

Research Article

Construction of Fisheye Lens Inverse Perspective Mapping Model and Its Applications of Obstacle Detection

Chin-Teng Lin,¹ Tzu-Kuei Shen,¹ and Yu-Wen Shou²

¹Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan

²Department of Computer and Communication Engineering, China University of Technology, Hsinchu 303, Taiwan

Correspondence should be addressed to Yu-Wen Shou, owen@cute.edu.tw

Received 1 December 2009; Revised 15 April 2010; Accepted 15 June 2010

Academic Editor: Yingzi Du

Copyright © 2010 Chin-Teng Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we develop a vision based obstacle detection system by utilizing our proposed fisheye lens inverse perspective mapping (FLIPM) method. The new mapping equations are derived to transform the images captured by the fisheye lens camera into the undistorted remapped ones under practical circumstances. In the obstacle detection, we make use of the features of vertical edges on objects from remapped images to indicate the relative positions of obstacles. The static information of remapped images in the current frame is referred to determining the features of source images in the searching stage from either the profile or temporal IPM difference image. The profile image can be acquired by several processes such as sharpening, edge detection, morphological operation, and modified thinning algorithms on the remapped image. The temporal IPM difference image can be obtained by a spatial shift on the remapped image in the previous frame. Moreover, the polar histogram and its post-processing procedures will be used to indicate the position and length of feature vectors and to remove noises as well. Our obstacle detection can give drivers the warning signals within a limited distance from nearby vehicles while the detected obstacles are even with the quasi-vertical edges.

1. Introduction

With the fast growing number of vehicles and traffic accidents in recent years, the advanced vehicle control and safety driving assistance in intelligent transportation systems (ITS) have been more and more important. It has played a significant role for the lateral obstacle detection system to improve the driving safety and assist drivers to reduce the dead angles of sight while driving. Moreover, the lateral obstacle detection could be integrated with that in the front or rear of a vehicle to make the obstacle detection system more robust and complete. In general, the objective of camera calibration is to extract the intrinsic and extrinsic information of the camera and the extracted information could be used to reconstruct the 3D world coordinate. Nevertheless, the performance of camera calibration would depend on the perspective effect, lens distortion, and the number of cameras. An alternative method, namely inverse perspective mapping (IPM), was proposed to reconstruct the 3D world coordinates by using a single camera only. Broggi

et al. [1, 2] utilized the IPM method and stereo cameras to detect obstacles in front of the vehicle, and implemented the parallel processor for image checking and analysis (PA-PRICA) system Single Instruction Multiple Data (SIMD) computer architecture, to construct their obstacle and lane detection system, called GOLD (Generic Obstacle and Lane Detection) [2]. The GOLD implemented in the ARGO (derived from Argo and Argus, a research group from Italy) experimental vehicle made automatic driving possible. Ji [3] utilized IPM to get the 3D information of the front vehicle, and Cerri and Grisleri [4] presented the stabilized subpixel precision IPM image and the time correlation to estimate the possible driving space on highways. Muad et al. [5] used IPM to implement lane tracking and gave discussions of the factors which might have the influences on IPM. Tan et al. [6] combined IPM and the optical flow to detect obstacles for the lateral blind spot of the vehicle. Jiang et al. [7] proposed the fast IPM algorithm and used it to detect lanes and obstacles. Nieto et al. [8] introduced how to stabilize IPM images by using vanish point estimation. However in their approaches

based on IPM, the planar objects such as lane markings were eliminated and the prominent objects like quasitriangle pairs were reserved. The performance of those detection methods would obviously depend on the height, width, distance, and shape of an obstacle.

There have been some other methods proposed for obstacle detection. Lai [9] used both of vision and the ultrasonic sensors on the mobile robot to detect the wall in the indoor environment. For the pedestrian detection, Curio et al. [10] used the contour, local entropy, and binocular vision to detect pedestrians. Bertozzi et al. [11] utilized stereo infrared cameras and three steps including warm area detection, edge-based detection, and v-disparity computation to detect pedestrians and used the morphological and thermal characteristics of heads to validate the presence of pedestrians. Though infrared cameras could perform well in either daytime or nighttime, the applications would be still restricted because of the higher prices of those cameras. There have existed many kinds of features such as symmetry, color, shadow, corner, Vertical/horizontal edges, texture, and vehicle light for vehicle detection [12]. Kyo et al. [13] used edges to detect possible vehicles and further validated the vehicles by the characteristics of symmetry, shadow, and differences in the gray-level average intensity, and Denasi and Quaglia [14] used pattern matching to detect and validate vehicles. These methods would usually fail if the obstacles did not match the defined models. For the general obstacle detection task, the optical flow-based and stereo-based methods have been most popular in recent researches. The optical flow based methods would detect obstacles by analyzing the differences between the expected and real velocity fields. Krueger et al. [15] combined the optical flow with odometry data to detect obstacles, but the optical flow-based methods would have the higher computational complexity and might fail if the relative velocity between obstacles and the detector was too small. For the stereo-based methods, Forster and Tozzi [16] utilized disparities of obstacles to detect obstacles and used a Kalman filter to track obstacles. However, stereo methods are highly dependent on the accuracy of identification of correspondences in the two images. In other words, searching the pairs of homogeneous points was much tougher for stereo-based methods.

In recent years, there were two important subjects, including improving the accuracy of compensation estimation and obstacle detection. After an IPM image was acquired, a serious problem on resolution between the original and remapped images might be caused. Therefore, how to get an appropriate compensation result would be difficult, especially in our fish-eye lens approach. In Yang et al. [17], the compensation estimation was gained by the recursive method in trials and errors. Firstly, he chose randomly two pixels with a predefined distance to compare the optical flow values until gaining twenty pairs, and then used the median pair to be the value of compensation estimation. However, the IPM remapped images may cause a serious problem for computing the optical flow values in case of the worse resolution. Furthermore, even if the recursion method was used to avoid choosing nonplanar pixels, it was still probably to get similar or nonplanar points

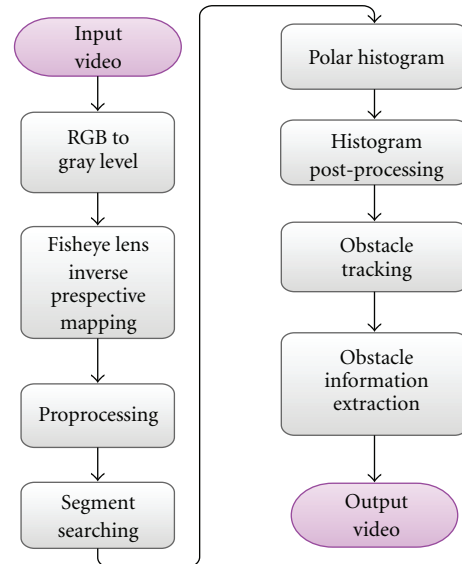


FIGURE 1: The overall system.

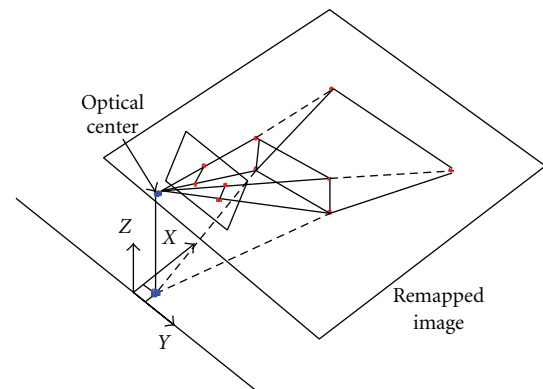


FIGURE 2: The vertical line projection of (1).

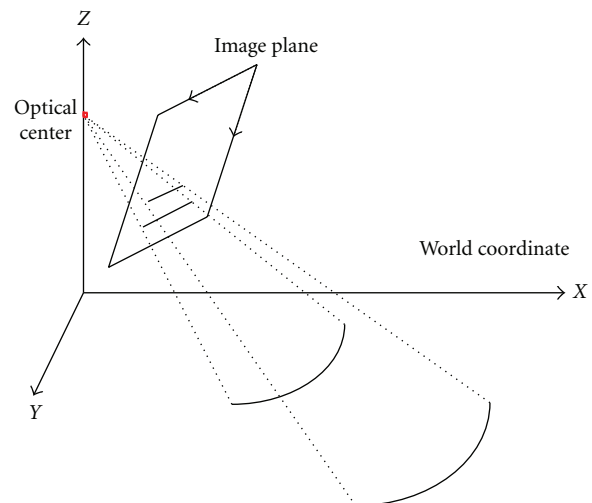


FIGURE 3: The projected results of (5).

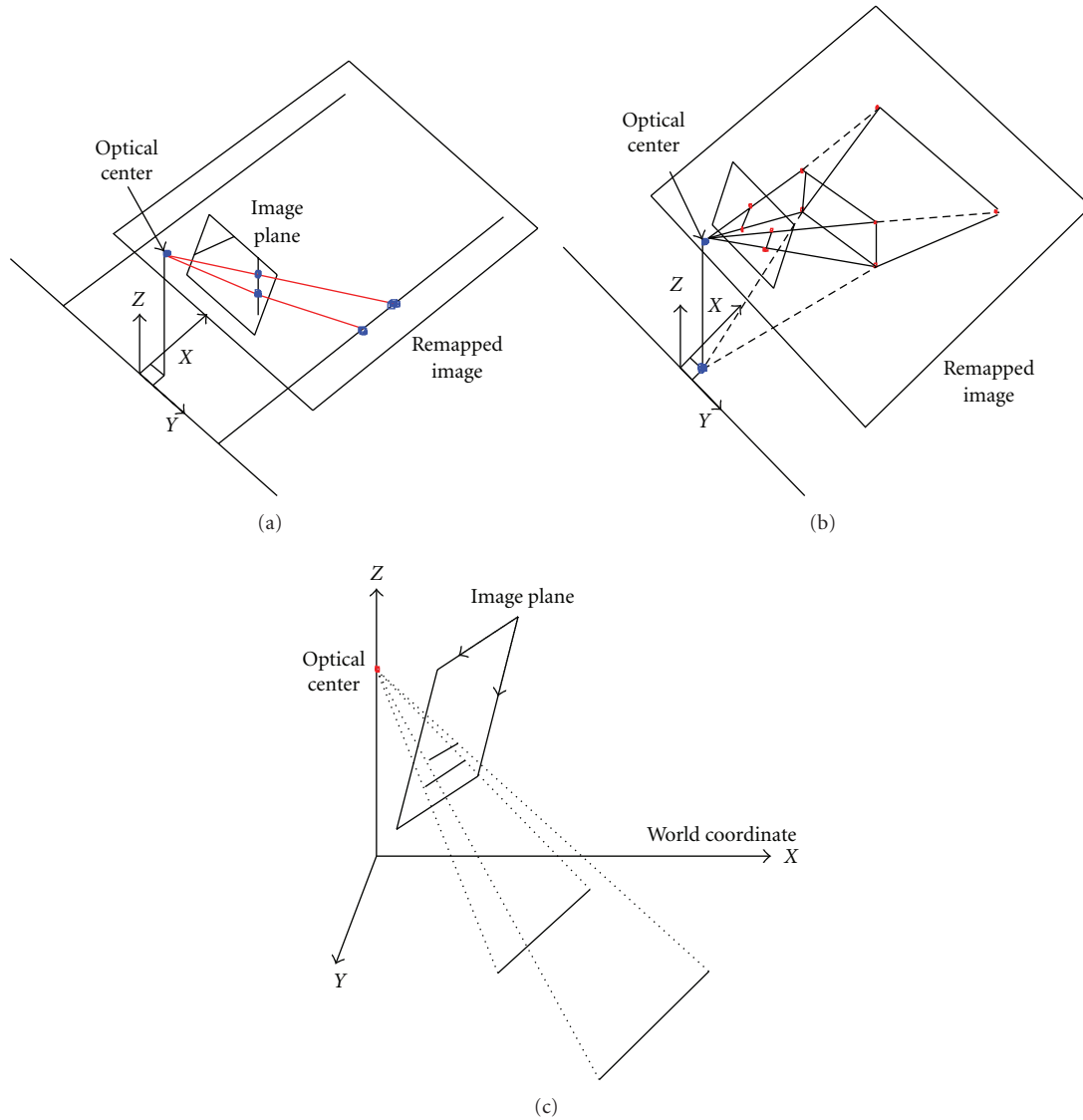


FIGURE 4: The figures and expected results for (a) perspective effect removing (b) a vertical straight line in the image will be projected to a straight line whose prolongation will pass the vertical projection point of the camera on the world surface (c) a horizontal straight line in the image will be projected to a straight line instead of an arc on the world surface.

when the values of optical flows were very close. In our approach, we adopted the edge features and images with time difference to improve the above problems in both static and dynamic environments. For dynamic environments, since the nonplanar edge features may change more vibrantly than planar edge features, the values of compensation estimation can be easily determined by the compensated image with the minimum number of candidate pixels of obstacles. To improve stability and robustness of our system, we considered both the time interval and the earlier k frames to average and update the latest compensation estimation. For obstacle detection, in Ma et al. [18] approach, he adopted the pedestrian features and symmetrical property to search the possible positions of obstacles in the region of interest. Although the performance of their system was acceptable, the results would be not stable and robust with the detection

rate in 58% ~ 92%. That was because the pedestrians' foot steps might be influenced by lane markings, shadows of trees, and any other planar noises. Our algorithm used the IPM's property; therefore, the polar histograms derived from the IPM images could help to obtain the information of images in 1-D distributions. For separating from nonplanar obstacles, we also constructed a novel method to detect and localize obstacles.

With the intrinsic and extrinsic parameters from camera calibration, the obstacle detection system could establish a transformation table for mapping the coordinates of real-road surfaces into the distorted image coordinates. The objective of IPM method was to remove the perspective effects caused by cameras, and the higher performance of IPM methods made it possible to achieve better image processing results. Since IPM methods have been proven to

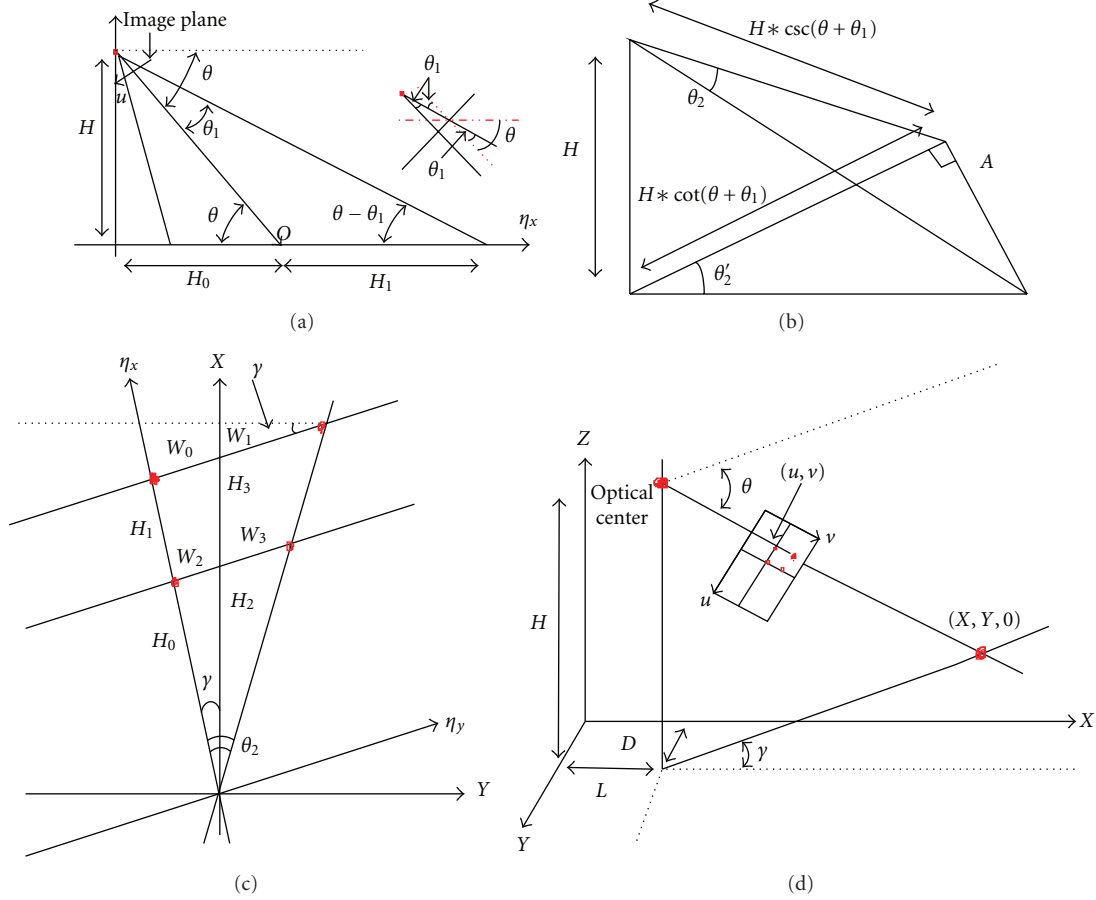


FIGURE 5: The geometrical relations of the image and world coordinate system for deriving our equations.

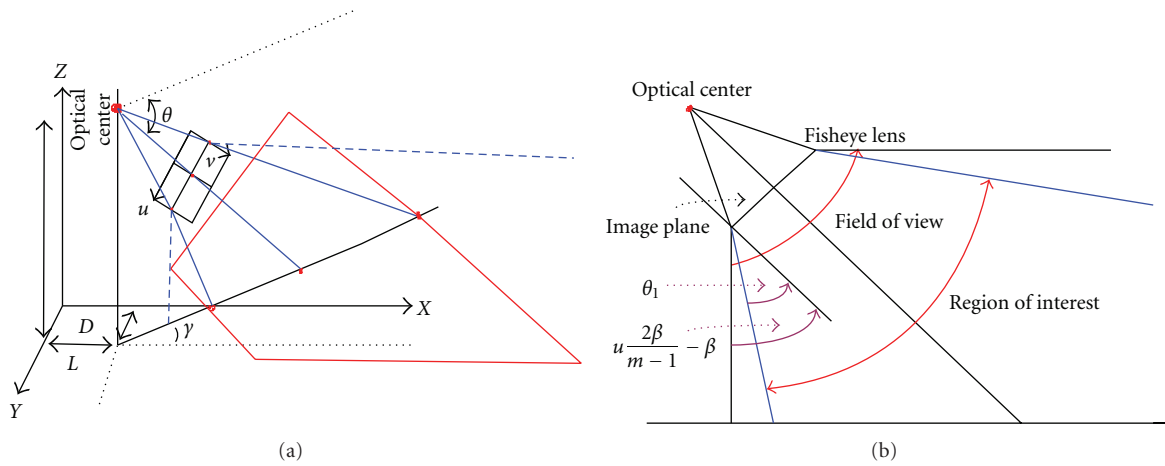


FIGURE 6: The original and adjusted scope.

be more efficient and applicable to real traffic conditions, we would focus on developing an accurate IPM algorithm for both normal lens and fisheye lens by improving the previous IPM methods. Our obstacle detection system aimed at detecting obstacles with either vertical or quasivertical edges. In fact, the obstacles with the significant height in vertical or quasivertical edges could be mapped to the radial lines

of the transformed bird-view images. As a result, we could deal with the transformed images to extract the profile of edges and obtain the polar histogram for post-processing. We have organized the following sections in this paper, including our systematic structure, the modified normal lens IPM method, fisheye lens IPM, obstacle detection algorithms, experimental results, and conclusions.

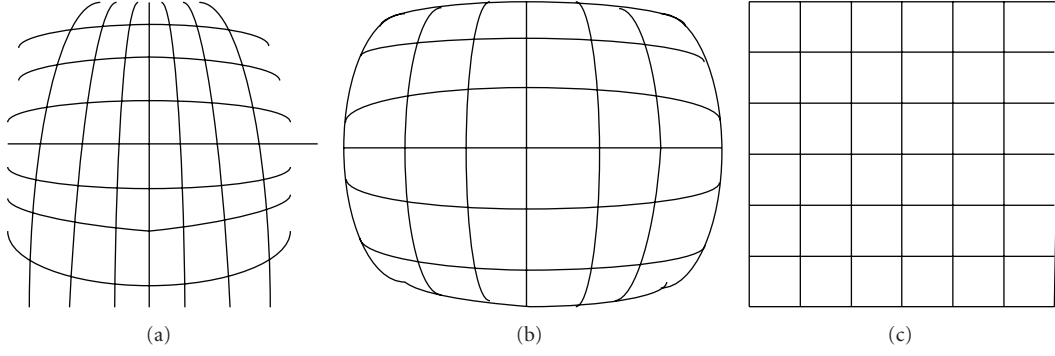


FIGURE 7: Illustrations for (a) the real scene image, (b) the distorted image, and (c) the desired image.

2. Our Systematic Structure

Our overall systematic structure is illustrated in Figure 1. The obstacle detection is performed after obtaining the bird-view images of road surfaces captured by the camera mounted on the lateral side of the vehicle. The edge profile of road surfaces in bird-view images or temporal FLIPM difference image should be acquired, and then the segment searching algorithm will use the edge profile to get the feature radial lines which indicate the obstacles. After searching the feature segments, the polar histogram which represents the direction and size of obstacles will be computed. The histogram post-processing will also be used to filter out some noises and obstacles with shorter height. We still have to identify the detected obstacles and extract the relative information of the obstacle after the obstacle tracing process. After all the processes, we can obtain the final results in the output videos.

3. The Modified Normal Lens Inverse Perspective Mapping Method

To find more practical applications and set up the appropriate mapping equations in our system, we modify the previous approaches proposed by Bertozzi and Broggi [2] and make the obstacle detection system more complete. Let u and v represent the image coordinate system and X, Y , and Z be the world coordinate system where $(X, Y, \text{and } 0)$ indicates the road surface. L, D , and H are the coordinates of the camera in the world coordinate system, while θ and γ are the camera's tilt and pan angles, respectively. α and β are the horizontal and vertical aperture angles. m and n indicate the height and width of an image. O is the optic axis vector, and η_x, η_y are the vectors representing the optic axis vector O projected on the road surface and its perpendicular vector

$$\begin{aligned} X &= H * \cot\left(\theta - \beta + u \frac{2\beta}{m-1}\right) * \cos\left(\gamma - \alpha + v \frac{2\alpha}{n-1}\right) + L, \\ Y &= H * \cot\left(\theta - \beta + u \frac{2\beta}{m-1}\right) * \sin\left(\gamma - \alpha + v \frac{2\alpha}{n-1}\right) + D. \end{aligned} \quad (1)$$

From (1), the vertical straight line in the image coordinate system can be represented by the set of pixels whose v

coordinate value is constant. If we assume that $Kv = \gamma - \alpha + v(2\alpha/(n-1))$ is constant, then (1) will be simplified to

$$\begin{aligned} X &= H * \cot\left(\theta - \beta + u \frac{2\beta}{m-1}\right) * \cos(Kv) + L, \\ Y &= H * \cot\left(\theta - \beta + u \frac{2\beta}{m-1}\right) * \sin(Kv) + D. \end{aligned} \quad (2)$$

After simple calculations, we can obtain (3) from (2), which is shown in Figure 2

$$X - L = (Y - D) * \cot(Kv). \quad (3)$$

Equation (3) means that a vertical straight line in the image which represents the vertical edge of obstacles or other planar markings in the world coordinate system will be projected into a straight line whose prolongation will pass the vertical projection point of the camera on the world surface.

Similarly, the horizontal straight line in the image coordinate system can be represented by the set of pixels whose u coordinate value is a constant. If we assume $Ku = \theta - \beta + u(2\beta/(m-1))$ is constant, then (1) will be also simplified to

$$\begin{aligned} X &= H * \cot(Ku) * \cos\left(\gamma - \alpha + v \frac{2\alpha}{n-1}\right) + L \\ &= K * \cos\left(\gamma - \alpha + v \frac{2\alpha}{n-1}\right) + L, \\ Y &= H * \cot(Ku) * \sin\left(\gamma - \alpha + v \frac{2\alpha}{n-1}\right) + D \\ &= K * \sin\left(\gamma - \alpha + v \frac{2\alpha}{n-1}\right) + D. \end{aligned} \quad (4)$$

Thus, we can derive (5) from (4), which is shown in Figure 3

$$(X - L)^2 + (Y - D)^2 = K^2. \quad (5)$$

Equation (5) means that a horizontal straight line in the image will be projected to an arc on the world surface. In order to modify the original IPM model, we propose a new pair of transformation equations for two expected results. First, a vertical straight line in the image will still be projected

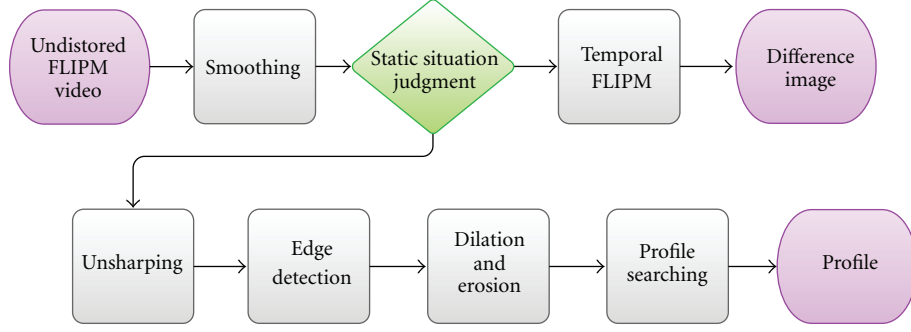


FIGURE 8: The flow chart of image pre-processing.

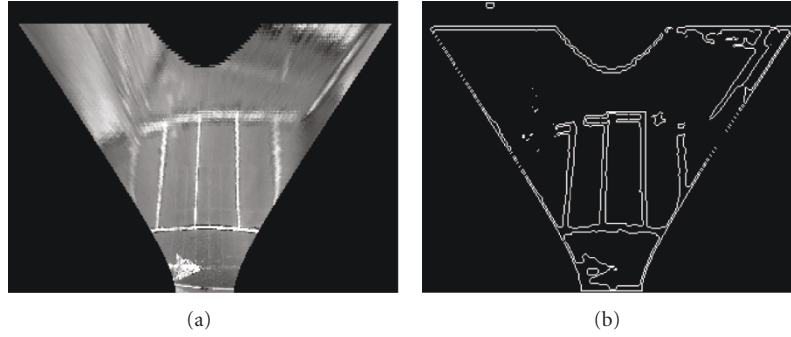


FIGURE 9: The results in the profile searching process, (a) the remapped image (b) the profile image.

to a straight line whose prolongation will pass the vertical projection point of the camera on the world surface. Second, a horizontal straight line in the image will be projected to a straight line instead of an arc on the world surface. The results can be verified by the similar triangle theorem. With some prior knowledge such as the assumptions on flat roads, intrinsic and extrinsic parameters, we will be able to reconstruct a 2D image without the perspective effect. The illustrated figures and expected results are shown in Figure 4.

By referring to the notations, the diagrams of relationship between the image coordinate system and the world coordinate system are shown in Figure 5. We will derive a new pair of transformation equations by simple mathematical computations in triangular functions. From Figures 5(a) and 5(b), we can obtain

$$\begin{aligned}
 &\rightarrow \theta_1 = u \frac{2\beta}{m-1} - \beta, \\
 &\rightarrow H_0 = H * \cot(\theta), \\
 &\rightarrow H_0 + H_1 = H * \cot(\theta + \theta_1), \\
 &\rightarrow \theta_2 = v \frac{2\alpha}{n-1} - \alpha, \\
 &\rightarrow \tan(\theta'_2) = \tan(\theta_2) * \sec(\theta + \theta_1).
 \end{aligned} \tag{6}$$

Figure 5(c) describes how the points in the first quadrant of the image coordinate system will be projected onto the road surface. If the world coordinate of camera is $(0, 0, H)$, we will

finally obtain (7) by the geometrical descriptions in Figures 5(c) and 5(d) and the length of each segment listed below:

$$\begin{aligned}
 &\rightarrow H_0 = H * \cot(\theta), \\
 &\rightarrow H_0 + H_1 = H * \cot(\theta + \theta_1), \\
 &\rightarrow H_2 = H_0 * \sec(\gamma), \\
 &\rightarrow H_2 + H_3 = (H_0 + H_1) * \sec(\gamma), \\
 &\rightarrow W_0 = (H_0 + H_1) * \tan(\gamma), \\
 &\rightarrow W_0 + W_1 = (H_0 + H_1) * \tan(\theta_2) * \sec(\theta + \theta_1), \\
 &\rightarrow W_2 = H_0 * \tan(\gamma), \\
 &\rightarrow W_2 + W_3 = H_0 * \tan(\theta_2) * \sec(\theta + \theta_1), \\
 &\Rightarrow X = H_2 + H_3 + W_1 * \sin(\gamma) = H * \cot(\theta + \theta_1) \\
 &\quad * [\cos(\gamma) + \sec(\theta + \theta_1) * \tan(\theta_2) * \sin(\gamma)], \\
 &\Rightarrow Y = W_1 * \cos(\gamma) \\
 &\quad = H * \cot(\theta + \theta_1) [-\sin(\gamma) + \sec(\theta + \theta_1) \\
 &\quad * \tan(\theta_2) * \cos(\gamma)].
 \end{aligned} \tag{7}$$

Now, we have obtained the forward transformation equations, and the backward transformation equations shown in (8) can also be obtained easily by some mathematical computations in inverse functions

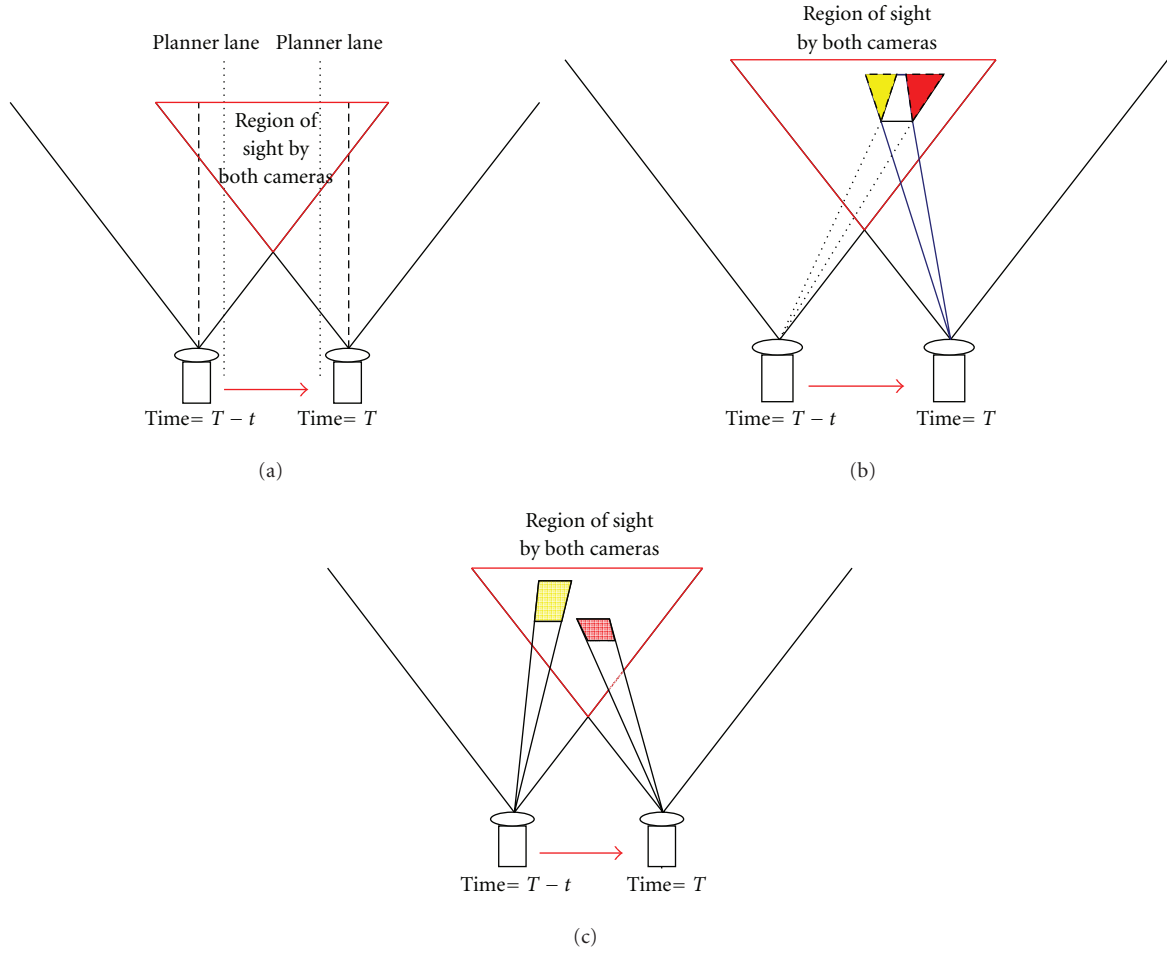


FIGURE 10: Illustrations for the temporal FLIPM difference image, (a) the planar object patterns and (b) nonplanar object patterns (c) Moving nonplanar object patterns.

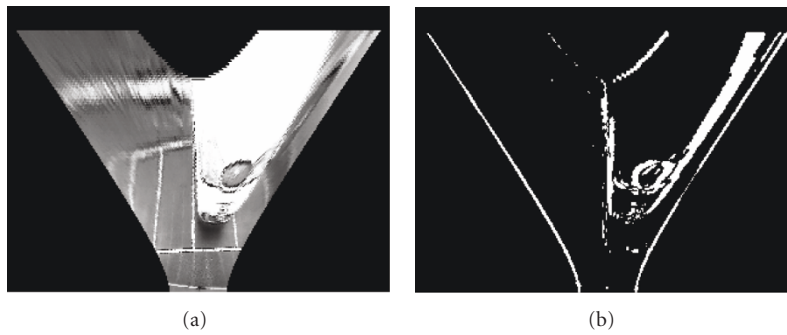


FIGURE 11: The results of the temporal FLIPM process, (a) the remapped image (b) the temporal difference image.

$$\begin{aligned}
 \Rightarrow \theta_1 &= \cot^{-1} \left(\frac{X * \cos(\gamma) - Y * \sin(\gamma)}{H} \right) - \theta, & * \left(\cot^{-1} \left(\frac{X * \cos(\gamma) - Y * \sin(\gamma)}{H} \right) - \theta + \beta \right), \\
 \theta_2 &= \tan^{-1} \left(\frac{X * \sin(\gamma) + Y * \cos(\gamma)}{H * \csc(\theta + \theta_1)} \right), & \\
 \Rightarrow u &= \frac{m - 1}{2\beta} & \Rightarrow v = \frac{n - 1}{2\alpha} * \left(\tan^{-1} \left(\frac{X * \sin(\gamma) + Y * \cos(\gamma)}{H * \csc(\theta + \theta_1)} \right) + \alpha \right). \tag{8}
 \end{aligned}$$

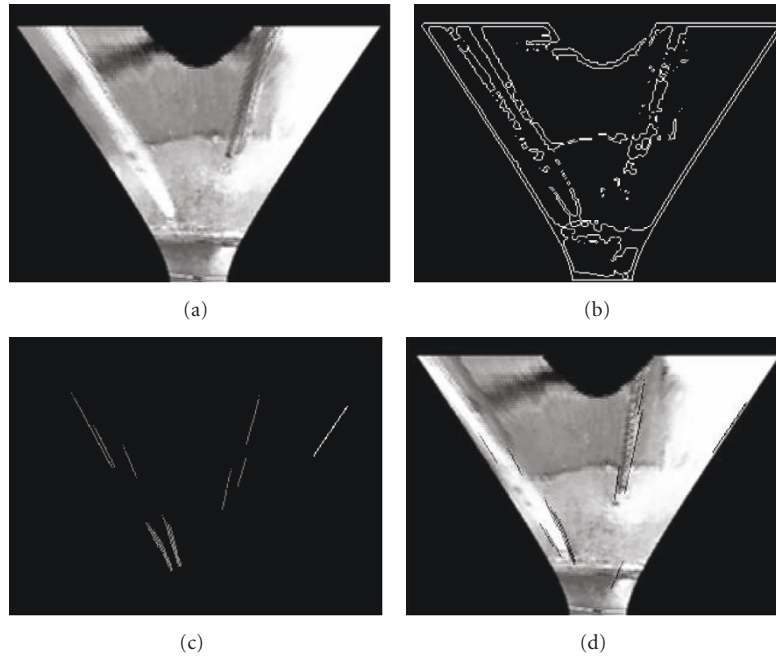


FIGURE 12: The results in the feature searching procedure by using profile images, (a) the sharpened remapped image, (b) the profile image, (c) the scanned feature segments, and (d) the scanned feature segments superposed on the sharpened remapped image.

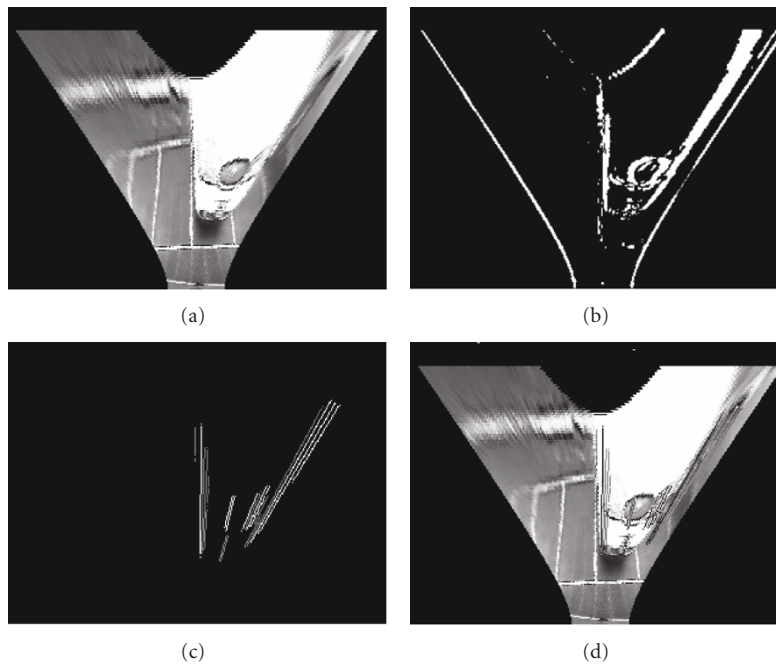


FIGURE 13: The results of the feature searching procedure using temporal difference FLIPM images, (a) the remapped image, (b) the temporal difference FLIPM image, (c) the scanned feature segments, (d) the scanned feature segments superposed on the remapped image.

4. Fisheye Lens Inverse Perspective Mapping (FLIPM)

4.1. *The Fisheye Undistortion Model.* Zhang and Fu[19] proposed a camera spherical projection model to implement the endoscope image formation process and utilized the warping

transformation equations to correct the radial distortion. The warping transformation equation pairs and its inverse pairs are shown in (9). The coordinate $(X, Y, \text{ and } Z)$ is the position of point in the 3D world coordinate system, (u_1, v_1) is the coordinate in the undistorted image, and (u, v) is the coordinate in the distorted one

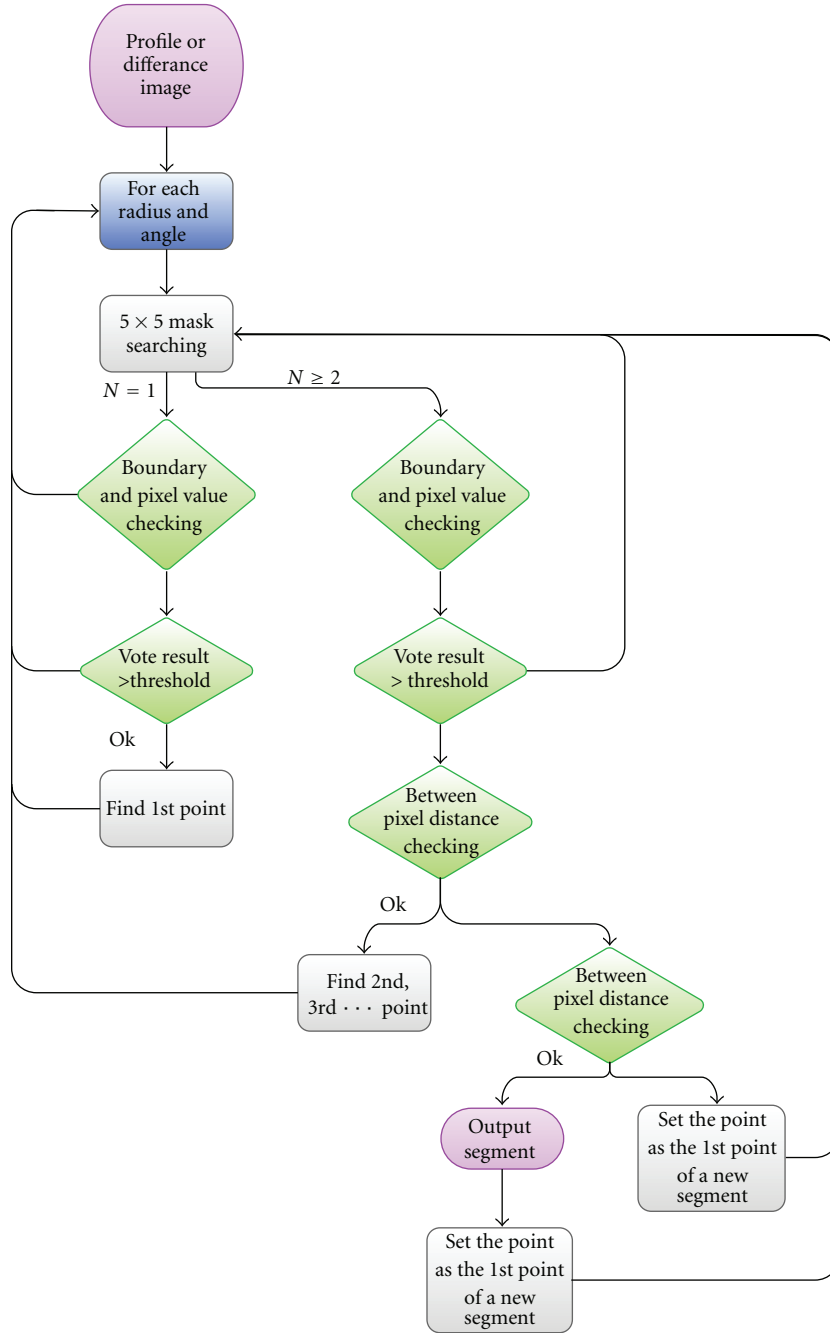


FIGURE 14: The flow chart of feature searching.

$$\begin{aligned}
 X &= \frac{f * u}{\sqrt{R^2 - u^2 - v^2}}; & Y &= \frac{f * v}{\sqrt{R^2 - u^2 - v^2}}, \\
 u &= \frac{R * X}{\sqrt{f^2 + X^2 + Y^2}}; & v &= \frac{R * Y}{\sqrt{f^2 + X^2 + Y^2}},
 \end{aligned} \tag{9}$$

where F is the focal length of camera, and R is the radius of the sphere. We modify and redefine that model for our applications in this paper. We regard the X_1 - Y_1 plane as an undistorted image plane and the u - v plane as the distorted one, thus we can derive the modified equations in (10)

$$\begin{aligned}
 u_1 &= \frac{f * u}{\sqrt{R^2 - u^2 - v^2}} = k_1 * \sqrt{\frac{u^2 + v^2 * \sin^2(\theta_1)}{\cos^2(\theta_1)}}, 0 \\
 v_1 &= \frac{f * v}{\sqrt{R^2 - u^2 - v^2}} = k_1 * \sqrt{\frac{v^2 + u^2 * \sin^2(\theta_2)}{\cos^2(\theta_2)}}, \\
 u &= \frac{R * u_1}{\sqrt{f^2 + u_1^2 + v_1^2}} = \frac{u_1}{k_1 \sqrt{1 + (\tan^2(\theta_1) + \tan^2(\theta_2))}}, \tag{10} \\
 v &= \frac{R * v_1}{\sqrt{f^2 + u_1^2 + v_1^2}} = \frac{v_1}{k_1 \sqrt{1 + (\tan^2(\theta_1) + \tan^2(\theta_2))}},
 \end{aligned}$$

where $k_1 = f/R$, and $\theta_1 = \sin^{-1}(u/\sqrt{R^2 - v^2}) = \tan^{-1}(u_1/f)$ and $\theta_2 = \sin^{-1}(v/\sqrt{R^2 - u^2}) = \tan^{-1}(v_1/f)$ are the angles between the lines connected the horizontal or vertical direction projection point of an image point with the optical center on the optical axis. Equation (11) instead of (13) will be used through this paper since (13) may produce many nonpixel-values of the image. We also can obtain the distorted or undistorted images no matter if the focal length is known or not by tuning the parameter k_1 .

4.2. The Complete Fisheye Lens Inverse Perspective Mapping. A fisheye lens inverse perspective mapping (FLIPM) algorithm will be complete by two parts, the forward and backward mapping algorithm. The objective of the forward mapping algorithm is to search the dimensions or ranges of remapped images, which can be illustrated in Figure 6.

The dimensions of scopes are only related to the view-ranges of a camera; that is to say, either the use of the normal lens or fisheye lens with fixed tile and pan angle will determine the factors of influences. In order to reduce the computational loadings in use of the tangent and secant triangular functions, we restrict the scope of a camera by narrowing down its view-range. Without loss of generality, we still keep the broadest range of scopes and minimize discarding far and fringe information. Furthermore, we narrow down the view-angles by using Snell's Law as shown in (11) where IR simulates the index of refraction and controls the scopes of resultant ranges. The range of IR is between 1.3~1.7 for glass-based lens

$$\begin{aligned} \sin\left(u\frac{2\beta}{m-1} - \beta\right) &= \text{IR} * \sin(\theta_1), \\ \sin\left(v\frac{2\alpha}{n-1} - \alpha\right) &= \text{IR} * \sin(\theta_2). \end{aligned} \quad (11)$$

The angles θ_1 and θ_2 can be substituted into (8) to compute the extreme values about the coordinate values of X and Y , and in this way, we will obtain the dimension of the remapped image. The backward mapping algorithm is different from the forward one because a plus of the radial distortion correction step should make it more rational. We firstly consider the ideas of the backward mapping algorithm by Figure 7.

Since the images captured by the fisheye cameras which can be shown in Figure 7(a) have the perspective effects and distortions, we have to remove those undesired effects to acquire the available images just like Figure 7(b) in pursuit of Figure 7(c), where the perspective effect and distortion have been completely removed. Thus so, we can derive the backward mapping algorithm by modifying (8) as (12). We also complete the distortion correction process by using (13) and the derived formulas of angles in (12). By tuning the parameter of IR and k_1 , we will

easily obtain the undistorted and perspective effect removed images

$$\begin{aligned} \theta_1 &= \cot^{-1}\left(\frac{X * \cos(\gamma) - Y * \sin(\gamma)}{H}\right) - \theta, \\ \theta_2 &= \tan^{-1}\left(\frac{X * \sin(\gamma) + Y * \cos(\gamma)}{H * \csc(\theta + \theta_1)}\right). \end{aligned} \quad (12)$$

5. Our Obstacle Detection Algorithm

In this section, we develop an obstacle detection algorithm by using both spatial and temporal information of the FLIPM method. We use a single fisheye camera mounted on the lateral side of the vehicle to detect obstacles. The definitions of obstacles in this paper are the objects with the height shorter than a threshold and with nonquasivertical edges. The straight line in the vertical direction in the images represents the vertical edges of obstacles in the world coordinate system and will be projected to a straight line whose prolongation will pass the vertical projection point of the camera on the world surface. To illustrate our systematic mechanism more clearly, we will introduce the obstacle detection algorithm in the following parts, including some image preprocessing steps, feature selection, histogram analysis, object tracking, and information extraction.

5.1. The Preprocesses. We have to simplify the image patterns for our following procedures by some image preprocessing techniques shown in Figure 8. At first, the remapped image will be smoothed by mean filter to reduce the noises resulted from FLIPM transformation. Our developed equations in FLIPM have the advantages of IPM in removing the information of height and can help to detect the obstacles on the surface of roads. We also propose two different strategies toward feature extraction. We use the profile image which will be introduced next to extract the feature series when the detected objects and our cameras are relatively motionless, otherwise we acquire the features by the obstacle-sensitive temporal FLIPM difference image which will be clarified in Section 5.3.

5.2. Profile Image. The obvious edges of obstacles will be essential for extracting the profile images. We hence enhance the edges by the unsharp mask at this time to make up for over-blurred images, and detect edges by simple Sobel operations. The binary images can be obtained by thresholding after edge detecting of the remapped image, and we have to use the morphological operations on dilation and erosion to get the useful edges for our processes. As for extraction of the feature segments, we remodel the thinning algorithm introduced in [20] in thinning the binary edges in order to meet our real-time needs in the applications of ITS. We turn to use the center pixel of a mask to extract the exterior profile of a pattern without checking the conditions of patterns iteratively. Figure 9 shows the processed results of our profile image searching.

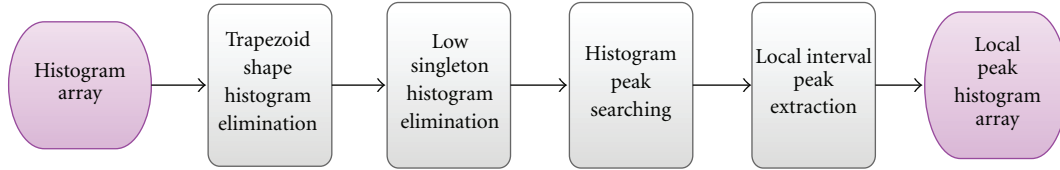


FIGURE 15: The flow chart of histogram postprocessing.

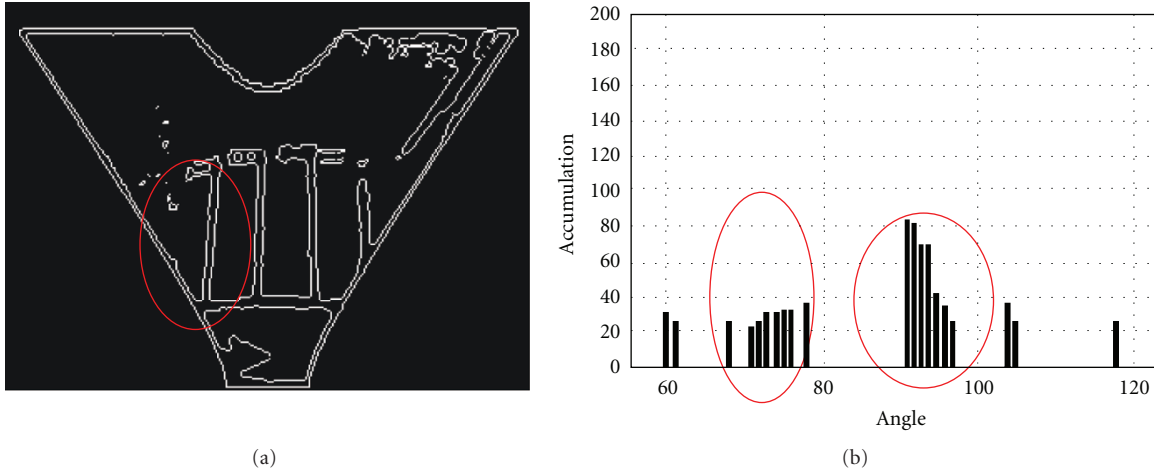


FIGURE 16: Illustrative figures of the trapezoid histogram distributions. (a) The figure of lane markings. (b) The trapezoid histograms.

5.3. *The Temporal Fisheye Lens Inverse Perspective Mapping Difference Image.* The objective of temporal FLIPM difference process is to simulate the stereo vision of captured images. The stereo IPM can keep the non-plane objects and remove the plane objects such as lane-markings, shadows by comparing the differences between the left and right remapped image, which will be illustrated in Figure 10.

According to the stereo IPM method [21], two cameras should be used to acquire the sufficient information of overlapped regions. Since this paper focuses on using a single camera, we take advantage of time difference to simulate the effects of stereo cameras. As Figure 11 shows, we have to address on two important issues, selections of time interval and the shift displacement of the remapped image. From the FLIPM, both the maximum movement of shift displacement and self moving speed have been restricted by the position of cameras. Therefore, we shall concentrate on deciding the time interval to compensate the remapped images. For instance, as shown in Figure 11(a), the remapped image can be estimated while the maximum shift displacement is the real horizontal distance of the remapped image. With the maximum moving speed, we can get the appropriate time interval by keeping the temporal difference images within an obvious displacement. We determine the value each time by the assumption that the compensated profile image has the minimum nonplanar pixels if the acquired value is the optimized one. As a result, we accumulate the movements in the earlier k frames to update the latest compensate displacement, and the value k is variant to different speeds. When the compensation method works in

the complex background such as many moving objects are in the field of views, the IPM effect of moving objects will cause a different projection between the front and rear frames as shown in Figure 10(c). Furthermore, to estimate the needs of compensation like moving directions of obstacles, we accumulate the movements of planar features of edges in k frames to obtain the adaptive value of compensation estimation. It is more difficult and decisive to determine the shift displacement of remapped images, for selecting the appropriate time interval may be easier for the expected performance of our obstacle detection system. We use the average displacement of remapped images from two cameras as the shift displacement in our temporal FLIPM method to obtain the “pseudo” stereo effect binary difference image.

5.4. *Feature Searching Algorithm.* As mentioned previously, we only prefer to search the features extracted from the objects with quasivertical edges in the remapped image. Based on the observation that those qualified features will always pass through the vertical projection point of cameras, we propose a feature searching algorithm and use polar histogram to accurately detect obstacles even for the noisy images. Our searching algorithm begins with the vertical projection point of a camera in the remapped image (denoted as CP). After that, we scan the acquired profile or temporal difference image angle by angle from the center to outmost border of a circle in the defined radius which can be determined by the information of remapped images. We then use a voting method in the mask searching and adjust

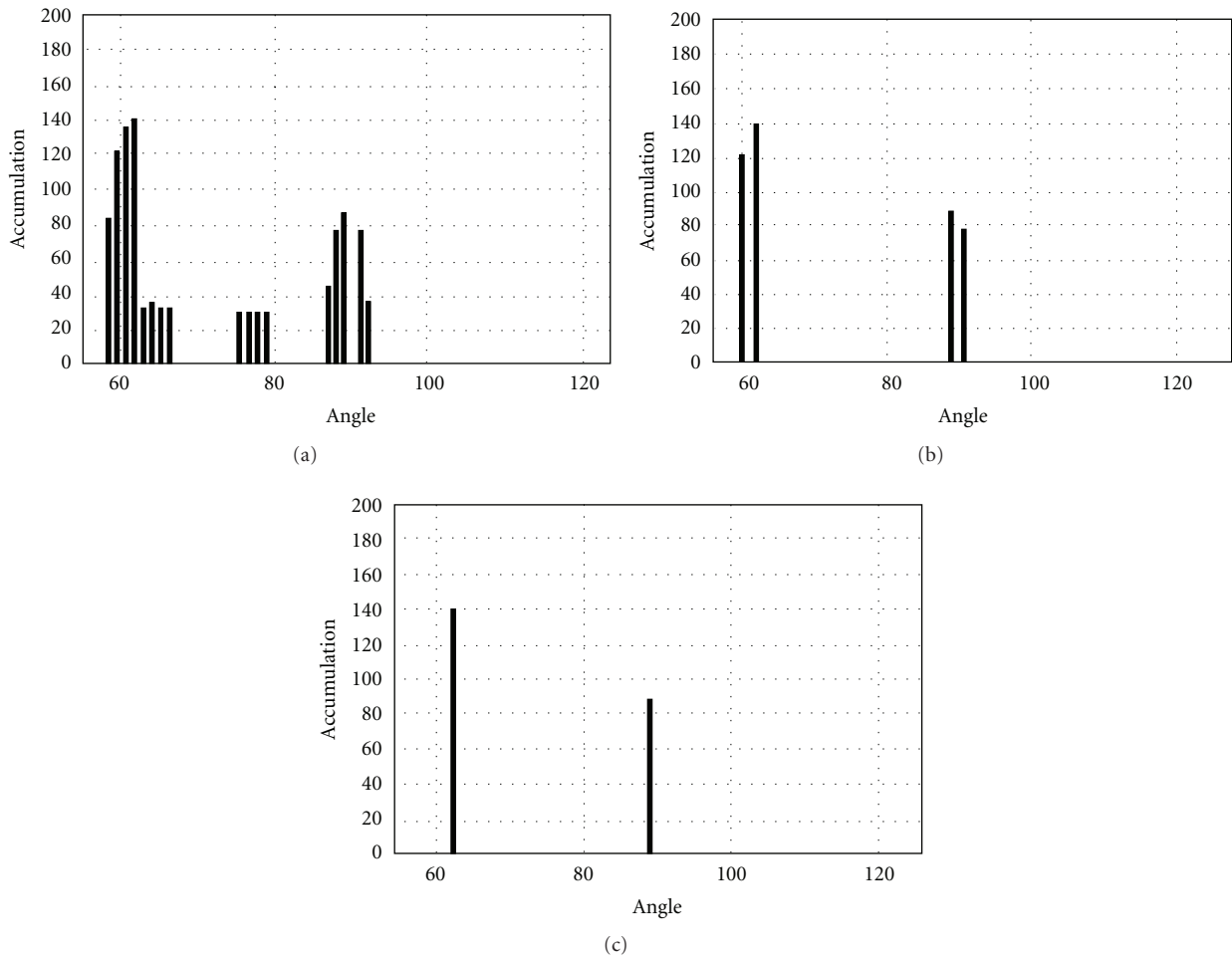


FIGURE 17: The processes of histogram post-processing, where x -axis and y -axis represent the angles of polar histograms and the accumulation amount on each angle, respectively. (a) The polar histogram of Figure 13. (b) The histogram of (a) after the trapezoid histogram elimination, low singleton histogram elimination and the peak searching procedure. (c) Local peak histogram.

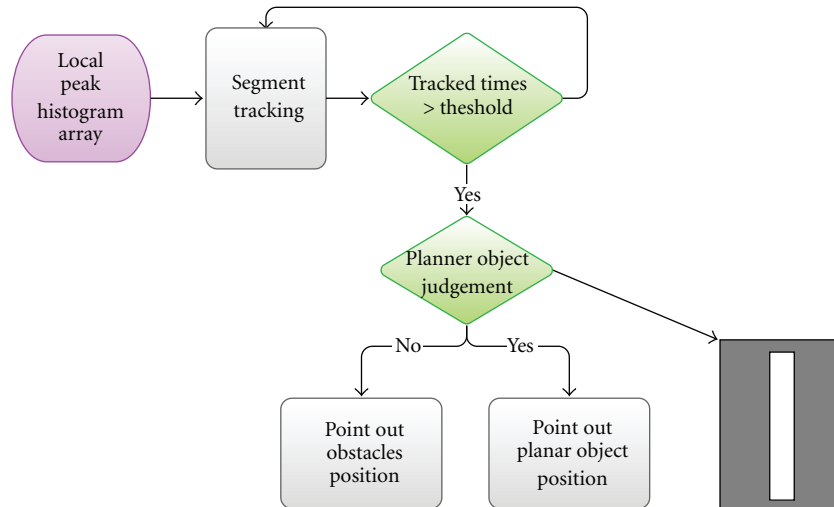


FIGURE 18: The flow chart of object tracking and information extraction.

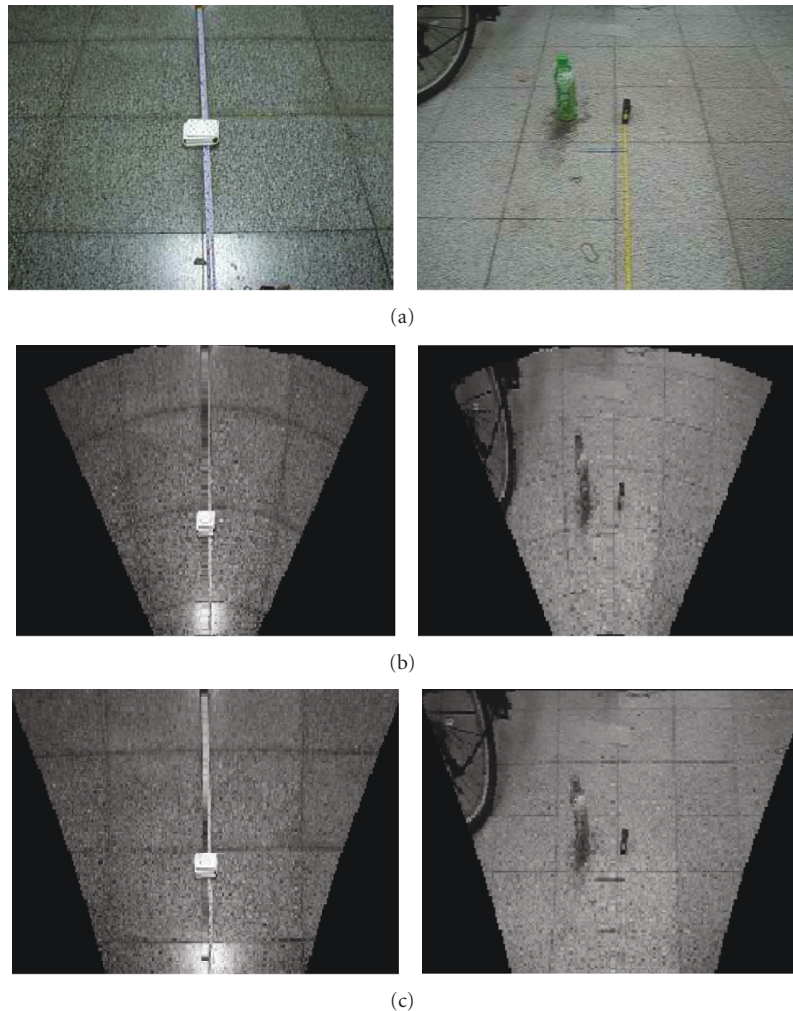


FIGURE 19: The results of the normal lens IPM equations (a) the original captured image (b) the bird-view image using Broggi's equations (c) the bird-view image using our equations.



FIGURE 20: The setup locations of cameras.

the searching space flexibly according to the intensities and relative distances between vehicles and obstacles. The voting threshold is fixed and can be determined by the half of the total elements in the mask. We can keep the major features, for the Gaussian weighting values in each 5×5 mask indicates the important regions in this mask. For each angle, a feature

segment will be taken only if its corresponding percentage is higher than some specific value in order to reduce the errors caused by statistics. The next searching point at the same angle must be close enough to the last searched segment so as to concentrate on the points close to CP. After the searching process at each angle, the number of points at each angle

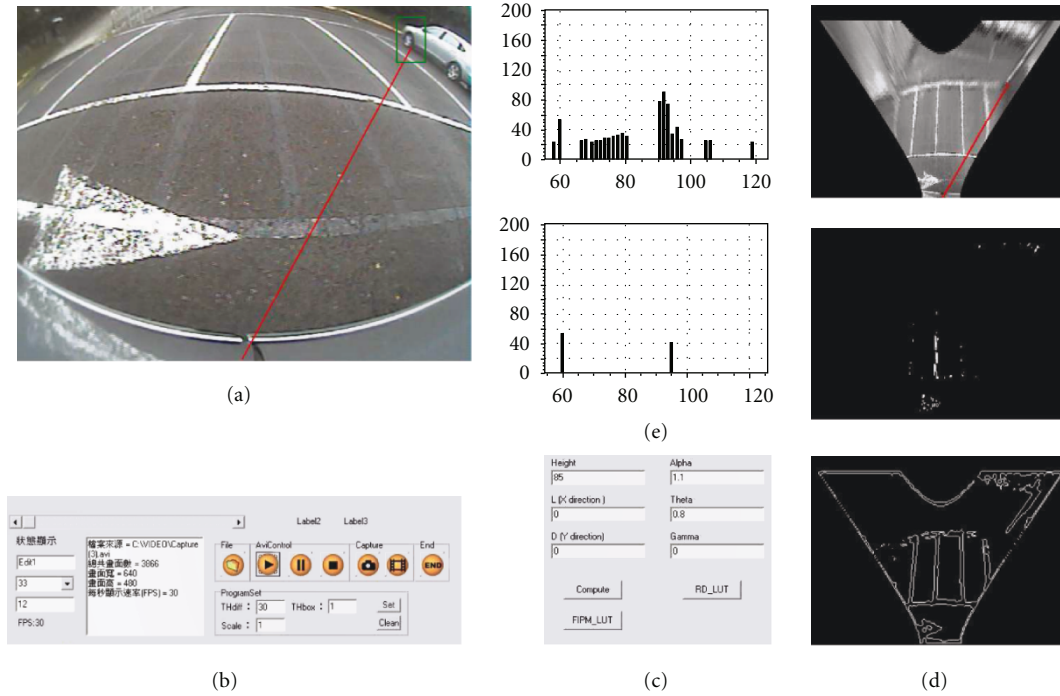


FIGURE 21: The interface of our program and the related information about adjustable parameters.

will be used to produce a polar histogram in our system. Some results in the feature searching procedure are shown in Figures 12 and 13 and the complete flow chart of our feature searching algorithm is shown in Figure 14. As Figure 14 shows, our algorithm can deal with two kinds of problems. One is that our systematic design can effectively improve the accuracy rate and reduce the possibility of lost pixels by using the features which are defined in each angle line by the results of mask searching to indicate the distance between objects and the camera. The other is that our proposed method can discriminate the meaningful pixels from others by the presented model for checking the length of searched feature segments. Therefore, the flaws of the polar histogram can be made up and our obstacle detection will make a great progress in performance.

5.5. Histogram Postprocessing. Figure 15 shows the processes in the histogram postprocessing. The postprocessing on histograms is necessary since we have to consider some important problems such as how to obtain the peak values of the histogram, how to reduce the influences of light, and how to find the best way in statistics for our applications. After we obtain the polar histogram of feature segments, we still need to find our desired histograms to remove the segments of planar objects and noises. Our procedures in histogram post-processing try to reduce the undesired information which may be produced in the polar histogram. For instance, the line segments belonging to planar objects will still be extracted in the polar histogram step. By observing the polar histogram, we can discover that the trapezoid histogram represents the planar objects. We can thus remove those clusters of bins in the histogram and we

do not process the oversmall histograms (the columns in the histogram are few) to avoid disturbances. After eliminating planar objects and noise, we will search the position of local maximum which represents the segment position of nonplanar object in the polar histogram. Also, we only pick a peak column at an angular interval to prevent from detecting too many obstacles at the same time. Some results in the histogram post-processing procedure will be shown in Figures 16 and 17. As Figure 16 shows, the regions in red circles (Figure 16(b)) are corresponding to the lane markings, as shown in Figure 16(a). In Figure 16(b), x -axis and y -axis represent the angles of polar histograms and the accumulation on each angle, respectively.

5.6. Object Tracking and Information Extraction. Our tracking procedure is used to confirm the detected objects. We choose the displacement and variation of angles in the extracted feature segment as the judgment conditions in the tracking process. If the feature segment has been extracted, we would judge again whether this feature segment belongs to the planar object by a pattern matching approach. Our tracking process and the pattern for representing the planar object are shown in Figure 18. We can finally confirm that the detected feature segment is an obstacle and also obtain the position of feature segments or other useful information.

6. Experimental Results

We arranged the information of the working platform and listed in Table 1 shown below. To show our experimental results more clearly, we categorized the experiments

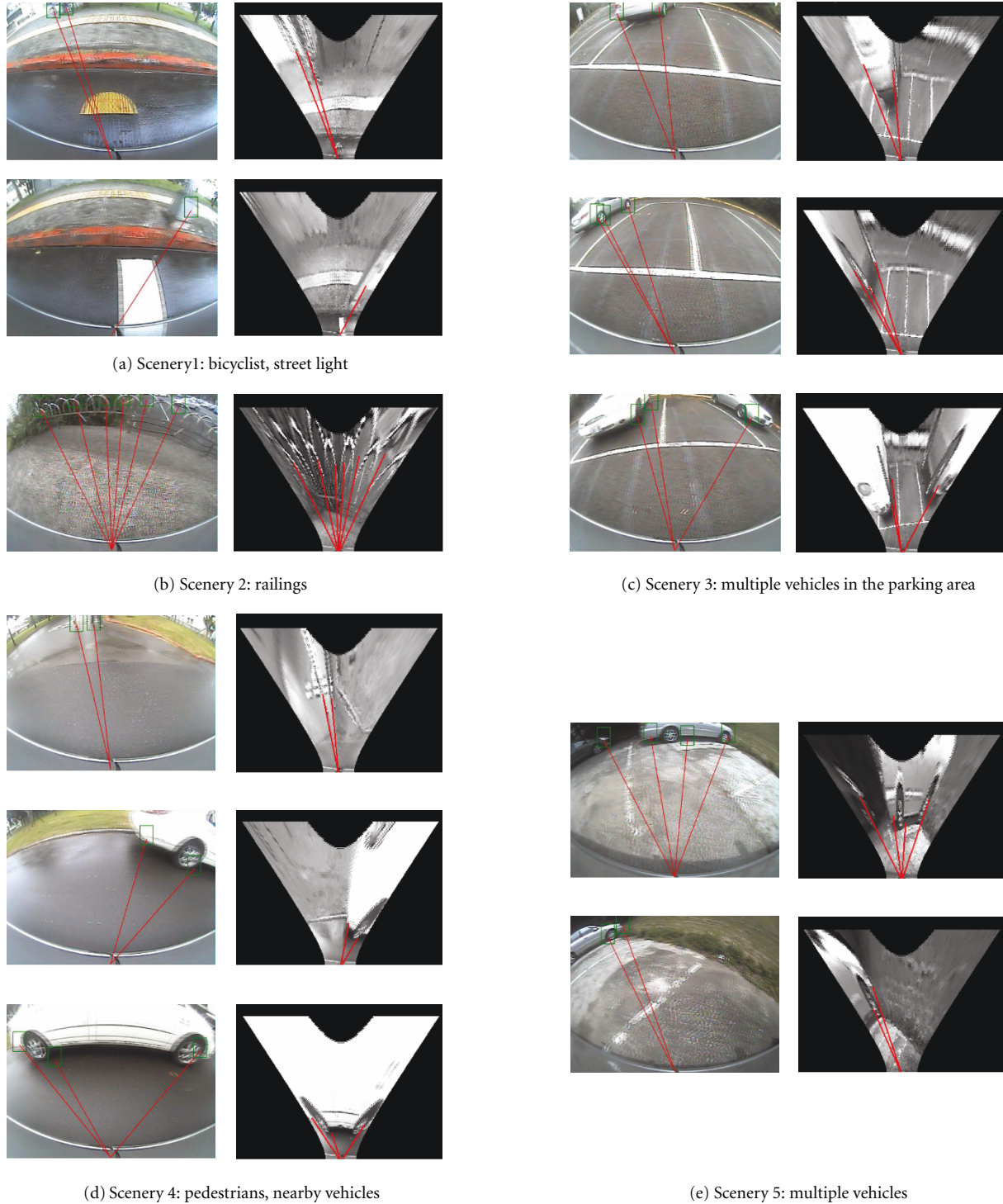


FIGURE 22: The experimental results of FLIPM and obstacle detection in different scenes (a) Scenery1: bicyclist, street light (b) Scenery 2: railings (c) Scenery 3: multiple vehicles in the parking area (d) Scenery 4: pedestrians, nearby vehicles (e) Scenery 5: multiple vehicles.

according to the proposed process and approach which have been introduced in the previous sections as follows.

Table 2 showed the runtime in each processing step defined in Figure 1. As Table 2 demonstrated, our system processed 15 frames per second. We tested the complete system by two parts, Input and Display stages. Therefore, the

performance of the whole system could be improved easily by upgrading the video I/O equipments and optimizing the FLIPM Kernel functions in regions of interest. Table 3 exhibited the performance of different obstacle algorithms. We gave the compared results in four parts, including the runtime, types of sensors, moving compensation, and field

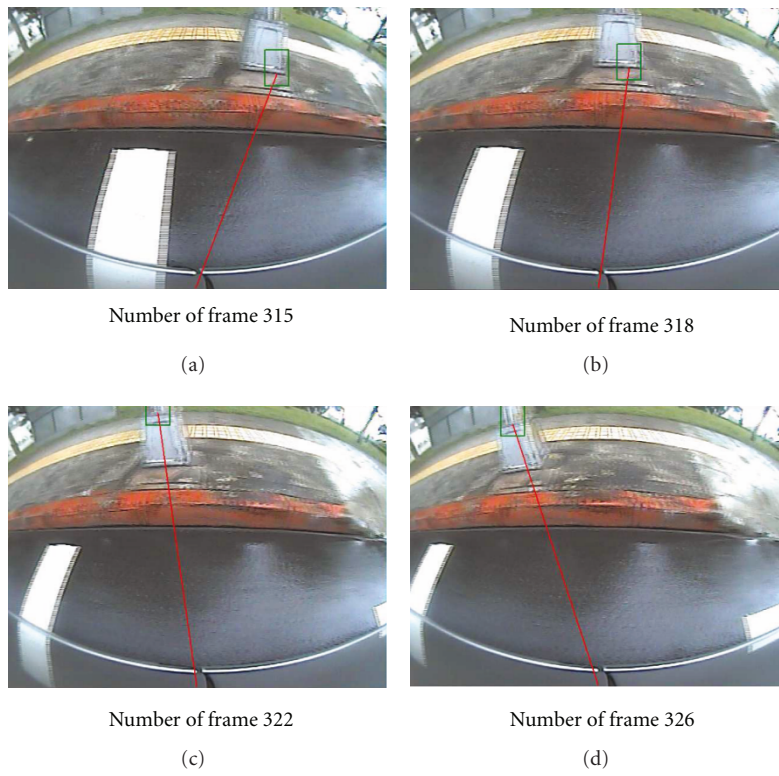


FIGURE 23: Results of obstacle tracking in Scenery 1.

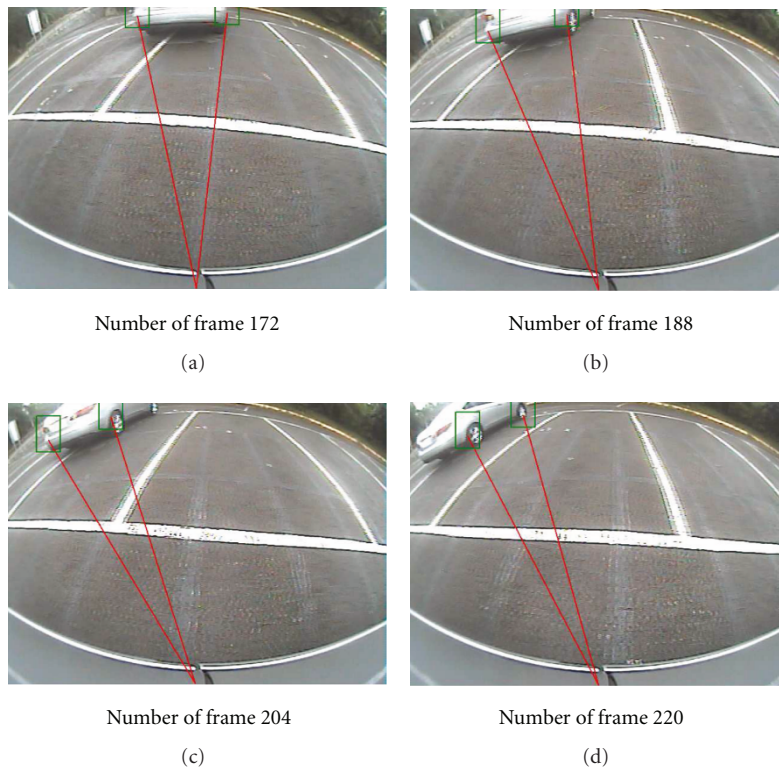


FIGURE 24: Results of obstacle tracking in Scenery 3.

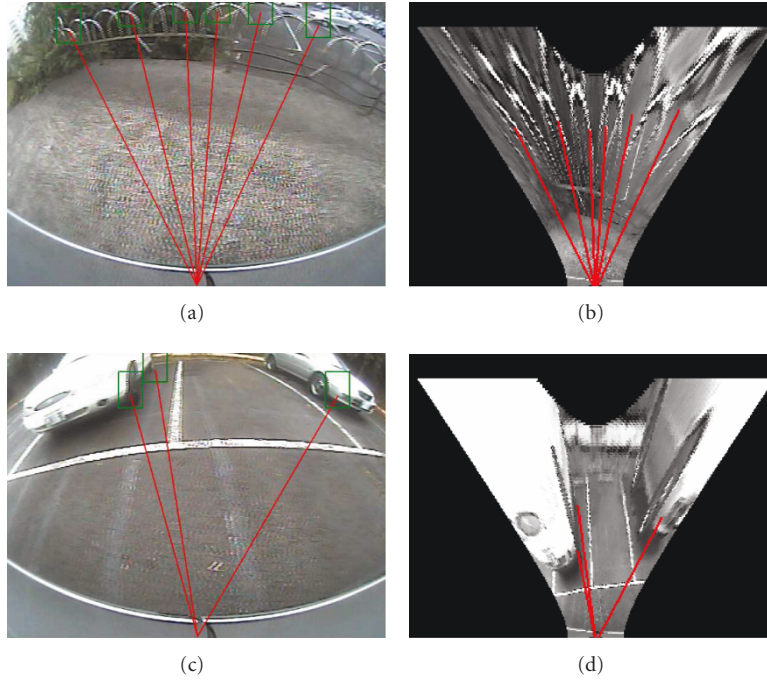


FIGURE 25: Results of obstacle warning in the lateral direction.

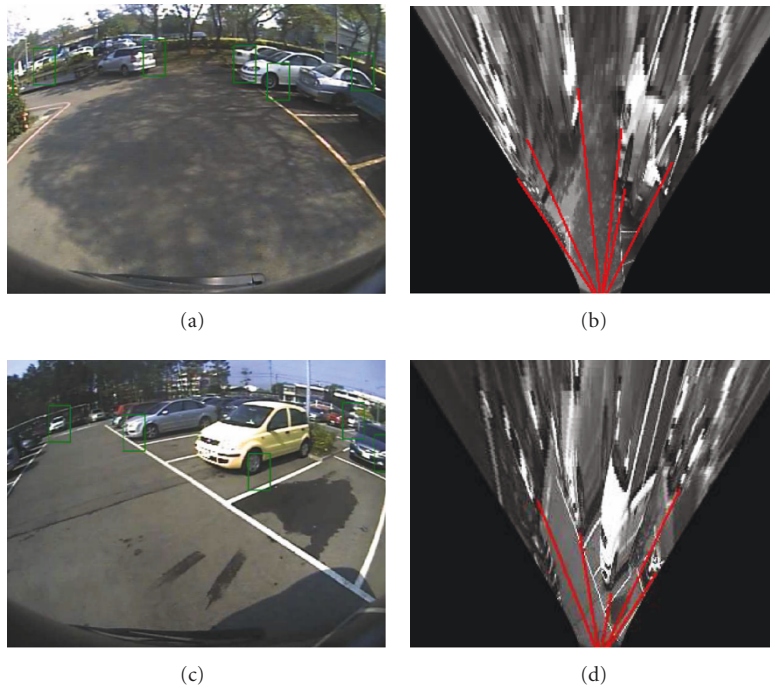


FIGURE 26: Results of obstacle warning in the rear direction.

of views. Although our approach included image I/O routine processes on common development platform, it still had the better performance than the others. On the detection module, we adopted the polar histogram to simplify the analytic step. It had two benefits where one was to reduce

the complexity and accelerate the processing speeds, and the other was to improve the detection rate and accuracy of obstacle detection. In Table 3, [2, 17], and our approach considered the polar histogram, however, our system had the better detection rate than the others and might not be

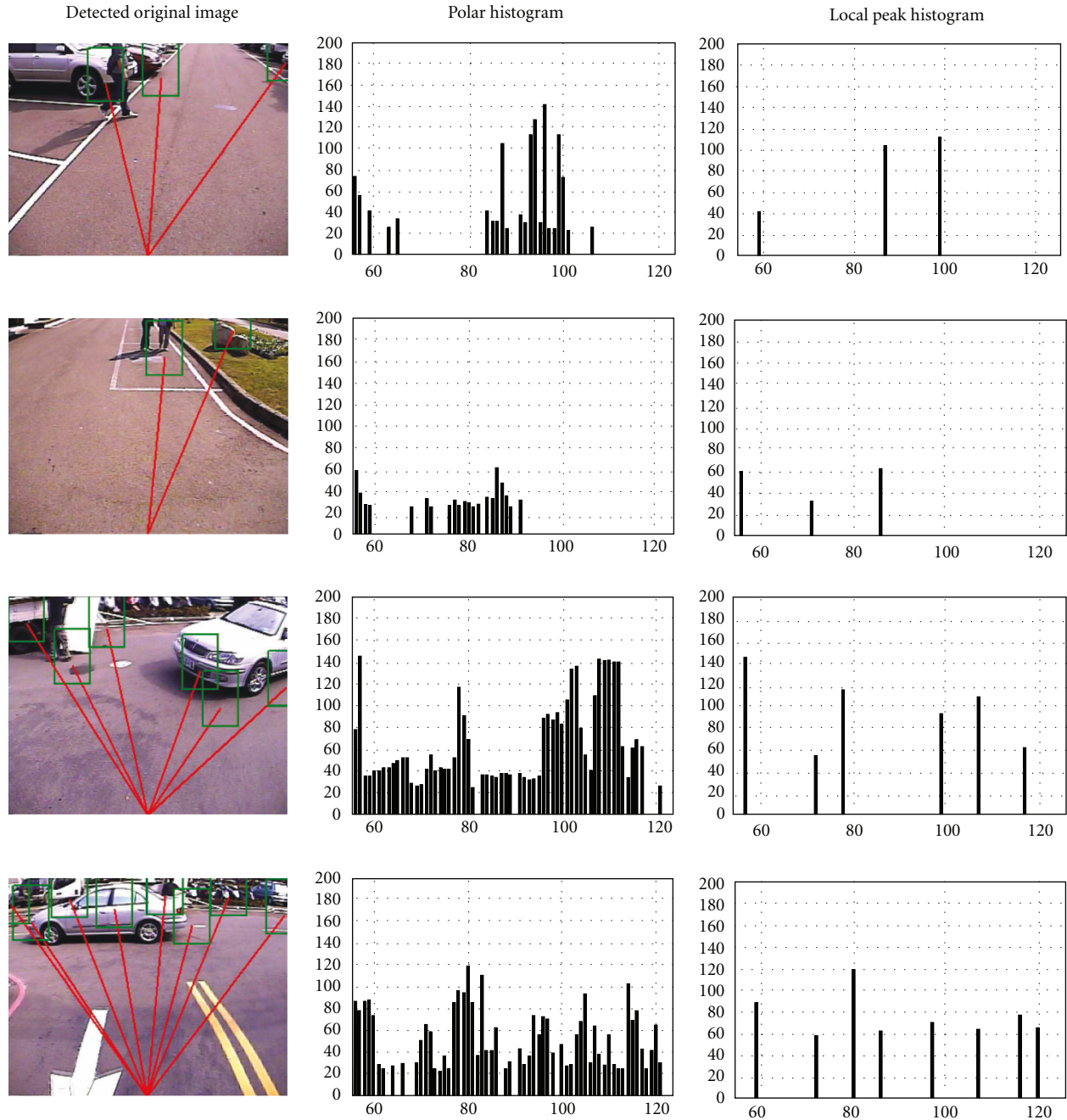


FIGURE 27: Results of obstacle warning with moving objects. Upper two rows are the results with moving humans. Bottom two rows are the results with a moving vehicle turning into the corner.

easily influenced by the planar markings, shadows, and other noises.

6.1. Comparisons About the Normal Lens Inverse Perspective Mapping Method. In Section 4, we proposed a modified forward and backward normal lens IPM equation pairs. The experimental results of our proposed approach and the most popular one by Broggi's equations were given in Figure 19. From Figure 19, the captured images by the normal lens camera were transformed to the bird-view images by using our equations. In Figures 19(b) and 19(c), the perspective

effect was eliminated by both of Broggi's and our equations. Nevertheless, the horizontal line in the original image would be transformed to an arc by Broggi's equations as shown in Figure 19(b). With our modified equations, the horizontal straight line in the original image could be transformed to a horizontal straight line in the bird-view image as shown in Figure 19(c).

6.2. The Experimental Configurations. For the experiments in obstacle detection, we mounted a fisheye lens camera at the center between two side doors with the appropriate

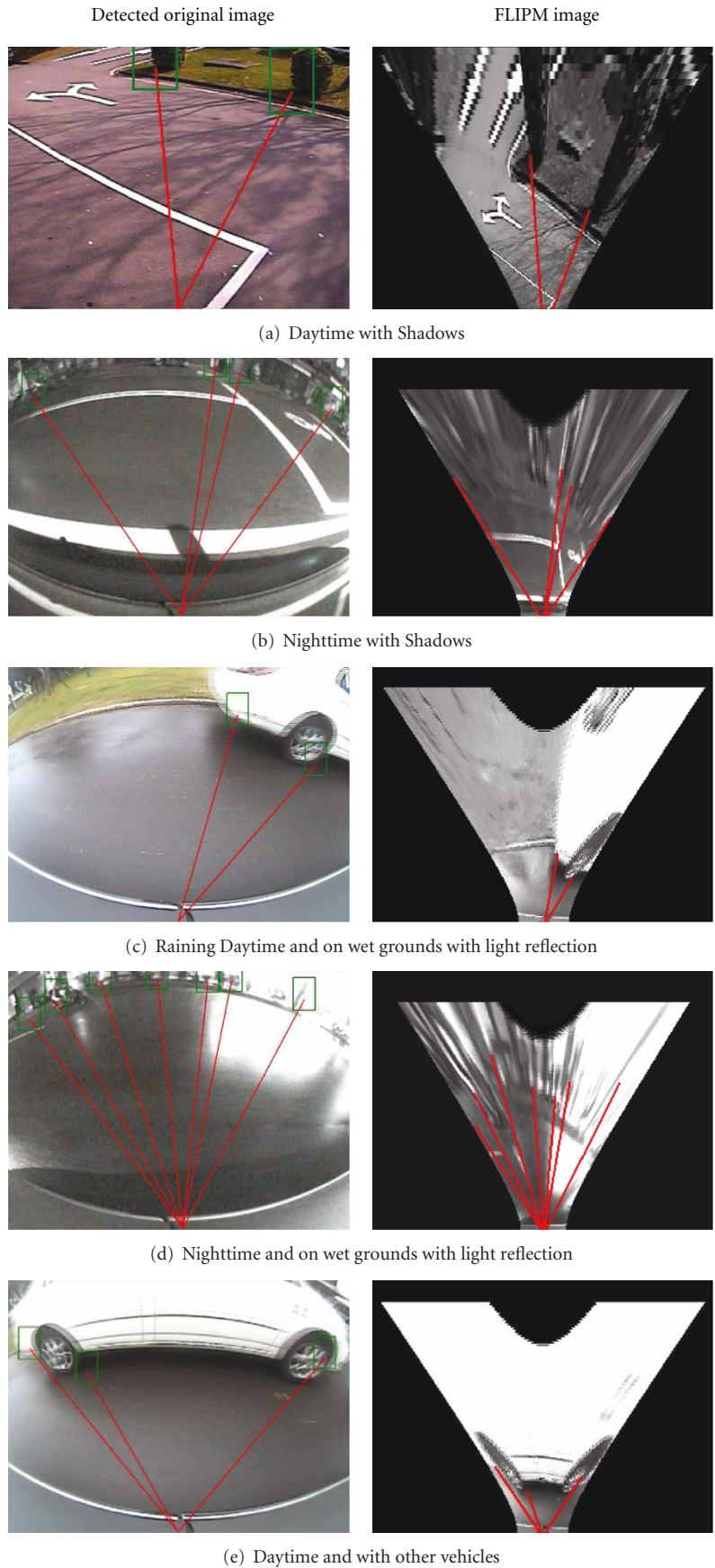
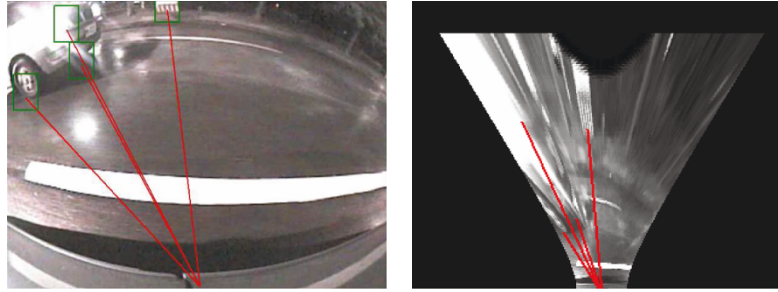
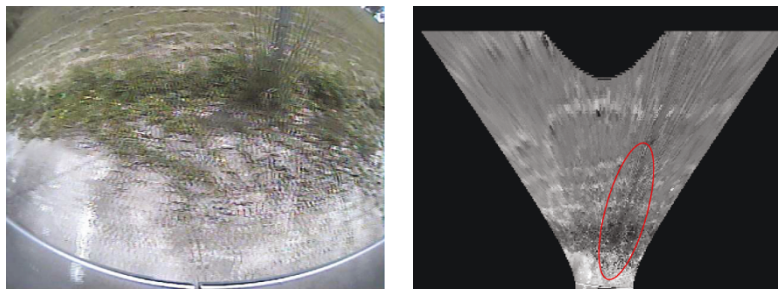


FIGURE 28: Continued.

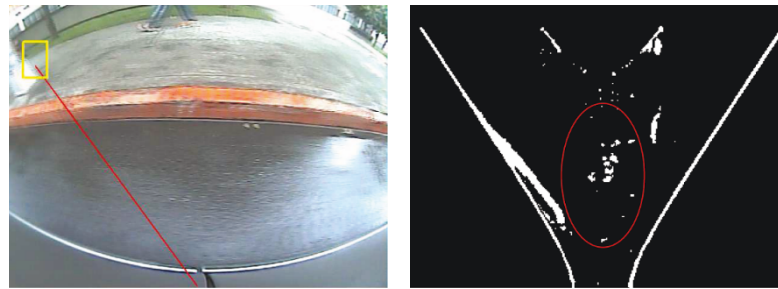


(f) Nighttime and with other vehicles

FIGURE 28: The results in different environments with heavy noise. (a) In the daytime with many shadow effect. (b) In the nighttime with self shadow projected by several different direction road lights. (c) In the raining daytime on wet grounds with light reflection. (d) In the raining night time on wet grounds with strong light reflection. (e) Daytime with other vehicles. (f) Nighttime with other vehicles.



(a)



(b)

FIGURE 29: Examples of erroneous detected results in our system.

TABLE 1: The specifications of our working platform.

CPU	Intel Core Duo T2050 1.6 GHz
Memory	1 GB DDR2 RAM
Programming tool	Borland C++ Builder 6.0
Operation system	Microsoft Windows XP
Video resolution	640 × 480
Camera frame rate	30 fps

TABLE 2: The runtime in each processing step in Figure 1.

Processing Function	Runtime/ frame (ms)
FLIPM (with the IPM remapped position table)	2.793
Input image, transfer gray-level image and setting dynamic array memory	31.98
Preprocessing	8.847
Segment searching and Polar Histogram	6.515
Histogram and Postprocessing	0.253
Obstacle tracking and extraction	0.347
Display	13.725
Total	64.46

height as shown in Figure 20 To avoid disorders of frames, we would only detect the objects whose heights are more than a threshold and whose edges are quasivertical. The

objects such as sidewalks, small balls and so on were excluded in our detection system. The experimental environments would also be constrained to the brighter backgrounds and

TABLE 3: Comparisons of different obstacle algorithms.

Compared Methods	Runtime (ms)	Sensor Type	Moving compensation	Field of View
Our approach	64.46 (CPU 1.6 GHz)	Fisheye Camera	YES	125°
Bertozzi and Broggi [2]	100 (FPGA based)	Stereo Camera	NO	28°
Ji [3]	66.7 (CPU 3 GHz)	Single Camera	NO	34°
Cerri and Grisleri [4]	950 (CPU 2.8 GHz)	Single Camera	YES	Normal Lens
Kyo et al. [14]	65 (multipile IMAP-VISION board)	Single Camera	NO	Normal Lens
Yang et al. [17]	50 (CPU 3.6 GHz)	Single Camera	YES	Normal Lens
Ma et al. [18] (pedestrian candidate detection module)	16 (CPU 3.6 GHz)	Single Camera	NO	48.8°

the speed of vehicles should be under a reasonable limit so that the objects between frames would not change too drastically.

Figure 21 showed our program interface where the block (a) gave the input frame in which the line and rectangle were used to indicate the position of obstacles, the block (b) had the relative information of videos, the block (c) contained the extrinsic and intrinsic parameters of the camera and specifications of look-up table for FLIPM, the block (d) displayed the image used in our obstacle detection algorithm, and the block (e) showed the obtained polar histogram. In the block (d) of Figure 21, the upper image was the remapped image, the middle one represented the judgment image for the static situations, and the lower one showed the input image for the feature-segment searching stage.

6.3. Results in Various Environments. As Figure 22 showed, we could accurately detect various kinds of obstacles with quasivertical edges by using our FLIPM methods and obstacle detection algorithm.

Our tracking process was carried out by iteratively checking the displacement and angular shift in the image, and we also demonstrated the results of the tracking process in successive frames as given in Figures 23 and 24. According to the FLIPM method, the 3D world coordinate value could be estimated from the remapped image. In other words, when we detected the obstacle in the remapped image, we could also estimate the position information. We hence set up an obstacle warning system on the lateral and rear of the vehicle to give a warning signal when the detected obstacles were over close to our vehicle. We showed the results of the obstacle warning system In Figures 25, 26, and 27 where the rectangles in the upper images and the lines in the below images indicated “the position of obstacles” and “the distance and direction between the vertical projection point of camera and the detected obstacle”, respectively.

In Figure 27, there were results of obstacle warning with moving objects. We had two simulation situations. One was that a pedestrian was walking form a parking region to another side while the demonstrated vehicle was leaving the parking region. The other situation contained a corner where a coming vehicle, the static obstacle and

clear planar markings existed. Here, we presented two issues in the moving obstacle detection and compensation estimation with rotation angles. We could get the results from distributions of the right polar histograms, localize each nonplanar obstacle by dominant peaks, and filter the lane markings by trapezoid distributions in the original polar histograms.

In Figure 28, there existed some simulated environments with heavy noises, such as shadows, light reflection, and light refraction from wet roads. In case (a) where there existed many shadows of trees on the grounds, we could obtain the remapping image as shown in the right one by the FLIPM transformation. By our approach, the shadows could be filtered out by the polar histogram if its accumulations on each angle were small and its shapes were in trapezoid. For case (b) where there were two road lights at the front and rear of our vehicle in the nighttime, we would find two different shadows on the ground. One was not clearly recognized from far light projection and the other was obvious due to near light projection. Our proposed method, however, could take advantage of FLIPM effects to remove the shadowing effect no matter how serious the illuminating conditions might be. Our compensation estimation could shift the new frame to the adaptive position to gain the minimum candidate pixels of obstacles. For case (c) and case (d), we demonstrated our results to be reliable and satisfactory in the raining conditions in the daytime and nighttime. In the nighttime, noises from light reflection were much more serious than those in the daytime, and our approach could successfully avoid misrecognizing them as obstacles in the fixed illumination condition. As for case (e) and case (f), we showed the experimental results in the common situations which simulated the roads in the daytime and nighttime. As a result of advantages of the fisheye camera in a wider angle of view, our obstacle detection algorithm could detect the range of the field of view up to 125 degrees which was much wider than other algorithms by common lens.

6.4. Discussions. Although the performance of our obstacle detection system based on FLIPM method was quite satisfactory, there have existed some disturbance factors as shown below. In Figure 29(a), the street light in the remapped image

were too unapparent to be detected because its texture was similar to that of the grassland behind it. In addition, the completeness of obstacle shape in the profile or temporal FLIPM difference image would be critical for the following obstacle detection process. Figure 29(b) showed the broken shape of obstacles in the temporal FLIPM difference image, which could lead to the erroneous detected results.

7. Conclusions

In this paper, we propose a complete and novel structure for the obstacle detection system. This brand-new structure includes three major parts, FLIPM algorithm, feature segment searching, and obstacle detection. With our modified normal lens IPM equations, the vertical/horizontal straight lines in the original image will be projected to a straight line whose prolongation will pass the vertical/horizontal projection point of camera in the remapped image. The resultant phenomenon has two advantages in removing planar objects and detecting obstacles. One is to give more information in predicting the compensation quantification between difference frames, which helps us to remove the planar objects such like shadow, water, lane line, and so on. The other one is to reinforce the feature points of obstacles, which efficiently reduces the computational loading in searching obstacles. Besides, we consider the fisheye lens distortion effect and provide a high efficient and all-direction feature searching method on polar histogram for both of the static and dynamic environments. We use the polar histogram to find the position and length of feature segments by referring to the edge and temporal difference images. We also present the histogram post-processing to exclude the planar lane markings and noises. Finally, all the experimental results of our proposed system show the satisfactory performance and provide the accurate detection rate. In the future, our obstacle detection system can be integrated into the driving assistance and safety system, including vehicle collision-free development, warning system, and lane departure warning system. Furthermore, we will work on different shapes of obstacles for those without quasivertical edges and speed up our detection system for more real-time applications.

References

- [1] M. Bertozzi, A. Broggi, and A. Fascioli, "Stereo inverse perspective mapping: theory and applications," *Image and Vision Computing*, vol. 16, no. 8, pp. 585–590, 1998.
- [2] M. Bertozzi and A. Broggi, "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62–81, 1998.
- [3] W.-L. Ji, *A CCD-based intelligent driver assistance system-based on lane and vehicle tracking*, Ph.D. thesis, National Cheng Kung University, Tainan, Taiwan, 2005.
- [4] P. Cerri and P. Grisleri, "Free space detection on highways using time correlation between stabilized sub-pixel precision IPM images," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, pp. 2223–2228, Barcelona, Spain, April 2005.
- [5] A. M. Muad, A. Hussain, S. A. Samad, M. M. Mustafa, and B. Y. Majlis, "Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system," in *Proceedings of the IEEE Region 10 Conference on Analog and Digital Techniques in Electrical Engineering (TENCON '04)*, vol. 1, pp. 207–210, Chiang Mai, Thailand, November 2004.
- [6] S. Tan, J. Dale, A. Anderson, and A. Johnston, "Inverse perspective mapping and optic flow: a calibration method and a quantitative analysis," *Image and Vision Computing*, vol. 24, no. 2, pp. 153–165, 2006.
- [7] G. Y. Jiang, T. Y. Choi, S. K. Hong, J. W. Bae, and B. S. Song, "Lane and obstacle detection based on fast inverse perspective mapping algorithm," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 2969–2974, Nashville, Tenn, USA, October 2000.
- [8] M. Nieto, L. Salgado, F. Jaureguizar, and J. Cabrera, "Stabilization of inverse perspective mapping images based on robust vanishing point estimation," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 315–320, Istanbul, Turkey, June 2007.
- [9] J.-H. Lai, *Development of an exploration system for a vision-guided mobile robot in an unknown indoor environment*, M.S. thesis, St. John's University, 2006.
- [10] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, and W. Von Seelen, "Walking pedestrian recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 155–162, 2000.
- [11] M. Bertozzi, E. Binelli, A. Broggi, and M. D. Rose, "Stereo Vision-based approaches for Pedestrian Detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 2005, p. 16, San Diego, Calif, USA, 2005.
- [12] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using optical sensors: a review," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (ITSC '04)*, pp. 585–590, October 2004.
- [13] S. Kyo, T. Koga, K. Sakurai, and S. Okazaki, "Robust vehicle detecting and tracking system for wet weather conditions using the IMAP-VISION image processing board," in *Proceedings of the IEEE/IEE/JSAI International Conference on Intelligent Transportation Systems*, pp. 423–428, Tokyo, Japan, October 1999.
- [14] S. Denasi and G. Quaglia, "Obstacle detection using a deformable model of vehicles," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '01)*, pp. 145–150, Tokyo, Japan, 2001.
- [15] W. Krueger, W. Enkelmann, and S. Roessle, "Real-time estimation and tracking of optical flow vectors for obstacle detection," in *Proceedings of the Intelligent Vehicles Symposium*, pp. 304–309, Detroit, Mich, USA, September 1995.
- [16] C. H. Q. Forster and C. Tozzi, "Towards 3D reconstruction of endoscope images using shape from shading," in *Proceedings of the 13th Brazilian Symposium on Computer Graphics and Image Processing*, pp. 90–96, 2000.
- [17] C. Yang, H. Hongo, and S. Tanimoto, "A new approach for in-vehicle camera obstacle detection by ground movement compensation," in *Processing of the 11th IEEE Conference on Intelligent Transportation Systems (ITSC '08)*, pp. 151–156, Beijing, China, October 2008.
- [18] G. Ma, D. Muller, S.-B. Park, S. Muller-Schneiders, and A. Kummert, "Pedestrian detection using a single-monochrome camera," *IET Intelligent Transport Systems*, vol. 3, no. 1, pp. 42–56, 2009.

- [19] S. Zhang and K. S. Fu, "A thinning algorithm for discrete binary images," in *Proceedings of the International Conference on Computers and Application*, pp. 879–886, Beijing, China, 1984.
- [20] Q.-T. Luong, J. Weber, D. Koller, and J. Malik, "An integrated stereo-based approach to automatic vehicle guidance," in *Proceedings of the 5th International Conference on Computer Vision*, pp. 52–57, Cambridge, Mass, USA, June 1995.
- [21] G. Ma, S.-B. Park, A. Ioffe, S. Müller-Schneiders, and A. Kummert, "A real time object detection approach applied to reliable pedestrian detection," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 755–760, Istanbul, Turkey, June 2007.