



2950 Niles Road, St. Joseph, MI 49085-9659, USA
269.429.0300 fax 269.429.3852 hq@asabe.org www.asabe.org



An ASABE – CSBE/ASABE Joint
Meeting Presentation

Paper Number: 1906189

Object Tracking and Collision Avoidance Using Particle Filter and Vector Field Histogram Methods

Ta-Te Lin, Professor

Dept. of Bio-Industrial Mechatronics Engineering, National Taiwan University, No. 1, Sec. 4,
Roosevelt Rd., Taipei 106, Taiwan, R.O.C., m456@ntu.edu.tw

Li-Kang Weng, M.S. Graduate Student

Dept. of Bio-Industrial Mechatronics Engineering, National Taiwan University, No. 1, Sec. 4,
Roosevelt Rd., Taipei 106, Taiwan, R.O.C., r02631009@ntu.edu.tw

An-Chih Tsai, Ph.D. Candidate

Dept. of Bio-Industrial Mechatronics Engineering, National Taiwan University, No. 1, Sec. 4,
Roosevelt Rd., Taipei 106, Taiwan, R.O.C., d96631003@ntu.edu.tw

Written for presentation at the

2014 ASABE and CSBE/SCGAB Annual International Meeting

Sponsored by ASABE

Montreal, Quebec Canada

July 13 – 16, 2014

Abstract. Object tracking and collision avoidance are important topics in various applications of autonomous vehicles and robotics. In previous studies, we have developed a real-time stereo vision system equipped with efficient algorithms for obstacle detection and recognition. To detect and locate obstacles, the estimated three-dimensional information of each pixel is projected onto a non-linear top-view map, and the blob segmentation is used to find the obstacle candidates in the top-view map. Detected obstacles can be further recognized based on their geometrical features and histogram of oriented gradient (HOG) feature using support vector machine method. The 3D information of obstacles and the relationship of obstacles are subsequently applied to track obstacle paths and to estimate the motion model. In this study, particle filter approach was applied to offer stable tracking of detected obstacles in different frames. This method was compared with Kalman filter approach under various experimental conditions. Finally, by combining various information including color and 3D information as well as motion models, the collision avoidance algorithm was developed based on the vector field histogram approach. The output of the collision avoidance algorithm includes the direction and predicted path of the vehicle and warning signals for the driver. The performance of the collision avoidance algorithm was analyzed and compared with the method based on the A-star path-finding algorithm. Experiments were carried out under different scenarios of driving conditions and obstacle states to compare the overall performance of the system. The proposed object tracking and collision avoidance method will help in enhancing the safety of operations of agricultural vehicles.

The authors are solely responsible for the content of this meeting presentation. The presentation does not necessarily reflect the official position of the American Society of Agricultural and Biological Engineers (ASABE), and its printing and distribution does not constitute an endorsement of views which may be expressed. Meeting presentations are not subject to the formal peer review process by ASABE editorial committees; therefore, they are not to be presented as refereed publications. Citation of this work should state that it is from an ASABE meeting paper. EXAMPLE: Author's Last Name, Initials. 2014. Title of Presentation. ASABE Paper No. --. St. Joseph, Mich.: ASABE. For information about securing permission to reprint or reproduce a meeting presentation, please contact ASABE at rutter@asabe.org or 269-932-7004 (2950 Niles Road, St. Joseph, MI 49085-9659 USA).

Keywords. stereoscopic, automatic guidance, off road vehicles, image processing, robotics.

Introduction

Object tracking and collision avoidance are the leading research areas in tracking and autonomous navigation. Sensors including radar, laser rangefinder, ultrasonic, and infrared sensors are mounted on the vehicle for surrounding environment sensing. However, these sensors only provide depth information comparing to cameras. On the other hand, stereo vision system can provide not only the depth information but also the color information. Based on the integration of this information, more information about obstacles can be provided to drivers for obstacle recognition and tracking. According to color and geometrical information of detected obstacles, the obstacle recognition and tracking are achievable even in complex applications. Based on the motion status of tracked obstacles, the collision avoidance method can warn drivers and suggest next action to take. In agricultural applications, the fields are usually too spacious to be managed by manpower. Therefore, farmers often use agricultural machineries including tractors or other off-road vehicles to carry out agricultural tasks. While driving agricultural automatic vehicles in the field, paying attention to the surrounding working area is important, especially the blind spots where higher probability of accidents would take place. Thus, the obstacle tracking, recognition and collision avoidance methods can be applied to enhance the safety. By these methods, the locations of obstacles can be detected and provided to drivers during operation.

Stereo vision technology has been applied in obstacle detection, recognition and tracking in recent years. Foggia et al. (2005) proposed a real-time stereo vision based system for moving objects and obstacle detection (MOOD). In this system, moving objects were segmented out by the optical flow method. The obstacles were detected from the disparity image by applying blob method and the method proposed by Pantilie et al. (2010). Pantilie et al. (2010) used the fusion of the stereo vision and optical flow in a depth-adaptive occupied grid framework to accurately detect moving obstacles. Obstacle tracking was then applied after the obstacles were detected and recognized. Yilmaz et al. (2006) synthesized the tracked objects and categorize them into three types: points, silhouette and kernel types. In this research, the point tracking type was applied. In addition, two widely-used tracking method, Kalman filter and particle filter were applied and compared. Kalman filter is a linear system, which uses location and velocity measurements of objects to filter noise out and estimate the locations of objects. Particle filter is a nonlinear system which applies sets of particles to characterize object features for the object position prediction. In real world, the dynamic systems are usually random distributed models; hence, the linear systems are not always applicable to describe the whole systems. In recent years, building a nonlinear and non-Gaussian dynamic system is more important as the nonlinear systems are more realistic to fit the real-world models. In a collision avoidance method, the motion planning is a critical part where A* search algorithm and vector field histogram (VFH) algorithm were applied in this research. A* algorithm was the best-first search algorithm proposed by Hart et al. in 1968. The path was planned where only the destination is needed by the map with surrounding information. VFH algorithm proposed by Borenstein et al. (1991) was applied for robotics navigation. The dynamic environment information was represented by one-dimensional polar histogram to save computation time. The path planned by the VFH is smoother by considering each position on the path.

This research aims to build an obstacle tracking and collision avoidance system utilizing the stereo vision system built in the previous study. In addition, the system should be capable of obstacle recognition and obstacle tracking for agricultural vehicles. With this application, the collision between obstacles and vehicles can be avoided. The safety of vehicles thus can be increased by the collision avoidance method. Kalman filter and particle filter were applied for obstacle tracking and compared for their performance. The A* search algorithm and VFH algorithm were applied for planning the path in the collision avoidance method.

System Configuration

The stereo vision system is composed of two cameras, the video acquisition device and the flexible mechanism. The system utilizes two 16 mm cameras mounted on the precision mechanism with resolution 640 x 480 pixels and 19.8° FOV angle as shown in Figure 1. Video acquisition device catches images at 30 frames per second (fps). According to the principle of stereo vision and simplifying the corresponding matching, optical axes of the dual cameras have to keep in parallel, and cameras have to be in the same horizontal axis. Hence, the precision mechanism was designed to fit these criteria. The mechanism fixes the cameras' lens on the bar to avoid the cameras' vibration caused by vehicle's movement. In addition, the baseline, the distance between the dual cameras, of the mechanism is adjustable to cope with different applications. The default baseline length is 15 cm in this research. With the precision mechanism, the parameters of cameras are held constant

after the cameras have been calibrated.



Figure 1. The flexible binocular stereo vision system with adjustable baseline length.

The system flowchart is shown in Figure 2. The images are acquired from cameras and 3D information is estimated from the pixel disparity values. The obstacles in front of the vehicle are detected as their images are segmented out using 3D information. They are projected onto the non-linear top-view for later process. Applying stereo vision to detect obstacles not only defines the locations of obstacles, but also obtains their color information. According to this information, the type of obstacles can be recognized by extracting their features and using the machine learning methods. Moreover, the velocity of obstacles can be measured via the estimated 3D information. Kalman filter and Particle filter are then applied to track the obstacles labeled in different frames using pattern matching methods; hence, the motion status of obstacles can be estimated. By the estimated 3D information, the obstacle avoidance system can monitor the frontal environment of vehicle and also guide the driver to avoid collision with the obstacle.

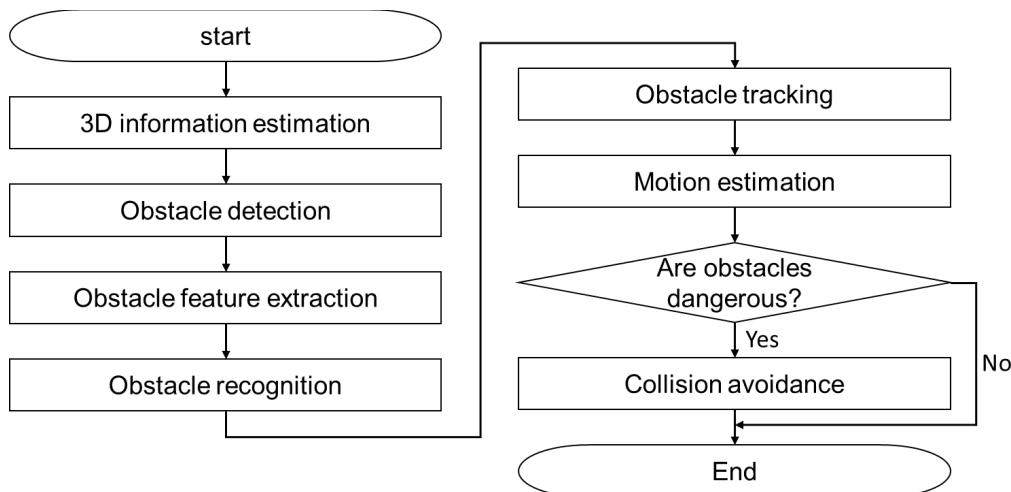


Figure 2. The system flowchart.

Obstacle Detection and Recognition

3D Information Estimation

From a pair of stereo images, 3D information of each pixel in image plane can be estimated using epipolar geometry and corresponding matching method. Epipolar geometry represents the geometry relationship between the object's location in the real-world and in the image plane. The locations of the objects can be estimated when pixels' disparity values are given (Lin et al., 2013). The disparity value represents the difference of the object's pixel value in left and right image planes. Thus, defining the correspondence of the object in two image planes is necessary to calculate the disparity value by a corresponding matching method. For the corresponding matching method, the semi-global block matching (SGBM) algorithm is implemented to calculate the disparity value of each pixel in the image using OpenCV (Intel®) library, the open source library for computer vision.

Using top-view image projection, the frontal environment is easier to be understood than displaying 3D information directly. The obstacles including vehicles, pedestrians and bikes are assumed to occupy certain volume and be perpendicular to the ground. The objects obtained from the disparity image and satisfied the predefined 3D constraints are projected onto the top-view image (occupied grid), which is proposed by Pocol et al. (2008). The top-view image is formed by two-dimensional histogram which is trapezoid. The size of each

grid represents the practical size in real-world as shown in Figure 3.

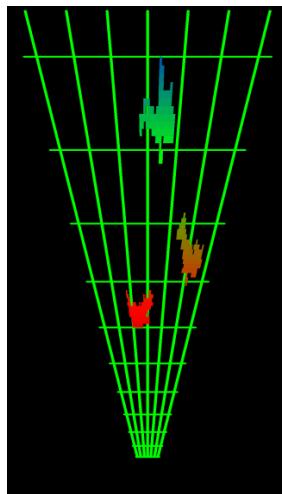


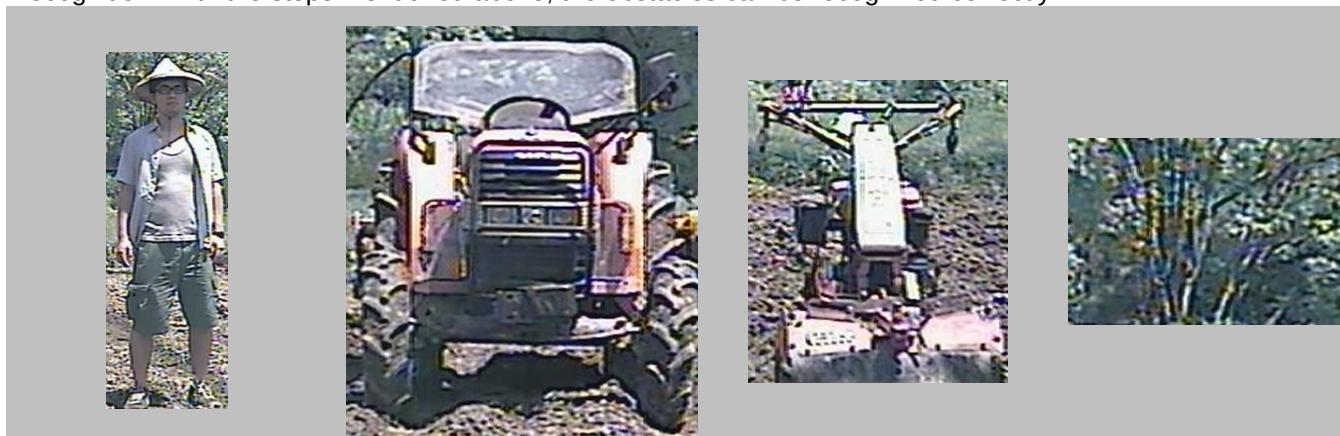
Figure 3. A typical top-view image (occupied grid) for obstacle detection.

Obstacle Detection

The obstacles are detected in the top-view image using the blob method and 3D constraints. Each pixel in the image plane is projected from the disparity image onto each grid in top-view image using the 3D information of each pixel. The grids which contained more than 300 pixels are considered as obstacle candidates. Then, the blob method is applied to the top-view image to connect the grids and label them as obstacle candidates. Though the blob method is applied, there are still some small obstacles due to corresponding matching errors. The height constraint is applied to filter out small obstacles whose heights are less than 30 cm. Sometimes, some obstacles are detected as several small parts due to the lack of texture information. For this reason, the merge method, a post-processing of obstacle detection, is applied to merge the small parts, whose geometrical information is similar, into an obstacle image to overcome this drawback.

Feature Extraction and Obstacle Recognition

Obstacle recognition is an essential application to provide drivers obstacle details after they are detected. Figure 4 shows four types of obstacles considered in this study. They are named as human (pedestrian), large agricultural vehicle (tractor), small agricultural vehicle (cultivator) and unknown obstacle. The obstacle recognition method is based on a hierarchical decision tree. In the first level of the tree, the obstacles were divided into two types whether it is a human or not. The histogram of oriented gradients (HOG) descriptor proposed by Dalal et al. (2005) is applied with its outstanding performance for human detection based on the magnitude and orientation of obstacle's gradient. The model of recognizing human and other obstacles is built from HOG features and support vector machine (SVM). In the second level, the obstacles recognized as non-human objects including large agricultural vehicles, small agricultural vehicles and unknown obstacles are recognized using geometrical features and linear discriminant analysis (LDA). The feature vectors composing the estimated distance and the aspect ratio of the obstacle image are mapped to the LDA space for easier recognition. With the steps mentioned above, the obstacles can be recognized correctly.



(a) (b) (c) (d)

Figure 4. The defined obstacles are (a) human, (b) tractor, (c) cultivator, and (d) unknown obstacle.

Obstacle Tracking and Motion Estimation

Obstacle tracking, providing the motion state of obstacles, is important and helpful for surveillance and collision avoidance. To track obstacles in adjacent frames, the features of detected obstacles are compared between adjacent frames and the most similar one is registered as the same obstacle. The obstacle feature is crucial to the tracking method. If the obstacle feature is not significant to represent the obstacle, the error caused by mismatching would result in unsatisfactory tracking. The tracking method based on geometric shape of obstacles is used due to its relative low computation cost. There are two steps in the obstacle tracking method: feature extraction and feature matching. In this research, the histograms of average distance and color on HSV color space are combined as the matching criteria. The Bhattacharyya distance, the method of measuring the similarity between two histograms, is applied to the feature matching process. The range of Bhattacharyya distance is from 0 to 1. Thus, the obstacle in different frames is more similar if the Bhattacharyya distance is close to 0, and vice versa. When the Bhattacharyya distance of obstacles is lower than a predefined threshold, the obstacle in this frame is considered as the same one in the previous frame. Caused by the change of the obstacle's velocity, Bhattacharyya distance may be influenced and cause a failure in tracking. Hence, adaptive Bhattacharyya distance method is applied to reduce this kind of tracking failures.

Once the relationships of obstacles in different frames are acquired, the motion states of obstacles such as locations and velocities are helpful to represent the dynamic environment at front. Figure 5 shows that the obstacle's traveling state is calculated and classified into two types: approaching and leaving. The real location of obstacle is calculated by the locations in last five frames. The obstacle tracking method proposed by Lin et al. (2013) is applied.

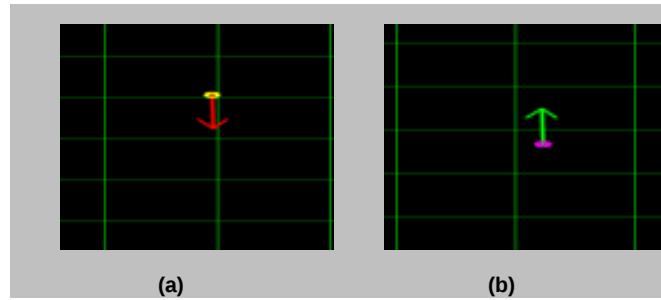


Figure 5. Traveling state of obstacle: (a) approaching, and (b) leaving.

Kalman Filter

Kalman filter proposed by Kalman (1960) is a recursive filter to solve discrete linear problems. It is an algorithm which makes use of imprecise data optimally in a linear system with noises to continuously update the best estimate of current state. There are two main applications: filtering and prediction. In particular, the latter one is widely-used in autonomous navigation field that needs prediction and position modification. In this study, the motion of each obstacle is assumed to be linear. Kalman filter is applied to estimate obstacle's motion and predict its possible position in successive frames.

Generally, Kalman filter can be inaccurate due to the inaccurate initial guess at the beginning of prediction. For the purpose of reducing the error, this research proposes to initialize the Kalman filter via the reliable velocity estimation approach mentioned above. The position at the time can be estimated when the velocity is obtained. It is helpful to reduce the error.

Particle Filter

Particle filter (Arulampalam et al., 2002) is a tracking method implementing by Bayesian recursion equations. This method is also called sequential Monte Carlo method which combines Monte Carlo method with a statistical method. It uses a grid-based approach. A set of weighted particles is constructed to describe the posterior probability density function (pdf) of state-space. Using particle filter approach, features are not restricted to be any form used for object tracking in cluttered environment. This method consists of two stages: prediction and update. Prediction stage predicts the state pdf from time to time. The prediction pdf is updated by refreshing the state pdf and the latest measurement. In this research, particle filter is applied to predict the

motion state of the obstacle. The objects' features are represented by a set of randomly distributed particles. To find the object, the similarity of the histogram is compared. X_k , the pdf of the object at time t for the nonlinear system, is written as

$$p(X_t | Z_{1:t}) \approx \sum_{i=1}^N w_t^i \delta(X_t - X_t^i), \quad (1)$$

where $Z_{1:t}$ is the prediction value from time 1 to t , N is the amount of particles, w_t^i is the weight of particle i in time t and X_t^i is the pdf of particles at time t and $\delta(\cdot)$ is the Dirac measure. w_t^i , the weight of each particle at time t , is related to the previous weight w_{t-1}^i and the current pdf $p(Z_t | X_t^i)$. The weight equation is given by

$$w_t^i \propto w_{t-1}^i \frac{p(Z_t | X_t^i) p(X_t^i | X_{t-1}^i)}{q(X_t^i | X_{t-1}^i, Z_t)}, \quad (2)$$

where q represents the importance density function.

The flowchart of the particle filter approach is shown in Figure 6. Particles are sprayed in the Gaussian distribution first. The feature of each particle is compared with the target one. In this research, the feature is the histogram calculated from the image in HSV color space. The similarities of histograms are summed and equalized. The new prediction of object is estimated by weighted average calculation. Before the next calculation, the new weights of particles are updated by the current weight. If the current weight is higher, the object has a higher probability to be at this location. The process is repeated until the objects are disappeared.

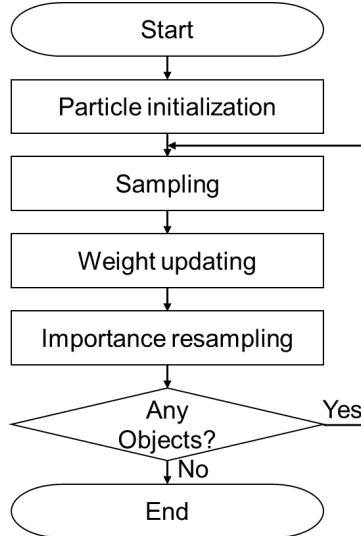


Figure 6. Flowchart of the particle filter approach.

Collision Avoidance Methods

The dynamic environment is obtained for drivers to realize the frontal situation. Obstacles ahead of the driver's vehicle are always dynamic. In order to estimate the obstacle motion, accumulating and observing obstacles' historical movement is one of solutions. Then the orientation, location and velocity of obstacles can be predicted by referring to prior movement information. When the obstacle motion changes rapidly, such as emergency brakes or unexpected jump in appearance, the motion planning algorithm is applied to prevent the vehicle from accidents and guide the vehicle. The collision avoidance method consists of obstacle prediction and motion planning. In this research, A* search algorithm and VFH algorithm are applied and compared to plan the path.

A* Search Algorithm

Hart (1968) proposed the best first search algorithm for motion planning. A* search algorithm is widely-used in autonomous navigation for its good performance by applying heuristics. The path is calculated using the costs of nodes on pre-established grid map until finding the one with lowest cost given the starting and end nodes.

Figure 7 shows the calculation process of A* search algorithm. Each grid represents a node. The backslash, slash, solid ones are the starting node, predefined end node and the obstacle respectively. The algorithm starts with the starting node. At each step of the calculation, the costs of adjacent nodes are calculated and recorded in the open set to find the lowest one. The lowest one is set to be the next node recorded in the close set.

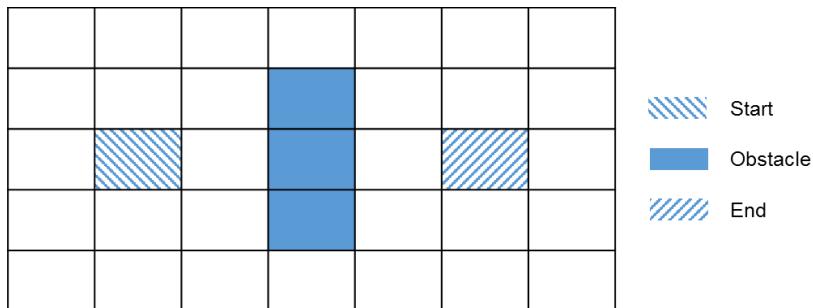


Figure 7. Grid map of motion planning.

The cost function calculated in open set is shown below,

$$F(n) = G(n) + H(n) \quad , (3)$$

where $F(n)$ reflects the current cost. $G(n)$ is the movement cost from the starting node to the current node which assumes the costs to be 10, 10, and 14 for horizontal, vertical movements and diagonal one, respectively. $H(n)$ is the heuristic evaluation function of the cost from the current node to the end node using the Manhattan method. The Manhattan method takes 8-adjacent movement cost without considering obstacles. Figure 8 shows an example of how A* search algorithm works. Starting from the starting node (the thick block), the costs of adjacent nodes are calculated and recorded in the open set. The right node is the lowest cost node

with $F(n)$ equals to 40 for next calculation and the center node is recorded as the parent node. In the next calculation, there is at the situation where the upper and bottom nodes cost the same. In this situation, both nodes can be selected resulting in the different paths but the costs will stay the same. The bottom node is chosen and the adjacent nodes are recorded in the open set, too. However, the cost is not the lowest in the open set. By comparing the $H(n)$ values between nodes, it is better to choose the bottom right node in the first calculation rather than taking two steps to get the same node and then the center node in this calculation is not recorded.

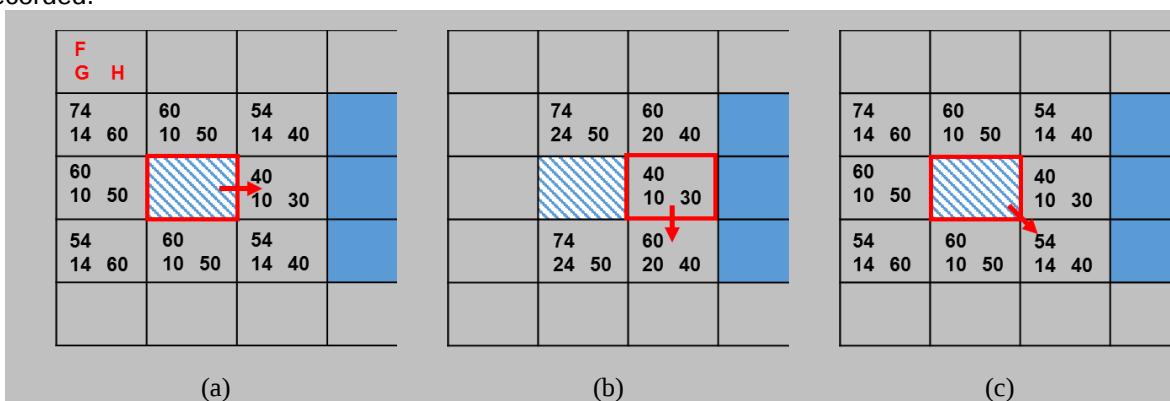


Figure 8. Costs of nodes in (a) first, (b) second calculation and (c) the result of comparing cost.

In this process, the cost of each grid is changed due to the updating parent node. By tracing the parent node,

the path (the solid line) is found when the end node is reached as shown in Figure 9.

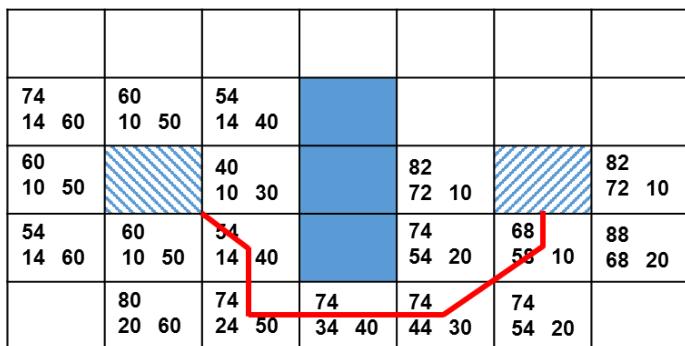


Figure 9. The result of A* search algorithm.

Vector Field Histogram Algorithm

VFH algorithm is a fast, all-rounded and best path search algorithm applied in robotics navigation. It is initially intended to reduce the memory usage and to accelerate the computing speed. For this reason, the information is converted from two-dimensional into one-dimensional. The vehicle information is considered to simulate the situation more realistic and to compute the safe spaces. The path can be planned smoother by the safest space in each frame.

Figure 10 shows how the path is planned. Firstly, the information around the vehicle is required. The danger level of obstacles, that will affect vehicle, is evaluated using this information. The free space for vehicle to move is also calculated. With the safe spaces, the path is planned by the most appropriate one selected in the movement direction. The graphical user interface (GUI) displays the guide sign for collision avoidance. If the obstacles are inescapable, the algorithm will warn the drivers to brake and display on the GUI.

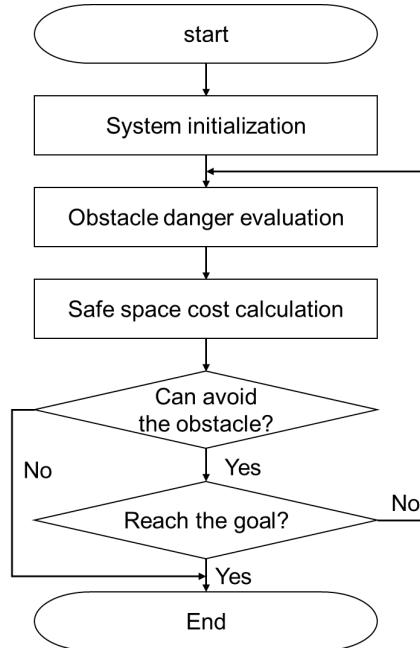


Figure 10. Flowchart of VFH algorithm.

System Initialization

For obstacle avoidance, it is essential to take the information, such as obstacles' locations, vehicle length, rotation angle and desired path into consideration. The desired path is the reference path for the algorithm. In this research, the desired path is the straight line at the middle of the map. The vehicle follows the path if there is no obstacle on the path. Otherwise, the vehicle detours around the obstacles occupying the desired path. The main concept of VFH is to estimate an appropriate path referring the desired path considering the vehicle's geometrical information. Figure 11 shows the environment map in front of the vehicle. The blue circle

represents the vehicle center point (VCP). The black blocks are the obstacles and the gray line is the desired path.

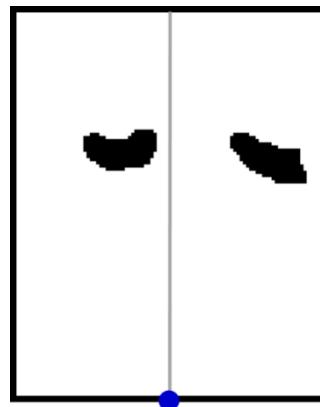


Figure 11. Environment map.

Obstacle's Weight Assignment

In the main concept of VFH, if there is no obstacle on the desired path, the vehicle will follow the desired path. If the obstacle detected and locates on the path, the estimated path will guide the vehicle to detour obstacles. When the vehicle is far away from the desired path to avoid the obstacle, the estimated path will still pull the vehicle back to the desired path. Figure 12 shows how VFH algorithm is applied to estimate the path. In Figure 12, two trajectories are illustrated. The black line reflect actual vehicle's trajectory and red dash line is the desired path. Referring to the desired path, the estimated path will pull the vehicle back to the desired path.

The closest point (P_a) is the point VCP projecting on the desired path. In Eq. (5), \vec{V} is the unit vector from P_a to P_{a+1} which is adjacent to P_a on the desired path. In addition, an avoidance factor, F , is used to define how far the vehicle is expected to move. In Eq. (6), the new point, P_{new} , can be calculated with estimated P_a and F . Then, the radius of the circle (blue circle in Figure 12) is defined and denoted as Eq. (7). In the actual environment, there are different types of obstacles on the map. Some of them are neglected because they are too far to obstruct the vehicle immediately. The obstacles whose distance to vehicle is within the predefined range are all considered in VFH algorithm. In contrast, the obstacles out of the circle are not considered. Then, the weight of the obstacle is assigned based on the distance from VCP to the obstacle. Obstacle's weight reflects how dangerous the obstacle is to the vehicle. The greater the danger is, the larger the obstacle's weight is.

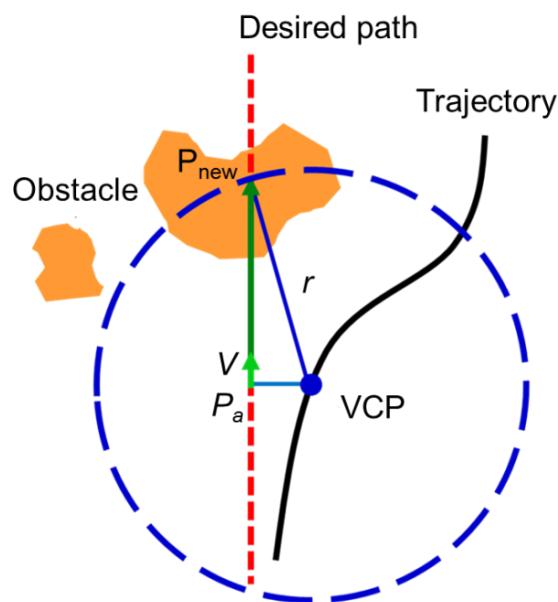


Figure 12. Obstacle danger evaluation diagram.

$$P_a = \operatorname{argmin} \|VCP - DesiredPath\| \quad (4)$$

$$V = P_{a+1} - P_a \quad (5)$$

$$P_{new} = P_a + V \cdot F \quad (6)$$

$$r = \|VCP - P_{new}\| \quad (7)$$

To assign the obstacle's weight, the all-round environmental information has been separated into 72 sectors clockwise as shown in Figure 13. In this research, sectors 36 to 72 are considered. The vehicle is assumed that it cannot reverse and drift while driving. The weight of obstacles are calculated by the magnitude equation to reflect the possible danger to the vehicle,

$$W_i = C^2(a - b \cdot \|VCP - Obstacle\|), \quad (8)$$

$$a = 1 + b \cdot r, \quad (9)$$

where a and b are positive constants and C is the certainty value of obstacle. The weight is higher when the obstacle is closer to the vehicle. In addition, the position of obstacle is converted into polar coordinates. The polar obstacle density (POD) is a type of histogram used to calculate the weight of obstacle based on the angle of obstacle' polar coordination (Eq. 8). The smoothing processing (Eq. (11)) below is implemented to dilate the spaces of obstacles to ensure obstacles are completely considered.

$$h_k = \sum_{i \in C} m_i \quad (10)$$

$$S_k = \frac{h_{k-l} + 2h_{k-l+1} + \dots + lh_k + \dots + 2h_{k+l-1} + h_{k+l}}{2l-1} \quad (11)$$

With POD, the safe spaces where the weights are zero are founded. Figure 14 (b) shows the safe spaces are 36-43 and 62-72. Hence, according to the distribution of obstacle's weight, the information in the frontal environment can be clearly understood.

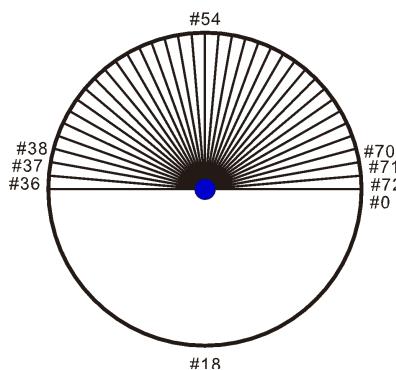


Figure 13. Dimension transition diagram.

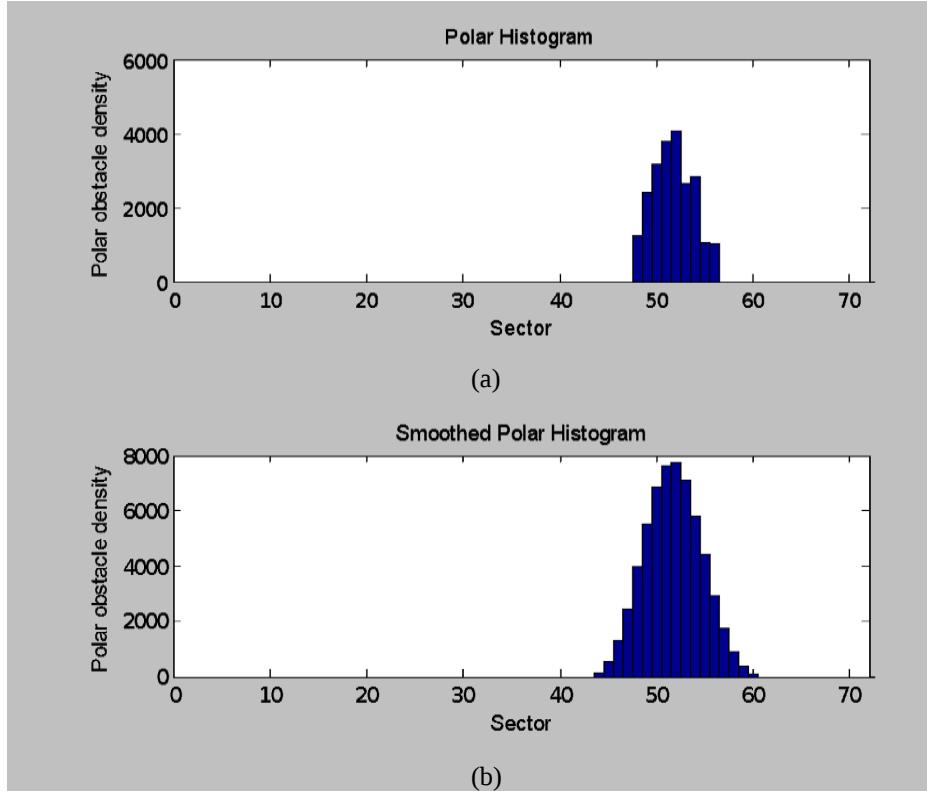


Figure 14. Polar coordinates histogram. The (a) original and (b) smoothed one.

Safe Space Cost Calculation

In order to plan the path, the costs of the safe spaces, φ_i , are calculated using Eq. (12). The set of possible safe spaces are denoted as S . To find the optimal and appropriate safe space in S , three constraints (Eq. (13), (14), (15)) are applied to define the optimal safe spaces considering with the k^{th} sector adjacent to the safe spaces, the orientation of VCP (Or_{VCP}), and previous rotation angle (k_{prev}). In Eq. (16), the angle (A_i) of φ_i whose cost is minimal is used as the rotation angle, A_{now} , the current vehicle's condition.

$$\varphi_k = c_1 E_1 + c_2 E_2 + c_3 E_3, \quad k \in S \quad (12)$$

$$E_1 = \min(k, k') \quad (13)$$

$$E_2 = \min(k, Or_{VCP}) \quad (14)$$

$$E_3 = \min(k, k_{prev}) \quad (15)$$

$$A_{now} = A_i, \quad i = \arg \min_{k \in S} (\varphi_k) \quad (16)$$

By the calculation mentioned above, A_{now} is computed at each position on the map. Therefore, the safest path is planned in the case of collision avoidance. Figure 15 displays the result of VFH algorithm, the path is smoother considering vehicle's rotation angle. Unlike VFH algorithm, the path planned by A* algorithm do not consider the vehicle's condition; thus, the orientation in the path is sometimes hard to be used in a practical condition.

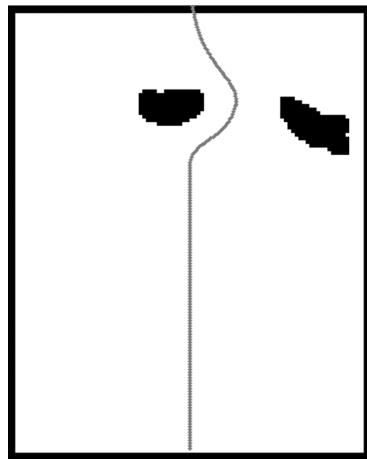


Figure 15. Result of VFH algorithm.

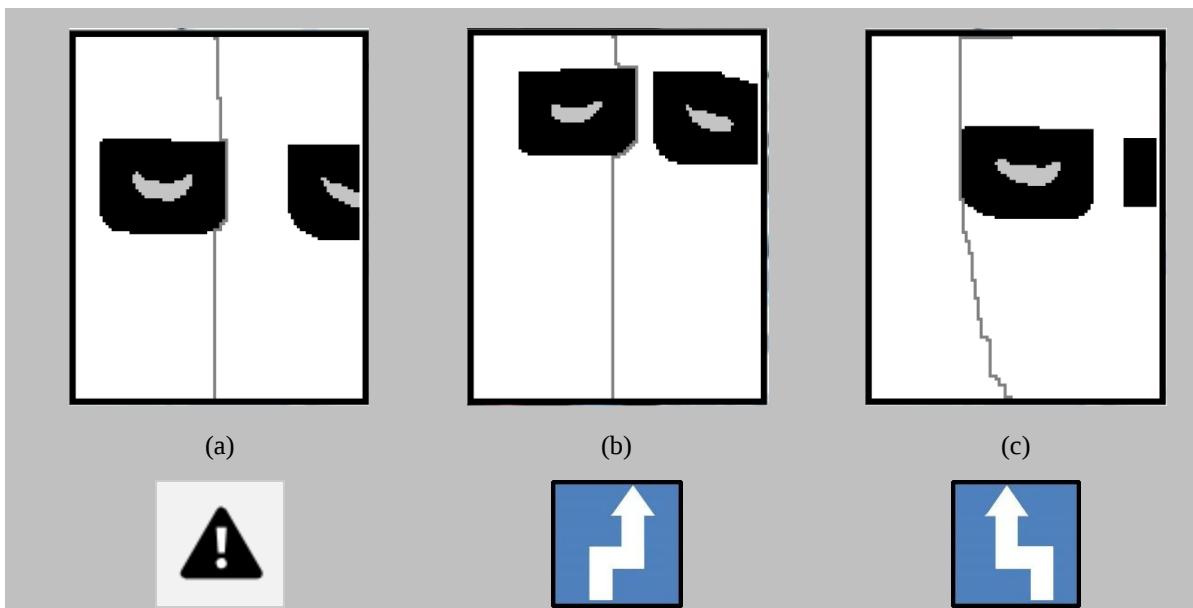
Avoidance Indication Display Method

When motion planning is accomplished, the suggested path is obtained. By the designed indications, the simple and easy-to-understand information can be displayed on the GUI. The whole map information is complicated and may cause driver distraction paying more attention to figure out. The avoidance indication was classified into three types: go straight, turn right and turn left as shown in Figure 16. The sum of bias (B_s) between desired path and estimated path was calculated written as

$$B_s = \sum_{\substack{P_a \in \text{Desired path} \\ VCP \in \text{Estimated path}}} \|P_a - VCP\|, \quad (17)$$

$$B_s \begin{cases} > T_R & , \text{turn right} \\ < T_L & , \text{turn left} \\ \text{otherwise} & , \text{go straight} \end{cases}, \quad (18)$$

where T_R and T_L were the threshold values.



(d)

(e)

(f)

Figure 16. Avoidance indication and the signs.

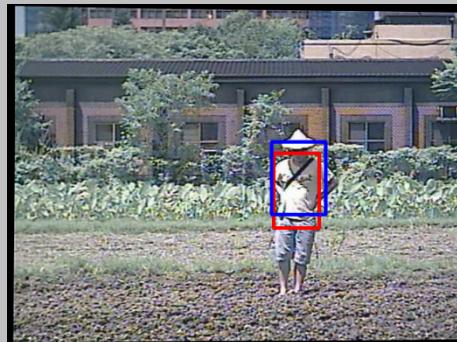
(a) Go straight, (b) turn right and (c) turn left with (d)-(f) the corresponding signs.

Experimental Results and Discussion

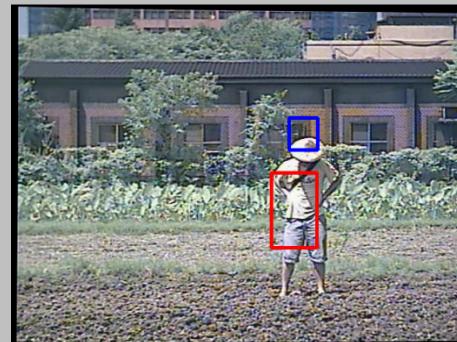
The experiments were conducted to evaluate performance of the proposed methods, and the performance of A* and VFH algorithm were also compared. In the beginning of the experiments, the camera calibration was accomplished. The intrinsic and extrinsic matrices of cameras were estimated to resist the distortion from lens. With the matrices and stereo vision theorem, each pixel's 3D information in the image plane can be estimated. The estimated depth error was around 5.2% on average with the detection range from 5 to 30 meters. The obstacles were then detected using the top-view image and the blob method. The resolution of top-view image is 125 x 100 pixels comparing with the entire disparity image (640 x 480 pixels). Thus, the computational loading was less when the detection method was executed in top-view image than in disparity image. For the whole system, the computation speed is faster than 10 fps. According to the experiments accomplished in our previous research (Lin et al., 2013), the accuracy of detection was 99.1%. The accuracy of obstacle recognition was 87.1% on average.

In the tracking experiments, the tracking result using Kalman filter was influenced by the matching method. When the obstacle was failed to match the correct obstacle using Bhattacharyya distance, the tracking result was then affected. Otherwise, particle filter was implemented since the object had been detected by the stereo vision method without using matching method. The nonlinear system of particle filter was used to diminish the tracking error caused by instant change of velocity. Figure 17 shows the tracking results of Kalman filter and particle filter. The blue and red blocks represent the result of Kalman filter and particle filter, respectively. The red block was defined as the correct tracking result and the blue block shown in Figure 17 (b) was the false tracking result. Table 1 shows the result of one of the experiments carried out under different scenarios. The correct tracking rate of Kalman filter and particle filter were 96.3% and 99.4%, respectively. Both Kalman and particle filter performed well in tracking and their RMSEs were 53.2 cm and 38.2 cm. However, in these experiments, particle filter was more robust than Kalman filter to dramatic changes of tracked objects in case of emergency brake or a rapid change in the movement orientation. The RMSEs of Kalman and particle filter were 66.8 cm and 44.6cm, and the results are shown in Figure 17.

Experiment 1



(a)



(b)



Figure 17. The tracking results of Kalman filter (blue) and particle filter (red) in different scenarios.

Table 1. Prediction results in (a) Kalman filter and (b) particle filter in experiment 1.

	Correct	False	Total	Correct rate
Kalman filter	156	6	162	96.3%
Particle filter	161	1		99.4%

The distance between obstacles and vehicle was classified into three ranges: 2.5~10 m, 10~20 m, over 20 m. When the distance is within ten meters, the collision avoidance method would warn drivers to brake immediately. The motion planning algorithms were used if the distance is within ten to twenty meters. The obstacles away from vehicle further than twenty meters won't cause the instant danger to the vehicle; hence, the method was not activated. Figure 18 shows the result of motion planning using A* algorithm and VFH algorithm. The images in left column illustrate different frontal conditions. If there was no obstruction on the set path, both paths were the straight line toward the destination (Figure 18 (a), (b) and (c)). The drivers were suggested to go right when the obstacle was on the left-hand side to avoid the collision. In contrast, the suggestion is to move left if the obstacle is on the right-hand side.

From the experiments, the algorithms were compared with their efficiency. In A* algorithm, the destination had to be predefined, and the weighting map was required to estimate the optimal path whose cost was lowest. However, the vehicle's geometrical information including length, width, and rotation radius were not considered. Instead of using vehicle's geometrical information to consider vehicle's condition, the detected obstacles in the weighting map were processed by dilation. In the middle column of Figure 8, the black boundary of each obstacle is the results of being processed by dilation method. However, the path planned by A* algorithm did not consider vehicle's rotation radius. With VFH algorithm, instead of a required weighting map, VFH needs a predefined path which is usually straight and forward to the destination. According to the main idea of VFH, when there are obstacles on the predefined path, the algorithm would change the path to detour the obstacles given vehicle's geometrical information. Considering the vehicle's rotation radius, the estimated path was smoother than the one estimated by A* algorithm as shown in Figure 8 (e), (f) and (h), (i). The average computing speed of A* algorithm and VFH algorithm were 50 ms and 30 ms, respectively. Additionally, the path length planned by A* algorithm and VFH algorithm were 150 pixels and 80 pixels respectively.

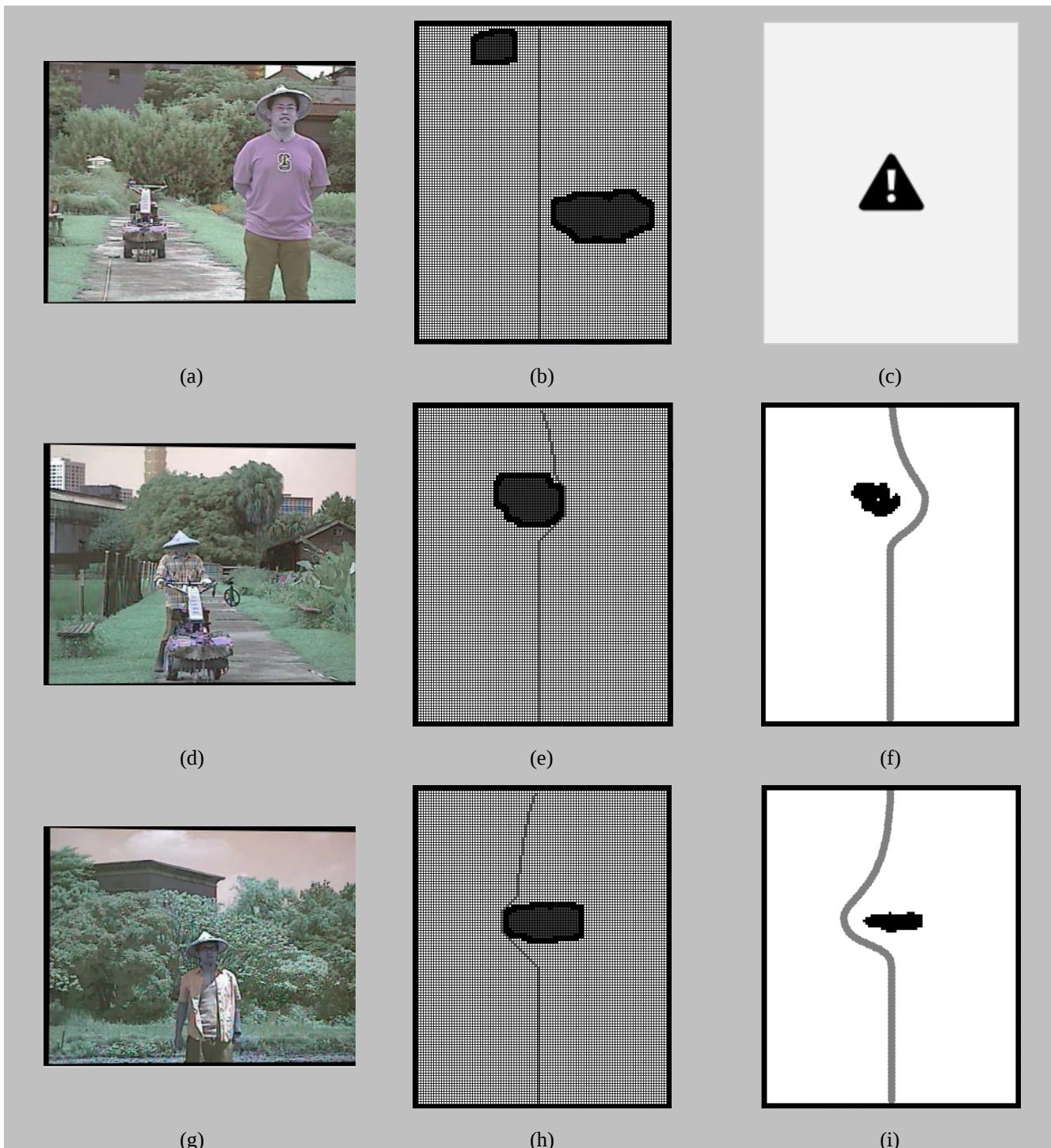


Figure 18. Results of motion planning in different conditions. (a), (d), (g) show the different frontal conditions. The middle column shows the estimated paths by A* algorithm (b, e, h) and VFH algorithm (c, f, i) for the right column.

The avoidance indications (shown in Figure 16) were applied when the collision avoidance method was active. The indication can provide signals to the driver to avoid the obstacles. The selection of appropriate indication was based on the estimated path. From Table 2, the average accuracy of correct sign displaying on the GUI was 93.5% using the proposed method.

Table 2. Results of avoidance indication displayed on GUI.

		Prediction indication	Go straight	Turn right	Turn Left	Correctness
Actual indication	Go straight	28	1	0	96.6%	
	Turn right	0	26	2	92.9%	
	Turn left	0	3	30	90.9%	
Average correctness						93.5%

Conclusions

The obstacle tracking and the collision avoidance system was developed and evaluated with experiments. The system was shown to be durable and efficient in different scenarios with nearly real-time speed (10 fps). In obstacle tracking, both Kalman filter and particle filter were applied to track objects in the experiments. Kalman filter sometimes failed to track the obstacles caused by errors in pattern matching. On the other hand, applying particle filter to track the object was more stable for rapidly change of obstacles. The RMSEs of Kalman filter and particle filter were 53.2 cm and 38.2 cm, respectively. Moreover, the RMSEs of Kalman filter and particle filter were 66.8 cm and 44.6 cm respectively when obstacles move suddenly or under emergency brake. According to the proposed method, the warning mechanism was established based on obstacle's condition. The proposed system would react and show emergency indication when the obstacle is too close to the user's vehicle. Otherwise, the motion planning method would estimate an appropriate path for the driver when detected obstacles locate in the defined range. The differences of using two motion planning methods A* algorithm and VFH algorithm, were compared. Considering the practical conditions like vehicle's geometrical information, the path planned by VFH algorithm was more efficient, smoother than the path planned by A* algorithm. In this study, VFH algorithm was proposed to be the better algorithm in motion planning. The avoidance indication was employed to display on the GUI and the correct guiding rate was over 93%.

References

- Arulampalam, M. S., S. Maskell, N. Gordon, and T. Clapp. 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174-188.
- Borenstein, J., and Y. Koren. 1991. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278-288.
- Dalal, N., and B. Triggs. 2005. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 20-26.
- Foggia, P., A. Limongiello, and M. Vento. 2005. A real-time stereo-vision system for moving object and obstacle detection in AVG and AMR applications. *Proceedings of the 7th International Workshop on Computer Architecture for Machine Perception*. 58-63.
- Hart, P. E., N. J. Nilsson, and B. Raphael. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100-107.
- Kalman, R. E. 1960. A new approach to linear filtering and prediction problems. *Journal of basic Engineering* 82(1):35-45.
- Lin, T. T., K. C. Chuang, A. C. Tsai, and Y. S. Chen. 2013. A real-time stereo vision system for obstacle recognition and motion estimation. *American Society of Agricultural and Biological Engineers (ASABE) Annual International Meeting*
- Lin, T. T., A. C. Tsai, K. C. Chuang, Y. C. Chen, and Y. S. Chen. 2012. A real-time stereo vision system for obstacle detection and recognition. *American Society of Agricultural and Biological Engineers (ASABE) Annual International Meeting*
- Pantilie, C. D., S. Bota, I. Haller, and S. Nedevschi. 2010. Real-time obstacle detection using dense stereo vision and dense optical flow. *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. 191-196

- Pocol, C., S. Nedevschi, and M.-M. Meinecke. 2008. Obstacle detection based on dense stereovision for urban ACC systems. *Proceedings of 5th International Workshop on Intelligent Transportation (WIT 2008)*.
- Yilmaz, A., O. Javed, and M. Shah. 2006. Object tracking: A survey. *ACM Computing Survey*. 38(4):1-45.