

Comparative Presentation of Real-Time Obstacle Avoidance Algorithms Using Solely Stereo Vision

Ioannis Kostavelis, Lazaros Nalpantidis and Antonios Gasteratos

Robotics and Automation Lab.,
Production and Management Engineering Dept.,
Democritus University of Thrace, Greece.

Abstract. This work presents a comparison between vision-based obstacle avoidance algorithms for mobile robot navigation. The issue of obstacle avoidance in robotics demands a reliable solution since mobile platforms often have to maneuver in arbitrary environments with high level of risk. The most significant advantage of the presented work is the use of only one sensor, i.e. a stereo camera, which significantly diminishes the computational cost. Three different versions of the proposed method have been developed. The implementation of these algorithms consists of a stereo vision module, which is common for all the versions, and a decision making module, which is different in each version and proposes an efficient method of processing stereo information in order to navigate a robotic platform. The algorithms have been implemented in C++ and the produced frame rate ensures that the robot will be able to accomplish the proposed decisions in real time. The presented algorithms have been tested on various different input images and their results are shown and discussed.

1. Introduction

The main purpose of this work is the development and the comparison of three vision-based obstacle avoidance algorithms. A successful obstacle avoidance algorithm should be able to adapt to local conditions and at the same time to be computationally efficient, even in unstructured and unknown environments. This behavior becomes more demanded due to the restricted computational resources that a mobile platform usually provides. The only sensor that has been used in the presented implementations is a stereo camera.

Stereo vision is a technique that offers a lot of information and can produce efficient results when applied to robot

navigation tasks. As previously mentioned, one of the implemented modules performs the required stereo processing. This module produces reliable and detailed disparity images, i.e. depth maps, providing depth information about the scenery in front of the mobile robot. The second module that has been developed takes advantage of the depth information previously acquired and finds the most appropriate direction for the robot in order to avoid any possible obstacles. The disparity images have been created using the C++ application program interface (API) of Point Grey Research [1], which is also the manufacturer of the used stereo camera. The decision making methods are also written in the C++ programming

language and comprise innovative methods for stereo vision obstacle avoidance.

2. Related Work

In mobile robot navigation lots of techniques are used such as odometry, active beacon, and GPS systems as extensively mentioned in [2]. These techniques can coexist as part of combinational efforts to define a mobile robot's position and determine the required navigation instructions. All the aforementioned methods demand a variety of sensors that should be installed on the platform [3]. There are also other hybrid implementations that involve stereo vision systems and ultrasonic sensors which are used in localization and mapping problems [4]. Furthermore, the use of solely stereo vision can be applied to the detection of 3D objects efficiently as it is described in [5]. Considering the above as a background the contribution of this work is the development of an algorithm for obstacle avoidance with the use of only one stereoscopic camera, shown in Figure 1. This choice has the additional advantage that the proposed system could be easily integrated with other vision-based methods such as object recognition and tracking.



Figure 1. The stereoscopic camera Bumblebee 2 of Point Grey Research.

3. Stereo Vision Module

The stereo vision equipment utilized in this work is the Bumblebee2 stereo camera by Point Grey Research. Point Grey Bumblebee2 stereo vision cameras

are factory calibrated. The Bumblebee2 uses two CCD image sensors and provides quality 3D data and real-time processing speed. It is able to produce as output 640x480 pixel images at 48 frames per second, or 1024x768 pixel images at 20 FPS, through its IEEE-1394 interface.

The stereo camera is used in order to capture two pre-calibrated images. The images are aligned and corrected in order to remove the lens distortion and make sure that epipolar lines are parallel to the horizontal axis. A successful alignment can ensure the production of correct disparity images because there is disparity only along the horizontal direction. The disparity is usually computed as a shift towards left of an image feature when viewed in the right image. A point that appears at the horizontal coordinate x , in the left image may be present at the horizontal coordinate $x-d$ in the right image, where by d is denoted the point's disparity in pixels. The obstacles that are closer to the stereo camera should have greater disparity values than the obstacles, which are located at the background of the scenery. In the present work the depth maps are calculated using the fixed functions, which are provided by the Point's Grey support development kit (SDK). The stereo SDK supplies an optimized fast-correlation stereo process that rapidly calculates the Sum of Absolute Differences (SAD) stereo correlation method. This is a very quick and robust method and produces dense disparity images. The reference (left) image of a self-captured stereo pair is shown in Figure 2a, while Figure 2b depicts the result of the stereo processing, i.e. the disparity map, of that stereo image pair.



(a) (b)
Figure 2. The reference image (a) and the produced disparity map (b) for a stereo pair.

4. Implementation Methods for the Decision Making Module

The second module of this work takes advantage of the information that is stored in the disparity image in order to navigate the mobile robot. When obstacles are detected the algorithm has to make the decision whether to move the robot forward or to steer it left or right. Three different possible methods have been developed and all of them have as common target to navigate the robot towards the direction with the fewer obstacles. Another common characteristic of the proposed methods is that all of them initially divide the disparity map into three horizontally tiled sub-regions or windows.

4.1. The mean estimation method

Firstly, the disparity map is divided into a left-side window, a central window and a right-side window, as shown in Figure 3. For each window, the average disparity value is calculated. The window having the smaller average disparity value indicates the direction with the fewer obstacles. For example, in the disparity which is shown in Figure 3 the mean values for each window are: Left = 78.7 pixels, Central = 79.2 pixels and Right = 44.5 pixels.

Comparing the three mean values, it is shown that the right window has the smaller one. As a result there should be the fewer obstacles in that way and the



(a) (b)
Figure 3. Reference image (a) and disparity map divided into three windows (b).

robot should make the decision to steer right. This method is very efficient when there are not many obstacles in one of the three windows. In order to verify this conclusion one more scene is tested, as shown in Figure 4. The processing of this image set gives the following mean disparity values for each direction: Left = 66.8 pixels, Central = 80.2 pixels and Right = 61.5 pixels.

In this case the algorithm would take the decision to steer right. However, as we can see there is enough space in front of the robot to move before it should have to steer in order to avoid a collision.

The conclusion is that occasionally the algorithm presents hesitantly behavior. As a consequence, there should be defined another method in order to overcome this behavior. Thus, the threshold estimation method has been developed.



(a) (b)
Figure 4. The reference image (a) and the produced disparity map (b).

4.2 The threshold estimation method

This method also divides the disparity map into three windows of pixels, as shown in Figure 3. The flow of this method is as follows:

1. In the *central* window, the pixels p whose disparity value $D(p)$ is greater than a defined threshold value T (i.e. $T = 120$), are enumerated.
2. Then, the enumeration result is examined. If it is smaller than a predefined rate r (i.e. $r = 20\%$) of all the central window's pixels, this means that there are not any obstacles and the robot can move forward.
3. On the other hand if this enumeration exceeds the predefined rate, the algorithm examines the other two windows and chooses the one with the smaller average disparity value.

The threshold value and the rate value can control the hesitance of the algorithm. If the threshold value become greater i.e. $T=140$ the algorithm will become less hesitant and will closer approach the obstacles. These parameters are very useful when the mobile robot has to maneuver in restricted areas. Let as apply this algorithm to the disparity map, which is depicted in Figure 4. The 20% of the central window's pixels is: 12.376. The pixels whose value is greater than $T=120$ are 10.037 pixels. As a result $10.037 < 12.376$ and the mobile robot decides to move forward and closer to the obstacle. The value $T=120$ is able to bring the robot approximately 50cm close to the obstacles, before indicating change of direction.

4.3 The multi-thresholds method

The third method is similar to the second method. The disparity map is also divided into three windows.

1. In *each* window the pixels p whose values $D(p)$ is greater than

a defined threshold value T , are enumerated.

2. The resultant values are compared and the window with the smaller value is selected. If this value is smaller than a predefined rate r (i.e. $r=20\%$) of the selected window's pixels the robot chooses to move towards to the corresponding direction.
3. In case that the all of the three values are greater than the predefined rate this means that the robot is very close to the obstacle and should accomplish a different routine such a 180° rotation or a movement to the previous position in order to take a different decision.

The most important difference from the previous methods is that this method first examines the traversability of the scenery and then takes the decision to move towards a direction. This prevents the algorithm from making a decision that will have as a result a collision to an obstacle. In Figure 5 a disparity map is shown that depicts a non-traversable terrain. The three methods have been applied on this disparity map in order to compare their results.

Concerning the mean estimation method, the average disparity values of each window are: Left = 143.1 pixels, Central = 165.8 pixels and Right = 128.9 pixels.

Thus, the algorithm makes the decision to steer the robot right. Taking a better look of the reference and the disparity image in Figure 5 it is easy to realize that there are also obstacles in the right window. For that reason the decision to steer right is not advisable. Applying the threshold estimation method on the same disparity image the pixels whose value

are greater than $T=120$ are 54.493 and are more than the 20% of the central window's pixels. The algorithm's next step is the examination of the side windows, comparing the average values as previous. The result in this case is once more that the robot has to turn right.

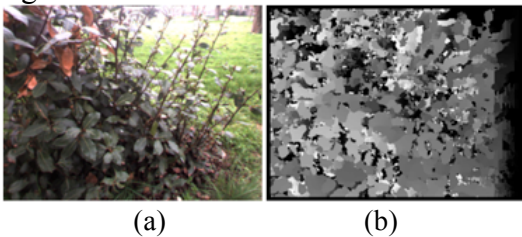


Figure 5. A non-traversable terrain with bushes (a) and its disparity image (b).

Finally, the third method is examined. The enumerations of the pixels whose values is greater than the threshold $T=120$ are for each window greater than the predefined rate ($r=20\%$). In this case the algorithm understood that the terrain is non-traversable and the output is a rotation of 180° . In that case the collision has been avoided.

5. Conclusions

A comparison of three efficient vision-based algorithms for obstacle avoidance was presented. Each algorithm constituted from a stereo vision module, which produces dense disparity maps, and three different decision making methods. The disparity images were acquired using the Point Grey's Research API. The very good quality of the produced disparity images provided the examined decision making methods with reliable input data. Out of the three decision making methods presented and examined, the multi-threshold method was proven to be the more efficient. It works properly in all the tested cases and avoids almost every collision with

obstacles. Despite its simple calculations this decision making algorithm exhibited a very good performance and can comprise a very stable first step in the solution of localization and mapping problems. Taking into consideration all the above, the proposed multi-threshold algorithm can be used for real-time obstacle avoidance and navigation requiring a minimum of computational cost and using only a stereo camera.

Acknowledement

This work has been supported by the View-Finder FP6 IST 045541 Project.

References

- [1] Point Grey Research. "Triclops Stereo Vision System Manual", Version 3.1, 2003.
- [2] J. Borenstein, H.R. Everett, L. Feng, and D. Wehe "Mobile Robot Positioning & Sensors and Techniques". Invited paper for the Journal of Robotic Systems, Special Issue on Mobile Robots. Vol. 14 No. 4, pp. 231 – 249, 2007.
- [3] N. Vandapel, R. Donamukkala and M. Hebert. "Experimental Results in Using Aerial LADAR Data for Mobile Robot Navigation", 4th International Conference on Field and Service Robotics, July 14–16, 2003.
- [4] S. Soumare, A. Ohya and S. Yuta. "Real-Time Obstacle Avoidance by an Autonomous Mobile Robot using an Active Vision Sensor and a Vertically Emitted Laser Slit" Intelligent Autonomous Systems 7, pp. 301-308, 2002.
- [5] D. Murray and J. Little. "Using Real-Time Stereo Vision for Mobile Robot Navigation." Autonomous Robots 8, pp. 161–171, 2000.