Distance estimation using fixation on a 2DoF robot

Anonymous CVPR submission

Paper ID *****

Abstract

The scope of this project is to explore the capabilities of active perception when predicting depth information in the scene. In the past, previous works demonstrated that analysing one image at once was not really effective. However, the latest works are based on a more biological approach where fixation is carried out. This process is the act of moving the camera while looking (or fixating) an object. In this project we will try to create a simple setup that mimics the fixation with a 2 DoF Robot.

1. Introduction

The scope of this project is to explore the capabilities of active perception when predicting depth information in the scene. In the past, previous works demonstrated that analysing one image at once was not really effective. However, the latest works are based on a more biological approach where fixation is carried out. This process is the act of moving the camera while looking (or fixating) an object. In this project we created a simple setup that mimics the fixation with a 2 DoF Robot.
The approach presented uses a frame-based camera which will acquire RGB image frames and using computer vision algorithms the displacement of the tracked object on the camera frame is computed. Using this information, the robot is commanded to always maintain the center of the ball in the center of the image.
In the upcoming sections we discuss the existing work. Then we present each of the three (3) algorithms developed and description of each. Finally, the test setup and the results are presented.

2. Related work

Estimating the depth is an hot topic in robotics as knowledge of the surrounding world and the object to manipulate is necessary to successfully control the robot. In this method, we use a fixation-based vision approach where the visual servoing is implemented computing the angle needed to keep the object fixated while moving via image alignment.

2.1. Image alignment

In order to know how much we need to rotate the camera to recenter the object in the image frame while moving the robot, a image alignment algorithm detection has been used. Although our domain is slightly different our approach is based on the the method used in [3]. In this work [3], an optimization algorithm seeks the best transformation (translation, rotation, scaling, and deformation) to align one image with a set of images with each other.

2.2. Fixation-based vision

Previous studies (e.g., [1], [2], [4],) have demonstrated the effectiveness of fixation algorithms in predicting object depth and environmental information. Our approach follows similar principles to achieve accurate measurements of relative distances but on a simpler setting. Although the principle remains the same, the methods mentioned above employ considerably more costly hardware, resulting in greater precision compared to our results.

3. Method

A robot is positioned in front of a stationary object and moves along the x-axis, capturing two images before and after each displacement. In order to align the two images, our group developed 3 different approaches which are translation, rotation with (RICK), and rotation with (Juan). For the translation method, the images are aligned by obtaining the pixel displacement along the x-axis. With the pixel displacement, the optimal rotation angle is calculated to recenter the image. The rotation method is done by aligning two images to determine the optimal rotation required to re-center the object within the image frame. This rotation information is then used to extract depth data from the images.

The following sections describe the algorithms and the optical flow equations and rotation matrix which are commonly used throughout most of the algorithms.

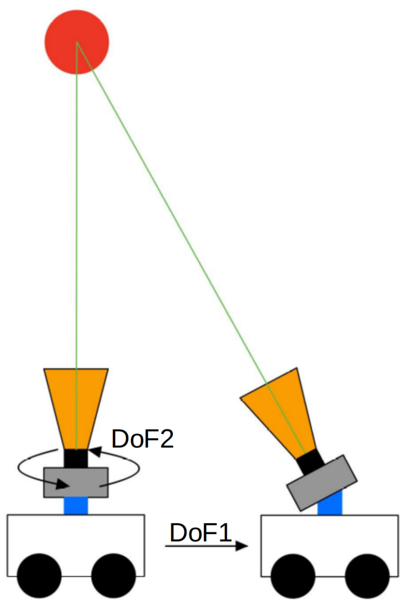

Figure 1. The robot rotates the camera after each displacement along the x axis

3.1. Matrix Based Rotation

3.1.1 Image alignment algorithm

Alignment of an object in an image is done by transforming an image and then comparing the dissimilarity to the original. Alignment is achieved once the dissimilarity reaches its smallest value. Therefore, we must slowly transform the image and compare iteratively until the alignment is finalized.

$$P' = M * P \tag{1}$$

Where P represents the original image coordinates and P' the transformed image. M denotes the transformation matrix, which includes the 2D-3D projection and the rotation [5].

$$M = T * R \tag{2}$$

The projection matrix, T, is expressed as shown below to account for the projection about the center axis of the image.

$$T = \begin{pmatrix} 1 & 0 & -w/2 \\ 0 & 1 & -h/2 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & w/2 & 0 \\ 0 & f & h/2 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3}$$

The transformation matrix for a rotation about the y-axis, as shown in Figure 2 is presented below:

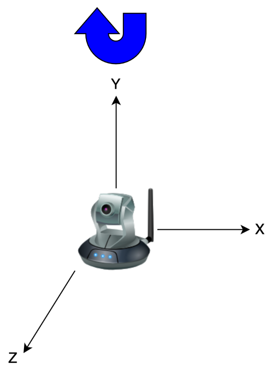

Figure 2. Camera coordinate plan

$$R = \begin{pmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

After projecting the image from 2D to 3D, we apply the transformation. However, the result remains in a 3D coordinate plane and we must revert the projection process. Therefore, taking advantage of matrix properties, we can find the inverse of the projection matrix:

$$T^(-1) = \begin{pmatrix} 1/f & 0 & (f-1)*w/(2*f) \\ 0 & 1/f & (f-1)*h/(2*f) \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

Once we obtain the transformed image, we then have to compare it to the original image, using Mean-Squared Error (MSE) method:

$$MSE(P, P') \tag{6}$$

Since we require an initial angle to produce a transformation, we must iterate by increasing the angle and then doing the comparison. We do this until the error value is minimized. Therefore, at each iteration the error must be decreasing, and once the error increases we revert to the last transformation, which produces the closes aligned image.
To further simplify the process we narrow the comparison of both images, by only comparing the area corresponding to the original image where the tracked object is located.

3.2. Rotation Flow

3.2.1 Image alignment algorithm

The process of aligning two images involves using a gradient descent algorithm to find the optimal transformation matrix that can be applied to the second image to align it with the first or vice versa. In our setup we aim to find only one parameter which is either the rotation around the y-axis or the translation along the x-axis. The optimization algorithm could be formulated as:

$$\theta^* = \arg\min_\theta f(\theta) \tag{7}$$

where $f(\theta)$ is the Mean Squared Error between the transformed image and the other one. For minimize this function, in every iteration the parameter θ gets update as follows:

$$\theta_{n+1} = \theta_n - \gamma \frac{\partial f(\theta)}{\partial \theta} \tag{8}$$

where learning rate γ is continuously getting adapted by the ratio between the current error and the maximum error over the whole optimization process and it is determined by:

$$\gamma = \frac{f(\theta_n)}{max(f(\theta_i))} \Delta\theta \tag{9}$$

and $\Delta\theta$ is maximum step. That allows us to adjust the size of our steps based on the distance to the minimum, taking larger steps when we are far away and smaller steps when we are close.

3.2.2 Image Transformation

To align the images, we have tried various methods that rely on using a transformation matrix to manipulate 3D points that have been projected from pixel coordinates.

In one of our approaches, firstly a basic perspective projection is used to map image pixels (x, y) onto a 3D world (X,Y,Y) space as follows:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f\frac{X}{Z} \\ f\frac{Y}{Z} \end{pmatrix} \tag{10}$$

where Z is set arbitrarily as in this case it will not influence the computations. Once we have the projection of the points in the 3D world frame (X, Y, Z) we apply the rotation matrix about the y-axis given the rotation angle θ:

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{11}$$

Finally, the 3D points are re-projected back to the image frame using the inverse of the formula 10 obtaining the transformed pixel location (x', y').

3.2.3 Depth estimation

Being the (U, V, W) the camera translation, (A, B, C) the camera rotation, (x, y) the image pixels, and Z the depth of the object, or goal is to compensate a translational motion with a rotation such that:

$$v_t = -v_r \tag{12}$$

where the translational flow v_t is

$$v_t = \frac{1}{Z} \begin{pmatrix} -fU + xW \\ -fV + yW \end{pmatrix} \tag{13}$$

and the rotational flow is equal to:

$$v_r = \begin{pmatrix} \frac{A}{f}xy - Bf - \frac{B}{f}x^2 + Cy \\ Af + \frac{A}{f}y^2 - \frac{B}{f}xy - Cx \end{pmatrix} \tag{14}$$

In our simplified setup, we assume that the robot's translation is limited to the x-axis ($U \neq 0$) and that rotation is performed around the y-axis ($A \neq 0$), with the object centered in the image frame ($x = 0$ and $y = 0$). With these assumptions, the expressions mentioned earlier can be simplified as follows:

$$-\frac{1}{Z} fU = fB \tag{15}$$

where the the depth information can be extracted as:

$$Z = -\frac{U}{B} \tag{16}$$

In our case, U refers to the robot's displacement along the X axis, while B represents the rotation angle required to align the second image with the first image.

3.3. Translation algorithm

Unlike the other two algorithms, the translation algorithm solely depends on the position of the objects in the image plane. The image alignment steps of the translation algorithm are listed below.

1. Apply the color mask to isolate the object (red ball) from the background
2. Extract the contour of the object from the masked image
3. Obtain the center of the contour
4. With the centers from the two images, calculate the distance between them
5. Substitute the distance to the equation 10 to find Z or the depth
6. Use equation 15 to obtain the optimal rotation angle

There are two main differences between the translation and the other approaches. The first difference is the alignment approach. The rotation algorithm is obtaining the optimal rotation angle first by aligning the two images through the rotation matrix. And from the angle, the depth is estimated. On the other hand, the translation approach is vise-versa of the rotation algorithms. It utilizes pixel displacement to be the main variable that requires to estimate the depth. With the depth, the rotation angle is obtained with the optical flow equation.

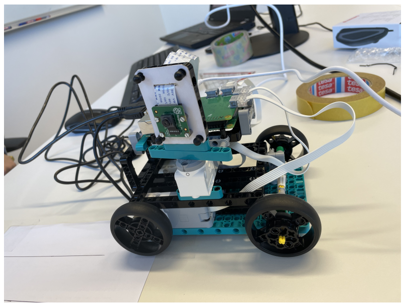

Figure 3. Lego Mindstorms Robot

4. Test Setup

To evaluate the 3-algorithms presented a lego-mindstorms robot, equipped with the pi-camera, was built. For the image processing a Raspberry-Pi 4 was used. The assembled robot is shown in Figure 3.

For this experiment a red ball was placed in front of a white board, and the robot was positioned at a predefined distance. The distances of 20, 40, and 60-cm were evaluated.
The experiment procedure is as follows:

1. Position robot at the appropriate distance and center camera. The robot movement shall be perpendicular to the camera FOV.
2. Capture image at initial position
3. Move robot to new position. Record distance.
4. Capture image at current position.
5. Execute congealing algorithm.
6. Rotate camera per the algorithm output
7. All previous steps shall be performed for 3 consecutive displacements.

This procedure shall be repeated for the 3 distances specified. The results for the 3 algorithms are shown in the next section.

5. Result

The group tested 3 different algorithms with the same setup. To verify the performance of the rotation and the depth estimation, 3 different depths, 20cm, 40cm, and 60cm, are selected. 3 camera positions are tested for each depth. The overall result of the experiment is shown in figure [4]. As labeled, the blue color is the actual distance measured with the measuring device. The Depth_Measurement_T denote for the result obtained by the translation algorithm. The Depth_Measurement_R1 is gathered with the Matrix Based Rotation Algorithm. Lastly, the Depth_Measurement_R2 is the result from the Rotation Algorithm. As shown in the trend of the graph, the measured depth, y-axis, diverge further away from the actual depth as expected. As all of the algorithms are highly depending on the pixel resolution of the camera, the farther the distance between the camera to the object would make less accurate pixel changes or variation. Therefore, such phenomena is unavoidable when using low resolution camera. Nevertheless, as shown in the graph, overall performance of the algorithms are acceptable as they do not vary that much from each other and with the actual depth.

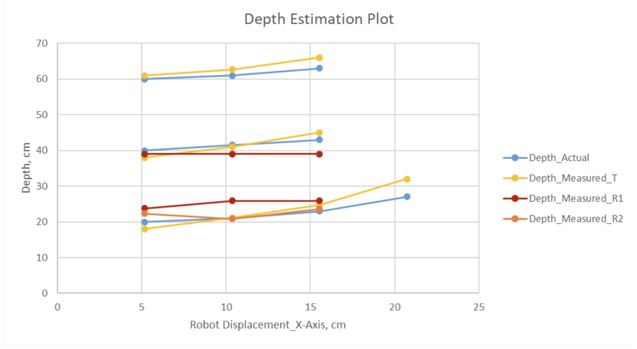

Figure 4. Depth estimation result plot

References

[1] Aravind Battaje and Oliver Brock. One object at a time: Accurate and robust structure from motion for robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3598–3603. IEEE, 2022. 1

[2] Angel Juan Duran and A.P. del Pobil. Robot depth estimation inspired by fixational movements. *IEEE Transactions on Cognitive and Developmental Systems*, 14:1356–1366, 2022. 1

[3] Erik G Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):236–250, 2005. 1

[4] Fabrizio Santini and Michele Rucci. Active estimation of distance in a robotic system that replicates human eye movement. *Robotics and Autonomous Systems*, 55(2):107–121, 2007. 1

[5] Axel Thevenot. How to transform a 2d image into a 3d space, Feb 2022. 2