

# Suspended Obstacle Detection for Plant Site Inspection Robots with Monoscopic Camera

Noppawit LERTUTSAHAKUL, Tohoku University

○ Keiji NAGATANI, Member, Tohoku University, keiji@ieee.org

Atsushi WATANABE, Tohoku University

In 2015, Field Robotics Laboratory at Tohoku University had joined a competition named “ARGOS” which stands for Autonomous Robot for Gas and Oil Sites. It is a challenge to develop a robot that can autonomously patrol and inspect a gas and oil plant site. During the second competition at the competition site, the robot “AIRK” had a problem that it could not detect a suspended obstacle. It represented off-limit region, and it located higher than the field of view of the laser scanner. To detect such obstacle with minimum remodeling, the authors decided to use the onboard forward-facing camera and a help of computer vision to detect an obstacle that was invisible to the laser scanner. In this paper, the basic ideas and evaluation results of the obstacle detection method are introduced.

**Key Words:** Inspection robot, Obstacle detection, Computer vision

## 1 Introduction

Field Robotics Laboratory at Tohoku University had joined a competition named ARGOS (Autonomous Robot for Gas and Oil Sites). It is a challenge to develop a robot that can autonomously patrol and inspect a gas and oil plant site, as shown in Figure 1. In the second competition, the robot “AIRK” had a problem that it could not detect a suspended obstacle. It represented an off-limit region, and was placed higher than the field of view of the laser scanner.

To detect obstacles placed at a higher position, there are several ways, such as adding another laser scanner, adding contact sensors, using a computer vision. After considering the possibilities, effectiveness, robustness, and minimum remodeling, the authors chose a computer vision approach. Other choices were either not effective or not convenient. With computer vision, only one front facing wide-angle camera that equips with the AIRK is necessary.

There are various methods of obstacle detection using a monoscopic camera such as use of support vector machine [1] and reinforcement learning [2]. However, both methods need training and do not give information about the height of the obstacle. The method proposed in this paper will yield location and height information and will not need training. Another idea is to detect the motion of the object in the frame. With this approach, the current image frame is compared with the previous frame. The pixel that changes (or the object that is moving) between two frames will appear as a blob of white contour in the “delta” frame. Then the delta frame is processed, by considering the size and intensity of each contour. If any contour that is big and intense enough, it will be treated as an object and tracked. Other small contours will be treated as noise. The method was simple, but it was unreliable in the case that the camera is mounted on a robot that moves.

Finally, we settle on an idea of finding a height and location of the object that is visible in the frame, based on features tracking method, as explained in the following.

In this paper, the basic ideas for suspended obstacle detection is introduced, and some evaluation results are reported for our ARGOS challenge.



**Fig.1** AIRK robot with the suspended obstacle.

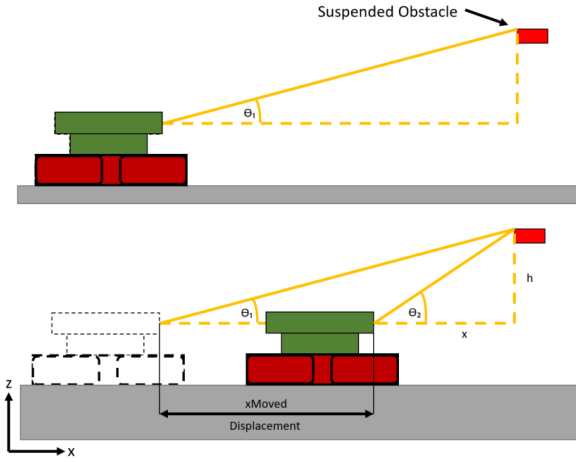
## 2 Principle

Here, the principle of suspended obstacle detection is described, particularly the height and distance of the target object.

Firstly, the algorithm requires to find a good feature of the obstacle and represent it as a point [3], such as a corner or a change in color. These points have a strong gradient in pixel intensity, and it is a feature of the point. This feature detector will find a dominant feature (a gradient in the pixel intensity) around a pre-defined point. Then, it will pinpoint the position of this corner with sub-pixel accuracy.

After feature points have been chosen as good features, when the point moves, the algorithm requires to know where the new location of each point is. In this approach, Lucas Kanade Method is applied [5] [6]. This feature tracker finds the optical flow of multiple images. The optical flow shows where the point moves based on 2 assumptions: (1) intensities of the pixel for the same object is almost constant between the frame, and (2) the nearby pixel will have a similar motion.

The mathematics behind this idea is simple. As the robot approaches to the obstacle, which will be in the upper area of the horizon of the frame, the obstacle will move upward in the frame of the camera or increases the angle viewed from the side. Now, the robot can compare the difference of the location of the top of the object between the current frame and the previous frame as shown in Figure 2.



**Fig.2** Principle of the calculation.

The robot knows the displacement ( $xMoved$ ), which is obtained from the odometry data. However, the key value used in the equation is the angle  $\theta_1$  and  $\theta_2$ , obtained from the image. Therefore, a pixel-angle conversion equation is necessary. In here, it is obtained experimentally, as shown in Figure 3.

With known angle of  $\theta_1$ ,  $\theta_2$  and displacement  $xMoved$ , the equation (1) and (2) can be formed from 2 image frames:

$$\tan \theta_1 = \frac{h}{xMoved + x} \quad (1)$$

$$\tan \theta_2 = \frac{h}{x} \quad (2)$$

$x$  is the distance to the object, and  $h$  is the height of the object.

Since it is the same object, the actual height of the object would remain unchanged. Therefore, the height and the distance from the object by:

$$x = \frac{\tan \theta_1 \cdot xMoved}{\tan \theta_2 - \tan \theta_1} \quad (3)$$

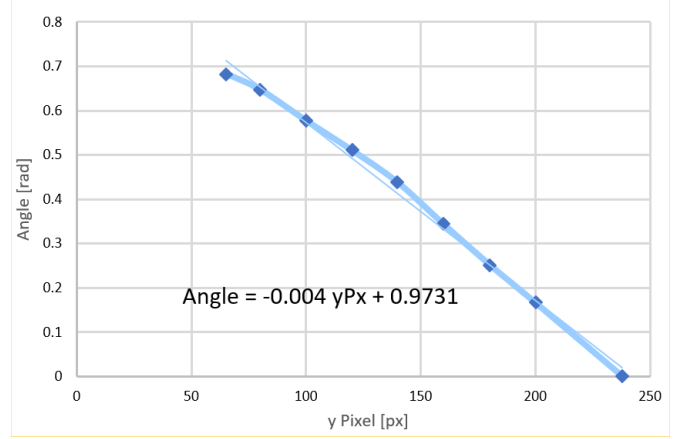
$$h = \frac{\tan \theta_1 \cdot xMoved}{\tan \theta_2 - \tan \theta_1} \cdot x. \quad (4)$$

So, the distance from the object is obtained by the equation (3), and the height of the object is obtained by the equation (4). With the known  $x$  from the object, the angle from the center of the object can be obtained by a conversion equation from x-axis pixel to angle. The conversion equation can be obtained from the process similar to the one for y-axis pixel. With 3 values available: distance, height, and an angle from the center of the camera, the 3D coordinate can be formed.

### 3 Implementation

The above mentioned method was implemented on the AIR-K ver1.5, based on the OpenCV [4] framework. The process of the obstacle detection includes the following 4 steps.

1. Find: Each point on an image has its own unique feature. By using *cornerSubPix* of OpenCV feature detector, it finds these features on an image around a pre-defined point. Then these features and point location on the image are logged to be used later in the process.
2. Track: As the robot moves, the image changes. The location of an object in an image changes, but the object does not. The same object has the same feature. By using the iterative Lucas-Kanade method, OpenCV feature tracker tracks these feature points as the robot moves and logs the location of the point. While the robot moves, it



**Fig.3** Pixel v.s. angle in radian.

knows how much does it displaced. This information is also logged into each feature points.

3. Calculate: After the robot has moved to a certain distance. The point in the image has a certain change of location. With these information available, a calculation can be made to find the location and height of an object relative to the robot.
4. Determine: With the location and height information of an object, a determination can be made whether that object is an obstacle or not.

According to the above procedure, some tests were conducted in indoor environment in Tohoku University. Figure 4 shows information of the detected point that represent an object for debugging and visualization purpose. In Figure 4, the robot detects feature points around the suspended chain, and in figure 5, it detects it as an obstacle located at the high position.

In this example, the camera was a standard USB camera with an ultra-wide angle lens. The height of the camera location is 32 cm from the ground. From the specification of the camera lens, the field of view (FOV) of the camera is  $70^\circ$  in vertical axis and  $120^\circ$  in the horizontal axis. However, it was very difficult to calculate around the edges of viewing angle.

### 4 Evaluation results

A series of evaluation tests was conducted to check for the accuracy of the detected value and detectability. The detection area is divided into region of interest (ROI) and trigger zone. ROI is the whole calculable area of an image; trigger zone is the area where obstacle will likely to be located.

There are three following tests. The experimental environment in case of angle accuracy test is shown in Figure 6, and results are summarized in Table 1.

1. Height and distance accuracy test

The test was conducted by varying the height of a stack of wooden blocks from 0.4 to 1.0 m. Then, the robot calculated the height and distance at the distance of 0.5 m, 0.8 m, and 1.0 m at each running distance of 0.3 m, 0.5 m, and 1.0 m.

2. Angle Accuracy test

The test was conducted by arranging a wooden block at the height of 0.7 m. Then, moved the wooden block side way to vary the calculated angle.

3. Maximum detectable height test

The test was conducted by moving the robot that was placed at 0.8 m, 1.2 m, or 1.6 m parallel to the obstacle.

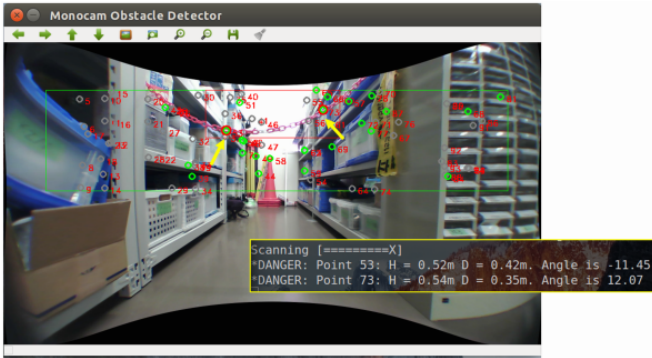


Fig.4 Suspended chain is tracked.



Fig.5 Suspended chain is detected.

Then, it headed toward the center line leading to the obstacle. After that, it made 90° turns toward to obstacle. During this motion, the maximum detectable height was evaluated.

According to the result of both height and distance accuracy test, it was not impressively accurate. However, it is acceptable for an obstacle detection. The angle accuracy is acceptable for an obstacle detection at  $\pm 3^\circ$  in the trigger zone. The detection test shows that the robot can detect an obstacle with a reasonable height at close distance.

## 5 Conclusion

Suspended obstacle detection has been a problem for AIRK team. With this research, the problem was almost solved by using a standard monoscopic camera and computer vision. The robot sees the obstacle by its camera, then try to calculate the height and location of that obstacle by using geometry.

Table 1 Evaluation Results.

Table 1: Test Result		
Height Accuracy	Trigger Zone (Direct Path)	$\pm 5\%$ or 0.1 m
	ROI	$\pm 15\%$ or 0.3 m
Distance Accuracy	Trigger Zone (Direct Path)	$\pm 15\%$ or 0.3 m
	ROI	$\pm 30\%$ or 0.5 m
Angle Accuracy	Trigger Zone (Direct Path)	$\pm 3^\circ$
	ROI	$\pm 5^\circ$
Maximum Detectable Height		0.5 m at 0.8 m
		0.8 m at 1.2 m
		0.9 m at 1.6 m

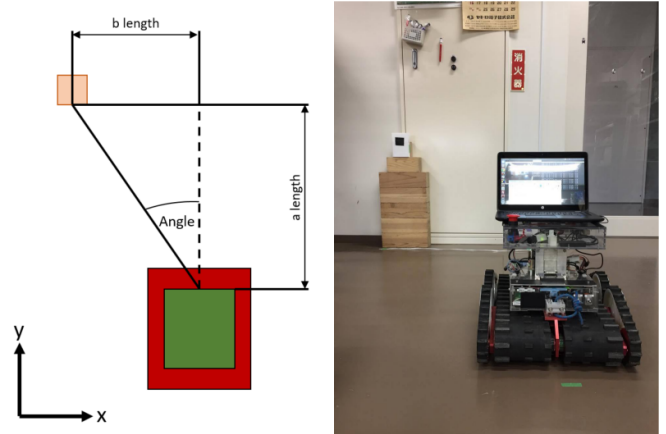


Fig.6 Experimental scene of angle accuracy test.

The calculated location was not as accurate nor as good as those of traditional obstacle detection sensor, such as laser range finder. However, this method of obstacle detection can detect obstacles with minimum remodeling. Future work includes expanding to multiple camera, improving algorithm, and improving the accuracy.

## References

- [1] Cooper Bills, "Vision-based obstacle detection and avoidance", <https://tinyurl.com/hzb2lxz> (2011).
- [2] Jeff Michels, "High speed obstacle avoidance using monocular vision and reinforcement learning" Proceedings of the 22nd Int'l conf. on Machine learning (2005), pp. 593-600.
- [3] David G. Lowe, "Distinctive image features from scale-invariant keypoints", Int'l Journal of Computer Vision, 60, 2 (2004), pp.91-110.
- [4] OpenCV Documentation, "Feature Detection", [http://docs.opencv.org/3.0-beta/modules/imgproc/doc/feature\\_detection.html](http://docs.opencv.org/3.0-beta/modules/imgproc/doc/feature_detection.html) (2014).
- [5] Dhara Patel, "Optical Flow Measurement using Lucas Kanade Method" Int'l Journal of Computer Applications, 61, 10 (2013), pp.6-10.
- [6] OpenCV Documentation, "Motion Analysis and Object Tracking", [http://docs.opencv.org/3.0-beta/modules/video/doc/motion\\_analysis\\_and\\_object\\_tracking.html](http://docs.opencv.org/3.0-beta/modules/video/doc/motion_analysis_and_object_tracking.html) (2014).