both BrnoCompSpeed dataset single-camera scenes and actual road cross-camera scenes. The main contributions of this paper are as follows: (1) A road space fusion algorithm in cross-camera scenes based on camera calibration is proposed to generate the panoramic image with physical information in road space, which can be used to convert multiple cross-camera perspectives into continuous 3D physical space. (2) A 3D vehicle detection algorithm based on geometric constraints is proposed to accurately obtain 3D vehicle projection centroids, which is used to describe vehicle spatial distribution in the panoramic image and to extract 3D trajectories. Compared with existing vehicle tracking methods, continuous 3D trajectories can be obtained in the

panoramic image with physical information by 3D projection centroids, which is helpful to applications in large scope road scenes.

However, 3D vehicle projection centroids obtained by the proposed algorithm in this paper is highly dependent on 2D vehicle detection results. When the vehicle is far from the camera, it is prone to be missed of detection and the accuracy will decrease when the camera pan angle is close to 0°. Moreover, the proposed algorithm cannot currently be adapted to various road situations and congested traffic. In future work, a more efficient method for road space fusion can be developed to generate the panoramic image and calculate vehicle spatial distribution more precisely and a more sophisticated vehicle detection network can be designed to fuse various types of geometric constraints to further improve the accuracy of 3D vehicle detection under different camera pan angles. In addition, only straight roads and simple traffic conditions are considered in this paper, which is necessary to be further extended to complex traffic scenes such as road-crossing (containing winding roads) and traffic congestion for more practical and advanced applications. Efforts are also needed to collect a large dataset of these complex traffic scenes for algorithm validation. This direction is a key and difficult point in the future work.

Author Contributions: Methodology, X.T.; resources, H.S.; validation, W.W.; writing–original draft, Y.Y.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62072053), the Fundamental Research Funds for the Central Universities (300102249317), Natural Science Foundation of Shaanxi Province (2019SF-258), National Key R&D Program of China (SQ2019YFB160023), and Key R&D project of Shaanxi Science and Technology Department (2018ZDXM-GY-047).

Acknowledgments: The authors would like to thank the researchers of Brno University of Technology for providing the public dataset BrnoCompSpeed which is used in our experiments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sivaraman, S.; Trivedi, M.M. Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1773–1795. [CrossRef]
2. Chen, Z.; Wu, C.; Huang, Z.; Lyu, N.; Hu, Z.; Zhong, M.; Cheng, Y.; Ran, B. Dangerous driving behavior detection using video-extracted vehicle trajectory histograms. *J. Intell. Transp. Syst.* **2017**, *21*, 409–421. [CrossRef]
3. Shokrolah Shirazi, M.; Morris, B.T. Vision-Based Turning Movement Monitoring: Count, Speed and Waiting Time Estimation. *IEEE Intell. Transp. Syst. Mag.* **2016**, *8*, 23–34. [CrossRef]
4. Ho, G.T.S.; Tsang, Y.P.; Wu, C.H.; Wong, W.H.; Choy, K.L. A Computer Vision-Based Roadside Occupation Surveillance System for Intelligent Transport in Smart Cities. *Sensors* **2019**, *19*, 1796. [CrossRef] [PubMed]
5. Dai, Z.; Song, H.; Wang, X.; Fang, Y.; Li, H. Video-Based Vehicle Counting Framework. *IEEE Access* **2019**, *7*, 64460–64470. [CrossRef]
6. Špaňhel, J.; Juránek, R.; Herout, A.; Novák, J.; Havránek, P. Analysis of Vehicle Trajectories for Determining Cross-Sectional Load Density Based on Computer Vision. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 1001–1006.
7. Hiribarren, G.; Herrera, J.C. Real time traffic states estimation on arterials based on trajectory data. *Transp. Res. Part B-Methodol.* **2014**, *69*, 19–30. [CrossRef]
8. Ma, Y.; Meng, H.; Chen, S.; Zhao, J.; Li, S.; Xiang, Q. Predicting Traffic Conflicts for Expressway Diverging Areas Using Vehicle Trajectory Data. *J. Transp. Eng.* **2020**, *146*, 1–10. [CrossRef]
9. Zhang, X.; Zhang, D.; Yang, X.; Hou, X. Traffic accident reconstruction based on occupant trajectories and trace identification. *ASME J. Risk Uncertain. Part B* **2019**, *5*, 20903–20914. [CrossRef]
10. Chen, X.; Kundu, K.; Zhang, Z.; Ma, H.; Fidler, S.; Urtasun, R. Monocular 3D Object Detection for Autonomous Driving. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2147–2156. [CrossRef]

11. Gu, I.Y.; Bolbat, M. Road traffic tracking and parameter estimation based on visual information analysis using self-calibrated camera views. In Proceedings of the 2013 Seventh International Conference on Distributed Smart Cameras (ICDSC), Palm Springs, CA, USA, 29 October–1 November 2013; pp. 1–6. [CrossRef]
12. Bullinger, S.; Bodensteiner, C.; Arens, M.; Stiefelhagen, R. Monocular 3D Vehicle Trajectory Reconstruction Using Terrain Shape Constraints. In Proceedings of the 2016 European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 1122–1128. [CrossRef]
13. Cao, M.; Zheng, L.; Jia, W.; Liu, X. Joint 3D Reconstruction and Object Tracking for Traffic Video Analysis Under IoV Environment. *IEEE Trans. Intell. Transp. Syst.* **2020**, 1–15. [CrossRef]
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the 2015 28th International Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 91–99. [CrossRef]
15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
16. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the 2016 European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37. [CrossRef]
17. Lin, T.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007. [CrossRef]
18. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. Available online: https://arxiv.org/pdf/1904.42607850v1.pdf (accessed on 16 April 2019).
19. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645. [CrossRef] [PubMed]
20. Sharma, S.; Jain, K. Image Stitching using AKAZE Features. *J. Indian Soc. Remote Sens.* **2020**, *48*, 1389–1401. [CrossRef]
21. Luo, X.; Li, Y.; Yan, J.; Guan, X. Image Stitching with Positional Relationship Constraints of Feature Points and Lines. *Pattern Recognit. Lett.* **2020**, *135*, 431–440. [CrossRef]
22. Lin, J.; Yang, C.K. Collaborative panoramic image generation from multiple mobile phones. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Hong Kong, China, 10–14 July 2017; pp. 339–344. [CrossRef]
23. Ma, J.; Zhang, J.; Sun, W. Research on Panoramic Image Mosaic Method Based on Camera Calibration. *J. Syst. Simul.* **2018**, *29*, 1112–1119. [CrossRef]
24. Hsu, C.; Chang, C.; Kang, L.K.; Fu, R.; Chen, D.; Weng, M. Fish-Eye Lenses-Based Camera Calibration and Panoramic Image Stitching. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Taichung, Taiwan, 19–21 May 2018; pp. 1–2. [CrossRef]
25. Hsu, C.Y.; Kang, L.W.; Liao, H.Y.M. Cross-camera vehicle tracking via affine invariant object matching for video forensics applications. In Proceedings of the 2013 IEEE International Conference on Multimedia and Expo (ICME), San Jose, CA, USA, 15–19 July 2013; pp. 1–6.
26. Castaneda, J.N.; Jelaca, V.; Frias, A.; Pizurica, A.; Philips, W.; Cabrera, R.R.; Tuytelaars, T. Non-Overlapping Multi-camera Detection and Tracking of Vehicles in Tunnel Surveillance. In Proceedings of the 2011 International Conference on Digital Image Computing: Techniques and Applications, Noosa, QLD, Australia, 6–8 December 2012; pp. 591–596. [CrossRef]
27. Straw, A.D.; Branson, K.; Neumann, T.R.; Dickinson, M.H. Multi-camera Realtime 3D Tracking of Multiple Flying Animals. *IEEE Trans. Smart Grid* **2010**, *6*, 1219–1226. [CrossRef]
28. Peng, J.; Shen, T.; Wang, Y.; Zhao, T.; Zhang, J.; Fu, X. Continuous Vehicle Detection and Tracking for Non-overlapping Multi-camera Surveillance System. In Proceedings of the International Conference on Internet Multimedia Computing and Service, Xi'an, China, 19–21 August 2016; pp. 122–125. [CrossRef]
29. Byeon, M.; Yun, S.; Ro, Y.; Jo, D.; Kim, K.; Choi, J.Y. Real-time scheme for 3-dimensional localizing and tracking of people in multiple camera settings. In Proceedings of the 2017 17th International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea, 18–21 October 2017; pp. 239–244. [CrossRef]

30. Qian, Y.; Yu, L.; Liu, W.; Hauptmann, A. ELECTRICITY: An Efficient Multi-camera Vehicle Tracking System for Intelligent City. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 2511–2519. [CrossRef]
31. Kanhere, N.K.; Birchfield, S.T. A Taxonomy and Analysis of Camera Calibration Methods for Traffic Monitoring Applications. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 441–452. [CrossRef]
32. Wang, W.; Zhang, C.; Tang, X.; Song, H.; Cui, H. Automatic Self-Calibration and Optimization Algorithm of Traffic Camera in Road Scene. *J. Comput.-Aided Des. Comput. Graph.* **2019**, *31*, 1955–1962. [CrossRef]
33. Wu, F.; Liang, H.; Song, H.; Jia, J.; Liu, L. Multi-Camera Traffic Scene Mosaic Based on Camera Calibration. *Comput. Syst. Appl.* **2020**, *29*, 176–183. [CrossRef]
34. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. Available online: https://arxiv.org/pdf/2004.10934.pdf (accessed on 23 April 2020).
35. Limits of Dimensions, Axle Load and Masses for Road Vehicles. Available online: http://www.miit.gov.476cn/n1146285/n1146352/n3054355/n3057585/n3057592/c5173956/part/5176262.pdf (accessed on 26 July 2016).
36. Sochor, J.; Juránek, R.; Špaňhel, J.; Maršík, L.; Široký, A.; Herout, A.; Zemčík, P. Comprehensive Data Set for Automatic Single Camera Visual Speed Measurement. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 1633–1643. [CrossRef]