

Article

Algorithm for Measuring the Outer Contour Dimension of Trucks Using UAV Binocular Stereo Vision

Shiwu Li, Lihong Han, Ping Dong * and Wencai Sun *

School of Transportation, Jilin University, 5988 Renmin Street, Changchun 130022, China

* Correspondence: dongping@suibe.edu.cn (P.D.); sunwencai@jlu.edu.cn (W.S.); Tel.: +86-180-4994-7392 (P.D.); +86-135-0080-6339 (W.S.)

Abstract: Promoting the management of the over-limit of freight transport vehicles plays an important role in the sustainable development of the highway industry. Vehicle outer contour dimension measurement is a key element in highway over-limit detection. The current detection approaches and research methods, however, are insufficient for high-precision flow detection. Therefore, this study proposes an algorithm for measuring the dimensions of a truck's outer contours, using unmanned aerial vehicle (UAV) binocular stereo vision. First, this study leverages a binocular camera mounted on a UAV to reconstruct the 3D point clouds of the truck. Second, the point cloud data are clustered using an FoF (Friends-of-Friends algorithm); this recognizes the cluster of truck points according to the truck's characteristics. Finally, the principal component analysis and the Gaussian kernel density estimation are used to generate the outer contour dimensions of the trucks. Twenty model vehicles are selected as test objects to verify the reliability of the algorithm. The average error of the algorithm is represented by calculating the average value of the difference between the real size and the predicted size of the three dimensions. The experimental results demonstrate that the average error of this measurement approach is less than 2.5%, and the method is both stable and robust. This approach aligns with national regulations for over-limit detection.



Citation: Li, S.; Han, L.; Dong, P.; Sun, W. Algorithm for Measuring the Outer Contour Dimension of Trucks Using UAV Binocular Stereo Vision. *Sustainability* **2022**, *14*, 14978. <https://doi.org/10.3390/su142214978>

Academic Editor: Armando Carteni

Received: 26 October 2022

Accepted: 11 November 2022

Published: 12 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: UAV; over-limit detection; vehicle outer contour dimension; principal component analysis; kernel density estimation

1. Introduction

When a truck is over-limit, it means that its length, width, height, and cargo mass exceed specified limitations. Over-limit transportation endangers traffic safety, damages roads, pollutes the environment, disrupts the normal economic order of the market, and hinders sustainable development in the highway industry. A vehicle's three-dimensional (3D) information recognition system is used to measure the dimensions of the outer contours of the vehicle and is an important part of its intelligent transportation system (ITS). The results generated by such a recognition system can be applied to detect over-limits on highways [1], recognize vehicle types [2], and support automatic driving [3]. This has important practical significance for road traffic safety, road protection, and road transport market order.

The mainstream vehicle 3D information recognition methods currently rely on LiDAR sensors to obtain accurate 3D information. LiDAR has significant advantages in accuracy and efficiency but is expensive, has a short life, and has limited perceptive abilities. As such, the vision sensor has attracted significant attention in academia and industry, due to its low cost and high resolution. There are two main types of vision-based vehicle 3D information recognition methods based on the number of vision sensors: monocular vision and binocular vision. First, for monocular traffic scenes, three main methods are used to identify a vehicle's 3D information: vehicle-model-based methods, deep-learning-based methods, and self-calibration-based methods. When considering the methods based

on vehicle models [4], the model-based method has too many computational variables. The deep-learning-based method [5–7] requires a large volume of data and requires the estimation of a priori information. The self-calibration method [8] highly depends on self-calibration results and needs to be calibrated for roadside monitoring in traffic scenes.

Second, for binocular traffic scenes, two main methods are used to identify a vehicle's 3D information: 3D object detection and 3D reconstruction. The 3D object detection method is mainly used in the field of autonomous driving [9–13] and is used to perceive traffic scenes. In addition, 3D vehicle information is generated in the form of enclosing frames, and the accuracy requirements for vehicle outer contour size are low. The 3D reconstruction method is based on the 3D coordinates of all or part of the point cloud on the surface of the object, to reconstruct all or part of the surface of the target object. This approach can be used to assist robots or UAVs in completing the measurement task of the target and can meet accuracy requirements better than 3D object detection. Each point in the point cloud data obtained through 3D reconstruction corresponds to a measurement point, which is the most realistic record of the object's geometric properties. In this study, a 3D reconstruction method based on binocular stereo vision is selected to obtain the real outer contour dimensions of the truck in a highway over-limit detection scenario. This involves obtaining the real point cloud data of the truck and calculating its outer contour dimensions using the outer-contour dimension-solving algorithm.

Most existing over-limit detection equipment is fixed. This kind of detection station is expensive, requiring large investments in labor and material resources, and provides little coverage. In contrast, a UAV can provide a mobile detection solution. It has the advantages of small size, as well as high efficiency and flexibility, and has a low level of environmental dependence with wide coverage. As such, it has been applied in the different aspects of the transportation field, such as regular data collection for intelligent transportation systems [14], the delivery of goods [15,16], vehicle tracking [17–21], and traffic monitoring [22,23].

In this study, to enable the mobile detection of highway over-limits, a UAV-based binocular stereo vision algorithm is proposed to measure the dimensions of a truck's outer contours. The larger goal is to provide algorithmic support for future UAV portable over-limit detection equipment. Figure 1 shows the highway over-limit movement detection scene based on the UAV.



Figure 1. Highway over-limit movement detection scene based on UAV.

The main contributions of this study are as follows:

- (1) A method is proposed to solve the ground plane equation by iteratively correcting the ground plane normal vector using the least square method. This method is verified to be robust in the over-limit detection scenario;
- (2) Considering the efficiency of applying UAVs for over-limit detection, this study proposes a point cloud segmentation algorithm based on FOF clustering. This approach is computationally efficient, and the number of clusters does not need to be artificially set;

- (3) To address the characteristics of the large length–width ratios and symmetries associated with truck bodies, this study proposes a method for calculating length and width using the principal component analysis and the Gaussian kernel density estimation.

The rest of this paper is organized as follows: Section 2 describes the process of obtaining the point cloud data of vehicles using ZED2. Section 3 details the algorithm for measuring the outer contour dimensions of trucks based on their 3D point cloud information, which mainly includes two parts: the extraction of the target vehicle's point clouds and the prediction of the outer contour dimension. Section 4 validates the proposed experimental algorithm and estimates the error of the algorithm. Finally, Section 5 presents the concluding remarks and possible future work. Figure 2 shows the overall workflow of this algorithm.

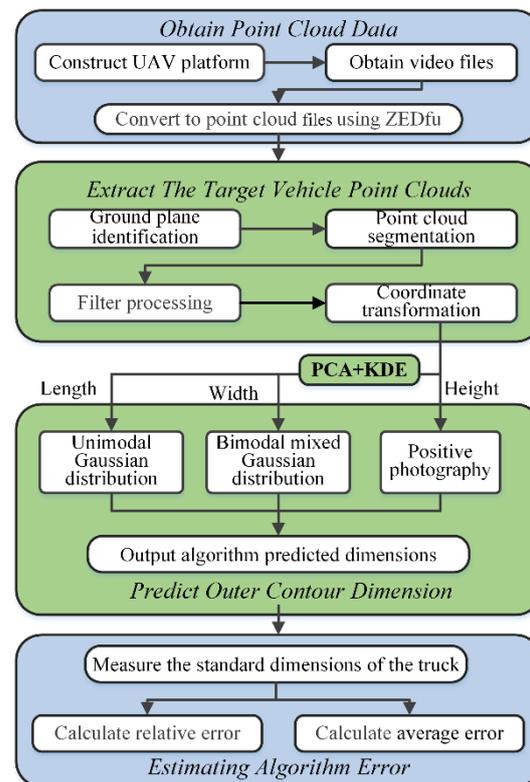


Figure 2. The overall workflow of the algorithm for measuring the outer contour dimension.

2. Vehicle 3D Point Cloud Acquisition

This study focuses on the methods for obtaining the true outer contour dimensions based on a reconstructed vehicle point cloud. It does not specifically study the 3D reconstruction of the vehicle itself. As such, the ZED2 stereo camera developed by Stereolab Labs was selected for the study. The SDK-ZEDfu provided by the ZED development team enables the 3D reconstruction of objects. Table 1 lists the most important features of the ZED2 camera. Figure 3 shows a schematic of the coordinate system using the ZED2 stereo camera.

Table 1. Technical specifications for the ZED2.

Parameters	Values
Dimensions	175 × 30 × 33 mm
Weight	166 g
Field of View	110° (H) × 70° (V) × 120° (D)
Depth Range	0.2–20 m
Output Resolution (side by side)	HD720: 1280 × 720 (60/30/15 FPS)
Operating Temperature	−10 °C to +45 °C

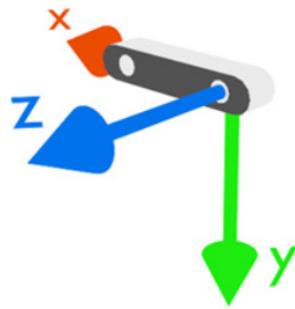


Figure 3. Coordinate system under ZED2 stereo camera.

3. Methods

3.1. Truck Point Cloud Segmentation

3.1.1. Ground Plane Identification

A classical algorithm for plane extraction is the random sampling consistency (RANSAC) algorithm. It estimates the plane parameters by randomly selecting three points and then calculates the rate of the inliers (i.e., points on the plane). After a specific number of iterations, the plane with the maximum rate of inliers is extracted [24]. However, the RANSAC algorithm is random when repeatedly detecting planes, it is not robust to outliers, and its accuracy heavily depends on the number of iterations. Therefore, this study proposed a method for solving the plane equations by iteratively correcting the ground plane normal vectors using the least square method based on the Bayesian principles. This method is robust to outliers and requires only a small number of iterations to achieve high accuracy.

The overall structure of the method is outlined in the pseudocode of Algorithm 1.

Algorithm 1 Ground plane identification

Input: $P = \{(x_j, y_j, z_j), j = 1, \dots, N_p\}$; threshold ξ

- (1) Initialization: $\mathbf{n}^{(0)} = (0, 1, 0)^T$
- (2) **For** each $\mathbf{n}^{(i)}$ **do**
- (3) $i = 0$
- (4) $W^{(i)} = P \bullet \mathbf{n}^{(i)}$
- (5) $\mu(w) \leftarrow$ Fit the probability density function corresponding to $W^{(i)}$ by nonparametric KDE
- (6) $w_y^{(i)} = \underset{w}{\operatorname{argmax}} \mu(w)$
- (7) $P_m^{(i)} \leftarrow$ Select the points within $w_y^{(i)} \pm \xi$
- (8) $\tau^{(i)} = n_g^{(i)} / N_p$
- (9) $\vec{n}^{(i+1)} \leftarrow$ Fitting plane by Equation (7) within $P_m^{(i)}$ and $d^{(i+1)} = -h_y^{(i)}$
- (10) **If** $\tau^{(i+1)} \geq \tau^{(i)}$
- (11) $i = i + 1$;
- (12) **Else**
- (13) $\tau^* = \tau^{(i+1)}$;
- (14) **End If**
- (15) **End For**

Output: $a^*x + b^*y + c^*z + d^* = 0$

The general form of the plane equation is expressed as $ax + by + cz + d = 0 (b \neq 0)$. The normal vector of the plane is $(a, b, c)^T$.

According to the point cloud data acquisition method and the binocular camera coordinate system, the positive direction of the y -axis in the coordinate system was selected as the initial normal vector direction $\mathbf{n}^{(0)}$. The point clouds obtained through 3D reconstruction are denoted as $P = \{(x_j, y_j, z_j), j = 1, \dots, N_p\}$. We then project P to $\mathbf{n}^{(0)}$. The resulting one-dimensional point set is denoted as $W^{(0)}$; the probability distribution corresponding to $W^{(0)}$ is fitted using the nonparametric kernel density estimation (KDE);

and the probability density function is recorded as $\mu(w)$. The points corresponding to the maximum probability density are found to be $w_y^{(0)}, w_y^{(0)} = \operatorname{argmax}_w \mu(w)$. The next steps are to set a neighborhood threshold ξ ; identify the original three-dimensional point clouds $P_m^{(0)} = \{(x_i^{(0)}, y_i^{(0)}, z_i^{(0)}), i = 1, \dots, n_g^{(0)}\}$ corresponding to the projected points in the range of $w_y^{(0)} \pm \xi$; and calculate the interior point rate. The interior point rate τ refers to the ratio of the number of point clouds in the plane to the total number of point clouds. The plane equation is fit within $P_m^{(0)}$ using the least square method.

$$(a^{(1)}, b^{(1)}, c^{(1)})^T = \operatorname{argmin}_{a,b,c} \|ax + by + cz - d\|_2^2 \quad (1)$$

To ensure the identifiability of the model, a, b, c, d in Equation (1) need to satisfy the following two equations:

$$(a^{(1)})^2 + (b^{(1)})^2 + (c^{(1)})^2 = 1 \quad (2)$$

$$d^{(1)} = -w_y^{(0)} \quad (3)$$

The normal vector of the fitted plane is obtained using the following expression: $\mathbf{n}^{(1)} = (a^{(1)}, b^{(1)}, c^{(1)})^T$.

The next step is to project the point cloud P in the direction of the new normal vector and repeat the previous steps. If the newly obtained interior point rate τ increases, then the normal vector is updated, and the iterations continue, stopping the process in the opposite direction. The final output is the optimal plane normal vector \mathbf{n}^* . The obtained ground plane equation is expressed as $a^*x + b^*y + c^*z + d^* = 0$.

To prevent the ground point clouds from interfering with the detection of the target in the subsequent clustering process, the detected ground plane is removed. Based on the clustering nature of the point clouds and the plane determination theorem, the ground thickness (defined as the size of the ground area where the truck is located after the 3D reconstruction of the point cloud in the direction of the normal vector of the ground plane) d_h is 3–4 times the resolution of the point clouds. In this study, the optimal ground thickness d_h^* is determined using the mean square error (MSE) function. The calculation equation is as follows:

$$d_h^* = \operatorname{argmin}_{d_h} \operatorname{MSE} = \operatorname{argmin}_{d_h} \left\| [l(d_h), w(d_h), h(d_h)]^T - [l_0, w_0, h_0]^T \right\|_2^2 \quad (4)$$

where $l(d_h), w(d_h), h(d_h)$ represents the truck's modeled length, width, and height, respectively; l_0, w_0, h_0 are the truck's real measured length, width, and height.

3.1.2. Target Vehicle Point Cloud Segmentation Based on FoF Clustering

The point clouds of the same object are consistently close to each other, and there is a distance between the point clouds of different objects. Due to this, the point cloud clustering algorithm can be used for segmentation according to the Euclidean distance of spatial points. In the clustering process, many neighboring points need to be searched for the data points according to the distance. The number of reconstructed point clouds reaches millions, making it inefficient to use the traversal methods such as K-means and DBSCAN. Therefore, this study adopts the FoF clustering method for point cloud segmentation; this method has a fast computing speed and low memory consumption. FoF clustering is mainly used for astronomy applications, where data are generally measured in terabytes or petabytes. Therefore, using this algorithm to process millions of data points improves the efficiency of the algorithm.

In cosmology, the FoF algorithm is used to identify the objects of interest and the quantitative structure in particle systems. It is a simple clustering algorithm that checks the distance between particles [25]. Figure 4 shows the principle of the FoF clustering

algorithm. If the distance between two particles is less than the threshold ε , the FoF defines them as friends. For example, in Figure 4, A and B are friends, and B and C are friends, but A and C are not friends. Two particles are friends of friends if they can be reached by traversing the transfer closure created by the friend relationship. If a particle has no friends, it is classified as being noise. For example, in Figure 4, A and C are friends of friends through B , and D is a noise point. To compute the clusters, the algorithm computes the passing closure for each unvisited particle's friend relationship. All the particles in the closure are marked as visited and are linked as a single cluster.

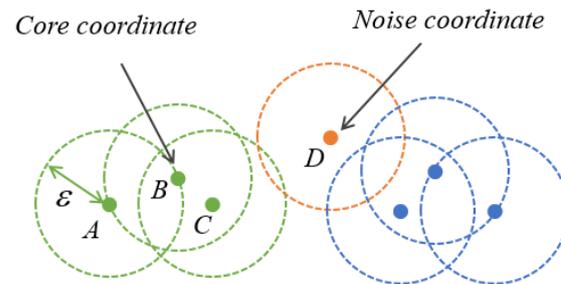


Figure 4. Schematic diagram of FoF clustering algorithm.

In this study, we first build an index based on kd trees to facilitate a fast search. A kd tree is a tree-like data structure that stores instance points in the k -dimensional space for fast retrieval [26]. Constructing a kd tree involves continuously slicing the k -dimensional space with a hyperplane perpendicular to the coordinate axes until there are no instances in two subregions. The resulting kd tree is then generated as output.

FoF clustering is then performed on the samples. The overall structure of the FoF clustering method is outlined in the pseudocode of Algorithm 2. In lines 1–7, the algorithm first finds all the core objects according to the given neighborhood parameter, and then in lines 10–24, with any core object as the starting point, finds the samples whose density can reach to generate the clusters until all the core objects have been visited. This involves first denoting the point cloud set P' as the sample set $S = \{x_1, x_2, \dots, x_{N_p}\}$. The neighborhood parameter ε for FoF clustering is determined based on point cloud resolution and clumping. For sample x_i , ε is set as the distance threshold to find friends. All samples are recorded except x_i in the sample closure as the neighborhood $N_\varepsilon(x_i)$ of sample x_i . Finally, all the point cloud clusters are output.

In terms of the parameter settings, the FoF clustering algorithm has only one threshold parameter. Tuning the FoF clustering parameters is a complex process, and finding optimal parameters and performing sensitivity analysis operations for FoF clustering is beyond the scope of this study. Instead, engineering heuristics are used to estimate these parameters (see Section 4 for the details of threshold determination). While the resulting parameters could be further optimized, they are sufficient for the key task of extracting the target vehicle point clouds. Compared with most clustering algorithms, the FoF clustering algorithm does not require the number of clusters in advance. Further, it removes some of the noise points and has a flexible cluster shape to meet the requirements of vehicle point cloud segmentation.

After clustering, the point cloud cluster needs to be filtered to extract the point clouds of the truck. In other words, after the point clouds of the truck are obtained, the outer contour dimensions can be calculated to determine whether the limit is exceeded. Two filtering methods were used in this study: cluster-like point cloud number filtering and circumscribed cuboid size filtering. Cluster-like point cloud number filtering determines the number of point clouds in each cluster-like after clustering, sets the minimum and maximum number of point clouds for a target object, and filters those clusters that are not within this range. Circumscribed cuboid size filtering is used to determine the size of each

minimum circumscribed cuboid after clustering and further filter the nontarget objects based on size.

Algorithm 2 FoF clustering

Input: Sample set $S = \{x_1, x_2, \dots, x_{N'_p}\}$; Neighborhood parameter ε

- (1) Initialize the core object collection: $\Omega = \emptyset$
- (2) **For** $i = 1, 2, \dots, N'_p$ **do**
- (3) Determine the neighborhood $N_\varepsilon(x_i)$ of the sample x_i ;
- (4) **If** $|N_\varepsilon(x_i)| \geq 1$; **then**
- (5) Add sample x_i to the core object collection: $\Omega = \Omega \cup \{x_i\}$
- (6) **End If**
- (7) **End For**
- (8) Initialize the number of clusters: $\alpha = 0$
- (9) Initialize the unvisited collection: $\Gamma = S$
- (10) **While** $\Omega \neq \emptyset$ **do**
- (11) Log the sample collection that is not currently visited: $\Gamma_{old} = \Gamma$;
- (12) Randomly select a core object $o \in \Omega$, initialize the queue $G = \langle o \rangle$;
- (13) $\Gamma = \Gamma \setminus \{o\}$;
- (14) **While** $G \neq \emptyset$ **do**
- (15) Take out the first sample g in queue G ;
- (16) **If** $|N_\varepsilon(g)| \geq 1$ **then**
- (17) $\Delta = N_\varepsilon(g) \cap \Gamma$;
- (18) Add the samples in Δ to queue G ;
- (19) $\Gamma = \Gamma \setminus \Delta$;
- (20) **End If**
- (21) **End While**
- (22) $\alpha = \alpha + 1$, $C_\alpha = \Gamma_{old} \setminus \Gamma$;
- (23) $\Omega = \Omega \setminus C_\alpha$
- (24) **End While**

Output: $C = \{C_1, C_2, \dots, C_\alpha\}$

3.2. Measurement of the Outer Contour Dimension of the Truck

3.2.1. Coordinate Transformation of Target Vehicle Point Clouds Based on the Ground Plane Equation

In the 3D point cloud data reconstructed by using the ZED2 binocular camera, the ground plane is not horizontal in the coordinate system. This affects the subsequent point cloud processing, making it necessary to calibrate. According to the identified point cloud data of the truck and the parameters of the ground plane equation, the corrected point clouds of the truck are obtained using coordinate transformation. This involves defining the point cloud coordinates of the truck in the original coordinate system as $p_{tr} = [x_{tr}, y_{tr}, z_{tr}]^T$. The corrected point cloud coordinates are defined as $p'_{tr} = [x'_{tr}, y'_{tr}, z'_{tr}]^T$. A set of the unit orthonormal bases of the original coordinate system, defined as coordinate system 1, is denoted as $[e_1, e_2, e_3]$. A set of the orthonormal bases of the corrected coordinate system (defined as coordinate system 2) is denoted as $[e'_1, e'_2, e'_3]$. According to the Euclidean transformation, this yields the following conversion relationship:

$$\begin{bmatrix} x'_{tr} \\ y'_{tr} \\ z'_{tr} \end{bmatrix} = \begin{bmatrix} e_1^T e'_1 & e_1^T e'_2 & e_1^T e'_3 \\ e_2^T e'_1 & e_2^T e'_2 & e_2^T e'_3 \\ e_3^T e'_1 & e_3^T e'_2 & e_3^T e'_3 \end{bmatrix} \begin{bmatrix} x_{tr} \\ y_{tr} \\ z_{tr} \end{bmatrix} = R_{12} \begin{bmatrix} x_{tr} \\ y_{tr} \\ z_{tr} \end{bmatrix} \quad (5)$$

$$R_{12} [x_{tr}, y_{tr}, z_{tr}]^T + t_{12} = [x'_{tr}, y'_{tr}, z'_{tr}]^T \quad (6)$$

where R_{12} refers to the rotation matrix that transforms the vector of coordinate system 1 into coordinate system 2, and t_{12} refers to the translation vector that transforms the vector of coordinate system 1 to coordinate system 2.

3.2.2. Vehicle Length and Width Solution Based on Principal Component Analysis and KDE

The solution to the height of the truck is relatively simple, compared with the length and width, and only needs to project the corrected vehicle point cloud data to the height direction. The distance from the highest point $[x_h, y_h, z_h]^T$ to the ground plane is the height of the truck. The equation to calculate this height is as follows:

$$h(d_h) = \frac{|a^*x_h + b^*y_h + c^*z_h + d^*|}{\sqrt{(a^*)^2 + (b^*)^2 + (c^*)^2}} \quad (7)$$

For the solution of vehicle length and width, the side of the truck body is perpendicular to the ground. This orientation provides a large amount of reliable length and width information. Therefore, based on the idea of three-dimensional to two-dimensional transformation, the truck point cloud is projected in the direction of the ground plane to obtain a two-dimensional point cloud. The length-to-width ratio of the truck is large; as such, the principal component analysis (PCA) can be used to determine the direction of the length and width of the truck.

PCA is a common data analysis method and is often used to reduce the dimensionality of high-dimensional data. The goal of PCA is to find the best linear projection to transform high-dimensional data into a low-dimensional subspace, by maximizing the variance of each projection dimension. This approach can be used to extract the main feature components of the data [27]. In this study, singular value decomposition (SVD) is used to perform PCA. The solved left singular eigenvector u_1 is the direction of the vehicle's length, and the left singular eigenvector u_2 is the direction of the vehicle's width. After determining the direction of the truck's length and width, a symmetry method is applied to measure the dimensions of the vehicle's outer contours, considering the body features of the truck.

The KDE method, developed by Rosenblatt [28] and Parzen [29], is applied in the symmetric solution. This is a nonparametric estimation method that does not use a priori knowledge for the data distribution, does not attach any assumptions to the data distribution, and enables the study of data distribution characteristics from the data sample itself. The Kernel density estimation theory [30] proposes that the accuracy of kernel density estimation depends on the bandwidth and the kernel function. When the bandwidth h is constant, the different kernel functions have little effect on the accuracy of kernel estimation. This makes the choice of bandwidth particularly important compared with the kernel function. If h is too large, it leads to a too-smooth kernel estimation and a large estimation bias. In contrast, if h is too small, it leads to large volatility in kernel estimation and an underbalanced state.

In this study, the two-dimensional point clouds are first projected onto the direction of length u_1 and the direction of width u_2 . Plotting the frequency histograms of the point cloud projection distribution in the length and width directions reveals that the point cloud projection in the truck width direction conforms to a bimodal mixed distribution. In contrast, the length direction conforms to a single-peaked distribution at the end of the truck (Figure 5). In this study, a smooth and continuous Gaussian kernel function is selected to estimate the kernel density of the point cloud distribution, and the cross-validation method is used to select the bandwidth. A positive feature of the cross-validation method is that the selected bandwidth automatically adapts to the smoothness of the kernel function [31]. The optimal objective function for bandwidth selection is formulated as follows:

$$CV(h) = \frac{1}{n^2h} \sum_i \sum_j K \star K\left(\frac{X_j - X_i}{h}\right) - \frac{2}{n(n-1)} \sum_i \sum_{j \neq i} K_h(X_i - X_j) \quad (8)$$

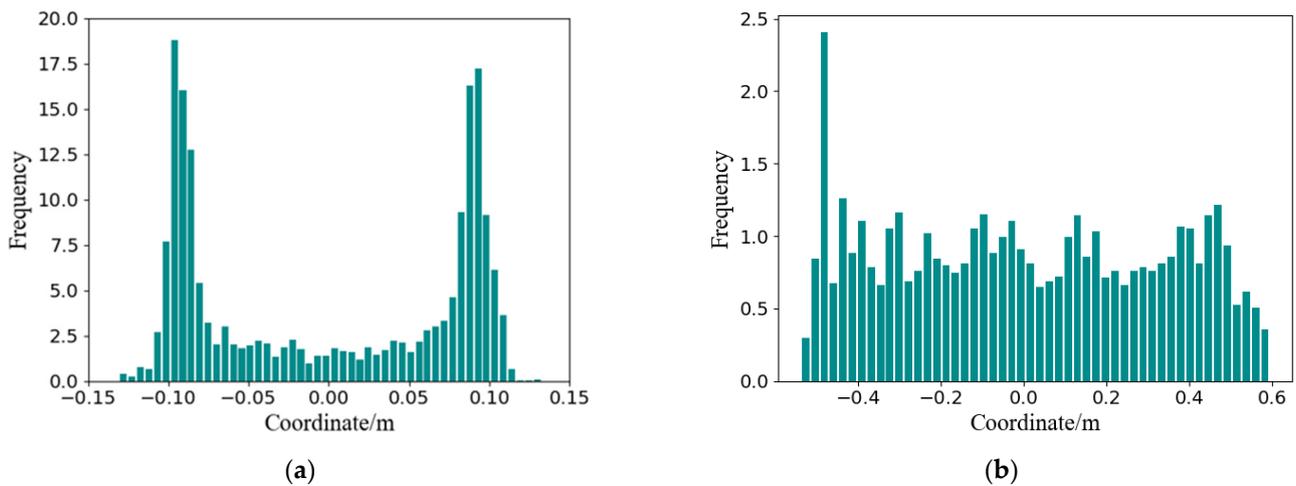


Figure 5. Frequency distribution histogram of point cloud projection: (a) width direction; (b) length direction.

This section does not specifically perform a posteriori tests on the kernel density estimation results [32]; instead, we directly verify the truck's final outer contour dimension measurements, indirectly verifying the validity of the method used to generate the vehicle's 3D measurements.

To better illustrate how to solve for length and width using kernel density estimation, we plot the Gaussian kernel density estimation of point cloud projections for a dataset based on the bandwidth selected using cross-validation, as shown in Figure 6. In the width direction, the points corresponding to the peaks v^* are solved based on the bimodal mixed Gaussian distribution. The distance between the two peaks corresponding to points v_1^* and v_2^* is the vehicle width. The solution equation is as follows:

$$v^* = \underset{v}{\operatorname{argmax}} f(v) \quad (9)$$

where $f(v)$ is the probability density function of the single-peaked region.

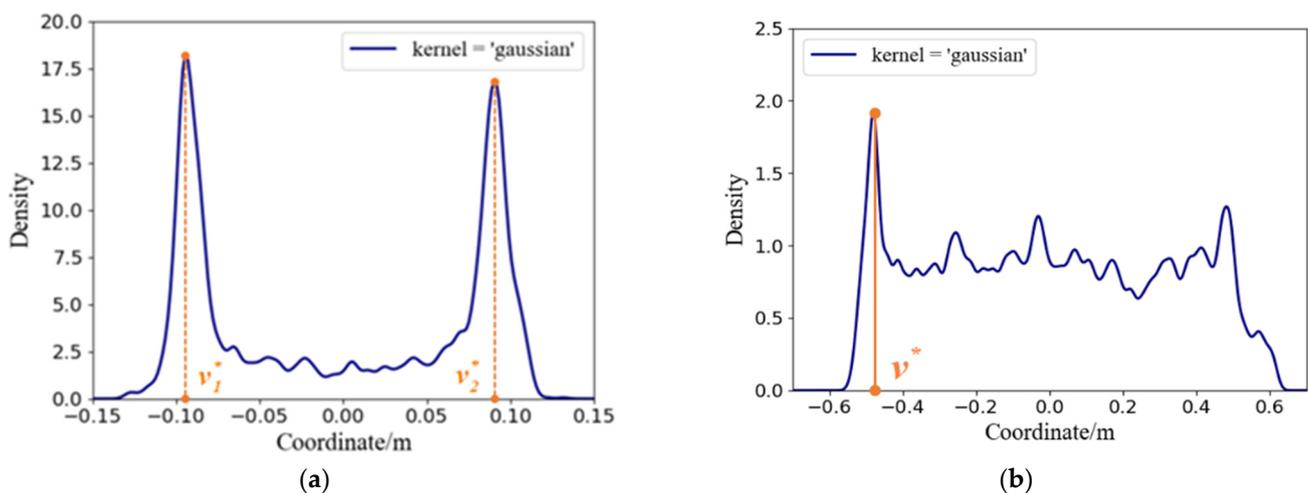


Figure 6. Gaussian kernel density estimation of point cloud projection: (a) width direction; (b) length direction.

The direction of the length of the truck is not symmetrical. As such, only the rear part of the truck conforms to the Gaussian distribution. Therefore, the rear part is solved for v^* based on the unimodal Gaussian distribution. The length of the truck is the distance from v^* to the front of the truck.

4. Experiments and Results

4.1. Experiments' Preparation

Construction of the UAV Platform and Experimental Implementation Scheme

A customizable industrial-grade quadcopter UAV was used to carry the ZED2 binocular camera. The binocular camera was connected to the UAV using a digitally encoded gimbal. The ZED2 was connected to an NVIDIA AGX Xavier single-board computer that provided the embedded system for processing the raw data acquired by the ZED2. A wireless gateway device was used to establish a two-way communication link between the client (ground station computer) and the server (AGX). Figure 7 shows the completed UAV platform. The client used NoMachine software to remotely control the server to acquire and process the data.



Figure 7. UAV platform.

In this study, we verified the proposed algorithm to measure the outer contour dimensions using model vehicles. The model vehicles included a training model vehicle and a test model vehicle. The training model vehicle was used to determine the parameters, and the test model vehicle was used to estimate the algorithm error. The data acquisition scheme is as follows: First, the Mission Planner was used to plan the path and flight speed for the UAV; the planned path circled the truck model. Figure 8 shows the experimental implementation scenario. The SDK-ZED Explorer provided by the ZED development team was used to obtain a video file containing the 360° view of the truck. ZEDfu was then used to convert the video files to point cloud files for output.



Figure 8. Experimental implementation scenario.

4.2. Algorithm Implementation

4.2.1. Determination of Optimal Ground Thickness Parameters in the Ground Plane Identification

During the experiment, the algorithm parameters were set according to the characteristics of the environment and the density characteristics of the reconstructed point clouds. When identifying the ground plane, the MSE function was used to determine the optimal

ground thickness; the ground thickness was 3–4 times the resolution of the point clouds. Therefore, we set eleven d_h values ranging from 0.06 to 0.08 m in steps of 0.002 m; those eleven values were used to predict the outer contour dimensions for each of the six training model vehicles. The MSE was calculated for each, and the average was calculated for the six model vehicles. Figure 9 shows the average MSE of the six model vehicles; the value corresponding to the minimum average MSE was selected as the best value for the ground thickness. This optimal value was 7 cm.

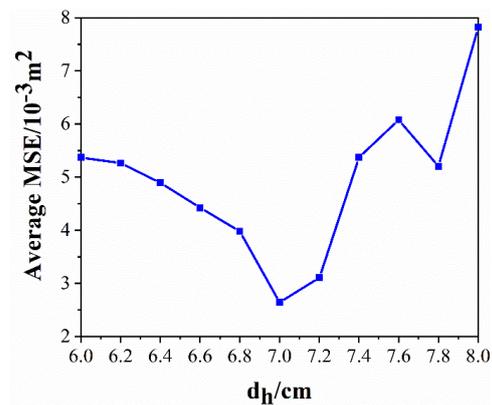


Figure 9. Variation curve of average MSE with d_h .

4.2.2. Determination of Threshold Parameters in FOF Clustering

The distance threshold parameter ε in FoF clustering has no effect on the accuracy of the vehicle's outer contour dimension measurements but has a significant effect on the time required for the UAV to complete the measurement. Given this, this study set ten values in steps of 0.01 m between 0.03 and 0.12 m to investigate the effect of different distance thresholds on measurement time. Figure 10a shows the variation in the measurement time with the distance threshold for the six training model vehicles. Figure 10b shows the average measurement time with the variation in the distance threshold. Figure 10a indicates that the number of reconstructed point clouds differed between the vehicles, and the measurement time also differed. Figure 10b shows that the average measurement time gradually decreased when the distance threshold ε was in the range of [0.03, 0.05] m and gradually increased when ε was in the range of [0.05, 0.12] m. Therefore, the average measurement time was at the minimum value when ε was 0.05 m. Therefore, the optimal distance threshold ε^* was set as 0.05 m in this study.

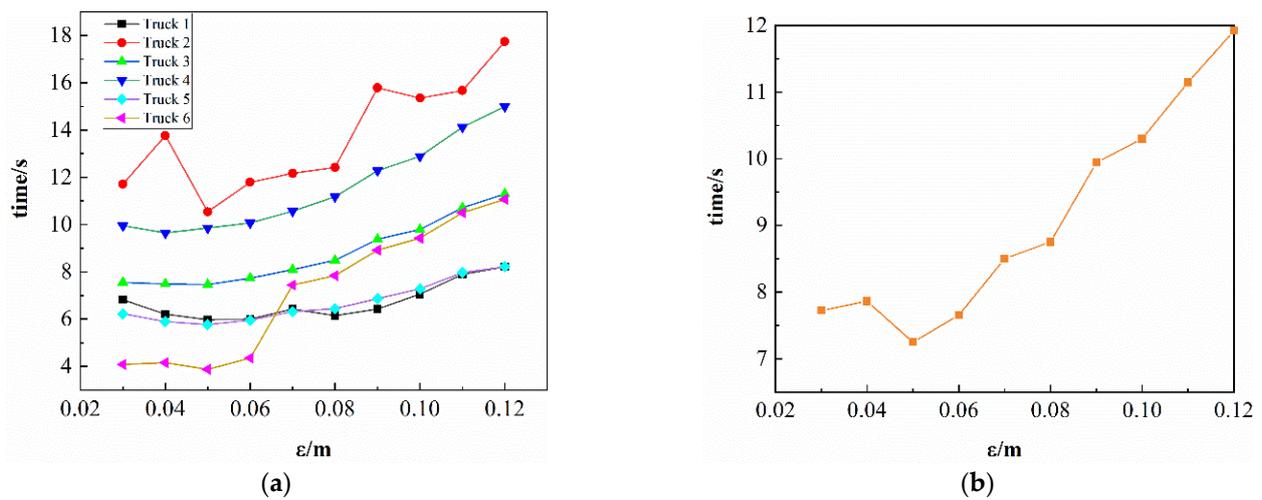


Figure 10. (a) Variation in measurement time with the distance threshold for each vehicle; (b) variation in average measurement time with the distance threshold.

4.2.3. Error Estimation of the Outer Contour Dimension Measurement Algorithm

Two evaluation metrics were used to evaluate the accuracy of the algorithm used to measure the vehicle's outer contours: the relative error and the average error. The relative error represents the prediction accuracy of the model for one-dimensional size. The average error represents the overall prediction accuracy of the model for three-dimensional size.

Twenty model vehicles were selected as the test objects to verify the reliability of the algorithm. Figure 11 shows the RGB image of the vehicle model and the point cloud image after 3D reconstruction. The test subjects are labeled as trucks 1–20. A plumb was used to mark the longest, tallest, and widest position of the truck model; the laser rangefinder was then used to measure the actual size of the truck as the standard size. Finally, the measurement algorithm proposed in this study was used to predict the outer contour dimensions of the twenty truck models. Table 2 shows the standard sizes, predicted sizes, and relative and average errors of the twenty model vehicles.

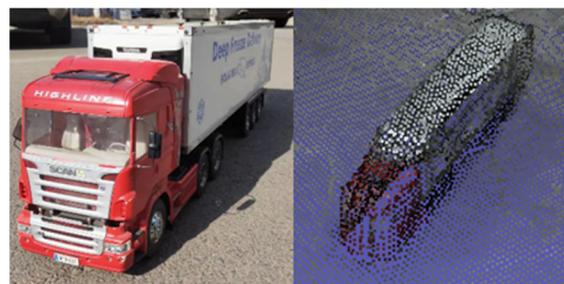


Figure 11. Truck model and reconstructed point cloud.

Table 2. Experimental results.

Subject	Standard Size			Predicted Size			Relative Error			Average Error	
	Length/m	Width/m	Height/m	Length/m	Width/m	Height/m	Length Error	Width Error	Height Error		
1	1.157	0.180	0.307	1.184	0.184	0.31	2.28%	2.17%	0.97%	1.81%	
2	1.151	0.179	0.316	1.184	0.184	0.32	2.80%	2.72%	1.27%	2.26%	
3	1.084	0.183	0.308	1.106	0.184	0.315	1.99%	0.54%	2.22%	1.58%	
4	1.071	0.177	0.324	1.106	0.184	0.325	3.18%	3.80%	0.32%	2.43%	
5	1.071	0.177	0.307	1.102	0.184	0.315	2.81%	3.70%	2.56%	3.02%	
6	1.065	0.182	0.323	1.102	0.184	0.325	3.32%	1.08%	0.64%	1.68%	
7	2.184	0.384	0.375	2.161	0.372	0.377	1.05%	3.13%	0.53%	1.57%	
8	2.184	0.384	0.385	2.112	0.372	0.394	3.30%	3.13%	2.34%	2.92%	
9	2.308	0.324	0.335	2.242	0.334	0.341	2.86%	3.09%	1.79%	2.58%	
10	2.106	0.384	0.38	2.025	0.371	0.375	3.85%	3.39%	1.32%	2.85%	
11	2.106	0.384	0.37	2.056	0.38	0.373	2.37%	1.04%	0.81%	1.41%	
12	2.102	0.384	0.335	2.098	0.374	0.331	0.19%	2.60%	1.19%	1.33%	
13	2.102	0.384	0.355	2.061	0.379	0.359	1.95%	1.30%	1.13%	1.46%	
14	2.384	0.352	0.42	2.357	0.361	0.411	1.13%	2.56%	2.14%	1.94%	
15	2.384	0.352	0.42	2.351	0.361	0.427	1.38%	2.56%	1.67%	1.87%	
16	2.408	0.352	0.42	2.351	0.345	0.431	2.45%	1.99%	2.62%	2.35%	
17	2.307	0.361	0.415	2.269	0.37	0.41	1.65%	2.43%	1.20%	1.76%	
18	2.307	0.37	0.415	2.262	0.378	0.406	1.95%	2.16%	2.17%	2.09%	
19	2.172	0.37	0.415	2.099	0.362	0.419	3.36%	2.16%	0.96%	2.16%	
20	2.172	0.37	0.415	2.097	0.367	0.421	3.45%	0.81%	1.45%	1.90%	
	Average							2.37%	2.32%	1.47%	2.05%

The experiment results showed that, for the one-dimensional dimension, the average relative error for the length was 2.37%, the average relative error for the width was 2.32%, and the average relative error for the height was 1.47%. For the three-dimensional stereo size, the average error of the model was 2.05%. The prediction accuracy of the height of the model was slightly higher than that of its length and width; the model's average relative error for predicting its length, width, and height was less than 2.5%. These results verify the accuracy and stability of the algorithm.

5. Conclusions and Future Work

In conclusion, this study proposed an algorithm for measuring the dimensions of a truck's outer contour based on the UAV binocular stereo vision. The algorithm uses UAV as the platform. Compared with the traditional over-limit detection technology, the UAV platform has the following advantages: It is portable, low cost, suitable for mobile law enforcement; operates over a long distance; has a wide field of view; and is strongly scalable. The UAV platform can also monitor vehicle volume, allowing for estimated weight changes and facilitating other freight vehicle applications before and after weighing.

Given the current experimental methods, algorithm performance is limited by the effect of point cloud 3D reconstruction. Future research work should focus on improving the algorithm associated with 3D reconstruction, improving the operational efficiency and accuracy of point cloud reconstruction, and conducting further experiments on real vehicles to explore the most appropriate flying height and speed of UAVs for the different types of trucks. This work demonstrates the ability to calculate vehicle outline dimensions based on 3D reconstruction. This is applicable to static vehicle scenes but not to real-time vehicle 3D information detection scenes. As such, future studies should also consider realizing the real-time, high-precision recognition of the dimensions of vehicles' outer contours. Despite this, this study provides valuable information about the use of UAV measurements to assess vehicle 3D information and the risk of over-limit conditions.

Author Contributions: Conceptualization, L.H. and W.S.; methodology, L.H.; software, S.L.; validation, P.D.; formal analysis, L.H.; investigation, S.L.; resources, W.S.; data curation, L.H.; writing—original draft preparation, L.H.; writing—review and editing, S.L. and P.D.; visualization, L.H.; supervision, P.D.; project administration, W.S.; funding acquisition, W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Jilin Province Transportation Innovation Development Support Project with grant number 2020-1-12, the General Program of National Natural Science Foundation of China with grant numbers 52172385 and 52131203, and supported by Exploration Foundation of State Key Laboratory of Automotive Simulation and Control.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions.

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

References

1. Wardhana, K.; Hadipriono, F.C. Analysis of Recent Bridge Failures in the United States. *J. Perform. Constr. Facil.* **2003**, *17*, 144–150. [[CrossRef](#)]
2. Corral-Soto, E.R.; Elder, J.H. Slot Cars: 3D Modelling for Improved Visual Traffic Analytics. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 889–897. [[CrossRef](#)]
3. Chen, Y.; Liu, S.; Shen, X.; Jia, J. DSGN: Deep Stereo Geometry Network for 3D Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020. [[CrossRef](#)]
4. Wang, L.; Li, H.; Li, S.; Bie, Y. Gradient illumination scheme design at the highway intersection entrance considering driver's light adaption. *Traffic Inj. Prev.* **2022**, *23*, 266–270. [[CrossRef](#)] [[PubMed](#)]
5. Kocur, V.; Ftáčnik, M. Detection of 3D bounding boxes of vehicles using perspective transformation for accurate speed measurement. *Mach. Vis. Appl.* **2020**, *31*, 1–15. [[CrossRef](#)]
6. Sochor, J.; Spanhel, J.; Herout, A. BoxCars: Improving Fine-Grained Recognition of Vehicles Using 3-D Bounding Boxes in Traffic Surveillance. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 97–108. [[CrossRef](#)]
7. Chabot, F.; Chaouch, M.; Rabarisoa, J.; Teuliere, C.; Chateau, T. Deep MANTA: A Coarse-to-Fine Many-Task Network for Joint 2D and 3D Vehicle Analysis from Monocular Image. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
8. Mousavian, A.; Anguelov, D.; Flynn, J.; Kosecka, J. 3D bounding box estimation using deep learning and geometry. In Proceedings of the IEEE 2017 Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [[CrossRef](#)]
9. Yang, S.; Scherer, S. CubeSLAM: Monocular 3-D Object SLAM. *IEEE Trans. Robot.* **2019**, *35*, 925–938. [[CrossRef](#)]
10. Peng, W.; Pan, H.; Liu, H.; Sun, Y. IDA-3D: Instance-Depth-Aware 3D Object Detection From Stereo Vision for Autonomous Driving. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020. [[CrossRef](#)]
11. Wang, Y.; Chao, W.-L.; Garg, D.; Hariharan, B.; Campbell, M.; Weinberger, K.Q. Pseudo-LiDAR From Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [[CrossRef](#)]
12. Li, P.; Chen, X.; Shen, S. Stereo R-CNN Based 3D Object Detection for Autonomous Driving. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [[CrossRef](#)]
13. Shi, Y.; Guo, Y.; Mi, Z.; Li, X. Stereo CenterNet-based 3D object detection for autonomous driving. *Neurocomputing* **2021**, *471*, 219–229. [[CrossRef](#)]
14. Zhou, J.; Jin, L.; Wang, X.; Sun, D. Resilient UAV Traffic Congestion Control Using Fluid Queuing Models. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 7561–7572. [[CrossRef](#)]
15. Estevez, J.; Lopez-Guede, J.M.; Graña, M. Quasi-stationary state transportation of a hose with quadrotors. *Robot. Auton. Syst.* **2015**, *63*, 187–194. [[CrossRef](#)]
16. Drones: A Vision Has Become Reality. 2019. Available online: <http://www.swisspost.ch/drones> (accessed on 1 July 2019).
17. Wang, S.; Jiang, F.; Zhang, B.; Ma, R.; Hao, Q. Development of UAV-Based Target Tracking and Recognition Systems. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3409–3422. [[CrossRef](#)]
18. Boudjit, K.; Larbes, C. Detection and Implementation Autonomous Target Tracking with a Quadrotor AR.Drone. In Proceedings of the Icimco 2015—12th International Conference on Informatics in Control, Automation and Robotics, Alsace, France, 21–23 July 2015. [[CrossRef](#)]

19. Liu, S.; Wang, S.; Shi, W.; Liu, H.; Li, Z.; Mao, T. Vehicle tracking by detection in UAV aerial video. *Sci. China Inf. Sci.* **2019**, *62*, 24101. [[CrossRef](#)]
20. Wang, L.; Chen, F.; Yin, H. Detecting and tracking vehicles in traffic by unmanned aerial vehicles. *Autom. Constr.* **2016**, *72*, 294–308. [[CrossRef](#)]
21. Guido, G.; Gallelli, V.; Rogano, D.; Vitale, A. Evaluating the accuracy of vehicle tracking data obtained from Unmanned Aerial Vehicles. *Int. J. Transp. Sci. Technol.* **2016**, *5*, 136–151. [[CrossRef](#)]
22. Kanistras, K.; Martins, G.; Rutherford, M.J.; Valavanis, K.P. Survey of Unmanned Aerial Vehicles (UAVs) for Traffic Monitoring. In *Handbook of Unmanned Aerial Vehicles*; Springer: Dordrecht, The Netherlands, 2015. [[CrossRef](#)]
23. Elloumi, M.; Dhaou, R.; Escrig, B.; Idoudi, H.; Saidane, L.A. Monitoring road traffic with a UAV-based system. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, 15–18 April 2018. [[CrossRef](#)]
24. Yang, L.; Li, Y.; Li, X.; Meng, Z.; Luo, H. Efficient plane extraction using normal estimation and RANSAC from 3D point cloud. *Comput. Stand. Interfaces* **2021**, *82*, 103608. [[CrossRef](#)]
25. Kwon, Y.; Nunley, D.; Gardner, J.P.; Balazinska, M.; Howe, B.; Loebman, S. Scalable Clustering Algorithm for N-Body Simulations in a Shared-Nothing Cluster. In Proceedings of the Scientific and Statistical Database Management, Heidelberg, Germany, 30 June–2 July 2010. [[CrossRef](#)]
26. Bentley, J.L. Multidimensional binary search trees used for associative searching. *Commun. ACM* **1975**, *18*, 509–517. [[CrossRef](#)]
27. Zhang, X.; Jiang, X.; Jiang, J.; Zhang, Y.; Liu, X.; Cai, Z. Spectral–Spatial and Superpixelwise PCA for Unsupervised Feature Extraction of Hyperspectral Imagery. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–10. [[CrossRef](#)]
28. Rosenblatt, M. Remarks on Some Nonparametric Estimates of a Density Function. *Ann. Math. Stat.* **1956**, *27*, 832–837. [[CrossRef](#)]
29. Parzen, E. On Estimation of a Probability Density Function and Mode. *Ann. Math. Stat.* **1962**, *33*, 1065–1076. [[CrossRef](#)]
30. Ahamada, I.; Flachaire, E. *Non-Parametric Econometrics*; Oxford University Press: Oxford, UK, 2010.
31. Härdle, W.; Werwatz, A.; Müller, M.; Sperlich, S. *Nonparametric and Semiparametric Models*; Springer: Berlin/Heidelberg, Germany, 2004.
32. Qin, Z.; Li, W.; Xiong, X. Estimating wind speed probability distribution using kernel density method. *Electr. Power Syst. Res.* **2011**, *81*, 2139–2146. [[CrossRef](#)]