Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 60 (2015) 1423 – 1432

Procedia
Computer Science

©19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

LIDAR and monocular based overhanging obstacle detection

Jeffery Young[a], Milan Simic[a]

[a]*School of Aerospace, Mechanical and Manufacturing Engineering, RMIT University, Melbourne, Australia*

Abstract

This paper presents an improved method for the detection of obstacles in the trajectory of autonomous ground vehicle (AGV). The novel approach requires fewer calculations, i.e. less computational time. Obstacle detection algorithms were investigated, in order to perform safe motion control, in an environment with unknown overhanging obstacles. We describe a two dimensional (2D) laser sensor application, and optimal sensor configurations for mounting a monocular camera to monitor path ahead clearance. Two different sensors are used, a vision sensor and a scanning laser, Light Detection and Ranging (LIDAR). While LIDAR measures the precise distance to the object, it cannot detect low objects and overhanging obstacles due to its predefined, constant, scanning height and angle. In contrast, vision sensor provides 2D scenery information with relatively poor distance information. To compensate for the drawbacks of these two sensors, the sensor fusion method for obstacle detection of AGV is proposed. Size expansion cue algorithm is deployed to achieve that goal. Proposed method is validated experimentally.

Keywords: LIDAR; Monocular camera; Scale Invariant Feature Transform (SIFT); overhanging obstacles detection;

1. Introduction

Obstacle detection and mapping are essential for unmanned autonomous driving. Historically, there is a large number of methods and sensing devices used for obstacle detection. Both, simple contact-sensitive mechanical bumpers, as well as the laser scanners, used today, have their limitations. They do not detect hanging objects that

* Corresponding author. Tel.: +61-03-9925 6242 fax:+61-03-992 56108
E-mail address: s3314581@student.rmit.edu.au

doi:10.1016/j.procs.2015.08.218

well. In this investigation we have identified several thin overhanging objects that usually appear in an industrial environment. A test scenario presented in Fig.1 shows how 2D line scanning sensors are typically mounted and oriented on vehicles to aid detection of obstacles. The lower or higher detection region near the vehicle would be undetected. In the Fig.2 the red dotted lines depict sensor scan-lines. From the side view, it is clear that 2D sensors of this type will not detect obstacles that are not within the sensor scan area. As shown, the laser scanner can miss an overhanging obstacle completely.

Each sensor has its advantages and disadvantages. Thus, one or more sensors of different technologies have to be brought together to perform a better decision of the positional measurements. In general, active sensors [1] like LIDAR offer limited spatial information in contrast with vision systems which have rich information regarding visual appearance of an object. A fusion between active and vision sensors is desired. LIDAR offers good accuracy in longitudinal distance measurement, but poor accuracy in lateral distance measurement. Vision has the opposite properties. Following that, a fusion process will bring good accuracy in both positional measurements. In type of 2D laser scanner it is hard to detect obstacles that have different latitude, because 2D laser scanners can detect obstacle only at a specific height [2]. A 2D LIDAR was used in previous industrial environment project [3], which was able to detect some objects, like the legs of the table but it did not detect the table top. In this, new experiment, a boom gate, block tape, overhanging manufacture robotic arm are expected in the scenario.

2. System layout

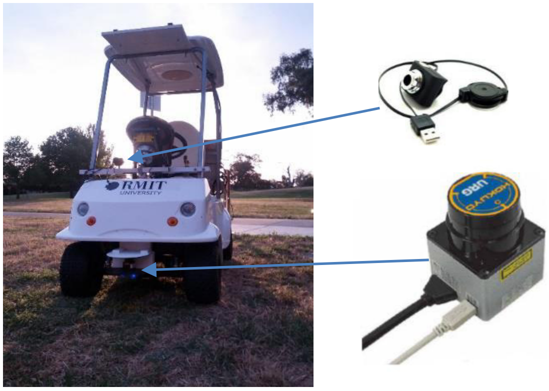


Fig.1 System layout showing Vision sensor and LIDAR.

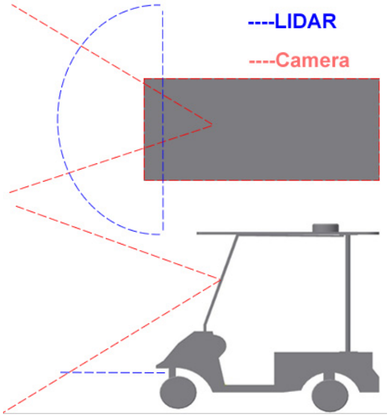


Fig.2 Top and Side view of 2 sensors' vision range.

In order to deploy described system, a modified electric golf cart is used as the autonomous ground vehicle (AGV) model. The real time localization data comes from the fusion of LIDAR and gyroscope data. LIDAR is also used for positive obstacle detection. The overhanging obstacles will be detected by a monocular camera. Our observation is that all overhanging obstacles have their spans less than *4m* from their base. Following that a Hokuyo URG lx01 with *4m* range is selected, while an USB web camera is used as the monocular camera. Camera and LIDAR are mounted on golf cart, at the front, as shown in Fig.1. LIDAR has scanning angle of *180°,* in *4 meter* range and angular resolution of *0.5°*. Monocular camera has resolution of *640*480 pixels* and *30 frames per second (fps)* update rate. Specific positions of sensors and their ranges are shown in Fig.2.

A Solid State Drive (SSD) is utilised as the primary storage device for the Laptop PC controller, due to the faster access times. The SSD is also far less susceptible to the effects of vibration and shock, as would arise from traversing harsh terrain. Read/write speed improvements positively affected the run speeds of algorithms requiring large amounts of memory access, such as mapping and stereo vision.

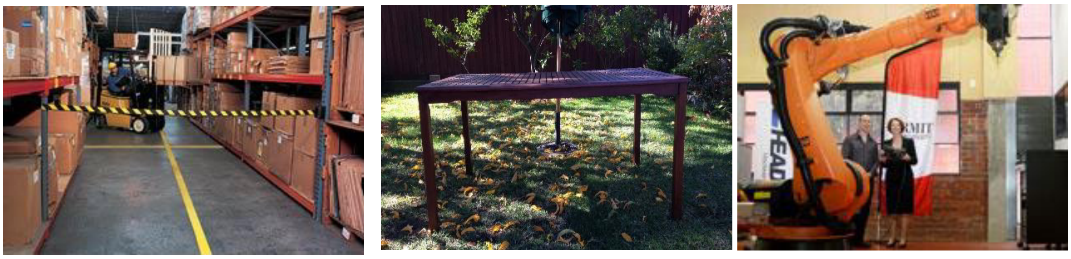
3. Obstacles definition

In general, obstacles are classified into three categories: positive, negative, and overhanging. A positive obstacle is one whose surface slope is steep. Examples of positive obstacles are stairs, wall, ground vehicles, etc.[4] Overhanging obstacles are objects that are located directly above the ground. A blocking tape like one shown in the Fig.3 (a), a larger working bench like presented in the Fig.3 (b), or robot arm from Fig.3 (c), could appear anywhere in the working area *I*, as shown in Fig.4. Negative obstacles refer to depressions on the ground, i.e., ditches, and will not be discussed.

Current vision obstacle detections are normally based on following assumptions proposed by I. Ulrich[5]. These are reasonable for a variety of environments.

- Obstacles differ in appearance from the ground.
- The ground is relatively flat.
- There are no overhanging obstacles.

The first assumption distinguishes obstacles from the ground, while the second and third assumptions can be helpful on estimating the distances of obstacles [2]. In this paper only the first two assumptions are considered. The third assumption: overhanging obstacles will be studied on, like illustrated in section 5. As the observation, in the manufacture, all those overhanging obstacles are usually in *4m* range to nearest positive obstacles. Following that, we assume that the overhanging obstacle are stretched out from nearest positive obstacle. We also assume that regions labeled "overhanging" are safe areas. These areas are referred to as "cover". If the height of cover does not meet the minimum clearance to the AGV, the regions are labeled as *being with obstacles*. The "cover" will be removed by resizing camera captured pictures, therefore "cover" will be automatically treated as safe areas in the mapping. In Fig.9, is shown that the captured picture size has been reduced to remove any possible covers.



(a) (b) (c)

Fig.3 Undetectable overhanging obstacles by 2D LIDAR sensor.

4. Environment and Task

A 2D LIDAR detects a positive obstacle when a single LIDAR pixel hits it. The lower, or higher detection region around the vehicle would be undetected. In Fig.3 (b), an undetectable obstacle scenario is presented. If AGV has only one 2D LIDAR, it will travel from current location to the umbrella straight away without avoid the big working bench. The single laser sensor showed in Fig.2 can only detect table legs as obstacles and record them into map on which a potential collision path will be generated like in
Fig.5 (a).

We assumed that the vehicle is travelling in the horizontal plane [6]. The test field is set up, as shown in Fig.4, to simulate a food manufacturing workshop, separated into two areas, of *10m*10m* each. In the left part of the workshop (area *I*) there is a large working bench. A boom gate, as one shown in the Fig.9, operates at the conjunction *G* presented in the Fig.4, between area *I* and area *II***.** While AGV is operating in area *II,* the nine work stations as shown in Fig.4 (c) are in stationary position, with robotic arm outstretched, which are typical examples of overhanging obstacles.

AGV has 3 tasks: Firstly, AGV runs from location A (charging point) to loading area B by wireless instruction order. Then, deliver parts and collect products from nine robotic arm work cell, W_i *(i=1-9)*. Finally, transports the collected products from robotic arm work stations back to the charging point. Following that, navigation system aims to enable robot to move safely and avoid all obstacles in the area *I* and robotic arm work stations in the area *II*.

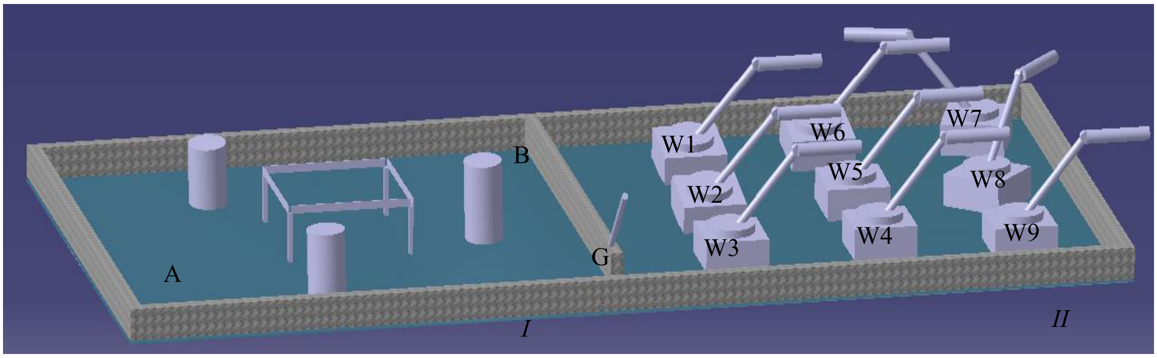
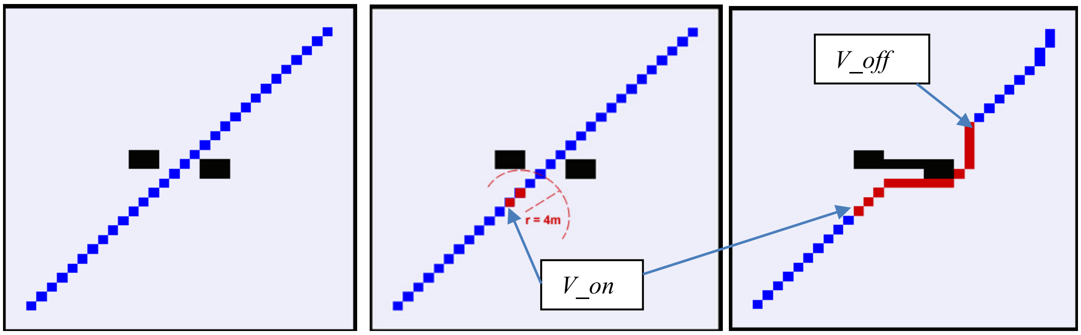


Fig.4 Test environment with 3 types of overhanging obstacles.

In the Fig.4, letter *A* stands for the charging, start and finishing point, *G* is the automatic door gate, columns represent unknown, random obstacles, like workers or other robots in the area *I,* and W_i *(i=1-9)*are stationary robots' working cells, similar to the one shown in the Fig.3 (c).

5. Overhanging obstacle objects detection methodology

To make overhanging obstacles visible, a novel method has been applied. AGV is mainly moving at a higher speed, relying just on laser data for the environment scanning. Using new approach, overhanging obstacles vision perception is switched on only after positive obstacle has been detected by LIDAR. It occurs at point *V_on* as shown in Fig.5 (b). The AGV velocity has to be dropped to a safe speed to achieve a constant time interval *T*, between each camera picture frame capture. If any frontal obstacles have been confirmed from camera vision data, AGV will update map in the real-time, for path rerouting. The flow chart and procedure are shown in Fig.6 and Fig.7. The vision perception will be switched off at the position corresponding *V_off,* if no frontal obstacles were found in *4m* range, to reduce computation load and AGV will be back to normal speed.



(a) LIDAR based Path planning. (b) Vision perception starts. (c) Regenerates map and

Fig.5. Overhanging obstacles vision perception region and AGV actions.

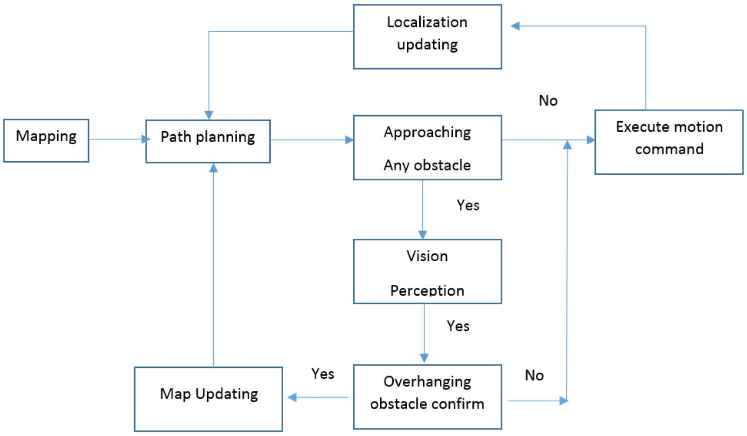


Fig.6 Overhanging obstacle detection and mapping flow chart.

```
input: LIDAR scanning and camera vision data
while  not back to A yet do
        if  LIDAR found frontal obstacles in 4m range do
              keep constant velocity &&
              Start vision perception
                 if overhanging obstacles confirmed
                       Update map
                end if
                else
             Path following
        end if
end while
Output: updated map with all positive and overhanging obstacles.
```

Fig.7 Overhanging obstacles detection algorithm.

This investigation focuses on detecting thin surface, like block tape, boom gate, even chain or wire connecting posts. Performance of optical flow techniques, particularly the performance of measurement accuracy will be studied and compared later. If an AGV can calculate the distance from the camera to an object, it is straightforward to determine which objects pose imminent threats.

A size expansion cue is useful to detect obstacles in front of the vehicle to avoid collisions. For example, when the distance from an obstacle reduces to *2m* from *4m*, the apparent size in the image becomes 2 times larger (See Fig.8). The features can be matched across multiple images and also work for larger sections of skipped features.

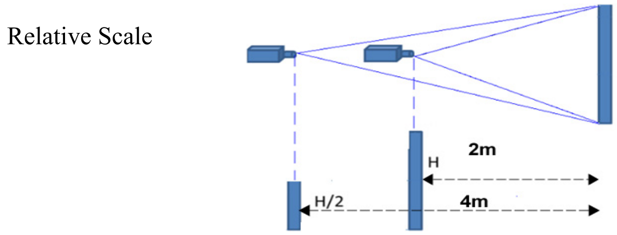


Fig.8: An example frontal obstacle size expansion in monocular camera.

Alenya et al [7] present a comparison of three methods that measure time to contact that include a looming detection method with scale invariant features. We develop a similar approach that tries to be more accurate in the scale detection and test it on an AGV that actually avoids obstacles. For real-time and reliable obstacle avoidance, we choose Scale Invariant Feature Transform (SIFT) features and template matching. Our approach uses SIFT features to detect the initial layouts of potential locations that are good to detect relative size changes. SIFT key points match even if their size has changed. Then, the algorithm uses the difference in the location and scale of the features to determine if there are any objects moving toward the camera, and if a collision is imminent.

SIFT flow is applied for extracting distinctive invariant features from images that can be invariant to image scale and rotation [8]. It is widely used in image mosaic, recognition, retrieval and etc. After Lowe, Ke and Sukthankar used PCA (Principal Component Analysis) to normalize gradient patch instead of histograms [9]. They showed that PCA-based local descriptors are also distinctive and robust to image deformations. But the methods of extracting robust features are still very slow. Bay and Tuytelaars speeded up robust features detections by using integral images for image convolutions and Fast-Hessian detector [10]. Their experiments have shown faster and reliable outcomes.

Speeded Up Robust Features (SURF) is fast and has good performance, the same as SIFT, but it is not stable to rotation and illumination changes [11]. Choosing an algorithm mainly depends on the application. Overhanging obstacle detection requires that the method used is robust with respect to the small visual changes, like scale and blur. From Table 1. Performances of 3 Table 1 [11], which also summarizes our investigation, we can see that the most suitable algorithm is SIFT.

Table 1. Performances of 3 algorithms.

Algorithm	Time	Scale	Rotation	Blur	Illumination	Affine
SIFT	common	**best**	**best**	**best**	common	good
PCA-SIFT	good	common	good	common	good	good
SURF	best	good	common	good	best	good

SIFT approximates objects based on their varying location in a series of images. The approach is very similar to optic flow-based approaches [12], except that SIFT is working at a feature level instead of a pixel level [13]. There are two assumptions: first is to assume that the camera only moves directly forward, and second that it moves orthogonal to the viewing plane. Camera turning was not covered by our investigation. SIFT would simply not be applied during the times that the camera is turning.

In addition to that, we assumed that all images are taken at constant time intervals, and that the velocity of the camera is constant, while all objects are stationary. SIFT does not require any particular unit of measurement, nor does it need to know the exact velocity of the camera. Experimental coefficient μ, as given in the equation (1) ,was evaluated during the test. The fixed time interval between frames was set to be one time step. The distances between the camera and each object were defined based on this unit of time.

To determine if a collision is imminent, we need to estimate the velocities of the camera v, and of all the objects in a series of images. We impose a 3-dimensional coordinate system to study this problem, see Fig.9 (b). The x-axis is aligned with the horizontal axis of camera images, while the y-axis aligns with the vertical axis. The z-axis, is orthogonal to the viewing plane of the camera. For this system to remain consistent from image to image, camera can only move directly along the z-axis, as we have made this simplifying assumption. The primary variable used in our implementation is the change in scale of the SIFT features. A disparity in location between two images may be the result of the object's motion. Thus, only scale can potentially determine the object's precise location. The scale of the object inversely corresponds to the distance from the camera. For example, an object that doubles in size from one frame to the next is half as far away, when camera is moving directly towards it. If the camera and an object are moving directly towards each other on the z-axis, then, again, this rate of change in scale is the only value necessary to compute the time to collision [13].

The input to the algorithm is a series of images from a moving camera taken at constant intervals. For the first image, there will be no previous images to compare. The only action taken on the first image is to extract the SIFT features. For subsequent images, we again extract the SIFT features, but we also match these features to the previous image. Matching is performed using the canonical SIFT matching algorithm [14]. Certain thresholds must be specified including the maximum allowable scale difference. We set a maximum allowable scale difference of 2, implying that a feature can be twice as large in the second image, and still match the first. The quality of matches degrades if the scale difference is larger [13].

When feature points are identified, persistent for 2 or more frames, we can interpolate their trajectories as explained in the previous section. Once its trajectory is calculated, it is straightforward to determine if a feature point is imminently dangerous. First we observe the relative velocity of the camera and the feature point along the *z*-axis. From this, we find the time to collision on the *z*-axis. If this value is below a threshold, a collision may happen. We set this threshold at 2.0, meaning that a collision is imminent if it is going to occur within the next 2 time steps. Then, for a collision to occur, the point and camera must have approximately the same *x* and *y* value at the time of the *z* collision. It is easy to determine if this is the case, because we know the *x* and *y* location as well as the *x* and *y* velocity of the feature point [13].

$$l_r = v * T * \sigma * s * \mu * f \qquad (1)$$

In the equation (1) variable l_r, gives the relative distance of overhanging obstacles to the AGV, where *v* is AGV velocity, *T* is time interval between two frames, σ is a threshold, s refers to time steps, μ is the experimental coefficient, mentioned before and *f* is focal length.

6. Experimental Performance

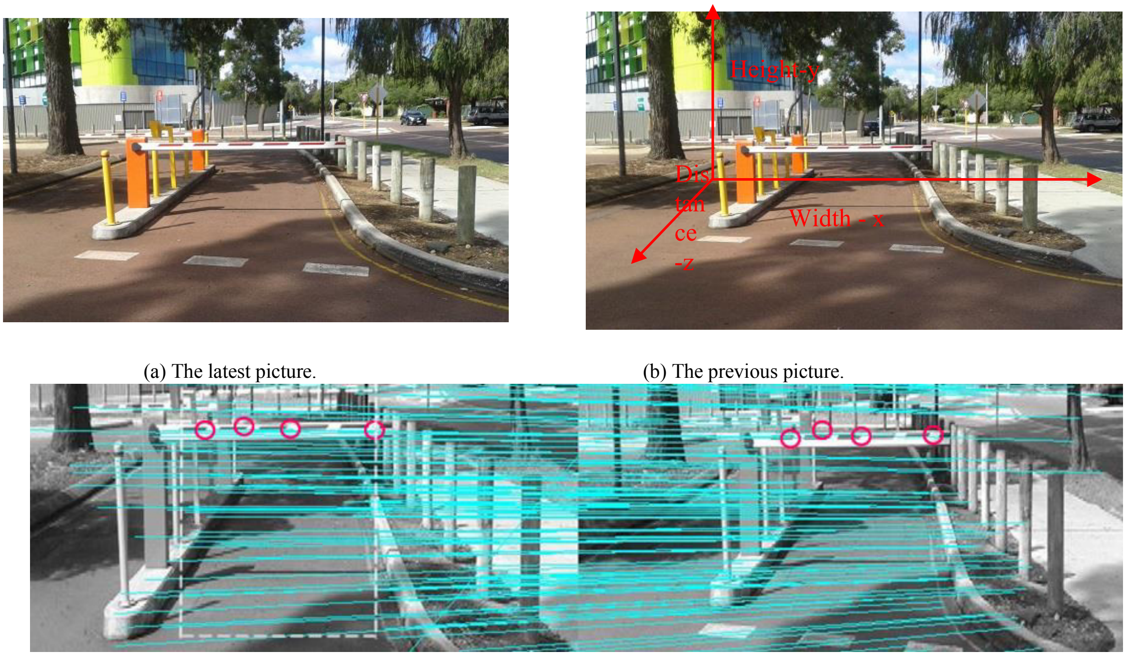
New method is tested in a near-collisions scenario as illustrated in Fig.9 (a). When AGV approaches boom gate vision perception procedure is activated, after any frontal positive obstacle has been detected in LIDAR range. The feature points are extracted from each SIFT gate feature shown in Fig.9 (c).

With the template matching algorithm [15], the ratio of feature point expansion can be detected. As shown in Fig.9 (c), first of all a pair of template images are drawn from 'previous picture' and 'latest picture'. Then the template sizes can be acquired from length of dash line rectangle square side, from both pictures. Finally, the ratio of template sizes will be calculated, and the proposed algorithm will determine whether the ratio is greater or less than the minimum alert threshold.

Two sets of parameters have to be decided based on empirical values. In general, by selecting the first set of parameters we are able to minimize both the obstacle detection time and AGV response time. On the other hand, selecting the second set of parameters assists us to determine other key factors, unrelated to time. The second set of parameters has mainly factors related to distance and angle, for instance, the frontal overhanging obstacle distance, the amount of obstacle avoidance maneuver angle, thresholds, number of time-steps and minimal frontal panic distance. In the first set of parameters, the size of SIFT gate is important. This is easily understood, as all obstacles have to be detected within the region of the SIFT gate. The proposed algorithm enables us to discriminate the expanding key points from the non-expanding key points, in a complicated background, with a monocular camera. Empirically, we do not study all pixels in a *640 *480* image. Alternatively a SIFT gate composed of *162 *218* pixels is applied. A reduced-sized SIFT gate supports detection of a minimal frontal clearance, which is still slightly greater than AGV's width and height.

Another important parameter in the first set is the minimum number of feature points to be detected for an obstacle. As we know, one single feature point is simply insufficient to report a collision. To the other end of extreme, a large number of feature points consume longer time in the matching phase. Subsequently the scale and location of the feature points may have been calculated imperfectly. To determine the appropriate number of feature points, we experimented on feature point verification by other nearby dangerous feature points. When a given dangerous feature point overlaps *3* other dangerous feature points, an imminent collision is reported. Hence, we set the minimum number of feature points as *4.*

This set of parameters is found working accurately on image sequences generated at the frame rates that exceed *4* frames per second. Fig. 10 shows the timing comparison results between solution 1 (*640*480* gate size with minimum of *10* feature points) and solution 2 (*162*218* gate size with minimum of *4* feature points). It can be seen that solution 2 is superior to solution 1 in terms of required processing time. We performed solution 2 over 100 times in our test environment with 3 types of overhanging obstacles, as shown in Fig. 4. The performance of the proposed method is evaluated by calculating average of the required time for the entire processing (from image loading, feature extraction, missing features exclusion, discarding non-informative points to template matching). Table 2 presents the average of total required time in detecting three different obstacles using solution 2.



(a) The latest picture.

(b) The previous picture.

(c) SIFT feature matching.

Fig.9. Sample images of the expanding selected key points.

In the second set of parameters, threshold values must be specified for this matching algorithm. Such thresholds include: (1) the maximum allowable scale difference, and (2) time step value. We set the maximum allowable scale difference equal to *2*, implying that a feature can be twice as large in the latest picture, and still match the previous picture. The quality of matches appears to degrade if the scale difference becomes larger. The time step is set as *0.25* second in practice.

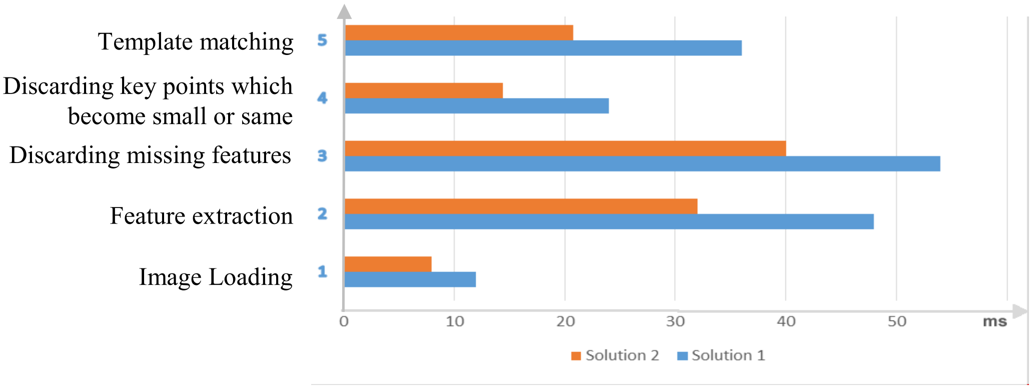


Fig. 10 Timing results comparison in different gate sizes and the least feature point solution.

Panic distance is another vital parameter to be determined. The avoidance method bases all of its calculations on the minimum panic distance from a frontal overhanging obstacle. This distance is bounded between *0* and *4m*. A weighted distance to obstacles in front of the robot is calculated where each angle is weighted using a Gaussian centred on straight ahead. The direction and amount to turns is calculated based upon a penalty term for left versus right. A turn direction is penalized by the total amount, by which obstacles to that side of the robot, are closer than maximum LIDAR range, which is *4m*.

Table 2 Summary of the experimental results for 3 types of overhanging obstacles.

Description	Boom gate	Robot arm	Block tape
Average of the total required performance time for all processes	0.453 sec	0.420 sec	0.518 sec

Should optimal value of σ and panic distance be determined, the golf cart can successfully avoid working table each time while moving at the speed of *v=1m/s*. These passes are the result of the combination of the scale expansion algorithm and the golf cart reactive behaviour. Fig.9 (c) is the image from the Golf cart when it is avoiding an obstacle. If it detects an obstacle during that motion, it will update the vision-detected obstacle into the map to regenerate a new path toward the goal.

The actual result shows that the avoidance is highly accurate: on its path, the AGV approaches close to the overhanging obstacles then avoids them successfully. The system used in the experiment optimizes the frontal overhanging obstacle vision perception speed as little time is lost for vision perception of the frontal obstacles. This enables AGV to move at a cruising speed without stopping.

7. Conclusion and future work

In this paper, a real time overhanging obstacle detection method is presented. Our method can detect frontal objects that are difficult to be detected by LIDAR. It works in real time with a low resolution camera on a raspberry pi board. In our experiments the algorithm has shown capabilities to discriminate overhanging obstacles with small surface. With our sensors configuration, the mapping and navigation system could drive the golf cart to avoid slim overhanging obstacles quickly and accurately.

The most important improvement presented in this paper, is the speed of the detector. A novel sensor configuration for fast detecting overhanging obstacle and a real-time computation with AGV running at *v=1m/s* have been demonstrated. This sensor configuration represents an important advantage for AGV applications in manufacturing and transportation. The simplicity and the use of reduced camera images make our descriptor competitive in terms of speed. Moreover, the SIFT makes the slim obstacles matching in the real time without stopping the AGV.

In the future, we intend to optimize experimental parameters in the equation (1) to increase accuracy and efficiency of SIFT overhanging obstacle perception of the autonomous vehicle. It is also planned to investigate new combinations of feature extractions like Multi-Scale Oriented Patches (MOPS) in order to improve the vision perception performances. We intend to improve chain or wire detection performance with a good camera system. Since a higher resolution image means more CPU time consumption, on-board processing may be more challenging. The approach could be implemented efficiently on a signal processor and field-programmable gate-array (FPGA) combination. This sensory system configuration could be the basis for many future driver assistance applications and for AGV systems as well.

8. References

1. Lefsky MA, Cohen WB, Parker GG and Harding DJ. Lidar Remote Sensing for Ecosystem Studies Lidar, an emerging remote sensing technology that directly measures the three-dimensional distribution of plant canopies, can accurately estimate vegetation structural attributes and should be of particular interest to forest, landscape, and global ecologists. *BioScience*. 2002; 52: 19-30.
2. Yang S, Lho H-S and Song B. Sensor fusion for obstacle detection and its application to an unmanned ground vehicle. *ICCAS-SICE, 2009*. IEEE, 2009, p. 1365-9.
3. YOUNG J, ELBANHAWI, E., and SIMIC,M. Developing a Navigation System for Mobile Robots. *Intelligent Interactive Multimedia*. Springer, 2015.
4. Chang T, Hong T, Legowik S and Abrams M. Concealment and obstacle detection for autonomous driving.

Proc of the Intl Association of Science and Technology for Development-Robotics and Application. Citeseer, 1999.

5. Michels J, Saxena A and Ng AY. High speed obstacle avoidance using monocular vision and reinforcement learning. *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, p. 593-600.

6. Iqbal U, Okou AF and Noureldin A. An integrated reduced inertial sensor system—RISS/GPS for land vehicle. *Position, Location and Navigation Symposium, 2008 IEEE/ION*. IEEE, 2008, p. 1014-21.

7. Alenyà G, Nègre A and Crowley JL. A comparison of three methods for measure of time to contact. *Intelligent Robots and Systems, 2009 IROS 2009 IEEE/RSJ International Conference on*. IEEE, 2009, p. 4565-70.

8. Lowe DG. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*. 2004; 60: 91-110.

9. Ke Y and Sukthankar R. PCA-SIFT: A more distinctive representation for local image descriptors. *Computer Vision and Pattern Recognition, 2004 CVPR 2004 Proceedings of the 2004 IEEE Computer Society Conference on*. IEEE, 2004, p. II-506-II-13 Vol. 2.

10. Bay H, Tuytelaars T and Van Gool L. Surf: Speeded up robust features. *Computer vision–ECCV 2006*. Springer, 2006, p. 404-17.

11. Juan L and Gwun O. A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*. 2009; 3: 143-52.

12. Souhila K and Karim A. Optical flow based robot obstacle avoidance. *International Journal of Advanced Robotic Systems*. 2007; 4: 13-6.

13. Chavez A and Gustafson D. Vision-based obstacle avoidance using SIFT features. *Advances in Visual Computing*. Springer, 2009, p. 550-7.

14. Lowe DG. Object recognition from local scale-invariant features. *Computer vision, 1999 The proceedings of the seventh IEEE international conference on*. Ieee, 1999, p. 1150-7.

15. Mori T and Scherer S. First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles. *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, p. 1750-7.