Development of a Lightweight Visual-based Pose Estimation Sensor: Implementation and Validation

Renato Severiano
renato.severiano@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

November 2018

Abstract

Following the demand for platooning solutions on current motorways, the need for new sensors emerges. Vision-based systems are generaly very flexible, however there is a lack of methods tested for platooning.

This dissertation presents a vision-based system working as a sensor for detection of position and orientation of a circular marker. Putting a marker of this sort on the backside of a vehicle should be enough to perform the detection of that vehicle by the sensor. Physically, the sensor is composed by a Raspberry Pi 3 and a dedicated camera. The Raspberry Pi runs the detection algorithm making use of the framework Robot Operating System to manage data input and output aiding to minimise response times. The computer vision algorithm used on the sensor is based on an existing one, with some improvements added to the target detection part. This improvements resulted in an increase of efficiency of the algorithm, enabling an easier detection of the target using fewer computing resources.

As a way of sensor validation, its data was compared with data obtained with a motion capture system. In order to prove the usability of the sensor, it was used as the positioning sensor on a leader-follower system using two omni-directional vehicles in order to emulate a small platoon. Both tests demonstrated the effectiveness of the proposed solution.

Keywords: Computer Vision, Pose estimation, Fiducial markers, Targets detection, Shapes identification

1. Introduction

Mobile robots are being created and adapted to work in a number of places of our lives, from the household to the road. The iRobot Roomba is helping people to keep their houses cleaner without effort. Several companies, like Waymo, are working to take the human out of from the driver's seat. In the field of transportation, there is an EU project called "Safe Road Trains for the Environment" (SARTRE) whose goal is to have formations of several cargo trucks as a train. The head truck is driven by a human and the others are following it as big mobile robots. Its main goal is to increase the safety in motorways while having the convenient side-effect of increased fuel efficiency.

1.1. Previous Work

Cucci presents a planar target composed by concentric circles with the addition of smaller circles inside the circular section as a coding to differentiate distinct markers following the same system[2]. The target was designed to be accurately found by a small computer on-board of a UAV. The objective was to allow it to follow the target, which was placed on top of a van. The procedure followed by Cucci began by applying the Canny detector and then finding contours according to Suzuki *et al.* [4]. A new circle detection algorithm is presented, assuming ellipses as irregular circles. Then the code is tested for each outer circular section and ellipses are fitted to the inner and outer ellipses. As the projection of the circle centre is not the same as the centre of the projected ellipses, an accurate method for estimating the circle centre is presented. Then the pose is determined similarly to a Perspective-3-Points problem (P3P)[3].

1.2. Objectives

This dissertation has the goal of building a turnkey solution for the relative pose estimation of a target relative to a camera. With the only sensor used to locate the target being a monocular camera. The algorithm should be tested on a small board computer such as the Raspberry Pi or the Odroid as a ground proof that the algorithm can be easily fitted on a mobile robot for platooning applications.

1.3. Outline

The next section of this document introduces some theoretical concepts used throughout the subsequent sections. Then the methodology followed for

pose estimation on this dissertation is succinctly explored, with a highlight on an improved target detection method. The following section touches some key tools used. The subsequent section states the results obtained. Finally, the conclusions with some sugestions for future work.

2. Theoretical Background

2.1. Target Reference Frame

Depending on the target used, its reference frame is placed conveniently following the rules stated next:

- The origin O_T is placed at its geometrical centre;
- The plane xy is coincident with the target plane;
- The y axis should be positive to the side of the target plane considered as being the bottom;
- The x axis should be positive to the side of the target plane considered as being the right side;
- The z axis must be negative on the side the target is visible.

2.2. Camera Reference Frame

The camera reference frame is a Cartesian frame defined in (x, y, z). The origin O_C of the camera frame is placed on its optical centre, with the z axis pointing outwards the camera. The plane xy is parallel to the projection plane.

2.2.1 Projection Frame

The projection plane is where the image in a camera get's projected. The projection frame is defined by the plane $x'y'$ as a translation of the plane xy along the negative semi-axis z by the focal length f.

2.3. Pose

A pose is a coupling of position and orientation. A position can be defined as a translation vector an an orientation as a rotation matrix.

2.3.1 Rotation Matrix

A rotation matrix defines the rotation between two reference frames. It is formed by 3 successive rotations along 3 non consecutive axis. In this work, a rotation about a static frame on axes X-Y-Z was used taking Euler angles as starting point $\mathcal{R} = \mathcal{R}_z(\varphi)\mathcal{R}_y(\vartheta)\mathcal{R}_x(\psi)$.

$$\begin{bmatrix} c_\varphi c_\vartheta & c_\varphi s_\vartheta s_\psi - s_\varphi c_\psi & c_\varphi s_\vartheta c_\psi + s_\varphi s_\psi \\ s_\varphi c_\vartheta & s_\varphi s_\vartheta s_\psi + c_\varphi c_\psi & s_\varphi s_\vartheta c_\psi - c_\varphi s_\psi \\ -s_\vartheta & c_\vartheta s_\psi & c_\vartheta c_\psi \end{bmatrix} \quad (1)$$

3. Methodology for Target Pose Estimation

The methodology for estimating the pose was highly similar to the methods applied by Cucci [2]. The most evident differences are detailed in this document.

The solution presented by Cucci showed very promising results that could be applied for the objectives in hand. Both methods make use of the same family of targets, these are composed primarily by a pair of concentric circles.

The algorithm for detecting the target can be partitioned in 5 stages. In short, those stages are:

- Image preprocessing
- Finding contours in image
- Detecting concentric circles
- Looking for the right target
- Extracting the pose of the target

A diagram of the algorithm with the respective placing of the referred stages can be seen in Figure 1.

3.1. Circular Target

The target is composed by a black circle with a concentric white circle with about half the radius inside it. This canvas make up the base for detection of the target. On top of the black ring some smaller white circles can be placed on its medium radius region. This smaller circles create a region similar to a simple bar-code along a constant radius and varying angle. Figure 2 has 3 circular targets with distinct circle codes.

3.2. Ellipse Detector

The circles of the target are projected onto the image plane as ellipses. As such, a good ellipse detector is needed in order to detect effectively the target. The ellipse detector work on the contours found after appling the Canny edge detector [1][4].

A new ellipse classifier is proposed here to improve the target detection method. The classifier proposed is based on the comparison of areas. The first area is the contour area A_c, which is the number of pixels inside the contour. The second area is the area of the ellipse A_e inscribed on the rotated rectangle of minimum area that contains the full contour. If it is the contour of a perfect ellipse the difference of the two areas is zero. In order to tolerate a finite resolution and a reasonably low lens distortion, a relative error should be tolerated.

$$\frac{A_c - A_e}{A_e} < e_{max} \quad (2)$$

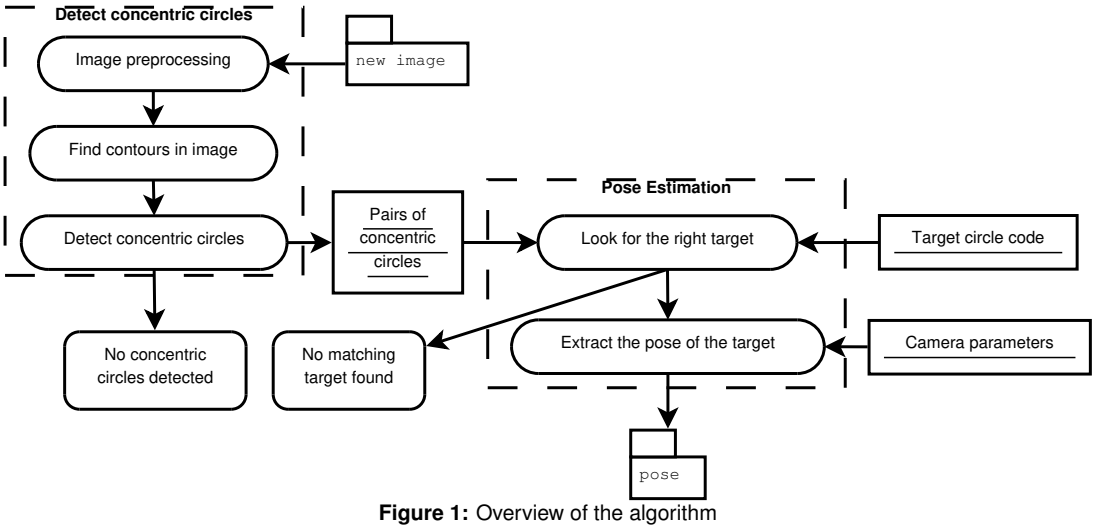

Figure 1: Overview of the algorithm

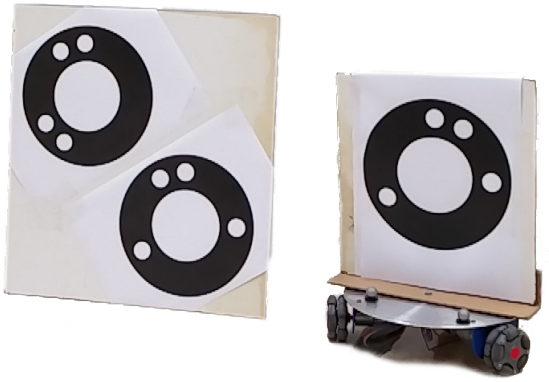

Figure 2: Left: plate with two dummy targets, **right**: OMNI-ANT equipped as Leader

To compare the enhanced method proposed with the one originally employed by Cucci, performance evaluation tests were conducted, to assess the ability of classification of elliptic shapes of both algorithms. Both algorithms ran on the same dataset, which has the same amount of elliptic shapes and non elliptic shapes.

Table 1: Test results of the previous ellipse classifier

Cucci		Actual	
		True	False
Result	Positive	479	0
	Negative	1635	1156

In Tables 1 and 2, the values in the "Result" rows represent shapes that were classified positively as ellipses or negatively as not ellipses, as a result of each classifier; the values in the "Actual" columns display the amount of shapes whose results were truly or falsely classified.

After fine tuning the threshold parameter to get the best results for both classifiers, the results shown in Table 2 were obtained, giving some evidence that the proposed classifier should be more reliable by failing to classify ellipses less often.

Table 2: Test results of the new ellipse classifier

Proposed		Actual	
		True	False
Result	Positive	1391	0
	Negative	1635	244

To quantify the quality of each classifier a performance measure is introduced. Accuracy is a quantity that can be directly compared in this situation, because the results were obtained using the same input data and also because there are as many positive input data as negative input data.

$$\text{Accuracy} = \frac{\text{total amount of truly classified data}}{\text{total amount of classified data}} \tag{3}$$

The method proposed by Cucci obtained an accuracy of $64,6\%$ and the method proposed here demonstrated an accuracy of $92,5\%$. Note that a blind classifier would have an accuracy of around 50%, so it is a large increase in classification performance.

In addition, the proposed approach has a considerable advantage in regard to computational efficiency. Comparing both solutions in respect to computing time, the total time for classifying with the proposed classifier was $122\,\text{ms}$ and the Cucci method took $570\,\text{ms}$, nearly five fold as much.

3.3. Proposed Algorithm

The proposed algorithm is inspired by Cucci [2]. It introduces a new method for classifying elliptic contours and adds instrumental enhancements on some parts of the algorithm, as described in the previous sections.

The target design and the solution to the problem were followed from Cucci without modifications, all the other steps were inspired by Cucci's findings

with more or less modifications.

The proposed algorithm achieves best results by following this steps:

Image steam stabilisation Open a stream of images from a camera with a constant rate, that rate should be adjusted to the resolution and computer used;

Image preprocessing Process each image, if its coloured convert it into greyscale;

Canny contour detector Detect all contours in the image being processed;

Concentric circles detection Find the contours that might define the outer circumference of a target;

Cross check code Check if the code if found on any of the possibly found targets by cross correlating the signal in the target with a reference signal, if it matches: the resulting lag indicates the z rotation of the target;

Find the centre Estimate the point of projection of the centre of the concentric circles of the target and refine it if needed;

Estimate the pose Compute the distance to the target and its rotations in x and y, the position of the centre of the target if given by $C^C = \overline{OC}\vec{v}_C$.

4. Implementation

4.1. Software tools

The computer vision algorithm was implemented in Python using OpenCV. The management of real-time data is done with Robot Operating System.

4.2. Omni-directional Vehicle

For demonstrations purposes it became convenient to use the promptly available platform OMNI-ANT. It is an omni-directional vehicle with a set of three wheels equally spaced mounted on a flat round aluminium plate.

This platform contains a low-level controller that accepts commands for velocity in x and y (in the directions indicated by the letters X and Y on the metal plate) and also a command for rotation speed control. The commands order changes in the reference velocity for the low-level controller, the command values are proportional but are not in any known units. The OMNI-ANT platform receives its input references wirelessly.

4.3. Final setup

Figure 5 depicts the setup used for follower in the final tests. It is composed by different parts:

Omni-wheels are designed to glide on its normal direction and grip when it is rotating, these are part of the OMNI-ANT platform;

Arduino is the low-level controller of the OMNI-ANT, receives velocity references and converts it into orders for the motors;

Raspberry Pi 3 is the small and lightweight computer that carries on the processing of the images captured from the camera and sends velocity references to the Arduino;

RPi Camera v1 feeds images to the Raspberry Pi for it to process;

Metal plate is the main structure of the OMNI-ANT;

Power-bank powers the Raspberry Pi with enough current with an embedded 5V regulated output;

LiPo battery pack powers the OMNI-ANT motors and their low-level controllers.

5. Results

5.1. Validation

In order to validate the proposed methodology, a concise and effective plan was designed:

- Reflective markers were placed around the camera and around the target, to allow detection of both with the robotic arena;
- The software of the robotic arena was set to start recording and the detector node was launched while recording the coordinate transforms calculated;
- The target or the camera were moved manually while trying to keep the target in the field of view of the camera.

The plots in Figure 3 and Figure 4 were obtained by following the mentioned procedure.

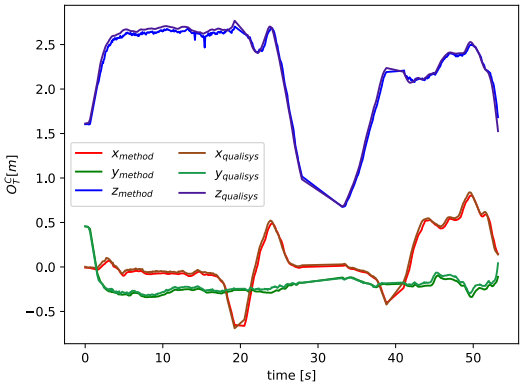

Figure 3: Position of target in the camera frame as detected by both methods

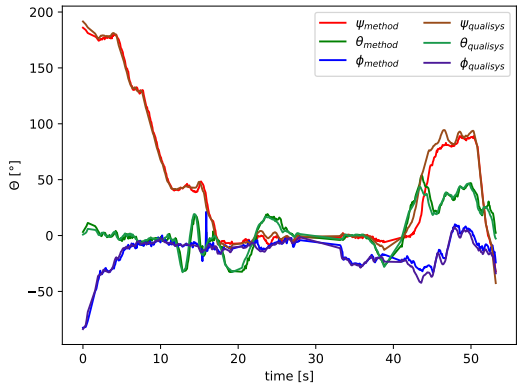

Figure 4: Orientation of target in the camera frame as detected by both methods

The standard deviation values in Table 3 were computed from the absolute error of each parameter in the same units as Figure 3 and Figure 4, taking the robotic arena values as ground truth. The standard deviation of z is higher than the standard deviations of x and y because for each computation the error of z is always larger that the x or y errors. The estimated translation errors are most frequently under the 5% margin, as the algorithm aims for. The error for ϕ is often double than the error for ψ and θ due to the nature of its computation, a lower error can be achieved by taking more data points to read the circle code. The ψ and θ values share close standard deviations as both are computed with the same steps, differing only in the values used.

The results obtained show that the solution proposed is on par with a much more sophisticated tracking system for the desired application, thus is well-suited for integration with mobile robotics applications, since the error is acceptable for most instances.

5.2. Limitations

From the starting position, different steps were taken to detect different limitations:

- Moving the **target** towards the camera until detection fails, to fetch the **closest** z distance possible;
- Moving the **target** outwards from the camera until detection fails, to fetch the **farthest** z distance possible;
- Rotating the **target** around its x axis, to extract the ψ detection limits;
- Rotating the **target** around its y axis, to extract the θ detection limits;
- Rotating the **camera** around its x axis, to find the yz field of view;
- Rotating the **camera** around its y axis, to find the xz field of view.

5.3. Testing Platooning

The ultimate testing procedure is to place the system on a platooning scenario to verify how it performs.

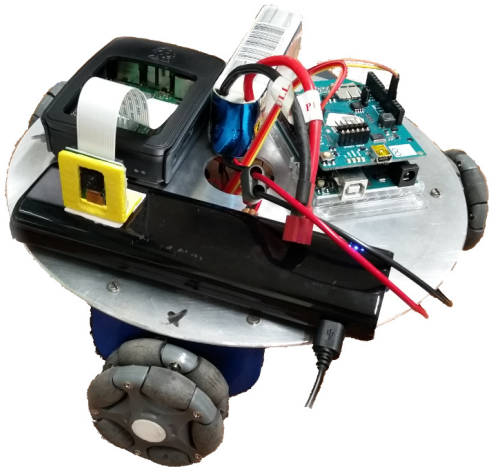

Figure 5: OMNI-ANT equipped as Follower

Using the proposed sensor as the only mean for the follower in Figure 5 to detect the position of the target mounted on the leader in Figure 2 a basic test platoon was structured. The leader was remote controlled using a smart-phone application.

A demonstration video can be watched at `https://youtu.be/5D9tKhgqPAQ`.

6. Conclusions

The objectives stated initially were globally accomplished, namely the turnkey solution for relative pose estimation with the possibility to be applied on platooning.

The preliminary validation of the algorithm by confrontation of its results with the robotic arena data proved it is a good pose estimation approach with better than expected results.

This solution was easily fitted on an existing mobile robot without adding any custom parts. It can be considered an inexpensive solution, as the sum of the parts added to the robot cost less than the Logitech camera used for validation alone.

As for the platooning, a preview demonstration was accomplished with only a leader carrying a target and a follower carrying the final solution with a power-bank powering a Raspberry Pi 3 with a camera module attached.

6.1. Contributions

Using Cucci [2] as baseline for a working marker detection algorithm, an improved method for binary ellipse classification was made with success.

A solution for autonomous OMNI-ANT vehicles was demonstrated with working results.

Table 3: Standard deviation values of the pose validation trial

$\sigma_x[m]$	$\sigma_y[m]$	$\sigma_z[m]$	$\sigma_\psi[^\circ]$	$\sigma_\theta[^\circ]$	$\sigma_\phi[^\circ]$
0.0296	0.0190	0.0435	3.34	3.82	6.31

Table 4: Limitations found when using the Raspberry Pi Camera v1

Resolution	Frequency	ψ Range	θ Range	z Range	xz FoV	yz FoV
640×480	10 Hz	$\pm 60^\circ$	$\pm 55^\circ$	0.6 m to 3.5 m	49.7°	36.8°

This document can be used as an introduction to ROS for the development of future work.

6.2. Future Work

There are several paths that can be taken from the point left by this work. Some works might focus on the algorithm while others might give more detail to the control scheme.

Using the same hardware solution: different algorithms, targets and control techniques can be employed and compared using this work as a baseline to improve upon.

Following the same algorithm several improvements could be achieved. One is particular could be the method for testing the target code, as cross correlation methods are very heavy for real-time applications. The approach employed by the unidimensional bar-codes seen everyday in product packaging could be studied to employ here as it would allow the identification of the target by a number instead of a signal, easing human readability while providing faster code checking.

The platooning preview can be extended to more vehicles, as there are more OMNI-ANTs available each could be mounted with a target facing backwards and the camera to the front. It could be coupled with a computer more powerful than the Raspberry Pi 3. It might also provide new control challenges by keeping each node independent, as each one should keep following the next while easing the movements to allow the previous to keep following.

References

[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, nov 1986.

[2] D. A. Cucci. Accurate optical target pose determination for applications in aerial photogrammetry. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3:257–262, jun 2016.

[3] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003.

[4] S. Suzuki and K. be. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, apr 1985.