Vehicle pose estimation algorithm for parking automated guided vehicle

Zhixiong Ning¹, Xin Wang¹, Jun Wang² and Huafeng Wen²

Abstract
Parking automated guided vehicle is more and more widely used for efficient automatic parking and one of the tough challenges for parking automated guided vehicle is the problem of vehicle pose estimation. The traditional algorithms rely on the profile information of vehicle body and sensors are required to be mounted at the top of the vehicle. However, the sensors are always mounted at a lower place because the height of a parking automated guided vehicle is always beyond 0.2mm, where we can only get the vehicle wheel information and limited vehicle body information. In this article, a novel method is given based on the symmetry of wheel point clouds collected by 3-D lidar. Firstly, we combine cell-based method with support vector machine classifier to segment ground point clouds. Secondly, wheel point clouds are segmented from obstacle point clouds and their symmetry are corrected by iterative closest point algorithm. Then, we estimate the vehicle pose by the symmetry plane of wheel point clouds. Finally, we compare our method with registration method that combines sample consensus initial alignment algorithm and iterative closest point algorithm. The experiments have been carried out.

Keywords
Parking AGV, vehicle pose estimation, 3-D point clouds, ground segmentation, registration algorithm

Date received: 17 April 2019; accepted: 24 October 2019

Introduction
Parking problem is becoming a major issue in urban development due to the increase in number of vehicles. Many smart parking systems have been developed in recent years, which mainly focus on parking safety¹,² and finding parking space.³ For space-saving parking, stereo garage is one of the solutions. Parking automated guided vehicle (AGV) is a flexible robot that has been used for automatic parking in flat parking lots. Since there is no need to reserve space for opening vehicle doors, entering and exiting parking stall, the parking space is greatly saved. Besides, parking lot accidents are avoided due to automatic parking and multi-robot cooperations will make parking more efficient.⁴

A typical parking AGV was designed and developed in 2014.⁵ It is bigger and lower flexibility than most mobile robots. In 2017, two types of pallet parking AGVs with lidar⁶ and vision navigation⁷ were developed, respectively. Like most parking AGVs, an extra mechanical platform for transporting vehicle is primarily provided and users is required to park the vehicle on it accurately. Different from these parking AGVs, clamp-on parking AGV does not rely on the extra mechanical platform and is only required to park the vehicle to parking buffer zone, which effectively reduces security risks and improves user experience. So the

¹ School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen, Shenzhen, People’s Republic of China
² Shenzhen Fine Automation Co. Ltd, Shenzhen, People’s Republic of China

Corresponding author:
Xin Wang, School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen, Shenzhen University Town, Nanshan District, Shenzhen, 518055, People’s Republic of China.
Email: wangxinsz@hit.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
vehicle pose estimation technology is required to be adopted. More than 10 years ago, many scholars have begun to research this topic in autonomous driving based on lidar data. In general, these methods can be roughly divided into two categories: feature-based\textsuperscript{8–11} and measurement model-based.\textsuperscript{12–15}

Since most vehicles are in the shape of a box, the partial profiles of them are like the letter L. Thus, L-shape feature can be used to represent them with a simple structure, which is widely used in feature-based methods. Gan et al.\textsuperscript{8} and Kim et al.\textsuperscript{9} proposed an efficient algorithm to fit a cluster of 2-D laser scan points with an L-shape and got the vehicle pose by this feature. Wang et al.\textsuperscript{10} determined the interest area of vehicle point clouds based on L-shape feature, and used RANSAC (Random Sample Consensus) algorithm to fit straight line of points in interest region to calculate the orientation of vehicle. Mertz et al.\textsuperscript{11} used line fit and corner fit for features extraction of a segment to estimate vehicle pose. However, just as stated in Morris et al.,\textsuperscript{12} feature-based methods are sensitive to noises and interferences.

In measurement model-based methods, vehicles are usually represented as some rectangles or cuboids, and sampling or optimization techniques are used to estimate the vehicle pose. Morris et al.\textsuperscript{12} proposed four-rectangles measurement model to represent vehicle point clouds and used gradient-based optimization method to obtain vehicle pose. Chen et al.\textsuperscript{13} combined a likelihood-field-based model with scaling series algorithm to realize real-time vehicle pose estimation based on four-rectangles model. Three-rectangles measurement model is proposed by Petrovskaya et al.\textsuperscript{14} and they used importance sampling technique to estimate vehicle pose. Wojke and Häselich\textsuperscript{15} extended Petrovskaya’s method in unstructured environments and realized vehicle pose estimation and tracking in real time. However, the results obtained by measurement model-based methods may be locally optimal because of the inappropriate initial value.

Figure 1 shows a novel clamp-on parking AGV that was introduced in 2018, which is a heavy load smart mobile robot and its maximum load can reach 3 tons but the height of the parking AGV is only 0.125 m. It works by moving to the bottom of the target vehicle and sensors are required to be mounted at a lower height, which greatly limits sensors’ view. Besides, in order to save parking space, the distance between target vehicle and parking AGV is not large. Thus, when we use lidar to collect data, the number of lidar beams hitting on vehicle body would be small and the profile information of the whole vehicle body cannot be obtained. Since feature-based methods and measurement model-based methods rely on the profile information of vehicle body, they cannot be applied in this case.

In 3-D point clouds processing, ground segmentation is very important, which is the precondition of other higher processes, such as obstacle detection, object classification, and recognition. Generally, approached of ground segmentation include cell-based methods\textsuperscript{16–19} and model-based methods.\textsuperscript{20–22}

Cell-based method was firstly proposed by Thrun et al.\textsuperscript{16} They project 3-D point clouds to 2.5-D grids and segment the ground point clouds by calculating the maximum height difference of each cell, which was widely used in the 2007 DARPA Grand Challenge competition.\textsuperscript{17} Because Thrun’s method relies on the minimum and maximum height of the points in the cell, it cannot segment the overhanging structures. Thus, Douillard et al.\textsuperscript{18} proposed an improved cell-based method based on cell’s gradient. If the gradient is less than the given threshold, the cell is the ground cell. Based on gradient threshold, Vu et al.\textsuperscript{19} combined adaptive break point detection with mean height evaluation to segment the ground point clouds. The algorithm obtains a better result in uneven and sparse regions than Douillard’s method. Because the cell-based methods only use the local information, they are sensitive to observation noises.

Model-based methods segment the ground by fitting geometric shape of point clouds. Himmelsbach et al.\textsuperscript{20} proposed a method for segmenting large-scale remote 3-D point clouds. The method divides the polar grid map into many segments and uses the line extraction algorithm to estimate local ground plane. Different from Himmelsbach’s method, Chen et al.\textsuperscript{21} used Gaussian process regression to classify ground and obstacles for each segment in polar grid map. Because the point clouds are divided into independent segments, the continuity of the overall ground height cannot be guaranteed. Since the ground is flat in most cases, Lam et al.\textsuperscript{22} used the plane RANSAC algorithm followed by the least square fitting on point cloud data to segment the ground point clouds. However, it will over-segment ground point clouds because the surface of some obstacles may be parallel to the ground.

In this article, we propose a vehicle pose estimation method based on the symmetry of wheel point clouds for the clamp-on parking AGV, which can be used to estimate vehicle pose when the mounting heights of sensors are very low. Before wheel point clouds segmentation, we combine Thrun’s cell-based method with support vector machine (SVM) classifier\textsuperscript{23} to segment ground point clouds. The global information and local features of point clouds are

![Figure 1. Clamp-on parking AGV. AGV: automated guided vehicle.](image-url)
used. It has a good performance and is not sensitive to the observation noises.

The rest of this article is structured as follows. In the next section, we introduce the ground segmentation algorithm based on cell-based method and SVM classifier. The third section elaborates the steps of the vehicle pose estimation algorithm with 3-D lidar. The experiments and results are offered in the fourth section. Finally, we summarize this article in the fifth section.

Research on ground segmentation algorithm

Figure 2 is the procedures of ground segmentation. Firstly, we use cell-based method and pass-through filter to pre-segment some ground point clouds and obstacle point clouds, respectively. Then, the normal vectors and height features from pre-segmented point clouds are extracted. Next, these features will be trained as the training data of SVM classifier. Finally, we use the optimal model of SVM classifier to predict the ground and obstacle point clouds.

Ground and obstacle pre-segmentation

We take a $W \times W$ plane in $xy$ plane and divide it into $N \times N$ cells, as shown in Figure 3. The point clouds are projected vertically to the cell plane. Thus, for a point $p (x, y, z)$ in lidar coordinate system, the cell it belongs to is

$$p \in C \left[ x \left( \frac{N}{W} + \frac{N}{2} \right), y \left( \frac{N}{W} + \frac{N}{2} \right) \right]$$

Figure 2. Procedures of ground segmentation.

Figure 3. Cell-based method.

where $W \times W$ is the plane size and $N \times N$ is the number of cells. The first cell is the lower-right cell. The threshold is set to $h$ and the maximum height difference $\Delta h$ of each cell is calculated. If $\Delta h \geq h$, the cell is set to a ground cell. All points in ground cell form the ground point clouds. The input point clouds and the ground pre-segmentation results are shown in Figure 4(a) and (b), respectively.

Because the height of obstacle point clouds is higher than ground point clouds, a lower threshold of pass-through filter in $z$-axis direction that is larger than the average height of the ground point clouds segmented by cell-based method is given and the upper threshold is 2 m. Besides, the points far from the lidar center are filtered. Thus, we can get a part of obstacle point clouds, as shown in Figure 4(c).

Ground and obstacle classification based on SVM classifier

Since the driving area is relatively flat, the normal vectors of ground point clouds will be approximately parallel to the $z$-axis of lidar coordinate system. Besides, the height of the ground is lower than the obstacles. Therefore, the normal
vectors and the height of the points are extracted as the SVM training features.

For a point, its normal vector coincides with the normal of the tangent plane of the surface formed by this point and its neighbors. Thus, it can be obtained by principal component analysis method.\textsuperscript{24} Let the $k$-nearest points set of a point $p$ is $P = \{p_1, p_2, \ldots, p_k\}$ and we can get the corresponding covariance matrix $C$ as follows

$$C = \frac{1}{k} \begin{bmatrix} p_1 - \bar{p} & \cdots & p_k - \bar{p} \\ \cdots & \cdots & \cdots \\ p_k - \bar{p} & \cdots & p_1 - \bar{p} \end{bmatrix}$$ \hfill (2)

where $\bar{p}$ is the center of gravity of $P$. Let $\lambda_1, \lambda_2, \text{ and } \lambda_3$ be the eigenvalues of the covariance matrix $C$, $0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3$, and the eigenvectors corresponding to the eigenvalues are $n_1, n_2, \text{ and } n_3$

$$C \cdot n_j = \lambda_j \cdot n_j, \ j \in \{1, 2, 3\}$$ \hfill (3)

where $n_j (n_1, n_2, n_3)$ is the normal vector of $p$. The normal estimation result of point clouds is shown in Figure 5.

After feature extraction, we can get the training data $x$

$$x = \begin{bmatrix} x_1 & y_1 & z_1 & h_1 \\ x_2 & y_2 & z_2 & h_2 \\ \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & z_N & h_N \end{bmatrix}$$ \hfill (4)

where $N$ is the number of pre-segmented point clouds. The ground point clouds are set to the positive samples and the obstacles point clouds are set to the negative samples. Thus, the training model is

$$f(W) = \text{sign}(W^T x + b)$$ \hfill (5)

The optimization function is

$$\text{max margin}(W, b) = \max_{W, b, x_i} \frac{1}{||W||} |W^T x_i + b|$$ \hfill (6)

After getting the optimal solution of $W$ and $b$, we can use the model to predict whether a point is a ground point or an obstacle point. In order to avoid under-segmentation, the points below the average height of ground point clouds will eventually be filtered.

**Vehicle pose estimation based on 3-D lidar**

**Wheel point clouds segmentation**

Since the lidar is mounted at a very low height, when parking AGV is behind the vehicle, the vehicle point clouds collected by the lidar mainly consist of two parts: wheel point clouds and chassis point clouds. As shown in Figure 6, the shape of the vehicle point clouds is like the letter $n$ and the wheel point clouds are segmented from obstacle point clouds based on this feature.

The minimum height of each cell can be calculated based on cell-based method. The vehicle cell is set to the suspended obstacle cell if the minimum height is greater than the given threshold. Because the distance between parking AGV and target vehicle is small, vehicle chassis point clouds contain more points than other suspended objects. We use Euclidean cluster algorithm\textsuperscript{25} to cluster the point clouds of suspended objects and select the largest cluster as the vehicle chassis point clouds. In Euclidean cluster algorithm, if the distance between two points is small, they belong to the same cluster.

As shown in Figure 7, the wheel point clouds is under the vehicle chassis point clouds. Thus, a sector region of interest based on vehicle chassis point clouds can be defined to segment wheel point clouds. The Gaussian
Correction of wheel symmetry

In theory, left and right wheels are strictly symmetrical. However, because the distances from lidar center to left and right wheels are always different, when using lidar to collect point cloud data, the number of points in left and right wheel point clouds will probably be different, which causes the symmetry errors of the wheel point clouds. Thus, it is necessary to correct the wheel point clouds before obtaining its symmetry plane. In this article, the symmetry of the wheel point clouds is corrected by registration algorithm.

To improve the registration accuracy, number of the left and right wheel point clouds should be balanced. The vertical angle $\omega$ of each point $p(x, y, z)$ can be calculated as follows

$$
\omega = \arctan\left(\frac{z}{\sqrt{x^2 + y^2}}\right)
$$

The points are classified into the same category with the same vertical angle. If the proportion that the number of points belong to the same wheel point clouds is small, the category will be filtered.

Assuming that the residual wheel point clouds $C$ is $\{p_1, p_2, \ldots, p_n\}$ and $p_i$ is a point, the center of gravity of the point clouds $C$ is given by

$$
W_g = \frac{1}{n} \sum_{i=1}^{n} p_i = \frac{1}{2} (L_g + R_g)
$$

where $L_g(x_{lg}, y_{lg}, z_{lg})$ and $R_g(x_{rg}, y_{rg}, z_{rg})$ are the center of gravity of the left and right wheel point clouds, respectively.

Let $W_g$ be $(x_{wg}, y_{wg}, z_{wg})$ and we can get the normal vector of the symmetry plane

$$
n_p = (A, B, C) = (x_{lg} - x_{rg}, y_{lg} - y_{rg}, z_{lg} - z_{rg})
$$

Therefore, the symmetry plane equation is

$$
Ax + By + Cz - (Ax_{wg} + By_{wg} + Cz_{wg}) = 0
$$

The symmetry point $p_i'$ of $p_i(x_i, y_i, z_i)$ with respect to the symmetry plane is

$$
\begin{bmatrix}
x'_i \\
y'_i \\
z'_i
\end{bmatrix} = -\left(\frac{Ax_i + By_i + Cz_i + D}{A^2 + B^2 + C^2}\right) \cdot
\begin{bmatrix}
A \\
B \\
C
\end{bmatrix} +
\begin{bmatrix}
x_i \\
y_i \\
z_i
\end{bmatrix}
$$

where $D = Ax_{wg} + By_{wg} + Cz_{wg}$.

The symmetry point is calculated for each point in left wheel point clouds and we can get the symmetry point clouds $L'$ of the left wheel point clouds $L$. Let the right wheel point clouds $S$ be $\{p_1, p_2, \ldots, p_n\}$ and the symmetry point clouds $L'$ be $\{p'_1, p'_2, \ldots, p'_n\}$.

We assume the Euclidean Transformation $T$ from $L'$ to $S$ is $(R, t)$. The iterative closest point (ICP) algorithm minimizes the distances between points in $S$ and $L'$. In each iteration, ICP computes the correspondences between $S$ and $L'$, and then $(R, t)$ is calculated to minimize the objective function

$$
\min_{R,t} J = \sum_{i=1}^{n} \sum_{j=1}^{m} \omega_{ij} \left\| p_i - (R p'_i + t) \right\|^2
$$

where $\omega_{ij}$ is the weight of correspondence. The maximum matching threshold $d_{max}$ of point pairs is given before registration. If the distance between a point pair is greater than $d_{max}$, $\omega_{ij} = 0$. Otherwise, $\omega_{ij} = 1$. After ICP registration, we can get the aligned point clouds. The aligned point clouds and the left wheel point clouds form the new wheel point clouds.

Vehicle pose estimation based on wheel symmetry

The homogeneous coordinates of a point in 3-D space can be represented as $p = [x \ y \ z \ 1]^T$. After ICP registration, we can get the new wheel point clouds $H_1$.
Thus, the normal vector of symmetry plane \( P_1 \) of point clouds \( H_1 \) is \( n_1 = [a_1, b_1, c_1]^T \), where \( c_1 > 0 \). The angle \( \alpha \) between \( P_1 \) and \( xy \) plane is
\[
\alpha = -\frac{b_1}{\|b_1\|} \cdot \cos^{-1}\left(\frac{c_1}{\sqrt{a_1^2 + b_1^2 + c_1^2}}\right)
\]  
(15)

In order to reduce the influence of unevenness on the ground, we rotate the wheel point clouds \( H_1 \alpha \) around the \( x \)-axis and we can get
\[
H_2 = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos \alpha & -\sin \alpha & 0 \\
0 & \sin \alpha & \cos \alpha & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \cdot H_1
\]  
(16)

The symmetry plane \( P_2 \) of point clouds \( H_2 \) is perpendicular to \( xy \) plane. Assuming that the normal vector of \( P_2 \) is \( n_2 = [a_2, b_2, c_2]^T \), where \( b_2 > 0 \), the horizontal azimuth of the target vehicle with respect to the lidar is given by
\[
\theta = -\frac{a_2}{\|a_2\|} \cdot \cos^{-1}\left(\frac{b_2}{\sqrt{a_2^2 + b_2^2 + c_2^2}}\right)
\]  
(17)

We rotate the wheel point clouds \( H_2 \theta \) around the \( z \)-axis of the lidar coordinate system and we can get the new wheel point clouds \( H_3 \) is
\[
H_3 = \begin{bmatrix}
\cos \theta & -\sin \theta & 0 & 0 \\
\sin \theta & \cos \theta & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \cdot H_2
\]  
(18)

The symmetry plane of \( H_3 \) is parallel to \( xz \) plane. Thus, the distance in \( y \)-direction between vehicle center and lidar center is
\[
\Delta y = \frac{1}{m+n} \left(\sin \theta \sum_{i=1}^{m+n} x_i + \cos \alpha \cos \theta \sum_{i=1}^{m+n} y_i - \sin \alpha \cos \theta \sum_{i=1}^{m+n} z_i\right)
\]  
(19)

We translate \( H_3 \) to \( \Delta y \) and we can get the final wheel point clouds. In theory, the symmetry plane of final wheel point clouds is coincident with \( xz \) plane. We suppose the \( xz \) plane of lidar coordinate system is coincident with the bilateral symmetry plane of parking AGV. Thus, in order to align to the target vehicle, the adjustment of parking AGV is
\[
P = [-\Delta y, -\theta]^T
\]  
(20)

where \( \Delta y \) is the traverse displacement and \( \theta \) is the rotation angle.

**Registration method combing SAC-IA and ICP algorithms**

If we have the source point clouds at the origin location, the object pose can be obtained by registration method.\(^{27}\) In ICP algorithm, the maximum matching threshold \( d_{\text{max}} \) should be given to get the correspondent points. If the distance between two point clouds is large, \( d_{\text{max}} \) should also be large. However, large matching threshold will cause incorrect correspondences, which will make registration failed. Thus, initial transformation should be calculated by rough registration if there is no overlap between two point clouds.

Point feature histograms (PFH) is a 3-D feature descriptor, which represents the surface model properties of a point. Fast point feature histograms (FPFH) is a simplified form of PFH and its computation is more time-saving than PFH.\(^{28}\) We assume the neighbors of point \( p \) is \( N = \{p_1, ..., p_k\} \), where points in \( N \) are in the sphere. Sphere’s radius is \( r \) and \( p \) is the center. For each pair of points \( p \) and \( p_k \) in \( N \), we calculate their normal vectors \( n \) and \( n_k \). The angular variation of \( n \) and \( n_k \) is
\[
\begin{bmatrix}
\alpha \\
\beta \\
\theta
\end{bmatrix} = \begin{bmatrix}
\cos^{-1}(\langle v \cdot n_k\rangle /||p_k - p||) \\
\cos^{-1}(\langle u \cdot (p_k - p)\rangle /||p_k - p||) \\
\tan^{-1}(w \cdot n_k, u \cdot n_k)
\end{bmatrix}
\]  
(21)

where \( u = n, v = (p - p_k) \times u, w = u \times v \). The angular variations of each point pair consist of the simplified point feature histograms (SPFH). Thus, the FPFH of point \( p \) is
\[
\text{FPFH}(p) = \text{SPFH}(p) + \frac{1}{k} \sum_{k=1}^{k} \omega_k \cdot \text{SPFH}(p_k)
\]  
(22)

where \( k \) is the \( k \) neighbors of point \( p \) and \( \omega_k \) is the distance between point \( p \) and point \( p_k \).

After getting the FPFH features of the whole point clouds, we use sample consensus initial alignment (SAC-IA) algorithm\(^{28}\) to register the point clouds. Different from ICP algorithm, SAC-IA algorithm finds the correspondence of a point by FPFH features and the rigid transformation \( T_i \) for each point pair will be calculated. The performance of \( T_i \) is evaluated by
\[
J = \sum_{i=1}^{n} H(e_i)
\]  
(23)

where \( H(e_i) \) is a Huber penalty measure
\[
H(e_i) = \begin{cases}
0.5e_i^2 & ||e_i|| \leq t_e \\
0.5t_e(2||e_i|| - t_e) & ||e_i|| > t_e
\end{cases}
\]  
(24)

where \( t_e \) is a given threshold and \( e_i \) is the distance between the target point clouds and the aligned point clouds transformed by the transformation of the \( i \)th matching point pair.
Finally, we find the best transformation from all transformations to minimize $J$.

After SAC-IA registration, we can get the initial transformation $T_1$ and the aligned point clouds $H_1'$. Then, we use ICP algorithm to register point clouds $H_1'$ to the target point clouds and we can get the transformation $T_2$. Thus, the transformation $T'$ from original point clouds to target point clouds is $T_2T_1$. If the target point clouds are the wheel point clouds whose symmetry plane is coincident with the $xz$ plane of lidar coordinate system, the adjustment of parking AGV is

$$P' = \begin{bmatrix} -\Delta y' & -\theta' \end{bmatrix}^T = \begin{bmatrix} -\sin^{-1}(T'[0][1]) & T'[1][3] \end{bmatrix}^T$$

(25)

**Experiments and results**

The lidar used in this article is Velodyne VLP-16 and the experimental platform is shown in Figure 9. In order to get the vehicle point clouds from the obstacle point clouds, the ground point clouds should be segmented first. Figure 10 shows the ground segmentation results. It can be seen that the ground point clouds below the vehicle are erroneously segmented to obstacles by cell-based method but there are almost no ground point clouds in obstacle point clouds by our method. Our method can get a better result.

Figure 11 is the results of vehicle segmentation. The vehicle chassis point clouds can be segmented effectively by suspended object extraction and Euclidean cluster algorithm, as shown in Figure 11(a). After that, the wheel point clouds are extracted as the nonsuspended objects from the sector region of interest, as shown in Figure 11(b).

Because the point clouds of the left and right wheels are not strictly symmetrical, the symmetry of the wheel point clouds should be corrected. Figure 12 shows the correction results. In Figure 12(a), white point clouds are the symmetry point clouds of the left wheel with respect to the symmetry plane of the wheel point clouds. We can find that the symmetry point clouds are not coincident with the right wheel point clouds and there are symmetry errors in wheel point clouds. After registration, the coincidence degrees of the wheel point clouds increase and the symmetry errors decrease, as shown in Figure 12(b).

To get the vehicle pose, the corrected wheel point clouds are transformed. Figure 13 shows the transformed results of wheel point clouds (white point clouds) by the vehicle pose estimation algorithm. It can be seen that the symmetry planes (white plane) of wheel point clouds are coincident with the $xz$ plane. In this case, lidar on parking AGV has aligned to the target vehicle. For comparison, we combine SAC-IA algorithm with ICP algorithm to register the wheel point clouds. The target point clouds are the wheel point

---

**Figure 9.** Experimental platform.

**Figure 10.** Results of ground segmentation. (a) Results of ground segmentation by cell-based method and (b) results of ground segmentation by our method.
The original point clouds are coincident with the target point clouds after registration.

In order to evaluate the precision of the vehicle pose estimation algorithm proposed in this article, we rotated and moved the lidar to different places. The lidar data were recorded by robot operating system (ROS) and we took out 100 scans from it to calculate the vehicle pose respectively.

Figure 15 is the estimation results of horizontal azimuth and Figure 16 is the estimation results of distance in y-direction. It can be seen that the results of our method are basically coincident with the registration method. Figure 17 is the Euclidean fitness scores of registration method, which are the sum of squared distances from aligned point clouds to target point clouds. The lower Euclidean fitness scores mean the better registration results and the vehicle pose estimation results are more accurate. From Figure 17, we can find that the Euclidean fitness scores are very low. Thus, the registration precision is high.
Figure 13. Vehicle pose estimation by transforming wheel point clouds.

Figure 14. Vehicle pose estimation by registration method combining SAC-IA algorithm with ICP algorithm. SAC-IA: sample consensus initial alignment; ICP: iterative closest point.

Figure 15. Estimation results of horizontal azimuth.

Figure 16. Estimation results of distance in y-direction.

Figure 17. Euclidean fitness score of registration method.

Figure 18. Processing time of vehicle pose estimation algorithm.
The processor used in this article is a standard laptop (Intel Core-i5-3749 CPU 3.2GHZ, 8G RAM) and the algorithm is run in ROS environment (C++). Figure 18 shows the processing time of vehicle pose estimation algorithm. It can be seen that the processing speed of our method is faster than registration method. The processing time consumption of our method is mainly in SVM classifier training and ICP registration.

Conclusions

By combining cell-based method with SVM classifier, the preparation procedure of training data is simplified and the results of ground segmentation are more exactly. The method by extracting the symmetry plane of the wheel point clouds solves the vehicle pose estimation problem in the case where the vehicle body information cannot be obtained. The registration method is given to provide a reference of vehicle pose because the registration accuracy is high according to the Euclidean fitness scores. The vehicle pose estimation results of our method are basically coincident with registration method and more time-saving. Besides, our method can estimate the pose of different vehicle models but registration method needs to build a database of different wheel point clouds models to provide the target registration point clouds. Experimental results indicate that the vehicle pose estimation method in this article is feasible and it has a high precision, which can be used on parking AGV.

In the future, we will consider data fusion of vision and lidar. Vehicle wheels recognition and regions of interest determination are implemented by vision, and then the vehicle pose is estimated based on lidar data. Besides, real-time performance of the algorithm should be improved for providing real-time feedback to the robot, which would probably improve the robustness and precision of the algorithm.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by grants from the Shenzhen Bureau of Science Technology and Information (Grant No. JCYJ20170413110656460).

ORCID iD

Zhixiong Ning @ https://orcid.org/0000-0002-9513-0486

References

1. Mahmud SA, Khan GM, Rahman M, et al. A survey of intelligent car parking system. J Appl Res Technol 2013; 11(5): 714–726.
2. Wang W, Song Y, Zhang J, et al. Automatic parking of vehicles: a review of literatures. Int J Auto Technol 2014; 15(6): 967–978.
3. Geng Y and Cassandras CG. New “smart parking” system based on resource allocation and reservations. IEEE T Intell Transp Syst 2013; 14(3): 1129–1139.
4. Serpen G and Dou C. Automated robotic parking systems: real-time, concurrent and multi-robot path planning in dynamic environments. Appl Intell 2015; 42(2): 231–251.
5. He T, Shao J, Zhang Y, et al. Recent Patents on automated transporting vehicle in automatic car parking system. Recent Pat Mech Eng 2016; 9(1): 2–8.
6. Wu Z, Wang X, Wang J, et al. Research on improved graph-based SLAM used in intelligent garage. In: 2017 IEEE international conference on information and automation (ICIA), Macau, China, 18–20 July 2017, pp. 592–597.
7. Xiao Z, Wang X, Wang J, et al. Monocular ORB SLAM based on initialization by marker pose estimation. In: 2017 IEEE international conference on information and automation (ICIA), Macau, China, 18–20 July 2017, pp. 678–682.
8. Gan Z, Wang C, and Yang M. A method for vehicle tracking and recognition based on scanning lidar radar. J Shanghai Jiaotong Univ 2009; 43(6): 921–926.
9. Kim D, Jo K, Lee M, et al. L-shape model switching-based precise motion tracking of moving vehicles using lidar scanners. IEEE T Intell Transp Syst 2018; 19(2): 598–612.
10. Wang X, Li K, Wang J, et al. Parameter identification of intelligent vehicle target based on 3D lidar radar. Automot Eng 2016; 38(9): 1147–1152.
11. Mertz C, Navarro-Serment LE, MacLachlan R, et al. Moving object detection with lidar scanners. J Field Robot 2013; 30(1): 17–43.
12. Morris DD, Hoffman R, and Haley P. A view-dependent adaptive matched filter for lidar-based vehicle tracking. In: International conference of robotics and applications, Cambridge, USA, 2–4 Nov 2009, pp. 1–9.
13. Chen T, Wang R, Dai B, et al. Likelihood-field-model-based dynamic vehicle detection and tracking for self-driving. IEEE T Intell Transp Syst 2016; 17(11): 3142–3158.
14. Petrovskaya A and Thrun S. Model based vehicle tracking for autonomous driving in urban environments. In: Robotics: Science and Systems (RSS), Zurich, Switzerland, 25–28 June 2008.
15. Wojke N and Häselich M. Moving vehicle detection and tracking in unstructured environments. In: IEEE international conference on robotics and automation (ICRA), St. Paul, Minnesota, 14–18 May 2012, pp. 3082–3087.
16. Thrun S, Monterdeiro M, Dahlkamp H, et al. Stanley: the robot that won the DARPA Grand Challenge. J Field Robot 2006; 23: 661–692.
17. Monterdeiro M, Becker J, Bhat S, et al. Junior: the Stanford entry in the urban challenge. J Field Robot 2008; 25: 569–597.
18. Douillard B, Underwood J, Melkumyan N, et al. Hybrid elevation maps: 3D surface models for segmentation. In: 2010 IEEE/RSJ international conference on intelligent robots and
19. Vu H, Nguyen HT, Chu PM, et al. Adaptive ground segmentation method for real-time mobile robot control. *Int J Adv Robot Syst* 2017; 14(6): 1–11.

20. Himmelsbach M, Hundelshausen FV, and Wuensche HJ. Fast segmentation of 3D point clouds for ground vehicles. In: *2010 IEEE intelligent vehicles symposium (IV)*, La Jolla, CA, 21–24 June 2010, pp. 560–565.

21. Chen T, Dai B, Wang R, et al. Gaussian-process-based real-time ground segmentation for autonomous land vehicles. *J Intell Robot Syst* 2014; 76(3–4): 563–582.

22. Lam J, Kusevic K, Mrstik P, et al. Urban scene extraction from mobile ground based lidar data. In: *International symposium on 3D data processing visualization and transmission*, Zadar, Croatia, 15–17 September 2010, pp. 1–8.

23. Chang YW, Hsieh CJ, Chang KW, et al. Training and testing low-degree polynomial data mappings via linear SVM. *J Mach Learn Res* 2010; 11: 1471–1490.

24. Rusu RB. Semantic 3D object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz* 2010; 24(4): 345–348.

25. Sparks DN. Algorithm AS 58: Euclidean cluster analysis. *J R Stat Soc C Appl* 1973; 22(1): 126–130.

26. Besl PJ and McKay ND. A method for registration of 3-D shapes. *IEEE T Pattern Anal* 1992; 14(2): 239–256.

27. Miao Y, Liu Y, Ma H, et al. The pose estimation of mobile robot based on improved point cloud registration. *Int J Adv Robot Syst* 2016; 13(2): 1–10.

28. Rusu RB, Blodow N, and Beetz M. Fast point feature histograms (FPFH) for 3D registration. In: *2009 IEEE international conference on robotics and automation (ICRA)*, Kobe, Japan, 12–17 May 2009, pp. 3212–3217.