Designing an innovative Vision for Autonomous Robots course

Arnoud Visser, Joey van der Kaaij, Qi Bi, and Shaodi You

University of Amsterdam, The Netherlands

Abstract. Robotics is a fast developing field, so it is important to keep robotics courses up-to-date. Without a good understanding of both the foundations and the latest developments, our students are not ready for the current scientific & societal challenges. With two new textbooks and a new robot, we designed a new master's-level robotics course. Our goal was to challenge students to demonstrate visual SLAM with their new robot by the end of the course, and to use the 3D-map generated in real-time to navigate a maze. That was an ambitious goal, yet the students demonstrated most of the components. This course design, with a build-up of the complexity with three assignments that were each challenging in their own right, could be an inspiration for other institutes that like to provide a state-of-the-art robotics vision course.

Keywords: Robotics Accessible for Everyone · Localization · SLAM.

1 Introduction

Robotics is a topic that is included in the curriculum of many different schools and universities. The focus could be on the electronics, mechanics, perception, or control aspect of the robot. Many textbooks try to cover all those aspects, although when multiple courses are given at the same university, it could be better to concentrate on one aspect.

At the University of Amsterdam, one robotics course is already given in the bachelor's degree in Artificial Intelligence, based on the textbook 'Introduction to Autonomous Mobile Robots' [15]. In six chapters, the fundamentals of mobile robots are covered. This course was accompanied by practical sessions based on a simulator and datasets. For the master Artificial Intelligence no such course existed, so a new one was designed. In the design we focused on the perceptual aspect, because this is the source of autonomous decisions, and apply the covered algorithms on a real mobile-robot platform (see Fig. 1).

Fig. 1. The RAE robot, build by Luxonis for OpenCV.

In 2023 OpenCV announced a new mobile-robot platform called RAE (Robotics Accessible for Everyone)[1], which had not only an impressive sensor suite, but also seemed very robust and within a price range that allowed a complete class of master students to have access to their own RAE robot (see Fig. 1). More details on the RAE robot in Sec. 4

2 Textbooks

In 2023 & 2024 two new editions of well-known robotics textbooks were published: 'Robotics, Vision and Control - Fundamental Algorithms in Python' [3] and 'Computational Principles of Mobile Robotics' [7].

2.1 Robotics, Vision and Control

The 'Robotics, Vision and Control' is a textbook with a long history [4], which started with creating a robotics toolbox [2]. In 2005, this toolbox was accompanied by a vision toolbox [5]. Those two toolboxes combined lead to the 1st edition of the 'Robotics, Vision, and Control' textbook.

The original toolboxes were MATLAB based. Porting them to Python was not trivial [4], yet once accomplished it allowed to create a Machine Vision toolbox which wraps both OpenCV and Open3D in a single user interface [10].

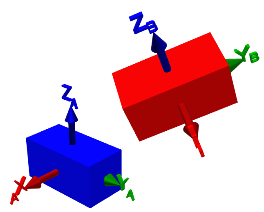

Fig. 2. A 3D-object moving from A to B. Sample code by Peter Corke.

The 'Robotics, Vision and Control' textbook is quite unique in the sense that it gives a more than solid theoretical background on algorithms, accompanied by the (python) code that created the illustrations and implemented the equations. The vision and control aspects are well integrated, in part because both need spatial math (the third toolbox provided by Peter Corke[2], partly because robotics and robotics vision involve the description of both position & orientation in 2D & 3D (see Fig. 2). The 2nd chapter of the 'Robotics, Vision and Control' textbook directly dives into subjects like transforms, twists and dual quaternions. Because both in Robotics and Vision movement is important, the 3rd chapter goes deeper into rigid-body transformation with speed and acceleration. Still, it is clear that the book is written by a robotics engineer, because at

[1] Kickstarter campaign, October 2023
[2] Spatial Maths for Python

the end of chapter 3 the transformations are illustrated with the inner working of Inertial Navigation Units.

As textbook 'Robotics, Vision and Control' was very useful for the theory, for the practical sessions a little bit less, because of the design decision to stay away from the Robotic Operation System ROS [4], which is the program interface of most current robots [12], including the RAE robot. ROS is known for its complexity and famous for its steep learning-curve, so there is some rational in this decision. Yet, it is also a missed chance, because one of the general concepts underlying ROS are the coordinate transformations from the transform library [9], which nicely match with the rigid-body transformations where the textbook starts with.

For our course, the chapters 13 and 14 of the textbook were most useful, where the topics Image Formation and Multiple Views are covered.

2.2 Computational Principles of Mobile Robotics

The textbook 'Computational Principles of Mobile Robotics' also has a long history, with a first edition published in 2000, based on a tutorial given in 1993. The third edition is quite recent [7], with for instance a new chapter on deep learning. Yet, for the practical sessions of our course, the most important aspect is the accompanying code[3], which is ROS-based. Unfortunately, the provided code was still in transition from ROS1 to ROS2 (some exercises based on ROS1 Noetic, other exercises based on ROS2 Foxy) which made the ROS learning curve even more challenging.

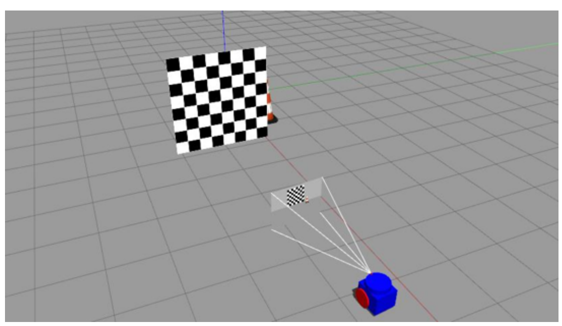

Fig. 3. An example of a simulated robot looking at a chessboard pattern. Courtesy Michael R. Jenkin.

Yet, the supporting material with the textbook contained a short introduction to ROS in general, which we combined with material from Francisco Martín Rico [13], to give our students a head start with the concepts behind the Robotic Operation System ROS.

[3] Code examples and exercises

Chapter 5 of this textbook covers visual sensors and their algorithms. The section on feature detectors covers classic edge detectors (see Fig. 3), which we actually used for our first assignment. Chapter 6 demonstrated how convolutional neural networks could be used to follow a line in the simulation. All exercises from this textbook were based on a simulated robot, while our ambition for the 'Vision for Autonomous Robots' course to do the practical sessions with a real robot. Yet, our first assignment was inspired by Jenkin, to demonstrate that a real RAE robot could follow a line in the ceiling, as suggested in Chapter 5.

3 Lectures

In the lecture series, a range of topics were covered. Although for each lecture the corresponding sections in the two textbooks [3,7] were indicated, the lectures themselves were a combination of slides from the courses Autonomous Mobile Robots & Vision Algorithms for Mobile Robotics, both from the ETH Zürich. Especially the topics covered in Vision Algorithms for Mobile Robotics were in line with our approach, concentrating on the geometrical aspects of vision.

Topics covered included:

- Perspective camera
- Camera calibration
- Lens distortion
- Edge detection
- Camera localization
- Kalman Filters
- Triangulation
- Epipolar geometry
- Structure from Motion
- Visual Odometry
- Visual SLAM
- Path Planning

Note that Peter Corke's textbook [3] also has a strong focus on the geometrical aspects of vision & robot control.

4 The RAE robot

Most commercial robots are now ROS based [12], so to allow the students to build up experience which such a ROS-based system, we selected the RAE. Luxonis built for OpenCV a mobile-robot platform called RAE (Robotics Accessible for Everyone). It is a robot that fits in your pocket (see Fig 1), which allows you to experiment with computer vision, machine learning and artificial intelligence.

It has a 4K camera at the front, stereo cameras at both the front and back and two wheels to drive around. The robot is an example of a differential-drive robot, controlled by the difference in speed between the two powered wheels. The balance of the robot is maintained by two additional passive rollerball wheels. All cameras are from the OpenCV AI Kit product line (OAK); the 4K camera is a IMX214 board, the stereo cameras are based on the OV9782 boards.

Quite specific to the RAE robot is its computing core, which is called Robotics Vision Core 3 (RVC3). The RVC3 was built on Intel's more experimental platform Movidius Keem Bay. Support for this intermediate version was limited, which also affected the support of the RAE robot. The support for the RAE was deprecated in 2024, resulting in minimal documentation that quickly became outdated. Luckily, the docker-image provided by Luxonis could be updated by the users, which allowed keeping the ROS-version inside the docker up-to-date.

Software-wise, the RAE platform was quite flexible. The major problems that the students encountered with the RAE robot were hardware-wise. Because the RAE robot is quite robust, it was difficult for the RVC3 to disseminate its heat. After a few minutes, the RVC3 reduced its core frequency from the initial 1.5 GHz, which affected the running ROS-services. In addition, the RVC3 running all ROS-services also depleted the internal battery quite fast.

5 Assignments

For the 8-week course, we created 3 assignments. The first assignment had to be finished in the 2nd week, the second assignment was extended from the 4th to the 5th week, the third assignment was reduced in complexity so that it could be finished in the 7th week. The grades of the second and third assignments were weighted twice as much as the grade for the first assignment.

The first assignment was designed to be a warm-up assignment, which allowed the students to become familiar with ROS and the RAE robot and its sensor suite. This process required more time and effort from our students, even with the tutorials provided. However, the experience gained during the first two weeks was very valuable in the following weeks.

5.1 Assignment I: Image Calibration & Line Following

Line Following is a classic task for a robot, although the RAE robot has two particularities that made it challenging enough for master students. The first challenge is that both the front and back cameras are slightly tilted upward (30 deg), resulting in limited visibility of the ground plane. The second challenge is the radial distortion of the fish-eye cameras, which causes straight lines in the real world to appear curved in the image (see Fig. 4).

Fig. 4. The ground view of the RAE robot. Courtesy Bekker *et al.*

This gave the opportunity to demonstrate the importance of a good calibration of a camera to the students. The students used OpenCV calibrateCamera to estimate the radial distortion of the RAE cameras. The OpenCV algorithm calibrateCamera is based on Zhang's method [19], which was covered in the lectures (together with Tsai's method [16]). OpenCV recommends at least 10 images to be recorded for the calibration, in the assignment description, we recommended at 30-40 images. In addition, it is important to cover all parts of the images, at different distances and with different orientations, with the chessboard pattern fixed to a solid surface.

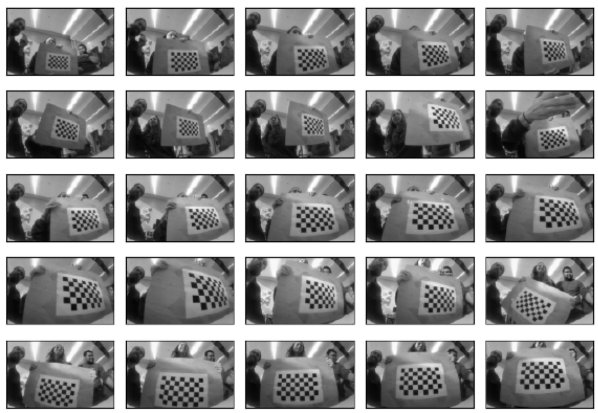

Fig. 5. A selection of images captured for calibration. Courtesy Bekker *et al.*

The students typically used 50 images for the calibration (see Fig. 5) and even repeated the calibration (with more diversity in the distances and orientations) until they were satisfied with the re-projection error. The students also experimented with other algorithms, such as OpenCV's fisheye . calibrate function, which uses Levenberg- Marquardt optimization to minimize re-projection errors. The result was not only a reduction of the mean reprojection error, but also fewer outliers. The focal length found with the fisheye . calibrate function was close to the 284.64 pixels reported by the Factory EEPROM of the StereoDepth camera.

The result of this calibration process was that the ceiling lights could now be recognized as bright straight lines (see Fig. 6). Note that not only the four ceiling lights are visible, but also the reflection on the right wall. Also, the windows at the left could be the sources of this sort of disturbance.

After that, the grayscale image is thresholded so that only high-luminance pixels remain, those pixels can be combined into the lines with the classic Canny edge detection algorithm [1]. Although the images are preprocessed, not all ceiling lights are connected, while also other regions of interest are still visible. (see the small red edges in Fig. 6). However, all ceiling lights have an equivalent orientation, which can be analyzed with the Hough transform [6]. The resulting

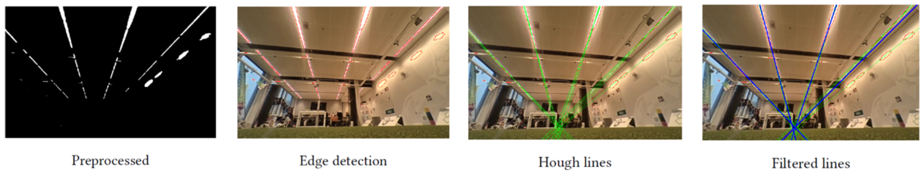

Fig. 6. Combining Canny edge with a Hough transform for line-detection. Courtesy de Jong *et al.*

clusters of Hough lines (indicated in green in Fig. 6) can be combined in a single line (indicated in blue in Fig. 6), which was found out to be more consistent across consecutive images and varied light conditions.

The selection of the most prominent line could also be done stochastically with an RANSAC algorithm [8], which gives even more robust results. Also, it is visible that the found ceiling lines all come together in an intersection point that is used by several student teams as the target to drive to when they have to align the RAE robot and follow the direction of ceiling lights. This was an approach chosen by several of the student groups (see Fig. 7).

Fig. 7. Using the vanishing point as target to drive the RAE robot to. Courtesy Sprott, Barták, Blokesch, den Braber and Kersten *et al.*

5.2 Assignment II: Detection & Localization

Localization is estimating the current position of the robot based on a provided map. For this assignment, the provided map was several AprilTag and ArUco markers, which were placed on objects around the field. The groundtruth position of each of these fiducial makers was measured with a digital laser rangefinder (Bosch Zamo 3). In total 24 of those makers were installed along the field, of different types and printed in different formats (big to small). As types of markers the 7x7 ArUco markers, the 36H11 AprilTag markers, and the MIP_36H12 ArUco markers were used.

From a good viewpoint, the markers could be easily detected, there are several libraries available specialized in detecting those markers. Yet, when the RAE robot started in the middle of the field, many of the markers were at the side and far away. Being at the side means that the makers were visible on the far edge of the image, which had as a prerequisite that the calibration of the camera was good enough so that radial distortion was not the source of error. The large distance made it hard to detect the markers printed in small formats, which forced the student to consider both false positives and false negatives. Examples of both events are visible in Fig. 8. In the left image, two small markers lower on the wall are not detected. In the right image not only false negatives are visible (due to obstruction or distance), but also a false positive above the curtain. With hyperparameter tuning, multi-scale processing and filtering with context information, the students were able to reduce the number of false positives and negatives, but incorporated in their localization algorithm that not always all markers that should be visible are actually visible.

Fig. 8. Examples of marker detection around the field. Courtesy de Jong *et al.*

The size of the fiducial markers was known, so both the angle and distance to the markers could be estimated. The students first measured the error in this estimation. The detection procedure could be optimized to a relative error of 10% in the distance estimate. Thereafter, the measurement of the distance was used to perform triangulation to compute the location from where the observation of multiple fiducial markers was made.

The goal of the assignment was to move as accurately as possible from a location in the center of the field to a location in the penalty area. While moving, the robot received observations of the markers (sometimes none are visible, sometimes two or three). Those observations were combined with the movement commands in an Extended Kalman filter to obtain a real-time position.

In most cases, the RAE robots could navigate accurately to the target location (see Fig. 9). However, not every behavior was very robust against disturbances. A few wrong observations could set the robot astray, which was difficult to recover from once the robot is no longer on its estimated position. A Kalman Filter is a good tracking algorithm, but has to be reinitialized once the track is lost.

5.3 Assignment III: Mapping & Route Planning

The last assignment was to navigate a maze, while building a visual map of the maze. Because our experience of the earlier weeks was that a lot of time

Fig. 9. Real-time updates of the location, and the final location (a point inside the penalty area, marked with a yellow duckie). Courtesy Becker *et al* and Barták *et al.*

was spent on getting good observations, this time a video of a drive through the mini-version of the maze was pre-recorded[4], which allowed the students to experiment with Structure-from-Motion and Visual SLAM algorithms directly. The recording through the full-size maze had to be recorded by the student teams.

As suggested, most students started with an implementation of the classic Structure-from-Motion algorithm, based on a graphical tool created in 2013 [18]. The graphical tool still functioned under Windows, for Ubuntu, an installation guide was created [17]. With this tool both a 3D point-cloud of both the mini- and full-size maze could be made (see Fig. 10).

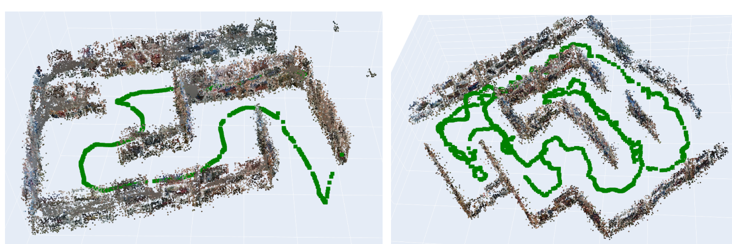

Fig. 10. 3D Reconstruction of both mazes with Structure-from-Motion, with additional points added in sparsely populated areas. Courtesy Bant *et al.*

The students also experimented with other Visual SLAM algorithms, such as COLMAP [14]. This allowed for an even more dense reconstruction (see Fig. 11). Because COLMAP depends on a more advanced matching algorithm, it was not computationally efficient enough to reconstruct the full-maze. As an alternative, the student teams looked to a more efficient version of the algorithm, such as GLOMAP [11].

[4] Small maze recorded with a RAE robot - doi:10.21942/uva.28554332

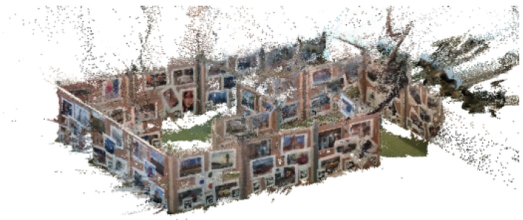

Fig. 11. A dense 3D reconstruction of the mini maze with COLMAP. Courtesy den Braber *et al.*

The 3D reconstruction could be projected onto a 2D grid. The students experimented with different data representations to represent obstacles and free space, such as a fixed grid, an adaptive grid or a nearest-neighbor graph implementation (see Fig. 12).

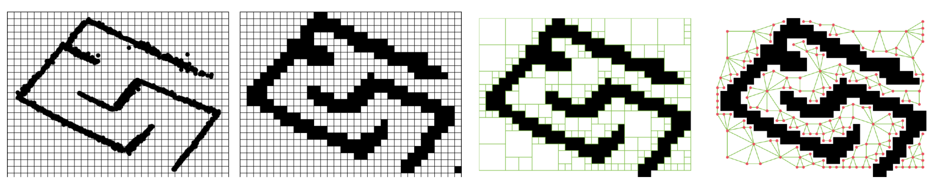

Fig. 12. Different representations to represent 2D free space. Courtesy Kersten *et al.*

Once a representation of the 2D free space is available (including enough margin to stay clear of the walls), path planning can be performed, including the classic Dijkstra and A* algorithm (see Fig 13).

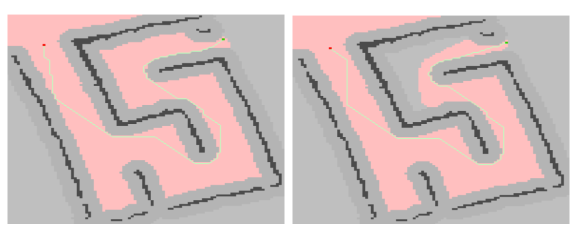

Fig. 13. Path-planning in the free space for the mini-maze. Courtesy Barták *et al.*

With such a planned path, the students could navigate the full maze, although this path was based on a previous generated map. There was not enough time left in the course to combine online mapping with real-time path-planning. This is not a surprise because well-known algorithms as Octomap, RTAB-Map and Cartographer can generate 3D-point clouds but are not designed to be used for navigation in real-time.

6 Conclusion

At the University of Amsterdam, a robotics course is designed, with a clear focus on the perception side. The master students were able to integrate the concepts covered in two recent robotics textbooks [3,7] directly on a recent RAE robot. After two weeks, the students switched without problems between ROS topics and were able to feed the sensor-streams into advanced OpenCV algorithms. So, a lesson learned for every lecturer who is preparing a course at this level: getting familiar with ROS doesn't conflict with going in depth with concepts of vision and control.

The major problem they encountered was with the design of the RAE robot. Due to its compact and robust design, the robot heated up fast, depleted its battery fast and was not able to send all sensor-streams in full resolution over wifi. The RAE robot as a product is also discontinued. As replacement we have already found another platform (see Fig. 14) with a more open design, more computing power and the same stereo camera as the RAE.

Fig. 14. The UGV Rover, build by WaveShare.

With this new platform, we like to work on the same assignments, which were perceived as interesting, challenging, and fun by the students. With those three assignments, topics like stereo vision, visual odometry and visual SLAM can be covered, which was the intention of designing this course.

Acknowledgement

We would like to thank all students who participated in the first edition of this course for their contribution. In this paper figures and highlights from their project reports were used: the illustrations are from Lisann Becker, Batu Helvacioğlu, Matey Krastev & Marina Orozco González; Juell Sprott, Robin Reitsma, Krijn Dignum & Thomas Komen; Gijs de Jong, Harold Ruiter, Macha Meijer & Jakob Kaiser; Patrik Barták, Jerrijn Mandjes, Erik Meijer & Raoul Ritter; Chimèene Blokesch, Lisa van Ommen, Simon Hüsgen & Niels Sombekke; Wouter Bant, Angelo Broere, Gidi Ozery & Valeria Sepicacchi; Thies Kersten, Thijs Stessen, David Boekestijn & Jochem van Oorschot.

References

1. Canny, J.: A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence **8**(6), 679–698 (Nov 1986)
2. Corke, P.: A robotics toolbox for matlab. IEEE Robotics & Automation Magazine **3**(1), 24–32 (Mar 1996)
3. Corke, P.: Robotics, Vision and Control: fundamental algorithms in Python, Springer Tracts in Advanced Robotics, vol. 146. Springer (May 2023)
4. Corke, P., Haviland, J.: Not your grandmother's toolbox – the robotics toolbox reinvented for python. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 11357–11363 (Oct 2021)
5. Corke, P.: The machine vision toolbox: a matlab toolbox for vision and vision-based control. IEEE Robotics & Automation Magazine **12**(4), 16–25 (Dec 2005)
6. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Commun. ACM **15**(1), 11–15 (Jan 1972)
7. Dudek, G., Jenkin, M.: Computational principles of mobile robotics. Cambridge university press (Feb 2024)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (Jun 1981)
9. Foote, T.: tf: The transform library. In: 2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA). pp. 1–6. IEEE (Jul 2013)
10. Haviland, J., Corke, P.: Robotics software: Past, present, and future. Annual Review of Control, Robotics, and Autonomous Systems **7**, 253–283 (Jul 2024)
11. Pan, L., Baráth, D., Pollefeys, M., Schönberger, J.L.: Global structure-from-motion revisited. In: European Conference on Computer Vision (ECCV). pp. 58–77. Springer (Nov 2024)
12. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y., et al.: Ros: an open-source robot operating system. In: ICRA workshop on open source software (May 2009)
13. Rico, F.M.: A concise introduction to robot programming with ROS2. Chapman and Hall/CRC (Sep 2022)
14. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 4104–4113 (Jun 2016)
15. Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D.: Introduction to autonomous mobile robots. MIT press, 2nd edn. (Feb 2011)
16. Tsai, R.: A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. IEEE Journal on Robotics and Automation **3**(4), 323–344 (Aug 1987)
17. Visser, A.: VisualSfM - Ubuntu installation. Tech. rep., Intelligent Robotics Lab - Universiteit van Amsterdam (Dec 2024)
18. Wu, C.: Towards linear-time incremental structure from motion. In: International Conference on 3D Vision – 3DV 2013. pp. 127–134 (Sep 2013)
19. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on pattern analysis and machine intelligence **22**(11), 1330–1334 (Aug 2002)