Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-22 (2018) 435–440

Robotic Autoscanning of Highly Skewed Ship Propeller Blades ⋆

Eirik B. Njaastad * **Njaal H. Munthe-Kaas** * **Olav Egeland** *

* *Department of Mechanical and Industrial Engineering, NTNU Norwegian University of Science and Technology, S. P. Andersens vei 5, NO-7491 Trondheim, Norway (e-mail: eirik.njaastad@ ntnu.no, njaalhm@ stud.ntnu.no, olav.egeland@ ntnu.no).*

Abstract: This paper presents an approach for a complete scanning of ship propeller blades. Inline quality control for advanced manufacturing or inspection of operational propeller blades requires automated and highly accurate 3D measurement systems. By combining an industrial robot with a high accuracy 3D camera and a laser distance sensor, the task at hand can be fully automated. The integrated system is described in detail, with focus on initial estimation of the propeller blade position and avoiding collisions with the blade while scanning. Validating experiments are performed where a highly skewed ship propeller blade is scanned with a resolution of 0.1 mm.

Keywords: Robot vision, Point Clouds, Precision measurements, Path planning, Intelligent manufacturing systems, Industry automation, Industrial robots, Reverse Engineering

1. INTRODUCTION

Measurement of the dimensions of a ship propeller is a time-consuming and skill-demanding task. It is usually performed by means of Coordinate Measuring Machines (CMM), preprogrammed from propeller design data sheets or from CAD models of the geometry. Due to the complex surface geometry, it is usually not feasible to do inline measuring of the propeller geometry during manufacturing. The 3D scanning process is further complicated by the highly reflective bronze surface of the propeller blades. Reconstruction or reverse engineering of previously made propellers is also complicated to automate, especially if descriptions or CAD data of the geometry is lacking.

Automation of the process of scanning propeller blades with a 3D camera has the potential to simplify and greatly reduce the time it takes to obtain geometry data of a manufactured propeller. It could also make it more feasible and effective to perform inspections of propellers in operation.

The presented work is part of a project that seeks to automate the process of casting propeller blades.

Various research have previously been done in the field of automatic 3D scanning of an object with a robot manipulator .

RoboScan, an automatic system for acquisition of complete 3D models using a laser-based 3D scanner mounted on an industrial robot is presented by Callieri et al. (2004). The system uses a known bounding volume of the object to be scanned to generate an automatic selection of scanner views and merging of range maps.

In Larsson and Kjellander (2008), a laser profile scanner mounted on an industrial robot is used for automatic scan path planning for scanning objects with curved surfaces. The method adapts to the shape of the object as well as to the constantly changing viewing direction of the scanner to find the scan orientations that produce the most accurate results. Other approaches based on laser profile scanners are more recently proposed by Chen et al. (2016) and Secil et al. (2017).

A line laser sensor is calibrated and integrated with an industrial robot to create a low-cost 3D scanning system in Yin et al. (2014). The system is used to automatically stitch together a point cloud from multiple scans of a car door panel.

In Ziegler et al. (2017), a low-cost RGBD sensor (Microsoft Kinect v2) in combination with a laser scanner mounted on a robot manipulator is used as part of a presented 3D scanning system. The acquired depth data is used to detect known objects and to automatically generate a robot trajectory in order to scan the objects.

Several methods have been proposed for autonomous view planning and collision-free path planning in robotic autoscanning. Recent examples include the work of Kriegel et al. (2015) and Phan et al. (2018). According to Zollhöfer et al. (2018), there are still many difficult open challenges that future work needs to address.

This paper presents an approach for a complete propeller blade 3D scanning process, including localization of the blade in the robot coordinate system as well as the initial point cloud processing after the finished 3D scanning.

The previous work presented assumes some prior knowledge about the boundaries of the measured object. The assumptions made about the scan object in this work, is

⋆ This work is supported by Oshaug Metall AS and the Norwegian Research Council under the OLIVER project, project number 261639.

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.11.595

Fig. 1. An overview of the robotic scanning system setup including the robot manipulator with 3D camera and a highly skewed propeller blade.

Fig. 2. Bounding spheres of the scanned point clouds are consecutively added to the robot simulation environment. Collisions with already scanned parts of the propeller blade can thus be avoided.

the fact that it is a propeller blade with a blade center axis running vertically upwards through a circular blade flange. It is also assumed that the robot can reach around the propeller in every direction.

The rest of the paper is organized as follows. Section 2 gives an overview of the robotic scanning system and the propeller blade used for initial testing. Section 3 presents a solution for initial estimation of the propeller blade position in the robot coordinate system. Section 4 gives an introduction to how the scanning process is performed. The details for how the resulting point clouds are processed are given in Section 5, while Section 6 concludes the paper and presents future work.

2. OVERVIEW

The general outline of the scanning process is as follows:

- Scan the part positioning table, on which the propeller blade is mounted.
- Estimate blade center axis and generate the initial scanning pose.
- Start scanning and continue navigation utilizing edge information for positioning the robot.
- Post-scanning point cloud processing with the goal of estimating propeller design parameters.

Measurements from a laser distance sensor are used for positioning the camera in the optimal working range before grabbing each image. Bounding spheres of the resulting point clouds are generated throughout the whole scanning operation in order to estimate the physical dimensions of the propeller. The bounding spheres are then input to a simulation which continuously analyzes robot reach and tries to predict potential collisions. The system also looks for blade edges for each camera pose.

2.1 Scanning setup

The physical scanning setup includes the following components:

- Zivid 3D camera with High Dynamic Range (HDR) imaging capabilities:
  A structured light 2.3 Mpixel 3D RGBD camera, with a field of view of 425×267 mm and a depth resolution of 0.1 mm at an image distance of 0.6 m (Zivid Labs, 2018). The optimal working distance of the camera is stated as between 0.6 and 1.1 m.
- Sick DT-35 Laser distance measurement sensor.
- KUKA KR 120 Industrial robot manipulator with its associated KRC4 robot controller.
- A central computer which processes the point clouds resulting from the 3D camera, generates the bounding spheres of point clouds, performs in-the-loop virtual simulation of the robot movements with reach analysis and collision detection, and controls the robot movements.

The setup is shown in Figure 1. Figure 2 shows the simulation environment with the generated bounding spheres.

2.2 Camera Calibration

The extrinsic camera parameters were found using standard functions in the OpenCV checkerboard calibration module (Bradski, 2000). The method used is based on Zhang (2000) and Bouguet (2002). A prerequisite is to have a good intrinsic calibration of the camera, since the calibration is done by using the RGB information and not the actual depth values.

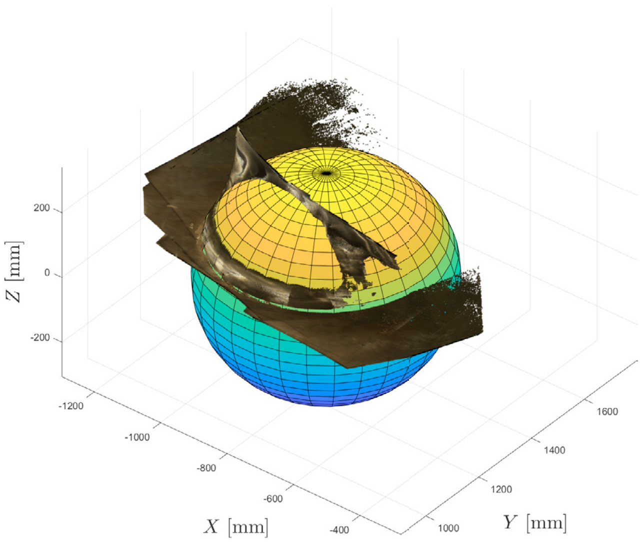


Fig. 3. The propeller blade center axis and radius of the blade foot is found by using MSAC to estimate a sphere from an initial safe-distance scan of the part positioning table. The sphere center point and radius are then used to set the starting pose for the propeller blade scanning.

2.3 The propeller blade

The propeller blade is designed for a Controllable Pitch Propeller with a total propeller diameter of 3600 mm. The blade is left-handed and cast in Ni-Al-Bronze. After the casting process the blade surface was polished. It is manufactured in accordance with the tolerances in ISO 484-1:2015 (E), accuracy Class 1. The blade has a skew angle of 29°, thus it is categorized as a highly skewed propeller. The blade is to be mounted on a propeller hub, along with three equal propeller blades.

3. INITIAL ESTIMATION OF BLADE CENTER AXIS

An initial estimate of where the propeller blade is located in the robot workspace is necessary in order to generate a meaningful scanning program and avoid collisions. The robot starts out by roughly scanning the top surface of a predefined positioning table from a safe distance.

The resulting point cloud is used for estimating a sphere located in the propeller blade foot centre. The estimation approach uses M-estimator SAmple Consensus (MSAC) (Torr and Zisserman, 2000) to find the sphere center point and radius. The MSAC algorithm is a variant of the RANdom SAmple Consensus (RANSAC) algorithm (Fischler and Bolles, 1987). An example of a sphere found in the blade foot center is shown in Figure 3.

The sphere center point $[x_c, y_c, z_c]$ is then assumed to be a point on the line through the center axis of the blade, with a unit vector pointing in the vertical direction. A cylinder with corresponding center axis is added to the virtual simulation as the initial guess of the blade bounding box, with a radius equal to that of the sphere. The starting pose for the following scanning sequence is then chosen as a point located at the proper scanning distance from the blade center axis, with the camera oriented with its principal axis perpendicular to the positioning table.

4. BLADE EDGE DETECTION AND NAVIGATION

The blade scanning algorithm is an uninformed (i.e. brute-force) grid based search algorithm. It generates the mapping grid (search tree) without using any domain-specific knowledge. The initial state of the algorithm is a pose automatically generated from the estimated propeller blade center axis. From there, the robot continues the scanning movements in an outward direction until an edge is detected. When detecting an edge, the robot moves up one scanning step length, and continues in the opposite direction. Before each scan is performed, the distance from camera to the surface is measured with the laser distance sensor. The camera is then moved to correspond with a fixed distance of 0.6 m from the propeller blade surface. The termination criteria of the algorithm is when a side and a top edge are found in the same image.

The camera is moved in the current scanning direction in predefined steps. In the scanning test performed in this project and with the 3D camera used, a good step length for safe navigation was found to be approximately 15 cm. Due to a specular surface, the HDR functionality of the 3D camera is used for each RGBD image. The Zivid camera has an adjustable diaphragm known as an iris diaphragm, used for taking 4 images with various iris opening in each camera pose. The 4 images are merged together to a single RGBD image before further processing.

4.1 Edge Detection based on Hough Transform

The Zivid camera outputs a pixel intensity image along with the $RGBD$ measurements. A Canny operator (Canny, 1987) is implemented to detect edges from the intensity image. Then, the Standard Hough Transform (Duda and Hart, 1972) is applied to detect straight lines at all orientations from the edges. In order to isolate features of a particular shape within an image, the Hough transform uses the parametric representation of a line:

$$\rho = x \cos\theta + y \sin\theta. \quad (1)$$

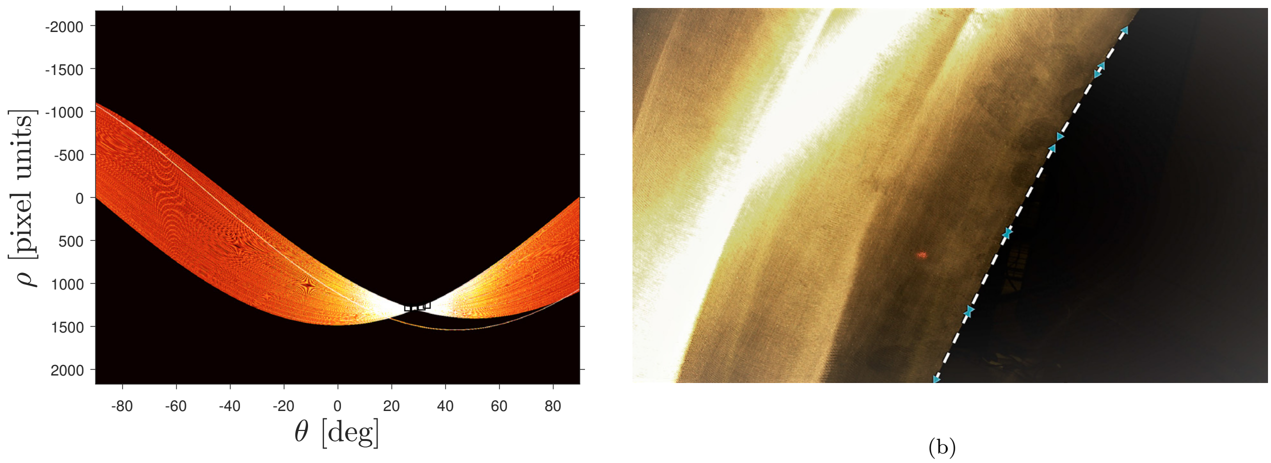
Figure 4a shows how the propeller blade edges are detected from the camera image using Hough transform. The underlying Hough matrix is shown in Figure 4b.

4.2 Edge classification

Based on the length and angle of the Hough lines found in the intensity image, the edges are classified as side or top edges. If the found edge is more horizontal then vertical (less than 45°), the edge is classified as a top edge. Likewise, a more vertical edge is classified as a side edge.

4.3 Estimating the Bounding Spheres

In order to do in-the-loop robot simulation and to avoid collisions, it is necessary to continuously keep track of the scanning results. By adding the information from the point clouds about where the blade is located with respect to the robot, a more accurate reach and collision



(a)

(b)

Fig. 4. Blade edge detection using Canny edge detector and Hough transform. In (a) the Hough matrix is shown. Peaks in the Hough transform are marked with squares. The resulting detected edge is marked on the input blade image in (b).

analysis can be performed before the robot moves to its next scanning pose. The optimal solution would be to compute the minimal bounding box. However, this is computationally expensive and a much easier solution is to compute a minimal bounding sphere for each point cloud. In this work, a near-optimal bounding sphere is found using Ritter's algorithm (1990). The computed sphere is about 5% larger than the ideal minimum-radius sphere.

4.4 Collision Detection and Reach Analysis

The robot simulation is performed in the simulation environment Visual Components 4.0 (Oy, 2000–2018). Via an application interface, the simulation is linked to the robot controller and point cloud processing program. Each robot movement is simulated based on the next desired scanning pose. The reach analysis and collision detection is performed before the robot is allowed to move to the next pose. Sequentially, the generated bounding spheres from each new scan are added to the simulation and volumetrically compared to the robot model geometry nodes. If the robot is not able to reach the desired scanning pose, the camera orientation angle around its principal axis is relaxed.

5. POINT CLOUD PROCESSING

The Cartesian reference frame of the propeller blade is defined so that the x-axis is positive along the propeller shaft axis in the forward direction of the ship. The y-axis is positive towards port of the ship. Since the coordinate system is left-handed the z-axis is positive upwards.

The propeller blade is aligned so that the z-axis runs through the center-point of the blade flange, and is coincident with the blade center axis. The origin of the coordinate system is in the center of the propeller hub, and for this project it is assumed that the hub diameter of the propeller is known.

The design parameters of the propeller are given for different sections of the blade at certain radii along the propeller reference line in the yz-plane, with the radii given as fractions of the total propeller radius from the origin to the largest span of the propeller. Each of the radial sections form a part of the surface of a cylinder with the specified radius and a cylinder axis coinciding with the x-axis.

To find these sections of interest in the point cloud of the propeller blade, an algorithm was implemented to segment out the points that are the correct distance away from the origin in the y and z directions. Since no points will be located exactly on the specified radial distance from the origin, a small threshold is applied to allow points that are within the threshold from the radial distance to be included in the segmentation. The result of the segmentation of these sections can be seen in Figure 6a. The figure shows the segmented cylinder sections at 0.3 to 0.95 of the propeller radius, with increments of 0.05, as well as the section at 0.975 of the radius.

To be able to calculate the design parameters for each cylinder section of the propeller blade, the cylinders need to be "opened out" as described by Carlton (2007). The points are then laying along a plane parallel to the xy-plane, and with a distance from the xy-plane equal to the radial distance of the cylindrical sections from the origin.

This is done by setting the new z-coordinate for each point in the section equal to the radial distance to the original point (r), and the new y-coordinate equal to the arc length spanning from the z-axis to the point with a radius of r.

If (x_1, y_1, z_1) are the coordinates of the original point in the cylinder section, the radius is equal to $\sqrt{y_1^2 + z_1^2}$ and the central angle (θ) of the arc equals $\cos^{-1}(z_1/r)$. The coordinates (x_2, y_2, z_2) of the flattened out section are found by:

$$x_2 = x_1, \tag{2}$$

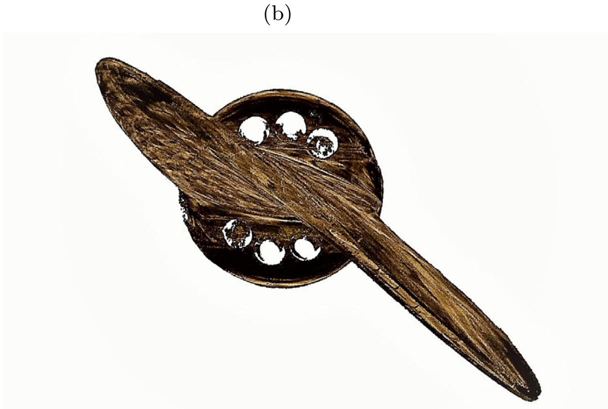
$$y_2 = \theta r \ \mathrm{sgn}(y_1), \tag{3}$$

$$z_2 = r. \tag{4}$$

The result of this process can be seen in Figure 6b. The figure shows the same cylinder sections as in Figure 6a,

(a)

(b)



(c)

Fig. 5. The resulting combined point cloud of the propeller blade. The propeller blade is seen from the front (a), side (b), and above (c).

but now the cylinders have been opened out to allow easier calculations of the propeller design parameters.

From each of these 2D sections as viewed in the xy-plane, the points that best describe the leading and trailing edges of the propeller blade can be determined. These points are essential to developing the chord line of the section which can further be used to calculate the different design parameters of the section, such as the chord length, pitch, skew, rake, as well as the thickness and camber distributions.

6. CONCLUSIONS AND FUTURE WORK

The purpose of this paper has been to present the integration of all operations necessary for a complete scanning process for the case of propeller blades. The described steps include initial propeller blade center axis position estimation, a summary of the scanning algorithm, as well as a description of how the resulting point cloud is processed.

While the different operations described in this paper have been functionally implemented and successfully experimented with, the integration into a complete process cycle remains to be done. Furthermore, some aspect of point cloud processing is still to be addressed in order to obtain the propeller design parameters - notably the estimation of the propeller blade origin which is not located on the blade itself, but in the center of the propellers rotational axis. Beyond that, the robustness and reliability required for an industrial installation has to be obtained. It is also beneficial to have external rotational axes in order to expand the usability of the system.

The aim of this system is to be able to reverse engineer propeller blades and potentially be used for inline inspection of propeller blades of various types. The highly skewed blade is one of the most complicated types of blades, which makes it a good test case for a future industrial-grade system.

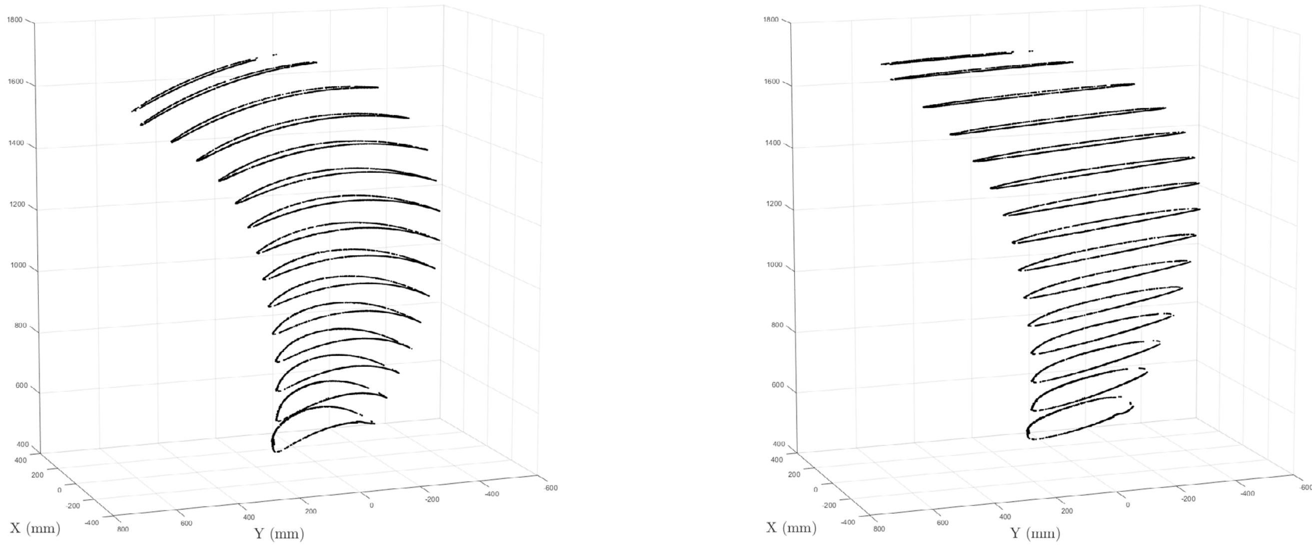
The developed system is able to 3D scan a highly skewed ship propeller blade with a resolution of 0.1 mm. The resulting point cloud can be seen in Figure 5.

From the results of the segmentation of blade sections, it should be possible to work further with the segments and try to estimate the design parameters of the propeller. The quality of these quantitative results will depend on the accuracy of the components in the scanning system, how well the point clouds are registered and aligned, as well as the level of noise in the data.

Until now, the point clouds are combined by using the robot's positional and orientation information. Experiments with external position measurement systems with better accuracy are planned.

ACKNOWLEDGEMENTS

The authors wish to thank the OLIVER project and industrial partner Oshaug Metall AS. The project is supported by the Norwegian Research Council under the BIA program, project number 261639. The authors would also like to thank Andrej Cibicik and Torstein A. Myhre for useful inputs.



(a) (b)

Fig. 6. In order to find the propeller design parameters, some post-processing of the resulting point cloud is necessary. The extracted cylinder sections from 0.3 to 0.975 of the total propeller blade radius are shown in (a). The propeller blade sections are opened out in order to calculate the propeller design parameters.

REFERENCES

Bouguet, J.Y. (2002). Camera calibration toolbox for matlab. *http://www.vision.caltech.edu/bouguetj/calib_doc/*.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Callieri, M., Fasano, A., Impoco, G., Cignoni, P., Scopigno, R., Parrini, G., and Biagini, G. (2004). Roboscan: an automatic system for accurate and unattended 3d scanning. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, 805–812. IEEE.

Canny, J. (1987). A computational approach to edge detection. In *Readings in Computer Vision*, 184–203. Elsevier.

Carlton, J. (2007). *Marine Propellers and Propulsion - Second Edition*. Elsevier Science.

Chen, L.C., Hoang, D.C., Lin, H.I., and Nguyen, T.H. (2016). A 3-d point clouds scanning and registration methodology for automatic object digitization. *Smart Science*, 4(1), 1–7.

Duda, R.O. and Hart, P.E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1), 11–15.

Fischler, M.A. and Bolles, R.C. (1987). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision*, 726–740. Elsevier.

ISO 484-1:2015(E) (2015). Shipbuilding - Ship screw propellers - Manufacturing tolerances. Standard, International Organization for Standardization, Geneva, CH.

Kriegel, S., Rink, C., Bodenmüller, T., and Suppa, M. (2015). Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects. *Journal of Real-Time Image Processing*, 10(4), 611–631.

Larsson, S. and Kjellander, J.A. (2008). Path planning for laser scanning with an industrial robot. *Robotics and autonomous systems*, 56(7), 615–624.

Oy, V.C. (2000–2018). Visual Components Premium 4.0: Proprietary. `visualcomponents.com/`.

Phan, N.D.M., Quinsat, Y., Lavernhe, S., and Lartigue, C. (2018). Scanner path planning with the control of overlap for part inspection with an industrial robot. *The International Journal of Advanced Manufacturing Technology*, 1–15.

Ritter, J. (1990). An efficient bounding sphere. In A.S. GLASSNER (ed.), *Graphics Gems*, 301 – 303. Morgan Kaufmann, San Diego.

Secil, S., Turgut, K., Soyleyici, C., Ozkan, M., Parlaktuna, O., Dutagaci, H., and Parlaktuna, M. (2017). A robotic system for autonomous 3-d surface reconstruction of objects. In *Control, Automation and Robotics (ICCAR), 2017 3rd International Conference on*, 188–191. IEEE.

Torr, P.H. and Zisserman, A. (2000). Mlesac: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1), 138–156.

Yin, S., Ren, Y., Guo, Y., Zhu, J., Yang, S., and Ye, S. (2014). Development and calibration of an integrated 3d scanning system for high-accuracy large-scale metrology. *Measurement*, 54, 65–76.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11), 1330–1334.

Ziegler, J., Gattringer, H., Kaserer, D., and Müller, A. (2017). Automated, depth sensor based object detection and path planning for robot-aided 3d scanning. In *International Conference on Robotics in Alpe-Adria Danube Region*, 336–343. Springer.

Zivid Labs (2018). Zivid 3D camera. `zividlabs.com`.

Zollhöfer, M., Stotko, P., Görlitz, A., Theobalt, C., Nießner, M., Klein, R., and Kolb, A. (2018). State of the art on 3d reconstruction with rgb-d cameras. In *Computer Graphics Forum*, volume 37, 625–652. Wiley Online Library.