3D cube-world construction robot

Final Report

C.H. Conroy
18072918

Submitted as partial fulfilment of the requirements of Project EPR402
in the Department of Electrical, Electronic and Computer Engineering
University of Pretoria

November 2021

Study leader: Mr. H. Grobler

Part 1. Preamble

This report describes work that I did *<to be completed>*.

Project proposal and technical documentation
This main report contains an unaltered copy of the approved Project Proposal (as Part 2 of the report).

Technical documentation appears in Part 4 (Appendix).

All the code that I developed appears as a separate submission on the AMS.

Project history
This project makes extensive use of existing algorithms on ... Some of the algorithms I used were adapted from ... Where other authors' work has been used, it has been cited appropriately, and the rest of the work reported on here, is entirely my own.

Language editing
This document has been language edited by a knowledgeable person. By submitting this document in its present form, I declare that this is the written material that I wish to be examined on.

My language editor was ______________________________.

______________________________ ______________

Language editor signature *Date*

Declaration

I, ______________________________ understand what plagiarism is and have carefully studied the plagiarism policy of the University. I hereby declare that all the work described in this report is my own, except where explicitly indicated otherwise. Although I may have discussed the design and investigation with my study leader, fellow students or consulted various books, articles or the Internet, the design/investigative work is my own. I have mastered the design and I have made all the required calculations in my lab book (and/or they are reflected in this report) to authenticate this. I am not presenting a complete solution of someone else.

Wherever I have used information from other sources, I have given credit by proper and complete referencing of the source material so that it can be clearly discerned what is my own work and what was quoted from other sources. I acknowledge that failure to comply with the instructions regarding referencing will be regarded as plagiarism. If there is any doubt about the authenticity of my work, I am willing to attend an oral ancillary examination/evaluation about the work.

I certify that the Project Proposal appearing as the Introduction section of the report is a verbatim copy of the approved Project Proposal.

C.H. Conroy Date

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

BRIEF	Binary robust independent elementary features
CNN	Convolutional neural network
FAST	Features from accelerated segment test
ORB	Oriented FAST and rotated BRIEF
RGB	Red, green and blue
RGBD	Red, green, blue and depth
ROI	Region of interest
SIFT	Scale-invariant feature transform
SURF	Speeded-up robust features
ToF	Time-of-flight

Part 2. Project definition: approved Project Proposal

This section contains the problem identification in the form of the complete approved Project Proposal, unaltered from the final approved version that appears on the AMS.

Part 3. Main Report

1. Literature study

The use of artificial systems to emulate tasks that humans find straightforward to perform, such as solving 2D puzzles, has been a long-standing practice since components of the solution system are often relevant in industrial applications [1]. This work focuses on a similar task that involves the construction of 3D shapes using small cubes. Such a task bears similarity to those tasks in the domain of pick and place robotics with the variation that object placement is dependent on the location of previously placed objects. Existing solutions in this domain typically consist of two primary components: a computer vision system to detect and localise the object of interest as well as a robot to alter the location and orientation of the object in 3D space [2].

A robot can be viewed as the combination of two core components, namely the robotic manipulator and the end-effector. The end-effector is the physical interface between the robot and the object of interest and is referred to as a robot gripper when its purpose is to grip the object to facilitate pose manipulation. The nature of the robot gripper depends on the physical characteristics of the object of interest and as such a wide variety of grippers have been developed. These include stiff finger grippers, flexible finger grippers, magnetic grippers and vacuum grippers which are best suited for objects with a flat surface [3]. The function of the robotic manipulator is to alter the position and orientation of the end-effector in 3D space. Robotic manipulators are categorised by the coordinate systems used to describe their movement mechanics which includes polar, cylindrical, articulate and Cartesian coordinates [4]. Cartesian robots have the benefit that accuracy of the robot is uniform throughout the robot's work envelope.

The purpose of the computer vision system is to detect the object of interest and localise it using the input image data captured from the robot's workspace, such that the robot has sufficient information to interact with the object. It is important that a distinction is made between object detection and object recognition. Object detection refers to the process of locating instances of a given object within an image while object recognition refers to the identification and classification of an object. This work is concerned with the former as the object of interest is known to be a cube. There are a wide range of approaches to the object detection problem. These can be broadly classified as being part of either the traditional computer vision domain or the deep learning domain [5]. In both cases, various techniques are used extract information from the image data in the form of features which are subsequently used to detect the objects of interest [6]. However, the distinguishing factor between these domains lies in the method of feature extraction. Traditional approaches incorporate a manual feature extraction step before the data is processed further. Deep learning approaches, on the other hand, integrate this step as part of the underlying model, such as a convolutional neural network (CNN). In this sense, such models can be viewed as highly integrated structures which take images as input after minimal preprocessing and produce the object recognition information as output.

An approach has been developed to detect generic rectangular cuboid objects in everyday scenes captured from a single perspective [7]. However, the generic nature of this task requires a highly sophisticated approach to achieve reasonable success.

The best solutions to computer vision problems that arise from unconstrained environments and require a great degree of generality are almost always found in the deep learning domain. However, when the problem is sufficiently constrained, solutions based on traditional techniques often exhibit performance that is comparable or even superior to that of deep learning approaches. In such cases, traditional approaches are often preferable since, unlike deep learning approaches, they do not require a massive training data set or a large degree of computational power. In general, a feature can be considered to be a piece of information present within the image input data. Edges, corners, blobs and ridges are examples of common low-level features that are considered within the traditional computer vision domain. Feature detectors are used to locate these fragments of information in the image input data. The Canny edge detector [8] and Harris corner detector [9] are examples of such methods which are popular for detecting edge and corner features respectively within an image.

The blurring of an image is a preprocessing step commonly employed with traditional approaches. This operation acts as a low-pass filter and filters out high-frequency noise which manifests itself as outlier pixel intensities. Improved performance of the feature detection stage is generally observed as a result. The conversion of an image to grey-scale is another common preprocessing step which reduces the complexity of subsequent operations when the color information of the image is insignificant. Thresholding is a useful method for obtaining shape-level feature information through image segmentation and is often applied following the preprocessing phase. The application of this technique to a grey-scale image results in a binary image which lends itself to further shape-level feature extraction. Since there are exposure inconsistencies that arise between images due to environmental light variability, it is usually prudent to incorporate an automatic threshold level determination mechanism when thresholding. In the ideal case, a grey image will exhibit a bimodal distribution of pixel intensities where the minima between the peaks corresponds to the ideal threshold value. However, such pixel intensity distributions are not necessarily guaranteed in most practical applications and, as a result, more robust automatic thresholding techniques have been developed, such as Otsu's method [10]. Existing automatic thresholding methods are usually classified as either histogram shape-based, clustering-based, entropy-based, object attribute-based, spatial or local methods [11].

Contours are another useful shape-level feature that can be used in service of traditional approaches to object detection. The bounding outline that captures the shape of an object in an image is considered to a contour. There are many different approaches to contour detection which, in general, can be categorised as either pixel-based, edge-based or region-based methodologies. Contours in images frequently correspond with discontinuities in grey-scale pixel intensity, particularly in the case of contours arising from luminance changes, which are detectable through the corresponding gradient magnitude information. A common approach to extract this information is to make use of a local filter which is convolved with the image. This results in a gradient space where the greatest gradient magnitudes are indicative of potential contours. However, this method is unreliable as it usually produces discontinuous contours and, therefore, often requires supplementary high-level feature information [12].

The contour detection problem is significantly simplified when the problem space is constrained to only binary images. In this case, gradient information is not required as with grey-scale images as pixel intensity discontinuities can be determined using only adjacent pixels. Furthermore, a binary image can be interpreted as consisting of a number of connected components where a connected component is defined as a set of pixels with identical intensity values which are interconnected through either 4-pixel or 8-pixel connectivity. Within this framework, the concept of a contour can be reduced to the sequence of pixels that define the boundary between adjacent but dissimilar connected components. The advantage of such contours is that they are guaranteed to be continuous, in contrast to the grey-scale image case. The border following algorithm is a longstanding approach to the detection of these binary image contours [13]. An extension to this approach exists whereby a more advanced border labeling method is employed to facilitate the extraction of topological structure information. Such information includes the hierarchical relationship between borders as well the distinction between outer and hole borders. This approach has also been adapted such that only the top-level outer borders in the hierarchy are detected which offers improved computational performance for applications that only require such information [14].

Contour detection, in conjunction with contour template matching, has been successfully used in robotic object detection and grasping applications using a single monocular camera [15]. There exist a number of other feature detectors which have applicability to cube detection. For objects with straight edges, the Hough transform is a useful image processing tool that can be used to capture these edges with parameterised straight lines in 2D space which is useful to determine the orientation of the object [16]. A more advanced and robust approach to determining useful features within an image involves the use of a feature descriptor which is a vector of values that describes the local region about a given image point. A number of feature descriptor algorithms have been developed such as the scale-invariant feature transform (SIFT) [17], speeded-up robust features (SURF) [18], features from accelerated segment test (FAST) [19], binary robust independent elementary features (BRIEF) [20] and finally the oriented FAST and rotated BRIEF (ORB) [21] algorithms.

The detection of an object within an image only forms the first stage in the robot's computer vision system. In order for the robot to interact with the object of interest in the physical world, the detected object needs to be localised such that its pose with respect to the robot's coordinate system is known. The object localisation methods available for robots when only red, green and blue (RGB) image input data is available can be categorised as either monocular vision and stereo vision approaches. With the monocular vision case, only a single RGB image is available as input at each time instance while in the stereo vision case, two or more RGB images are available [22]. The primary drawback of monocular vision approaches is the loss of depth information that arises during the projection of the 3D world onto a single 2D image. An additional piece of information, such as the size or world plane of the object, is required in order to recover the depth information. Stereo vision approaches, on the other hand, are able to recover the depth data based on the disparity between images that arises due to difference in pose of the cameras used to capture the images [23]. However, stereo vision approaches require more hardware and greater computational resources than monocular vision approaches. An alternative approach is to make use of a device that capture red, green, blue and depth (RGBD) data directly such as a time-of-flight (ToF) camera or integrated binocular stereo camera.

In order to relate the object detection information derived the camera input data to the world frame, the pose of the camera with respect to the world coordinate system needs to be determined. This information is represented by an extrinsic camera matrix which encompasses the rotation and translation parameters of the camera's pose with respect to the world frame. The extrinsic matrix can be used to map points in the world coordinate system to 3D camera coordinate system. In order to map points from the 3D camera coordinate system to the 2D homogeneous coordinates in the image, an intrinsic camera matrix is used [24]. Intrinsic parameters describe internal properties of the camera and are based on the pinhole camera model. These include the camera's inherent principal point offset, focal length and axis skew. The skew of the sensor axes occurs as a result of the optical axis not being exactly perpendicular to the sensor plane. However, for practical purposes this parameter is often discarded. The extrinsic matrix and intrinsic matrix can be multiplied to form the projection matrix which is used to project any point in the world frame to the image frame provided that the pinhole camera model is used and no lens distortion effects are present [25].

The intrinsic and extrinsic camera parameters need to be determined in order to make use of the pinhole camera model in practical applications. Camera calibration is used to estimate the intrinsic characteristics of the camera while camera localisation is used to estimate the extrinsic parameters of the camera. A popular approach to camera calibration involves the use of a planar pattern with known dimensions of which multiple images are captured at various different poses [26]. Either the pose of the planar pattern or the camera may be altered between calibration images. The application of this algorithm to a given set of such images will produce an estimate of the intrinsic parameters of the camera as well as the radial distortion of the camera. Real-world cameras have lens-induced distortion effects that are not included as part of the pinhole camera model. Radial distortion is observed when the degree to which light rays bend when incident on the lens is not consistent with the distance from the optical centre of the lens. Tangential distortion is observed when a degree of misalignment exists between the image plane and lens. These distortion effects are described by the radial and tangential distortion coefficients respectively [27].

2. Approach

- Contour detection is only concerned with outer border detection but does use the approach of border labeling over border marking - In this implementation, once the fiducials have been detected and the bounding region has been formed, it is assumed that all remaining contours are cubes or are a collection of cubes due to the constrained nature of the region of interest (ROI).

. . .

3. Design and implementation

3.1 Design summary

. . .

3.2 Theoretical analysis and modelling

. . .

4. Results

4.1 Summary of results achieved

...

4.2 Qualification tests

...

5. Discussion

5.1 Interpretation of results

5.2 Critical evaluation of the design

5.3 Design ergonomics

5.4 Health, safety and environmental impact

5.5 Social and legal impact of the design

6. Conclusion

6.1 Summary of the work completed

6.2 Summary of the observations and findings

6.3 Contribution

6.4 Future work

7. References

[1] G. C. Burdea and H. J. Wolfson, “Solving jigsaw puzzles by a robot,” *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 752–764, Dec. 1989.

[2] G. S. Sharath, N. Hiremath, and G. Manjunatha, “Design and analysis of gantry robot for pick and place mechanism with Arduino Mega 2560 microcontroller and processed using pythons,” *Materials Today: Proceedings*, vol. 45, pp. 377–384, Jan. 2021.

[3] G. Lundstrom, “Industrial robot grippers,” *Industrial Robot*, vol. 1, pp. 72–82, Feb. 1973.

[4] R. Miller, *Robots and Robotics Principles, Systems, and Industrial Applications*. New York: McGraw-Hill Education, 2017.

[5] The MathWorks, Inc., “Object recognition,” Accessed Oct. 30, 2021. [Online]. Available: https://www.mathworks.com/solutions/image-video-processing/object-recognition.html

[6] A. Kumar, “An overview of visual servoing for robot manipulators,” Control Automation, May 15, 2020. [Online]. Available: https://control.com/technical-articles/an-overview-of-visual-servoing-for-robot-manipulators/

[7] J. Xiao, B. C. Russell, and A. Torralba, “Localizing 3D cuboids in single-view images,” *Advances in Neural Information Processing Systems*, vol. 1, 2012.

[8] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679–698, Nov. 1986.

[9] C. G. Harris and M. J. Stephens, “A combined corner and edge detector,” in *Alvey Vision Conference*, vol. 15, 1988.

[10] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[11] P. Guruprasad, “Overview of different thresholding methods in image processing,” in *Proc. TEQIP Sponsored 3rd Nat. Conf. ETACC*, Jun. 2020, pp. 1–4.

[12] X. Y. Gong, H. Su, D. Xu, Z. T. Zhang, F. Shen, and H. B. Yang, “An overview of contour detection approaches,” *International Journal of Automation and Computing*, vol. 15, pp. 656–672, Jun. 2018.

[13] S. Suzuki and K. be, “Topological structural analysis of digitized binary images by border following,” *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[14] S. Yokoi, J. ichiro Toriwaki, and T. Fukumura, “An analysis of topological properties of digitized binary pictures using local features,” *Computer Graphics and Image Processing*, vol. 4, no. 1, pp. 63–73, 1975.

[15] H. Wei and B. Y. Chen, “Robotic object recognition and grasping with a natural background,” *International Journal of Advanced Robotic Systems*, vol. 17, pp. 1–17, Mar. 2020.

[16] N. Aggarwal and W. C. Karl, “Line detection in images through regularized Hough transform,” *IEEE Transactions on Image Processing*, vol. 15, pp. 582–591, Feb. 2006.

[17] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, p. 91–110, Nov. 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029664.99615.94

[18] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.

[19] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443.

[20] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary robust independent elementary features,” in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792.

[21] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[22] K. Liu, W. Shang, S. Du, and S. Cong, “6-DOF object localization by combining monocular vision and robot arm kinematics,” in *2017 36th Chinese Control Conference (CCC)*, 2017, pp. 6575–6580.

[23] P. Azad, T. Asfour, and R. Dillmann, “Stereo-based 6D object localization for grasping with humanoid robot systems,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 919–924.

[24] R. Szeliski, *Computer Vision : Algorithms and Applications*. London: Springer, 2011.

[25] OpenCV team, “Camera calibration and 3D reconstruction,” Accessed Oct. 31, 2021. [Online]. Available: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

[26] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[27] The MathWorks, Inc., “What is camera calibration?” Accessed Nov. 01, 2021. [Online]. Available: https://www.mathworks.com/help/vision/ug/camera-calibration.html

Part 4. Appendix: technical documentation

HARDWARE part of the project

Record 1. System block diagram

Record 2. Systems level description of the design

Record 3. Complete circuit diagrams and description

Record 4. Hardware acceptance test procedure

Record 5. User guide

SOFTWARE part of the project

Record 6. Software process flow diagrams

Record 7. Explanation of software modules

Record 8. Complete source code

Complete code has been submitted separately on the AMS.

Record 9. Software acceptance test procedure

Record 10. Software user guide

EXPERIMENTAL DATA

Record 11. Experimental data