Conference Paper

Depth Estimation of an Underwater Object Using a Single Camera[1]

Benjamin Champion[1]*, Mo Jamshidi[2], and Matthew Joordens[1]

[1]School of Engineering, Deakin University, Waurn Ponds, VIC, Australia
[2]Collage of Engineering Electrical Engineering, University of Texas at San Antonio, TX, USA

Abstract

Underwater robotics is currently a growing field. To be able to autonomously find and collect objects on the land and in the air is a complicated problem, which is only compounded within the underwater setting. Different techniques have been developed over the years to attempt to solve this problem, many of which involve the use of expensive sensors. This paper explores a method to find the depth of an object within the underwater setting, using a single camera source and a known object. Once this known object has been found, information about other unknown objects surrounding this point can be determined, and therefore the objects can be collected.

Keywords: Underwater robotics, single camera depth, underwater object retrieval

Corresponding Author:
Benjamin Champion; email:
benjamin.champion@deakin
.edu.au

Received: 28 November 2016
Accepted: 4 December 2016
Published: 9 February 2017

Publishing services provided by Knowledge E

Selection and Peer-review under the responsibility of the DesTech Conference Committee.

OPEN ACCESS

1 Introduction

In the field of robotics, determining where an object is relative to the detecting device, generally a robot, is not a simple problem. It becomes even more vital when object collection is considered, as the position of the object needs to be determined before any work can be done towards collecting it. This task, while still challenging, can be accomplished on the land and in the air by using a whole range of sensors such as laser range finders, cameras, ultrasonic sensors etc. In the underwater domain this issue becomes much harder as the range of sensors available is significantly smaller. Because of this, a method to localize an object by utilising a single camera has been investigated.

The reasoning behind using a single camera is this sensor is already present on the VideoRay Pro 3 robot, the device currently being used for this experiment and the need to install extra external cameras is removed. Generally, to be able to construct a depth map by using a camera, two or more devices are required. Robust techniques can then be used to generate a fairly accurate map, such as Depth from Stero and Depth from Defocus [1,2]. Due to only one camera being available, these techniques were discounted. Throughout the literature many different techniques have been trailed, each giving significantly different outcomes. A system using a rotating mirror has been investigated where objects of different depths will be represented by observing the change in pixel speed of the rotating image [3]. Using predefined information about the scene is often

required to then train an algorithm by either using associated depth maps [4] or known information about a structured object, such as hands or feet [5]. Techniques have been proposed where a light source is added into the environment in the forms of either points [6] or patterns [7], and by measuring how these light sources are changed, the depth two the object can then be determined. An application was shown where an object would use the changes in the pattern of the ground to determine where objects are relative to themselves, though this method required a predefined knowledge of the grounds colours and textures [8] as well as having the camera continuously pointed at the ground. The shapes from shading method can also be used to approximate depth, but is difficult to apply in situations that do not have a very uniform intensity and textures [9]. A Marcof filter can also be trained to determine the depths of known image characteristics, and then applied to segments of the image to get an overall depth map, though a significant amount of information is required for this training process [10].

Of the researched methods, a simpler approach was determined necessary for the proposed application. This application is for a swarm of robots to be able to generate a map of known objects that can easily be shared among the agents in the system, and therefore be used to determine the location of unknown objects relative to these points in future applications. The following paper explains the detection method used to find the depth objects within the image obtained from the camera, and how the depth and position of these object can be determined.

2 Vision System

The initial challenge was which toolkit was to be used to process the image data being received from each of the connected robots. To accomplish this task, the open source toolkit EMGU was used. This toolkit is a wrapped version of the commonly used opencv toolkit for .NET applications, and therefore benefited from the large community and code base that opencv brings. There are other C# vision API that can be used to achieve the same, or similar results as the one proposed, such as AForge.Net, but EMGU was decided upon due to opencv being the most popular vision api.

This leads to the proposed problem, this being how to determine the depth of an object relative to the robot, without knowing any information about the object beforehand and also without the use of a ranging system, such as sonar. As described in the introduction, there have been many different approaches proposed to solve this problem, many of which are not suitable for this application. The proposed method involves populating the environment with objects of a known size that are significantly different to the rest of the environment, in this case tennis balls. In a real world application objects similar to this, these being uniform objects preferably spheres, can be utilised. By using the known parameters of these objects, the depth that these objects are at relative to the robot can be found. Utilising this information, the depth and size of an

Figure 1: Image before distortion removal.

object in proximity to one of these depth markers can be determined, and used for the collection of the objects.

2.1 Detection of the Object

The first problem that needed to be overcome is how to detect the depth objects, in this case a tennis ball, under the water by simply using the camera attached to the robot. In the end this process was able to be accomplished by simply using the inbuilt functions that come with the EMGU and the opencv api. Most of the work to separate the ball from the surrounding environment is accomplished by using the binary thresholding method. Initially the RGB colour space was trailed to find the object. It was quickly found, and expected, that by using this colour space too much noise was introduced. To overcome this issue, the HSV colour space was used. This enabled a mask containing only the ball to be generated with good repeatability.

Whilst looking at the data coming back from the ball, it was noticed that the image would warp at the edges of the camera. This was due to the fish eye effect, something that is quite common in older cameras. To remove this noise, the standard camera calibration methods used by opencv, and in this case EMGU, were performed. This involves moving either a circular or checkerboard pattern in front of the camera so that the function can determine how the image is being warped and attempt to remedy the image. The function is able to accomplish this as it knows what it expects from the patterns, and can compare this to what it is receiving from the camera data. After this was preformed, it was found that the very edges of the image were still being distorted a small amount. It was determined that this is caused by the large dome that is placed infront of the camera in the robot design. To overcome this, the edges of the image were cropped in, so only the calibrated section of the image is being considered. Cropping the image also allowed for the removal of some corrupted data being received from the bottom of the video feed, as depicted in figure 1 and corrected in 2.

To be able to access the data a simple blob filter was used which returns the x and y components of the ball, relative to that of the camera. By analysing the data obtained

Figure 2: Image after distortion removal.

Figure 3: Unfiltered, information overlaid.

from the blob filter, it was clear that more processing was required. To solve this, the size of the blobs was filtered. One of the main advantages of using a spherical object like a ball means that any valid blobs that are of the ball must have a bounding box that is square in nature from all viewing angles. Therefore, any objects where the length and width of the bounding box not within %30 of each other were discounted as noise in the image. The camera that is attached to the VideoRay Pro 3 is of relatively low definition, meaning that the width and height of the bounding box were generally small, hence requiring a relatively large error margin. If a higher resolution camera was used, this range could be tightened to reduce noise even further using this technique. The size of the blobs was also limited, as if the blobs were too small they could be considered noise, as they might only be a couple of pixels in area, and were discarded. Conversely if the area was too big, it was again removed as the system might be detecting something like the floor or wall, even though if proper thresholding was conducted this should not be an issue. Unfortunately this did limit the range of the system, but it was determined that it was an acceptable loss, as when the tennis ball was only a few pixels on the screen, accurate calculations of its depth relative to the camera were not able to be performed.

Finally, as it is known that the edges of the object should be circular no matter the approach angle if a sphere is used, a hough circle transform was applied. As it was found that the hough circle transform is a relatively expensive method to apply to an image, a mask containing only the remaining blobs after the previous filters were applied was used. After significant testing it was found that the size and locations of the circles generated from this function could not be relied upon when the camera was moved

Figure 4: Filtered image of depth indicator.

around, away-from and towards the depth marker. It was discovered however that in all of these cases, the hough transform was able to detect at least a portion of the ball as a circular image. This method also meant that the entire ball had to be in frame for it to be detected, something the blob tracking method was lacking. Therefore, it was decided to combine both the hough circle method and the blob tracking methods together, by again filtering out any of the blobs that did not also contain a centre point detected by the hough circle method. After all of these filters have been combined together, the resulting blobs only contained that of the tennis balls located in the environment. There are some significant downsides to this method, such as if there was another object in the environment that appeared to be similar to a tennis ball in both shape and colour, it could be detected and therefore incorrectly used as a depth marker. It was determined that was an acceptable complication as when choosing an object to be used as the depth marker, these other environmental aspects should also be considered. Figure 3 and 4 depict the before and after shots of the image that is generated by the camera. It can be seen that by using this simple method, the ball can be easily extracted from the environment. The information depicted above the ball also shows how the depth of the object can also be determined, which is explained in the next section.

2.2 Calculating the Objects Position

Once the object has been found within the image the depth to the object, relative to the camera, can be obtained. In this case the advantage is that all of the information about the object is already known, such as the size of the object and colour of the object. Using this information, it is then possible to find where the object is in space relative to that of the camera. Firstly, the size of the object needs to made relative to that of the blob in the camera. This is a very simple process, after all of the aforementioned processing has been achieved, and can be accomplished outside of the water making it significantly easier to undertake. The depth object is placed a known distance away from the camera, preferably in the centre of the image. The average of the blob's side lengths is taken, and the distance away from the camera the ball currently is at is also found. Therefore, the distance that the ball is away from the camera can be found, at

any point in the view, by using the following equation:

$$depth = C_d \frac{2C_h}{B_w + B_h}$$

Where C_d is the distance used to calibrate the height of the ball, C_h is the height in pixels of the calibrated ball and B_w and B_h are the width and height of the detected blob respectively.

2.3 Finding the Depth Object in the local Frame

Once the depth of the ball has been determined in the global frame, the coordinates of the ball, in mm, needs to also be found in the local frame. Fortunately, assuming that the camera has been calibrated correctly, it can be assumed that each of the pixels will be a predefined width and height in mm. As the size of the blob will reduce the further away from the camera the object is, the information calculated in equation 1 does not need to be considered at this point, as only the centre point of the object needs to be found. As the blob detection method only provides the top right hand corner of the blob's bounding box, it needs to be shifted into the centre of the blob. This is achieved in the y axis using the following equation:

$$y_{coord} = \frac{H(2B_y - F_r)}{2B_w} + \frac{H}{2}$$

Where H is the actual height of the ball, B_y is the y coordinate of the ball, F_r is the amount of rows the frame contains, also the frame's height in pixels and B_h is the height of the bounding box. Similarly, the x coordinate of the can be obtained in mm by using the following equation:

$$x_{coord} = \frac{W(2B_x - F_c)}{2B_w} + \frac{W}{2}$$

Where W is the width of the ball, which should be about the same as the height of the ball if the object is spherical, B_x is the x coordinate of the bounding box and F_c is the amount of columns in the image, also known as the image width in pixels.

After the positional information of the object in the camera's frame of reference has been collected, a simple transformation matrix can be used to get this information into the global frame so that other robots within the swarm will be able to relate any objects that they detect to the calculated position of the depth indicator. This will also allow for a very basic depth map of the entire area to be quickly generated, as each robot contributes its own findings to this global depth map, negating the robots having to search over area that has been previously explored by other robots within the swarm. The only point of note that should be taken out of this is that the coordinate axis also changes, as when the camera is considered, the z axis is pointing away from the robot when it is sitting on a flat plane, and the y axis points up. This is in contrast to the global map, where generally the Y axis pointing away from the robot, and the z axis is indicating the depth/height of the robot.

3 Conclusion and Future Work

This paper has presented a method to both detect and an object that can be used by a robot for depth calculation, as well as how the position of this object can be calculated with the frame of reference of the camera. The position of these objects can then be transferred onto a global map so that multiple robots within the swarm can utilize this information to collect objects within a proximity to these markers.

Future work can be completed upon using this method. It has been found that when processing vision data under the surface of the water, in this case a pool, that the light source is diffused enough that adaptive thresholding has not been required. Unfortunately, in some scenarios this will not be the case, therefore a method of adaptive thresholding should be included to enable the algorithm to function in a more diverse lighting environment, such as caves or popular shipping channels.

Currently the range in which the objects can be detected is relatively short. It was found that the main reasoning behind this was that the resolution of the camera was such that the far away objects would simply become a couple of pixels and therefore indistinguishable from the noise in the environment. If a higher resolution camera was employed it would be possible to find the objects at a significantly longer distance.

This work will be expanded to allow the robots within the swarm to use this depth information to be able to collect objects on the same level as the depth markers. The depth markers are required as these objects will be ones such that their shapes are not uniform or constant, and therefore the distance away the robots are from these objects cannot be easily determined, unless objects of a known size are within their vicinity.

Finally, a filter, such as an extended kalman filter could be applied to the detected objects to remove any variation of the objects position, particularly their depth, that might be introduced by drift or other inaccuracies that is obtained by the detecting robots ability to localize itself within the environment, or introduced by the robots other sensors, such as their own depth sensor. This could also help if two or more robots detect a single depth marker, and place it in slightly different positions, potentially causing conflict.

[1]Work was supported, in part, by grant number FA8750-15-2-0116 from Air Force Research Laboratory and OSD through NCS&T State University.

References

[1] Y. Wei, Z. Dong, and C. Wu, Depth measurement using single camera with fixed camera parameters, *IET Computer Vision*, **6**, 29–39, (2012), 10.1049/iet-cvi.2010.0017.

[2] R. Mohedano, and N. Garcia, Robust multi-camera 3d tracking from mono-camera 2d tracking using bayesian association, *IEEE Transactions on Consumer Electronics*, **56**, 1–8, (2010), 10.1109/TCE.2010.5439118.

[3] J. Song, S. Na, K. Hong-Gab, H. Kim, and L. Chun-shin, A depth measurement system associated with a mono-camera and a rotating mirror, in Pacific-Rim Conference on Multimedia, 2002, pp. 1145–1152.

[4] J. Michels, A. Saxena, and A. Y. Ng, High speed obstacle avoidance using monocular vision and reinforcement learning, in Proceedings of the 22nd international conference on Machine learning, 2005, pp. 593–600.

[5] T. Nagai, T. Naruse, M. Ikehara, and A. Kurematsu, Hmm-based surface reconstruction from single images, in Image Processing. 2002. Proceedings. 2002 International Conference on, 2002, pp. II-561-II-564 vol. 2, (2002).

[6] C. Willert, and M. Gharib, Three-dimensional particle imaging with a single camera, *Experiments in Fluids*, **12**, 353–358, (1992), 10.1007/BF00193880.

[7] D. An, A. Woodward, P. Delmas, G. Gimelfarb, and J. Morris, Comparison of active structure lighting mono and stereo camera systems: Application to 3d face acquisition, in 2006 Seventh Mexican International Conference on Computer Science, 2006, pp. 135–141.

[8] G. C. Gini, and A. Marchi, Indoor robot navigation with single camera vision, in PRIS, 2002, pp. 67–76.

[9] M. Shao, T. Simchony, and R. Chellappa, New algorithms from reconstruction of a 3-d depth map from one or more images, in Computer Vision and Pattern Recognition, 1988. Proceedings CVPR'88., Computer Society Conference on, 1988, pp. 530–535.

[10] F. Liu, C. Shen, G. Lin, and I. Reid, Learning depth from single monocular images using deep convolutional neural fields, (2015).