IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 10, 2020, accepted April 29, 2020, date of publication May 11, 2020, date of current version June 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993934

Omni-Directional Obstacle Detection for Vehicles Based on Depth Camera

XIANGMO ZHAO, HUAYUE WU, ZHIGANG XU, AND HAIGEN MIN
School of Information Engineering, Chang'an University, Xi'an 710064, China

Corresponding authors: Huayue Wu (whyinvr@126.com) and Zhigang Xu (xuzhigang@chd.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1864204, and in part by the Natural Science Foundation of Shaanxi Province under grant 2018JQ6035.

ABSTRACT At present, the vehicle obstacle detection system usually uses different devices or sensors to perceive and obtain the obstacle information. However, omni-directional obstacle detection is difficult to realize because these devices or sensors are usually easy to be affected by environmental lighting and the material properties of the obstacle surface. Furthermore, most sensors have limited information regarding distance, which limits their application to omni-directional obstacle detection. To solve this problem, this paper proposes a method using depth camera for omni-directional obstacle detection. A method applying region growth for depth image and a fast inpainting method for depth image are proposed to extract and repair the obstacle regions in the depth images obtained by installing depth cameras around the car body. An improved method applying iterative normalized cut is also proposed to cluster and segment fragmentary and irregular obstacle regions to generate the complete obstacle regions. Finally, the obstacle regions are overviewed using a three-dimensional visualization method to realize omni-directional obstacle viewing. The results of experiments conducted in an environment with different obstacles during the day and night demonstrate that, compared with other methods, our proposed approach can effectively promote the ability to detect complex obstacles, and largely improve the detection speed; furthermore, obstacle detection using our method will be unaffected by environmental lighting. Each of these advantages provided by our method can significantly promote the driving safety of unmanned or other types of vehicles.

INDEX TERMS Omni-directional obstacle detection, Depth camera, Binocular vision, Normalized cut, Optical flow

I. INTRODUCTION

The numbers of vehicles on urban roads have increased with the development of transportation infrastructure, and more accidents are therefore occurring, making vehicle safety an important issue. For obstacle detection, many researches have focused on robots and autonomous vehicles. Numerous successful research institutes such as Mobileye, Daimler, and KIT Autonomous Vision Group (AVG) have proposed preferable research approaches in this area.

Vehicle environmental obstacle detection is a hot issue in the field of intelligent vehicle research. It is a prerequisite for intelligent vehicle to realize automatic driving, autonomous navigation and other functions. Among the methods for vehicle environmental obstacles detection, lidar, millimeter wave radar(MMW radar) and ultrasonic radar have been widely used.

The lidar detects the scene in front of the vehicle by emitting infrared laser beam, and measures the time delay of the reflected light to calculate the distance between the obstacle and the vehicle. The lidar has high measurement accuracy and wide detection range, but it is sensitive to bad weather (such as heavy rain and fog) and heavy and expensive, and can interfere with the laser signals emitted by other vehicles.

MMW radar can measure distance by detecting reflected wave, and it can also achieve enough accuracy under harsh conditions. However, MMW radar has a narrow view field (usually has the field with milli-radians level)and low lateral information accuracy, which means that unless the vehicle is far ahead, the radar can not detect vehicles from adjacent lanes.

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo.

Ultrasound radar is also a common obstacle detection method. However, ultrasonic radar is not suitable for omni-directional obstacle detection because of its poor long-range detection ability, vulnerability to noise interference, poor environmental perception accuracy and robustness.

Comparing with radar method, computer vision based method can accurately extract the three-dimensional contour of the target. It has the advantages of wide detection range, complete road information and remote sensing. It is a very important sensor in vehicle obstacle detection. However, as with other sensors, visual sensors are sensitive to weather and illumination conditions. At the same time, the calculation of disparity map and the analysis and reconstruction of three-dimensional scenes have large computation, which limits the application of binocular stereo vision in omni-directional obstacle detection.

At present, the depth camera gradually plays an important role in obstacle detection. It has the advantages of large detection distance, rich depth information, not affected by the environmental light, low cost and so on. Therefore this study proposes an omni-directional obstacle detection method using depth cameras. After installing depth cameras at different locations around the vehicle and determining the ROI, the obstacle area within the ROI is extracted using the proposed region growth method for depth images. In addition, an improved iterative normalized cut method is also proposed to cluster and segment the obstacles. Finally, omni-directional obstacle viewing is realized using 3D reconstruction and an overview of the obstacle regions.

II. RELATED STUDIES

To prevent traffic accidents, numerous methods have used sensors such as ultrasound, cameras, or lidar to obtain a real-time perception of the surrounding environment. Among such sensors, ultrasound achieves a strong reflectivity, but its point-to-point ranging method, slow sound wave propagation speed, required planar surface of the reflector, and interference between ultrasound waves limit its application in omni-directional obstacle detection. At present, studies on obstacle detection have focused on computer-vision [1] and lidar [2]–[8] based methods.

In the case of computer-vision based methods, the approach in [9] segments the obstacle region by using a given judgment threshold to obtain the binary image of binocular depth data; however, this method depends on the selected threshold, and thus it cannot segment the complete obstacles, and noise will seriously affect the obstacle detection. The methods in [5] and [10]–[12] use the V-disparity algorithm to detect obstacles. In [12], the original UV-disparity algorithm is applied to detect obstacles in an urban environment. However, this method is more sensitive to large obstacles, and will lead to a massive loss in the detection of small obstacles. Obstacle detection using a binocular camera and requiring binocular matching, has also been commonly applied. In [13]–[15], preferable matching methods were proposed. The method in [13] obtains an accurate binocular matching by training a convolutional neural network to compare image blocks. However, binocular matching is a time-consuming process, and incorrect matching will lead to incorrect obstacle detection. The approaches in [16]–[21] use an optical flow method to detect obstacles through monocular vision. In [18], an obstacle ROI region is established according to the different characteristics of the optical flow between the obstacle region and the background. The method described in [19] distinguishes between different obstacles using a combination of the optical flow and dense disparity information. In [21], the features of KLT are used to track and detect obstacles. However, the optical flow is very complex and time-consuming, and the method is therefore unsuitable for real-time omni-directional obstacle detection. In [31], inertial measurement unit (IMU) and other synchronized sensors are used to detect obstacles. In [22], fewer RGB cameras are applied to realize omni-directional obstacle detection. However, the use of fewer cameras causes a serious fisheye distortion, and the method suffers from other shortcomings, such as the need for a large amount of camera calibration and an inability to work at night. In addition, computer-vision based method is extremely sensitive to light, and obstacle detection under dark lighting is even worse, which may result in the shadows of obstacles being detected as new obstacles.

As examples of lidar-based method, the methods in [4], [5], [28] and [29] use lidar to detect obstacles, where [4] processes point clouds as 2D obstacles using the iterative least squares method. In [8], a Gauss mixture model based on 3D lidar is applied for obstacle detection. However, using lidar for omni-directional obstacle detection is not only expensive it is also easily affected by rainwater and dust. Furthermore, lidar typically has a large volume and blind areas when detecting obstacles closer to the car body, and is thus poorly suited for omni-directional obstacle detection.

In addition, deep learning has been increasingly applied to image region segmentation, such as in [23], [24], [30]. Because a deep learning method requires a large amount of sample data for training, it is difficult to achieve the requirement of real-time detection. The methods in [25], [27] apply machine learning to detect obstacles. Although this method obtains a better effect, its real-time performance cannot reach the real-time requirement of an omni-directional obstacle detection system when the vehicle is moving.

Table 1 summarized the characteristics of different methods mentioned above based on their different features. Besides the shortcoming of the computer-vision method in the table, it cannot work under low illumination environment. The method we proposed uses the depth camera to perceive the surrounding environment, which is not affected by environment lighting and the obstacle detection accuracy and speed are also greatly improved.

Although the depth camera has the advantages of large detection distance, rich depth information, not affected by the environmental light, low cost and so on, there are few studies regarding obstacle detection by depth camera. Obstacle detection by depth camera mainly uses binary threshold

TABLE 1. Characteristics of different methods mentioned above.

Detection mode	Detection method	Detection characteristic
Ultrasound		Poor environmental perception accuracy and robustness
Computer-vision	Binary threshold	Cannot completely segment the obstacles
	UV-disparity	Sensitive to large obstacles and easy to loss small obstacles
	Convolutional neural network	Large amount of sample data required and poor real-time performance
	Optical flow	Very complex and time-consuming
	Multi-camera	Suffer from fisheye distortion, and a large amount of camera calibration
	Deep learning	Large amount of sample data required and poor real-time performance
Lidar		Heavy and expensive

segmentation, multi frame background segmentation and clustering segmentation methods to extract obstacles, but the detection effect using these methods is not very well and the some extracted obstacles have incomplete shapes. To solve these problems, this paper proposes a new method using depth camera for omni-directional obstacle detection, which greatly improved omni-directional obstacle detection accuracy and speed in the vehicle environment.

III. SYSTEM OVERVIEW

The omni-directional obstacle detection method proposed in this paper can be used to obtain a depth image of an omni-directional environment through an evenly distributed installation of ten depth cameras placed on the car body. The installation positions are shown in Fig. 1 (a). Fig. 1 (b) shows the overall process used to extract omni-directional obstacles with this method. In order to ensure that the detection covers the whole omni-directional space, the view field of each depth camera is set to 36 degrees(this degree depends on the number of the depth camera and the type of depth camera. Ten depth cameras are used in this study), which ensures the coverage of omni-directional space. When detecting obstacles, all the depth cameras generate depth images at the same time. After all the cameras have generated the depth images, a panoramic depth image will be formed.

The processing steps required include collecting the original image of the scene from the depth cameras, which means obtaining depth images from the ten depth cameras; Note that the detection range of different depth cameras is different, the obstacle depth data within the detection range will be received normally, and the obstacle depth data outside the range will not be received. Therefore, the depth image generated by each depth camera within the detection range is the real and valid depth data without any data outside the detection range. So except for noise and holes, there are no other detection errors in the depth data. Determining the ROI region for the depth images obtained; and denoising the ROI regions using a fast adaptive median filter, which can effectively eliminate holes and pulse noises in the depth image and helps with the subsequent processing. After the denoising, as the exact position(including position and rotating angle) of the cameras in relation to the car coordinate system is known, the transformation matrix of each depth camera can be obtained, therefore a panoramic depth image can be generated by transforming the depth points in ten depth images into the car coordinate system using their own transformation matrices(Discussed in section III.F). A newly proposed method of region growth for depth image and a fast inpainting method are applied, which can extract connected obstacle regions and fast repair or fill in the holes in depth image. Meanwhile, Because the connected obstacle region will include a large number of small and fragmented obstacle regions, in this study an improved iterative normalized cut algorithm is proposed to cluster and segment all obstacles that extracted, and thus the small and fragmentary obstacles can be merged into the surrounding obstacles, which is conducive to the entire treatment of obstacles in the later steps. After clustering and segmenting using iterative normalized cut, an omni-directional obstacle distribution map is obtained by overlooking 3D point clouds in the depth images.

A. DETERMINATION OF THE ROI IN DEPTH IMAGE

Among the different image segmentation algorithms available, the region of interest (ROI) algorithm is very important for image processing. On the one hand, it indicates the image range that requires processing, and on the other, it improves the processing efficiency. For obstacle detection around a vehicle using the depth image, the ROI applied in this study determines the region through which the vehicle can pass, further reducing the region to be processed and thereby improving the efficiency.

For the 3D point cloud in a depth space captured by a depth camera, the vehicle attitude plane equation is defined to segment the space of the 3D point cloud, as shown in Fig. 2. The vehicle attitude plane is parallel to the vehicle chassis, and this plane determines the maximum height of the obstacles that the vehicle can pass. Fig. 2 (a) and (b) shows the status of the ROI region at different times while the vehicle is moving. The region below the vehicle attitude plane is the non-ROI region, where obstacle detection is not required. The region above the vehicle attitude plane is the ROI region, where the objects in this region are obstacles and need to be detected.

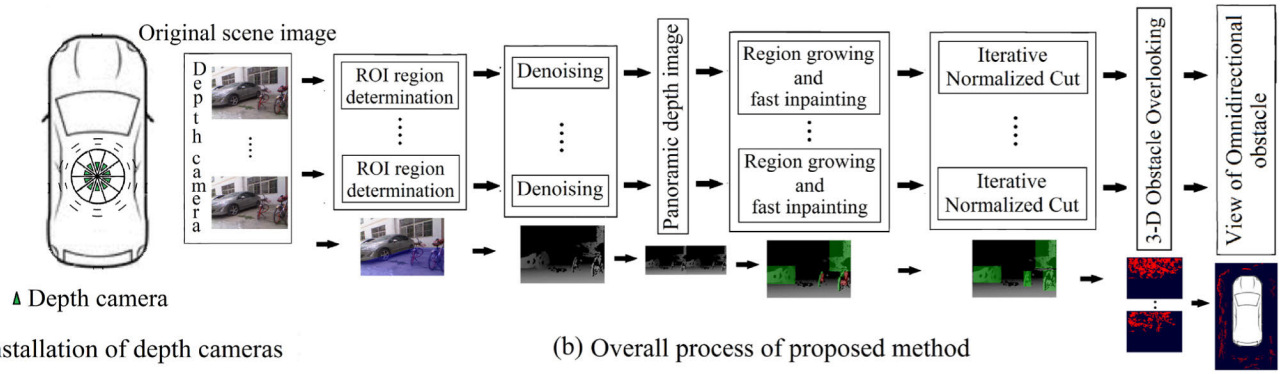


FIGURE 1. Installation of depth cameras and overall process of proposed method.

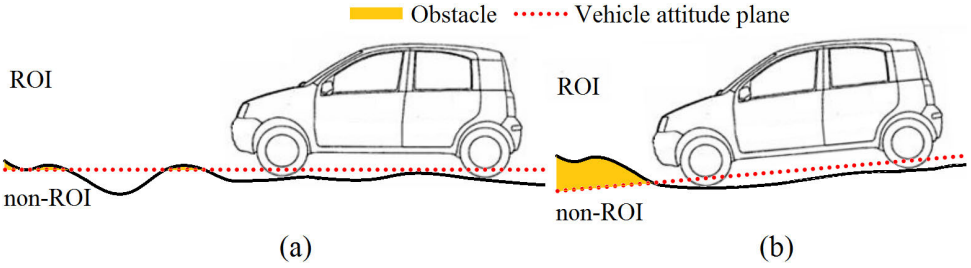


FIGURE 2. Establishment of vehicle attitude plane.

The vehicle attitude plane is defined as

$$Ax + By + Cz + D = 0 \tag{1}$$

and the coordinates of a certain point in the point cloud of the depth space are indicated by (x_0, y_0, z_0). If

$$Ax_0 + By_0 + Cz_0 + D > 0 \tag{2}$$

then this point belongs to the ROI space; otherwise, it is in the non-ROI space.

B. DENOISING OF THE DEPTH IMAGE

For a TOF(Time of flight) camera, such as the depth camera, the depth camera sends the light pulse to the target continuously, then receives the light returned from the object with the sensor, and obtains the distance between the target and the object by detecting the flight (round-trip) time of the light pulse. Because the reflectivity of different surfaces differs, when the reflectivity is low, the amount of light returned from the incident light will be reduced, and thus there will be a smaller number of holes in the depth image, similar to a black surface absorbing a large amount of light and forming holes in the image. We assume that the light of each pixel is reflected from a single location; in practice, however, the light reflected from other locations may be reflected to the same pixel many times, which will cause an error. At the edge of a stereo object, namely, the foreground and background, the returned light is a mixture of the foreground and background, and noise will be generated during the process of distinguishing between the two. For the omni-directional obstacle detection requiring a real-time performance described in this paper, a fast adaptive median filter is used to remove noise.

The adaptive median filter first detects whether each pixel in the ROI space is a noise point. By providing a window with the initial size, the median value of the pixels in the neighborhood is calculated, and whether this value is between the maximum and minimum pixel value in the neighborhood is determined. If this condition is satisfied, then the point is not a noise point and will not be filtered by the median filter. Otherwise, the range of the neighborhood is enlarged, and this process is repeated. For those pixels that are consistently judged as noise points, the enlargement of the neighborhood will not stop until the neighborhood size reaches the setting threshold. Fig. 3 shows the denoising effect when using the adaptive median filter on a certain depth image. It can be seen that noises, such as holes in the depth image, have been effectively removed.

C. METHOD OF REGION GROWTH FOR DEPTH IMAGE

As described in the previous section, adaptive median filtering is used to remove the holes and other noises in the depth image, which makes the obstacles in the depth image easier to be extracted. Each pixel in the depth image has its own depth value, and thus, the entire depth image can be seen as a large number of 3D point clouds in space. Therefore, this study proposes a new region growth method that is suitable for the depth image, and preliminarily extracts obstacles in the depth image.

1. For each point in the depth image, $ID(x, y, z)$ is defined as the number of the region, where the point is located, the initial value of which is zero, and x, y, and z indicate

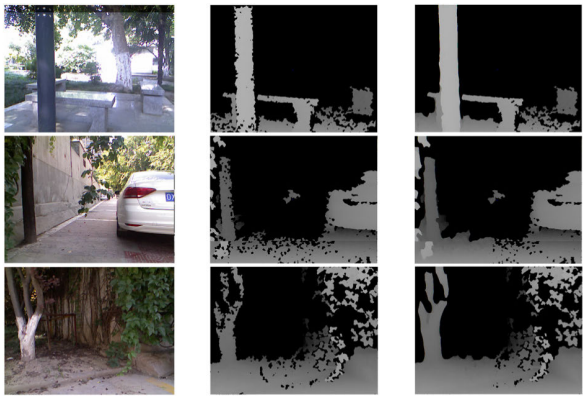


FIGURE 3. Denoising effect of adaptive median filter applied to a certain depth image.

the 3D coordinates of this point. In addition, $S(ID(x, y, z))$ is defined as its area, the initial value of which is also zero.

2. The similarity threshold is defined as T_s. Starting from the coordinates $(0, 0, 0)$ in the depth image, the image points are traversed along the three-dimensional coordinate axis in turn. When a depth point P_0 is encountered, it is confirmed whether there is a depth point P_1 in the radius T_s of P_0. If P_1 exists, then P_0 and P_1 are merged into the same region. The coordinate of P_0 is set as (x_0, y_0, z_0), the number of the region where P_0 is located is set as $ID(P_0) = ID(x_0, y_0, z_0) = 1$ and the area of P_0 is set as $S(ID(x_0, y_0, z_0)) = 1$. In addition, the coordinates of P_1 are set as (x_1, y_1, z_1) because P_1 and P_0 belong to the same region, and thus $ID(P_1) = ID(P_0)$ is set, that is $ID(x_1, y_1, z_1) = ID(x_0, y_0, z_0) = 1$ and the area of the region increases on its own, that is $S(ID(x_1, y_1, z_1)) = S(ID(x_0, y_0, z_0)) + 1 = 2$.

3. Continuing to traverse the depth image. If the next depth point is P_1, it is confirmed whether depth point P_1' exists within the radius T_s of P_1 and if so then P_1' and P_1 are merged into the same region. If the coordinate of P_1' is set as (x_1', y_1', z_1'), then $ID(x_1', y_1', z_1') = ID(x_1, y_1, z_1) = 1$ and the area self-increases, that is $S(ID(x_1', y_1', z_1')) = S(ID(x_1, y_1, z_1)) + 1 = 3$. If the next depth point is P_2, it is confirmed whether depth point P_2' exists within the radius T_s of P_2. If the next depth point is P_3, it is confirmed whether depth point P_3' exists within the radius T_s of P_3. The check continues until all depth points have been investigated.

4. At a certain time, if there are no depth points belonging to a certain region within the radius T_s of P', then P' is an isolated point or a start point of a new region, the number of regions self-increases, which means $ID(P')$ self-increases, and the area of the region is set as 1, that is $S(P') = 1$.

5. Steps 3 and 4 are repeated until all depth points are traversed.

Fig. 4 shows a process diagram of this method, in which a block represents an obstacle point. In (a), obstacle point p_0 is scanned sequentially because there are no points that belong to a region within the radius T_s of p_0, therefore p_0 is a starting point of a new region; thus, the number of regions that the point belongs to is $ID(p_0) = 1$ and the area of this point is $S(ID(p_0)) = 1$. Then, obstacle points P_1, P_2, and P_3 are searched sequentially. For p_1, because there are no points that belong to a region within the radius T_s of P_1, the number of the region self-increases, that is $ID(p_1) = ID(p_0) + 1 = 2$ and the area of this point is $S(ID(p_1)) = 1$ which means a new region is found. The red arrow indicates that the next obstacle point searched is P_2, and because P_1 exists within its radius, thus p_2 belongs to the region of p_1, that is $ID(p_2) = ID(p_1) = 2$ and the area self-increases as $S(ID(p_2)) = S(ID(p_1)) + 1 = 2$.

For p_3 which is sequentially searched, its number is the same as p_2 and the area self-increases. As shown in Fig. 4 (b), the process continues with the traversing of obstacle points along with the coordinate axis. Because p_4 exists within the radius of p_0, p_4 is merged into p_0, and the area self-increases by the largest area of the current region. When p_5 is searched, because p_4 exists within the radius of p_1p_5 is merged into p_1, and the area self-increases by the largest area of the current region. Fig. 4 (c) shows the growing result of all obstacle points, and that there are two obstacle regions generated. The number next to the obstacle region indicates the number of the obstacle region, and the number in the box indicates the search order of region growth. In addition, Fig. 4 (d) shows the area of the obstacles. As indicated, the area of both two obstacles increases gradually, indicating that the regions are growing.

This method is described as the following pseudo code, where the input is the depth image from the depth camera and the output is a set of obstacles.

Algorithm 1 Method of Region Growing for Depth Image

```
INPUT: points P from depth image, similarity threshold T_s
OUTPUT: obstacle segments Γ = {C_1, C_2, ..., C_n}
INITIALLY: ID(x, y, z) ← 0 to reset the number of the obstacle region
           S(ID(x, y, z)) ← 0 to reset the area of the obstacle region
Foreach p_i ∈ P do
    If p_i is a depth point and no depth point in the radius T_s of p_i Then
        ID(p_i) = ID(p_i) + 1
        S(ID(p_i)) = 1
        C ← C ∪ {p_i}
        If p_1 is a depth point and distance (p_1, p_i) < T_s Then
            ID(p_1) = ID(p_i)
            S(ID(p_1)) = S(ID(p_i)) + 1
            C ← C ∪ {p_1}
        End
    End
End
```

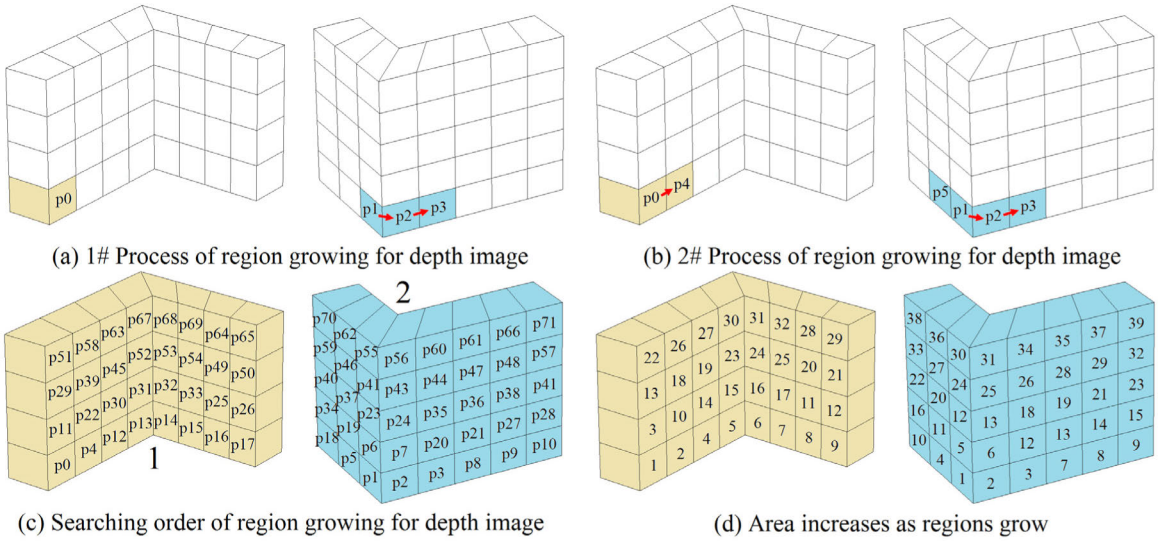


FIGURE 4. Process of region growing for depth image.

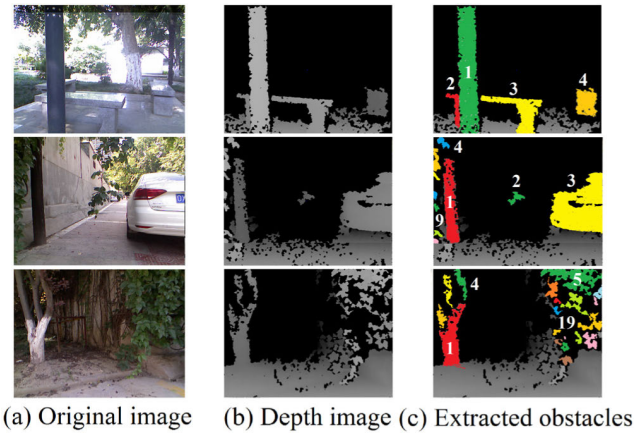


FIGURE 5. Obstacle extraction using region growth for depth image.

Each pixel point is traversed to check whether it is a depth point. If the pixel point is a depth point and there are no other depth points within the radius T_s of p_i, it is considered an isolated point or a start point of a new region, and the *ID* of this point then self-increases and its area is reset to 1. If the pixel point is a depth point and there does exist another depth point within the radius T_s of p_i, then these two depth points are merged, and the *ID* of each point is set to the same number, and the area of this region self-increases.

Fig. 5 shows the obstacle regions extracted from a certain depth image using the region growth method. Different obstacles are represented by different colors, and the number next to or in the obstacle represents the number of the region. As can be seen from Fig. 5, the region growth method can effectively extract obstacles in the depth image.

D. FAST INPAINTING METHOD FOR DEPTH IMAGE BASED ON KNN AND SPATIAL REGION GROWTH

Although the denoising and filtering of the depth image can eliminate some of the noise and holes, large holes still exist, which will affect the obstacle extraction in a later step. Generative Adversarial Networks (GAN) are State Of the Technical Art in image inpainting, however this method takes a long processing time and it is more important to quickly repair the holes using the obstacles information directly from the method of region growth in the previous step, which could accelerate the process of repairing the holes in depth image.

As described in the previous section, the obstacle region in a depth image is roughly extracted using spatial region growth. Because a real-time performance is required for vehicle obstacle detection, a fast inpainting method for the depth image based on KNN(k-NearestNeighbor) and spatial region growth is proposed herein.

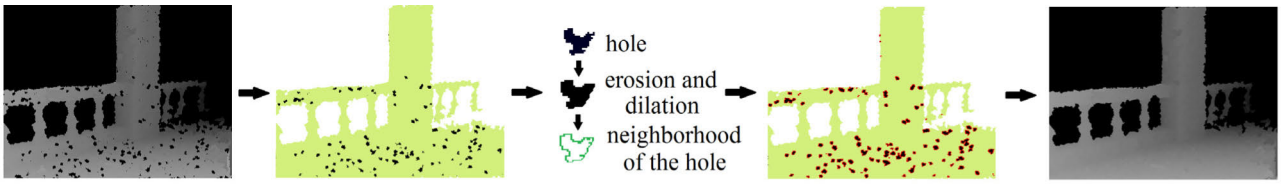
Because the infrared reflectivity of different objects differs and the occlusion between objects always occurs, there are numerous holes on the obstacle surface in the depth image. An inpainting of the depth image is applied to repair or fill in these holes. A spatial correlation is also analyzed according to the obstacle range obtained during the previous step, and the pixel that best matches the pixel in the hole is determined.

For the depth data, the depth value is defined as $D(u, v)$, and the point with a depth value $D(u, v)$ of zero is a hole. The depth image is then binarized, and the formula applied is as follows:

$$D(u,v) = \begin{cases} 0 & D(u,v) = 0 \\ 255 & D(u,v) > 0 \end{cases} \tag{3}$$

The result of binarization is shown in Fig. 6 (b). The holes $H_{()}$ in the figure are shown in black, and the obstacle range R_E extracted using the spatial region growth is indicated in green.

For each hole, the treatment of the morphological corrosion and expansion is first carried out to expand the range of the hole area, allowing the neighborhood of the holes to be included. The neighborhood is defined that it includes a hole as R_{ED} ($R_{ED} \in R_E$), the neighborhood range of the hole,



(a) Depth image with holes (b) Mask image with holes (c) Erosion and dilation of a single hole (d) Neighborhood of the holes in depth image (e) Depth image after inpainting

FIGURE 6. Process of inpainting of depth image.

which is defined as R_{NP}, can be obtained by determining the difference between the region after the treatment of the corrosion expansion and the original area through the following formula:

$$R_{NP} = R_{ED} - H_{()} \tag{4}$$

The neighborhood R_{NP} is shown in the red region in Fig. 6 (d).

To fill in a hole, it is necessary to determine its depth. Because the depth changes uniformly within a neighborhood, in this study, a new method for calculating the depth of the holes based on the use of KNN is proposed. To estimate the depth of a hole close to the neighborhood, it is given the same value as the depth the of neighborhood itself. For a pixel A in a hole of a depth image, its eigenvector is defined as $A_{ij}(X_i, X_j)$ and its estimated depth is D_A.

Pixel A is selected as the center point, and the related attribute values of $n^2 - 1$ pixels in its $n \times n$ dimensional neighborhood (where n is an integer of larger than 2) are calculated. The reference pixel is marked as B_r, its characteristic vector as $B_{rij}(X_{ri}, X_{rj})$, and its depth value as D_{Br}, where $r = 1, 2, \ldots, n^2 - 1$.

Because the depth value changes evenly within the hole region and the depth value is related to the distance of the neighborhood points of the hole, the ratio of Euclidean distance d_r is used to represent the correlation between the hole point and the neighborhood point, which is defined as the weight coefficient ω_r and the sum of the weights in $n \times n$ dimensional neighborhood is 1, that is,

$$\sum_{r=1}^{n^2-1} \omega_r = 1 \tag{5}$$

The weights of each hole point and its neighborhood point can be calculated as follows:

$$\omega_r = \frac{d_r}{\sum_{r=1}^{n^2-1} d_r} \tag{6}$$

where the Euclidean distance d_r is expressed as follows.

$$d_r = sqrt[(x_{ri} - x_i)^2 + (x_{rj} - x_{ij})^2] \tag{7}$$

Here, x_{ri} is the ith dimensional coordinate of the rth point, and x_{rj} is the jth dimensional coordinate of the rth point. The smaller the Euclidean distance d_r is between the two points, the larger the weight coefficient and the higher the similarity.

The depth value D_A of the pixel in a hole is calculated using the valid depth information of the neighboring points, which is expressed as follows:

$$D_A = \sum_{r=1}^{n^2-1} (\omega_r D_{Br}) \tag{8}$$

That is,

$$D_A = \sum_{r=1}^{n^2-1} \left[\frac{D_{Br} \cdot sqrt[(x_{ri} - x_i)^2 + (x_{rj} - x_{ij})^2]}{\sum_{r=1}^{n^2-1} sqrt[(x_{ri} - x_i)^2 + (x_{rj} - x_{ij})^2]} \right] \tag{9}$$

The calculated depth value D_A of the hole pixel is filled into the corresponding position of the hole to complete the inpainting of the depth image. The final inpainting result of a depth image with holes (Fig. 6 (a)) is shown in Fig. 6 (e), which indicates that the holes are well repaired.

E. CLUSTERING SEGMENTATION USING ITERATIVE NORMALIZED CUT

In the previous section, different obstacles are obtained using the region growth method for the depth image. However, the method is easily affected by the similarity threshold T_s and may result in the fragmentation of obstacles if this threshold is too small, which is not conducive to the overall extraction of obstacles. Herein, an improved iterative normalized cut is proposed to create the normalized segmentation and merging the fragmented obstacles.

For obstacles obtained using region growth for the depth image, an undirected graph with weights is constructed using the iterative normalized cut algorithm for these obstacles. Each node in the graph represents an obstacle, and the weights on the edges represent the approximate relationship between obstacles.

By establishing a normalized minimum cut on the established graph, the obstacles in the graph are segmented for the first time, and this segmentation is carried out iteratively until no new areas can be segmented. Finally, the clustering segmentation of all obstacles is completed.

An undirected graph $G = (V, E)$ is constructed for obstacles obtained by region growth for the depth image, where V represents the obstacles extracted from the previous step, and E represents edges, which are the connections between obstacles. Assume that graph G can be divided into disjointed parts A and B ($V = A \cup B$). One partition of graph G can be defined as follows:

$$C_w(A, B) = \sum_{i \in A, j \in B} w(i, j) \tag{10}$$

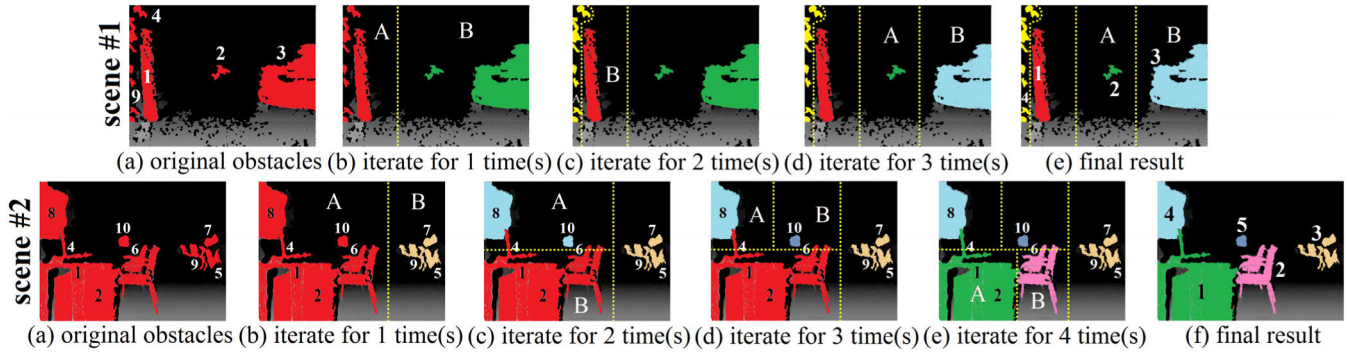


FIGURE 7. Process of iterative normalized cut.

where $C_w(A, B)$ denotes the weights between regions A and B, and $w(i, j)$ denotes the weights between obstacles i and j. The weights are defined as the spatial distance between obstacles. Iterative normalized cut achieves the purpose of normalized segmentation by calculating the normalized weights. The calculation formula is as follows:

$$NC_w(A, B) = \frac{C_w(A, B)}{\sum_{i\in A, j\in V} w(i, j)} + \frac{C_w(A, B)}{\sum_{i\in B, j\in V} w(i, j)} \tag{11}$$

Here, $\sum_{i\in A, j\in V} w(i, j)$ and $\sum_{i\in B, j\in V} w(i, j)$ represent the sum of the weights from A and B to all nodes in the graph, respectively. Therefore, the optimal segmentation of A and B can be achieved by finding the minimal value of $NC_w(A, B)$. The minimal value of $NC_w(A, B)$ can be solved by solving the eigenvalues and eigenvectors of the matrix. Let the number of obstacles be n, let $x = (x_1, x_2, \cdots, x_n)$, where $x_i = -1$ denotes that B contains node i, and $x_i = 1$ denotes that A contains node i. Let W be a symmetric matrix of $n \times n$, whose element w_{ij} is $W(i, j)$ in (10). Let $D(i, j) = d_i$ be the diagonal matrix, where $d_i = \sum_j w(i, j)$. Let $k = \frac{\sum_{x_i>0} d_i}{\sum_i d_i}$, and $NC_w(A, B)$ can thus be deduced as follows:

$$NC_w(A, B) = \frac{(1 + x)^T (D - W)(1 + x)}{4k1^T D1} + \frac{(1 - x)^T (D - W)(1 - x)}{4(1 - k)1^T D1} \tag{12}$$

where 1 denotes a matrix of dimensions $[N, 1]$ with each of its elements having a value of 1. Let $b = k/(1 - k)$ and $y = [(1 + x) - b(1 - x)]/2$, then the minimal value of $NC_w(A, B)$ can be obtained by solving the eigenvalues and eigenvectors of formula (13).

$$(D - W)y = \lambda Dy \tag{13}$$

where y is the eigenvector and t is the eigenvalue. Then, the location of the obstacle to be segmented is the eigenvector corresponding to the second smallest eigenvalue.

Initially, the iterative normalized cut algorithm is used to cluster and segment all obstacles obtained through the region growth method for the depth image, and two segmentations A and B of the obstacles are obtained. Iterative normalized cut is then applied again for both sections A and B, and the separated obstacle continues to be iteratively clustered and segmented by the algorithm until it can no longer be segmented.

Fig. 7 shows the process of the iterative clustering and segmentation of obstacles in two scenes using the iterative normalized cut algorithm after extraction using the region growth method for the depth image. Fig. 7 (a) shows the obstacles extracted, in which a total of ten obstacles are found in scene #2 and nine are found in scene #1. Fig. 7 (b) shows the results of clustering and segmentation applied for the first time, in which all obstacles are divided into sections A and B; the objects in both sections are, then clustered and segmented into new sections A and B, respectively.

For scene #1, Fig. 7 (c) shows the clustering and segmentation results of section A, and Fig. 7 (d) shows the results for section B. In Fig. 7 (c), section A is segmented into new sections A and B, and in Fig. 7 (d), section B is also segmented into new sections A and B. The final clustering and segmentation results are shown in Fig. 7 (e). The results indicate that the nine obstacles are clustered into one to four obstacles, which effectively reduces their fragmentation.

In scene #2, because there are no obstacles that can be segmented in section B, section A is clustered and segmented, the results of which are shown in Fig. 7 (c). In Fig. 7 (c), section A is segmented into new sections A and B. Iterative segmentation for part A is shown in Fig. 7 (d), and that for part B is shown in Fig. 7 (e). The final clustering and segmentation results are shown in Fig. 7 (f). As the results indicate, the ten obstacles are clustered into one to five obstacles, which also effectively reduces their fragmentation.

F. DEPTH IMAGE REGISTRATION AND VISUALIZATION OF OMNI-DIRECTIONAL OBSTACLES

As the exact position(including position and rotating angle) of the cameras in relation to the car coordinate system is known, the transformation matrix $[T]$ of each depth camera can be obtained, and the inverse transformation matrix $[T]'$ could also be deducted from this matrix.

When detecting obstacles, all the depth cameras generate depth images at the same time. After all the cameras have

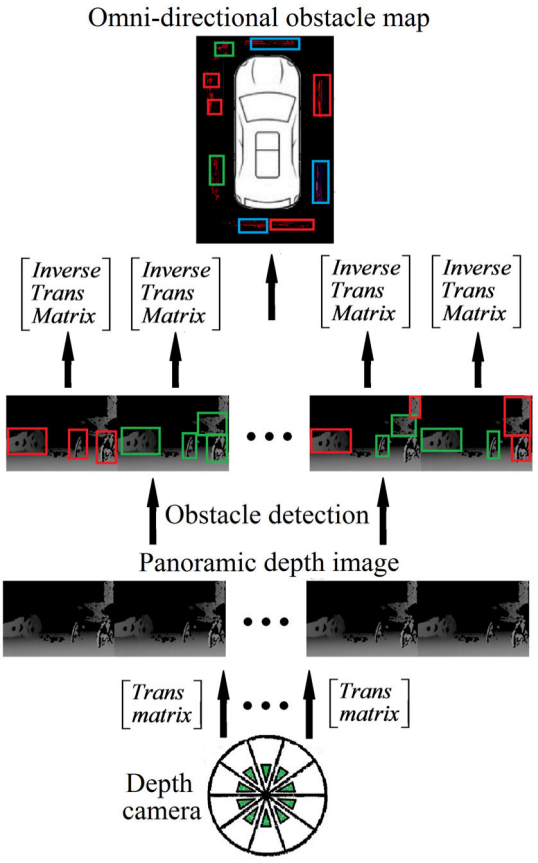


FIGURE 8. Generation of the panoramic image and the omni-directional obstacle map.

generated the depth images, depth points in ten depth images are transformed into the car coordinate system using their own transformation matrices $[T]$, which completes the depth point cloud registration in order to generate the panoramic depth image.

After the obstacles in the panoramic depth image are extracted using our method mentioned above, the obstacles in the panoramic depth image are transformed according to the inverse transformation matrix $[T]'$ deducted from transformation matrix $[T]$, in order to generate a omni-directional obstacle map. These two processes as shown in Fig.8.

3D visualization technology is used to show an omni-directional scene by overviewing the obstacles. By rendering each depth point and constructing obstacle point clouds, the purpose of a real-time obstacle overview can be achieved.

IV. EXPERIMENT AND ANALYSIS

A. OMNI-DIRECTIONAL OBSTACLE EXTRACTION

In this experiment, to verify the effectiveness of the proposed method, the effect of omni-directional obstacle detection under different scenarios is tested. Depth cameras are installed at different locations of the vehicle, as shown in Fig. 1 (a). The figure also shows that ten depth cameras are installed around the car body to collect depth information of the omni-directional environment. An on-board computer is used for the experiment and is equipped with an Intel i3-4130p processor with a main frequency of 3.4 GHz and 4G memory, Intel realsense D430 module is used as the depth camera.

The effect of omni-directional obstacle detection is tested under both daytime and nighttime environments, and the results are compared with current mainstream obstacle detection methods. Fig.9 shows the results of omni-directional obstacle detection using our method during the daytime. The depth image of the environment is obtained using depth cameras installed in ten locations, and the obstacles are then detected using our proposed method. The detected obstacles are represented by region using a different color in the original and depth images. Among these regions, red indicates the primary obstacle region, where the obstacle is nearest to the vehicle. As indicated, our method can extract obstacles more completely, and small and fragmentary obstacles (such as tree crowns and branches) can also be detected more completely. Obstacles are segmented coherently and reasonably as a whole. No detection errors in which one obstacle is segmented into different regions occur, or different regions of different obstacles are merged into new obstacles. A top view of the obstacles represents an overview of the extracted obstacle point cloud through a 3D visualization. The term ''Omni'' indicates the overview of all obstacle point clouds showing the obstacles that exist in an omni-directional environment.

Fig. 10 shows the results of omni-directional obstacle detection using our method at night. Because the environment is purely dark without any interference of other lighting, the depth camera can work well. Our method is also used to detect obstacles in this type of environment, and detected obstacles are represented by regions with different colors in both the original and depth images. Among these regions, the red region is also the primary obstacle region, representing the obstacle that is nearest to the vehicle. The experiment results show that the obstacles detected by our method at night are clearer and more complete than those detected during the daytime, and the obstacles are also more reasonably segmented without any abnormalities. The top view image of the obstacles using 3D visualization shows the distribution of omni-directional obstacles in a dark environment.

At present, vision-based obstacle detection focuses on obstacle extraction using binocular vision. The binocular vision method usually uses the binary threshold segmentation, clustering analysis, UV-disparity map, and the optical flow to extract obstacles.

Binary threshold segmentation simply sets a distance threshold to segment the pixels in the binocular depth image according to the distance threshold, such as retaining the pixels within the distance threshold and removing the pixels outside the distance threshold. The clustering analysis method combines the pixels with the same attributes into the same region through a certain statistical method of pixel distance and density to achieve obstacle extraction. UV-disparity map

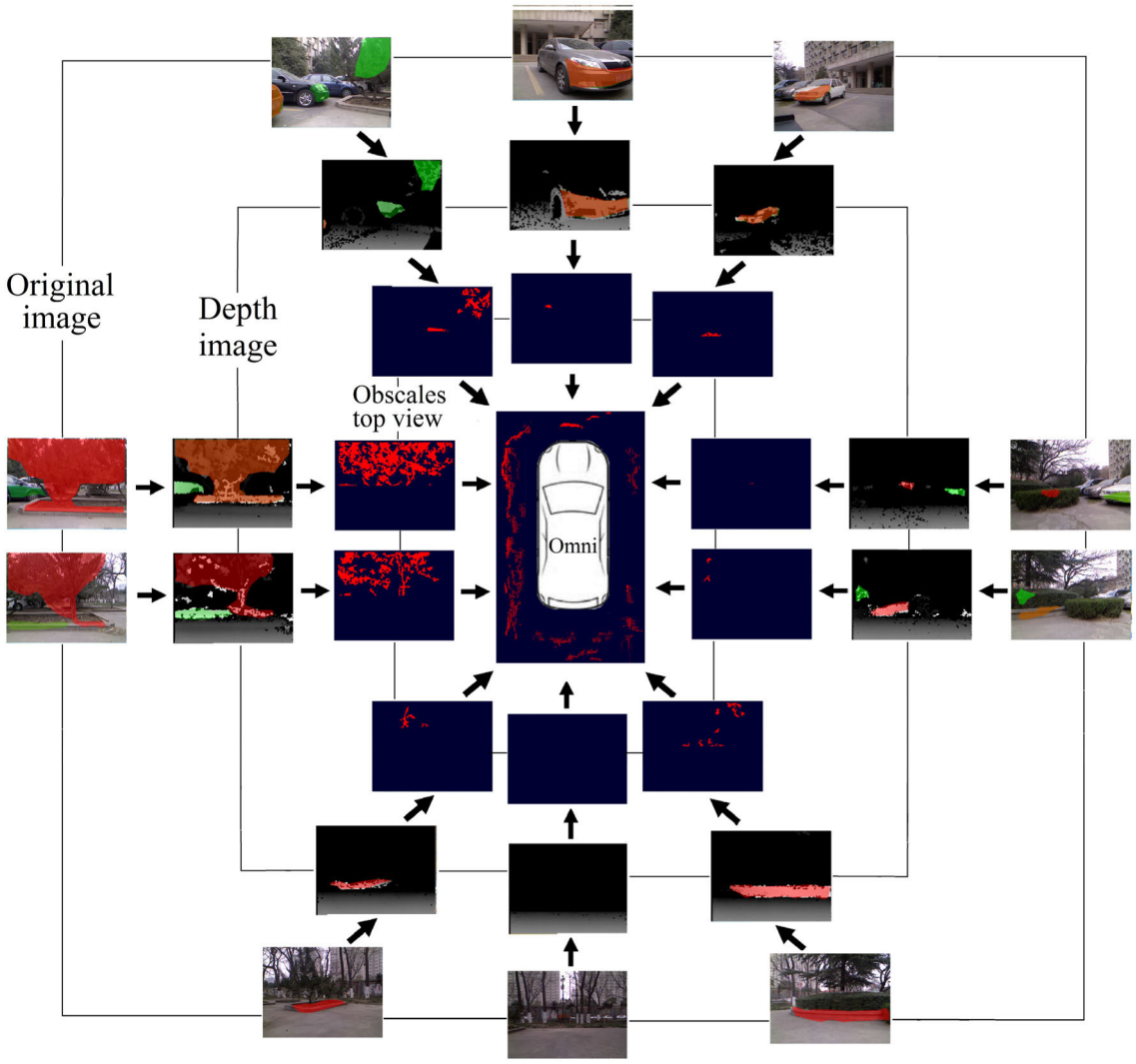


FIGURE 9. Omni-directional obstacle detection using our method during the daytime.

determines the position of obstacles by counting the number of disparity on the u-map and v-map respectively and detecting the straight lines on the u-map and v-map.The optical flow method assigns a velocity vector to each pixel in the image, thus forming a motion vector field. The velocity vector formed by moving object is different from that of background, so the position of moving object can be calculated.

The detection results using the methods above are compared with those of our method. Because the vision-based method cannot work at night, the experiment was only carried out during the daytime. Fig. 11 (a) shows the experiment results of obstacle extraction using binary threshold segmentation. The experimental scene applied is the same as that shown in Fig. 9. The segmentation threshold is set as $T = 1.0$, which indicates that an obstacle can be considered to exist when the distance between the points in the binocular depth image and the camera (i.e., the vehicle) is less than 1 m. We can see from the segmentation result that all of the segmented obstacle are the depth points closest to the vehicle. However, because only the threshold is used for segmentation, the segmented obstacles are incomplete and fragmented, which makes it impossible to analyze and achieve obstacle avoidance in a further step. Fig. 11 (b) shows the experiment results of the clustering segmentation. We use the K-means clustering algorithm to segment and extract obstacles from the binocular depth images. After the extraction of the obstacles, the purpose of extracting obstacles closer to the vehicles is achieved by retaining obstacles with smaller depth values. However, the K-means algorithm is limited by the value of K, and this clustering segmentation algorithm is more inclined toward the overall segmentation based on the spatial location; however, the obstacle segmentation of the depth image should be more inclined toward segmentation based on the connected region. Because the segmentation is based on the spatial location, we can see from Fig. 11 (b) that obstacles such as the tree crown are not extracted, and the trunk and other obstacles are clustered into an entire obstacle. Because the spatial location of the tree crown is far from the obstacles on the lower

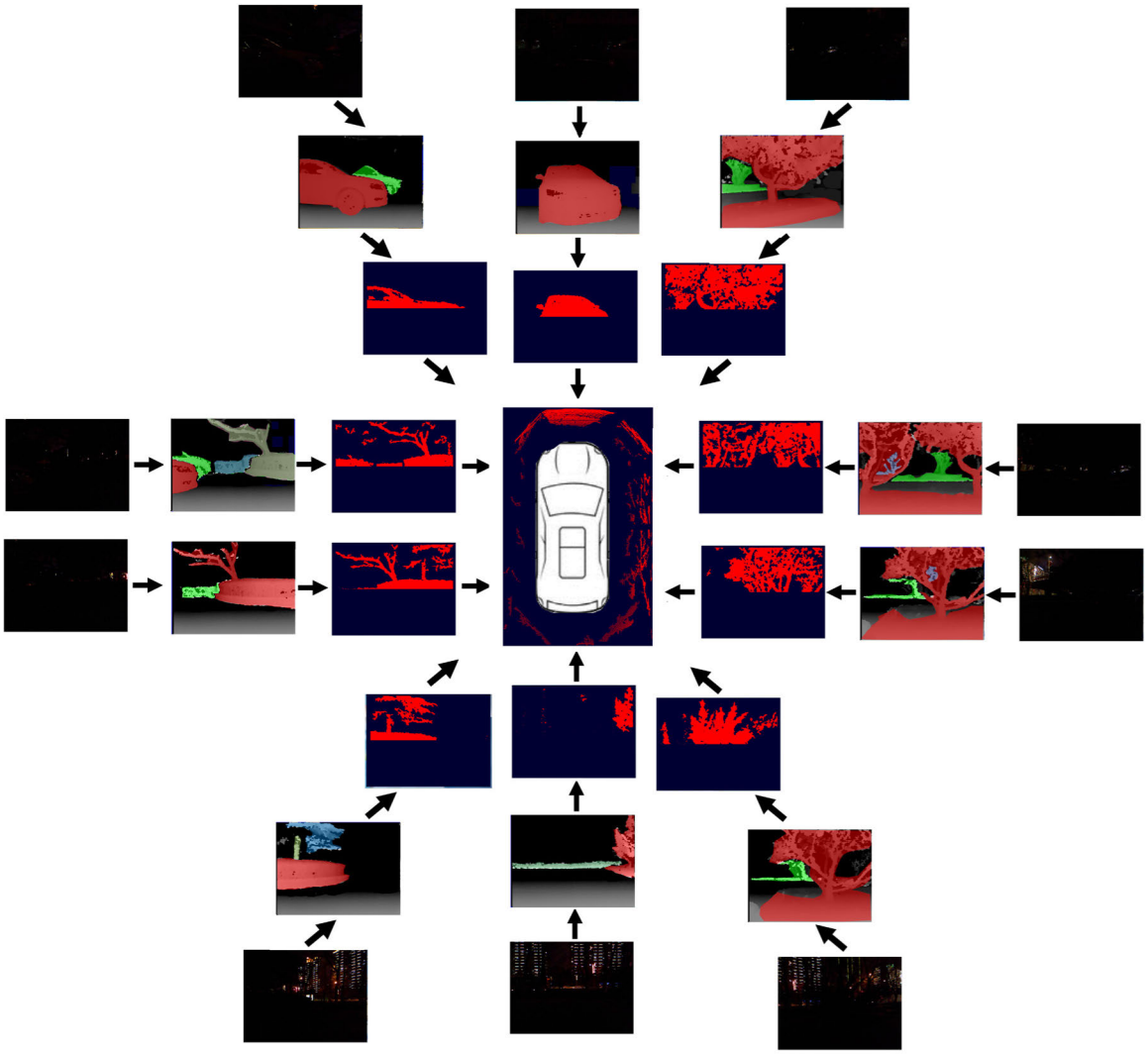


FIGURE 10. Omni-directional obstacle detection by our method at night.

side, the K-means algorithm incorrectly divides the trunk and other obstacles on the lower side into the same obstacles. Fig. 11 (c) shows the result of extracting obstacles in the depth image using UV-disparity maps. The UV-disparity map determines the range of obstacles by detecting the lines in the map. However, because numerous fragmentary obstacles (such as trunks and leaves) exist in the test scene, it will be quite difficult to detect lines in the UV disparity map, and the detection error will be quite large. Because the UV-disparity map determines the obstacle range by detecting the line in the U and V maps, respectively, it can better extract regular obstacles. For complex and irregular obstacles such as leaves, trunks, and grasses in this experiment, the UV-disparity map can only roughly determine their locations and cannot accurately segment different obstacles; therefore, the false recognition rate is very high. In Fig. 11 (c), we detect lines that clearly exist. From the results, we can see that the overall range of the obstacles can be determined, although the obstacles cannot be accurately segmented, and the spatial overlapping obstacles cannot be distinguished. Therefore, the UV-disparity map is not suitable for obstacle detection in complex scenes. Fig. 11 (d) shows the obstacle detection result using the optical flow method. In addition, the yellow point in the image is the direction of the optical flow. Because the optical flow method first needs to extract image feature points, different lighting and color will affect the extraction of the feature points, and therefore corner and feature points with significant changes can be properly extracted, and the obstacles that belong to these points can also be well extracted. Therefore, we can see from the results in Fig. 11 (d) that parts of the obstacles are extracted, although the obstacles without distinct feature points such as trees and grasses are not well detected. Fig. 11 (e) shows the obstacle detection result using lidar and deep learning. In the training phase, as it is impossible to recognize all of the obstacles in the surrounding environment due to limited samples, only cars and trees with different shapes were trained for this experiment. However, as the samples trained are very limited,

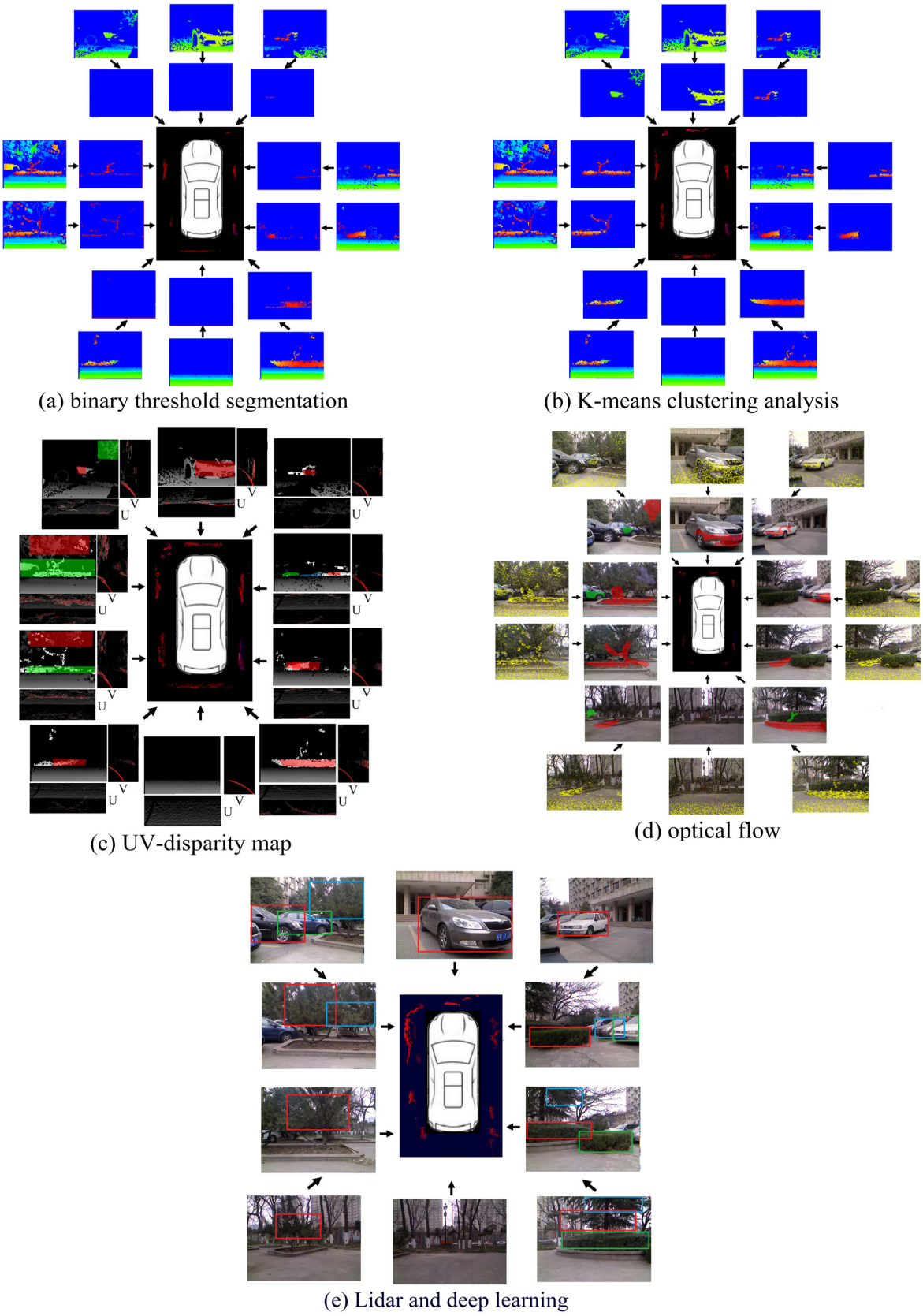


FIGURE 11. Omni-directional obstacle detection using different methods.

TABLE 2. Comparison of efficiency of omni-directional obstacle detection using different methods.

	Total	Maximum time	Minimum time	Average time
Binary threshold segmentation	<9 s	<1 ms	<1 ms	<1 ms
K-means clustering analysis	9945.714 s	1336 ms	874 ms	1105 ms
UV-disparity map	6021.253 s	701 ms	637 ms	669 ms
Optical flow	12421.5 s	1493 ms	843 ms	1168 ms
Deep learning	2205.4s	387ms	103 ms	245 ms
Our method	126.017 s	23 ms	5 ms	14 ms

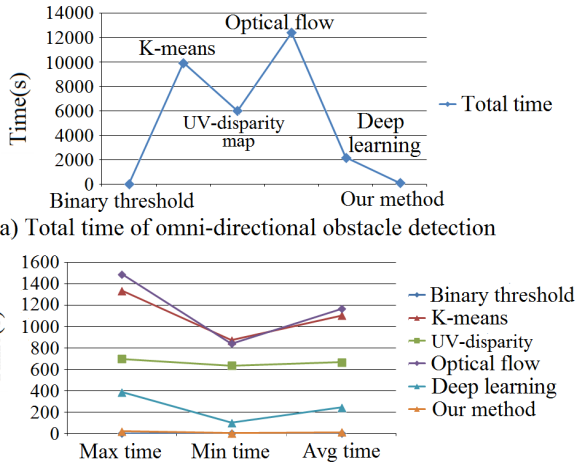
this method cannot detect all the obstacles and small obstacles are not completely detected.

B. COMPARISON OF EFFICIENCY AND ACCURACY OF OMNI-DIRECTIONAL OBSTACLE DETECTION

For omni-directional obstacle detection for vehicles, the detection efficiency will affect the driving safety. Therefore, we compare the efficiency of omni-directional obstacle detection with our method and the methods tested above. In this experiment, the frame rates of the depth camera and the RGB camera were both 30 fps. The vehicle drove for 5 minutes in an environment with complex obstacles and recorded 9,000 frames of depth and binocular images. The obstacles in each frame were detected using our proposed method and the binary threshold segmentation, K-means clustering analysis, UV-disparity map, and optical flow methods, respectively, and the total time required to complete the detection of 9,000 frames, and the maximum, minimum, and average times required to detect a single frame, were calculated.

Table 2 shows the efficiency of the different methods. Fig. 12 shows the graph of the time required for omni-directional obstacle detection for all frames and a single frame based on the table. Because K-means is a process of iterative convergence, it takes a long time to complete the detection. The UV-disparity map method needs to scan every pixel in the U map that corresponds to the column of the original depth map and analyze the number of statistical disparity values. Each pixel in the V map that corresponds to the row of the original depth map also needs to be scanned and the number of disparity values are then analyzed; therefore, the UV-disparity map method is also time-consuming. Binary threshold segmentation is in fact not an obstacle detection method. A partial depth image is preserved using a predefined threshold, and it is therefore meaningless to calculate the required time. The optical flow method takes a long time to process each frame because of its complexity and numerous computations required. The deep learning method also takes a long time because of its complexity. In contrast, our method has a faster processing speed and is suitable for real-time omni-directional obstacle detection.

For the accuracy of omni-directional obstacle detection, we collected a large number of omni-directional depth images



(a) Total time of omni-directional obstacle detection

(b) Time of omni-directional obstacle detection in a single frame

FIGURE 12. Time of omni-directional obstacle detection.

and their corresponding binocular images in different environments with different obstacles, and detected the obstacles in these images using both our method and other methods mentioned above, and manually marked the locations of the major obstacles in each environment to establish a dataset. We used the Intersection Over Union (IOU) as the evaluation metrics for this dataset. When marking the locations of the major obstacles in each environment, the ground-truth boxes that cover the obstacles were also marked. The established data set consists of 5000 depth images and their corresponding binocular images(2500 depth images and 2500 corresponding binocular images), and these images include 9 scenes and 257 obstacles. This data set as shown in Table 3.

TABLE 3. Established data set.

Environment number	Total number of obstacles	Environment number	Total number of obstacles	Environment number	Total number of obstacles
Scene 1#	24	Scene 4#	43	Scene 7#	19
Scene 2#	27	Scene 5#	17	Scene 8#	36
Scene 3#	35	Scene 6#	32	Scene 9#	24
Total			257		

We considered correctly detection(true positives), wrong detection(false positives) and undetected(false negatives) to calculate accuracy, precision and recall. When each obstacle was detected, we also use a box that can cover the obstacle, and then the area of the overlap of this box and the ground-truth box was calculated to compare the detection similarity. We define the ratio of this area to the area of the ground-truth box as the detection similarity, and we calculated the average similarity after all the images in data set were detected.

The experiment results are shown in Table 4, and Fig. 13 shows a graph of the accuracy, precision, recall rates and average similarity based on the table.

In Table 4, ‘‘Correct detection’’ indicates the number of the images that all of the obstacles in each image are correctly

TABLE 4. Comparison of accuracy of omni-directional obstacle detection using different methods.

	Correct detection	Accuracy rate	Precision rate	Recall rate	Average similarity
Binary threshold segmentation	1053	21%	84%	21%	49%
K-means clustering analysis	3761	75%	85%	81%	79%
UV-disparity map	1627	32%	67%	37%	62%
Optical flow	3406	68%	88%	68%	86%
Deep learning	2742	54%	96%	55%	89%
Our method	4842	96%	97%	98%	93%

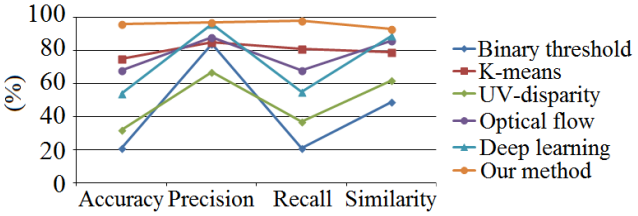


FIGURE 13. Omni-directional obstacle detection, precision and recall rate with different methods.

detected. Because binary threshold segmentation only preserves a portion of the disparity region, it does not have the ability to extract obstacles as a whole, and therefore this method is basically unable to achieve a proper obstacle detection. As this method segments the depth image though a set threshold, only the obstacle regions in the threshold can be detected, which causes most of the obstacles will not be detected and thus the accuracy and recall rate are very low. As this method can not really segment the obstacles, therefore most of the detected obstacles have incomplete shapes, which cause the detection similarity to the ground-truth box are also very low. The K-means clustering analysis can extract obstacles as a whole but is more inclined toward an overall segmentation according to the spatial location. However, the obstacle segmentation of the depth image is more inclined toward determining the connected regions of the obstacles. Therefore, K-means results in numerous obstacles being missegmented into different obstacles. Although obstacles are divided into different parts, most obstacles that marked in the data set were detected, and only a few wrong or incomplete detections exist. Therefore this method has a relatively high precision and recall rate. By contrast, the UV-disparity map detects obstacles by detecting lines in a UV graph. If there are numerous fragmentary and complex obstacles in the environment, the detection of the lines will be extremely difficult to achieve, resulting in a number of incorrectly detected obstacles, which lead to poor accuracy, precision and recall rate. Because UV method detects the lines to judge the obstacle, therefore detected obstacles usually have incomplete shapes, which will cause the area of the detected obstacles largely differ from the area of the ground-true boxes, and thus the detection similarity is also very low. Because the environmental light and color of the obstacle affect the extraction of the corner and feature points, the optical flow method cannot reliably extract all obstacles, and a few of the obstacles will not be completely extracted, which cause the accuracy, recall rate and average similarity are not very high. But for the obstacles that have the strong feature points, this method could well detect obstacles, so the precision rate can reach a high level. For deep learning, because only the sample of vehicle and tree were trained for the experiment, therefore not all the obstacles are correctly detected. But for the trained samples, almost all the obstacles are correctly and completely detected, therefore it has a high precision rate and high similarity to the ground-truth box. Because there are many other obstacles are not detected in the experiment environment due to limited samples, the accuracy and recall rate are relatively low. By contrast, first as our method could effectively extract small and irregular obstacles in complex scenes using depth image region growth, the number of detected obstacles are largely increased, which avoid small obstacles being missing and the shape of detected obstacles being incomplete, and thus our method could improve the accuracy and recall rate. Second, as a large number of fragmentary obstacles can be merged into the nearest obstacles through the proposed iterative normalized cut algorithm, it ensures that the detected obstacles could have a more complete shape and thus improves the precision rate and average similarity to the ground-truth box. All of these advantages make our method could more accurately and completely detect obstacles in all kind of complex environment, and more conducive to the subsequent overall treatment of the obstacles for analysis and avoidance.

V. DISCUSSION AND CONCLUSION

To solve the problem in which the obstacle detection around a vehicle is easily affected by the environment light and the material of the obstacle, this study proposes using a depth camera to obtain the depth image of the environment and extract obstacles from this image. A depth camera has a large detection coverage and can work at day or night, and is therefore more suitable for omni-directional obstacle detection than ultrasound and visual sensors.

In this study, the application of the ROI for a depth point cloud was proposed, which avoids collisions with obstacles by defining a plane parallel to the chassis, and the processing efficiency of the depth point cloud was promoted. The transformation matrix of each depth camera is used to generate the panoramic depth image. A fast inpainting method for depth image was also proposed, which effectively eliminates the noise and holes on the obstacle surface in the depth image. A region growth method for the depth image

was also proposed. By analyzing the connected region in a depth image, the depth image is segmented, and obstacles are roughly extracted. An improved iterative normalized cut method was also proposed to cluster and segment all obstacle regions extracted using the region growth method for the depth image. The fragmented and small irregular obstacles are merged with the nearest obstacles in the spatial position to generate more complete obstacles, which is conducive to the subsequent overall treatment of the obstacles for analysis and avoidance. Finally, the inverse transformation matrix deducted from transformation matrix of their original depth camera is applied to generate a omni-directional distribution map for obstacles around the vehicle, thereby increasing driving safety. Based on an obstacle detection experiment conducted in a complex environment during both day and night, the effect of our detection method during the daytime was shown to be more effective than that of other mainstream methods in extracting complete obstacles, whereas for fragmentary and small obstacles such as branches, crowns, and leaves, our method can also achieve a complete extraction. Because the depth camera can work better in a dark environment, the detected obstacles have a more complete shape and clearer segmentation level at night, whereas the other mainstream methods cannot work at all at night.

Using a depth camera and a binocular camera to collect the depth and binocular images of the environment with complex obstacles, and applying different methods to detect obstacles in each image and calculate both the time required and the detection rate, the results indicate that our method has a faster detection speed than that of the other methods, as well as higher detection, precision, and recall rates. Each of these advantages shows that our method is more suitable for real-time omni-directional obstacle detection for use in vehicles.

ACKNOWLEDGMENT

Authors X. Zhao and H. Wu conceived the study and experiments, drafted the manuscript and guided experiments, Z. Xu carried out the experiments and edited the manuscript. The other co-authors contributed analysis, discussion, manuscript editing, and help with performing the experiments.

REFERENCES

[1] L. C. León and R. Hirata, "Car detection in sequences of images of urban environments using mixture of deformable part models," *Pattern Recognit. Lett.*, vol. 39, pp. 39–51, Apr. 2014.

[2] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard, "Motion-based detection and tracking in 3D LiDAR scans," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 4508–4513.

[3] Q. Li, B. Dai, and H. Fu, "LIDAR-based dynamic environment modeling and tracking using particles based occupancy grid," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, Aug. 2016, pp. 238–243.

[4] M. Himmelsbach, F. V. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3D point clouds for ground vehicles," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 560–565.

[5] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2147–2156.

[6] H. Zhu, M. Fu, Y. Yang, X. Wang, and M. Wang, "A path planning algorithm based on fusing lane and obstacle map," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1442–1448.

[7] J. Žbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, pp. 1–32, Apr. 2016.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[9] Y. Liu, Z. Wang, Y. Liu, Z. Zhang, and H. Xu, "Reversing obstacle detection based on binocular vision image," *J. Chongqing Jiaotong Univ. Natural Sci.*, vol. 37, no. 3, pp. 92–98, Mar. 2018.

[10] S. Q. Marlow and J. W. Langelaan, "Local terrain mapping for obstacle avoidance using monocular vision," *J. Amer. Helicopter Soc.*, vol. 56, no. 2, p. 22007, Apr. 2011.

[11] Y. Gao, X. Ai, Y. Wang, J. Rarity, and N. Dahnoun, "U-V-disparity based obstacle detection with 3D camera and steerable filter," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 957–962.

[12] B. Musleh, A. Escalera, and J. M. Armingol, "U-V disparity analysis in urban environments," in *Proc. Int. Conf. Comput. Aided Syst. Theory*, Feb. 2011, pp. 426–432.

[13] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1592–1599.

[14] C. C. Pham, V. Q. Dinh, and J. W. Jeon, "Robust non-local stereo matching for outdoor driving images using segment-simple-tree," *Signal Process., Image Commun.*, vol. 39, pp. 173–184, Nov. 2015.

[15] A. Teichman, J. Levinson, and S. Thrun, "Towards 3D object recognition via classification of arbitrary object tracks," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 4034–4041.

[16] D. Held, J. Levinson, and S. Thrun, "A probabilistic framework for car detection in images using context and scale," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 1628–1634.

[17] Y. Yi, Y. Guang, Z. Hao, F. Meng-Yin, and W. Mei-Ling, "Moving object detection under dynamic background in 3D range data," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 394–399.

[18] J. Hariyono, V.-D. Hoang, and K.-H. Jo, "Moving object localization using optical flow for pedestrian detection from a moving vehicle," *Scientific World J.*, vol. 2014, Jul. 2014, Art. no. 196415.

[19] C. D. Pantilie and S. Nedevschi, "Real-time obstacle detection in complex scenarios using dense stereo vision and optical flow," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 439–444.

[20] P. Lenz, J. Ziegler, A. Geiger, and M. Roser, "Sparse scene flow segmentation for moving object detection in urban environments," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 926–932.

[21] M. Muffert, T. Milbich, D. Pfeiffer, and U. Franke, "May i enter the roundabout? A time-to-contact computation based on stereo-vision," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 565–570.

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[23] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[24] B. Pepikj, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3286–3293.

[25] Y. Wei, J. Yang, C. Gong, S. Chen, and J. Qian, "Obstacle detection by fusing point clouds and monocular image," *Neural Process. Lett.*, vol. 49, no. 3, pp. 1007–1019, Jun. 2019.

[26] C. Häne, L. Heng, G. H. Lee, F. Fraundorfer, P. Furgale, T. Sattler, and M. Pollefeys, "3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection," *Image Vis. Comput.*, vol. 68, pp. 14–27, Dec. 2017.

[27] T.-J. Lee, D.-H. Yi, and D.-I. Cho, "A monocular vision sensor-based obstacle detection algorithm for autonomous robots," *Sensors*, vol. 16, no. 3, p. 311, 2016.

[28] D. Xie, Y. Xu, and R. Wang, "Obstacle detection and tracking method for autonomous vehicle based on three-dimensional LiDAR," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, Mar. 2019, Art. no. 172988141983158.

[29] L. Li, Q. Zhang, G. Qi, K. Zhang, M. Li, D. Yu, and J. Liu, "Algorithm of obstacle avoidance for autonomous surface vehicles based on LIDAR detection," in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Mar. 2019, pp. 1823–1830.

[30] G. Qi, H. Wang, M. Haner, C. Weng, S. Chen, and Z. Zhu, "Convolutional neural network based detection and judgement of environmental obstacle in vehicle operation," *CAAI Trans. Intell. Technol.*, vol. 4, no. 2, pp. 80–91, Jun. 2019.
[31] B. Bovcon, R. Mandeljc, J. Perš, and M. Kristan, "Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation," *Robot. Auto. Syst.*, vol. 104, pp. 1–13, Jun. 2018.

XIANGMO ZHAO received the B.S. degree from Chongqing University, China, in 1987, and the M.S. and Ph.D. from Chang'an University, China, in 2002 and 2005, respectively. He is currently a Professor and the Vice President of Chang'an University, China. He has authored or co-authored over 130 publications and received many technical awards for his contribution to the research and development of intelligent transportation systems. His research interests include intelligent transportation systems, distributed computer networks, wireless communications and signal processing.

HUAYUE WU received the B.S. degree in Computer Science and Technology from Xidian University, Xi'an, in 2013 and the M.S. degree in Computer Technology from Chang'an University in 2015. He is currently a Ph.D. student in the School of Information Engineering, Chang'an University, Xi'an, China. His research interests include intelligent transportation systems, and connected and automated vehicle.

ZHIGANG XU received the B.S. degree in automation, and the M.S. and Ph.D. degrees in traffic information engineering and control from Chang'an University, China, in 2002, 2005, and 2012, respectively. He is currently an Associate Professor at Chang'an University, China. His research focuses on connected and automated vehicle, Intelligent Transportation Systems, and nondestructive testing of infrastructures.

HAIGEN MIN received the B.S. and M.S. degrees in the Department of computer science and is currently pursuing the Ph.D. degree in the Department of Traffic Information Engineering and Control from Chang'an University, China. His research interests include localization and navigation system for intelligent vehicle and test methodology for intelligent and connected vehicle.