

D I S S E R T A T I O N

Efficient Part Feature and Object Detection by Fitting Geometric Models to Range Image Data

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften unter der Leitung von

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Markus Vincze
E376

Institut für Automatisierungs- und Regelungstechnik

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informatikstechnik

von

Dipl.-Ing. Georg Biegelbauer
geb. am 24.01.1973
Matr. Nr.: 9325775
Erdbergstr. 52-60/1/5
A-1030 Wien

Wien, im Oktober 2006

Georg Biegelbauer

Abstract

Industrial automation is characterized through the use of robots utilizing large batch size productions. Indeed, the trend in the last decade points towards the flexible use of the robot for pure customization up to lot size one production. Sensors plays in this content a major role for an autonomous behaviour, where vision sensors get more and more important to acquire the robot's environment. Laser range sensors are predestinated to precisely obtain range images of the objects to be handled. In industrial applications mainly single-view range images can only be obtained and therefore, the challenging image processing task is to handle the object- and self-occlusions.

Therefore, reliable object detection is necessary for flexible robot handling, especially for grasping tasks. Due to the one-view characteristic of the range images the rear-side of the objects is not present and dealing with this restriction a model-based approach is desired for an object detection in 3D space. In the literature there are a lot of expensive methods fitting volumetric object descriptions to range data, but industrial requirements need fast and robust detections. Hence, this thesis investigates efficient part feature and object detection methods by fitting geometric models to range image data, suited for industrial use.

The work contributes with a low- and a high-level processing detection approach. The first approach introduces a novel RANSAC-based sequential cylinder fit for a fast bore hole detection. This part feature detection of industrial parts is needed for a sensor-guided robotic inspection application, where an endoscope has to be reliably inserted into a bore hole for a quality check of the inner bore surface. The second approach presents an efficient object detection method for robotic grasping tasks. The object modeling with one or several Superquadrics describes the approximated overall shape of the object. The approach is embedded in a hierarchical two level search structure, where randomly pose hypotheses are generated. These hypotheses are then verified using three quality of fit criteria within a voting procedure.

The proposed methods are described in detail and the experimental evaluation demonstrate the effectiveness and robustness in terms of a rapid and reliable approach. Comparisons between the two methods themselves as well as with standard fitting techniques show the potential to be effectively used in an industrial environment. At the end of this thesis an outlook describing the prototype's fair presentation and feasible applications of the introduced methods are given. As outcome a new concept of a fully automated object handling framework should be introduced, closing the gap between shape learning, object detection and tracking and should show that this research work is again the basis for new investigations.

Kurzfassung

In der industriellen Automation ermöglicht der Gebrauch von Robotern die Produktion von großen Stückzahlen. Allerdings versuchte man im letzten Jahrzehnt, durch den flexiblen Einsatz des Roboters, die Stückzahlen zu reduzieren oder gar die Losgröße Eins zu erreichen. In diesem Zusammenhang spielen jedoch Sensoren für das eigenständige Verhalten des Roboters eine wichtige Rolle, wobei den bildgebenden Verfahren für das Erfassen der Roboterumgebung eine besondere Bedeutung zukommt. Laser Sensoren sind dafür besonders gut geeignet, da diese die zu handhabenden Objekte präzise erfassen können. Allerdings ist es in den industriellen Anwendungen oft nur möglich die Objekte von einer Seite zu scannen, wobei die dadurch entstehenden Abschattungen für die Bildverarbeitung eine besondere Herausforderung darstellen.

Deshalb ist eine verlässliche Objektdetektion für den flexiblen Robotereinsatz, besonders aber für Greifoperationen, notwendig. Aufgrund der Tatsache, dass die Objekte nur von einer Seite gescannt werden, fehlt die Information der Rückseite und ein modellbasierter Ansatz zur Objektdetektion ist notwendig. In der Literatur findet man viele aufwendige Methoden volumetrische Objektbeschreibungen in Tiefenbilddaten einzupassen, jedoch verlangen industrielle Anforderungen schnelle und robuste Objektdetektionen. Deswegen beschäftigt sich diese Dissertation mit industriell einsetzbaren und effizienten Methoden zur Teile- und Objektdetektion durch Einpassen von geometrischen Modellen in Tiefenbilddaten.

Als wissenschaftlichen Beitrag dieser Arbeit wird sowohl ein grundlegender als auch einen komplexerer Ansatz zur Objektdetektion präsentiert. Als Erstes wird ein neuartiger RANSAC-basierter, sequenzieller Ansatz vorgestellt, Zylinder für eine Bohrloch-Detektion einzupassen. Diese Merkmals-Detektion von industriellen Werkstücken ist notwendig, um eine sensorgeführte, endoskopische Qualitätssicherung von den Bohrungsinnenwänden mit einem Roboter durchzuführen. Der zweite Ansatz stellt eine effiziente Objekt-Detektionsmethode für die Anwendung des Robotergreifens vor. Die ungefähre Form des Objekts wird hierbei mit einem oder mehreren Superquadrics modelliert. Die Methodik besteht in einer zweistufig hierarchischen Suchstruktur, deren erste Stufe zufällig Positionshypothesen generiert, die dann mittels drei Qualitätskriterien in einem Auswahlverfahren verifiziert werden.

Die in dieser Arbeit vorgestellten Methoden werden im Detail beschrieben und die experimentelle Evaluierung zeigten die Effektivität und Robustheit der Ansätze hinsichtlich ihrer Geschwindigkeit und der Zuverlässigkeit. Vergleiche zwischen den Methoden einerseits und den Standardtechniken andererseits zeigt das Potential der Ansätze, diese im industriellen Umfeld einzusetzen. Der abschließende Ausblick beschreibt einerseits die

Messepräsentation des Prototyps, als auch die möglichen Anwendungen, die sich aus den vorgestellten Methoden ergeben. Als Ergebnis dieser Arbeit wird noch ein vollautomatisches Konzept zur Objektverarbeitung vorgestellt, das die Lücke zwischen der visuellen Formerkennung, der Objekt-Detektion und -Verfolgung schließt. Dies soll zeigen, dass diese Forschungsarbeit ihrerseits wieder die Basis für weiterführende Arbeiten darstellt.

Acknowledgement

I would like to express my gratitude to Markus Vincze for supervising my project employment and for his support and advise during the thesis work. I am also indebted to Helmut Pottmann for his review of this thesis as second supervisor. I further want to thank my parents for their support during all the years of my study and finally, I am grateful to all the people who supported me with valuable comments and discussions or with their patience and appreciation.

Contents

1	Introduction	1
1.1	Motivation and Problem Statement	1
1.1.1	Feature Detection for Automated Inspection	2
1.1.2	Object Detection for Robot Handling	2
1.2	Contributions	4
1.2.1	Laser Stripe Profiler Calibration	4
1.2.2	Rapid Cylinder Fitting	4
1.2.3	Superquadric-Based Object Detection	5
1.3	Thesis Overview	5
2	State of the Art	7
2.1	Parametric Model Description	7
2.1.1	Geometric Primitives	8
2.1.2	Generalized Cylinders	11
2.1.3	Superquadrics	12
2.1.4	Geons	16
2.1.5	High Parametric Models	17
2.2	Model Fitting	22
2.2.1	Least-Squares Metric	22
2.2.2	Nonlinear Least-Squares Methods	25
2.3	Model-Based Object Detection	30
2.3.1	Cylinders	30
2.3.2	Superquadrics	31
3	Automated Industrial Production	33
3.1	FlexPaint	35
3.1.1	The FlexPaint Idea	35
3.1.2	Detection of Process Relevant Features	36
3.1.3	Planning of the Robot Painting Motions	37
3.1.4	Anti-Collision Detection of the Robot Motions	38
3.2	Lot Size One Calibration	39
3.2.1	Calibration Approach	40
3.2.2	Multi-View Calibration Object	40
3.2.3	System Calibration Method	41

3.3	FibreScope	43
3.3.1	System Overview	43
3.3.2	Robust Bore Localization	46
3.3.3	Bore Inspection	47
4	Bore Hole Detection	48
4.1	Method Description	49
4.1.1	Preprocessing and Normal Vector Calculation	50
4.1.2	Range Image Segmentation	53
4.1.3	Sequential Cylinder Fit	55
4.1.4	Bore Pose Determination	58
4.2	Experimental Evaluation	59
4.2.1	Sensor Setup	60
4.2.2	Algorithm Performance Evaluation	65
4.2.3	Experimental Detection Results	71
4.2.4	Experimental Performance Comparison	77
5	Object Detection	81
5.1	Method Description	82
5.1.1	Preprocessing: Dominant Plane Removal	83
5.1.2	Hierarchical Object Detection	84
5.1.3	Compound Model: Connectivity Verification	88
5.2	Experimental Evaluation	88
5.2.1	Sensor Setup	89
5.2.2	Simple Model Description	98
5.2.3	Composite Model Description	110
5.2.4	Bore Hole versus Model-Based Detection	114
6	Applications	117
7	Conclusion	122
	Glossary	124
	Bibliography	126

List of Figures

2.1	Qualitative representation of the hierarchical coherence of the parametric models ranked by the number of parameters faced to their shape complexity.	8
2.2	Set of geometric primitives: from (a) to (e) sphere, cylinder, cone, torus, cube.	8
2.3	Illustration of a generalized cylinder. (a) A generalized cylinder is defined as a volume formed by sweeping a cross-section along an axis. b) A generalized cylinder with an arbitrary shaped axis and the displayed sweeping function [29].	12
2.4	Basic Superquadric shapes varying ε_1 and ε_2 with $a_1 = a_2 = a_3$.	14
2.5	Visualization of taper deformation using a cube-shaped Superquadric ($\varepsilon_1 = \varepsilon_2 = 0.3$ and $a_1 = a_2 = a_3 = 1$).	15
2.6	List of the 36 basic Geons introduced by [8].	17
2.7	Hyperquadrics with (a) 5 faces. (b) 5 faces and one $\gamma_i < 1$. (c) 6 faces. (d) 6 faces and one $\gamma_i < 1$ from [28].	18
2.8	Example of a fitted screwdriver shaft using an implicit polynomial [63].	20
2.9	Composite fit of an Egypt bust using implicit polynomials of the 4 th , 6 th and 8 th order [15].	20
2.10	Example of spherical harmonic surfaces with the degree of 1, 3 and 7 [21].	21
2.11	Some examples of spherical harmonic surfaces used in medical and biological applications: (a) cardiac [117], (b) insulin [82], (c) protein [107].	21
2.12	Illustrating the difference between the algebraic and Euclidean distance illustrated using an ellipse.	23
3.1	Example of parts to be painted automatically: (a) car mirrors from FICO, (b) ROSSI gearbox with motor, (c) FINI compressor tank, (d) and a truck steering column from MAN.	35
3.2	Flow Chart of the FlexPaint system.	36
3.3	Detection of process relevant features out of the captured range image (a): cavities (b) and rib section (c).	36
3.4	(a) Generation of the main faces on the motor and gearbox (reduced VRML format). (b) Calculated paint strokes located on the main faces.	38
3.5	(a) Original paint trajectory generated by the PaintPlanner. (b) Paint trajectory after it has been modified by the ToolPlanner.	39

3.6	(a) Calibration object viewed from the bottom side in the darkened sensor cell. Note the projection of the laser plane on the aluminum profiles. (b) Sketch of the sensor cell consisting of four cameras and an arbitrary laser plane.	41
3.7	(a) Simulated range images of the non-calibrated sensor setup in Figure 3.6(b). The range images from the cameras are color coded. (b) Result of the calibration performed on the simulated data set.	43
3.8	(a) The robot arm of the FibreScope prototype. (b) The sensor setup consists of a circular scanning sensor, the endoscope, the CCD-camera and the clearance check.	44
3.9	Flow Chart of the bore inspection sequence.	44
4.1	Sketch of the processing steps of the first bore hole detection approach by fitting a circle to the bore boundary.	49
4.2	Flow chart of the bore hole detection approach.	50
4.3	Scanned range image with the indicated region of interest.	51
4.4	Close up of the calculated normal vectors in the region of interest.	52
4.5	Sharp corners lead to a systematic calculation error causing unwanted normal vector deviations.	52
4.6	Illustration of the normal vector calculation in the case of non planar boundary area, causing normal vector deviations.	53
4.7	Illustrating the segmentation process. (a) Gaussian image representing a noisy cylinder surface and a plane. (b) Result of the segmentation process separating the cylinder surface fragment and the plane.	54
4.8	Idea of the radius calculation by circumscribing a triangle built of three randomly picked points of the segmented bore surface.	57
4.9	Result of the axis fit. Note the visualized normal vectors whose ends determine the bore hole axis.	58
4.10	Final result of the bore hole detection process.	59
4.11	Cross section of a cast part where radial notches are visible which have an effect to the cylinder fit result.	59
4.12	Circular scanning sensor from Oxford Sensor Technology.	60
4.13	Close up of the scanning process (a) and the acquired range image (b). Note the laser beam rotation.	61
4.14	Drawing of the sensor setup showing the dimensions and the world and sensor coordinate system.	62
4.15	Working principle of the circular scanning sensor. Note the main part of the sensor, the rotating asymmetric lens.	63
4.16	Drawing of the CSS simulation.	64
4.17	(a) Industrial cast part with a bore diameter of $38mm$. (b) Simplified CAD model, notable the detailed modeled bore surface with several notches. (c) Synthetic range image of the circular scanning sensor.	64
4.18	Test bore with a diameter of $10mm$ to evaluate the algorithm performance. Note, the segmentation result of the plane and bore surface.	65

4.19	Averaged absolute deviation of the orientation estimation.	66
4.20	Boundaries of the maximum and minimum deviation of the estimated orientation.	67
4.21	Averaged estimated radius.	68
4.22	Boundaries of the maximum and minimum estimated radius.	68
4.23	Cylinder surface raw data points with (a) 1254 points (full-resolution), (b) 631 points, (c) 313 points, (d) 124 points and (e) 50 points.	69
4.24	Least-squares circle fit of the 2D mapped full resolution bore surface raw data points.	69
4.25	Comparison of the RSF (10^3 samples) and the LSF method concerning the processing time and the accuracy.	70
4.26	Range image of a bore with a diameter of 10mm. Note, the segmentation result of the plane and bore surface.	71
4.27	Timing bar chart of the apportioned processing steps according to Figure 4.2 of the bores #1 and #2 of Table 4.4.	72
4.28	Detection results of metallic cast parts. Note the disturbing raw data points caused by a radial notch in the bore.	73
4.29	Bore detection of a metallic high speed cutting part. The specular metallic edge caused a lot of ghost points in the range image. Nevertheless, the ghost points do not have an effect to the detection result.	74
4.30	Detection results of a bore with a rounded chamfer ($r = 2.5mm$) (a) and a 45° slanted bore (b).	74
4.31	Experiment on a noisy range data detecting a bore with a diameter of $12mm$	75
4.32	Two experiments investigating bores in close relationship and a bore with notches whose diameter is larger than the scan area.	76
4.33	Experiment demonstrating the detection-problem of thick-walled pipes.	76
4.34	Cylinder and Superquadric fits to the raw data points of the bore surface fragment of Figure 4.23(a). (a) Top row: Initial pose estimation using PCA and fit results after 10 and 200 iterations. Bottom row: Again fit results after 10 and 200 iterations, but started with the CAD pose. (b) Model recovery of a tapered Superquadric. (c) Cylinder fit result with a cylindrical constrained Superquadric.	77
4.35	Evaluation of the cylinder and Superquadric fit performance on full-resolution and sub-sampled raw data sets.	79
5.1	A wooden bowl (a) and its approximated Superquadric model (b).	81
5.2	Flow chart of the proposed object detection approach.	82
5.3	(a) Intensity image of a typical scene consisting of a ceramic and a wooden bowl, a mug, a screwdriver and a rubber-mallet. (b) The corresponding range image is shown with the ground plane detected, indicated with the green raw data points. Note the typical camera and laser shadows of a single-view scan.	83
5.4	Illustration of $n = 10$ RANSAC pose estimations.	85
5.5	Refinement steps (red) and final candidates for pose verification (green).	86

5.6	Final result after the voting process.	87
5.7	The main axis steradian α_k and the distance d_k between the Superquadric centers describe the connectivity of a composite object (rubber-mallet in Fig. 5.3(a)).	88
5.8	Drawing of the sensor cell design with its rough dimensions and labeling.	90
5.9	The sensor cell assembly is presented on the left image in overview and on the right image in detail showing the ranger camera, the color camera and the laser source.	90
5.10	In the top row the processing steps of a scan are shown. First the range image is captured before the picture of the mallet is taken. After that the color texture is mapped onto the range image data (bottom picture).	91
5.11	Visualization models of the camera, the CCD-chip and the laser source with its local coordinate frames (red: x -axis, green: y -axis, blue: z -axis).	92
5.12	On the left the modeled object scene saved in the STL-format is shown. On the right this scene is visualized with the virtual sensor setup.	93
5.13	Synthetic range image of the modeled object scene in Figure 5.12. Note that the objects are considered transparent underneath.	93
5.14	Setup of the mobile scanning head development consisting of the pan-tilt head, the laser source, the sensor base and the camera.	95
5.15	Schematic drawing of the mobile scanning head design.	96
5.16	Diagram showing the transformation between the sensor and the world coordinate system.	97
5.17	Range image showing a typical table top scene consisting of a plate with cookies, a knife, a tea pot and a mug.	98
5.18	(a) Experimental setup of a scene consisting of a cylinder, a box, a sphere and a cone. (b) Virtual one-view scan of the geometric primitives.	99
5.19	Examples of true positive detections of the four geometric primitives.	100
5.20	(a) Percentage of true positive detections faced to the number of hypotheses and (b) the averaged corresponding computational effort.	100
5.21	Histogram of the absolute orientation deviation of the detection results compared to the ground truth orientation.	101
5.22	Plot of the absolute position deviation of the true positive results compared to the ground truth position.	102
5.23	Histogram of the distance d distribution.	103
5.24	Relative position of the sphere detection results shown in the xy - and yz -plane using the methods A–D.	105
5.25	Relative position of the cylinder detection results shown in the xy - and yz -plane using the methods A–D.	105
5.26	Inaccurate cylinder detections classified as false negative detections using the RANSAC method with refinement (B).	106
5.27	True positive detection rates and the processing time shown for the sphere and the cylinder model comparing the methods A–D.	106

5.28	(a) Three spheres are presented with radii of $48mm$, $50mm$ and $52mm$. (b),(c) Added Gaussian noise with $\sigma = 0.3mm$ and $\sigma = 3mm$ presenting a true positive detection result.	107
5.29	True positive detection rates facing different noise level.	108
5.30	Gaussian noise ($\sigma = 5mm$) corrupted scene of Figure 5.18(b).	108
5.31	Detection results of some geometric primitives occluding each other.	109
5.32	Detection of an occluded wooden bowl.	110
5.33	Sensor prototype scanning a desktop scene and presented detection results of small objects on the table – top left: a computer mouse, top right: a sponge, bottom left: a cup, bottom right: a mug.	110
5.34	Detection of a mug with different handle orientations, modeled with two Superquadrics.	111
5.35	Two detection examples of the scene in Figure 5.32(a) – (a) a coffee mug and (b) a rubber-mallet.	112
5.36	Detection result of a rubber-mallet in a cluttered tool scene of similar objects exploiting the relation of the two modeled Superquadrics.	113
5.37	Detection of an arbitrarily shaped object consisting of three similar paper-roles. (a) Model to be detected. (b) Occluded and cluttered scene. (c) Detection result.	113
5.38	Method comparison with two examples. (a) Bore #1: Cast part of Figure 4.1(a). (b) Bore #2: Slanted bore hole. Both bores have a diameter of $11mm$ and the raw data points which are processed are marked red.	114
5.39	Bore detection results of of bore #1 using 100 attempts of the LSF (blue +) and the RSF (red ×) method.	115
5.40	Bore detection results of of bore #2 using 100 attempts of the LSF (blue +) and the RSF (red ×) method.	115
6.1	Prototype of the bore inspection system presented at an industrial fair. . .	117
6.2	Concept of our perceptual system: The fully automatic sequence starts with the object capturing where the size and shape parameters are gained that are used for subsequent object detection and tracking in an occluded and cluttered scene.	118
6.3	Experiment 1: Fig. (a) to (c): Capturing the object’s parameters; Fig. (d) to (f): Detection of the object in the scene; Fig. (g) to (j): Some tracking frames.	119
6.4	Experiment 2: Handling of a cylinder. Capturing model (first row), Detecting (second row) and Tracking (last row). The reprojected pose is depicted as mesh-grid.	120
6.5	Experimental 3: Fig. (a) and (b): Learning the object in a scene with no disturbances; Fig. (c) and (d): Detection of the object in the cluttered scene; Fig. (e): Starting pose of the object in the scene for the tracker; the pose is reprojected and depicted as white lines; Fig. (f): A frame of the tracking sequence. Note the white dots marking the interest points.	121

List of Tables

2.1	Summary of the geometric primitives' shape and pose parameter.	9
2.2	Superquadric description of basic shapes with dependency on ε_1 , ε_2 , a_1 , a_2 and a_3	13
2.3	Geon classification attributes.	16
4.1	Summarized results of Figure 4.19 and 4.20 with the absolute value of the estimated axis orientation, the margins of deviation and the outlier-limits for 10^3 and 10^5 number of samples.	67
4.2	Summarized results of Figure 4.21 and 4.22 with the estimated radius, the margins of deviation and the outlier-limits for 10^3 and 10^5 number of samples.	68
4.3	Summarized results of Figure 4.25 of the probes seen in Figure 4.23 with the processing times and the margins of deviation of r of the RSF and LSF method.	70
4.4	Summarized key data and processing time of the bores in Figure 4.18 and 4.26.	72
4.5	Comparison of the accuracy and the computational effort of a cylinder fit using LSF and RSF method.	78
4.6	Summary of the recovered Superquadric and cylinder parameter in comparison with the LSF and RSF method.	79
5.1	Summarized size and shape parameter of the model description in Figure 5.1.	82
5.2	Size and shape parameters of the geometric primitives in Figure 5.18.	99
5.3	Ground truth pose parameter of the primitives in Figure 5.18. \mathbf{p} denotes the Superquadric's center and \mathbf{v} the Superquadric's z -axis vector.	100
5.4	Summarized results of Figure 5.22 given the mean of the position deviation and the distance d and the standard deviation of the distance d	102
5.5	Summarized results of the detection results testing the methods A–D including the averaged processing time.	107
5.6	Summary of the algorithm and Superquadric parameters of the object detection results in Fig. 5.31, Fig. 5.32 and Fig. 5.33.	109
5.7	Summary of the algorithm and Superquadric parameter of the object detection results in Figure 5.34 and 5.35(a).	111
5.8	Performance evaluation of object detection in Figure 5.36 and summary of the algorithm and Superquadric parameter.	112

5.9	Performance evaluation of the object detection in Figure 5.37 and summary of the algorithm and Superquadric parameter.	113
5.10	Summarized results of the evaluated comparison of bore #1 and #2.	116

Chapter 1

Introduction

The dissertation addresses the problem of three dimensional feature and object detections in range image data. In 3D computer vision, objects are usually scanned by laser range scanners or stereo cameras, and 3D coordinates of points on object surfaces are obtained. A large number of points are often desired in many computer vision tasks. In order to minimize the efforts in many applications of computer vision a model representation of higher granularity is needed. Such an object representation step is indispensable to robot manipulation tasks or other processes where robots are involved.

This thesis gives a background in industrial automation utilized by the use of robots. Exploiting the robot's flexibility 3D vision systems are needed for high performance automated processes. 3D image processing is the important link between the robot's vision and its autonomous behaviour. Hence, application-driven development and industrial embedded research often requires efficient methods for fast and reliable processing. Dealing with these constraints this work firstly introduces a novel method for a fast and robust detection of bore holes for industrial inspection tasks using geometric primitives. The second part presents a rapid object detection approach using a Superquadric model description consistent with human perception for robot handling tasks.

In the following sections, a motivation for this research is given ending up in a presentation of the problem statement in Section 1.1. The contributions of this dissertation are described in Section 1.2 and finally, the organization of this dissertation is provided in Section 1.3.

1.1 Motivation and Problem Statement

Vision systems get more and more important for robotic applications to enhance the robot's flexibility by acquiring its environment and task specific information. Beside passive vision systems such as cameras and stereo systems, active vision systems are increasingly used in industrial robotics. Laser range sensors scan the objects by acquiring a point cloud of an object's surface. Due to the high amount of point data, efficient methods for range image processing are needed to satisfy the industrial requirements concerning the processing time. This brings us to the motivation of this thesis, rapidly detecting features and objects for industrial and robotic applications.

1.1.1 Feature Detection for Automated Inspection

Quality control of mechanical components in general is vital for manufacturers in order to meet requirements of safety-critical products and components, e.g., in automotive, transportation, aerospace or aeronautics, and in order to increase customer satisfaction by assured quality. Bores and internal threads play a critical role in mechanical systems. Bores are integral parts of bearings and engines, hydraulic and pneumatic systems, and pumps, seals, and valves. Porous or rough bore surfaces and defects such as blow-holes, chips, or scratches lead to increased friction, abrasion, degrading closeness and even leakage. Thus, the energy consumption of the devices increase. Defective bores cause instabilities of connections of components. In both cases, the devices' reliability and the span of life is reduced. Therefore, industry demands ever higher quality of components' bore surfaces. For SMEs, which are often suppliers of these components, inspection of bore surfaces is a crucial factor for their competitiveness since the dependability and quality standard of their products can be increased. The need for this quality control of bores is currently met mainly by manual inspection, often with endoscopes. However, manual inspection of bores is exhausting, time-consuming and expensive and thus, often limited to samples.

So, a key requirement to ensure a fully automated 100% inspection of bores in components with almost arbitrary shape, is the reliable localization and detection of the bore in 5 DoF. To achieve the goal of inspecting bores with diameters ranging from 4 to 50mm an accurate compensation of the positioning errors using sensor-based servoing with respect to the bore is needed. Acceleration of bore inspection up to 900% (compared to a manual process) requires rapid and robust 3D image processing methods to keep the robotic inspection in process real-time. Safety-critical applications will not be limited anymore to sample inspection due to economical reasons.

The difficulty is to cope with the sparse data of not more than 120 degrees of partial information of the bore hole caused by the occlusions of the laser range sensor. Further problems to deal with are the difficulty of the nature of the top surface, which can be bumpy due to cast iron material or curved with a diameter larger than the bore hole. Bore holes can also be slanted towards the top surface and bore hole edges can have different types from sharp to rounded edges or chamfers. Furthermore, bore hole localization must be robust to any noise or outliers from the range sensor, e.g., caused by specular surfaces. Some bore holes have additional notches, which must be handled separately, since they result in a second bore diameter.

1.1.2 Object Detection for Robot Handling

Detecting and localizing objects is a fundamental task of robotic systems. The task is of great importance in industrial applications for the automation of part production and the ultimate goal to achieve a lot size of one or pure customization. The task is also of great relevance for assistance systems for helping handicapped or elderly persons with the delivery and handling of daily-life objects. Such systems enhance the user's independence and improve quality of life. The automatic localization of a wide variety of differently shaped objects scanned from one-view in a cluttered setting is necessary for robotic applications. The determination of the object's pose enables a grasp planning of a desired robot arm.

The classical approach is to use intensity or color cameras to exploit the appearance of objects for the detection task. Because shape is not directly encoded, this problem is in general difficult or ill-posed [6]. However, recent progress in invariant feature extraction is the basis to obtain first good results in realistic settings [38, 66, 75]. Although these approaches are rather fast, they do not work satisfactorily in cluttered scenes and “inherit the major problem of intensity-based systems, that is, dependency on lighting conditions” [61].

To overcome these problems with intensity images the 3D shape of the objects has to be directly recovered from range images. The model-based approach keeps the detection method simple and efficient where a time consuming range image segmentation is not necessary. The question is now how to model the object for detection where human perception gives us valuable hints to solve tackle that problem.

The human representation of visual shape is part-based [115]. Human vision parses shapes into component parts, and it organizes them using these parts and their spatial relationships. From computational perspective, parts are useful for many reasons. First many objects are articulated: A part-based description allows one to decouple the shapes of the parts from the spatial relationships of the parts – hence providing a natural way to present and recognize articulated objects. Second, one never sees all of an object in one view: the rear side of an object is not visible due to self-occlusions, and its front may be occluded by other objects. Representing shapes by parts allows the recognition process to proceed with those parts that are visible.

One theoretical approach defining parts, is to postulate that human vision uses general computational rules, based on the intrinsic geometry of shapes, to parse visual objects [115]. This approach separates the issue of finding parts from the issue of describing them. A visual system decomposes a shape into a hierarchy of parts. Parts are not chosen arbitrarily. The mental category “part” of shapes is based upon a regularity of nature discovered by differential topologists-transversality. Part decomposition segments an object into its constituent parts consistent with human perception. A regularity of nature called transversality regularity proposed by Hoffman et al. [52] defines a simple partitioning rule. When two arbitrarily shaped surfaces are made to interpenetrate, they always meet at a contour of concave discontinuity of their tangent planes.

Multiple parametric models have been introduced that are suited to describe parsed objects according to the rule of transversality, for 3D object recovery – numerating most relevant: geometric primitives, generalized cylinders, Superquadrics, Geons, implicit polynomials and spherical harmonic surfaces (see Section 2.1). Superquadrics are perhaps the most popular due to several reasons. The compact shape can be described with a small set of parameters ending up in a large variety of different basic shapes. The recovery of Superquadrics has been well investigated and even global deformations can be easily adopted [116]. They can be used as volumetric part-based models desirable for robotic grasping operations. These advantages can not be found in other geometric entities which predestines the Superquadric model for part-based object description.

As it is the natural way of human perception, in robotics often one-view scans of the scene of interest are taken. The acquired range data is sparse due to the single-view scan, exhibiting the typical laser and camera shadows, and the objects are only partly visible

in cluttered scenes typically encountered in realistic scenarios. Detecting an object in this range data is difficult and the task given to the robot is to find and locate an object – within a few seconds – specified by the user. Handling these problems with the object’s occlusions and intersections is a difficult task to be solved.

The second problem: Due to the single-view range image the information of the objects’ rear side is not visible. What can be exploited is the symmetry of the real-world objects. Furthermore, keeping the set of model parameters low it is not possible to model complex shaped object with a single Superquadric or a set of Superquadrics with global deformations. So the model should describe the approximated overall shape of the object where the details in shape are moreover not present in the sparse range data.

1.2 Contributions

Fitting geometric models is crucial in range image processing and an efficient way for scene segmentation and understanding, object decomposition, detection and recovery and reverse engineering. Model fitting is mostly a non-linear problem often extensive to solve depending on the complexity of the model respectively the number of fit parameter. Hence, this thesis spots the problem of efficient model fitting under the constraint of achieving fast and robust results. The contribution of this work shows the efficient use of plane fitting for a novel geometric laser stripe profiler calibration, a rapid cylinder fitting method and a fast object detection method using a Superquadric-based model description.

1.2.1 Laser Stripe Profiler Calibration

Contributions in the field of laser stripe profiling have been made introducing a novel geometric calibration method using robust plane fitting. Since this work has been done as extension to my basic research work it is only briefly described in Section 3.2. The novelty is based on the geometrical approach using the non-calibrated range data from several views without the use of markers. We introduced a new design for a calibration object where six planes are visible nearly independent from the view point. Robust plane fitting is applied to reconstruct the calibration object from the range data. This allows a simultaneous calibration of several laser stripe profilers iteratively solving a linear system. Averaging the calibration results of each laser stripe profiler increases the overall calibration accuracy for the whole system. This method is designed for a fully automatic use without user interaction and is fully described in [118].

1.2.2 Rapid Cylinder Fitting

The most popular method for fitting cylinders to range data is the least-squares technique. But this is a non-linear problem which has to be tackled iteratively. Methods to solve such problems are introduced in Section 2.2.2. Least-squares fitting works well with a sufficient good estimation of the starting pose, noiseless range data and enough raw data points around the cylinder surface (compare Section 4.2.4). But these requirements are often not provided in industrial applications. Furthermore, if we have dense range data the

computational cost of least-squares fitting is approximately linear with the number of raw data points (see Figure 4.35). The contribution of this thesis is a fast and robust method optimized for fitting cylinders in sparse and noisy range data. Exploiting the surface normal vectors which are needed for segmenting the cylinder surface the method introduces a novel sequential cylinder fit based on random samples to handle the outliers. First part of the approach is to calculate the radius by averaging the surface perpendicular circumscribed circle of cylinder sample points. As second step the normal vector elongated with the radius length crosses the cylinder-axis which can be calculated by line fitting of the vector's end points. Chapter 4 describes the approach in detail and demonstrates the robustness and computational effectiveness compared with the least-squares fitting of cylinder and Superquadric models.

1.2.3 Superquadric-Based Object Detection

Superquadric-based object recovery and detection has been well investigated during the last decade – see Section 2.3 – but there is a lack in computational effectiveness, which is tackled in this thesis. The contribution is a reliable detection and pose determination of known objects in nearly real-time. It is shown that a hierarchical search in the sense of a coarse to a fine processing (sub-scaling the range data) increases computational efficiency and saves valuable processing time [25]. The proposed approach, achieving fast results, is embedded in a hierarchical structure starting with a RANSAC-based object search to find pose hypotheses followed by a pose refinement. In the second step these hypotheses are verified executing a ranked voting procedure [95] over the sorted quality-of-fit criteria (the measurement of fit, the number of interior points and the number of points on the Superquadrics surface) to robustly select the final object pose. Moreover, it is shown that searching for a tapered Superquadric with fixed size and shape parameter makes the Levenberg-Marquardt [85] minimization process less complex, less time consuming and more stable. Using a model description with several Superquadrics extends the ranked voting process with two additional parameters describing the relation between the Superquadrics, the main axis steradian and the center distance. Chapter 5 describes this method in detail and demonstrates the robust detection results by optimizing the computational effort.

1.3 Thesis Overview

The remainder of this dissertation is organized as follows. Starting with the state of the art in Chapter 2 giving an overview of parametric models for part and object description in range images. Further, the state of the art gives the theoretical background of geometric fitting describing the error metric and the mathematical methods for least-squares minimization. An overall summary of part and object detection using geometric primitives and Superquadrics concludes the state of the art.

Chapter 3 gives an introduction to the process of automated industrial production. This is necessary to understand the motivation of this dissertation, by introducing the

application driven research work of three industrial projects, where range image scanning and the geometric interpretations are needed.

After the presentation of the project work the main contribution of this thesis is given in Chapter 4 and Chapter 5. First a novel method is introduced for a rapid bore hole detection in range image data. This low-level 3D image processing is evaluated in detail in terms of a performance analysis and is compared with the standard least-squares cylinder and Superquadric fitting. My second work presents a high-level method for a fast model-based object detection using Superquadrics. The experiments evaluate the method in terms of robustness and computational performance.

The outcome of the methods presented in Chapter 4 and 5 is given in Chapter 6 outlining the developed prototype and future research topics, e.g., a concept for a fully automated object learning, detecting and tracking system. Finally this work is concluded in Chapter 7 with a discussion on reliable and rapid part feature and object detection methods.

Chapter 2

State of the Art

This chapter presents the state of the art of the research literature on 3D part and object detection. The introduced model-based detection approach requires a three-dimensional object description which is reviewed in the first part of this chapter. Section 2.1 presents the parametric models that are used in literature for object recovery and description dealing with range image data. This includes a review of the application areas where these volumetric entities are used. Because of the use of range images – acquired by laser range scanner – the fitting of the geometric models to the point cloud is another important issue within a fast detection process. Section 2.2 gives the state of the art in least squares fitting by reviewing the fitting metrics and describing the common mathematical methods. Concluding this chapter, Section 2.3 reviews the literature on model-based object detection focused on cylinders and Superquadrics, that are related to my work.

2.1 Parametric Model Description

Reviewing the research literature a quite high amount of parametric models can be found that are used in 3D computer vision, especially dealing with point clouds. The level of detail in these models ranges from simple descriptions to highly complex representations. The number of shape parameters varies a lot depending on the complexity of the modeling. Figure 2.1 gives a qualitative survey of the hierarchical order of the models presented in this section. Models with low complexity, corresponding to a few number of parameter for shape description are grouped to the geometric primitives entity. Generalized cylinders, Superquadrics and Geons are models whose number of parameters can vary, mostly within the generalized cylinder model, but the complexity in shape is limited as well as the number of parameters. In contrast to that the class of high parametric models can represent any arbitrary shape. The level of detail depends on the number of parameters which is in fact not limited. Recovering objects with up to 500 parameters is state of the art using spherical harmonic surfaces.

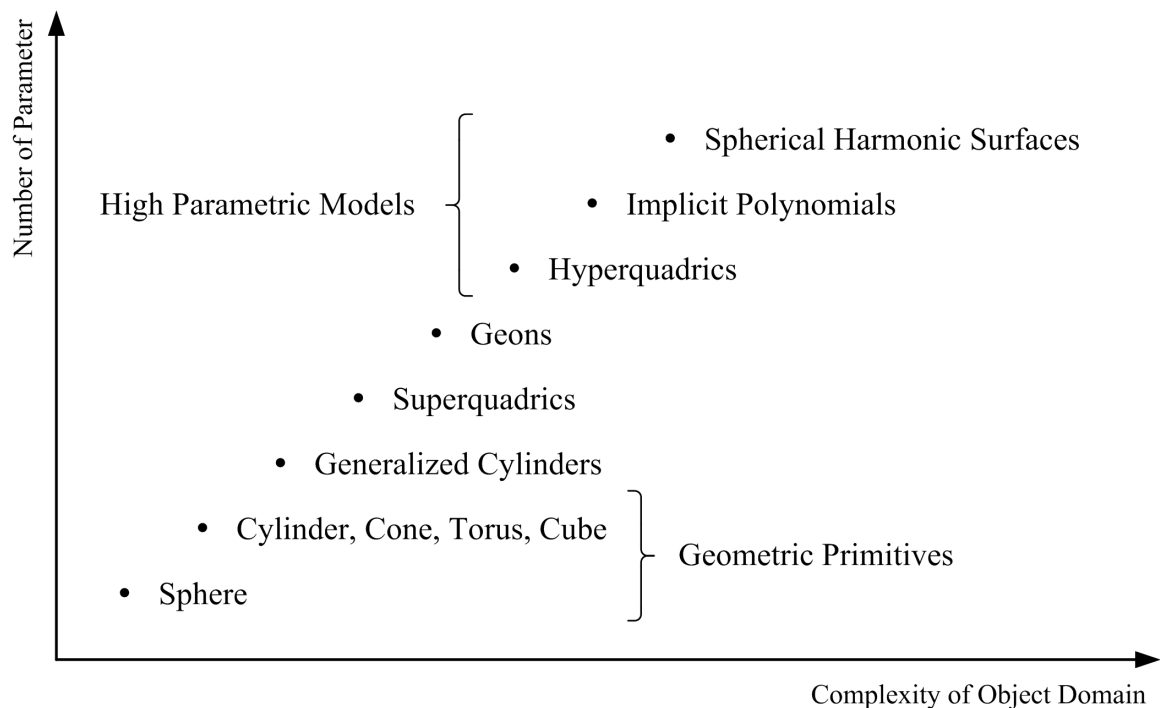


Figure 2.1: Qualitative representation of the hierarchical coherence of the parametric models ranked by the number of parameters faced to their shape complexity.

2.1.1 Geometric Primitives

Geometric primitives are the basic models used in computer vision for object recovery, recognition and detection. Figure 2.2 illustrates the models by its increasing number of parameters. Starting with the sphere, the cylinder, the cone, the torus and finally the cube. Table 2.1 summarizes the number of minimum parameters for a fully 3D description, with r, R, h, s and Ψ as shape parameters and $\mathbf{x} = (x, y, z)^T$ and $\mathbf{a} = (a_x, a_y, a_z)^T$ as position point respectively as orientation vector. Instead of using the orientation vector \mathbf{a} , which contains redundant orientation information for rotational symmetric objects, two angles α and β can describe the orientation of the objects z -axis, by defining a rotation around the x - and y -axis of the world coordinate system.

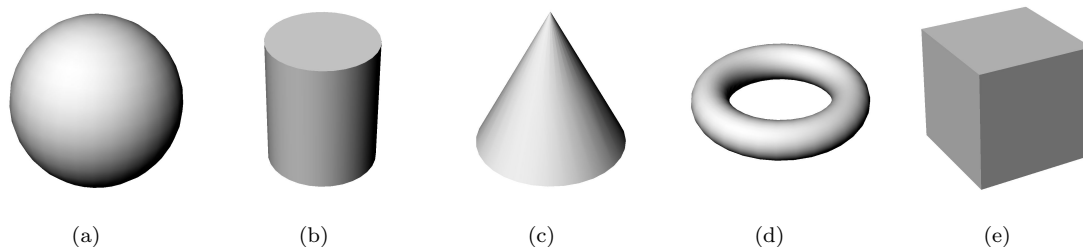


Figure 2.2: Set of geometric primitives: from (a) to (e) sphere, cylinder, cone, torus, cube.

The cube is not a rotational symmetric object, so the orientation vector for a unique

	Shape	Pose	# Parameters
Sphere	r	\mathbf{x}_m	4
Cylinder	r, h	$\mathbf{x}_a, \alpha, \beta$	7
Cone	h, Ψ	$\mathbf{x}_a, \alpha, \beta$	7
Torus	r, R	$\mathbf{x}_m, \alpha, \beta$	7
Cube	s	$\mathbf{x}_m, \alpha, \beta, \gamma$	7

Table 2.1: Summary of the geometric primitives' shape and pose parameter.

pose determination must be defined with three angles, α , β and γ . The orientation vector goes through the center point \mathbf{x}_m and is perpendicular to the top surface. With the side length s the cube is fully described in 3D space.

In the past decade much work has been made describing range data with geometric primitives [37, 77, 80, 126] using all primitives except the cube. This can be easily explained, because rotational symmetric primitives can be described with an implicit closed form, while the cube model has to be composed of six planes [64, 125]. There is no simple closed form description available. But note that the cube can be modeled with a more complex model description: the Superquadric (2.15), the Hyperquadric (2.28) or the spherical harmonic surface (2.35).

The following sections introduce the primitives with their model description including the parameters and the distance function which is needed to obtain the objective function for the least-squares fitting (compare (2.39)). For the sake of completeness two functions u_i and v_i should be introduced defining the orthogonal distance from a point \mathbf{x}_i to a line respectively to a plane. The vector \mathbf{a} represents either the line direction or the plane's normal vector and \mathbf{x} is a point defining the position of the line respectively the plane. Note, that these two distance functions can be used for least-squares fitting of a solid line or a plane and are furthermore used to describe the distance functions of the primitives, even if not using the minimum number of parameters. For a simple mathematical description of the geometric primitives, the rotation axis of the cylinder, the cone and the torus are defined with the vector \mathbf{a} . In the case of the cube each face normal vector is described with the vector \mathbf{a} .

$$u_i = f(\mathbf{x}_i, \mathbf{x}, \mathbf{a}) = |\mathbf{a} \times (\mathbf{x}_i - \mathbf{x})| \quad (2.1)$$

$$v_i = g(\mathbf{x}_i, \mathbf{x}, \mathbf{a}) = \mathbf{a} \cdot (\mathbf{x}_i - \mathbf{x}) = a_x(x_i - x) + a_y(y_i - y) + a_z(z_i - z) \quad (2.2)$$

Sphere

The sphere is the simplest geometric model with a large power of object description. Many of our daily objects consist of this compact shape description. Equation 2.3 gives the basic shape of a sphere.

$$x^2 + y^2 + z^2 = r^2 \quad (2.3)$$

The surface of a sphere can be described in 3D space with the center point \mathbf{x}_m and the radius r where the signed distance function is simply the radial length from an arbitrary

point \mathbf{x}_i to the surface.

$$d_{Sphere}(\mathbf{x}_i, \mathbf{x}_m) = |\mathbf{x}_i - \mathbf{x}_m| - r \quad (2.4)$$

Cylinder

The second important geometric model to describe common objects in daily life is the cylinder. The cylinder consists of two shape parameters, the radius r and the height h . Equation 2.5 gives the basic shape of the cylinder, but note that this equation describes slices (circle lines) of the cylinder surface perpendicular to the axis. The z coordinate is constant for each surface circle within the range of $0 \leq z \leq h$. The cap surfaces are not represented in Equation (2.5).

$$x^2 + y^2 = r^2 \quad (2.5)$$

The position in 3D space is determined with an arbitrary point \mathbf{x}_a on the cylinder axis and the vector \mathbf{a} of the axis itself. Whenever fitting a cylinder to range data the shell surface is the important one. So it is not necessary to define a distance function of the cap surfaces.

$$d_{Cylinder}(\mathbf{x}_i, \mathbf{x}_a, \mathbf{a}) = u_i - r \quad (2.6)$$

Cone

The cone is the next geometric primitive to be introduced. It can be described with the height h and the cone's apex semi-angle Ψ . The circular cap surface is as well not included in the shape Equation (2.7) as already shown with the cylinder Equation (2.5). Note that the shape description is only valid for z coordinates within the interval of $0 \leq z \leq h$.

$$\sqrt{x^2 + y^2} = z \tan \Psi, \quad \{z \in \mathbb{R} \mid 0 \leq z \leq h\} \quad (2.7)$$

The orientation and position is again defined with a vector \mathbf{a} for the axis and an arbitrary point \mathbf{x}_a on it. The distance function of a point \mathbf{x}_i is defined as the perpendicular distance to the shell surface. As for the same reasons as with the cylinder the cap surface is not included in the distance function.

$$d_{Cone}(\mathbf{x}_i, \mathbf{x}_a, \mathbf{a}) = u_i \cos \Psi + (v_i - h) \sin \Psi \quad (2.8)$$

Torus

The torus is an object which often appears in the literature for model fitting but the applications to real world objects is limited. Nevertheless the torus can be fully described with the radius r of the "donuts" cross-section and the radius R of the circular axis perpendicular intersecting the cross-section.

$$\left(R - \sqrt{x^2 + y^2}\right)^2 + z^2 = r^2 \quad (2.9)$$

To describe the pose a center point \mathbf{x}_m on the axis \mathbf{a} through the "donut" hole is sufficient. The distance function of an arbitrary point \mathbf{x}_i in 3D space always describes the orthogonal distance onto the torus surface.

$$d_{Torus}(\mathbf{x}_i, \mathbf{x}_m, \mathbf{a}) = \sqrt{v_i^2 + (u_i - R)^2} - r \quad (2.10)$$

Cube

As last geometric primitive the cube is presented. As already mentioned a simple surface description of a cube can only be achieved with a set of three orthogonal parallel plane pairs with the shape parameter of its side length s . To describe the whole surface of the cube in 3D space a center point \mathbf{x}_m and the orientation of the z -axis is required. Then, all plane parameters can be uniquely derived.

A least-squares fit of a cube in range data is not as simple as with the other primitives. In the case of a cube each segmented plane in the range image has to be fitted using v_i (2.2) as distance function. After that the spatial relations of the planes can be linked to assign a cube model to the range data.

Note that the cube can be easily described with a more complex model where a closed-form distance function is available, e.g. the Superquadric (see Section 2.1.3) and its inside-outside function (2.15).

2.1.2 Generalized Cylinders

Generalized cylinders are the first dedicated part-level models in computer vision [14]. They have influenced much of the model-based vision research in the past two decades. As shown in Figure 2.3(a), a generalized cylinder is formed from a volume by sweeping a two-dimensional set along an arbitrary space curve. The set may vary along the curve (axis). Therefore, definitions of the axis and the sweeping set are required to define a generalized cylinder. The axis can be defined as a function of arc length s in a fixed coordinate system (x, y, z)

$$a(s) = (x(s), y(s), z(s)). \quad (2.11)$$

The sweeping set is more conveniently expressed in a local coordinate system which is defined at the origin of each point of the axis $a(s)$. The sweeping set can be defined by a cross-section boundary, which is parameterized by another parameter r

$$\text{sweeping set} = (x(r, s), y(r, s)). \quad (2.12)$$

This definition is very powerful and describes a large variety of shapes (see Figure 2.3(b)). To limit the complexity and simplify the recovery of generalized cylinders from images, constraints are often incorporated. Straight axes and constant sweeping sets are two main constraints added to generalized cylinders to represent regular shapes.

Generalized cylinders are particularly attractive for representing elongated shapes where an axis is easy to define. In such cases, the axis of generalized cylinders often provides an intuitive method to conceptualize the design of an object, and a method of reliably recovering useful statistics about the shape of the object.

An often cited early vision system which applied generalized cylinders is the ACRONYM system [23]. This is a model-based system for interpretation of airport scenes. Recovery of generalized cylinders from intensity images has been studied by many researchers. Especially notable for their work in this area are Rao et al. [105], Mohan et al. [84], and Zerrouh et al. [141]. The recovery of generalized cylinders from intensity images seems

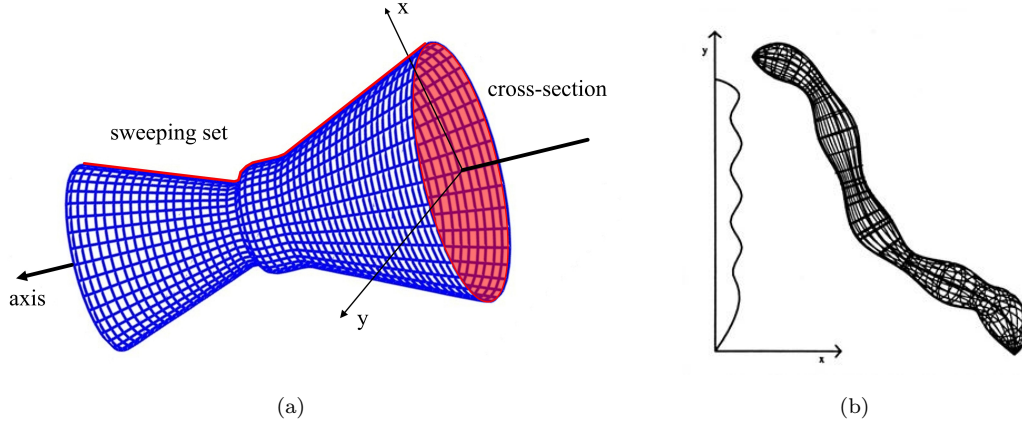


Figure 2.3: Illustration of a generalized cylinder. (a) A generalized cylinder is defined as a volume formed by sweeping a cross-section along an axis. b) A generalized cylinder with an arbitrary shaped axis and the displayed sweeping function [29].

to be overly complex since it must rely on complicated rules to group low-level image features, such as edges, corners and surface normals, into features of larger granularity (i.e., symmetrical contours or cross-sections), and to finally assemble them into generalized cylinders. These problems are due partly to the complicated parameterization of generalized cylinders, and to the lack of a fitting function that would provide a direct evaluation criteria on how well the model generalized cylinder fits the image data [57]. Consequently, only the recovery of a restricted subset of generalized cylinders such as straight homogeneous generalized cylinders [101] has been completed in the literature to date. Tackling the problem of recovering generalized cylinders from intensity some work has been done extracting generalized cylinders with symmetric cross-sections directly from range data as shown in [30] and [121]. This recovery seems to be more robust than using intensity images, because the range image provides geometric shape information.

2.1.3 Superquadrics

Superquadrics are a family of parametric shapes which have been first introduced in computer graphics by Barr [3] in 1981. Superquadrics can be classified into Superellipsoid, Supertoroid and Superhyperboloid with one and two parts. In this work we focus on the Superellipsoid which is useful for a volumetric part-based object description, because they are compact in shape and have a closed surface. For a detailed mathematical description of the Superquadric and its properties you are referenced to [57].

The explicit form of the Superquadrics surface is defined by the spherical product of two parameterized quadric curves (Superellipses) $s_1(\eta)$ and $s_2(\omega)$ which is given by

$$\mathbf{x}(\eta, \omega) = s_1(\eta) \otimes s_2(\omega) = \begin{bmatrix} \cos^{\varepsilon_1}(\eta) \\ a_3 \cdot \sin^{\varepsilon_1}(\eta) \end{bmatrix} \otimes \begin{bmatrix} a_1 \cdot \cos^{\varepsilon_2}(\omega) \\ a_2 \cdot \sin^{\varepsilon_2}(\omega) \end{bmatrix} \quad (2.13)$$

$$\mathbf{x}(\eta, \omega) = \begin{bmatrix} a_1 \cdot \cos^{\varepsilon_1}(\eta) \cdot \cos^{\varepsilon_2}(\omega) \\ a_2 \cdot \cos^{\varepsilon_1}(\eta) \cdot \sin^{\varepsilon_2}(\omega) \\ a_3 \cdot \sin^{\varepsilon_1}(\eta) \end{bmatrix} \quad \text{with} \quad \begin{array}{l} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \omega < \pi \end{array}, \quad (2.14)$$

where ε_1 and ε_2 are the shape parameters and a_1 , a_2 and a_3 are the scale parameters along the x -, y - and z -axis of the Superquadric. Varying the shape and scale parameter a lot of different shapes with a closed surface can be obtained which are summarized in Table 2.2. Figure 2.4 illustrates these basic Superquadric shapes.

ε_1	ε_2	a_1, a_2, a_3	description
0	0		box
0	0	$a_1 = a_2 = a_3$	cube
< 1	< 1		cuboid
< 1	1		pillow shape
1	< 1		cylindrical
1	1		ellipsoid
1	1	$a_1 = a_2 = a_3$	sphere
2	2	$a_1 = a_2 = a_3$	octahedron
ε_1 or $\varepsilon_2 > 2$			pinched
ε_1 or $\varepsilon_2 = 2$			bevelled

Table 2.2: Superquadric description of basic shapes with dependency on ε_1 , ε_2 , a_1 , a_2 and a_3 .

Substitution of η and ω in Equation (2.14) leads to the implicit Superquadric equation which is referenced to an inside-outside function F , necessary for the objective function in the least-squares fitting procedure. Note, that the expression F denotes the algebraic distance function of a Superquadric (see Section 2.2.1),

$$F(\mathbf{x}) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\varepsilon_1}} \quad (2.15)$$

with the property of determining the position of a point $\mathbf{x} = (x, y, z)^\top$ related to the Superquadrics surface.

$$F(\mathbf{x}) < 1, \quad \mathbf{x} \text{ inside the Superquadric} \quad (2.16)$$

$$F(\mathbf{x}) = 1, \quad \mathbf{x} \text{ on the Superquadrics surface} \quad (2.17)$$

$$F(\mathbf{x}) > 1, \quad \mathbf{x} \text{ outside the Superquadric} \quad (2.18)$$

To cover most of the object part shapes global deformations of the Superquadrics are required which were also introduced by Barr [4]. In this work global tapering is used and should be briefly introduced. Given a point $\mathbf{p}_{SQ} = (x, y, z)^\top$ on the Superquadrics surface,

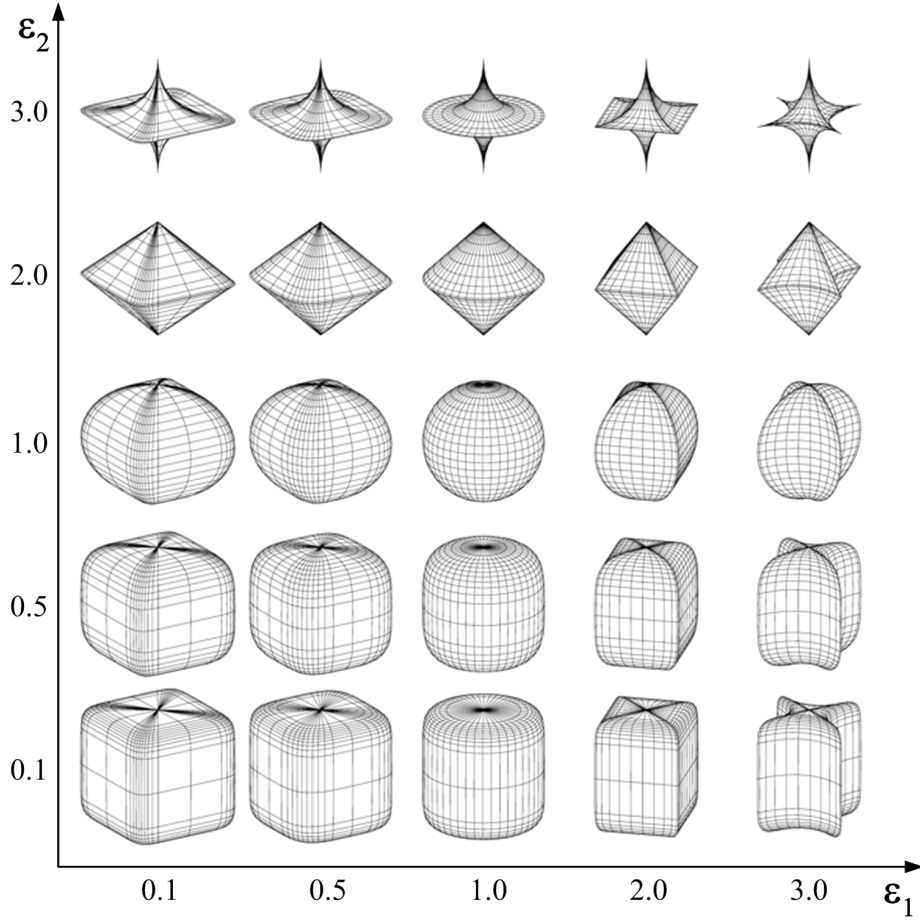


Figure 2.4: Basic Superquadric shapes varying ε_1 and ε_2 with $a_1 = a_2 = a_3$.

$\mathbf{P}_{SQ} = (X, Y, Z)^\top$ is the corresponding tapered point along the Superquadrics z -axis and is defined by

$$X = f_x(z) \cdot x \quad (2.19)$$

$$Y = f_y(z) \cdot y \quad (2.20)$$

$$Z = z \quad (2.21)$$

with the tapering functions

$$f_x(z) = \frac{k_x}{a_3} z + 1 \quad \text{and} \quad (2.22)$$

$$f_y(z) = \frac{k_y}{a_3} z + 1. \quad (2.23)$$

The tapering coefficients k_x and k_y are constrained within the range of $-1 < k_x, k_y < 1$. If $k_x = 0$ and $k_y = 0$ no deformation is applied. Figure 2.5 illustrates the taper behaviour of a cube-shaped Superquadric with $\varepsilon_1 = \varepsilon_2 = 0.3$ and $a_1 = a_2 = a_3 = 1$. A tapered Superquadric in general position (p_x, p_y, p_z) and orientation (ϕ, θ, ψ) has the ability to represent shapes of a sphere, a cuboid, a cylinder, a pyramid, a cone, a frustum of cone

and pyramid, a wedge, an anvil and all shapes in between with the parameter set Λ in general position and orientation.

$$\Lambda = \{a_1, a_2, a_3, \varepsilon_1, \varepsilon_2, \phi, \theta, \psi, p_x, p_y, p_z, k_x, k_y\} \quad (2.24)$$

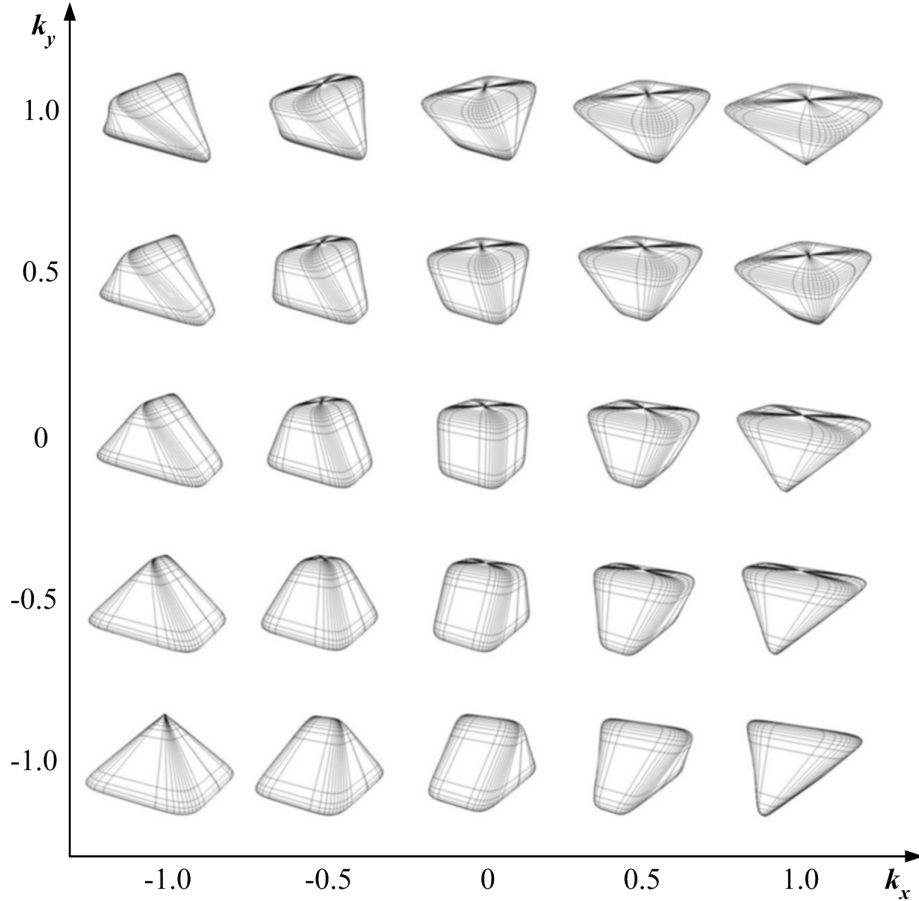


Figure 2.5: Visualization of taper deformation using a cube-shaped Superquadric ($\varepsilon_1 = \varepsilon_2 = 0.3$ and $a_1 = a_2 = a_3 = 1$).

Pioneer work on recovering Superquadrics in range image data was introduced by Solina et al. [116] in 1990. A lot of contributions have been published building on this work. A detailed state of the art on Superquadric recovery will be given in Section 2.3, but some selected works of the last years should be mentioned here. Zhang et al. demonstrated the part recovery and object representation of automotive parts [143] while Katsoulas showed the detection of parabolic deformed Superquadrics in a point cloud [61]. Concluding the selected works, Krivic et al. presented an object detection approach using part based Superquadric representations [67].

2.1.4 Geons

In 1987, Biederman proposed the Recognition-by-Components theory which offers an explanation of human visual object recognition [8]. The fundamental assumption of this theory is that there exists a small number (i.e. 36) of fundamental part primitives, whose boolean combinations can represent more complicated objects for the purpose of “primal access”. The Geons consist of 36 basic types which are classified on the basis of four qualitative shape attributes of generalized cylinders: Symmetry, Size, Shape Edge, and Axis (see Table 2.3). *Symmetry* defines the characteristic of the Geons’ shape, *Size* describes the characteristics of the cross-section along the axis, *Shape Edge* characterizes a round or angular cross-section, while *Axis* describes the shape of the axis. Figure 2.6 illustrates all 36 Geons classified according Table 2.3.

	++	+	-	--
E (<i>Edge Shape</i>)	-	Curved	Straight	-
Sy (<i>Symmetry</i>)	Reflection and Rotation	Reflection	Asymmetric	-
Si (<i>Size</i>)	Constant	-	Increasing	Increasing and Decreasing
A (<i>Axis</i>)	-	Straight	Curved	-

Table 2.3: Geon classification attributes.

Geons themselves are no parametric models but represent a set of shapes for a volumetric object description. Dealing with Geons in computer vision requires a parametric description. Raja et al. [104] related Superquadrics and Geons together and explored the recovery of Geons from single-view range images by classifying the actual parameters of Superquadrics. A more precise Geon modeling is discussed by Zhou et al. [145] proposing extended Superquadrics which can be deformed in any direction because it extends the exponents of Superquadrics from constants to functions of the latitude and longitude angles in the spherical coordinate system. The extended Superquadric is defined by

$$\mathbf{x}(\eta, \omega) = \begin{bmatrix} a_1 \cdot \cos^{f_1(\phi)}(\eta) \cdot \cos^{f_2(\theta)}(\omega) \\ a_2 \cdot \cos^{f_1(\phi)}(\eta) \cdot \sin^{f_2(\theta)}(\omega) \\ a_3 \cdot \sin^{f_1(\phi)}(\eta) \end{bmatrix}, \quad \begin{matrix} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \omega < \pi \end{matrix} \quad (2.25)$$

with $f_1(\phi)$ and $f_2(\theta)$

$$f_1(\phi) = \sum_{i=0}^n P_{1i} B_i^n \left(\frac{\phi + \frac{\pi}{2}}{\pi} \right), \quad f_2(\theta) = \sum_{i=0}^n P_{2i} B_i^n \left(\frac{\theta + \pi}{2\pi} \right) \quad (2.26)$$

with P_i as control points for the Bernstein polynomials B_i^n of degree n .

$$B_i^n(t) = \frac{n!}{(n-i)!i!} t^i (1-t)^{n-i}, \quad n = 0, 1, \dots, n \quad (2.27)$$

Thus, extended Superquadrics can model more complex shapes than solid Superquadrics (2.14) which predestinates the extended Superquadric for Geon description. Representative for all Geon-based object representation two works are presented for Geon recognition using edge-junction graphs of range images [87] or Support Vector Machines [138].

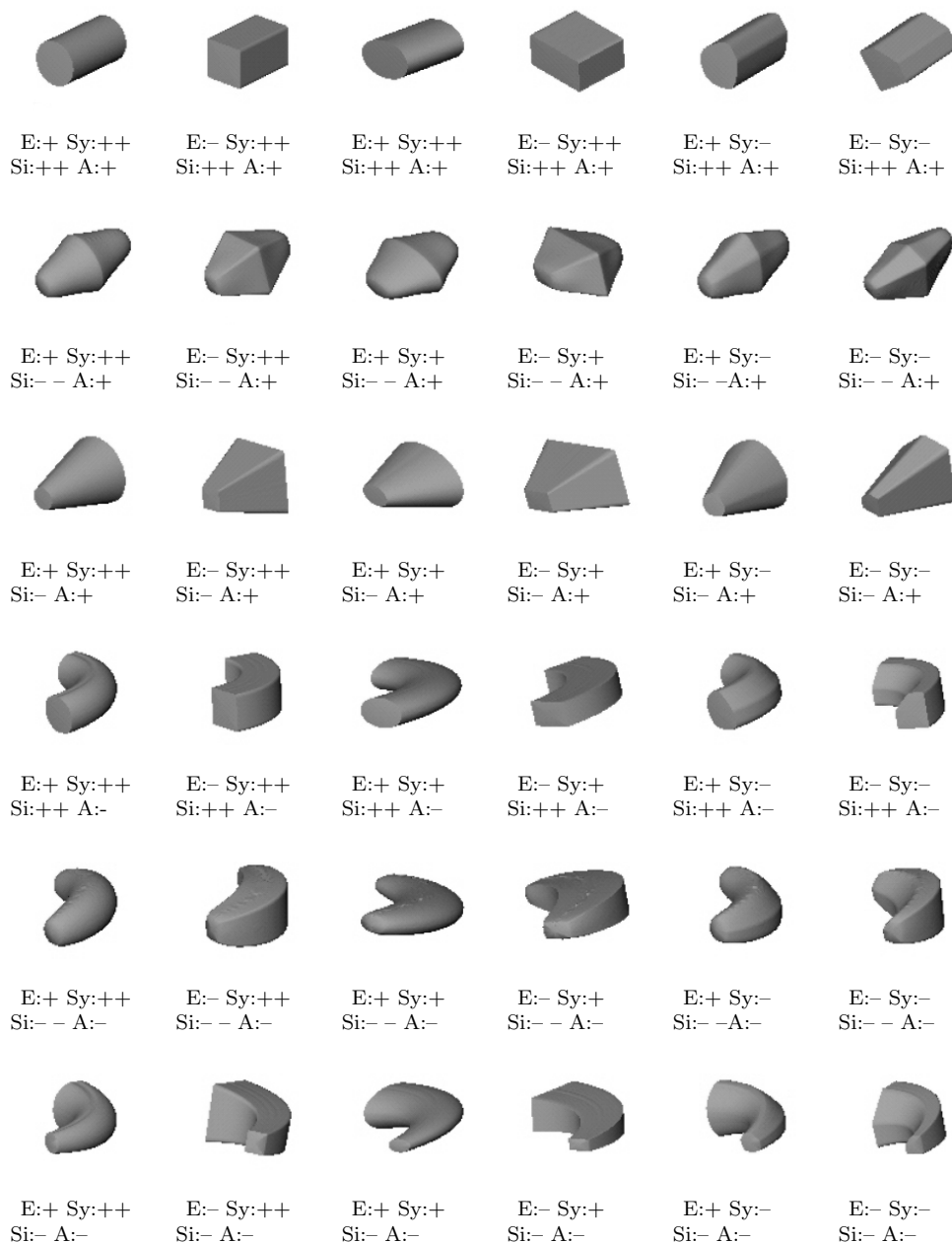


Figure 2.6: List of the 36 basic Geons introduced by [8].

2.1.5 High Parametric Models

In contrast to the models in the previous sections the number of parameters is not limited using high parametric models. Thus implies that the level of shape complexity can be refined with the increasing number of parameters in the modeling process. These models are suitable for free-form description of objects with a closed surface. In this section Hyperquadrics, implicit polynomials and spherical harmonic surfaces are introduced.

Hyperquadrics

Implicit surface models used in computer vision are mainly based on Superquadrics. These closed surface descriptions are included in a more general class of shape modeling, the Hyperquadric model, introduced by A.J. Hanson [47]. The Hyperquadrics are obtained by considering a sum of an arbitrary number of linear terms raised to powers. This generates shapes whose bounding polytopes have an arbitrary number of faces. In contrast to Superquadrics, Hyperquadrics are not limited to symmetric shape description and their model is defined by the set of points satisfying

$$H(x, y, z) = \sum_{i=0}^N |H_i(x, y, z)|^{\gamma_i} = 1, \quad (2.28)$$

where N is an arbitrary number of linear forms H_i ($N > 3$ for a closed surface)

$$H_i(x, y, z) = A_i x + B_i y + C_i z + D_i, \quad (2.29)$$

where A_i, B_i, C_i, D_i and γ_i are constants ($\gamma_i > 0$). Note, as mentioned before, that a Superquadric can also be written as

$$H(x, y, z) = (A_1 x)^{\gamma_1} + (B_2 y)^{\gamma_2} + (C_3 z)^{\gamma_3}. \quad (2.30)$$

Equation 2.30 is a special case of Hyperquadrics with $N = 3$, that represents a superellipsoid where $\varepsilon_1 = \varepsilon_2 = 1$ (compare equation 2.15), $\gamma_1 = \gamma_2 = \gamma_3 = 2$, $A_1 = \frac{1}{a_1}$, $B_2 = \frac{1}{a_2}$, $C_3 = \frac{1}{a_3}$, and all other parameters are set to zero.

Unlike the Superquadrics, Hyperquadrics do not have a general explicit parameterization because the number of terms N exceeds in general the number of spherical variables. In the Hyperquadric model the shape is defined globally by the different linear forms. These forms define the convex bounding polytope which gives a geometric description of the shape. As we have seen, this model allows the description of a wide variety of shapes since we can use an unlimited number of linear forms. Figure 2.7 illustrates the shape of Hyperquadrics with 5 respectively 6 linear forms (25 respectively 30 shape parameter).

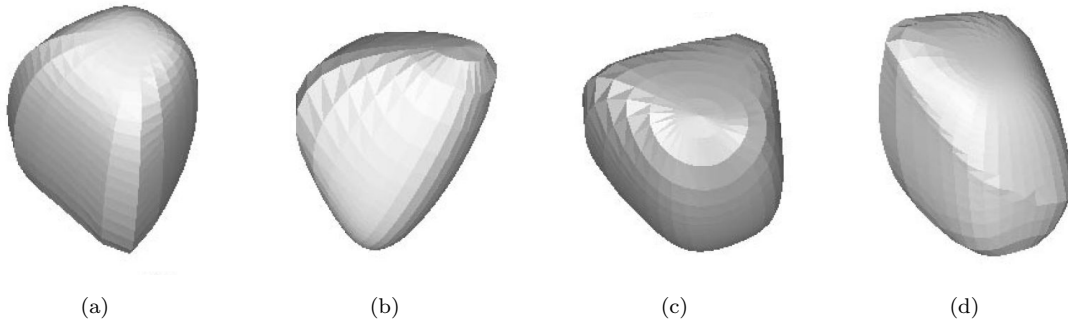


Figure 2.7: Hyperquadrics with (a) 5 faces. (b) 5 faces and one $\gamma_i < 1$. (c) 6 faces. (d) 6 faces and one $\gamma_i < 1$ from [28].

In the literature, the modeling of object parts or the object itself using Hyperquadrics is done with a fitting process to range image data. As Hyperquadrics can describe non-symmetric shapes all modeling approaches need full range data of the objects. First work fitting simple Hyperquadrics with $N = 3$ or $N = 4$ linear forms to range data was presented by [28] and [46]. Recovering more complex objects such as a duck or a hand was shown by [68] using Hyperquadrics with up to $N = 6$ linear forms. Tackling the problem to recover concave parts and objects Ohuchi et al. introduces extended Hyperquadrics [91] by replacing the linear form (2.29) with a parabolic form

$$H_i^e(x, y, z) = A_i(m_{11}x + m_{12}y + m_{13}z)^2 + B_i(m_{21}x + m_{22}y + m_{23}z)^2 + C_i(m_{31}x + m_{32}y + m_{33}z) + D_i. \quad (2.31)$$

Accurate recovering of a human femur is demonstrated using $N = 12$ parabolic forms with a very time consuming process, taking more than three hours for recovering.

Implicit Polynomials

In the past few years, implicit representations have been used more frequently, allowing a better treatment of several problems. One example is the point classification problem, which is easily solved with implicit representation. It consists of a simple evaluation of the implicit functions, although the determination of the distance error between the measured and the model surface is considerably more involved. Furthermore, implicit representations imply surfaces of desired smoothness with the lowest possible degree suitable for appropriate object model description. Finally, when we restrict ourselves to polynomial functions, the implicit representation is more general than the parametric representation. Indeed, it is well known that parametric equations can be converted to implicit ones through the process of implicitization [113], but not always the reverse. The equation to describe an implicit polynomial of an arbitrary degree n is defined by

$$F_n(x, y, z) = \sum_{0 \leq i+j+k \leq n} a_{ijk} x^i y^j z^k = 0. \quad (2.32)$$

In 1994, Keren et al. presented the first work on robustly fitting implicit polynomials to range data [62]. He showed that an implicit polynomial with the 4th degree can easily approximate Superquadric fits. But the drawback is that 34 parameters for a fourth order polynomial are computational expensive to fit compared to the solid Superquadric model (compare Section 2.1.3). Keren improved his work and demonstrated fit results on geometric objects for modeling tasks in reverse engineering applications [63]. Compare Figure 2.8 for an example of a fitted model of a screwdriver shaft.

The implicit polynomials have more power in arbitrary and complex shape description, this high parametric model can also be used for free-form surface reconstruction. But note that there has been made less work found in the literature on higher level implicit polynomial models for complex entire free-form shapes because of the lack of tractable computational procedures for obtaining and analyzing such models. One of these contributions is presented by Blane et al. [15] introducing the 3L-algorithm for a faster fitting of implicit polynomials. Figure 2.9 illustrates a composite fit of an Egypt bust using polynomials of the 4th, 6th and 8th (cap, neck and face) order to model the object.

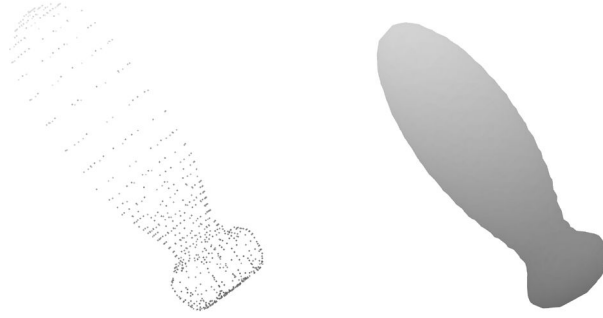


Figure 2.8: Example of a fitted screwdriver shaft using an implicit polynomial [63].

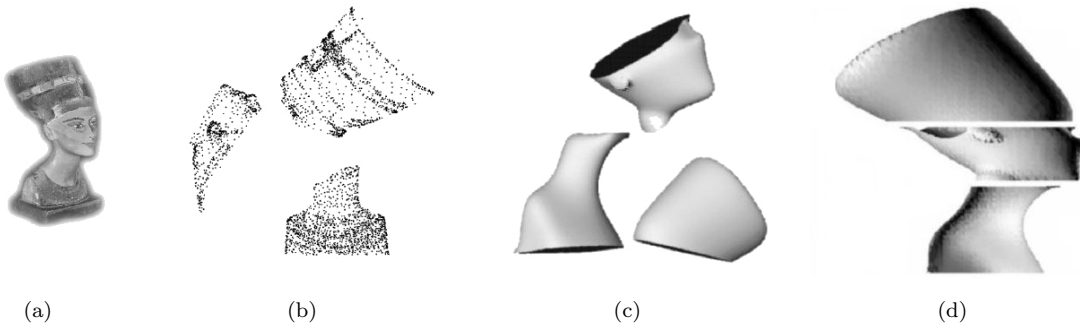


Figure 2.9: Composite fit of an Egypt bust using implicit polynomials of the 4th, 6th and 8th order [15].

Helzer et al. tackled the problem concerning the fit robustness and improved the stability compared to the introduced 3L-algorithm [50] achieving free-form fits with a better computational performance.

Spherical Harmonic Surfaces

Spherical harmonic surfaces are a subclass of the family of fourier surfaces using the spherical harmonic function Y_l^m of degree l and order m

$$Y_l^m(\theta, \phi) = \sqrt{\frac{2l+1}{4} \frac{(l-m)!}{(l+m)!}} P_l^m(\cos\theta) e^{im\phi}, \quad (2.33)$$

$$Y_l^{-m}(\theta, \phi) = (-1)^m Y_l^{m*}(\theta, \phi),$$

with the associated Legendre polynomials $P_l^m(x)$

$$P_l^m(x) = \frac{(-1)^m}{2^l l!} (1-x^2)^{\frac{m}{2}} \frac{d^{m+l}}{dx^{m+l}} (x^2-1)^l. \quad (2.34)$$

The parameterization of $\mathbf{r}(\theta, \phi)$ allows shape description of arbitray complex object surfaces running $0 \leq \theta < \pi$ and $0 \leq \phi < 2\pi$. Equation 2.35 gives the representation of a

spherical harmonic surface

$$\mathbf{r}(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l \mathbf{c}_l^m Y_l^m(\theta, \phi) \quad (2.35)$$

with the coefficients \mathbf{c}_l^m formed by the inner product of \mathbf{r} with the basis function.

$$\mathbf{c}_l^m = \int_0^\pi \int_0^{2\pi} \mathbf{r}(\theta, \phi) Y_l^m(\theta, \phi) \, d\phi \, \sin \theta \, d\theta \quad (2.36)$$

Figure 2.10 illustrates the reconstruction of the partial spherical harmonic series of $\mathbf{r}(\theta, \phi)$ using coefficients with the degree of 1, 3 and 7. Note that the spherical harmonic surface of first degree represents a Superquadric (2.15) with $\varepsilon_1 = 1$ and $\varepsilon_2 = 1$.

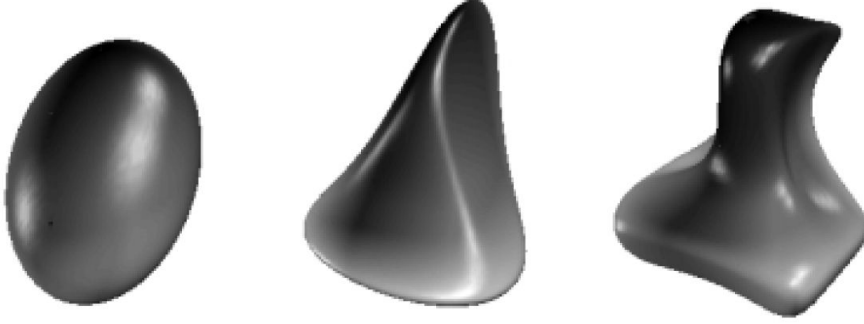


Figure 2.10: Example of spherical harmonic surfaces with the degree of 1, 3 and 7 [21].

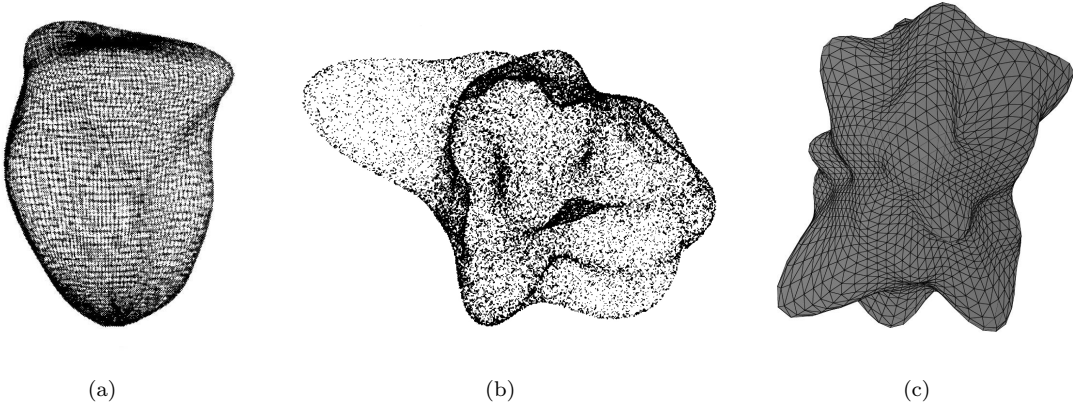


Figure 2.11: Some examples of spherical harmonic surfaces used in medical and biological applications: (a) cardiac [117], (b) insulin [82], (c) protein [107].

Spherical harmonic surfaces are mainly used in the medical and biological field of science where smoothed complex structures with a global and object-centered shape description are needed. The main applications of such a 3D description is the cardiac modeling

and the representation of molecular structures. Staib et al. [117] presented the recovery of a cardiac out of 3D computer tomography images illustrated in Figure 2.11(a). Note that the cardiac is computed up to an order of 8 with the large number of 321 parameter. Also much work has been done on the microscopic level modeling molecular structures. Max et al. [82] described in his work insulin structures with 8th order spherical harmonic surfaces (Figure 2.11(b)). More complex structures are recovered by Ritchie [107] using 12th order harmonic functions to describe protein molecules (Figure 2.11(c)). With that detailed modeling he tried to calculate the docking of several proteins using the surface shape to find matches [108].

2.2 Model Fitting

Generally, the literature on fitting can be divided into two general techniques, the clustering [7] and the least-squares fitting [70]. While the clustering methods are based on mapping data points to the parameter space, such as the Hough transform [54] and the accumulation methods, the least-squares methods are centered on finding the sets of parameters that minimizes some distance measures between the data points and the curve or surface. Because of the high computational cost of the clustering methods they are mainly used to fit lines and curves to point data sets in 2D space [33], while the least-squares method is the preferred technique of fitting geometric model's to 3D point data sets. As for this reason the least-squares fitting is reviewed in detail, first giving an overview of the used metrics to define the distance between a given point and the models surface and second, the state of the art of the least-squares methods is discussed related to the use in 3D point cloud processing.

2.2.1 Least-Squares Metric

Using the least-squares technique a distance measure is needed to compute the least-squares sum of the data points to the geometric model. In this section several metrics are presented to be used within the objective function of the minimization process. Numerating, these are: the algebraic distance, the Euclidean distance – also known as geometric distance – and the Taubin's approximation. Figure 2.12 clarifies the difference between the algebraic distance d_A and the Euclidean d_E distance demonstrated on the example of an ellipse.

Unfortunately, the minimization of the Euclidean distance from the data points to a general curve or surface is computationally impractical, because there is no closed-form expression and iterative methods are required to compute it. Thus, the Euclidean distance has been approximated with a first order approximation, suggested by Taubin [123]. However, experiments shown in [36] demonstrate the best results regarding quality and accuracy is obtained using the Euclidean distance, while using Taubin's approximation or the algebraic distance leads to a bias in the fitting results.

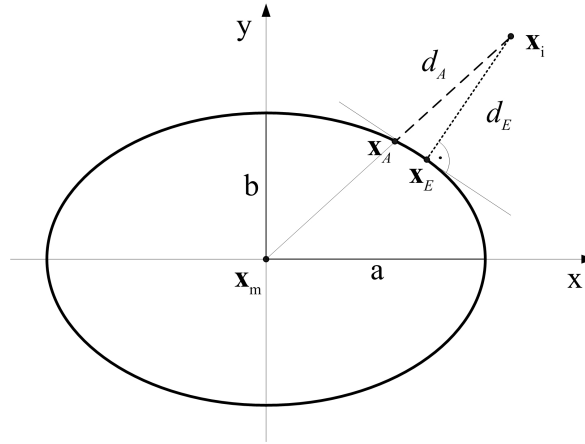


Figure 2.12: Illustrating the difference between the algebraic and Euclidean distance illustrated using an ellipse.

Algebraic Distance

The big advantage of the use of algebraic distances is the gain in computational efficiency. Closed-form solutions can usually be obtained providing a non-iterative fit measure approximating the Euclidean distance. However, the results are in some cases not satisfactory. There are at least two major reasons. First the function to minimize is usually not invariant under Euclidean transformations. This is a feature we dislike, because we usually do not know in practice where the best coordinate system to represent the data is. And second a point may contribute differently to the parameter estimation depending on its position on the surface. If a priori all points are corrupted by the same amount of noise, it is desirable for them to contribute the same way.

Consider the example of an ellipse in Figure 2.12, one can see that there is a significant difference between the algebraic and the Euclidean distance, which results in a different fit accuracy. Bookstein [19] gives the algebraic distance function of conic section. In this case the equation for the algebraic distance of an ellipse is given by

$$d_A(\mathbf{x}_i) = |\mathbf{x}_i - \mathbf{x}_m| \left(1 - \frac{|\mathbf{x}_i - \mathbf{x}_m|}{\sqrt{\left(\frac{x_i}{a}\right)^2 + \left(\frac{y_i}{b}\right)^2}} \right). \quad (2.37)$$

It is thus clear that a point at the high curvature sections contributes less to the conic fitting than a point having the same amount of noise but at the low curvature sections. This is because a point at the high curvature sections has a large value of $|\mathbf{x}_s - \mathbf{x}_m|$ and its squared distance d_A is small, while a point at the low curvature sections has a small $|\mathbf{x}_s - \mathbf{x}_m|$ and its squared distance d_A is higher with respect to the same amount of noise in the data points. Concretely, methods based on algebraic distances tend to fit better a conic to the points at low curvature sections than to those at high curvature sections. This problem is usually termed as high curvature bias.

Taubin's Approximation

A variant of the algebraic distance is given by Taubin [123] providing better fit results. This alternative approximate solution of the minimization problem is to replace the Euclidean distance from a point to an implicit curve or surface by the first order approximation [124]. There, the Taylor series is expanded up to the first order in a defined neighborhood, truncated after the linear term and then the triangular and the Cauchy-Schwartz inequality were applied.

An implicit curve or surface is the set of zeros of a smoothed function $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ of n variables: $\mathcal{Z}(f) = \{\mathbf{x} : f(\mathbf{x}) = 0\}$. In the applications for computer vision, especially for range image analysis we are interested in three special cases: $\mathcal{Z}(f)$ is a planar curve if $n = 2$ and $k = 1$, it is a surface if $n = 3$ and $k = 1$, and it is a space curve if $n = 3$ and $k = 2$. The Taubin's distance is then defined by

$$d_T(\mathbf{x}_i, \mathcal{Z}(f)) = \frac{|f(\mathbf{x}_i)|}{\|\nabla f(\mathbf{x}_i)\|}. \quad (2.38)$$

Euclidean Distance

To overcome the problems with the algebraic distances, it is natural to replace them by the Euclidean distances which are invariant to transformations in Euclidean space and which do not exhibit the high curvature bias. The Euclidean distance d_E between a point \mathbf{x}_i and a surface is the distance between \mathbf{x}_i and the point \mathbf{x}_E on the surface whose tangent is orthogonal to the line joining \mathbf{x}_i and \mathbf{x}_E (see Figure 2.12).

For primitive surfaces like planes, spheres, cylinder, cones and torus a closed form expression exists for the Euclidean distance from a point \mathbf{x}_i to the zero set (compare the distance Equations (2.2), (2.4), (2.6), (2.8) and (2.10) in Section 2.1.1). However, as the expression of the Euclidean distance to other curves or surfaces (e.g., the Superquadric) is more complicated and there exists no closed form expression, hence an iterative procedure to estimate the Euclidean distance must be carried out, suggested by Faber et al. [35]. This simple algorithm is briefly discussed:

1. Select the initial point $\mathbf{x}_t^{[0]}$. In the first step we determine the initial solution by intersecting the curve or surface with the straight line defined by the center point \mathbf{x}_m and the point \mathbf{x}_i . By the initial solution, the upper bound for the distance is estimated.
2. Update the actual estimation $\mathbf{x}_t^{[k+1]} = F(\mathbf{x}_t^{[k]})$. In the second step a new solution is determined. The search will be determined by the gradient of the curve or surface. Thus, the method is an adaption of the steepest descent method (see Section 2.2.2). As result of this step we get two possible solutions, $\mathbf{x}_t^{[k]}$ and $\mathbf{x}_t^{[k+1]}$, and we have to decide which one will be accepted as new local solution.
3. Evaluate the new estimation $\mathbf{x}_t^{[k+1]}$ if the distance is smaller and we accept this as a new local solution. Otherwise, we calculate a new estimation $\mathbf{x}_t^{[k+1]}$ with a smaller step in the current descent direction. Then, the algorithm will be continued with

step 2 until the difference between the distance of the old and the new estimation is smaller than a threshold.

As we have seen that the Equation (2.15) of the distance measure of the Superquadric represents the algebraic distance, a closed form expression of the Euclidean distance does not exist. Zhang evaluated in an experimental comparison two Superquadric objective functions using the algebraic and the radial Euclidean distance [142]. The radial Euclidean distance achieves better results but is not comparable with the real Euclidean distance. Tackling this problem Furong et al. [42] improved the Superquadric fit using an estimation of the Euclidean distance performed on Superquadrics with global deformations. The drawback of this method is the high computational cost, but achieving a high fit accuracy.

2.2.2 Nonlinear Least-Squares Methods

Fitting parametric models, such as described in Section 2.1, to range image data sets is usually made within an optimization process. Finding the global minimum is very hard to solve in general. The state of the art in model fitting is to solve a simpler problem of finding the local minimum. This implies that a starting solution is required to prevent the minimization process to be trapped in an arbitrary local minimum which does not correspond with the global minimum. Solving a least-squares problem is the standard method resulting in a minimum value of the so called objective function or cost function.

The least-squares problem is defined by finding \mathbf{x}^* , a local minimizer for

$$F(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^m (f_i(\mathbf{x}))^2, \quad (2.39)$$

where $f_i : \mathbb{R}^n \mapsto \mathbb{R}$, $i = 1, \dots, m$ are given functions, and $m \geq n$.

In the literature many methods can be found to solve the minimization problem using least-squares sums. But only four of them are recently used to recover geometric models in range data, that are: the *Steepest Descent* method, the *Newton's* method, the *Gauss-Newton* method and the *Levenberg-Marquardt* method. The first two methods are members of the so called descent methods finding a local minimum for general cost functions. A more tuned solution for least-squares problems are presented with the last two methods. Note that the Levenberg-Marquardt method has become the standard for nonlinear optimization routines. A detailed description of all of these methods can be found in [58, 89].

Decent Methods

All methods for non-linear optimization are iterative: From a starting point \mathbf{x}_0 the method produces a series of vectors $\mathbf{x}_1, \mathbf{x}_2, \dots$, which (hopefully) converges to \mathbf{x}^* , a local minimizer for the given function. Most methods have measures which enforce the descending condition

$$F(\mathbf{x}_{\mathbf{k}+1}) < F(\mathbf{x}_{\mathbf{k}}). \quad (2.40)$$

This prevents convergence to a maximizer and also makes it less probable that we converge towards a saddle point. If the given function has several minimizers the result will depend

on the starting point \mathbf{x}_0 . We do not know which of the minimizers will be found. It is not necessarily the minimizer closest to \mathbf{x}_0 .

In many cases the method produces vectors which converge towards the minimizer in two clearly different stages. When \mathbf{x}_0 is far from the solution we want the method to produce iterates which move steadily towards \mathbf{x}^* . In the final stage of the iteration, where \mathbf{x}_k is close to \mathbf{x}^* , we want faster convergence. One step from the current iterate consists in

1. Find a descent direction \mathbf{h} , and
2. find a step length α giving a good decrease in the F -value.

Consider the variation of the F -value along the half line starting at \mathbf{x} and with direction \mathbf{h} one can say that \mathbf{h} is a descent direction if $F(\mathbf{x} + \alpha\mathbf{h})$ is a decreasing function of α at $\alpha = 0$.

Steepest Descent Method When we perform a step $\alpha\mathbf{h}$ with positive α , then the relative gain in function value satisfies

$$\lim_{\alpha \rightarrow 0} \frac{F(\mathbf{x}) - F(\mathbf{x} + \alpha\mathbf{h})}{\alpha\|\mathbf{h}\|} = \frac{1}{\|\mathbf{h}\|} \mathbf{h}^\top \mathbf{F}'(\mathbf{x}) = -\|\mathbf{F}'(\mathbf{x})\| \cos \theta, \quad (2.41)$$

where θ is the angle between the vectors \mathbf{h} and $\mathbf{F}'(\mathbf{x})$. This shows that we get the greatest gain rate if $\theta = \pi$, that is if we use the steepest descent direction \mathbf{h}_{sd} given by

$$\mathbf{h}_{sd} = -\mathbf{F}'(\mathbf{x}). \quad (2.42)$$

The method based on (2.42) is called the *Steepest Descent* method or *Gradient* method. A method like this converges, but the final convergence is linear and often very slow. For many problems, however, the method has quite good performance in the initial stage of the iterative process. Considerations like this have led to the so-called hybrid methods, which - as the name suggests - are based on two different methods. One which is good at the initial stage, like the gradient method, and another method which is good at the final stage, like Newton's method. A major problem with a hybrid method is the mechanism which switches between the two methods when appropriate. An implementation in C of the Steepest Descent algorithm is given in [103] on page 318 and the following.

Newton's Method We can derive this method from the condition that \mathbf{x}^* is a stationary point satisfying $\mathbf{F}'(\mathbf{x}^*) = 0$. This is a nonlinear system of equations, and from the Taylor expansion

$$\begin{aligned} \mathbf{F}'(\mathbf{x} + \mathbf{h}) &= \mathbf{F}'(\mathbf{x}) + \mathbf{F}''(\mathbf{x})\mathbf{h} + O(\|\mathbf{h}\|^2) \\ &\simeq \mathbf{F}'(\mathbf{x}) + \mathbf{F}''(\mathbf{x})\mathbf{h}, \quad \text{for } \|\mathbf{h}\| \ll \end{aligned} \quad (2.43)$$

we derive *Newton's* method: Find \mathbf{h}_n as the solution to

$$\mathbf{H}\mathbf{h}_n = -\mathbf{F}'(\mathbf{x}), \quad (2.44)$$

with \mathbf{H} the Hessian

$$\mathbf{H} \equiv \mathbf{F}''(\mathbf{x}) = \left[\frac{\partial^2 F}{\partial x_i \partial x_j}(\mathbf{x}) \right], \quad (2.45)$$

and compute the next iteration by

$$\mathbf{x} := \mathbf{x} + \mathbf{h}_n. \quad (2.46)$$

Newton's method is very good at the final stage of the iteration, where \mathbf{x} is close to \mathbf{x}^* . One can show that if the Hessian at the solution is positive definite and if we are at a position inside the region around \mathbf{x}^* where $\mathbf{F}''(\mathbf{x})$ is positive definite, then we get quadratic convergence. On the other hand, if \mathbf{x} is in a region where $\mathbf{F}''(\mathbf{x})$ is negative definite everywhere, and where there is a stationary point, the basic Newton method (2.44) and (2.46) would converge (quadratically) towards this stationary point, which is a maximizer. We can avoid this by requiring that all steps taken are in descent directions.

We can build a hybrid method, based on Newton's method and the Steepest Descent method. The Newton step is guaranteed to be downhill if $\mathbf{F}''(\mathbf{x})$ is positive definite, so a sketch of the central section of this hybrid algorithm could be:

$$\begin{aligned} &\text{if } \mathbf{F}''(\mathbf{x}) \text{ is positive definite} \\ &\quad \mathbf{h} := \mathbf{h}_n \\ &\text{else} \\ &\quad \mathbf{h} := \mathbf{h}_{sd} \\ &\quad \mathbf{x} := \mathbf{x} + \alpha \mathbf{h} \end{aligned} \quad (2.47)$$

Non-Linear Least-Squares Problems

In this section methods for nonlinear least-squares problems are discussed. Given a vector function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ with $m \geq n$. We want to minimize $\|\mathbf{f}(\mathbf{x})\|$, or equivalently to find

$$\mathbf{x}^* \operatorname{argmin}_{\mathbf{x}} \{F(\mathbf{x})\}, \quad (2.48)$$

where

$$F(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^m (f_i(\mathbf{x}))^2 = \frac{1}{2} \|\mathbf{f}(\mathbf{x})\|^2 = \frac{1}{2} \mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}). \quad (2.49)$$

Least-squares problems can be solved by general optimization methods, but we shall present special methods that are more efficient. In many cases they achieve better than linear convergence, sometimes even quadratic convergence, even though they do not need implementation of second derivatives.

The two methods explained in this section use the Hessian \mathbf{H} and the Jacobian \mathbf{J}

$$\mathbf{J}(\mathbf{x}) = \left[\frac{\partial f_i}{\partial x_j}(\mathbf{x}) \right] \quad (2.50)$$

to show that

$$\mathbf{H} = \mathbf{F}''(\mathbf{x}) = \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) + \sum_{i=1}^m f_i(x) \mathbf{f}_i''(\mathbf{x}). \quad (2.51)$$

Gauss-Newton Method This method can be regarded as an approximation to Newton's method and is the basis of the very efficient Levenberg-Marquardt method that will be discussed in the next paragraph. It is based on implemented first derivatives of the components of the vector function. In special cases it can give quadratic convergence as the Newton-method does for general optimization.

The *Gauss-Newton* method is based on a linear approximation to the components of \mathbf{f} in the neighbourhood of \mathbf{x} : For small $\|\mathbf{h}\|$ one can see from the Taylor expansion that

$$\mathbf{f}(\mathbf{x} + \mathbf{h}) \simeq \ell(\mathbf{h}) \equiv \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\mathbf{h}. \quad (2.52)$$

Inserting this in definition (2.49) of F one can see that

$$\begin{aligned} F(\mathbf{x} + \mathbf{h}) &\simeq L(\mathbf{h}) \equiv \frac{1}{2}\ell(\mathbf{h})^\top \ell(\mathbf{h}) \\ &= \frac{1}{2}\mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) + \mathbf{h}^\top \mathbf{J}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) + \frac{1}{2}\mathbf{h}^\top \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x})\mathbf{h} \\ &= F(\mathbf{x}) + \mathbf{h}^\top \mathbf{J}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) + \frac{1}{2}\mathbf{h}^\top \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x})\mathbf{h} \end{aligned} \quad (2.53)$$

The Gauss-Newton step \mathbf{h}_{gn} minimizes $L(\mathbf{h})$,

$$\mathbf{h}_{gn} = \operatorname{argmin}_{\mathbf{h}} \{L(\mathbf{h})\}. \quad (2.54)$$

Further, the matrix $\mathbf{L}''(\mathbf{h})$ is independent of \mathbf{h} . It is symmetric and if \mathbf{J} has full rank, then $\mathbf{L}''(\mathbf{h})$ is also positive definite. This implies that $L(\mathbf{h})$ has a unique minimizer, which can be found by solving

$$(\mathbf{J} \equiv \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}))\mathbf{h}_{gn} = -\mathbf{J}(\mathbf{x})\mathbf{f}(\mathbf{x}). \quad (2.55)$$

This is a descent direction for F since

$$\mathbf{h}_{gn}^\top \mathbf{F}'(\mathbf{x}) = \mathbf{h}_{gn}^\top (\mathbf{J}(\mathbf{x})\mathbf{f}(\mathbf{x})) = -\mathbf{h}_{gn}^\top (\mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}))\mathbf{h}_{gn} < 0. \quad (2.56)$$

Thus, we can use \mathbf{h}_{gn} for \mathbf{h}_d in the descent methods. The typical step is

$$\begin{aligned} &\text{Solve } (\mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}))\mathbf{h}_{gn} = \mathbf{J}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) \\ &\mathbf{x} := \mathbf{x} + \alpha \mathbf{h}_{gn} \end{aligned} \quad (2.57)$$

where α is found by line search. The classical Gauss-Newton method uses $\alpha = 1$ in all steps. In the description of the Newton's method one can observe a quadratic convergence for the optimization. This is normally not the case with the Gauss-Newton method.

Levenberg-Marquardt Method Levenberg [73] and later Marquardt [79] suggested to use a damped Gauss-Newton method. The step \mathbf{h}_{lm} is defined by the following modification to (2.55),

$$(\mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) + \mu \mathbf{I})\mathbf{h}_{lm} = -\mathbf{g}, \quad \text{with } \mathbf{g} = \mathbf{J}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) = \left[\frac{\partial F}{\partial x_i}(\mathbf{x}) \right] \quad (\text{Gradient}) \quad \text{and } \mu \geq 0. \quad (2.58)$$

The damping parameter μ has several effects:

- a) For all $\mu > 0$ the coefficient matrix is positive definite, and this ensures that \mathbf{h}_{lm} is a descent direction, compare (2.56).
- b) For large values of μ we get

$$\mathbf{h}_{lm} \simeq -\frac{1}{\mu}\mathbf{g} = -\frac{1}{\mu}\mathbf{F}'(\mathbf{x}), \quad (2.59)$$

that is a short step in the steepest descent direction. This is good if the current iterate is far from the solution.

- c) If μ is very small, then $\mathbf{h}_{lm} \simeq \mathbf{h}_{gn}$, which is a good step in the final stages of the iteration, when \mathbf{x} is close to \mathbf{x}^* . If $F(\mathbf{x}^*) = 0$ (or very small), then we can get (almost) quadratic final convergence.

Thus, the damping parameter influences both the direction and the size of the step, and this leads us to make a method without a specific line search. The choice of initial μ -value should be related to the size of the elements in $\mathbf{A}_0\mathbf{J}(\mathbf{x}_0)^\top\mathbf{J}(\mathbf{x}_0)$, e.g. by letting

$$\mu_0 = \tau \cdot \max_i \{a_{ii}^{(0)}\}, \quad (2.60)$$

where τ is chosen by the user.¹ During iteration the size of μ can be updated which is controlled by the gain ratio

$$\varrho = \frac{F(\mathbf{x}) - F(\mathbf{x} - \mathbf{h}_{lm})}{L(0) - L(\mathbf{h}_{lm})}, \quad (2.61)$$

where the denominator is the gain predicted by the linear model (2.53),

$$\begin{aligned} L(0) - L(\mathbf{h}_{lm}) &= -\mathbf{h}_{lm}^\top \mathbf{J}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) - \frac{1}{2} \mathbf{h}_{lm}^\top \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) \mathbf{h}_{lm} \\ &= -\frac{1}{2} \mathbf{h}_{lm}^\top (2\mathbf{g} + (\mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) + \mu \mathbf{I}) \mathbf{h}_{lm}) \\ &= \frac{1}{2} \mathbf{h}_{lm}^\top (\mu \mathbf{h}_{lm} - \mathbf{g}). \end{aligned} \quad (2.62)$$

Note that both $\mathbf{h}_{lm}^\top \mathbf{h}_{lm}$ and $-\mathbf{h}_{lm}^\top \mathbf{g}$ are positive, so $L(0) - L(\mathbf{h}_{lm})$ is guaranteed to be positive.

A large value of ϱ indicates that $L(\mathbf{h}_{lm})$ is a good approximation to $F(\mathbf{x} + \mathbf{h}_{lm})$, and we can decrease μ so that the next Levenberg-Marquardt step is closer to the Gauss-Newton step. If ϱ is small (maybe even negative), then $L(\mathbf{h}_{lm})$ is a poor approximation, and we should increase μ with the twofold aim to get closer to the steepest descent direction and reduce the step length.

The stopping criteria for the algorithm should reflect that, at a global minimizer, we have $\mathbf{F}'(\mathbf{x}^*) = \mathbf{g}(\mathbf{x}^*) = 0$, so we can use

$$\|\mathbf{g}\|_\infty \leq \varepsilon', \quad (2.63)$$

¹The algorithm is not very sensitive to the choice of τ , but by rule of thumb, one should use a small value, e.g. $\tau = 10^{-6}$ if \mathbf{x}_0 is believed to be a good approximation to \mathbf{x}^* . Otherwise, use $\tau = 10^{-3}$ or even $\tau = 1$.

where ε' is a small, positive number, chosen by the user. Another relevant criterion is to stop if the change in \mathbf{x} is small,

$$\|\mathbf{x}_{new} - \mathbf{x}\| \leq \varepsilon''(\|\mathbf{x}\| + \varepsilon''). \quad (2.64)$$

This expression gives a gradual change from relative step size ε'' when $\|\mathbf{x}\|$ is large to absolute step size ε''^2 if \mathbf{x} is close to 0. Finally, as in all iterative processes we need a safeguard against an infinite loop,

$$k \geq k_{max}. \quad (2.65)$$

Also ε_2 and k_{max} are chosen by the user.

The last two criteria come into effect example given if ε' is chosen so small that effects of rounding errors have large influence. This will typically reveal itself in a poor accordance between the actual gain in F and the gain predicted by the linear model (2.53), and will result in μ being augmented in every step. The strategy for augmenting μ implies that in this case μ grows fast, resulting in small $\|\mathbf{h}_{lm}\|$, and the process will be stopped by (2.64). Detailed information about the algorithm are described in [85] and the C source code can be found in [103] on page 542 and the following.

2.3 Model-Based Object Detection

This section gives the state of the art in model-based object detection using cylinders, as members of the geometric primitives, and Superquadrics. This literature review focuses on object detection dealing with 3D point clouds acquired from one-view laser range scanners which paves the way to the methods presented in this thesis.

2.3.1 Cylinders

Dealing with geometric primitives in range images for segmentation, object recognition and detection tasks is mainly based on estimating the mean and the Gaussian curvature (H-K estimation) [133] as shown in the model-based object detection approach in [18]. Here, we concentrate on cylindrical shapes and in literature a lot of curvature-based approaches have been introduced to detect cylinders in range images. Yokoya et al. [140] achieved an improved detection robustness by using a hybrid approach combining H-K segmentation with a jump- and roof-edge detection. Restricting the search for cylindrical shapes in range images the estimation of the radius of curvature is sufficient for a geometrical description as Goulette demonstrated with an application of pipe segmentation [43]. Hameiri et al. [45] based his work on principal curvature histograms with the result of detecting cylinder fragments within a set of arbitrary shaped objects. Another approach introducing a surface type classification for object detection was proposed by Taylor et al. [126] analyzing the Gaussian image and the convexity of surface patches. In contrast to curvature-based approaches, Marshall et al. [80] follows a different approach by fitting least-squares models to segment a scene for object detection, while Attene et al. [1] approximates the object with cylinders, an object description which is based on a mesh representation.

While there is a large body of work on segmenting and localizing convex shaped objects, e.g., [53], there are only a few approaches to cope with concave shapes, especially cylindrically shaped bore holes. The intuitive approach to find a bore hole is to detect the circle of the bore hole entrance. This can be achieved by detecting the bore edge applying the scan line approximation [59]. This method works for simple bore holes with the restriction that the bore hole must be perpendicular to the surrounding surface and that the bore entrance has a sharp edge (no rounding or chamfer). Hence, that edge detection is not robust on non-perpendicular crossing scan lines, because detected edge points are not well distributed around the fitted circle and adulterate the results.

An improved approach leads to a bore hole surface segmentation based on the idea of the fast cylinder extraction algorithm [26] with a subsequent cylinder fit. This method has the advantage that the bore hole might be slanted and the bore entrance might be rounded (cast parts) or chamfered. Due to real time specification the cylinder surface fitting using the Euclidean or algebraic distance or using Taubin's approximation is too time consuming [35]. Furthermore these fittings need a rather large cylinder surface for good results. Also considered was the segmentation and fitting method [80], but this algorithm is too complex and time consuming for fast industrial applications. The idea of the robust axis determination of rotational symmetric parts [139] is used for an improved and advanced determination of the cylinder pose. Building on these techniques Chapter 4 introduces a sample based approach, such as the RANSAC algorithm [40], for a cylinder estimation which satisfies the required accuracy, processing time request and operating on sparse data.

2.3.2 Superquadrics

Solina et al. pioneered work in recovering single Superquadrics with global deformations in a single-view point cloud [116] and demonstrated that the recovery of a Superquadric from range data is sensitive to noise and outliers, which renders a stable object recognition difficult. To overcome this problem many approaches assume that full 3D data of the objects is available to estimate the complete set of model parameters [27, 39, 71]. On the other hand, full 360° views are difficult to obtain in practice and for object grasping it is sufficient to estimate pose parameters while the task constraint specifies shape and size parameters [111, 127].

Much progress has been made in the last decade in tackling the recognition problem by acquiring a single-view range image and interpreting the scene using Superquadrics. Leonardis et al. introduced the recover and select paradigm for segmenting a scene with simple geometric objects without occlusions [72]. This method aims at a full search with an open processing time incompatible to robotic applications. Subsequent work has improved this segmentation, e.g. Krivic et al. demonstrates the recovery of a known complex object in a scene using the connectivity information of the Superquadrics [67] handling the scene occlusions by using the redundancy information of the part connections. Tao et al. has also presented an improvement of Leonardis' segmentation [122] using random samples with the focus on speeding up the Superquadric fit in noisy range data. This approach comes close to filling the needs of robotic applications but only using undeformed

Superquadrics and concentrating his work on detecting pipes in sonar range images. Taylor et al. first segments the image for describing the scene [126] but saves processing time with the limitation of detecting single geometric primitives disclaiming the flexibility of Superquadrics to describe objects.

Katsoulas proposes a novel object detection approach searching for box-like objects using parabolically deformable Superquadrics [61]. He weakened the bottleneck of the scene segmentation using a 3D edge detector and achieved some improvement in processing time, but this method cannot handle non box-like objects and scene occlusions.

All these approaches use segmentation to analyze the scene. It is however argued that a bottom up segmentation is in conflict with purposeful object detection. The detection of a known object does not require segmentation. Moreover, segmentation wastes valuable processing time as shown in the experimental comparison of different range image segmentation algorithms [53]. One of the main contributions of the approach, proposed in Chapter 5, achieving fast object detection is the lack of necessity for any segmentation.

Chapter 3

Automated Industrial Production

In the past three decades the robot plays the main role in industrial automation. The robot is used in a large variety of different application areas programmed to execute manufacturing tasks such as assembly, inspection, welding or painting. Handling large batch sizes of industrial parts was necessary for a profitable use of the robot. Integrating more and more sensors on the robot increased its flexibility to operate in an adaptive production environment. One of the most powerful sensors are vision systems, which gives the robot the ability to work autonomously and for small batch size. In the last recent years serious research work has been made in the aspect of robotic lot size one production. This was the main motivation of my work providing the robot a 3D vision system to operate flexibly in terms of a fast and robust way.

My project and research work was related with the development of three robotic projects dealing with low batch sizes. I was involved in these projects contributing work in range sensing and calibration as well as in the range image processing. This chapter should give an overview of my work's background where fast 3D object detection is needed for autonomous robot handling.

The first robotic project that is presented in this chapter is a fully automated spray painting application for low batch sizes with a very high number of part variants called *FlexPaint*. Generally, industrial spray painting is characterized by off-line robot programming and manual teach-in methods. Achieving lot size one production FlexPaint introduced the “inverse” approach: on-line robot path planning based on scanned 3D sensor data from the part to be painted [9, 99, 136]. The FlexPaint approach is based on formalizing the technological knowledge in a geometry library and a process library. Laser range sensors are used to obtain an accurate scan of the part. Process-relevant classes of features are detected as specified in the geometry library. Feature classes are linked in the process library to basic paint strategies, which are grouped to automatically generate the robot paint tool trajectory. Finally collisions-free and executable robot motions are automatically obtained for the actual robot kinematics.

Lot size one manufacturing implies an automated and highly flexible system with less user interaction as possible. Due to the great success in the research work of the project

FlexPaint a spin-off project was started to commercialize the system. One important open task within the following project called *Lot Size One* was the calibration of the sensor cell because the procedure of the FlexPaint prototype was rather complicated and time consuming. In an industrial environment the range sensors need a fully automated calibration to be quickly re-calibrated, which is often an requirement in high quality production. Tackling this problem we investigated a novel geometric approach to simultaneously calibrate a multi-view laser stripe profiler [118]. The first and most important enhancement is a new design of the calibration object. Basically it is a “wire-frame model” of a cube, where the edges are simply standard aluminum profiles. This allows establishing point correspondences with high precision, overcoming the systematic errors when using circular marks. Furthermore this hollow kind of a calibration object allows simultaneous multi-view calibration since enough points of six different planes can be seen from many directions. Given a rough calibration the scanned points can be automatically allocated to the respective planes of the calibration object. The second enhancement is a new way of solving the linear model of a laser stripe profiler. By firstly solving the camera-independent geometry these values can be averaged over the multiple views, which further reduces the local offsets of point clouds and therefore enhances the output of a subsequent mesh generation.

The last project to be introduced is **FibreScope**. The focus on this work was the automation of a robotic inspection task for high variant parts [11, 13]. As well as in the project FlexPaint a 3D vision system is used to guide an endoscope in a robotic bore surface inspection application [12]. Bores and internal threads are critical parts of connections, bearings and engines, hydraulic and pneumatic systems and defective bore surfaces can cause problems. Current systems miss a high flexibility or are mainly driven by human interaction and control. Therefore the main target of this work is to develop an automatic robotic system for rapid and flexible 100% surface inspection of bores with diameters from 4 to 50mm. The main focus is to automatically detect and localize bores on arbitrary metallic objects. Using a laser scanner range images are acquired, the bore is automatically located, and an automatic endoscopic inspection with a vision system is started. The main focus is on easy operating of the system achieved with rapid programming for the user.

We used a highly flexible CAD-based approach with a robot to move an endoscope in 6 DoF to inspect each bore around the work piece. To compensate large uncertainties in 3D space we used a laser range sensor, which is also mounted on the robot arm, to manage a sensor-based detection of the position and orientation of the bore. We chose a very rapid, ergonomic and simple way of programming the system with new work pieces via a CAD editing tool, which enables the user to efficiently generate the robot program within a few minutes. This also includes a collision avoidance check of the robot movement. While inspection the endoscope is protected with a new developed clearance check that retires the endoscope immediately in case of a collision while inspecting.

The three projects are now briefly introduced and will be described more in detail in the next three sections starting with the project FlexPaint, followed by the project Lot Size One and finally concluding with the project FibreScope.

3.1 FlexPaint

First approaches to obtain an automatic tool path in 3D are known from milling turbine blades by considering planar cross sections [69, 114] or using a grid cell approach [135]. Assisting the programmer with virtual reality tools and automated path planning in simulations is another approach [41].

Further related work is the automatic generation of a 3D paint path, which has been attempted in the SmartPainter project. The painting motion was generated by virtually folding out the surfaces to be painted, putting on the painting motion, folding back the surfaces and letting the painting motions follow the unfolded surfaces [51, 94]. However, this strategy is only applicable when 3D models of the objects are available and the curvature of the objects is relatively low.

The patented technology from Advanced Robotics Technologies uses a 2D digital photo as input [US patent no. US 5,429,682]. The user decides on the screen where to apply paint strokes. The path planning for a robot is then done automatically.

3.1.1 The FlexPaint Idea

The FlexPaint approach is based on the observation that the parts in Figure 3.1 comprise a large number of *elementary geometries* with typical characteristics for an entire product family. Examples are rib-sections (cooling ribs), cylindrical surfaces (both shown on the motor in Figure 3.1(b)) and cavities (shown at the top of the gear-box in Figure 3.1(b) and at the steering column in Figure 3.1(d)). Another type of surface is the surface of the car mirror. These surfaces are smooth free-form surfaces, which are very difficult to represent by the use of simple geometric attributes such as cylinders, spheres and boxes. Hence, the goal is to specify these elementary geometries in such a way that generic methods for detecting and for path planning can be developed and that the variety of geometries seen in the applications is encompassed. Figure 3.2 sketch the functionality the FlexPaint system works.

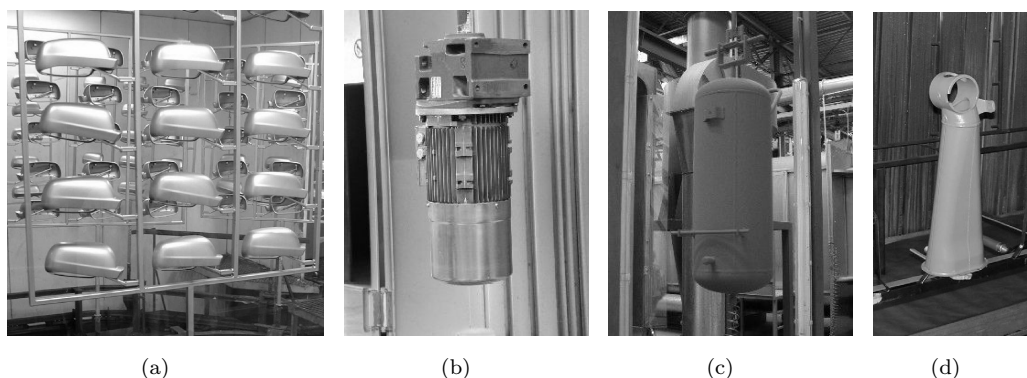


Figure 3.1: Example of parts to be painted automatically: (a) car mirrors from FICO, (b) ROSSI gearbox with motor, (c) FINI compressor tank, (d) and a truck steering column from MAN.

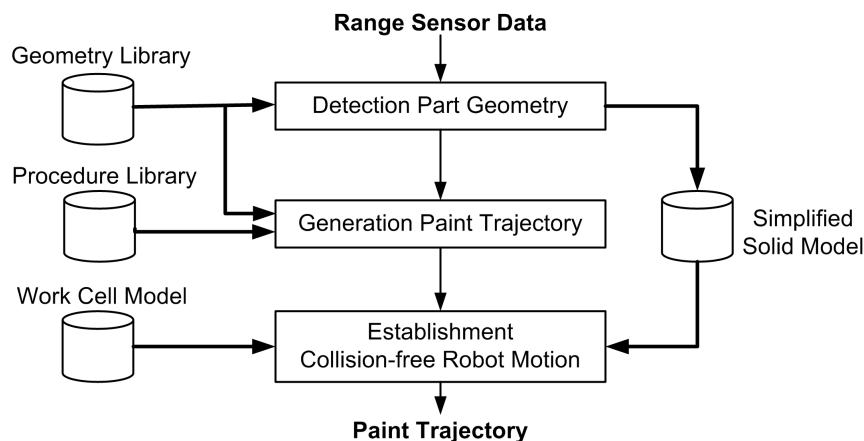


Figure 3.2: Flow Chart of the FlexPaint system.

The module “Detection Part Geometry” uses the geometric definitions of the *Geometry Library* and links them to the process knowledge, which is specified in the *Procedure Library* (compare Figure 3.2). Additionally a simplified solid model is calculated, which represents a convex hull approximation of the part. It is utilized to model the part when generating collision-free motions.

3.1.2 Detection of Process Relevant Features

The idea is to link elementary geometries to a process model. The elementary geometry types are defined in the *Geometry Library* and related to the process knowledge, which is specified in the *Procedure Library* (compare Figure 3.2). For example, a free-form surface needs painting normal to the surface, a cavity needs spraying into the cavity, and a rib section parallel to the ribs.

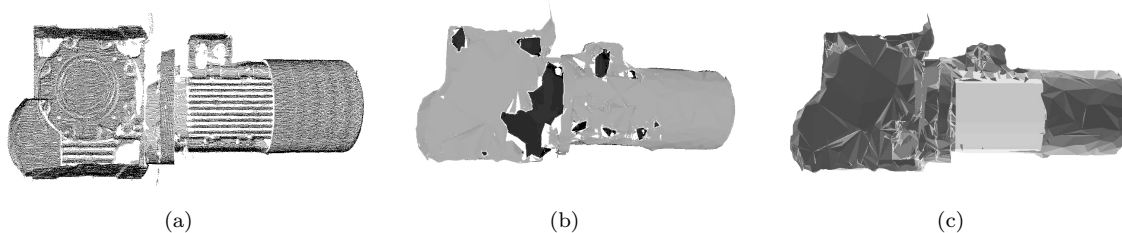


Figure 3.3: Detection of process relevant features out of the captured range image (a): cavities (b) and rib section (c).

Range image processing typically starts from segmentation, e.g., see comparison in [53]. Finding features is then constrained to defined geometric properties such as planes or conics [110, 80]. However, for the task at hand another approach is needed, because the geometry of the features is not known. The features are defined by attributes (lower than surrounding rim, parallel ribs). Hence, a more generic approach to feature detection is required, which detects *classes of features* [9, 137]. The tool developed is referred to as

“FeatureFinder”. Figure 3.3 shows the acquired range image and the detected cavity and rib section.

Cavity Detection

A cavity is defined as a region where surface points are locally lower (in the sense of an outward surface normal) than a surrounding rim. The challenge is to develop a robust procedure to handle noise and shadows of the range data. The triangulation range sensor presents the data with a natural neighborhood of points in the scan line and normal to the scan line. For each range image pixel these two orthogonal directions are investigated searching for locally lower data points. This procedure is repeated for each scan line and is also executed on lines in the scan direction to find a closed rim around classified cavity points. The cavity points classified are clustered using VTK methods. Cavity points are triangulated using mesh generation in a small neighbourhood. The cavity is then represented as the mesh that covers the opening or lid of the cavity. The mesh representation also has the advantage that it allows significant data reduction to render automated generation of the painting trajectory fast. See the result in Figure 3.3(b).

Rib Detection

A rib section is defined by a minimal number (five) of equidistant parallel lines. The first step is to use VTK to generate a pseudo 3D gray scale image out of the 3D range data. Then an edge map is generated using the Canny algorithm [24]. The challenge is now to extract the rib section out of this edge image by finding long, parallel and equidistant lines. The described methods in [55, 74] have been adopted to solve this problem. First the short edge segments are grouped to a set of long lines which are then grouped to a set of parallel long lines. After that the set of long lines is filtered to find sets of long lines with similar length and relative closeness. This procedure detects rib sections and the rib lines are mapped back into the 3D-space of the range image. In the last step the groups are merged and the boundary of the merged regions calculated using VTK. Finally, the rib section is again represented with a mesh (see Figure 3.3(c)).

3.1.3 Planning of the Robot Painting Motions

The module “Generate Painting Trajectory” shown in Figure 3.2 which is referenced as “PaintPlanner”, specifies a trajectory of the spray gun, which satisfies the requested paint quality. In this module only spray gun motions are considered in relation to process quality.

One input for the PaintPlanner, provided by the FeatureFinder, is a VRML file containing the mesh geometry of the free-form surface as well as of the cavity and rib section. Second, the process related input is a procedure library which specifies a number of painting procedures for use in robotic painting such as distance, speed and angles as well as the parameters of the spray gun. The last input concerns the relation between the specific geometric part specifications and the painting procedures specified in the geometry library. For instance, a free-form surface requires another treatment in order to meet the respective paint quality than any kind of cavity structure.

The approach of generating the paint trajectories is to substitute the original part geometry specification by mesh surfaces into a set of virtual surfaces (planes) having fixed directions in cartesian space, and to relate the painting procedures to these virtual surfaces instead of relating them to the original surfaces (compare Figure 3.4(a)). This gives the advantage that continuous paint strokes can be achieved even though the surface has an irregular shape. The optimized paint strokes for the virtual faces are then specified in a file describing start and stop positions of the spray gun calculated by homogeneous transformations and correspondence to each stroke. Figure 3.4(b) presents the calculated paint strokes located on the main faces.



Figure 3.4: (a) Generation of the main faces on the motor and gearbox (reduced VRML format). (b) Calculated paint strokes located on the main faces.

3.1.4 Anti-Collision Detection of the Robot Motions

The output from the PaintPlanner is a specification of the required paint trajectories. The collision avoidance software is responsible for converting this task description into a robot program that results in the specified paint strokes while it avoids collision and respect the limitations of the robot actually used in the paint cell. The collision avoidance software (MotionPlanner) consist of three independent successive modules: The ToolPlanner, the RobotPlanner, and the OutputGenerator. These modules gradually refine the task description provided by the PaintPlanner until the generic task description is finally converted into robot program.

The ToolPlanner will make the modifications of the task to remove the collisions. The ToolPlanner has the freedom to modify the pose of the paint gun by translating and rotating it around the specified tool center point within certain limits specified by the paint procedure. In addition to the task description the ToolPlanner also takes geometry files with spatial information about the paint tool and part being painted as input. Figure 3.5 shows an example of a paint trajectory generated by the PaintPlanner and the resulting paint trajectory after it has been modified by the ToolPlanner.

The RobotPlanner further refines the task description by selecting proper robot configurations and inserting intermediate targets so that the complete robot motion is fully defined. The input to the RobotPlanner is the collision-free paint trajectories and the output is a robot motion that is guaranteed to be collision-free and within the limitations of the robot, that is joint limits and joint speed limits. The PathPlanner also, indirectly, handles singularities. The singularity problem is that joint speeds can reach the upper



Figure 3.5: (a) Original paint trajectory generated by the PaintPlanner. (b) Paint trajectory after it has been modified by the ToolPlanner.

limit and hence the planner forces the tool motion to slow down when the robot approaches a singular configuration.

Finally the OutputGenerator converts the collision-free robot motion into a robot program in the specific language used by the robot controller. The resulting robot program will be sent directly to the robot controller and executed when the part reaches the paint cell.

3.2 Lot Size One Calibration

Physical models of a laser stripe profiler need less modeling effort to deliver accurate calibration results [22], hence we concentrate here on methods based thereon, see also [83] for a more complete treatment of calibration methods. Linear (physical) models neglect lens distortions. In [129] and [83] a linear model is used for generating a start value. Their non-linear models for refining the calibration only consider radial distortion with one parameter, which seems to be sufficient for state of the art lenses, even with a rather wide angle of view.

Since in a linear model the relation between corresponding points in the laser plane and points in the sensor plane can be modeled by a projective mapping, it suffices to find the corresponding matrix [106] without knowing the intrinsic (pixel scale, focal point, focal length, pixel skew) and extrinsic (position, optical axis) camera parameters.

Many calibration methods are at least partly based on first establishing correspondences between points in 3D and points in the sensor plane. An often used way of establishing point correspondences is to use the centers of circular marks that are mapped to elliptic regions in the sensor plane. But since a projective mapping does not keep proportions (only the cross-ratio is an invariant) the center of mass of the elliptic regions in the sensor plane will not exactly correspond to the aimed center of the original circle. Hence, the use of circular marks, as done in [129], introduces a systematic error into the calibration.

In [106] a calibration object, consisting of 6 planes, is used (basically a “cube in the corner of a room”) but the calibration method described there is not elaborated to full generality. Also this calibration object does not allow simultaneous multi-view calibration since already rather small deviations from the optimal point of view will lead to the

occlusion of at least one necessary plane.

In [60] a self-calibration method for a multi-view a laser stripe profiler is introduced. But this approach needs a pre-computed calibration and a highly geometric structured calibration object, because the proposed calibration refinement is based on a simultaneous multi-view registration of the scans. Also the registration process is very time consuming and not usable for industrial applications.

3.2.1 Calibration Approach

The proposed calibration approach is designed to work fully automated and simultaneously for several totally opposite camera views (even from different sides of the laser plane, see Figure 3.6(b)). The non-calibrated range data obtained from the calibration object together with a precise measurement of the axis movement between profiles and rough measurements of the scanning cell layout (often available from existing layout designs) are the only input to our calibration method, where no more interaction of an operator is needed for calibration.

For establishing point correspondences we use a hollow calibration object that allows seeing enough points of six different planes from most points of view. Triples of these planes (fitted with RANSAC) are used to extract feature points with high (sub-pixel) accuracy. Since planes in world coordinates are not mapped precisely onto planes in the (non-Cartesian) sensor coordinate frame, we do the plane fitting and intersection after transforming the scanned points to world coordinates with the available start value for the calibration. A non-perfect calibration does not totally restore the plane-incidence of points originating from one plane, but iteratively repeating the computation of the calibration based on thus generated point correspondences converges rather fast (3 iterations mostly suffice).

For the actual computation of the calibration we could use one of the existing methods mentioned above. Nevertheless we propose a new solution for the linear model¹. Our approach to solving the linear model easily allows to average the results of different cameras. We firstly compute the camera independent geometry (normal vector of laser plane, vector of axis movement), from which we can compute the intersection points of the line of travel with the laser plane. These intersection points we refer to as *piercing points*. The perspective projection for each camera view is calculated using known feature points on the calibration object whose piercing points (in moving direction with the laser plane) are mapped onto the image plane of the cameras. The piercing points can be averaged from the data of different cameras, which improves the local matching of the calibrated point clouds.

3.2.2 Multi-View Calibration Object

The calibration method works on pure range data and therefore there is no need for markers or intensity information. At least five significant points on the calibration object have to

¹If the radial or tangential lens distortions are not negligible, they can be estimated by available tools [49, 134, 144, 146] and eliminated in a preliminary step.

be extracted with high precision for each camera view. Points can be best detected by intersecting three planes each. When looking at a solid cube in general position, only three faces are visible and so only one corner can be detected with high precision. Additional planes would have to be attached to the cube to get more corners as intersection of planes (see [106], for instance, a solid cube positioned in the corner of a room). But this method is cumbersome (unless impossible) for simultaneous multi-view calibration. In order to see enough planes from several points of view we use a kind of “wire-frame model” of a cube that can be built rather easily from standard aluminum profiles (see Figure 3.6(a)). The ratio of the cube length to the strength of a frame bond should be 10:1 in order to see all six planes from almost all directions and to get enough raw data points from each plane due to laser and camera occlusions. Note that each camera sees three “outer” and three “inner” planes of the calibration object, which has to be taken into account when setting up the world coordinates of the feature points.

Care has to be taken when mounting the calibration object on the conveyer system. All cameras must see sufficient many points of all six planes. That means, that the calibration method fails in some configurations of the sensor setup, due to self occlusions of the calibration object. Secondly the piercing points of the feature points in the laser plane may not coincide and should have as few collinearities as possible. A well working setup for four cameras regarding these constraints is shown in Figure 3.6(b). One quadruple of edges of the calibration object is almost parallel to the moving axis, each tilted by approximately 3° around x - and z -axes.

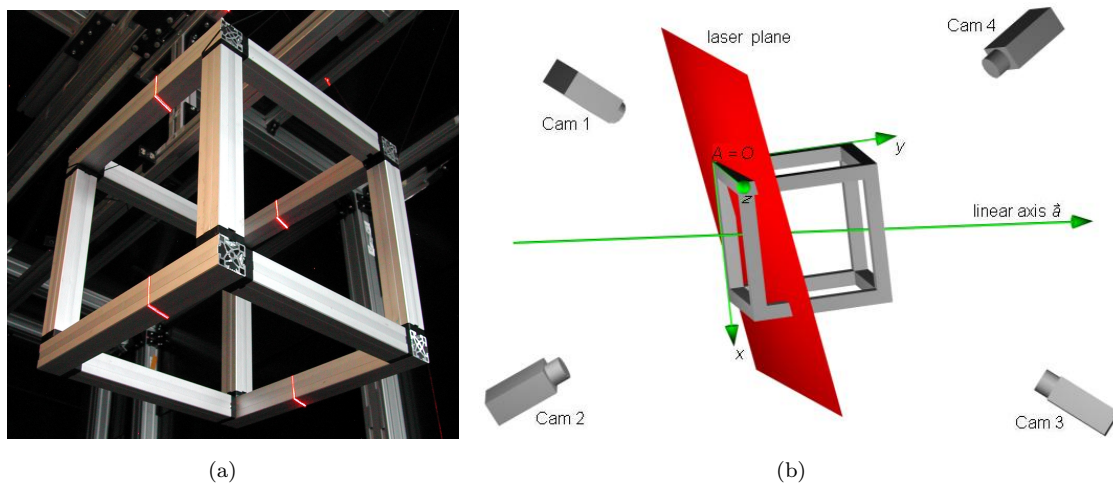


Figure 3.6: (a) Calibration object viewed from the bottom side in the darkened sensor cell. Note the projection of the laser plane on the aluminum profiles. (b) Sketch of the sensor cell consisting of four cameras and an arbitrary laser plane.

3.2.3 System Calibration Method

Our calibration method is based on establishing rather few (with our calibration object: 8) point correspondences in the raw data and the real world coordinate system, but these with

very high precision. Basically we use the points in the raw data (from the sensor) of the calibration object to extract planes with high precision, which are then intersected. Since a plane in world coordinates is not mapped to a perfect plane in the ranger coordinate system, the raw data are mapped to world coordinates first. This mapping is, of course, only an approximation (in the beginning we only use the rough estimation of the sensor cell to gain a starting value for the calibration), which still does not re-establish perfect planes, but as long as the error induced by the imperfect planes is smaller than the error in the calibration itself, successive iterations of the calibration method will converge. Our experiments show that after at most three calibration iterations the deviation of the raw data mapped to world coordinates from perfect planes is negligible.

A fast and robust method to extract a plane out of a point cloud is the RANSAC algorithm [40]. To improve the accuracy of the plane fit only candidate points are allowed whose surface normal vectors (computed by using the minor component analysis of a surface patch, see [92]) are in about the same direction as the normal vector of the hypothesis plane. After a final plane fit all points near this plane are removed from the range data. This procedure is repeated six times to extract the six most pronounced planes corresponding to the six visible planes of the calibration cube. By intersecting corresponding triples of these planes the corners of the cube can be extracted with high precision.

The actual calibration (i.e. the determination of the axis movement and the projective mapping \mathbf{M} from the sensor plane to the laser plane in world coordinates) is divided into two steps. At first the camera-independent geometry (the normal vector of the laser plane and the axis movement) is determined. This allows the computation of piercing points. The elements of the matrix \mathbf{M} are then the solution to a linear least squares problem arising from the projective relation between points in the sensor plane and corresponding piercing points.

After three iterations of separate calibrations we perform an additional iteration, this time averaging the piercing points before computing the matrices \mathbf{M} for each camera. This can degrade the overall precision of the calibrations, but improves in many cases the local matching of the point clouds.

Considering the real-world complications (unwanted laser reflections, moderate oscillations of the calibration object) the obtained accuracy was better than expected and is useful for several industrial robotic applications which deal with lot size one products. The accurate multi-view 3D model of the scanned part enables a feature extraction to apply a painting process.

Summarized evaluated results after calibrating the sensor cell with the method: In a workspace of one cubic meter the average accuracy of the acquired range images was $1.3mm$. The average accuracy on the simulated data set was $0.015mm$. The range images are calculated using the simulator described in Section 5.2.1. Figure 3.7 shows the starting position and the final result of the calibration on simulated range data sets.

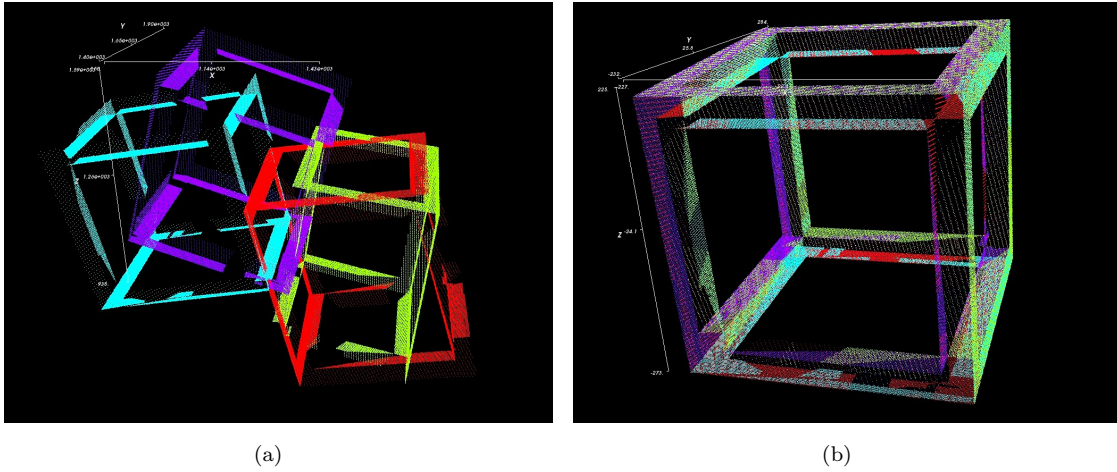


Figure 3.7: (a) Simulated range images of the non-calibrated sensor setup in Figure 3.6(b). The range images from the cameras are color coded. (b) Result of the calibration performed on the simulated data set.

3.3 FibreScope

State of the art in inspection of critical bores and internal threads are the usage of endoscope, coupled with image processing to automatically detect defects [34, 56]. Limitation is that the parts must be positioned at very high precision in 5 DoF underneath the endoscope, resulting in high costs for part handling that can only be used at very high batch sizes - incompatible to the requirements in several aerospace and pneumatic applications. Therefore this project aims to use a robotic sensor-guided system that can deal with small and medium lot sizes, being the advantage compared to manual inspection systems. State of the art in industrial robotics is program generation by teaching and OLP, using sensors to compensate the pose-uncertainties. The methods are too slow, for small lot sizes and large deviations cannot be handled, especially when requiring configuration changes of the arm. Recent research proposed a solution to these problems by automatic path planning. Examples are the projects NOMAD – presenting CAD-based welding [65], AUTOFETT – presenting a flexible off-line automatic fettling [17] and FlexPaint – presenting the paradigm of feature-based robot programming for spray painting (see Section 3.1), represent pioneering work towards this project.

3.3.1 System Overview

The system consists of several hard- and software components, which will be briefly introduced. On the hardware side there are four main components, the robot arm, the Circular Scanning Sensor (CSS), the endoscope with the CCD camera and the clearance check. The software side consists of six components, the MiniCAD module, the path planner, the CSS software, the endoscope software, the Motion Control Unit (MCU) and the system control. Figure 3.8(a) shows the prototype with the robot arm where the endoscope, the laser range sensor, the CCD-camera and the clearance check are mounted which are shown

in detail in Figure 3.8(b). Figure 3.9 gives a rough flow chart of a bore inspection sequence.

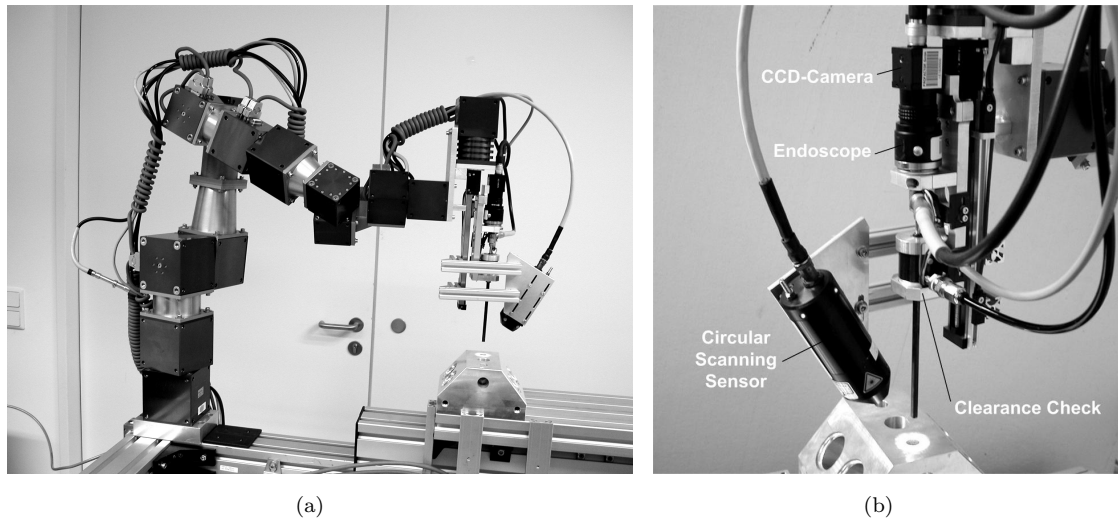


Figure 3.8: (a) The robot arm of the FibreScope prototype. (b) The sensor setup consists of a circular scanning sensor, the endoscope, the CCD-camera and the clearance check.

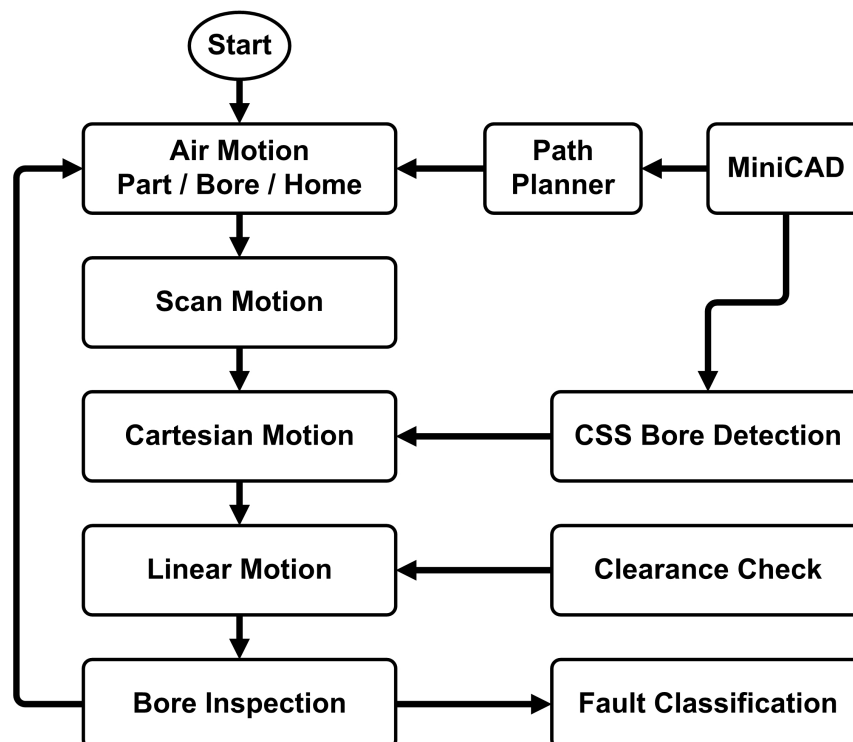


Figure 3.9: Flow Chart of the bore inspection sequence.

Hardware Components

The robot arm is build up with a modular system, that can be easily reconfigured. In this case a configuration is used for positioning a linear module with the mounted sensors over a bore hole in 6 DoF.

The CSS sensor (Figure 3.8(b)) is using the triangulation principle combined with a rotation motion to detect edges. Originally it was developed to guide automatic welding guns, but it is usable for several positioning tasks. In this special case the sensor is used to detect the position of the bore entrance, e.g., the axis of the bore related to an ideal CAD-model.

The endoscope is a standard forward-looking endoscope coupled with a CCD-Camera to perform the surface inspection (the vertical black pipe in Figure 3.8(b)). The endoscope is a precision instrument that has to be protected while inserting into the bore. A collision with the work-piece or a congested bore causes a damage of the endoscope. The clearance check detects motion resistance towards the linear motion direction and the endoscope retires immediately if a threshold of 5 Newton is exceeded.

Software Components

The MiniCAD module reads files in the STL format from commercial CAD systems or from reverse engineering. Using a GUI, the CAD data is enriched with FibreScope specific information concerning localization and inspection of bores. This information is used for the CSS bore-localization software and for the path planner which calculates a collision and singularity free motion of the robot arm. Therefore a complete CAD-model of the robot cell is integrated into the planner.

The main program activates the application CSS software of the laser range sensor. After each scan it provides the new bore target position and orientation in world coordinates related to the robot. The endoscope software is an image processing application, which detects surface faults out from gray scale images. The inspection is performed when the endoscope moves out of the bore.

The bore inspection cycle is called within a main application control loop. Modules are activated and deactivated by the specified CORBA interfaces and commands. The application control implementation allows adaptation to new robots, parts, and environment, thus the application does not need to be reprogrammed for different bore-inspection applications.

Inspection Cycle

During the bore inspection process several different motions are performed. The air and scan motion is processed by the path planner calculating the joint values for all positions of the arm on its way to the bores. The planner also checks the scan motion to prevent any kind of collision. The scanning motion is a simple rotation of the last joint with its angle depending on the bore diameter.

The cartesian motion is performed after detection of the position deviation. The CSS transfers the new bore pose to the system control. The new pose is used in a cartesian

motion to eliminate the offset and to align the endoscope with the bore. Finally the simplest motion is the linear motion, which imposes the endoscope into the bore and back. During this process all joints of the arm are on gridlock.

After introducing the different motions a bore detection sequence is introduced by carrying out the following steps (compare Figure 3.9). At the beginning the robot arm is moved to the part respectively the selected bore. The last joint is now located above the bore and the scanning can be performed by pivoting the rotating laser beam over the bore entrance. The CSS detects the bore center entrance in 5 DoF. Then the deviation of the bore pose is compensated. The linear module with the endoscope is deployed into the bore according to the parameter “bore depth” out from the MiniCAD module. If the clearance check detects a collision the linear motion is stopped and the bore is classified as not detectable. On the bottom of the bore the protection sleeve of the clearance check is pulled back and the inspection is done on the way out of the bore to make sure that there is no obstacle that stops the image grabbing. Afterwards the robot arm is moved to the next bore to start a new sequence or recapture the home position.

Rapid Application Programming

For programming, the system has to be adjusted only at the off-line side. An exact model of the robot cell needs to be built up inside the path planner. It has to be as similar as possible to the real robot and need to be made only once. The next step is to do the CAD-enrichment in MiniCAD for a certain part to get two text files with bore parameters. Afterwards the path planning can be executed. The result-file includes an executable joint motion path for the robot extended with all relevant motion parameters. The second file (bore entrance environment) is not manipulated and will be directly used by the localization software.

3.3.2 Robust Bore Localization

Due to imprecise positioning of the work piece on the feed system and unexact robot movement while positioning the endoscope axially parallel to the bore, the endoscope might fail to insert into the bore. Because of the high risk of damaging the endoscope, robust bore localization must be previously passed. This procedure enables the robot controller to compensate the deviation in 5 DoF. The bore localization is done by a contact free machine-vision-based method, namely a laser range sensing measurement. The Circular Scanning Sensor, CSS, is a multi-axis machine vision sensor, which produces a highly accurate 3D model of the work piece (resolution 0.2mm). The sensor is pivoting over the expected bore position enforced by a rotation of the last robot arm axis. To acquire a range image the helical movement of the laser beam is transformed to equidistant points in world coordinates. For a better treatment of the sensor setup see Section 4.2.1.

The accurate localization of the bore is based on detecting the bore entrance followed by fitting a cylinder model to the 3D range data. The algorithm is designed for real time processing. To achieve this goal, the method uses a limited search area given by approximated bore position of the MiniCAD module. An exact description of the range image segmentation and the novel fast cylinder fit procedure is given in [10] and in Chapter 4.

3.3.3 Bore Inspection

The task of the bore inspection is to detect shrink-holes, chips and scratches by image processing methods. The image sequence is taken by retracting the endoscope out of the bore. Each image of the sequence is rolled off and one pixel row, which is equivalent to a circle at the perimeter, in the focused field of view, is extracted. To acquire one image of the whole bore surface all extracted pixel rows are added, triggered by the feed speed of the endoscope.

After acquiring the gray scale image the image processing can be done. The first step of finding the faults is to search for areas where the gray scale value changes significantly and these areas must hold a minimum size. Methods of finding such areas are shown in [5]. The second step is to classify these fault areas [100]. Scratches are from longish shape while shrink-holes and chips are more circular shaped. If faults are detected in the surface these bores are marked as failed and the type of fault and their depth is denoted.

Chapter 4

Bore Hole Detection

The bore hole detection process is embedded in the FibreScope prototype framework (compare Section 3.3) where CAD data from the scanned workpiece is available which indicates the approximate pose of the bore hole to be inspected. This information is used to scan around the expected bore location and to process only range data in a circle around this location (compare Fig. 4.1(c)), which helps to speed up the segmentation process. As a second constraint the surrounding surface of the bore may deviate from a perfect plane and exhibit a bumpy surface typical of cast parts or a slight curvature compared to the bore diameter. The method is a special solution for this industrial inspection application. For that reason the method must work in process real time, hence, all processing steps must be optimized for fast processing. The focus of the method is not the segmentation process rather than the contribution of a novel sequential cylinder fit. Nevertheless, this chapter presents an efficient approach for bore hole detection dealing with noisy and sparse range data, whereas the approach can be adopted for generality, recovering cylindrical primitives in range data.

Inspired through the work of Robertson et al. [110], demonstrating a method for hole detection in noisy and fragmented range data by exploiting additionally obtainable knowledge about the part, the first approach to detect the bore hole within the FibreScope project was to find the bore hole boundary by fitting a circle. The problem to deal with are the rounded and chamfered edges at the bore hole entry, which makes a robust circle fit difficult. Figure 4.1 sketches the idea of the first approach visualizing the main detection steps. Starting from an industrial part the neighborhood of the bore is scanned (Figure 4.1(a) and 4.1(b)). Then the approximated bore pose from the CAD data is used to extract annulus points (Figure 4.1(c)) to fit a plane and then, all associated points within a certain tolerance are mapped to this plane (Figure 4.1(d)). Now, the task of resolution is to extract the boundary points of the projected bore hole representing the inner diameter of the bore. This is done by generating a mesh (Figure 4.1(e)) on the sub-sampled plane points using the Delaunay triangulation [31, 93] and afterwards calculating the mesh boundaries (Figure 4.1(f)). Extracting the vertices of the inner boundary and applying a least-squares circle fit determines the position of the bore in 3D space (Figure 4.1(g) and Figure 4.1(h)). The orientation of the bore is given by the normal vector of the plane.

This first approach seems to work quite well on good range data and perpendicularly

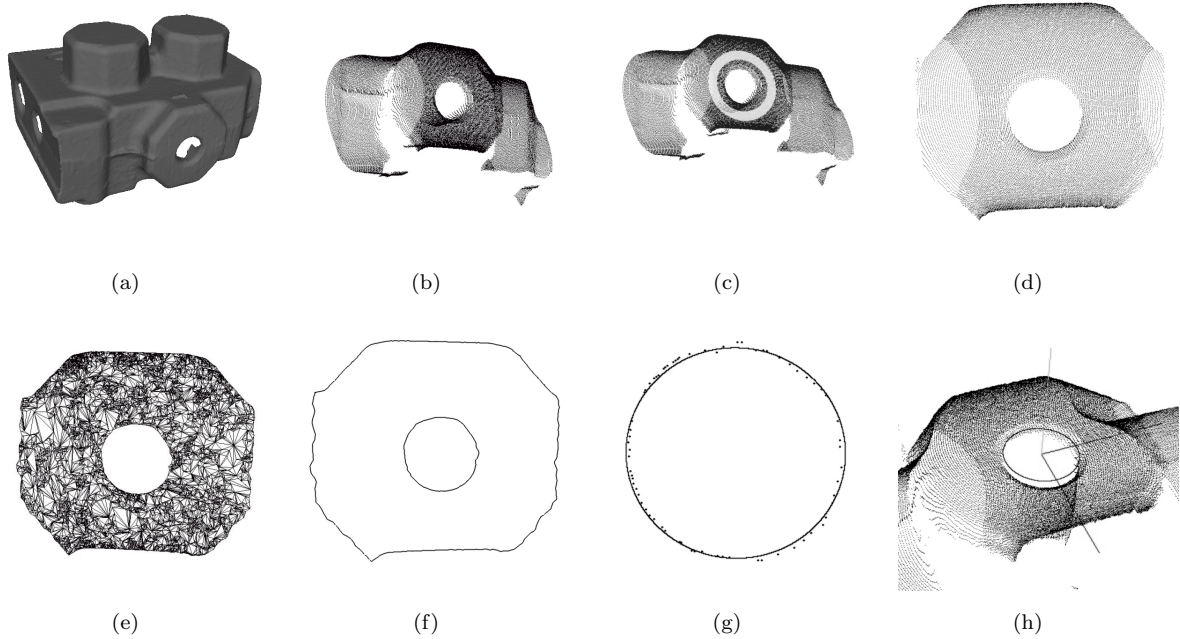


Figure 4.1: Sketch of the processing steps of the first bore hole detection approach by fitting a circle to the bore boundary.

drilled bore holes related to the bore entrance plane. Nevertheless, the method has some major drawbacks. The main disadvantage is that the method can not cover the detection of bores with an arbitrary orientation. Furthermore, range artifacts due to specular laser reflections, noise and incomplete range data of the bore hole boundary, fails the detection dramatically. The lack of robustness in the 3D detection results can be traced back because of reducing the problem to a two-dimensional one. So the solution of the bore detection seems to be a geometric fitting problem in 3D space. In Figure 4.1(h) the fragment of the inner bore surface can be seen, which gives an idea of the challenging task to fit a cylinder to the sparse range data.

The method described in the next section introduces a fast and robust approach to detect any arbitrarily orientated bore hole for a sub-sequent inspection task. Section 4.2 evaluates the proposed method and demonstrates the efficiency and robustness compared to standard fitting techniques.

4.1 Method Description

This section presents the proposed method in detail starting with the flow chart in Figure 4.2 outlining the overall approach. It shows the main processing steps from a CAD-estimated 3D pose to the measured, actual 3D bore pose. As second input the range image is required containing the scanned bore hole. Detailed information about the range image acquisition is given in Section 4.2.1. The method starts with a preprocessing step followed by the normal vector calculation in Section 4.1.1. The normal vectors are the basis for

the segmentation process (Section 4.1.2) extracting the bore entrance plane and the bore surface segment, as well as the normal vectors are needed for the sequential cylinder fit in Section 4.1.3. The last section describing this method deals with the problem of determining the pose of the bore in the presence of radial notches in the inner bore surface.

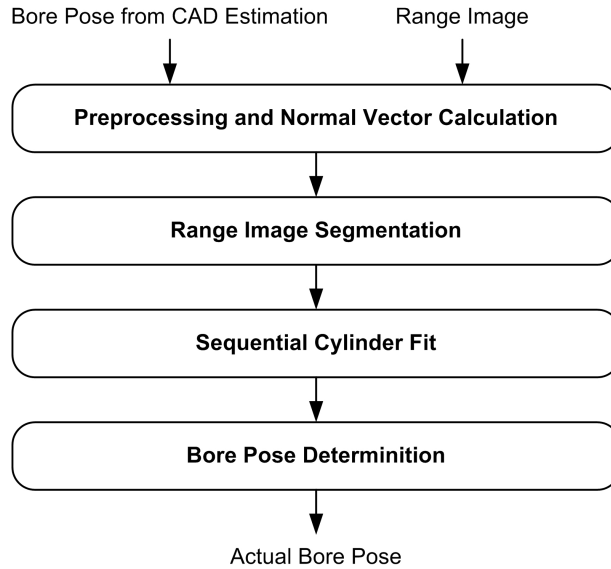


Figure 4.2: Flow chart of the bore hole detection approach.

4.1.1 Preprocessing and Normal Vector Calculation

First the surface normal vectors are calculated, which are used in the second step to segment the bore hole as well as in the third step for the sequential cylinder fit. The results below will show that the method used for calculating the normal vectors is robust against noise and outliers in the range image. This is important because a good segmentation is the basis for the sub-segmentation accuracy of the cylinder fit.

The requirement for the detection method was to process in real time. Because of the dense raw data points, provided by the scanner, the amount of data has to be reduced to achieve the goal of fast processing. The first preprocessing step is to extract the region of interest in the range image where the bore is probably located. Figure 4.3 indicates the region of interest given by the estimated CAD pose of the bore (doubled radius of the bore hole). As a second procedure to speed up the method the selected range image segment is again sub-sampled by reducing the amount of raw data points equally distributed by 80%.

The normal vector of a given point is approximated by a planar surface patch close to the observed point, instead of a local quadric fitting which does not improve the result. This is a robust method because all points within a radius R (in our case 2mm) give their vote to the plane fit which compensates the outliers (compare Fig. 4.6). To calculate

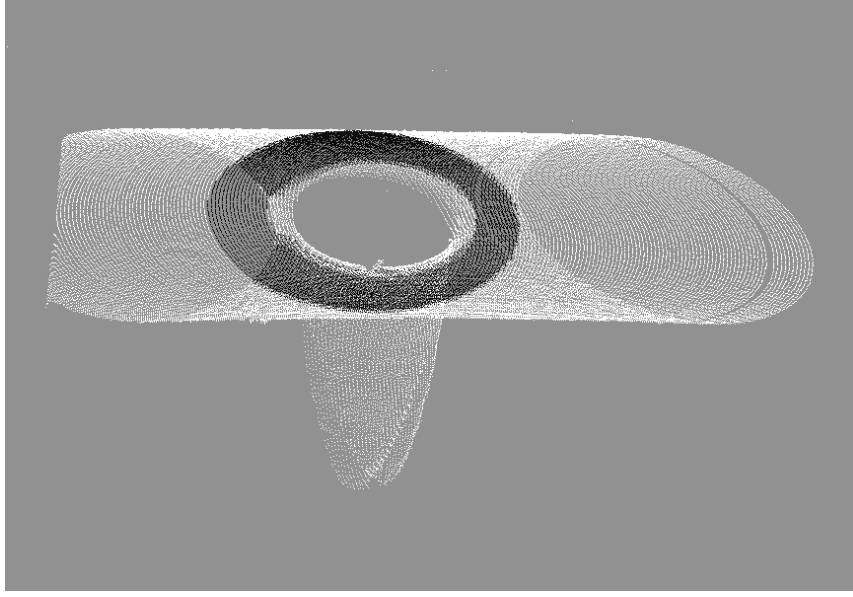


Figure 4.3: Scanned range image with the indicated region of interest.

the surface patch plane principal component analysis (PCA) [92] is applied to the set of surrounding points. Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ be a set of points which represent the patch, where $\mathbf{x}_i^\top = (x_i, y_i, z_i)$ is a 3D point of this set. Further, the mean of the point set is defined by

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}, \quad \bar{y} = \frac{\sum_{i=1}^n y_i}{n}, \quad \bar{z} = \frac{\sum_{i=1}^n z_i}{n} \quad (4.1)$$

and the covariance of a coordinate pair is given by

$$\text{cov}(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y}) \quad (4.2)$$

This leads us to the covariance matrix

$$C = \begin{bmatrix} \text{cov}(x, x) & \text{cov}(x, y) & \text{cov}(x, z) \\ \text{cov}(y, x) & \text{cov}(y, y) & \text{cov}(y, z) \\ \text{cov}(z, x) & \text{cov}(z, y) & \text{cov}(z, z) \end{bmatrix} \quad (4.3)$$

which is the initial for the PCA. The normal vector of the surface patch corresponds to the vector \mathbf{n} which is determined by the eigenvalue problem $C\mathbf{n} = \lambda_{min}\mathbf{n}$, where λ_{min} is the smallest eigenvalue and C is the covariance matrix of the point set (4.3).

Using the eigenvectors to determine the normal vector can result in two possible opposite directions. This is not important for the segmentation as in contrast for the cylinder fit where a wrong orientation yields a wrong fit. The correct orientation of the normal vector is found using the sensor coordinate frame, where the z -axis points into the sensor aligned with the optical sensor axis. The correct normal vector is the normal vector with the smaller angle to the z -axis (compare Figure 4.14).

Fig. 4.4 shows 5% of the calculated normal vectors around the bore hole. This normal vector calculation is an estimation and yields a deviation to the correct vector on sharp corners or boundaries of a curved surface of the range data.



Figure 4.4: Close up of the calculated normal vectors in the region of interest.

Normal vectors with a deviation caused by sharp edges are filtered in the segmentation process. Figure 4.5 demonstrates the deviations in the case of a roof edge. In this example each normal vector is calculated using six neighboring raw data points. The dashed normal vectors indicate the correct vectors. In contrast to that, normal vector deviations caused by non planar surface boundaries (see Figure 4.6) are eliminated by a post processing step that is described next.

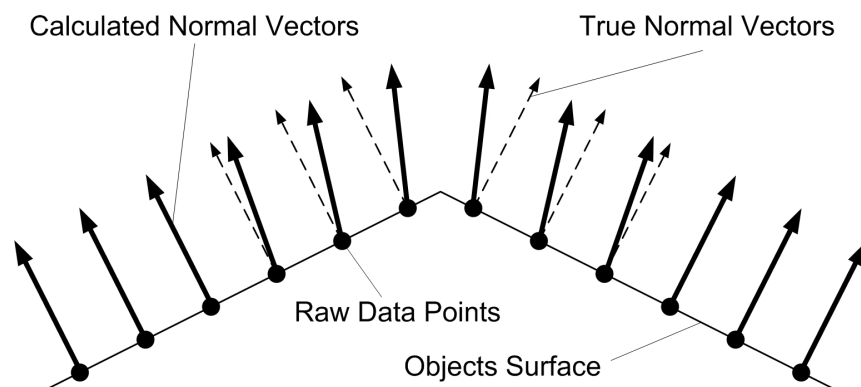


Figure 4.5: Sharp corners lead to a systematic calculation error causing unwanted normal vector deviations.

A criteria for a sufficiently accurate normal vector is the center of gravity of the point

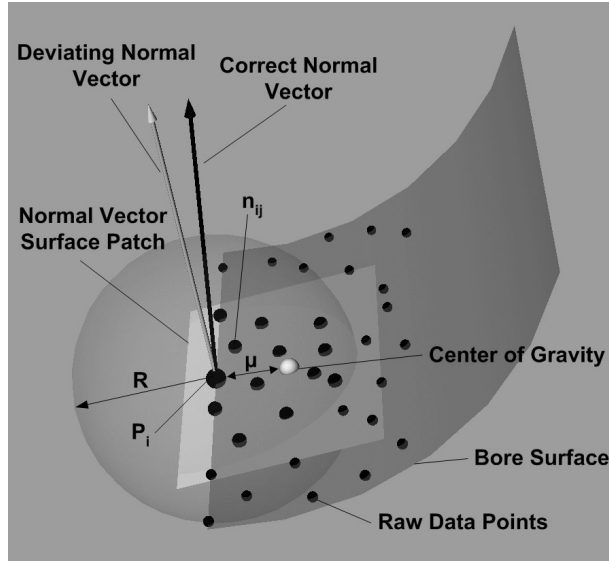


Figure 4.6: Illustration of the normal vector calculation in the case of non planar boundary area, causing normal vector deviations.

set X_i which is used to calculate the normal vector \mathbf{n}_i . If the difference μ

$$\mu = \left| \mathbf{p}_i - \left(\frac{1}{n} \sum_{j=1}^n \mathbf{x}_{i_j} \right) \right| \quad (4.4)$$

is less than 10% of the radius R that selects the points for the normal vector calculation, this normal vector is retained. Note that normal vectors in the boundary area will not be used for axis determination due to this criteria (see Figure 4.9).

4.1.2 Range Image Segmentation

Remember that in our case the bore hole localization needs to cope with sparse data, bumpy and curved top surfaces, rounded edges and chamfers, slanted bore holes, and bore holes with notches (indicated in Section 1.1.1). The Gaussian image [32] is an appropriate way to deal with these challenges and is a common method to estimate the orientation of geometric primitives such as planes, cylinders or cones [120]. Here, all raw data points are mapped onto the unit sphere according to the orientation of their normal vectors. In the Gaussian image, cylinders are represented as great circles and planes as points. In Fig. 4.7(a) the Gaussian image of the range image in Fig. 4.4 is shown.

Some clustering methods have been proposed in the past, where in [26] the segmentation is based on finding accumulated great circle points in the Gaussian image which have a minimum distance to a plane through it. This approach can be used if cylinders only can be expected. In our case a major plane and a cylinder fragment are expected in the range image respectively in the Gaussian image. If the segmentation has to deal with more than one geometric primitive a more complex segmentation algorithm is required. The

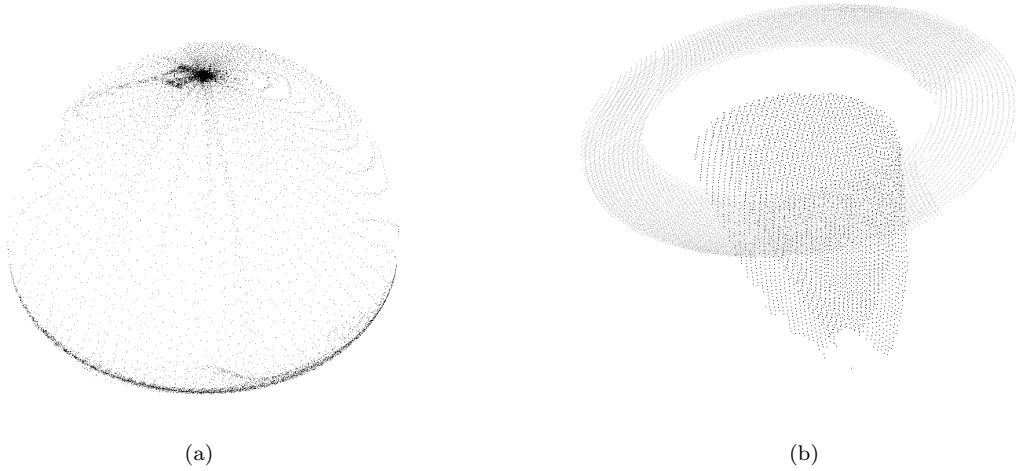


Figure 4.7: Illustrating the segmentation process. (a) Gaussian image representing a noisy cylinder surface and a plane. (b) Result of the segmentation process separating the cylinder surface fragment and the plane.

problem is that the normal vectors of parallel shifted or concentric objects have the same mapping to the Gaussian image and therefore, cannot be separated. A possible method of resolution is to extend the mapping of the Gaussian image to the \mathbb{R}^4 space – the so-called Blaschke model [16]. This is a combination of the classical Hough transformation [54] with the Laguerre geometry [102]. Applying the Principal Component Analysis (PCA) [92] to the Blaschke mapping the range image can be uniquely segmented into geometric primitives, such as planes, spheres, rotational cones and cylinders, general cones and cylinders, and general developable surfaces [96, 97].

Figure 4.3 illustrates that the range image consists of one major plane and one cylinder fragment whose data points can be clustered in the Gaussian image (Figure 4.7(a)). Due to computational efficiency the orientation of the bore hole and the plane can be exploited from the available CAD data, in order to avoid a complex algorithm using the Blaschke cylinder. By clustering the points on the unit sphere and using the expected inaccuracy of up to 5° from the reference orientation, the cylinder surface and the plane can be separated. This works as well for expected slanted bore holes, where the great circle is also slanted in the Gaussian image. All other points are ignored. These constraints simplify the segmentation where as first step the major plane in the range image is detected calculating the angle α of all normal vectors,

$$\alpha = \arccos \left(\frac{\mathbf{n}_i \cdot \mathbf{n}_{CAD}}{|\mathbf{n}_i| \cdot |\mathbf{n}_{CAD}|} \right), \quad (4.5)$$

where \mathbf{n}_i is a raw data normal vector and \mathbf{n}_{CAD} is the CAD normal vector of the plane where the bore is located. If $\alpha < 5^\circ$ the associated raw data point is classified as plane point. As second step, points belonging to the bore surface are clustered in the Gaussian

image by calculating the angle β for all remaining normal vectors

$$\beta = \arccos\left(\frac{\mathbf{n}_i \cdot \mathbf{a}_{CAD}}{|\mathbf{n}_i| \cdot |\mathbf{a}_{CAD}|}\right), \quad (4.6)$$

where \mathbf{n}_i is again a raw data normal vector and \mathbf{a}_{CAD} is the CAD bore axis vector. If $|90^\circ - \beta| < 5^\circ$ the associated raw data point is classified as bore surface point.

Using the expectation from CAD data helps to render segmentation very robust. An example of the result of the segmentation process is shown in Fig. 4.7(b). Note that only points which belong uniquely either to the cylinder surface or to the plane are clustered. It can also be seen that the curved bore hole rim is missing due to eliminating these normal vector directions after segmentation.

4.1.3 Sequential Cylinder Fit

The next step after the segmentation is the fit of a cylinder model to the cylinder surface points. Usually a fit of geometric objects to range data is done iteratively using global or local optimization methods, e.g., the Levenberg-Marquardt method (see Section 2.2.2). The problem with this approach is that all parameters are estimated at once, which requires a good initial value and which is time consuming because of iteratively comparing the whole point cloud with the fit criteria. Also, having in maximum of 120° of the cylinder surface and a short cylinder length makes it difficult to robustly apply standard fit techniques.

The approach proposed is a sequential cylinder parameter fitting starting with the orientation of the axis followed by the radius and finally the position of the axis. The rationale is that segmentation has properly clustered the bore hole and step wise fitting perfectly copes with the remaining noise.

Due to the fact that the normal vectors have been previously calculated they should be used for the cylinder fit as for reasons of computation efficiency. The task is now to fit a cylinder based on candidate points which are filtered through the normal vector computation and the segmentation process. Note that the depth of the bore hole cannot be estimated because of the missing raw data information deeper in the bore hole. The information of the depth is taken from the CAD data. But note, that this information is only used for the inspection task and is not necessary for the cylinder fit method.

Orientation Calculation

Cylinder fitting can start with estimating the orientation of the cylinder, because it is independent of the radius and can be calculated directly from the normal vectors. The cylinder orientation is given with an uncertainty of 5° from the Gaussian image.

To compensate the noise uncertainties we calculate the axis orientation \mathbf{u} with the average of 1,000 randomly-picked normal vector pairs complying to the constraint (4.9) where all normal vectors must point into the same Gaussian hemisphere

$$\mathbf{u} = \frac{1}{k} \sum_{k=1}^{1000} \mathbf{o}_k. \quad (4.7)$$

Trials with up to 100,000 samples show that more samples do not significantly improve the accuracy but waste useful processing time. See Section 4.2.2 for the experimental evaluation of this assumption.

The orientation \mathbf{o} of one sample in Equation (4.7) of the cylinder axis is determined by the cross product $\mathbf{o} = \mathbf{n}_i \times \mathbf{n}_j$ of the two normal vectors from the points \mathbf{p}_i and \mathbf{p}_j . Due to the noise level the orientation calculation might fail if the two normal vectors are too close to each other. Using the angle γ

$$\gamma = \arccos \left(\frac{\mathbf{n}_i \cdot \mathbf{n}_j}{|\mathbf{n}_i| \cdot |\mathbf{n}_j|} \right) \quad (4.8)$$

between two normal vectors \mathbf{n}_j and \mathbf{n}_i we have a stability criteria for the orientation estimation when constraining α to

$$15^\circ < \alpha < 75^\circ. \quad (4.9)$$

Radius Calculation

After estimating the axis orientation the radius can be calculated. One possibility is to map any two normal vectors to a plane perpendicular to the axis \mathbf{u} and calculate the intersection point \mathbf{m} of the two lines built by $l_i : f(\mathbf{p}_i, \mathbf{n}_i)$ and $l_j : f(\mathbf{p}_j, \mathbf{n}_j)$. Having the intersection point \mathbf{m} the radius \tilde{r} can be calculated with

$$\tilde{r} = \frac{|\mathbf{m} - \mathbf{p}_i| + |\mathbf{m} - \mathbf{p}_j|}{2}. \quad (4.10)$$

This method is not robust enough, even by calculating the average with a high number of samples, because small uncertainties of the normal vectors and acute intersections lead to deviations of up to 10 degrees. Also using least-squares circle fitting techniques applied to the perpendicular to \mathbf{u} mapped bore points, fail because of the circle fragment and due to the noise level in the range image, the fit does not meet the accuracy requirements (see Section 4.2.2).

A robust method of estimating the radius can be achieved by circumscribing a circle to a triangle: First three randomly selected points build a triangle which is again mapped to a plane where \mathbf{u} is its plane normal vector. Hence, the radius of the circumscribed circle is the radius of the cylinder. Figure 4.8 sketches the idea of the radius calculation.

The radius of one triangle is calculated by

$$\tilde{r} = \frac{a \cdot b \cdot c}{4 \cdot A}, \quad (4.11)$$

where a , b and c are the side lengths of the triangle and A is Heron's formula for the triangle area where s is half the triangle side lengths, that is,

$$s = \frac{a + b + c}{2}, \quad (4.12)$$

$$A = \sqrt{s \cdot (s - a) \cdot (s - b) \cdot (s - c)}. \quad (4.13)$$

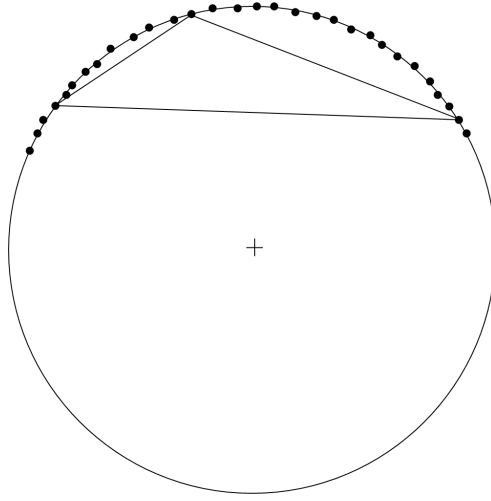


Figure 4.8: Idea of the radius calculation by circumscribing a triangle built of three randomly picked points of the segmented bore surface.

To eliminate the effect of the noise while calculating \tilde{r} , A must have a minimum size exploiting the expected radius \hat{r} from the CAD data.

$$A > \frac{\hat{r}^2 \pi}{10} \quad (4.14)$$

The constraint (4.14) has been evaluated experimentally and leads to a radius estimation with less than 1% error compared to the radius \hat{r} .

To make the estimation of the radius r again robust the average of 1000 radius calculations is computed with the same arguments as for Equation (4.7).

$$r = \frac{1}{k} \sum_{k=1}^{1000} \tilde{r}_k \quad (4.15)$$

Axis Pose Estimation

As last step the 3D pose of the cylinder axis must be determined. In [139] the pose of the axis is determined using the 3D Hough transformation [54] which is very time consuming and usually used to search for more than one line in 3D space. This problem can be solved using the cylinder surface normal vectors \mathbf{n}_i and the radius r . With the knowledge that all normal vectors cross the cylinder axis, this axis can be calculated. Points on the cylinder axis can be found by multiplying the cylinder surface normal vectors with the previously calculated radius r .

Figure 4.9 displays 5% (for reasons of clarity) of all segmented normal vectors. Also the axis fit is shown. The axis fit is applied again using the PCA method where $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ is now the set of axis points. Using Equation (4.2) and (4.3) the covariance matrix C of the axis point set can be calculated. The axis of the bore hole corresponds to

the vector \mathbf{v} which is determined by the eigenvalue problem $C\mathbf{v} = \lambda_{max}\mathbf{v}$, where λ_{max} is the largest eigenvalue of the point set. The bore hole axis is defined by the vector \mathbf{v} and a point \mathbf{p}_v which is the center of gravity of the point set X calculated with Equation (4.1).

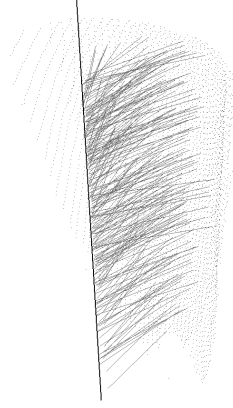


Figure 4.9: Result of the axis fit. Note the visualized normal vectors whose ends determine the bore hole axis.

4.1.4 Bore Pose Determination

The orientation of the bore axis is not the final bore pose. The robot needs as well the bore entrance position. This is the intersection point of the bore axis with the plane fit of the point set which is the result of the plane segmentation process.

The final result of the bore hole detection is shown in Figure 4.10. A model of the estimated bore hole is visualized and the axis starts at the bore entrance position pointing outwards.

The bore holes of the FibreScope's industrial partners' workpieces have many different shapes and surface qualities. Some bore holes, mainly from cast parts, have radial or axial notches. See Figure 4.11 where a cross section of a cast part is shown to see how radial notches can look like. These notches are also scanned when they are located near the bore entrance. Hence, the segmentation algorithm must distinguish between cylinder surfaces with different radii. Otherwise a notch let the cylinder fit fail: the orientation of the resulting cylinder axis is slightly slanted and the radius is larger than expected.

Because of possible notches in the bore hole the fit result is examined. Starting from the cylinder axis determined from each cylinder surface point \mathbf{p}_i the perpendicular distance d to the axis is calculated.

$$d = \left| \frac{\mathbf{v}}{|\mathbf{v}|} \times (\mathbf{p}_v - \mathbf{p}_i) \right| \quad (4.16)$$

And if the distance is larger than $1.03 \cdot r$ (from Equation (4.15)) these points are removed and a second cylinder fit is applied with the remaining points. The threshold of 1.03 is experimentally evaluated and depends on the notch size and shape. If no points exceed the threshold no second fit is required.

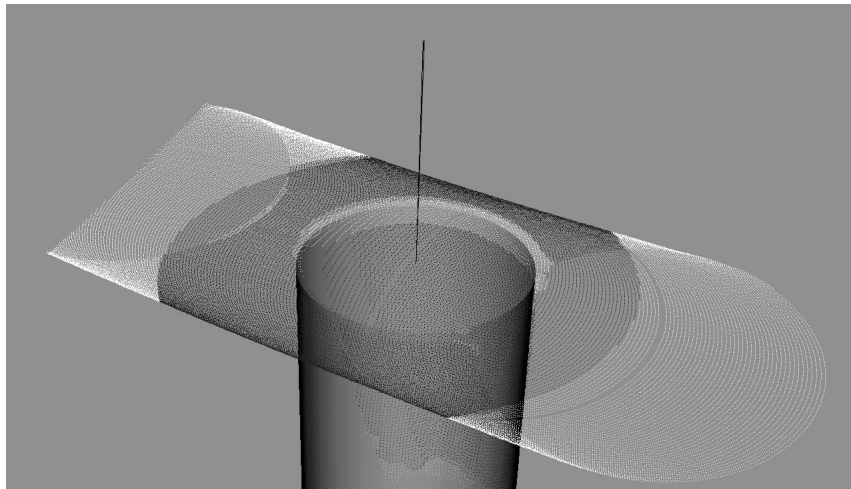


Figure 4.10: Final result of the bore hole detection process.

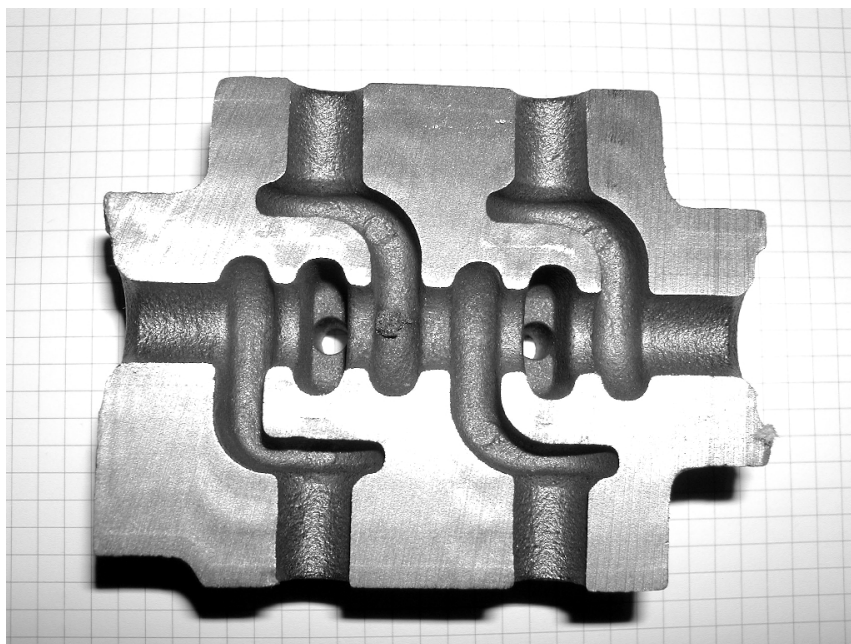


Figure 4.11: Cross section of a cast part where radial notches are visible which have an effect to the cylinder fit result.

4.2 Experimental Evaluation

The experiments are carried out acquiring range images using the FibreScope prototype to evaluate the behavior and the robustness of the detection method on real settings. In order to conduct experiments without using the prototype hardware or to test other parts a sensor simulator has been used. This is important, because the prototype hardware

was not available all the time and to test CAD-based parts a virtual environment of the sensor hardware is needed. This section starts with the introduction of the sensor setups describing the circular scanning sensor and the simulator in detail. Then the performance of the algorithm is tested as well as the bore detection method is evaluated and compared on real and synthetic range images.

4.2.1 Sensor Setup

In the project FibreScope the circular scanning sensor from OST has been used to compensate the positioning uncertainties while bore inspection. Section 4.2.1 describes this sensor in detail as well as the range image acquisition. In the course of the FibreScope development a simulation has also been realized to compute off-line tests and to be able to capture range images of not present parts (see Section 4.2.1). Details on these developments and settings for the experiments are given in the following sections.

Circular Scanning Sensor

The sensor described in this section was used in the project FibreScope (see Section 3.3). To detect the bore pose with the required accuracy a high resolution sensor is needed. Furthermore, the sensor must be compact in shape and must be lightweight because it is mounted at the end of the robot arm to cover concave work pieces with bore holes on all sides of the object (compare Figure 3.8).

The circular scanning sensor, CSS¹, which is shown in Figure 4.12, is a multi-axis machine vision sensor, that produces a 3D point cloud of the work piece's surface. The resolution of the sensor is 512 points per laser beam rotation at an average scan diameter of $27mm$. The laser range sensor has a beam rotation frequency of $7Hz$ with a lateral resolution of $165\mu m$ and $60\mu m$ in depth. Figure 4.13(a) illustrates the scanning distance of the laser range sensor over a bore with a diameter of $12mm$. Note the retired endoscope covered with the clearance check. The resulting range image of the scan is shown in Figure 4.13(b) consisting of more than 75,000 data points.



Figure 4.12: Circular scanning sensor from Oxford Sensor Technology.

¹For more information see <http://www.oxfordsensor.com>.

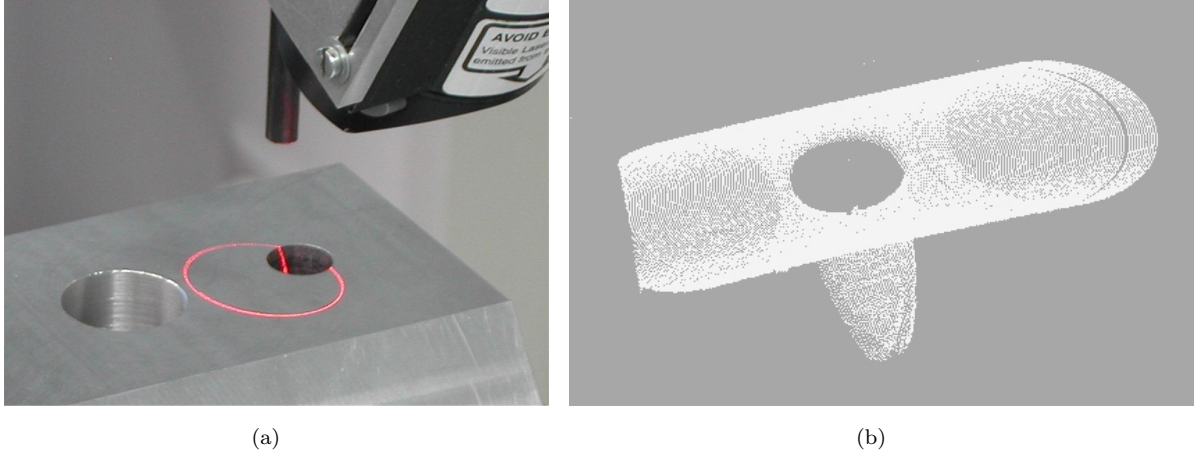


Figure 4.13: Close up of the scanning process (a) and the acquired range image (b). Note the laser beam rotation.

To acquire a range image the endoscope is axially positioned over the bore hole. Then, all robot joints are on gridlock except the last one, which is pivoting the sensor over the expected bore position enforced by a rotation of the last robot arm joint ($\pm\delta$). Figure 4.14 shows the important dimensions of the sensor setup and defines the world and sensor coordinate system.

The sensor coordinate frame is attached to the sensor body where the z -axis is aligned with the sensor axis. The sensor is calibrated factory-made with a nominal standoff r of 88.5mm . (For details on calibration of the robot arm related to the laser sensor see [12].) While scanning, the sensor coordinate frame is moving in 3D space on a circular trajectory described with the path angle ω . To obtain a range image, as it is shown in Figure 4.13(b), a transformation to a world coordinate system has to be applied in order to be able to apply a further transformation to the robot coordinate system (not described here). The transformation from a raw data point \mathbf{x}_s in the sensor coordinate frame to a point \mathbf{x}_w in the world coordinate frame is given by

$$\mathbf{x}_w = \mathbf{R}_y \cdot (\mathbf{R}_x \cdot \mathbf{x}_s + \mathbf{t}) \quad (4.17)$$

where \mathbf{R}_x is the rotation around the x_s -axis by the angle $\eta = 28.8^\circ$

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\eta) & -\sin(\eta) \\ 0 & \sin(\eta) & \cos(\eta) \end{bmatrix}, \quad (4.18)$$

\mathbf{R}_y is the rotation around the y_w -axis describing the scan path with the angle ω

$$\mathbf{R}_y = \begin{bmatrix} \cos(\omega) & 0 & \sin(\omega) \\ 0 & 1 & 0 \\ -\sin(\omega) & 0 & \cos(\omega) \end{bmatrix} \quad (4.19)$$

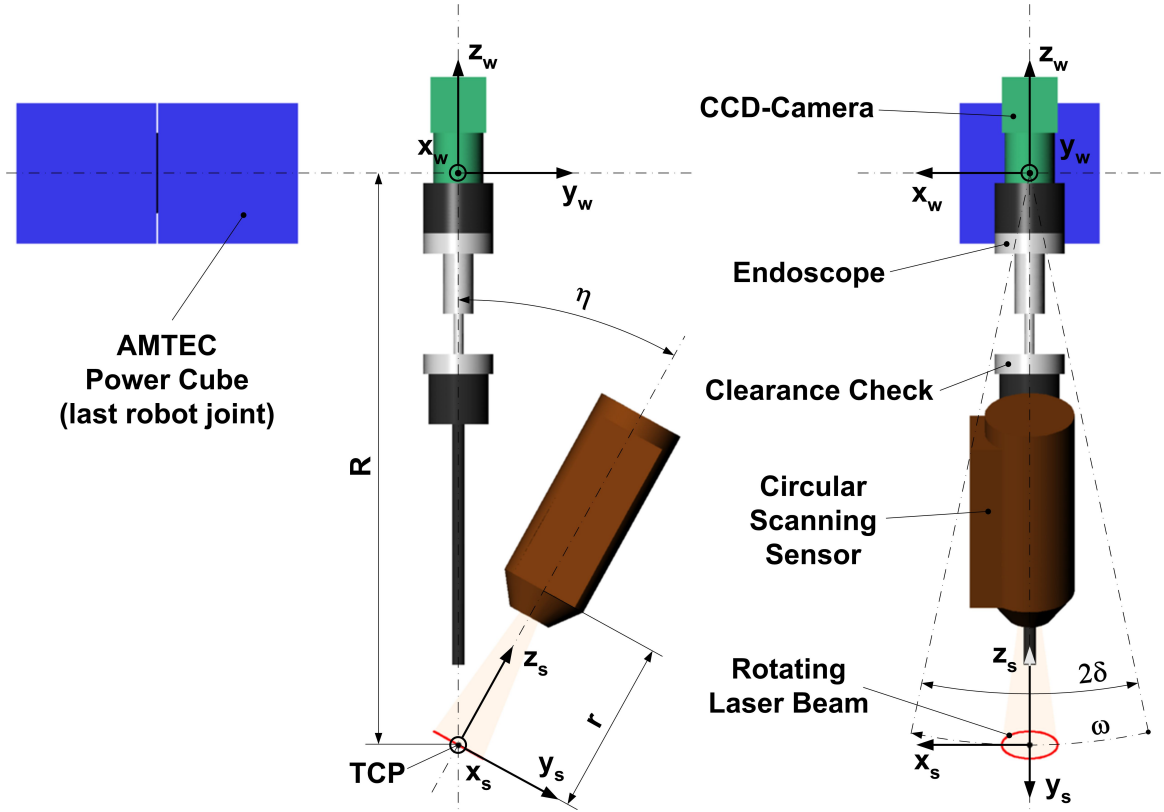


Figure 4.14: Drawing of the sensor setup showing the dimensions and the world and sensor coordinate system.

and t is the translation vector from the sensor in the world coordinate frame with $R = 285.8mm$.

$$\mathbf{t} = \begin{bmatrix} 0 \\ 0 \\ R \end{bmatrix} \quad (4.20)$$

The angle of the scan deflection δ is set to 5° which covers bore diameters from 4 to 50mm. In order to accurately transform the helical movement of the laser beam while scanning, the angle ω has to be determined precisely. This is done by using time stamps with a resolution of one μs at the beginning t_{start} ($\omega = 0$) and the end of the scan trajectory t_{end} ($\omega = 2\delta$). Additional time stamps, taken after each scan rotation, provide an accurate calculation of ω corresponding to each raw data point on the scanning path. The speed vector \mathbf{v} of the tool center point (TCP) is aligned with the x -axis of the local sensor coordinate frame and its speed $v = |\mathbf{v}|$ is given by

$$v = \frac{d\omega}{dt} \cdot R = \frac{\delta \cdot \pi \cdot R}{90 \cdot \Delta t}, \quad \text{with } \Delta t = t_{end} - t_{start}. \quad (4.21)$$

Acquiring range images with a scan speed $v = 2mm/s$ and $\delta = 5^\circ$ take about 25 seconds with a resolution between each scan circle in scan direction of $285\mu m$. This resolution is

sufficient for a reliably detection of the bore hole with a pose accuracy of $\pm 0.3mm$ in position and less than 0.5° in orientation.

Synthetic Range Image Generator

This section introduces the virtual scanner of the FibreScope prototype. The demand of the synthetic range image generator arises, because the prototype was not available during the test periods and the testing of parts, which are only available in CAD models, was necessary to evaluate the bore detection method.

The task was not to simulate the behaviour of the circular scanning sensor in detail rather than to compute a qualitative range image concerning the field of view of the camera and laser shadows. The simulator does not take the quantization noise, the brightness of the laser light and the object surface properties into account. In order to model the circular scanning principle one must understand the operating of the scanner.

Figure 4.15 gives an idea of the sensor's functionality. The main parts of the CSS are the laser diode, the CCD line camera and the rotating off-axis lens. This asymmetric lens deflects the laser beam to a circular motion and on the other hand the lens deflects the reflected laser light in parallel rays onto the CCD line, where the focus is set to infinity.

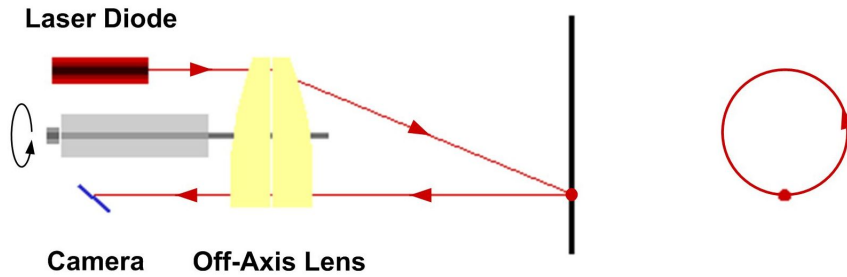


Figure 4.15: Working principle of the circular scanning sensor. Note the main part of the sensor, the rotating asymmetric lens.

The sensor is modeled as simple cylinder with its local sensor coordinate system located on the cylinder axis at a distance r to the sensor body, where the z -axis is orientated towards the sensor and the x -axis gives the scan direction. Note that r is the nominal standoff distance of the sensor ($88.5mm$) as described in Figure 4.14. The working standoff of the sensor is given by r_{min} and r_{max} with a range of $65mm$ to $130mm$. Figure 4.16 sketches the principle how a synthetic range image is processed.

In order to model the physical behaviour of the laser light the computation can be simplified. It is sufficient to define two points on the sensor body where the laser beam is transmitted \mathbf{p}_{LD} (Laser diode) and received \mathbf{p}_{CCD} (CCD-camera). These points are set parallel to the x -axis with a symmetric distance $d = 27mm$ with respect to the y -axis. Note, that d is the nominal scan diameter. Then the simulator calculates the scanning

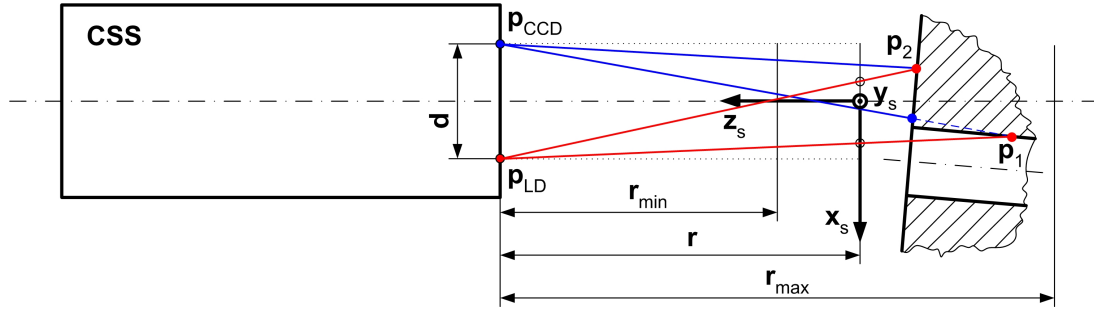


Figure 4.16: Drawing of the CSS simulation.

motion according to the settings and dimensions described in Figure 4.14. The virtual laser beam rotates with a frequency of $7Hz$ calculated with a centered circular point motion around the TCP in the xy -plane. The direction of the transmitted laser beam is given by the second point of the ray, that is \mathbf{p}_{LD} . Now this line is intersected with the CAD model (STL-format) which results in the point \mathbf{p}_1 . To calculate the visibility of the virtual raw data point a line is calculated with the points \mathbf{p}_{CCD} and \mathbf{p}_1 and the CAD model is again intersected with this line. If the result of the first intersection is the same as \mathbf{p}_1 , this point is stored as a point in the synthetic range image with the constraint of the working range boundaries (r_{min}, r_{max}). The transformation into the world coordinate system is described with Equation 4.17. The examples of two laser beams demonstrate with \mathbf{p}_1 a corrupt and with \mathbf{p}_2 a valid virtual raw data point.

The result of a complete synthetic range image process can be seen in Figure 4.17. Here an iron cast part (Figure 4.17(a)) is CAD-modeled in a simplified version (Figure 4.17(b)). Only the inner bore surface and the bore chamfer is modeled in detail. The virtual scanning result of the bore with a diameter of $38mm$ is shown in Figure 4.17(c). Note the radial notches of this complex cast part.

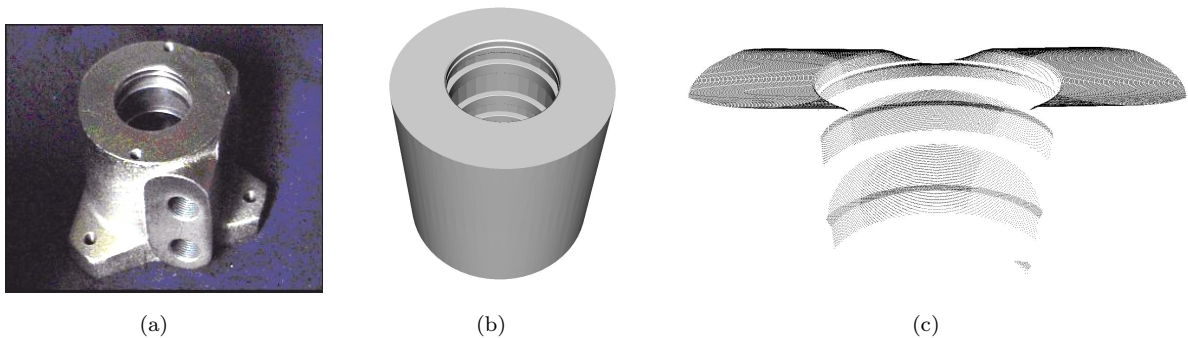


Figure 4.17: (a) Industrial cast part with a bore diameter of $38mm$. (b) Simplified CAD model, notable the detailed modeled bore surface with several notches. (c) Synthetic range image of the circular scanning sensor.

4.2.2 Algorithm Performance Evaluation

Before evaluating the results of the bore detection method and a comparison with other standard fitting techniques the performance of the algorithm itself must be evaluated. This section demonstrates that an approach based on random samples is very efficient concerning the computational cost. Nevertheless, the method still remains robust and does not suffer from inaccuracy. The experiments are carried out on a bore hole with a diameter of $10mm$. Figure 4.18 shows the evaluated bore whose range image simply consists of a plane and the perpendicular bore.

All tests have been performed on a $2.4GHz$ Pentium IV processor with $1GB$ RAM. For visualization of the results and the diagrams VTK and MATLAB have been used. All methods have been implemented in MSVC++ 6.0 which is necessary for time measurements and performance comparisons.

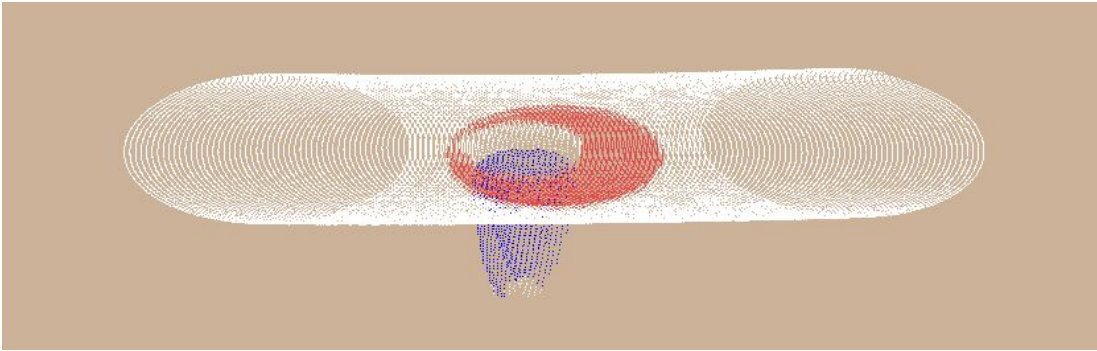


Figure 4.18: Test bore with a diameter of $10mm$ to evaluate the algorithm performance. Note, the segmentation result of the plane and bore surface.

Accuracy Versus Number of Samples

Fischler and Bolles first introduced a random samples and consensus method RANSAC [40] which became a standard technique in computer vision in 1981. Based on this work a wide variety of hypothesis and verifyign methods are presented [81, 86, 88, 130, 131] which all find a fast optimal solution to various vision problems, which there are: wide/short baseline stereo matching, motion segmentation, detection of geometric primitives, structure and motion estimation and object recognition.

In the bore detection algorithm the sequential cylinder fit is also based on two random sample processings, the initial orientation estimation of the cylinder axis and the radius estimation. In Equation 4.7 and 4.15 the random samples are limited to 1,000. This assumption will be verified in the following paragraphs.

Orientation Estimation As a remark, the initial orientation estimation of the cylinder axis is needed for the following circle fit to estimate the radius (see Section 4.1.3). To evaluate the robustness and accuracy of the orientation estimation the number of samples

in the test varies from 10^1 to 10^5 . As this method is a random process each estimation with a particular number of samples is repeated 1,000 times and the results are averaged. This is necessary, to be able to give a reliable interpretation of the result's tendency, especially on estimations with a low number of samples. Figure 4.19 shows the absolute averaged deviation of the orientation vector \mathbf{u} (4.7) compared to the real orientation of the bore.

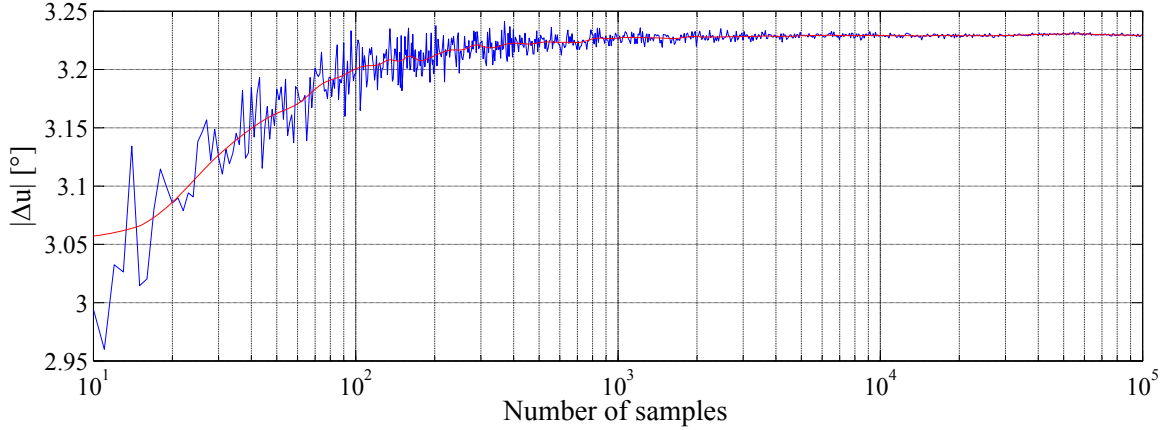


Figure 4.19: Averaged absolute deviation of the orientation estimation.

The result is quite interesting, because the axis orientation converges towards a rather high deviation of more than 3° , but the final bore axis is determined with a deviation of less than 1° . As in Section 4.1.3 described, the orientation is estimated with the cross product of two random normal vectors of the cylinder surface which have an uncertainty of 5° due to the tolerance in the segmentation step. This is the reason why the axis estimation is quite inexact compared to the final deviation. A second reason is the constraint (4.9) limiting the angle between the normal vectors. Normal vector pairs with an angle at the lower limit (15°) contribute cross product vectors with a larger error of deviation than vector pairs whose directions cross perpendicularly. Indeed, if the lower limit is raised to 45° an orientation deviation of 2° is the result. But the accuracy of the final bore axis orientation is almost untouched. This fact can be explained because the deviation of the orientation estimation is not sensitive to the following radius estimation, where the 3D points of the cylinder surface are projected to a plane perpendicular to the estimated orientation axis (2D). The uncertainty of the orientation axis is transformed to a larger noise level, which is partly compensated by the circle fit (see Figure 4.24).

Averaging over 1,000 trials compensates the maximum outliers. Figure 4.20 shows the extremal values of the orientation vector \mathbf{u} represented with smoothed curves. What can be clearly seen is, that the deviation is up to 5° for a small number of samples, but the boundaries converge rapidly to the average orientation. The question is now which number of samples is sufficient for the orientation estimation. A trade-off between the accuracy and the computational effort has to be found. A quantitative expertise of Figure 4.19 and 4.20 gives a break even result at 1,000 samples. This is the number of samples where the orientation \mathbf{u} reaches the final convergence, where the margin of deviation $\tau_{\mathbf{u}}$ in Figure 4.19 significantly decreases and where the maximum outlier deviation is limited to

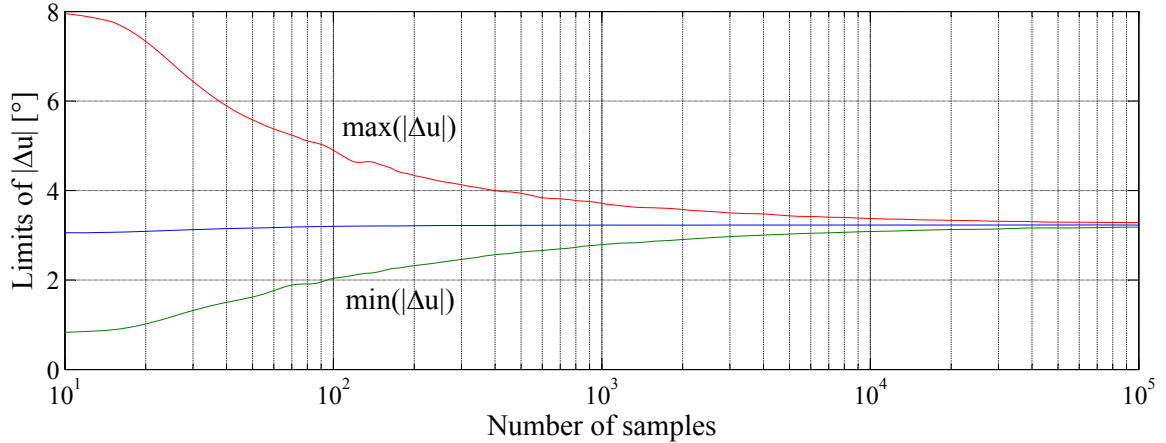


Figure 4.20: Boundaries of the maximum and minimum deviation of the estimated orientation.

0.5°. Furthermore, these deviations do not affect the final accuracy of the radius estimation and therefore 1,000 samples is the optimum number of samples. Table 4.1 summarizes the given implications and shows that an increased number of samples does not significantly improve the result.

number of samples	$ \Delta\mathbf{u} $	$\tau_{\mathbf{u}}$	$\max(\Delta\mathbf{u})$	$\min(\Delta\mathbf{u})$
10^3	3.227°	$\pm 0.001^\circ$	3.711°	2.791°
10^5	3.229°	$\pm 0.0005^\circ$	3.282°	3.182°

Table 4.1: Summarized results of Figure 4.19 and 4.20 with the absolute value of the estimated axis orientation, the margins of deviation and the outlier-limits for 10^3 and 10^5 number of samples.

Radius Estimation The radius is estimated using a circumscribed circle of three random points projected to a plane perpendicular to the estimated cylinder orientation (see Section 4.1.3). To evaluate the robustness and accuracy of the radius estimation the number of samples in the tests ranges again from 10^1 to 10^5 . As well as in the orientation estimation the tests are repeated 1,000 times with a particular number of samples. The result of the radius estimation is shown in Figure 4.21.

Below 10^3 samples the estimated radius is strongly oscillating whereas the estimation is rapidly converging towards $r = 4.963\text{mm}$ for samples above 10^3 . This is a relative error of 0.72% compared to the nominal radius of the bore (5mm). The error of the extremal values of the outliers with $\pm 0.075\text{mm}$ is still quite high and converges slowly against the averaged radius.

Again a quantitative expertise of Figure 4.21 and 4.22 gives a break even result at 1,000 samples, where the radius r reaches the final convergence, and the margin of deviation τ_r in Figure 4.21 significantly decreases. Note that the radius already reaches the final value

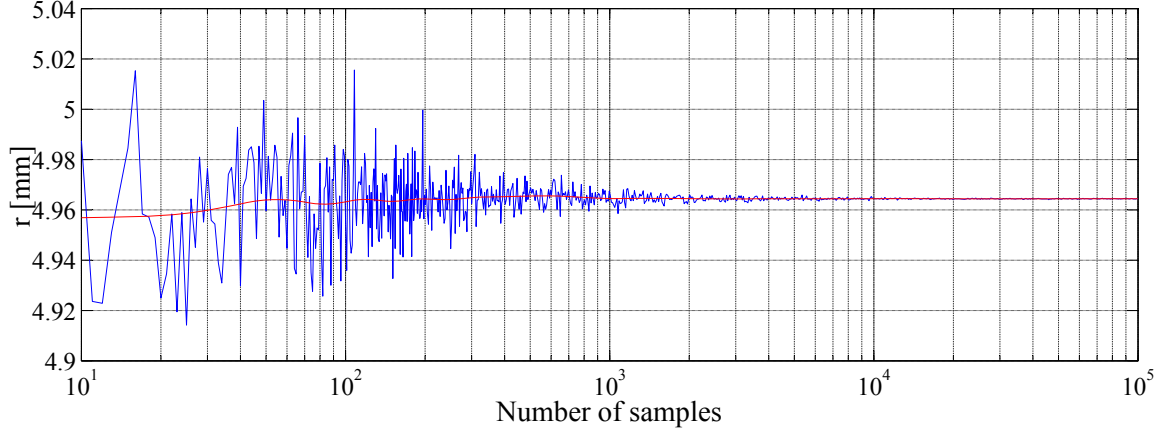


Figure 4.21: Averaged estimated radius.

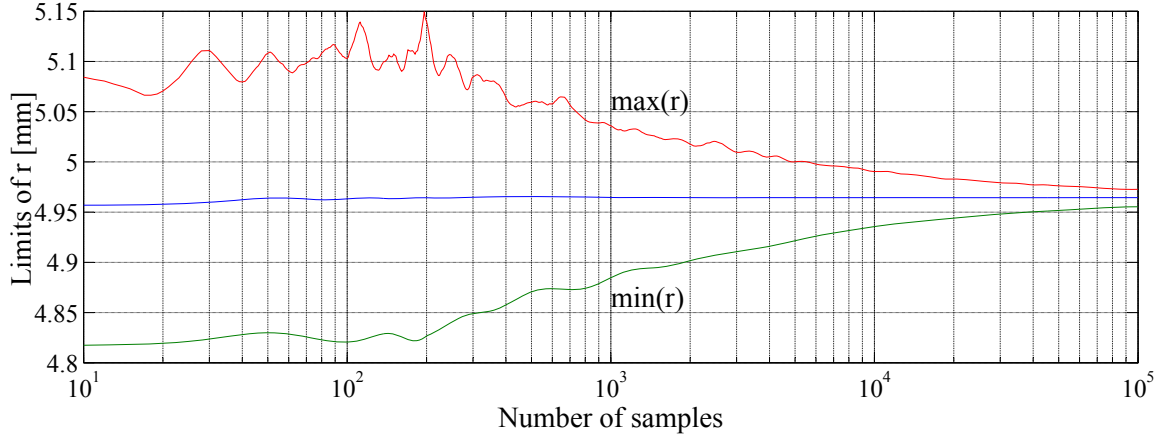


Figure 4.22: Boundaries of the maximum and minimum estimated radius.

number of samples	r	Δr	τ_r	$\max(r)$	$\min(r)$
10^3	$4.963mm$	$-0.036mm$	$\pm 0.0025mm$	$5.036mm$	$4.885mm$
10^5	$4.963mm$	$-0.036mm$	$\pm 0.0mm$	$4.973mm$	$4.955mm$

Table 4.2: Summarized results of Figure 4.21 and 4.22 with the estimated radius, the margins of deviation and the outlier-limits for 10^3 and 10^5 number of samples.

at 1,000 samples. Table 4.2 summarizes the given implications and show that an increased number of samples does not improve the result, except that the extremal outlier values slightly decrease. Also note, that the margins of deviation τ_r tend to go to zero at 10^5 samples.

Radius Estimation Versus Least-Squares Circle Fitting

In Section 4.1.3 a sample based method for the radius estimation is introduced. Calculating the radius with three random points and their associated circumscribing circle, one can ask: Why not using a standard least-squares circle fitting technique to calculate the radius? This section should answer this question by comparing the proposed *Random Sample Fitting* (RSF) with the standard *Least-Squares Fitting* (LSF) method.

In the two previous paragraphs the optimum number of samples for this application is discovered with 1,000. So the accuracy and the time consumption of the RSF with 1,000 samples is compared with the LSF of a circle. The evaluation is performed on full-resolution and sub-sampled raw data sets of the segmented cylinder surface fragment shown in Figure 4.18. Figure 4.23 shows the extracted cylinder surface fragment from (a) to (e) in 100%, 50%, 25%, 10% and 4% resolution. The 2D mapping of the full-resolution data onto the plane perpendicular to the estimated axis orientation \mathbf{u} is shown in Figure 4.24 with the corresponding least-squares circle fit. Note the noisy data with some outliers deeper in the bore.

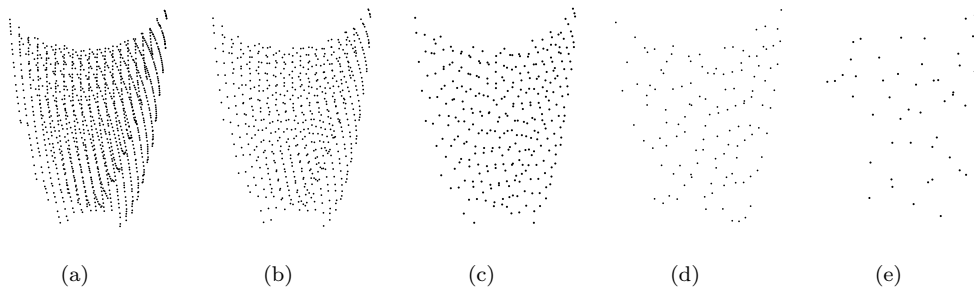


Figure 4.23: Cylinder surface raw data points with (a) 1254 points (full-resolution), (b) 631 points, (c) 313 points, (d) 124 points and (e) 50 points.

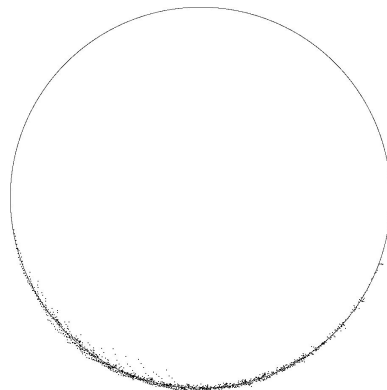


Figure 4.24: Least-squares circle fit of the 2D mapped full resolution bore surface raw data points.

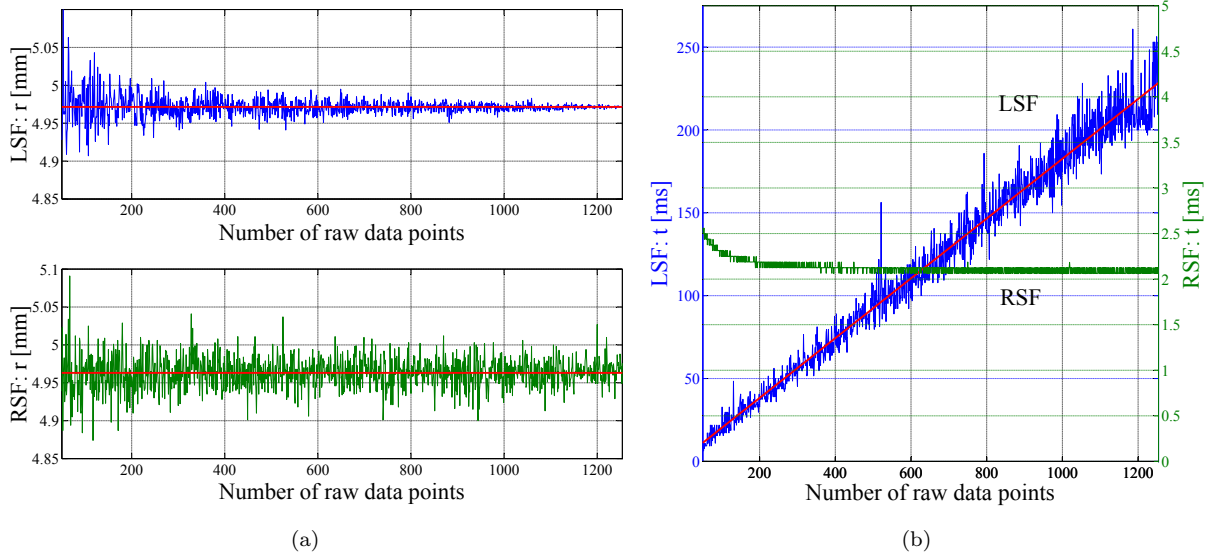


Figure 4.25: Comparison of the RSF (10^3 samples) and the LSF method concerning the processing time and the accuracy.

number of points	t_{LSF}	t_{RSF}	$\tau_{r_{LSF}}$	$\tau_{r_{RSF}}$
1254	228.4ms	2.09ms	$\pm 0.0005mm$	$\pm 0.023mm$
631	116.1ms	2.09ms	$\pm 0.018mm$	$\pm 0.023mm$
313	58.7ms	2.16ms	$\pm 0.022mm$	$\pm 0.023mm$
124	24.6ms	2.25ms	$\pm 0.031mm$	$\pm 0.023mm$
50	11.2ms	2.56ms	$\pm 0.062mm$	$\pm 0.023mm$

Table 4.3: Summarized results of Figure 4.25 of the probes seen in Figure 4.23 with the processing times and the margins of deviation of r of the RSF and LSF method.

The accuracy evaluation of the radius r is presented in Figure 4.25(a). The tests are performed on a continuous sub-sampling beginning from 50 points up to 1254 points. The result is, that both methods find nearly the same mean radius with $r = 4.971mm$ for the LSF and $r = 4.963mm$ for the RSF. The obvious difference is that the margins of deviation of the LSF method converges towards the mean radius. The big advantage of the RSF method can be seen in Figure 4.25(b), that is the computational cost. The processing time for 1,000 samples is nearly constant over the number of raw data points and is approximately 2ms. For a few number of data points the processing time raises a bit, due to the constraint (4.9). The likelihood that two random points do not match the constraint (4.9) is reciprocal to the number of data points. Hence, finding two matching points on a few number of data points takes more time and raises the overall processing time. In contrast to that the LSF method shows an increasing arithmetically characteristic corresponding to a raising number of raw data points. The oscillating behaviour is needed to converge towards the optimum fit because of the different number of iteration steps. Using full-resolution the RSF method is more than 100 times faster than the LSF method.

Even if 10^5 samples are used to increase the accuracy, the RSF method ($210ms$) is still faster than the LSF method ($228ms$). Table 4.3 summarizes the results of Figure 4.25 concerning processing time versus accuracy.

4.2.3 Experimental Detection Results

Before presenting detection results on real and synthetic range images the computational cost of the bore detection method should be evaluated. The timing tests are performed on two different bore holes with $r_1 = 5mm$ (bore #1, see Figure 4.18) and $r_2 = 10mm$ (bore #2, see Figure 4.26). Note the colored segmentation results in both figures. To speed up the computation time the approximated CAD pose is used to cut off all unnecessary raw data points. The red circle indicates (doubled bore diameter) the borders of the points which are processed. Table 4.4 summarizes the key data of the two bores with their processing times.

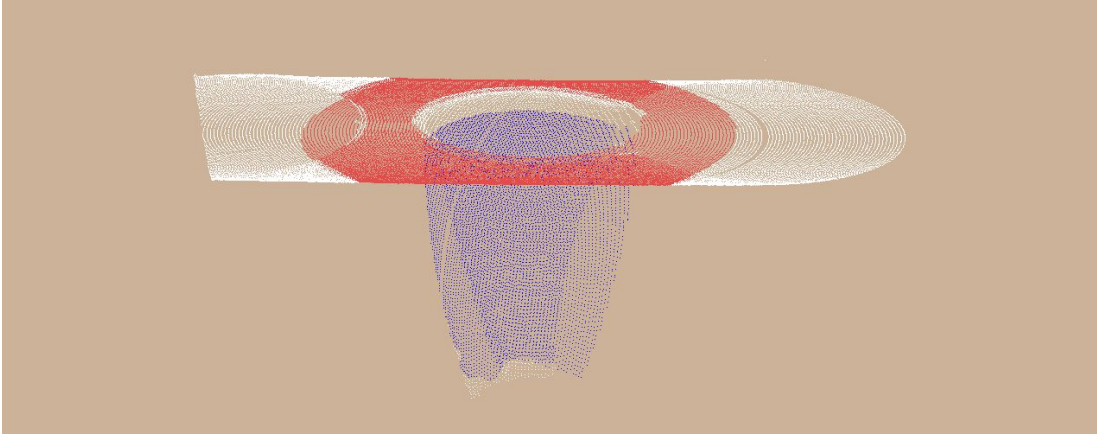


Figure 4.26: Range image of a bore with a diameter of 10mm. Note, the segmentation result of the plane and bore surface.

The obvious difference is the range image size and especially the number of the effective raw data points which are processed. It can be seen that the number of points directly affects the processing time. The demand to compute the bore pose in less than one second can only be achieved in sub-sampling the raw data points. The computation time is given in percentage of the resolution and the processing times in Table 4.4 indicate impressively the effectiveness of the sub-sampling approach without losing the accuracy of the pose detection.

As you can remember, Figure 4.2 outlined the main processing steps of the bore detection approach. According to these steps the method is analyzed concerning the time consumption. The timing results of both bores are shown in Figure 4.27. The most distinctive time consumption is used for the normal vector calculation and the more raw data points are processed the longer the processing takes. This bar chart shows that the normal vector calculation has the largest potential for tuning the method. Sub-sampling is one possibility. The second is to reduce the radius of the surface patch for the local

4. BORE HOLE DETECTION

bore	radius	range image size	resolution	processing points	processing time
#1	5mm	54,966 points	100%	7,481 points	1,386ms
#1	5mm	54,966 points	50%	3,740 points	723ms
#1	5mm	54,966 points	25%	1,870 points	418ms
#1	5mm	54,966 points	10%	748 points	235ms
#1	5mm	54,966 points	4%	299 points	418ms
#2	10mm	71,137 points	100%	45,688 points	16,626ms
#2	10mm	71,137 points	50%	24,344 points	7,797ms
#2	10mm	71,137 points	25%	12,172 points	4,016ms
#2	10mm	71,137 points	10%	4,569 points	1,641ms
#2	10mm	71,137 points	4%	1,827 points	767ms

Table 4.4: Summarized key data and processing time of the bores in Figure 4.18 and 4.26.

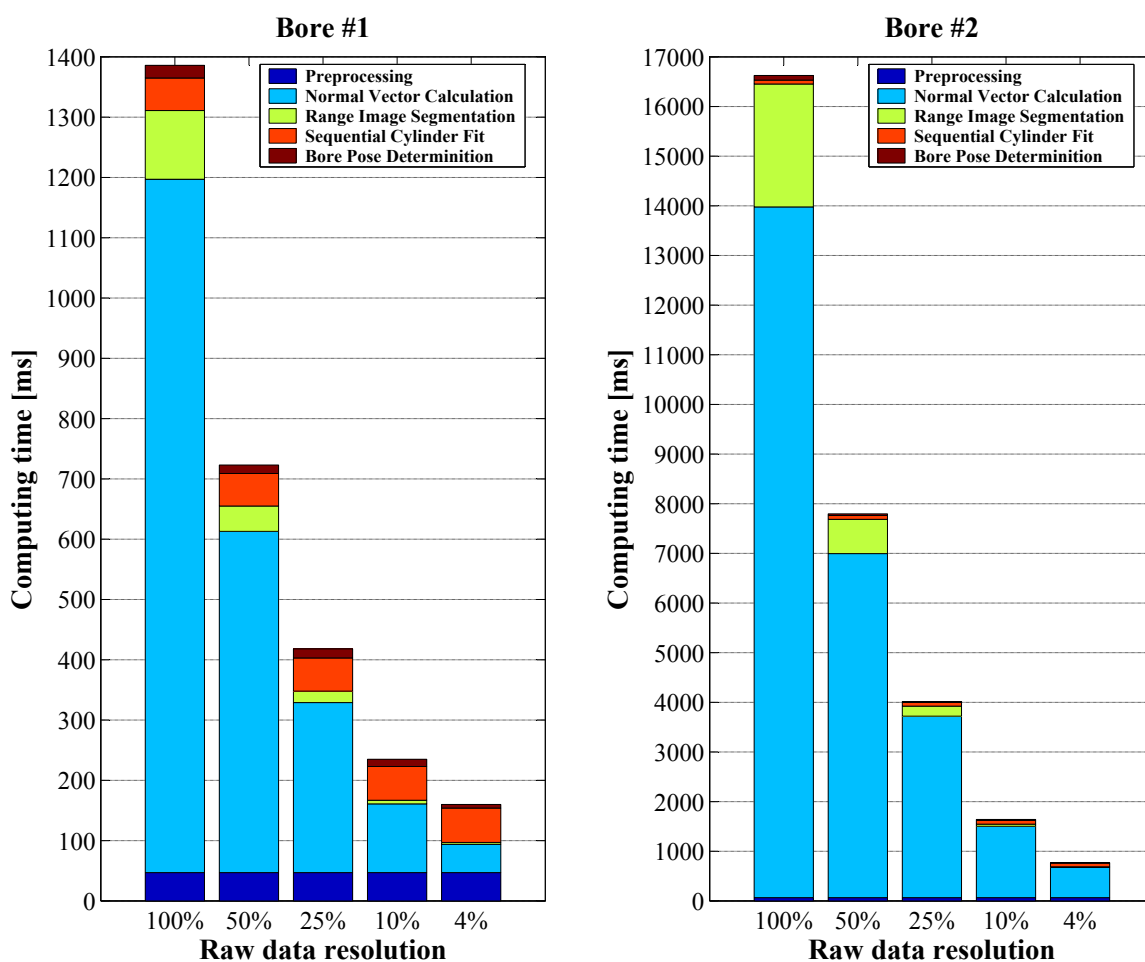


Figure 4.27: Timing bar chart of the apportioned processing steps according to Figure 4.2 of the bores #1 and #2 of Table 4.4.

normal vector calculation. In this performance evaluation the radius is set to $2mm$. Using a radius with $1mm$ reduces the normal vector calculation by more than 50%. The general reason for the high computational effort to calculate the normal vectors is, that for each sub-sampled raw data point the neighboring points are located to extract the local surface patch. To ensure highly accurate normal vectors the point location is still done on the full-resolution data.

Further interpreting the timing bar chart, one can see that the preprocessing and the sequential cylinder fit are independent from the resolution, as expected. The bore pose determination and the segmentation step (Gaussian image) are correlated with the number of processing points but can be neglected in most cases.

Real Range Data

The bore hole detection has been tested with a wide range of work pieces including metal cast and high-speed cutting parts. Tests verified that the bore hole detection algorithm is robust due to an accurate ($0.2mm$) range image quality. Even ghost points and noise in the range image do not affect the detection result.

Figures 4.28-4.30 demonstrate the robustness of the algorithm by presenting some selected examples. In Figure 4.28 two cylinder fits of metallic cast parts are shown. These bores are difficult to detect, because of shifted casting molds, inaccuracies and burrs are the result. Note the range data in the middle of the cylinder in Figure 4.28(b) which is caused by a radial notch. Here a second cylinder fit was necessary to precisely locate the bore.

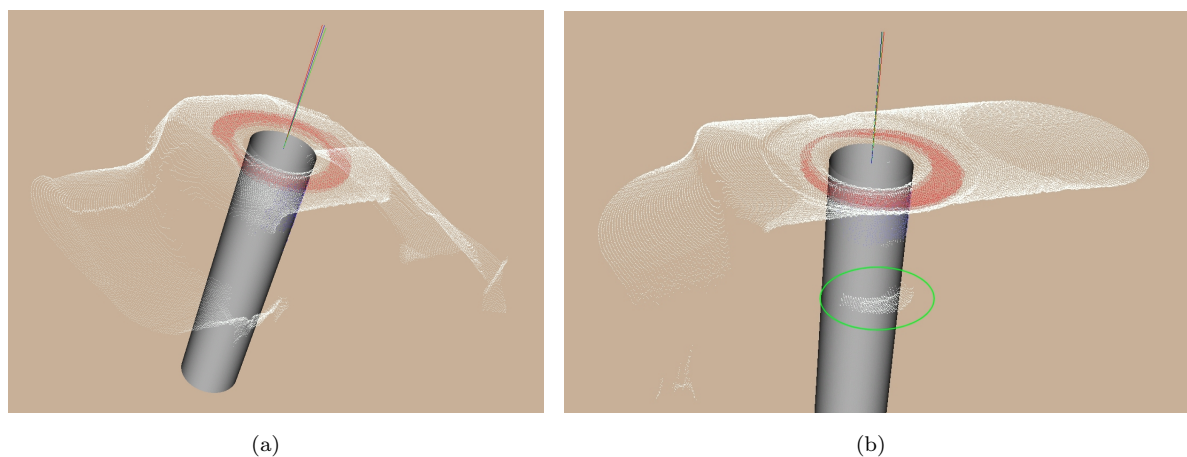


Figure 4.28: Detection results of metallic cast parts. Note the disturbing raw data points caused by a radial notch in the bore.

Figure 4.29 shows the bore hole detection result of a specular metallic high speed cutting part. In the range image, ghost points due to ambient light reflections of the long sharp metallic edge are present. These ghost points do not have an effect to the correct cylinder fit. They are filtered out in the segmentation process because of the wrong normal vector orientation.

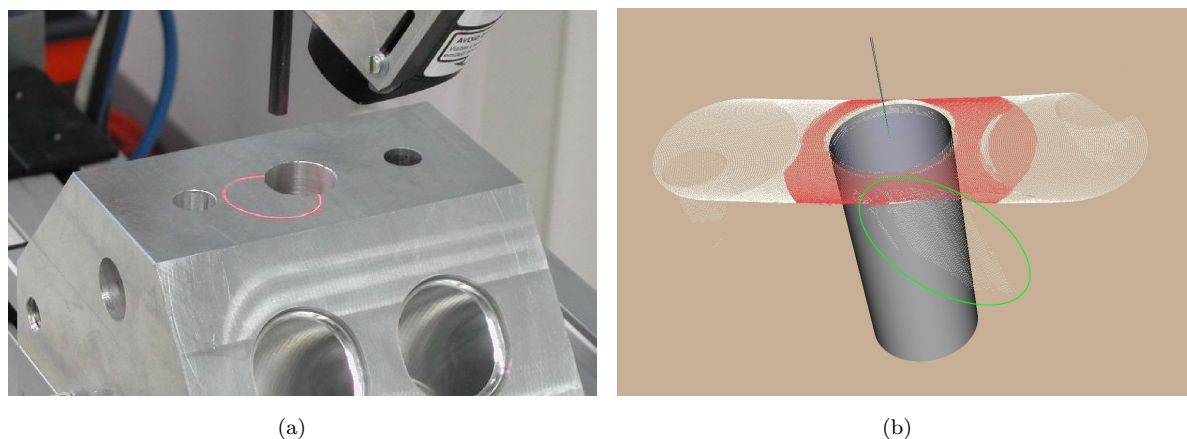


Figure 4.29: Bore detection of a metallic high speed cutting part. The specular metallic edge caused a lot of ghost points in the range image. Nevertheless, the ghost points do not have an effect to the detection result.

Figure 4.30 shows the last two examples underlining the robust detection. In Figure 4.30(a) a bore with a rounded chamfer ($r = 2.5mm$) has been correctly localized. Chamfers are often part of the bore and must not have an effect to the detection result. Note the correct segmentation of the bore surface fragment. Raw data points, which can be associated with the rounded chamfer are not segmented. This bore has a diameter of $15mm$ and was detected with an axis deviation of 0.447° and a radius of $7.477mm$. This approach has the capability to detect bores with any orientation, hence, Figure 4.30(b) visualizes a correctly detected slanted bore hole (45°). Despite the fact that the bore hole is scanned only partly due to a large part positioning error, the approach is robust enough to handle this sparse data.

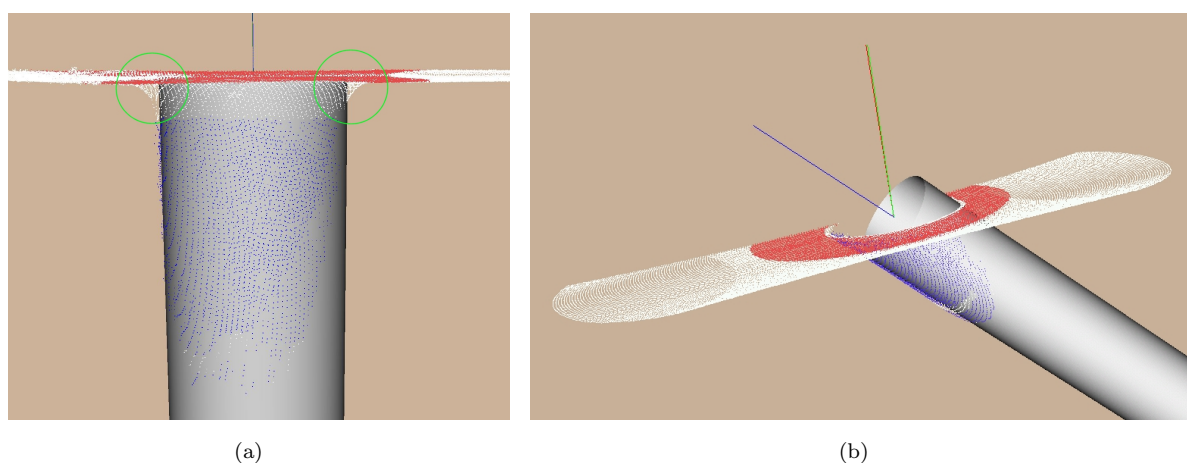


Figure 4.30: Detection results of a bore with a rounded chamfer ($r = 2.5mm$) (a) and a 45° slanted bore (b).

These tests verified that the pose detection obtained a position accuracy of $\pm 0.3mm$ and the axis orientation can be estimated with less than 0.5° deviation. Bore holes with a larger diameter tend to give better results in shorter processing time because of more raw data points available on the cylinder surface fragment and the capability of a larger sub-sampling rate. Bore holes with radial or axial notches can be robustly detected by automatically applying a second cylinder fit.

Simulated Range Data

Virtual tests have also been performed to evaluate the method under atypical conditions. The first test investigates the noise behaviour of the bore detection method. Therefore, a probe with three equal ($r = 6mm$) and equidistant ($15mm$) bores has been simulated. To ensure a realistic scan, Gaussian noise ($\sigma = 0.15mm$) has been added to the data. To create a bad range image quality additionally salt and pepper noise has been added to 5% of the raw data points with a maximum amplitude of $\pm 10mm$ along the sensor's z -axis. The first fit result was ill-posed, because of the cylinder surface points of the neighboring bores in the red marked search area (see Figure 4.31). Hence, an automatic second cylinder fit was applied resulting in a satisfying pose determination. The cylinder was fitted with a radius of $6.049mm$ and an axis deviation of $5 \cdot 10^{-6}$ degrees. The same experiment without added noise resulted in a radius of $5.999936mm$ with the same orientation deviation.

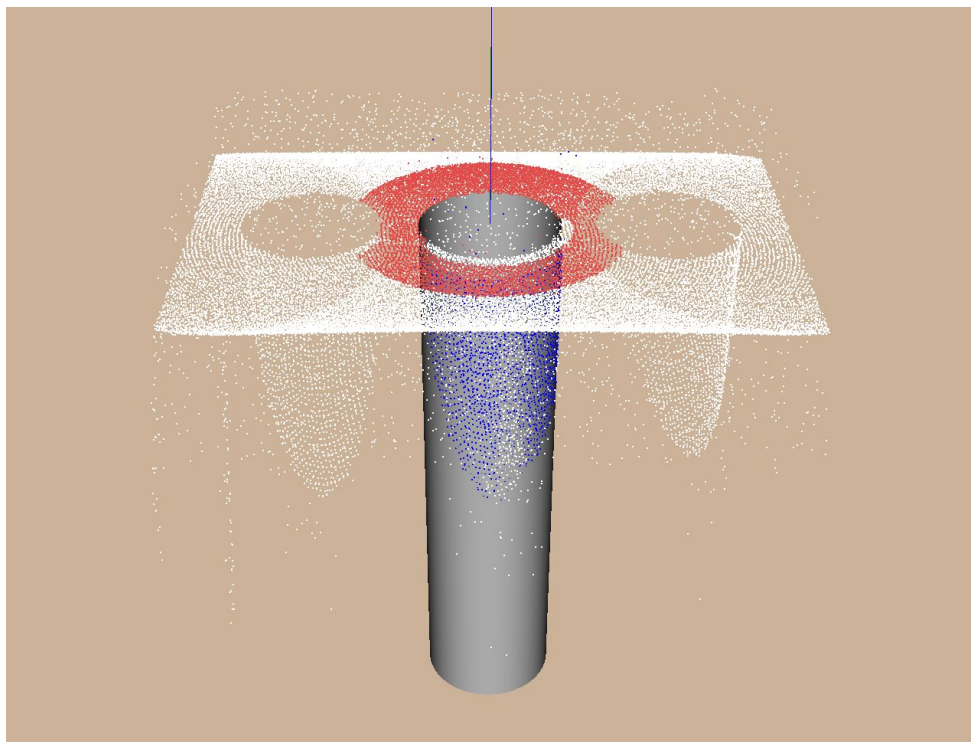


Figure 4.31: Experiment on a noisy range data detecting a bore with a diameter of $12mm$.

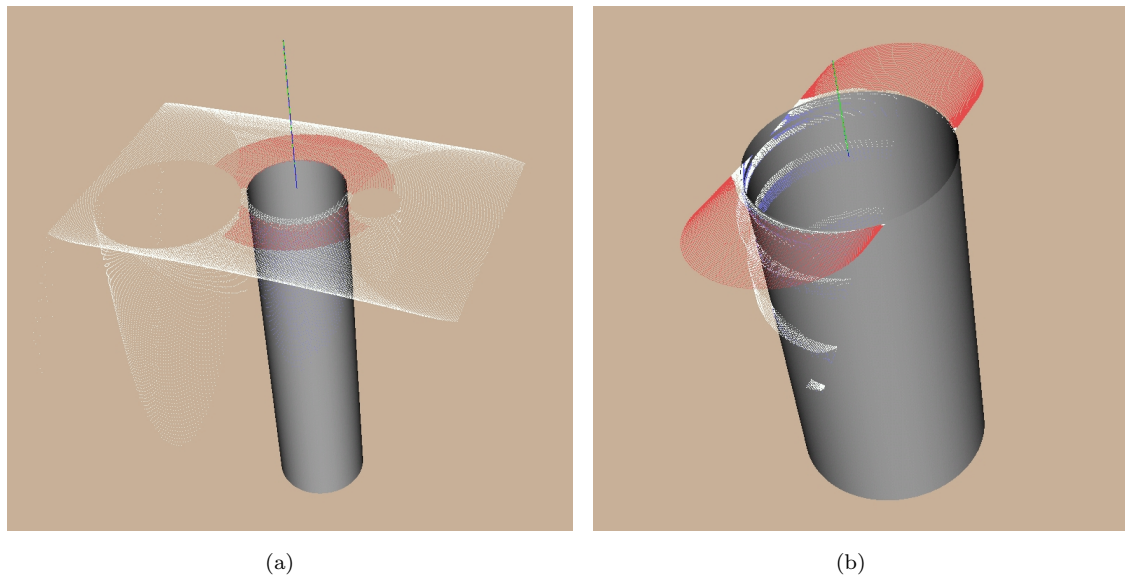


Figure 4.32: Two experiments investigating bores in close relationship and a bore with notches whose diameter is larger than the scan area.

Recognizing that bores in close relationship can cause troubles, an experiment has been conducted investigating three bores located directly nearby (distances between the bores: $1mm$) as shown in Figure 4.32(a). Although the segmentation process detects range data of the other bore hole surfaces the result is within the required accuracy again because of the two step cylinder fitting.

Large bore holes can be also detected even if the bore diameter, here $38mm$, is larger than the scan area. Figure 4.32(b) shows the detection result of the virtual modeled part in Figure 4.17. The difficulty in this detection process are the radial notches, but they have again no effect to the correct result because a second cylinder fit was applied.

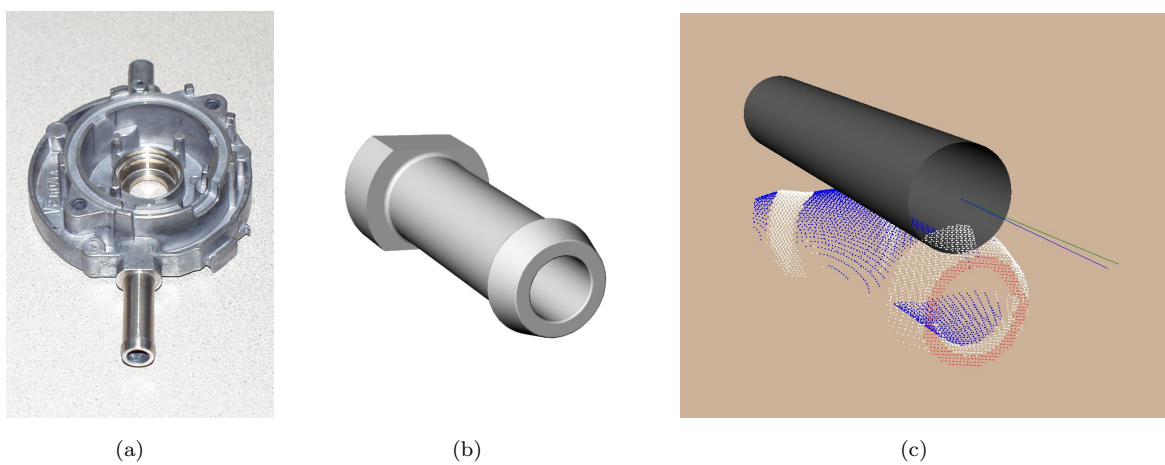


Figure 4.33: Experiment demonstrating the detection-problem of thick-walled pipes.

As last experiment the inspection task of a thick walled-pipe should be tested. Figure 4.33(a) shows a flexible tube adapter with the modeled adapter shown in Figure 4.33(b). Figure 4.33(c) presents the detection result which presently fails. The approach proposed is not able to handle concentric cylinders because the Gaussian image cannot separate concave and convex surface normal vectors. Hence, improvement of the method is necessary implementing a post processing filter that separates normal vectors of inner and outer cylindrical surfaces based on the divergence or convergence of neighboring normal vectors.

4.2.4 Experimental Performance Comparison

As last performance evaluation the sequential cylinder fit is compared with a least-squares cylinder and Superquadric fitting. As fitting criteria the radius and the final orientation of the cylinder and the computational effort are investigated. As already shown in Section 2.2.2 the least-squares fitting of geometric models requires an iterative optimization process, which is in most cases of non-linear nature. In this performance study the Steepest Descent method is used for the fit of the cylinder model and the Levenberg-Marquardt method for the Superquadric fit (compare Section 2.2.2). To keep the results comparable with the other tests, the bore in Figure 4.18 is used again. The fits are based on the bore surface fragment shown in Figure 4.23(a) (fully raw data resolution). Figure 4.34 shows the result of the cylinder and Superquadric fittings and should point out the problems to deal with.

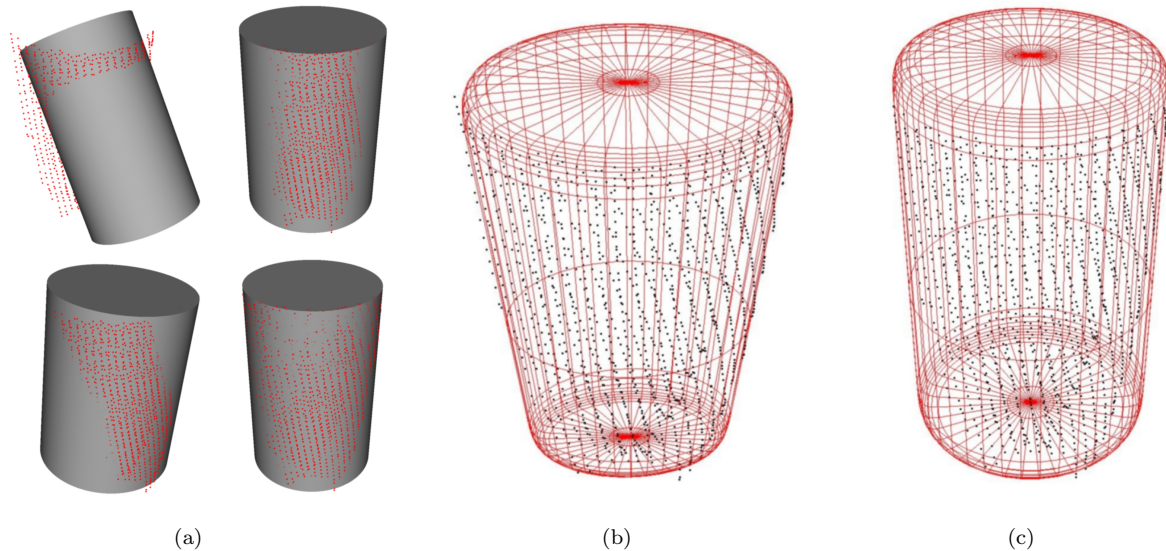


Figure 4.34: Cylinder and Superquadric fits to the raw data points of the bore surface fragment of Figure 4.23(a). (a) Top row: Initial pose estimation using PCA and fit results after 10 and 200 iterations. Bottom row: Again fit results after 10 and 200 iterations, but started with the CAD pose. (b) Model recovery of a tapered Superquadric. (c) Cylinder fit result with a cylindrical constrained Superquadric.

Each minimization process searching for a local minimum requires a sufficient starting solution to hopefully converge against the global minimum. As first step the least-squares cylinder fit is considered in detail before investigating the Superquadric fit.

Observing the bore surface fragment, one can see the sparse raw data representing in maximum 120° of the cylinder surface. This brings us to the first problem finding a good starting pose of the cylinder axis. The first attempt to get a good estimation is to use the PCA algorithm. This method works well for full point data of a cylinder, but dealing with this sparse raw data set the misplacement in the first iteration steps is rather high, which can be seen in the top row of Figure 4.34(a). On the left side the fit result is shown after 10 iterations and on the right side the result after 200 iterations. Improving the placement of the starting pose the CAD pose of the bore can be used for cylinder axis estimation and the corresponding results can be seen in the bottom row again presenting 10 and 200 iterations. The improvement can be observed on the equal distribution of the raw data points on the cylinder surface. But note, that the fit does not obligatory converge to the global minimum. If the displacement is too large the optimization process is trapped in a local minimum. Table 4.5 summarizes the accuracy of the LSF results with the different starting pose estimations compared to the RSF method. In this case $|\Delta\mathbf{v}|$ the deviation of the final cylinder axis denotes to the ground-truth bore axis (compare Equation 4.16). It can be clearly seen that the results are in the same range but the RSF method is approximately 450 times faster. Also note that the fit with the PCA estimation still needs more iteration for a complete convergence.

method	pose estimation	iterations	r	Δr	$ \Delta\mathbf{v} $	time
LSF	PCA	200	5.581mm	+0.581mm	0.612°	972ms
LSF	CAD	200	4.987mm	-0.013mm	0.911°	972ms
RSF	CAD	-	4.962mm	-0.038mm	1.096°	2.09ms

Table 4.5: Comparison of the accuracy and the computational effort of a cylinder fit using LSF and RSF method.

Fitting a Superquadric is no simple task by simultaneously minimizing 13 parameter, the Levenberg-Marquardt method is well suited to solve this problem. Indeed, fitting a Superquadric is an overhead detecting of a cylindrical bore, but the reader should get a better understanding in the next chapter. The Levenberg-Marquardt method needs a starting pose estimation, as well. Because of the fast convergence of this algorithm the PCA estimation is sufficient to estimate the starting pose. For the detailed implementation of a Superquadric fit to range data see [116]. Figure 4.34(b) shows the first attempt of fitting a Superquadric with global taper deformation. The fit is well aligned with the data points, but the recovered model has no cylindrical shape – it has a shape of a pinched frustum. This Superquadric model consists of 13 parameters that can cover a wide variety of differences. Nevertheless, the Superquadric can also be used for precise geometrical measurements as shown in [2]. Constraining the parameter set Λ (2.24) to $r = a_1 = a_2$, $\varepsilon_1 = 0.1$, $\varepsilon_2 = 1.0$ and $k_x = k_y = 0$ the shape is fixed to the cylinder and the fit result is shown in Figure 4.34(c). The fit parameters are summarized in Table 4.6 and again compared to the RSF method, with the same fit result except the processing time. The

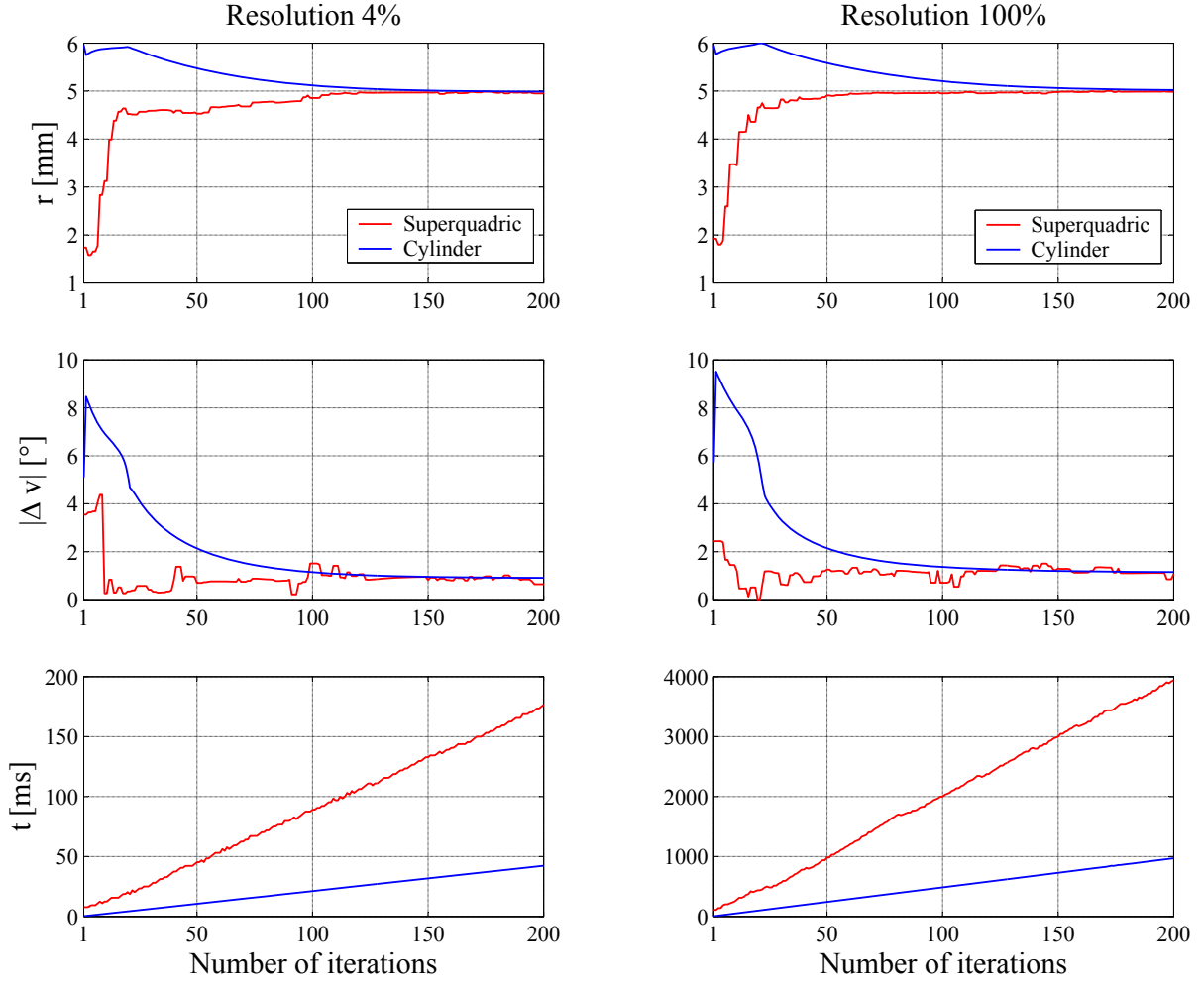


Figure 4.35: Evaluation of the cylinder and Superquadric fit performance on full-resolution and sub-sampled raw data sets.

RSF method is approximately 2,000 times faster, but note, that constraining the parameter in the Levenberg-Marquardt algorithm wastes more processing time.

method	a_1 [mm]	a_2 [mm]	a_3 [mm]	ε_1	ε_2	k_x	k_y	r [mm]	Δr [mm]	$ \Delta v $ [°]	t [ms]
LSF Fig. 4.34(b)	4.224	3.026	7.657	0.1	0.998	-0.144	-0.306	-	-	7.257	3235
LSF Fig. 4.34(c)	4.962	4.962	8.406	0.1	1.0	0	0	4.962	-0.038	1.079	3938
RSF	-	-	-	-	-	-	-	4.962	-0.038	1.096	2.09

Table 4.6: Summary of the recovered Superquadric and cylinder parameter in comparison with the LSF and RSF method.

The last diagrams in Figure 4.35 compare the accuracy of the final axis orientation and the radius against the number of iterations, as well as the computation time is evaluated. The diagrams in the left column represent the results of sub-sampled raw data points (compare Figure 4.23(e)), while the right column shows the result in full resolution. As expected (compare Section 2.2.2), the Levenberg-Marquardt method converges faster than

the Steepest Descent method. But fitting a Superquadric is more time consuming than fitting a cylinder. The time consumption is linear with the number of iterations. The processing is quite faster on sub-sampled data, but takes again more time to achieve a comparable accuracy than with the RSF method. On the sub-sampled raw data set at least 130 iterations are needed, that result is $28ms$ for the cylinder fit and $115ms$ for the Superquadric fit.

Concluding the experimental evaluation, the proposed detection approach achieves reliably robust results at a fraction of time compared to other standard fitting approaches. Achieving such low processing times has two reasons. The first is the random sample based approach achieving fast results independently from the raw data points, and second, the precalculated normal vectors are used for the fit. One can say that a least-squares fit does not need to calculate normal vectors, but the segmentation needs it and therefore it must be calculated. At this point you are referenced to the experimental evaluation of Chapter 5 where the Superquadric-based detection approach is applied to detect bores without previous segmentation but with the knowledge of the cylinder model parameter (see Section 5.2.4).

As final remark, some words should be mentioned concerning the detection accuracy. In the FibreScope project the bores are located with a position accuracy of $\pm 0.3mm$ and with an axis orientation offset smaller than 0.5° . In Table 4.5 and 4.6 the orientation deviation is larger than 0.5° . In this chapter only the detection method is presented. The missing link to achieve highly accurate localization is the calibration between the robot arm and the sensor, which is not part of this thesis. A description of the calibration approach can be found in [12].

Chapter 5

Object Detection

In this chapter an efficient object detection approach using Superquadrics is presented. Any task-based system needs to first learn about the relevant object. This is achieved by specifying a simplified Superquadric model of the wanted object or a unique part of it. The size, shape and taper parameter and potentially the Superquadric relations are set according to the best object description. A standard method to describe this model is to scan the target object alone, decompose it into the main parts and fit the Superquadric model. A comprehensive description of the modeling process is partly given in Chapter 6 and 7.

The power of the detection approach is that it is sufficient to describe the approximated overall shape with one or a few number of Superquadrics with global deformations. As an example, the wooden bowl in Figure 5.1(a) and its approximated model description will guide us through the description of the method. Table 5.1 gives the size and shape parameter of the Superquadric model in Figure 5.1(b).



Figure 5.1: A wooden bowl (a) and its approximated Superquadric model (b).

Constraining the approach we first assume that the object to be searched for is in the scanned area. Otherwise, the object which is most similar to the searched object will be

a_1	a_2	a_3	ε_1	ε_2	k_x	k_y
60mm	60mm	20mm	0.4	1.0	0.3	0.3

Table 5.1: Summarized size and shape parameter of the model description in Figure 5.1.

selected. And secondly, the scan is only available from one-view, as it is given in robotic applications (compare Figure 5.3). So the approach has to deal with the missing rear information of the objects and must handle laser- and camera-shadows as well as self and object occlusions in the range data.

5.1 Method Description

The system, which is outlined in Figure 5.2, receives the object model and the scene which is represented by an unstructured point cloud scanned from one-view (see Figure 5.3). The preprocessing removes the dominant plane from the range image which is in most cases the table or the ground plane. This information is not necessary for detection, but wastes valuable computation time in the detection process. In order to achieve fast detection results a probabilistic approach is used to generate pose hypotheses and for keeping the computational effort low the search process is structured in a two-level hierarchy.

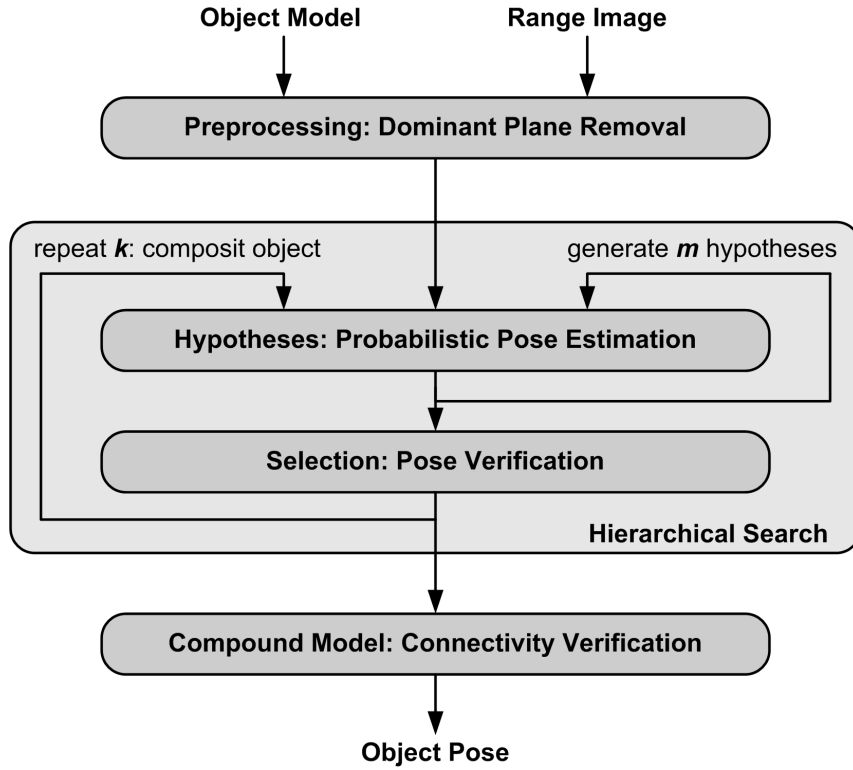


Figure 5.2: Flow chart of the proposed object detection approach.

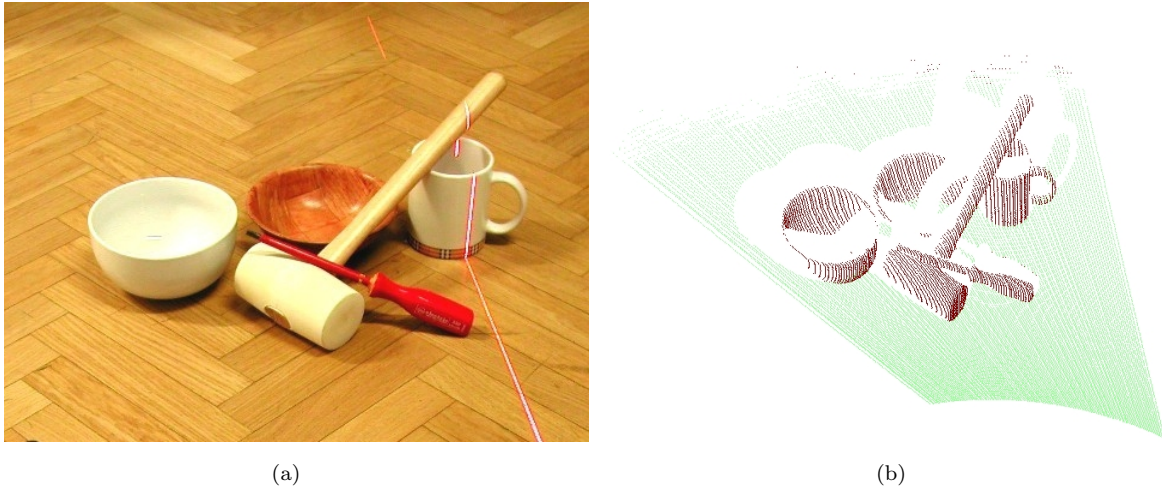


Figure 5.3: (a) Intensity image of a typical scene consisting of a ceramic and a wooden bowl, a mug, a screwdriver and a rubber-mallet. (b) The corresponding range image is shown with the ground plane detected, indicated with the green raw data points. Note the typical camera and laser shadows of a single-view scan.

First the low-level search (probabilistic pose estimation) is RANSAC-based [40] with n samples on sub-scaled raw data to speed up the Superquadric recovery using the Levenberg-Marquardt (compare Section 2.2.2) minimization. The best fit of the low-level search is iteratively refined, which generates one pose hypothesis. To ensure a true positive detection result m hypotheses are computed, where m indicates how often the object could appear in the view of the given scene.

Second, a high level selection (pose verification) uses the full resolution image to verify the object existence. To achieve this a voting [95] for the pose hypotheses is proposed considering three quality of fit constraints: the measure of fit, the number of points on the Superquadric surface and the number of the Superquadric's interior points.

If the model describes a composite object ($k > 1$) the pose verification has to be extended to the compound model pose. An additional voting of the Superquadrics' relations, the main axis steradian and the center distances, is used. Detailed information about the processing steps are given in the following sections.

5.1.1 Preprocessing: Dominant Plane Removal

Most of the range images of a table scene consist of the table plane. However, this plane is not needed for the object detection and, so much the worse, it slows down the object detection process and raises the likelihood of false detections. Hence, the first step is to detect and remove the raw data points associated with the ground plane. Furthermore, the information of the plane can be used as reference for the robot. We define the ground plane as the dominant plane in the range image associating more raw data points than in the rest of the scene.

Finding the dominant plane is achieved by fitting local plane patches and extending

them to verify a global plane. We radically decrease the resolution of the range image to 100 seed points equally distributed over the scene. Each point is now a seed for fitting the plane patch to the points in the close neighborhood using the principal component analysis (PCA) [92]. The normal vector of the surface patch corresponds with the vector \mathbf{n} which is determined by the eigenvalue problem $C\mathbf{n} = \lambda_{min}\mathbf{n}$, where λ_{min} is the smallest eigenvalue and C is the covariance matrix of the point set (see Equation 4.3). The plane point is the mean of the surface patch point set. Each plane hypothesis is now verified against the other 99 points by calculating the normal distances to the plane. A point belongs to the plane if the distance is smaller than the median of all normal distances. The plane hypothesis with the most number of supporting points is the dominant plane. If it contains more raw data points than the rest of the scene the raw data points of this plane are removed, otherwise no ground plane is detected. Fig. 5.3(b) shows the robust detection of the ground plane in the cluttered scene shown in Fig. 5.3(a).

5.1.2 Hierarchical Object Detection

The computational bottle neck of the processing is the nonlinear least-squares minimization routine. Given a set of points $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ the pose parameter of the parameter set Λ (2.24) are recovered applying the Levenberg-Marquardt algorithm in a least-squares minimization. Extending the inside-outside function (2.15) to the parameter set Λ the following expression must be minimized

$$\min \sum_{k=0}^n (F^{\varepsilon_1}(x_k, y_k, z_k; \Lambda) - 1)^2. \quad (5.1)$$

The exponent ε_1 in Equation (5.1) is necessary for optimal minimization to have a correct distance measure independent to the shape parameter. Note the missing factor $\sqrt{a_1 \cdot a_2 \cdot a_3}$ for a volume minimizing recovery as described in [116]. In this object detection approach the shape, scale and taper parameters are given by the target object and the pose (position and orientation) parameters are estimated.

Therefore, the computational effort of the iterative Levenberg-Marquardt algorithm is reduced by only minimizing six variables simultaneously (ϕ , θ , ψ , p_x , p_y and p_z in Equation 2.24). This still time consuming algorithm is executed as often as samples are computed. Its processing time depends on the number of points to be fitted and on the number of iterations to converge the optimization.

The classical RANSAC algorithm is a robust and reliable method to detect an object. Using the hierarchical structure it is shown that an increased detection robustness is achieved in less time than $n \cdot m$ RANSAC samples. The reason being is that the method works on different scaling levels applying different evaluation criteria (see Section 5.2.2).

An important minor detail for every Superquadric fit minimization should be mentioned. While recovering a set of points the initial z -axis – the orientation of the Superquadric – must be assigned to a local coordinate system. The initial orientation is obtained from the central moments of the point cloud to be fitted. In [116] the authors tried to define a rule to distinguish between a longish and a flat-shaped Superquadric to set the orientation of the z -axis. Due to occlusions, intersections and spare range data

of an object the orientation of the Superquadric cannot be estimated correctly according to this rule. The solution proposed here is to execute three estimations (sub-sampled Superquadric fits with less iterations) where the z -axis is aligned with their central moment directions. Three fits are executed and the central moment direction which causes the smallest squared error sum of the inside-outside function (2.15) is chosen as initial z -axis. This eliminates the shape ambiguities in a robust way.

Detailed information about the hierarchical processing – finding the hypotheses and selection – is given in the following sections.

Hypotheses: Probabilistic Pose Estimation

For finding first hypotheses in the low-level search the RANSAC algorithm is exploited. The number of hypotheses m that will be sampled depend on the ratio of the Superquadric size and the range image dimensions to allow to covering the whole scene. Each sample calculation is started by picking a random point in the raw data set. Then a set of 20 randomly chosen points within a radius of the smallest size parameter of the Superquadric around the seed points neighborhood is extracted from the range image. Using this point set a Superquadric is recovered, where the algebraic distance \mathcal{M} is the measure of fit [44]. \mathcal{M} is the mean of the algebraic distance calculated with the inside-outside function of Equation (2.15), that is,

$$\mathcal{M} = \sqrt{\frac{1}{n} \cdot \sum_{k=0}^n (F^{\varepsilon_1}(x_k, y_k, z_k; \Lambda) - 1)^2}. \quad (5.2)$$

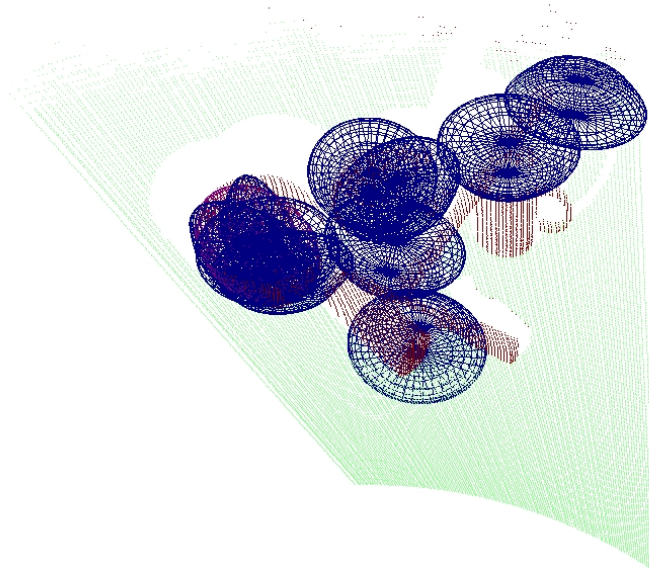


Figure 5.4: Illustration of $n = 10$ RANSAC pose estimations.

Figure 5.4 illustrates the low-level search of $n = 10$ RANSAC samples with a model fit to the surface patch of the n seed points. The pose estimation with the smallest value of \mathcal{M} is then used for a refinement step. This is necessary because the pose hypotheses found by the RANSAC algorithm are not well aligned with the raw data points. The refinement is an iterative process on the local full resolution data points to achieve a better adjustment of the Superquadric for the local surface fitting. All raw data points in the neighborhood of the recovered Superquadric, that are points within the radius $r = 1.2 \cdot \max\{a_1, a_2, a_3\}$ of the Superquadric's center, are used to fit the Superquadric model. This iterative step is repeated until no more improvement of the measure of fit \mathcal{M} can be achieved. The result of the probable pose estimation step are m object pose hypotheses. Figure 5.5 illustrates the refinement process by presenting the initial RANSAC pose (red) and the final refinement pose. Note that two iteration steps are necessary in Figure 5.5(a) where only one step is required in Figure 5.5(b) and 5.5(c).

Selection: Pose Verification

The found and refined pose hypotheses from the low-level search are now candidate poses for the final pose. These poses, the green results in Figure 5.5, are now verified using voting with additional criteria exploiting full resolution data. The measure of fit \mathcal{M} is not a sufficient criteria for a correct object detection. Shape and size ambiguities in the scene may lead to false detections (compare the fit results of the ceramic and the wooden bowl in Figure 5.5(a) and 5.5(b)), because it is a fit to a local surface patch, hence, additional information is needed to robustly verify the pose hypotheses. Exploiting the fact that the Superquadric object model represent the surface of the entire object no raw data points of the single-view range image should be located inside the recovered Superquadric. And secondly, the probability that a recovered Superquadric represents the searched object increases with the number of raw data points located on the Superquadric's surface.

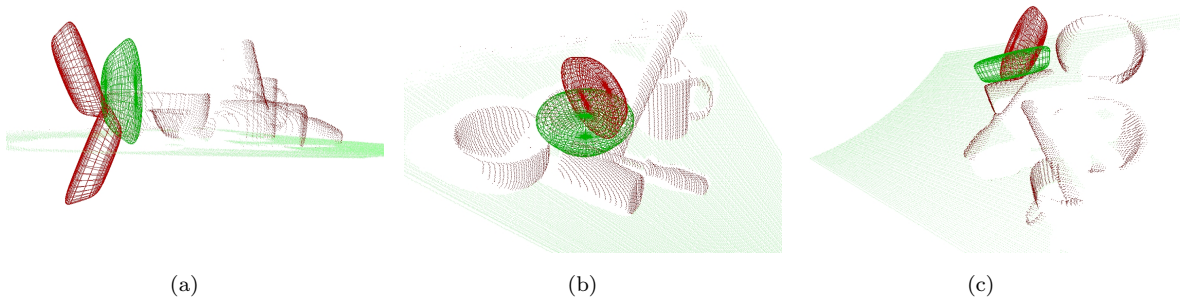


Figure 5.5: Refinement steps (red) and final candidates for pose verification (green).

Let \mathbf{X}_n be again a set of raw data points in the neighborhood of the recovered Superquadric, that are points within the radius $r = 1.2 \cdot \max\{a_1, a_2, a_3\}$ of the Superquadric's center. We can define the evaluation criteria \mathcal{I} for the number of the Superquadric's inte-

rior points and \mathcal{S} for the Superquadric's surface points by

$$\mathcal{I} = \left\{ \sum_{k=0}^n \mathbf{x}_k \in X_n \mid F^{\varepsilon_1}(x_k, y_k, z_k; \Lambda) < 1 - \gamma \right\}, \quad (5.3)$$

$$\mathcal{S} = \left\{ \sum_{k=0}^n \mathbf{x}_k \in X_n \mid 1 - \gamma \leq F^{\varepsilon_1}(x_k, y_k, z_k; \Lambda) < 1 + \gamma \right\}, \quad (5.4)$$

where γ can be selected depending on the application specification required in all expressions. In most cases γ is set to 0.2 allowing a 20% deviation.

A ranked voting using these three criteria \mathcal{M} , \mathcal{I} and \mathcal{S} is the key to achieve a robust hypothesis selection (compare Section 5.2.2). The voting process is performed using the sorted ranks of \mathcal{M} , \mathcal{S} and \mathcal{I} . The hypothesis with the lowest sum of all ranks is selected. This procedure does not need any parameters but uses the natural sorting, and it combines three criteria, which individually describe only specific object characteristics but which together give a quite complete constraint to select the best hypothesis.

To summarize, this hierarchical two-level search achieves a fast and robust detection result especially in cluttered scenes. Because of fitting the object model to local surface patches and verify them globally within the verification step, disconnected surface patches can be associated to one part. This enables the robust detection of partly occluded objects, as demonstrated in Figure 5.6. Note the heavily occluded bowl whose results in a large number of missing raw data points as can be seen in Figure 5.5(c).

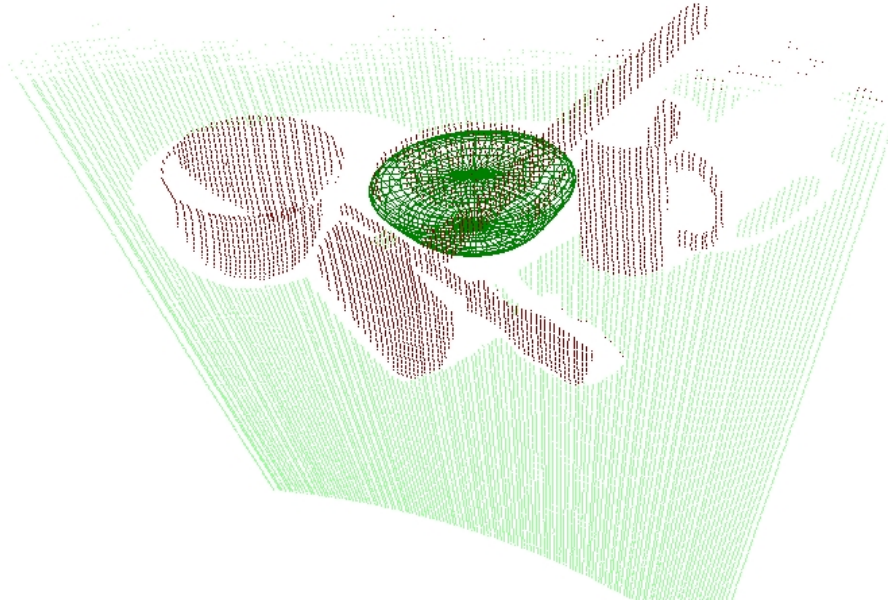


Figure 5.6: Final result after the voting process.

5.1.3 Compound Model: Connectivity Verification

A composite object using a model description with two or three Superquadrics needs an extension in the voting process to include the relationship between the Superquadrics. Krivic et al. use two vectors for the joint positions and a ZYZ rotation [67] to describe linked Superquadrics. For our purpose we show that it is sufficient to use a more efficient connectivity description despite the fact that the description is not unique. We found that the steradian between the two neighboring Superquadric's z -axes (\mathcal{A}) and the distance between the Superquadric's centers (\mathcal{D}) is sufficient (see Figure 5.7). This reduces the connection parameters from 6 to 2, compared to Krivic's approach. These two evaluation criteria are exploited to increase the detection robustness using again voting [95]. Defining the vector \mathbf{a} as Superquadric z -axis and the point \mathbf{x}_c as Superquadric center brings us to the following criteria,

$$\mathcal{A} = \left| \alpha_k - \arccos \left(\frac{\mathbf{a}_i \cdot \mathbf{a}_j}{|\mathbf{a}_i| \cdot |\mathbf{a}_j|} \right) \right|, \quad (5.5)$$

$$\mathcal{D} = \left| d_k - \sqrt{(x_{c_i} - x_{c_j})^2 + (y_{c_i} - y_{c_j})^2 + (z_{c_i} - z_{c_j})^2} \right|. \quad (5.6)$$

The evaluation criteria \mathcal{A} and \mathcal{D} are the absolute differences between the permutations of all m low level fits and the nominal angle α_k and the distance d_k specified in the model (see Figure 5.7). Again the sorted ranks \mathcal{A} and \mathcal{D} for the final voting are calculated and stored in a list. The lowest sum of the ranks of each evaluation criteria \mathcal{A} and \mathcal{D} with their associated result \mathcal{M} , \mathcal{I} and \mathcal{S} represents the parameter set Λ_k of the object detected.

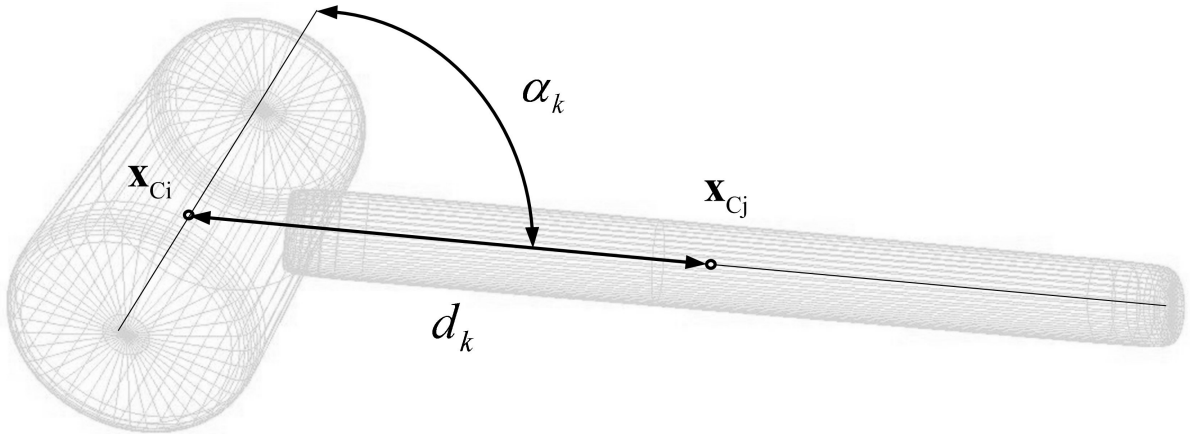


Figure 5.7: The main axis steradian α_k and the distance d_k between the Superquadric centers describe the connectivity of a composite object (rubber-mallet in Fig. 5.3(a)).

5.2 Experimental Evaluation

The experiments are carried out using two different laser range sensors to evaluate the behavior and the robustness of the detection method on real settings. Additionally a sensor

simulation has been used to evaluate the accuracy of the detection method considering the realistic one-view characteristic of the used sensors. This section starts with the introduction of the sensor setups describing the sensors and the simulator in detail. Then the evaluation method using real and synthetic range images is presented and compared. Experiments detecting part features or simple objects using a single Superquadric are as well conducted as experiments to analyze the detection of composite models using several linked Superquadrics. Concluding the experiments a comparable evaluation of the bore detection approach and the model-based detection using Superquadrics is given.

5.2.1 Sensor Setup

During the project FlexPaint two sensors from IVP have been purchased to realize the sensor cell for the part scanning (see Section 3.1). After the projects runtime these sensors were available for further research work¹. We developed a sensor cell to capture highly accurate colored range images and a mobile scanning head using a fast pan-tilt unit capable for mobile robotics use. In the course of these developments a simulation has also been realized to compute off-line tests and to be able to capture range images of not present parts. Details on these developments and settings for the experiments are given in the following sections.

Sensor Cell

The main motivation for the development of the sensor cell was to have a highly accurate laser range sensor for research purposes. An additional requirement was the feature to capture range images with additional texture information. The request was a robust sensor realized with low cost components compared to commercial solutions such as the Minolta VI-910 or the Riegl LMS-Z420i. The sensor cell should be easily operated and calibrated. For detailed information you are referenced to [78].

In the literature previous work can be found implementing low cost range sensors with color information. Two of these main contributions should be mentioned that are most related to our sensor cell implementation. In [48] a sensor is introduced that uses a similar ranger camera system from IVP as we do. The laser plane is projected onto the object with a rotating mirror. The ranger camera (MAPP2200) captures the same image as the color CCD camera using a semi-transparent mirror. This has the advantage that to each 3D raw data point, a corresponding color pixel can be associated. The disadvantage is the rather complex system architecture to synchronize the laser plane movement with the image capturing. The second implementation uses a xy -measurement table, a commercial scanner head and a color web-camera [109]. The sensor is calibrated with a calibration object that consists of circular markers and 3D half-spheres to align the web-camera and the ranger scanner coordinate system. In order to compute the correspondence between the 3D raw data points and the color information of the CCD pixels the line of sight method is used by function interpolation over many landmark sites [132]. This implementation is a truly good system design with the drawback of a long computation and a complex

¹Thanks to MAN Trucks Austria for providing the sensors at no charge.

5. OBJECT DETECTION

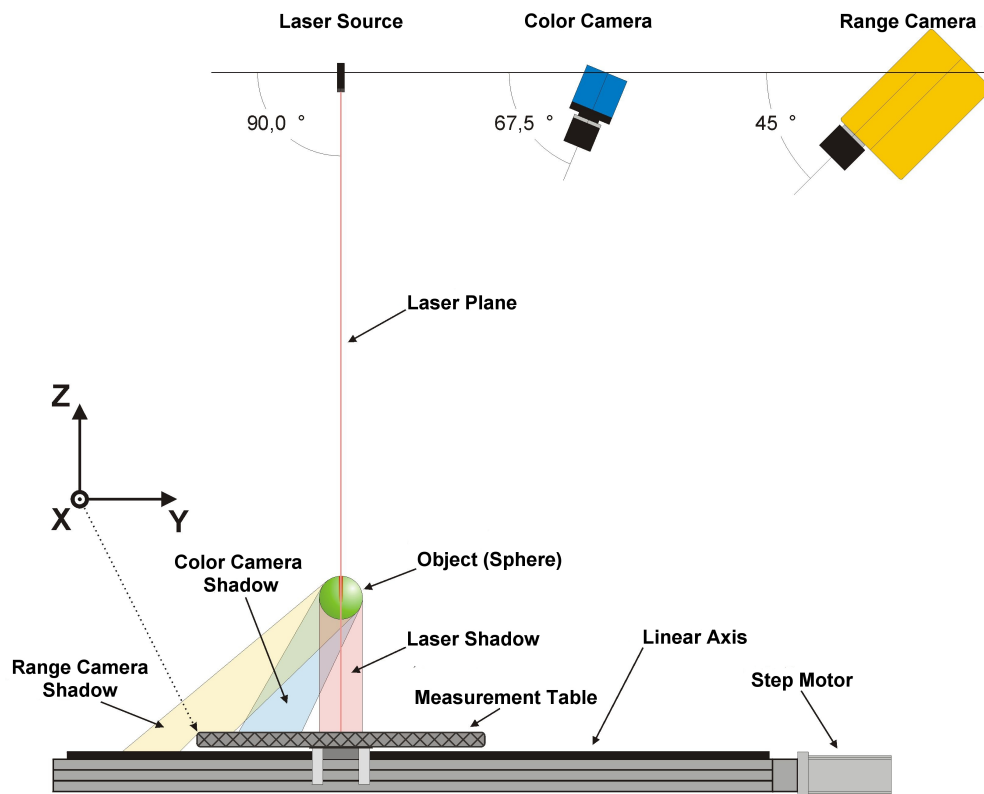


Figure 5.8: Drawing of the sensor cell design with its rough dimensions and labeling.

calibration method. Calculating the texture of 5,000 raw data points takes about 30 seconds.

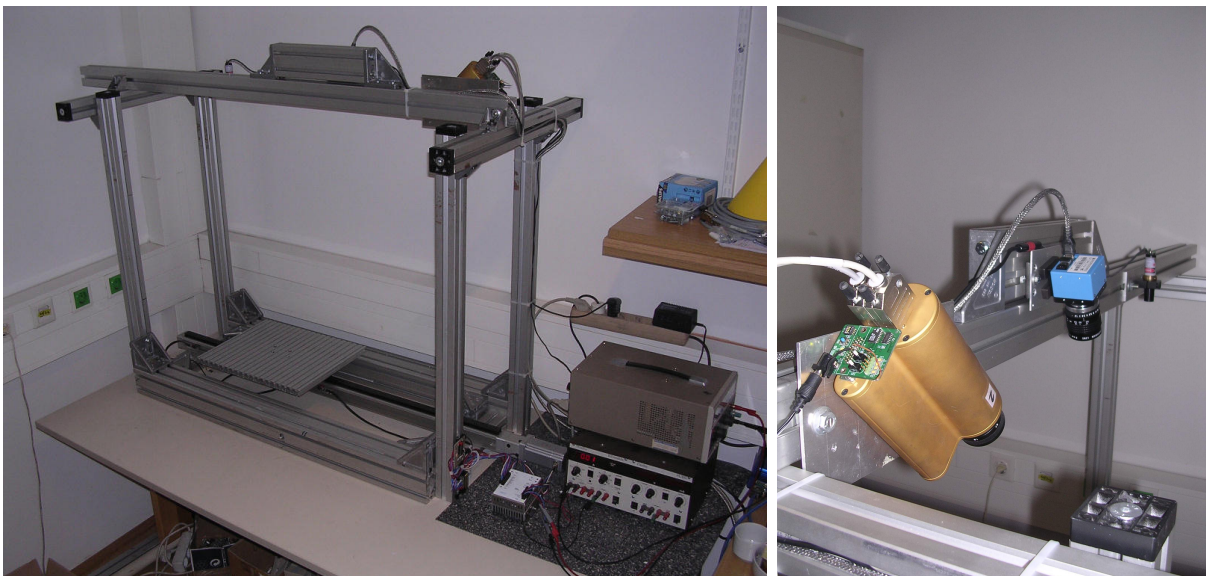


Figure 5.9: The sensor cell assembly is presented on the left image in overview and on the right image in detail showing the ranger camera, the color camera and the laser source.

Our contribution to the sensor design is the combination of the advantages of the above introduced sensors to achieve a robust color scanner with a fast processing and accurate result. Figure 5.8 illustrates the setup of the sensor cell we built. It consists of a linear axis (FESTO), a measurement table, a laser source (LASIRIS), a firewire color camera DFK 21F04 (The Imaging Source) and the MAPP2500 ranger camera (IVP). Figure 5.9 shows the real assembly of our sensor cell design.

For capturing accurately range images, a calibration of the laser scanner is needed applying the geometrical approach. Using a 3D calibration object with markers on at least two different planes the pose of the laser plane and the extrinsic parameters of the camera can be calculated as described in [48]. The color camera is calibrated with the tool *Camcalb*, introduced in [146]. This tool provides the intrinsic camera parameters in order to undistort the camera images for enhancement of the texture overlay and additionally gives the extrinsic parameters (position and orientation of the calibration plate) for fulfilling the last calibration step. The laser coordinate system and camera coordinate system are finally registered via transformation between the respective world coordinate systems. This leads to the possibility of transforming the raw data points obtained by the laser scanner into the coordinate system of the color camera using the line of sight method.

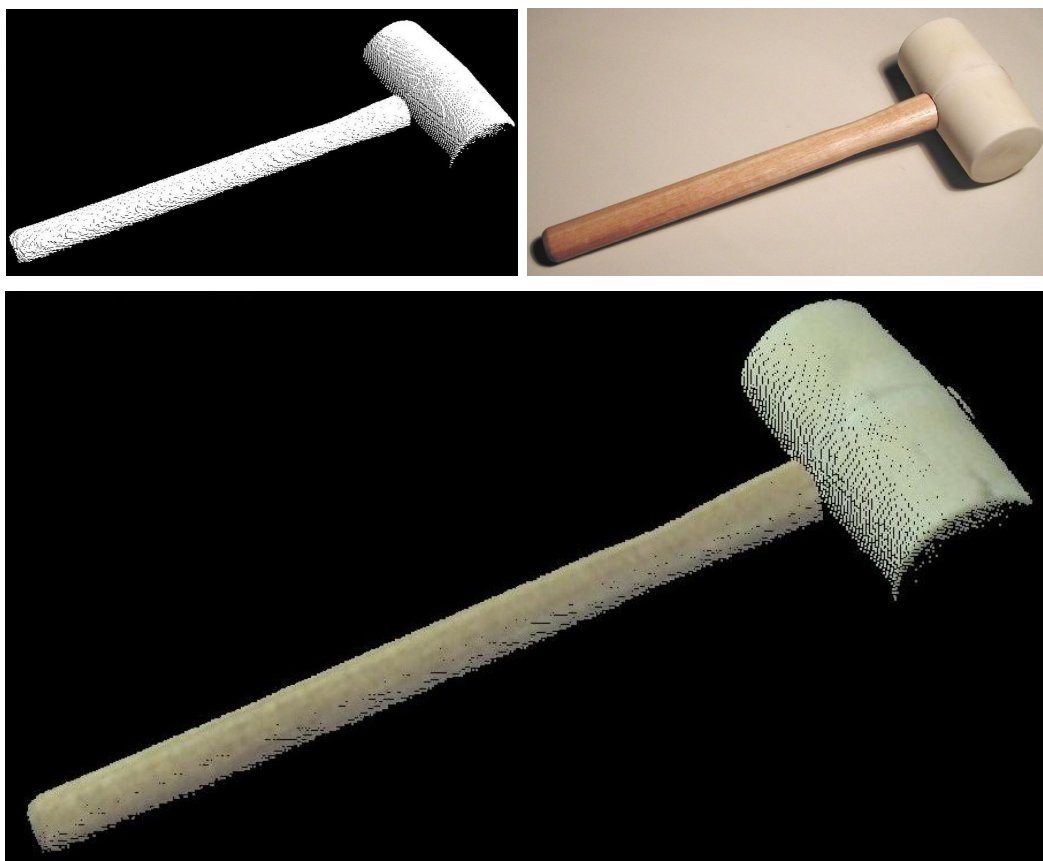


Figure 5.10: In the top row the processing steps of a scan are shown. First the range image is captured before the picture of the mallet is taken. After that the color texture is mapped onto the range image data (bottom picture).

The procedure to capture a range image is quite easy by placing an object on the measurement table. The maximum speed of the table is 6.3mm/s with a maximum resolution of 0.05mm (y -axis). Scanning the whole workspace (length, width, height: $400 \times 300 \times 300\text{mm}$) takes 63.5s . To compensate the color camera shadows up to 15 images are taken from the scene while the table moves back to its home position. The texture mapping of 170,000 points per second is very fast compared to the scanning process on a PC with 450MHz and 256MB RAM. The resolution in x - and z -direction is 0.5mm with an accuracy of $\pm 0.3\text{mm}$ [78]. Figure 5.10 demonstrates the scan result of a rubber-mallet consisting of 5,600 3D points.

Synthetic Range Image Generation

At the end of the project FlexPaint (see Section 3.1) and within the calibration task of the project Lot Size One (compare Section 3.2) the demand raised on a sensor simulation to obtain off-line range image data of any kind of object. This was necessary to push the *Lot Size One* research ahead, because most of the time a real sensor setup was not available. So the request was a sensor simulator that produces most realistic results for further processing.

The sensor simulator was designed to handle one laser plane and up to four cameras, all placed at an arbitrary pose. For the laser plane up to four laser sources can be used to avoid the unwanted laser shadows. Scan results are obtained performing a linear motion of the objects through the defined laser plane.

The sensor simulator consist of a design and a simulation part. As first step the visualization of the sensor cell design is important to get a fast estimation of the cameras' field of view related to the object and the pose of the laser plane. The cameras² can be set in 6 DoF as well as the laser plane and the laser sources. As example of the visualization Figure 5.11 shows the camera model, the sensor CCD-chip and the laser source with its local coordinate frames to place it in 3D.

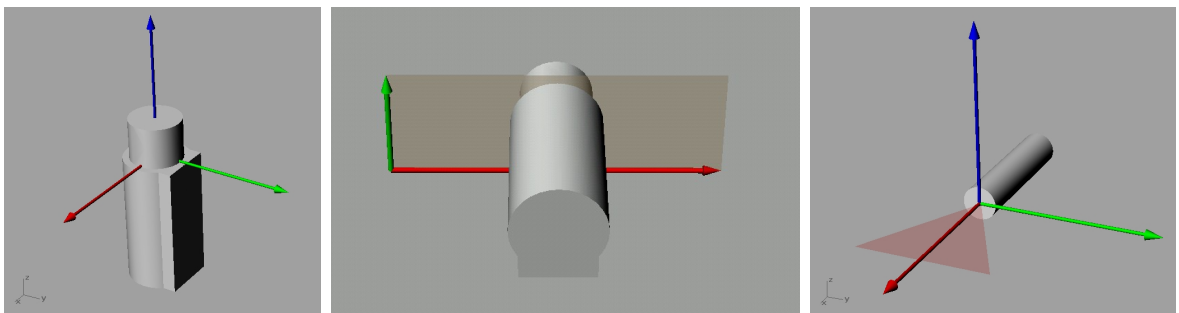


Figure 5.11: Visualization models of the camera, the CCD-chip and the laser source with its local coordinate frames (red: x -axis, green: y -axis, blue: z -axis).

The object scene must be previously modeled with a commercial CAD editing tool (e.g. AutoCAD, Rhinoceros) and must be exported in STL-Format. An example of a

²We simulated the properties of the M50 laser range scanner from IVP.

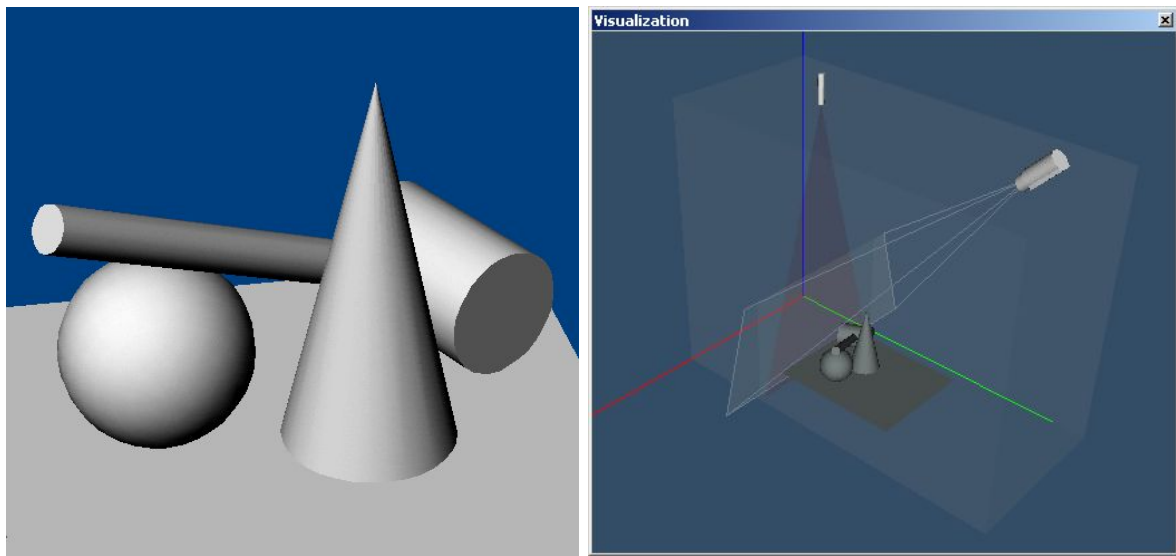


Figure 5.12: On the left the modeled object scene saved in the STL-format is shown. On the right this scene is visualized with the virtual sensor setup.

cluttered scene to be virtually scanned is shown in Figure 5.12 consisting of a sphere, a mallet and a cone. Figure 5.12 right shows a screen-shot of the modeled sensor setup of the static laser range scanner. Note the visualized laser plane with a fan angle of 30° and the camera's field of view on the laser plane. The synthetic range image is calculated by moving the objects through the laser plane parallel to the y -axis (green axis) of the work space.

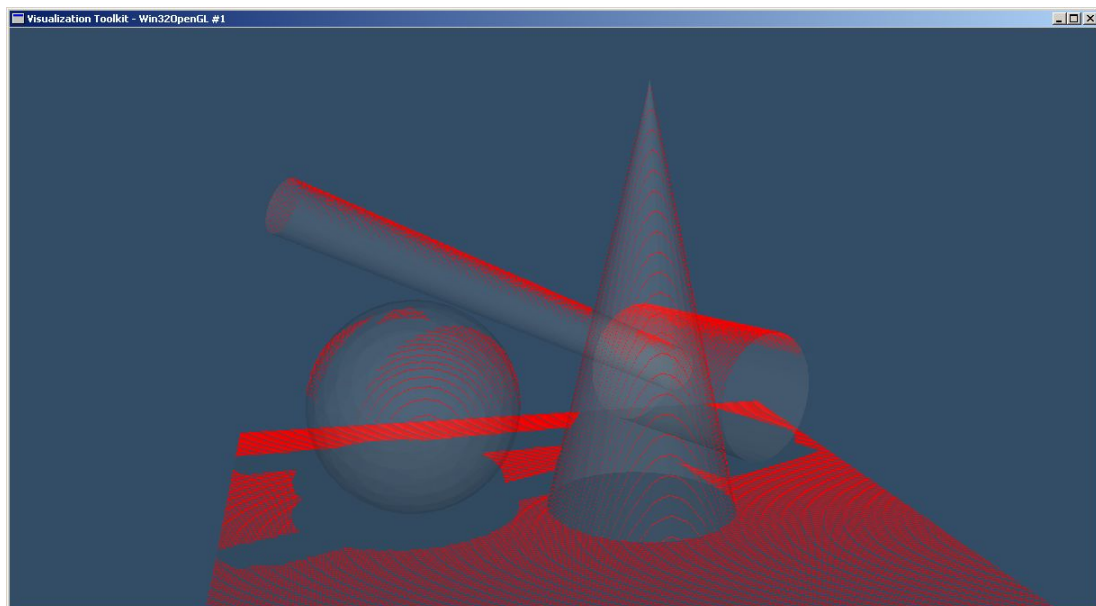


Figure 5.13: Synthetic range image of the modeled object scene in Figure 5.12. Note that the objects are considered transparent underneath.

After designing the virtual sensor setup and modeling the object scene the generation of the synthetic range image can be started. To achieve a most realistic simulation the geometry of the optical rays have to be implemented using the ray-tracing method. The shadows of the laser light are as well calculated as the self-occlusions due to the camera view. The quantization errors of the sensor chip (1536×512 pixels) are calculated for the given focal length. All internal algorithm parameter of the M50 camera, such as thresholds and the peak detection, are considered. The drawback of the simulation are the unimplemented lens distortions, the brightness of the laser light and the object's surface properties. Neglecting these constraints a perfect but realistic range image is computed without noise, apart from the quantization noise. Figure 5.13 shows the range image of the objects in Figure 5.12.

Mobile Scanning Head

The need of a fast laser range sensor usable for mobile robotics was the motivation to develop this scanner. The idea was to use a pan-tilt mounted sensor to capture a range image by sweeping the laser plane across the scene. Additional features of capturing gray-scale images and the ability to map gray-scale information on the range image data makes this sensor useful for a lot of robotics and image processing tasks. Details on the implementation can be found in [90].

Many sensors have been realized in recent years tackling the problem to develop fast and flexible laser range scanners to be used in the field of mobile robotics. Three developments of the last years should be picked out that are closely related to our mobile scanning head approach. In [128] the authors introduced a stereoscopic laser range scanner for a humanoid upper torso robot. They also used a pan-tilt head to sweep the laser plane across the scene but use two color cameras for depth measurement and texture capturing. The novelty of this sensor is the approach how to reduce sensor noise, cross talk and spurious specular reflections using two cameras. The drawback of this robust solution is the complexity of the system and the high computational effort to increase the accuracy and the range image quality. In contrast to the previous sensor [119] presents a simple hand-held laser scanner which can also be mounted on a pan-tilt head and used for robotic applications. This sensor is characterized by a simple design, a fast measurement and a novel geometric calibration method using the raw data of several scans of a plane without markers. But the fastest sensor sweeping a laser plane over the scene to capture a range image is introduced in [20]. The benefit of this design is the on-board depth processing on the CCD-chip in cooperation of a synchronized rotating laser plane. This novel approach achieves capturing rates of up to 50 range images per second. The advantage of the sensor's real time property weakens the range image resolution and accuracy.

Our contribution is to find the compromise between fast range image capturing and a high resolution and accuracy on the one hand and a simple and cheap concept and design on the other hand. Additional features of capturing gray-level images with a corresponding pose and the ability to capture the texture of a range image within a second sweep over the scene turn this development into a powerful sensor for the application to scan table top scenes for robotic applications. Figure 5.14 shows the flexible range image sensor with

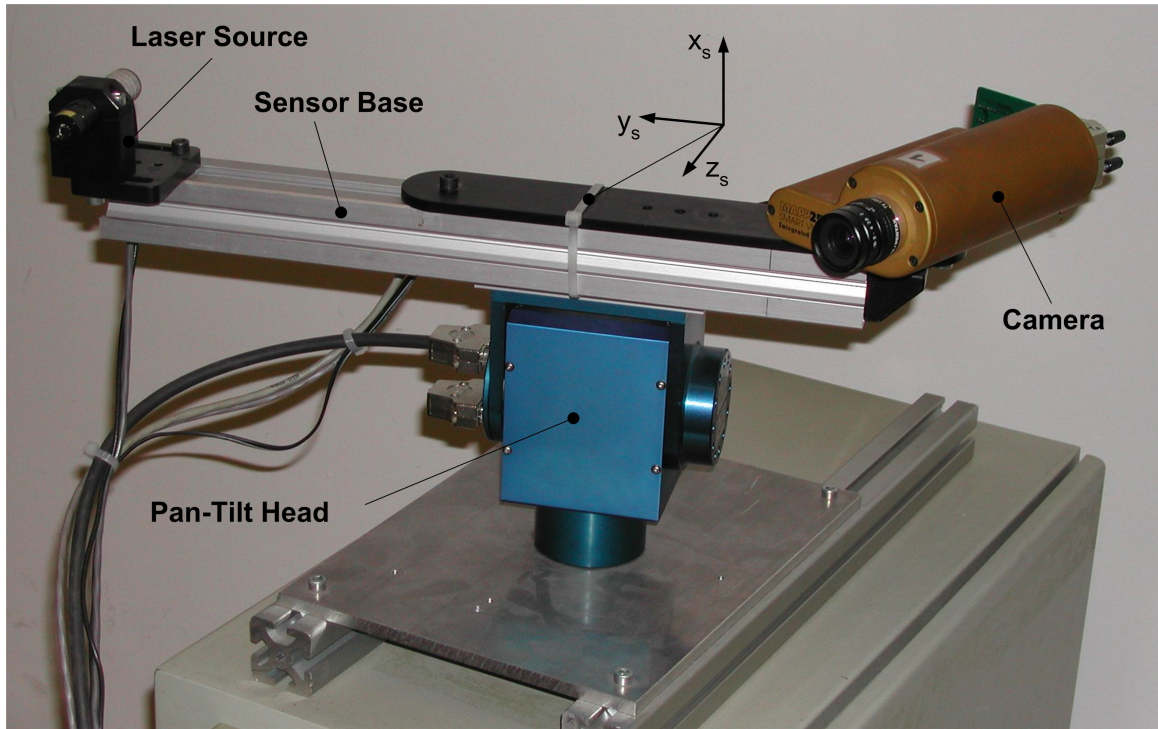


Figure 5.14: Setup of the mobile scanning head development consisting of the pan-tilt head, the laser source, the sensor base and the camera.

its components.

Calibration of a laser range scanner is a necessary but well investigated procedure. There are basically two approaches for calibration using geometric objects and their properties or using markers in at least two different planes on a calibration object. As an example in the wide range of publications [106] shows that a laser scanner of a mobile robot can be calibrated using a cube placed in a room corner. Using the 3D coordinates of the seven visible cube corners the system can be calibrated by capturing a range image. In contrast to that [83] demonstrates the calibration of a laser scanner with marker on a cube. The procedure is more complex than [106] but results in an accurate calibration because of compensating the radial lens distortions. For our purpose it was sufficient to use the calibration described in [48], a simple combination of the geometrical approach and markers but disregarding the lens distortions.

To keep the costs low we used industrial standard components to build the scanning head. The pan-tilt head is from AMTEC, the laser source from LASIRIS and the MAPP2500 camera from IVP (compare Figure 5.14). In Figure 5.15 the schematic assembly of the mobile scanning head is shown. Varying the baseline $B = B_1 \approx B_2$ and the angle α of the camera the dimensions of the workspace can be influenced [90]. The setup $\alpha = 60^\circ$ and $B = 425mm$ results in a workspace with a minimum distance of $A_1 = 344mm$ and in a maximum of $A_2 = 2685mm$. This is sufficient to scan table top scenes as well as the near neighborhood of the mobile robot. The capturing rate of the camera is limited by the integration time and typical values are up to 100 profiles a second at a rotation

speed of up to $360^\circ/s$ of the pan-tilt head. Each profile consists of 512 points.

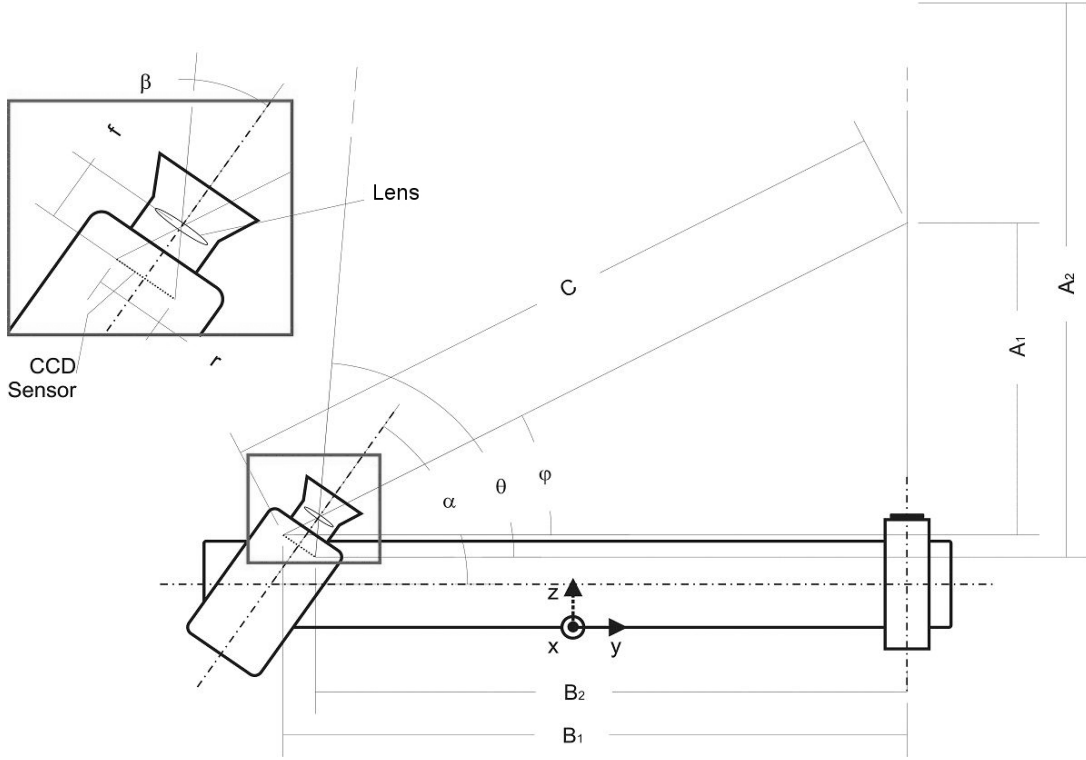


Figure 5.15: Schematic drawing of the mobile scanning head design.

The way of capturing a range image is quit simply by sweeping the laser plane over the scene and capturing as many profiles as possible. If a textured range image is required a second sweep in reverse direction is performed. From all, with the laser illuminated pixel on the cameras CCD-chip, the gray-value is taken and mapped into the range image. Because of the fan like nature of the profiles due to the rotation a transformation into a world coordinate system must be applied. Figure 5.16 shows the transformation from the local sensor coordinate system (x_s, y_s, z_s) into the world coordinate system (x_w, y_w, z_w) . The mathematical description transforming a point \mathbf{x}_s into \mathbf{x}_w is given by

$$\mathbf{x}_w = \mathbf{R}_x \cdot \mathbf{R}_y \cdot (\mathbf{x}_s + \mathbf{t}_s) + \mathbf{t}_w \quad (5.7)$$

where \mathbf{R}_x is the pan rotation

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}, \quad (5.8)$$

\mathbf{R}_y the tilt rotation

$$\mathbf{R}_y = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (5.9)$$

and \mathbf{t}_s and \mathbf{t}_w are the translation vectors from the sensor respectively to the world coordinate system.

$$\mathbf{t}_s = \begin{bmatrix} x_{ts} \\ y_{ts} \\ z_{ts} \end{bmatrix}, \quad \mathbf{t}_w = \begin{bmatrix} x_{tw} \\ y_{tw} \\ z_{tw} \end{bmatrix} \quad (5.10)$$

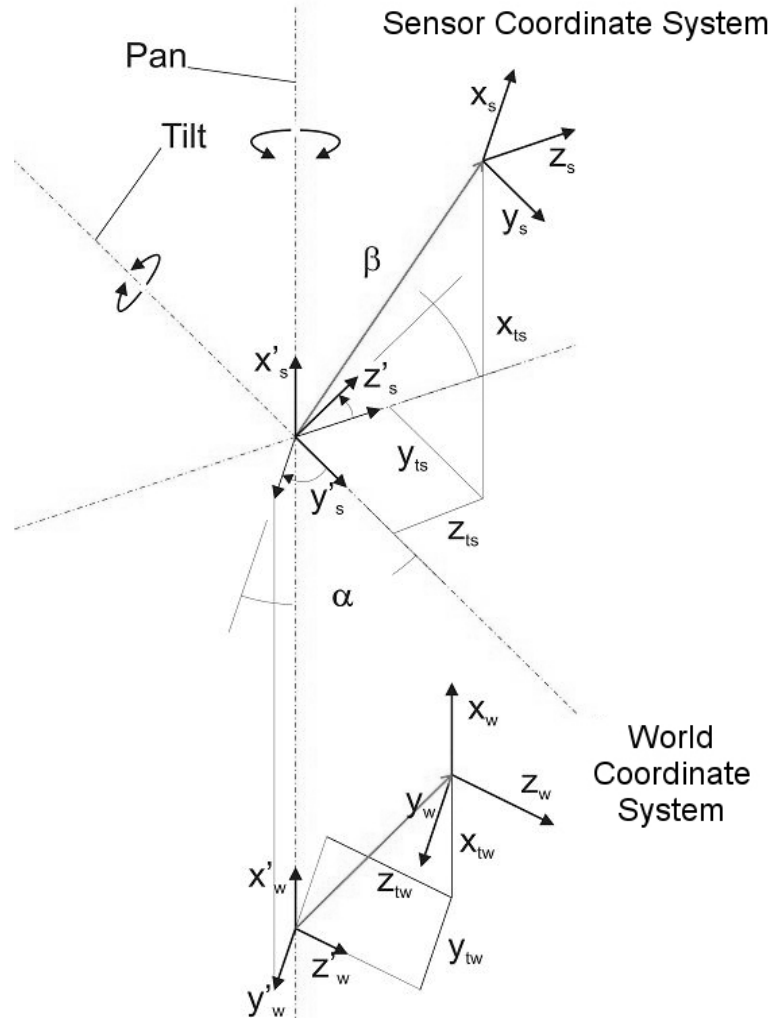


Figure 5.16: Diagram showing the transformation between the sensor and the world coordinate system.

Figure 5.17 illustrates the result of a scan transformed to world coordinates. This range image shows a typical table top scene consisting of a plate with cookies and a knife, a tea pot and a mug. The range image was captured within a scan angle of 50° and with an integration time of $17ms$. The rotation speed was set to $10^\circ/s$ which results in a range image of approximately 90,000 raw data points. The resolution in x -direction (compare Figure 5.15) is in the range of $1.0mm$ to $3.5mm$ across the workspace. In z -direction the resolution is 1.5 to $7.0mm$, whereas the resolution in y -direction is dependent on the rotation speed with a lower bound of $0.1mm$.

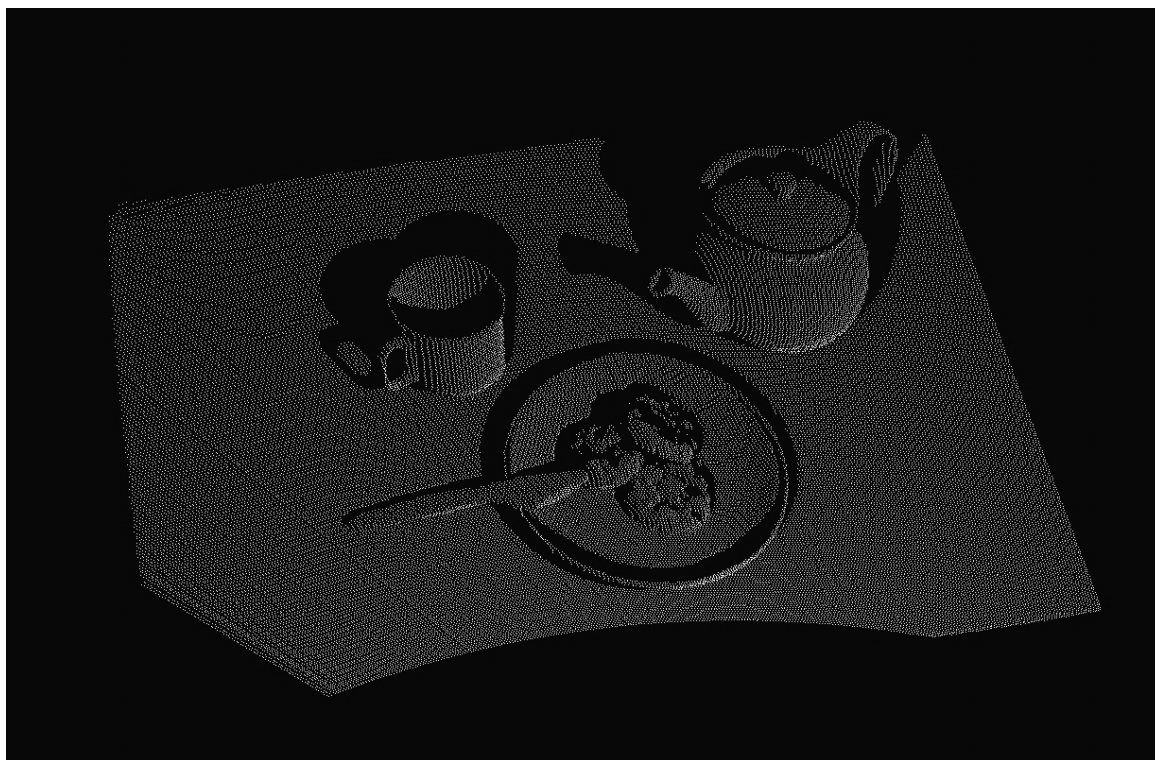


Figure 5.17: Range image showing a typical table top scene consisting of a plate with cookies, a knife, a tea pot and a mug.

5.2.2 Simple Model Description

This section starts the experimental evaluation of the proposed method using a single Superquadric for model description. First the performance of the approach is investigated in terms of the detection accuracy as well as the hierarchical approach is evaluated. Then the behaviour of the method concerning the impact of noise is investigated and finally detection results of daily life objects are presented.

The following section focuses the evaluation on the detection of composite models using several Superquadrics exploiting their relationships. The last section is an attempt to bridge the gap between the introduced methods of Chapter 4 and 5 giving an interesting experimental evaluation comparing the performances of the fast bore detection approach versus the model-based detection method applied on selected industrial parts of the FibreScope project.

If nothing else is stated, the standard algorithm parameters are set as followed: Calculating the initial pose for each Superquadric fit a sub-sample of 30 data points is used applying 10 minimization iterations. Finding one pose hypothesis, 10 RANSAC trials with each 50 data points and 20 Levenberg-Marquardt iterations and a final refinement on full resolution data is applied. The tolerance factor γ defining the quality of fit criteria (compare Equation 5.3 and 5.4) is set to 20%.

Evaluation and Detection Results

For a serious evaluation of the method, investigating the detection accuracy and the computational effort, the detection results must be compared to ground truth data. Obtaining ground truth information from real scanned objects is usually difficult and incorrect. Hence, a range image is generated with the simulation tool described in Section 5.2.1. To achieve precise detection results a set of geometric primitives is virtually scanned. Figure 5.18 shows an arrangement of a cylinder, a box, a sphere and a cone and the generated one-view synthetic range image consisting of 141,485 raw data points. Table 5.2 and 5.3 list the size and shape parameters of the Superquadric models as well as the ground truth position and orientation of the geometric primitives of Figure 5.18.

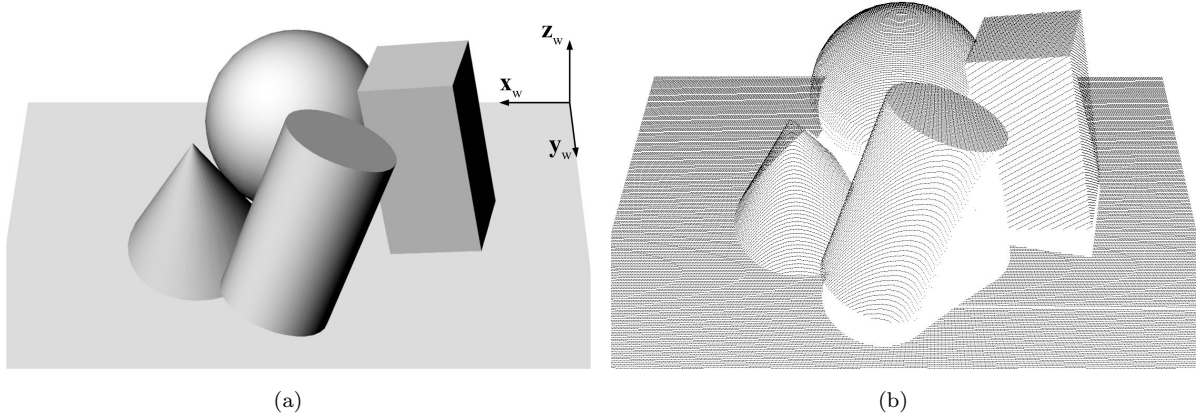


Figure 5.18: (a) Experimental setup of a scene consisting of a cylinder, a box, a sphere and a cone. (b) Virtual one-view scan of the geometric primitives.

Object	a_1	a_2	a_3	ε_1	ε_2	k_x	k_y
Cylinder	30.0mm	30.0mm	50.0mm	0.1	1.0	0.0	0.0
Box	25.0mm	25.0mm	50.0mm	0.1	0.1	0.0	0.0
Sphere	50.0mm	50.0mm	50.0mm	1.0	1.0	0.0	0.0
Cone	20.0mm	20.0mm	40.0mm	0.1	1.0	-1.0	-1.0

Table 5.2: Size and shape parameters of the geometric primitives in Figure 5.18.

In order to analyze the results a criteria for a true positive detection has to be defined. In our experimental setup, a pose deviation of 3mm in 3D space is allowed for a true positive detection. Figure 5.19 demonstrates some of the correct detection results of the primitives' defined in Table 5.2. In these experiment the method has been tested with 100 runs for each of the four models.

The first investigation evaluates the percentage of the true positive detections and the corresponding computational effort. All tests have been carried out with $n = 10$ RANSAC trials and 20 iterations for each Levenberg-Marquardt minimization. The number m (see Figure 5.2) has been varied from 1 to 25 hypotheses to demonstrate that a

5. OBJECT DETECTION

Object	p_x	p_y	p_z	v_x	v_y	v_z
Cylinder	153.646mm	146.348mm	58.275mm	-0.35028	-0.33886	0.87319
Box	88.923mm	94.128mm	59.173mm	0.18692	-0.29300	0.93767
Sphere	160.0000mm	65.000mm	50.000mm	-	-	-
Cone	213.074mm	110.4380mm	40.000mm	0.00000	0.00000	1.00000

Table 5.3: Ground truth pose parameter of the primitives in Figure 5.18. \mathbf{p} denotes the Superquadric's center and \mathbf{v} the Superquadric's z -axis vector.

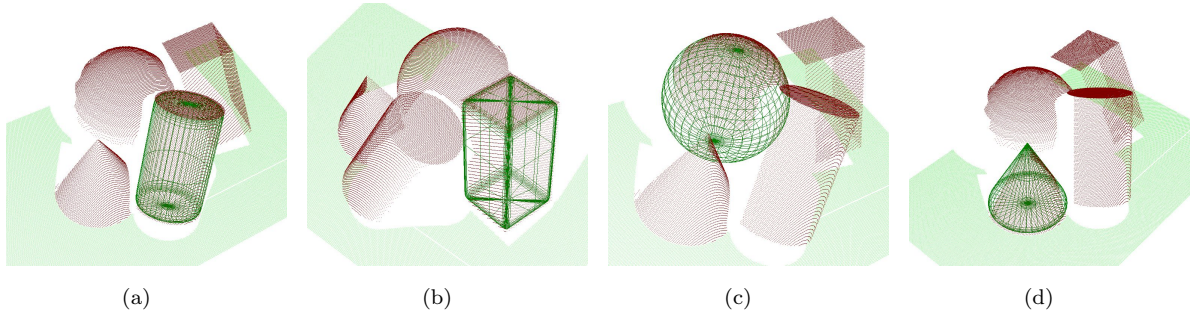


Figure 5.19: Examples of true positive detections of the four geometric primitives.

RANSAC-based approach finds the correct solution – it is only a matter of the amount of trials. Figure 5.20(a) illustrates the true positive detections faced against the number of calculated hypotheses.

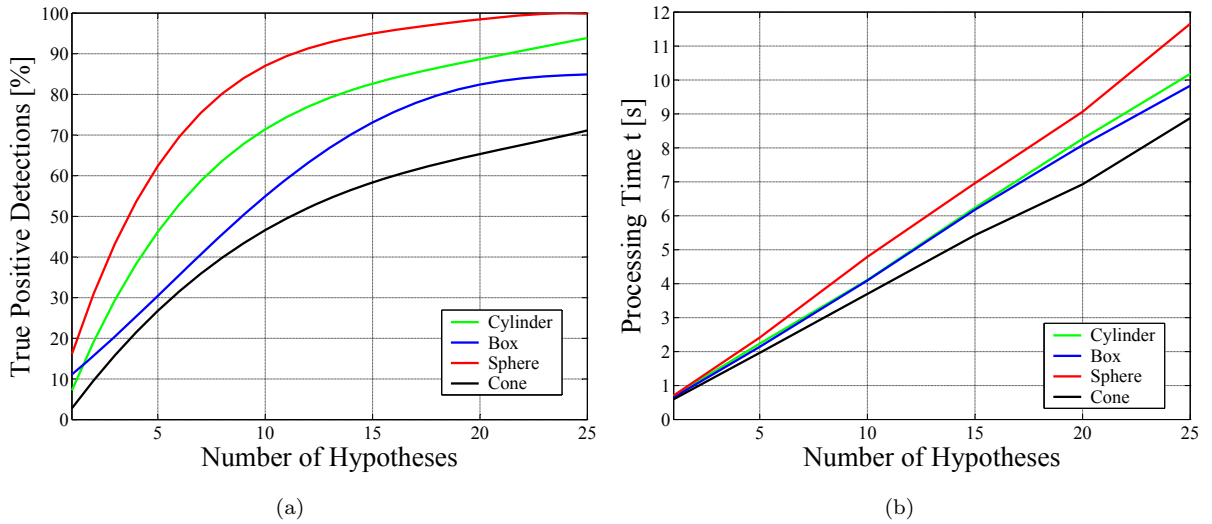


Figure 5.20: (a) Percentage of true positive detections faced to the number of hypotheses and (b) the averaged corresponding computational effort.

What clearly can be seen is, that the detection process gets more robust with the rising number of verified pose hypotheses. The sphere, as the simplest geometric primitive, achieves the most robust results and reaches 100% true positive detections with

22 hypotheses. It can be observed that the true positive results with an increased model complexity go down. Also interesting is the correlation between detection robustness and the averaged processing time, shown in Figure 5.20(b). The reason being is, the lower the model complexity the more pose refinement steps have to be executed, because of the shape-pose ambiguities. On the other hand, the higher the model complexity the less refinement steps are required to obtain no further pose improvement, which result in a higher pose uncertainty. That leads to a lower detection robustness, a behavior that can be clearly observed with the cone experiment, where additionally global deformation parameter are needed for the model description. The rather high computational effort is due to the dense raw data points usually not obtained from the scanner described in Section 5.2.1.

The criteria of a true positive detection in Figure 5.20 is only of qualitative nature. The next question is, how accurate the true positive detections are. The detected model-pose is made up of an orientation and a position vector. The error of the model's main axis is shown in Figure 5.21 denoted with the solid angle α , which gives the absolute deviation between the ground truth vector \mathbf{v} and the recovered object's main axis. The evaluation of the true positive results with $m = 25$ hypotheses is shown in this figure. Using a histogram the spreading of the true positive detection results can be evaluated. Note, that the sphere has no orientation, hence, the sphere histogram is missing.

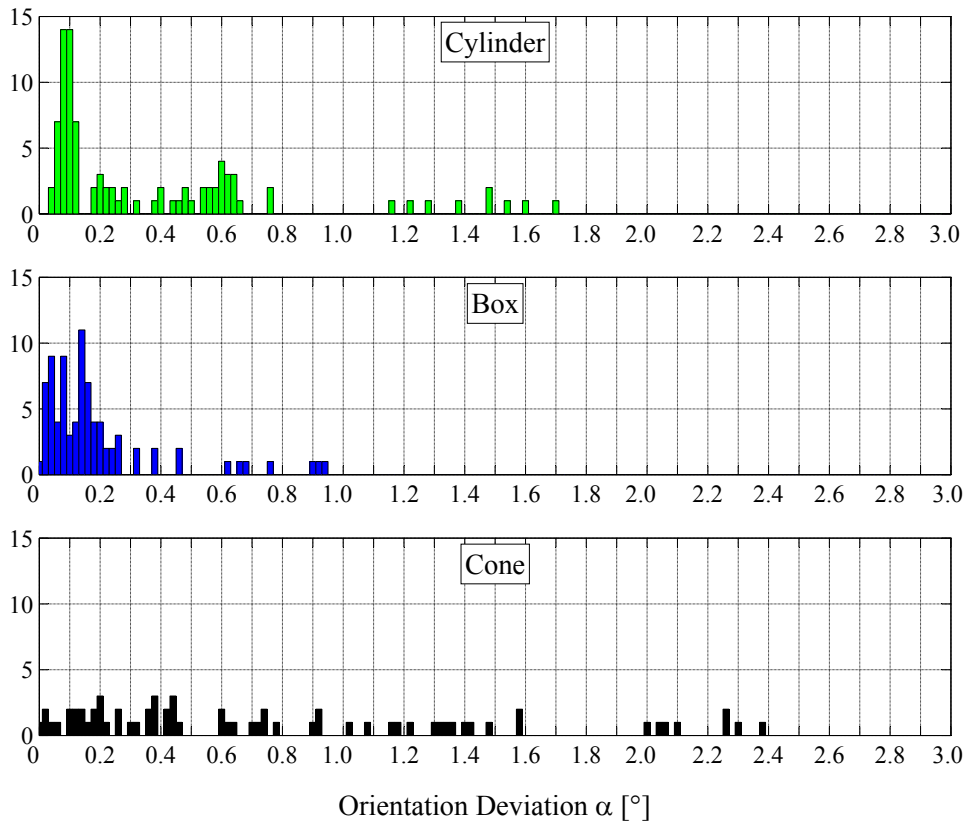


Figure 5.21: Histogram of the absolute orientation deviation of the detection results compared to the ground truth orientation.

The experiments with the cylinder and the box shows a significant peak between 0 and 0.2 degrees, where most of the detection results are located. The orientation deviation of the cone experiment is spread up to 2.4 degrees which indicates that the detection of global deformed models is less accurate than of solid Superquadric primitives. This observation can be verified investigating the position accuracy of the four models. Figure 5.22 presents the positioning errors in the xy - and yz -plane. As expected, the tests verified a high detection accuracy for the cylinder and the box and a little spreading of the cone's position errors. Surprising is the sphere experiment with widely and equally distributed results, but with no outliers. The reason being is that the result of the pose is determined in 6 DoF, but the orientation can not be fixed within the minimization process. This orientation drift influences the position of the sphere while fitting.

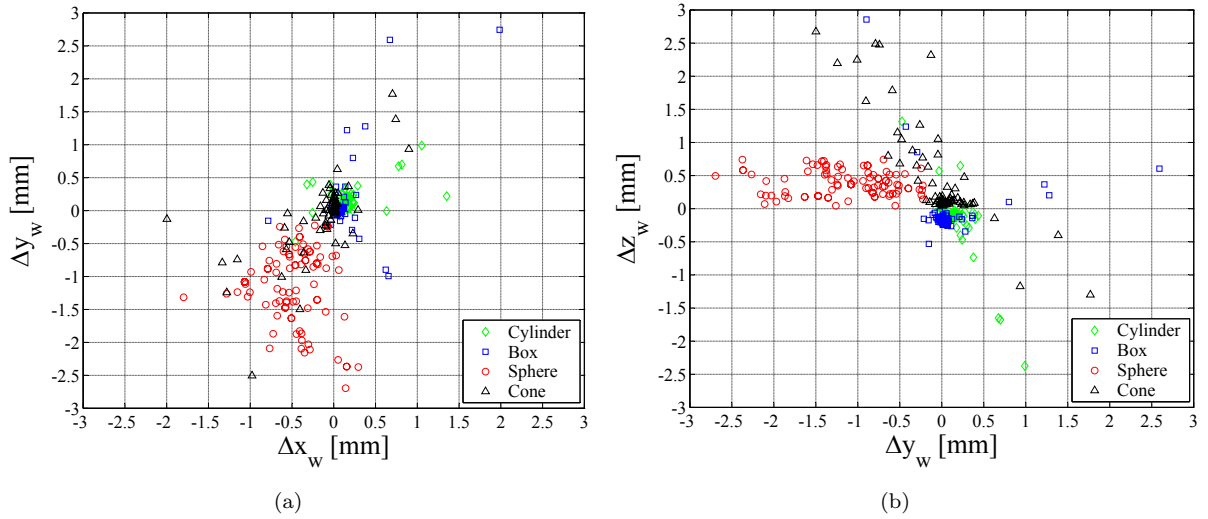


Figure 5.22: Plot of the absolute position deviation of the true positive results compared to the ground truth position.

Object	\bar{x}	\bar{y}	\bar{z}	$\text{std}(d)$	\bar{d}
Cylinder	0.135mm	0.128mm	-0.132mm	0.414mm	0.242mm
Box	0.079mm	0.077mm	-0.088mm	0.475mm	0.244mm
Sphere	-0.495mm	-1.124mm	0.385mm	0.296mm	0.622mm
Cone	-0.112mm	-0.040mm	0.428mm	0.667mm	0.507mm

Table 5.4: Summarized results of Figure 5.22 given the mean of the position deviation and the distance d and the standard deviation of the distance d .

The main parameters of the position experiment are summarized in Table 5.4, with the barycenters of the detection results \bar{x} , \bar{y} and \bar{z} . To obtain a measure of degree of the results spreading the distance d to each result from the barycenter is calculated.

$$d = \sqrt{(\bar{x} - \Delta x_{w_i})^2 + (\bar{y} - \Delta y_{w_i})^2 + (\bar{z} - \Delta z_{w_i})^2} \quad (5.11)$$

The standard deviation and the mean of the distance d are also expressed in Table 5.4. In Figure 5.23 the distribution of the distance d is shown using as well a histogram. Here, the good detection accuracy of the cylinder and the box can clearly be seen. The cone's detection accuracy is a little bit worse with some outliers. The larger distribution of the sphere's results can also be well seen.

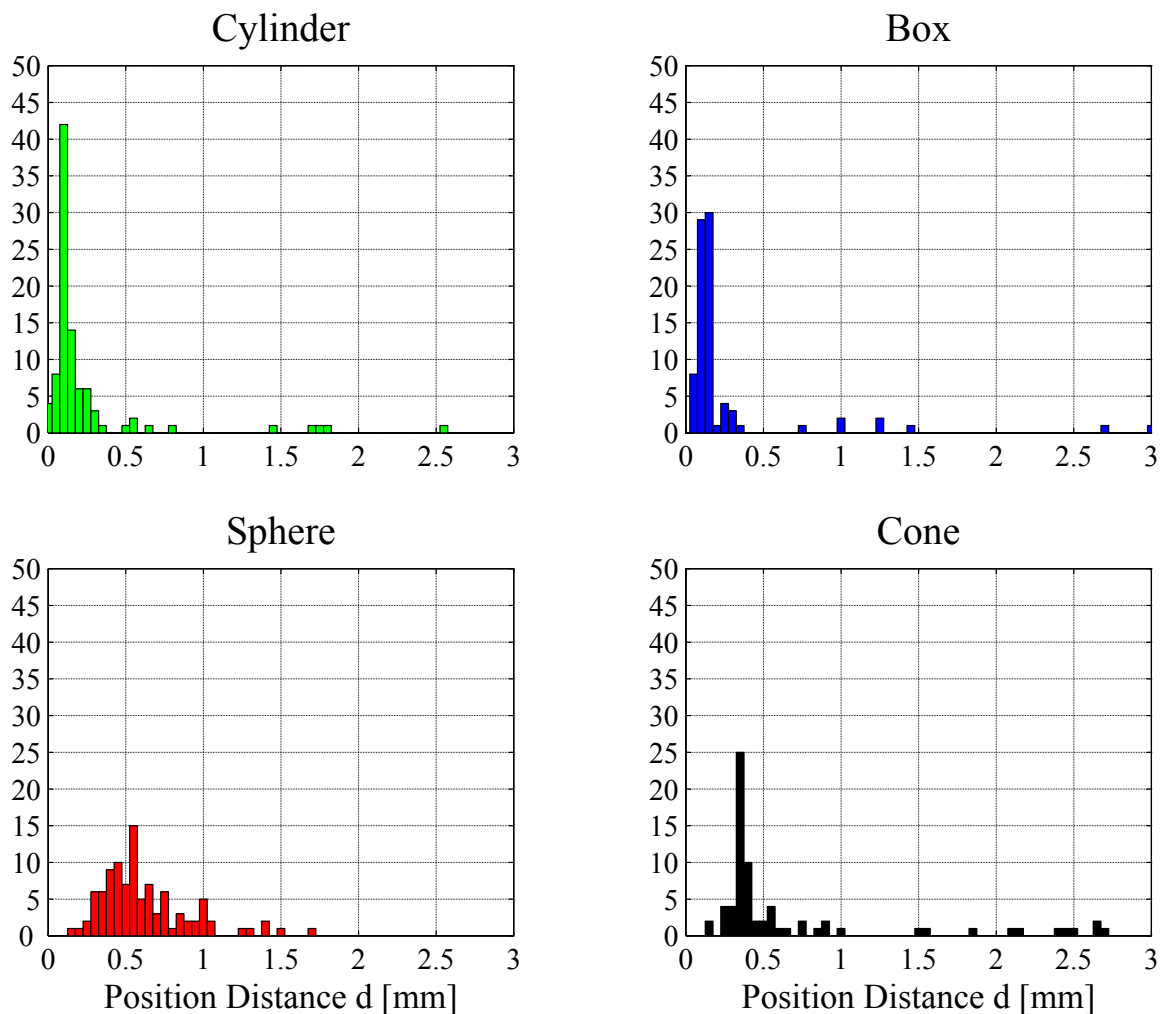


Figure 5.23: Histogram of the distance d distribution.

Hierarchical Approach Evaluation

As previously mentioned, the classical RANSAC algorithm is a robust and reliable method for object detection. In Section 5.1.2 it is asserted that a hierarchical search structure achieves a better performance in less time than with the same amount of simple RANSAC trials. This claim should be evaluated in this section.

To keep the comparability to the previous experiments, the sphere and the cylinder model (see Figure 5.18(a)) are used to evaluate the RANSAC-based hierarchical structure. The detection behavior is analyzed using four different methodically approaches. Each

method is based on 100 random searches with different criteria to determine the best fit. For each Levenberg-Marquardt minimization 20 iterations are calculated. In the following, the properties of the methods are briefly described.

Method A: Here the classical RANSAC method is implemented calculating 100 random samples. The fit with the best measure of fit criteria (5.2) is then used for a final refinement step to improve the object's fitting.

Method B: This method is again a variant of the RANSAC approach. The difference is, that after each random sample fit a refinement step on full resolution data is applied. The criteria for the best fit result is again the measure of fit.

Method C: This method is the one which is described in Section 5.1. Here the hierarchical approach verifies 10 pose hypotheses using the criteria (5.3) and (5.4). Each hypothesis is found with 10 RANSAC trials using the criteria (5.2).

Method D: The last method generates 100 hypotheses with one random sample and a refinement step. For the pose verification all three criteria (5.2), (5.3) and (5.4) are evaluated using voting.

Figures 5.24 and 5.25 show the relative position deviation of the detection results in 3D-space. The pure RANSAC approach (method A) indicates a very poor performance in both experiments. Evaluating the sphere experiment, all other methods (B, C and D) achieve mainly true positive results. Considering the cylinder experiment, the results show that the methods B and C have a quite high amount of true negative detections, whereas method D performs very well. Investigating Figure 5.25 in detail one can see a very interesting phenomenon, which occurs mainly on cylinder- and box-like objects. The object is detected with the correct orientation but with a shift along the Superquadric main-axis. This effect can be very well observed with the method B where a lot of detections are located along the cylinder axis. Figure 5.26 visualizes in fact two examples of correct detections, but with a position shift. These results occur rather often when using exclusively the measure of fit criteria and that is exactly the reason why this phenomena can mainly be observed with method B. Because of introducing two additional criteria, the number of points on the Superquadrics surface and the number of points inside the Superquadric, this effect is under control and increases the detection robustness significantly. From now on these detections are called false negative detections, if the position is located on the main-axis and the orientation deviates not more than 2 degrees (see Figure 5.26).

Also of interest is the processing time for each method. In Figure 5.27 the method performance in terms of true positive detections ($d < 3mm$ of Equation 5.11) is faced against the processing time. Interpreting these diagrams three conclusions can be made. First, the RANSAC approach is fast but is only suited for a pre-selection of potential pose hypotheses candidates due to inaccurate results. Second, the refinement steps are very time consuming especially on dense range data, because they are calculated on full resolution. And third, the voting process with additional quality of fit criteria makes the detection robust. Hence, a hierarchically structured search algorithm with a low-level RANSAC search and a high-level pose hypotheses verification achieves the best detection

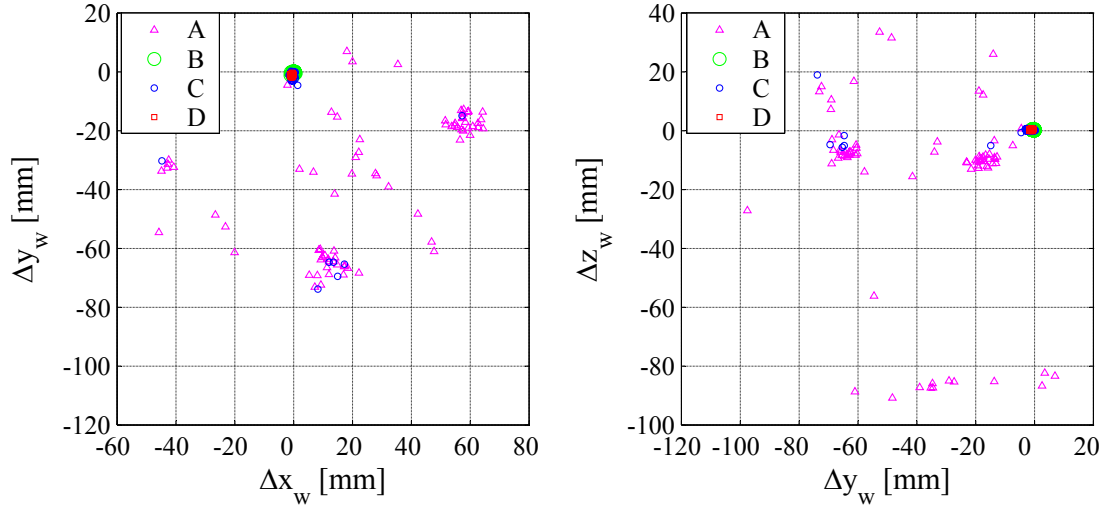


Figure 5.24: Relative position of the sphere detection results shown in the xy - and yz -plane using the methods A–D.

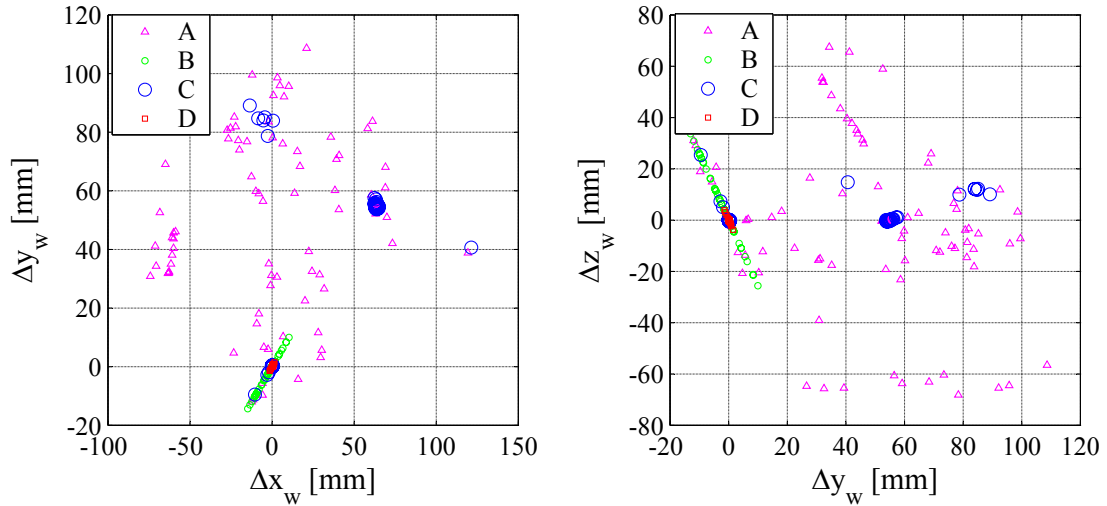


Figure 5.25: Relative position of the cylinder detection results shown in the xy - and yz -plane using the methods A–D.

performance. Method C is a good trade-off between the true positive detection rate and processing time. Of course, these results can be improved by tuning the number of RANSAC trials and hypothesis, depending on the application whether fast or robust results are needed.

Table 5.5 summarizes the detection results of the sphere and cylinder experiment by listing the true positive (TP), the true negative (TN), the false negative (FN) and the processing time. The maximum shift of all false negative results is 39.5mm along the cylinder axis with a length of 100mm . In the case that visual servoed grasping application can compensate this position uncertainty (the orientation is still correct) method B and D achieve a 100% detection rate.

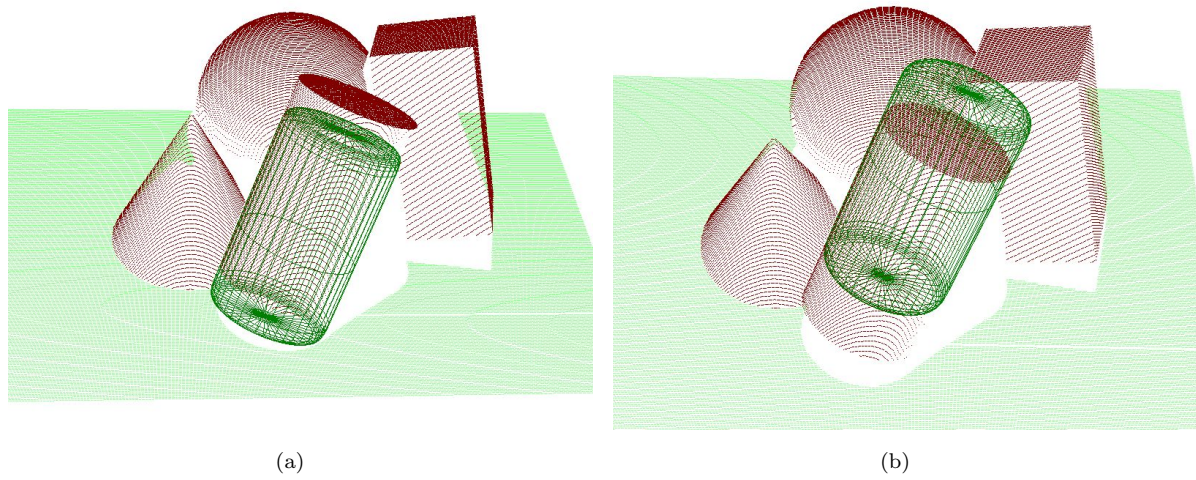


Figure 5.26: Inaccurate cylinder detections classified as false negative detections using the RANSAC method with refinement (B).

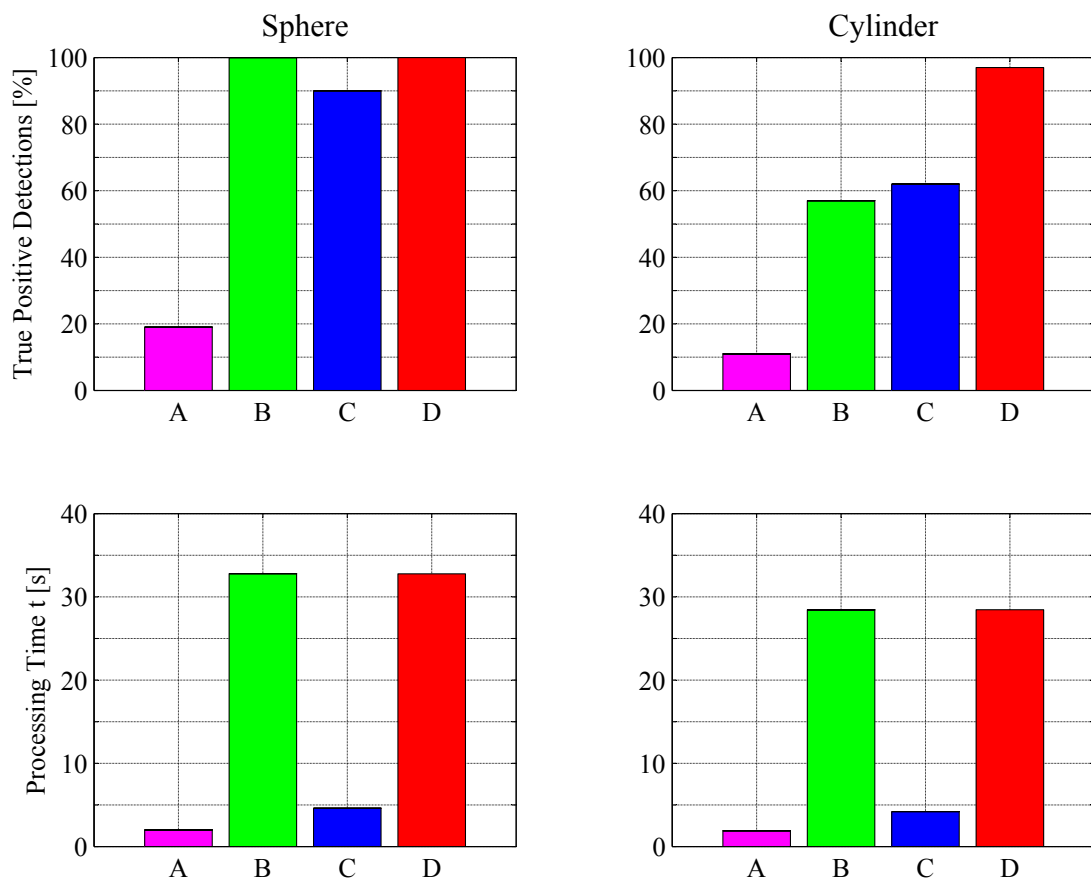


Figure 5.27: True positive detection rates and the processing time shown for the sphere and the cylinder model comparing the methods A–D.

Method	Sphere				Cylinder			
	TP	TN	FN	time	TP	TN	FN	time
A	19%	81%	0%	1.989s	11%	86%	3%	1.904s
B	100%	0%	0%	32.792s	57%	0%	43%	28.441s
C	90%	10%	0%	4.607	62%	35%	3%	4.197s
D	100%	0%	0%	32.767s	97%	0%	3%	28.457s

Table 5.5: Summarized results of the detection results testing the methods A–D including the averaged processing time.

Detection Behavior on Noisy Range Data

In this section the detection behavior in noisy raw data will be evaluated in terms of true positive results and investigating different noise levels. Here again, synthetic range data is used to be able to compare the results with ground truth information.

The first experiment investigates three similar spheres, with radii of $48mm$, $50mm$ and $52mm$, spacing each other with $10mm$ (see Figure 5.28(a)). The middle sphere with a radius of $50mm$ should be detected, with 10 calculated hypotheses and 20 iterations for each Levenberg-Marquardt minimization. A true positive detection is quoted when the position deviation in x , y and z -direction is less than $1mm$. In order to evaluate the noise behavior Gaussian noise is added in x , y and z -direction to the raw data points with a zero mean value and a standard deviation σ . Figure 5.28(b) and 5.28(c) shows the raw data points with a noise level of $\sigma = 0.3mm$ respectively $\sigma = 3mm$ each with a true positive detection result.

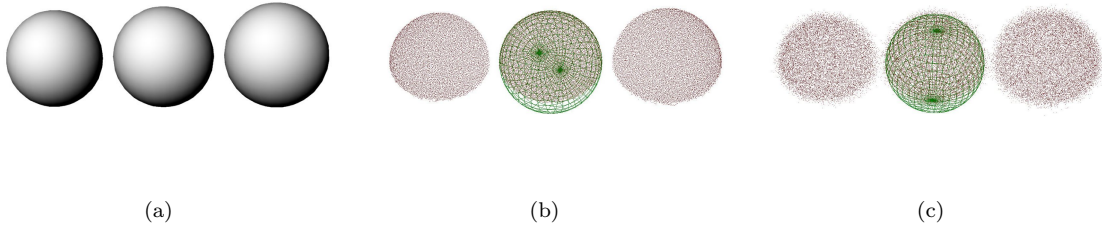


Figure 5.28: (a) Three spheres are presented with radii of $48mm$, $50mm$ and $52mm$. (b),(c) Added Gaussian noise with $\sigma = 0.3mm$ and $\sigma = 3mm$ presenting a true positive detection result.

Figure 5.29 shows the true positive detections by adding Gaussian noise ranging from $\sigma = 0.001mm$ to $\sigma = 100mm$. In face of having de facto no noise the hit probability of a true positive detection result is up to 95%. Furthermore, it can be seen that up to a noise level of $\sigma = 0.1mm$ the results are satisfactory. From this point on the noise weakens the uniqueness of the shape where the sphere radius is differing by $2mm$. Note that the method degrades gracefully.

The attention should now be focused on the scene presented previously in Figure 5.18. Here, a sphere with a radius of $50mm$ is also arranged among other primitives. If the

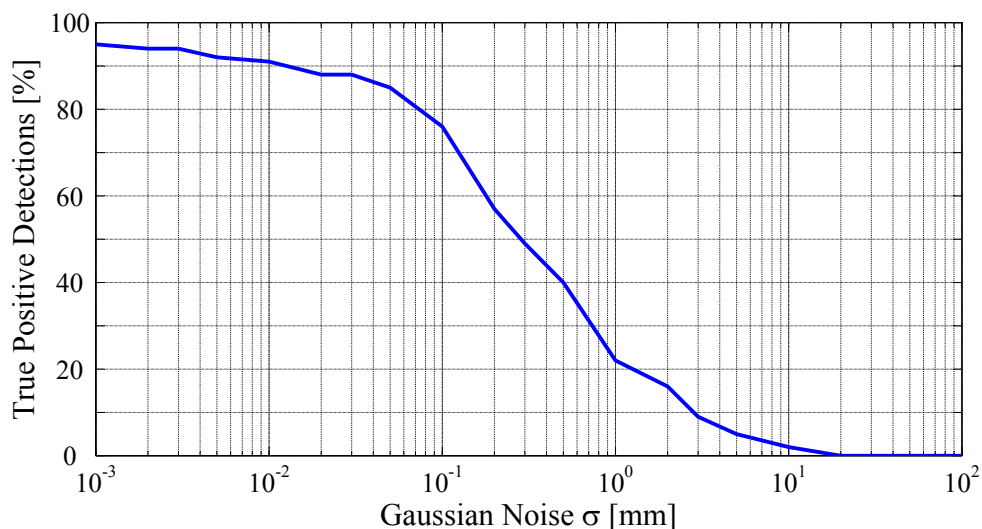


Figure 5.29: True positive detection rates facing different noise level.

searched object is uniquely present in the scene the detection method tolerates a much higher noise level. To keep the comparability with the true positive detection results in Figure 5.20(a) the detection tolerance is now $3mm$. Adding Gaussian noise with a σ of $1mm$ the true positive detection rate is 90%. With $\sigma = 5mm$ the rate is 52% and even with $\sigma = 10mm$ the detection rate is 7%. Figure 5.30 illustrates the heavily noise corrupted scene with $\sigma = 5mm$.

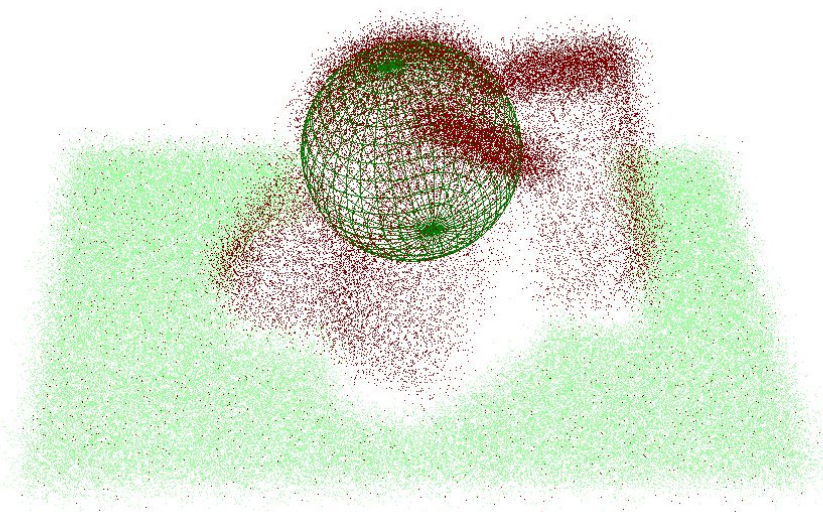


Figure 5.30: Gaussian noise ($\sigma = 5mm$) corrupted scene of Figure 5.18(b).

Concluding these two experiments a qualitative statement concerning the noise behavior can be given. The detection approach searching for an approximated shape has the ability to cope with a certain noise level in the range image. The tolerance is even higher if no similar objects are present in a scene.

Detection Examples on Daily Life Objects

In this section the detection of real objects is investigated using one Superquadric for model description. These experiments shall show the capability of the proposed approach for rapidly detecting a wide range of daily life objects handling occlusions in cluttered scenes. The objects are scanned with the mobile scanning head introduced in Section 5.2.1.

Three scenes are investigated where in all experiments the processing time for the table/ground plane removal was about 0.1 seconds. The model parameters for the following experiments, the number of calculated hypotheses, the range image size and the processing time are summarized in Table 5.6.

part	Fig. #	a_1	a_2	a_3	ϵ_1	ϵ_2	k_x	k_y	m	image size	process time
Cuboid	5.31(b)	15 mm	25 mm	35 mm	0.1	0.1	0.0	0.0	10	13360 points	1.3s
Sphere	5.31(c)	25 mm	25 mm	25 mm	1.0	1.0	0.0	0.0	10	13360 points	1.6s
Cylinder	5.31(d)	25 mm	25 mm	35 mm	0.1	1.0	0.0	0.0	10	13360 points	1.7s
Pyramid	5.31(e)	8 mm	8 mm	22 mm	0.1	0.1	-0.9	-0.9	18	13360 points	2.6s
Bowl	5.32(b)	60 mm	60 mm	20 mm	0.4	1.0	0.3	0.3	11	37708 points	1.8s
Mouse	5.33	45 mm	35 mm	20 mm	0.8	0.5	-0.3	0.0	100	35652 points	15.9s
Mug	5.33	40 mm	40 mm	50 mm	0.1	1.0	0.0	0.0	37	35652 points	6.0s
Cup	5.33	34 mm	34 mm	68 mm	0.1	1.0	0.2	0.2	38	35652 points	7.2s
Sponge	5.33	42 mm	25 mm	11 mm	0.1	0.1	0.0	0.0	100	35652 points	16.4s

Table 5.6: Summary of the algorithm and Superquadric parameters of the object detection results in Fig. 5.31, Fig. 5.32 and Fig. 5.33.

The first experiment demonstrates the detection robustness on a set of geometric primitives (see Figure 5.31) occluding each other. Note, the sparse data from the cuboid which is sufficient for a robust detection result.

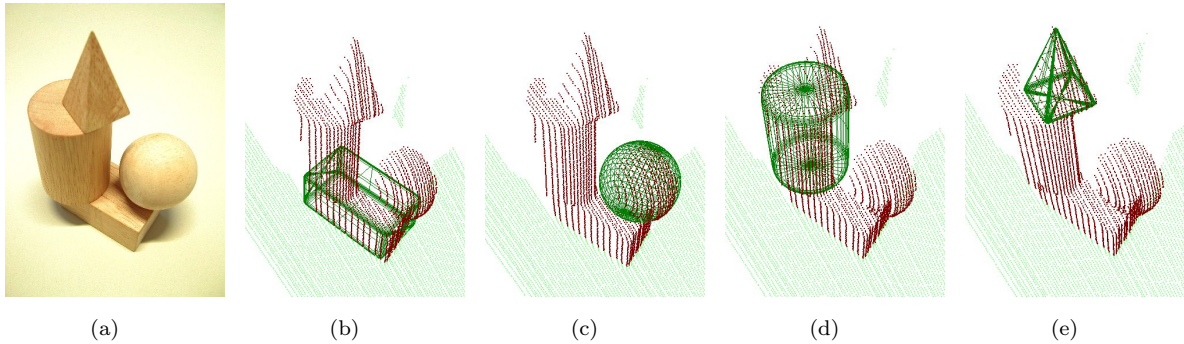


Figure 5.31: Detection results of some geometric primitives occluding each other.

The second experiment investigates the detection behavior in case of similar objects present in one scene. Figure 5.32(a) show the intensity image of the scene with the laser scanner. Figure 5.32(b) show the correctly detected wooden bowl with a similar white ceramic bowl located to the right of it. Note the partly occluded bowl. Using the hierarchical sub-scale approach speeds up the detection time nearly independent of the range image size. Detecting the wooden bowl is 1.4 times slower than the detection of the cuboid but note that the range image size is 3 times larger (Table 5.6).

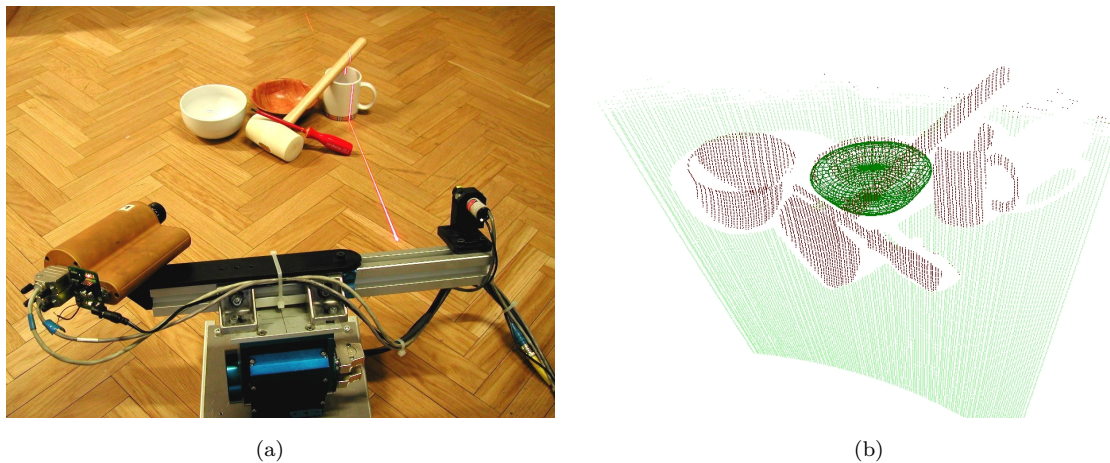


Figure 5.32: Detection of an occluded wooden bowl.

The last experiment, the desktop scene in Figure 5.33, demonstrates the detection of small and simple objects (computer mouse, sponge, cup and mug) in a real world scene. Due to the high number of different objects in the large scene the processing time is longer than in the previous experiments. Also note that objects are smaller than in the previous examples. Nevertheless, the loss of details in the range image does not reduce the robustness of the detection, which can be seen in the similar shapes of the computer mouse and the sponge.

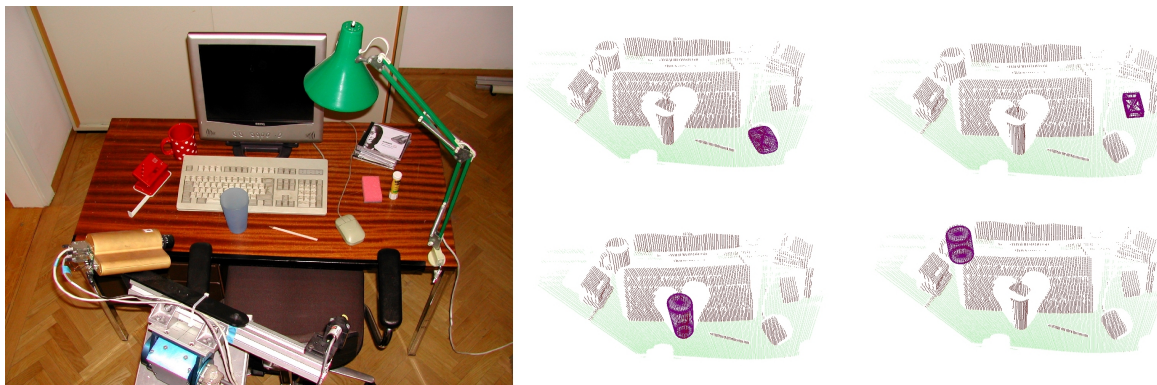


Figure 5.33: Sensor prototype scanning a desktop scene and presented detection results of small objects on the table – top left: a computer mouse, top right: a sponge, bottom left: a cup, bottom right: a mug.

5.2.3 Composite Model Description

The experiments in this chapter address the object detection using model descriptions with several Superquadrics. For mobile robotics in a home or office environment an often used object is the coffee mug, hence it will be further investigated. For a rough localization it is not necessary to model the mug in detail. In fact, modeling the mug with a cylinder

is widely sufficient (compare Figure 5.33). In order to localize the mug in 6 DoF the handle must be modeled and detected as well. This is a challenging task, because in the majority of cases only sparse raw data from the handle is available. Nevertheless, modeling the handle with an approximately flattened Superquadric is sufficient for localization and even for grasping tasks.

In order to demonstrate that the composite mug model is sufficient and well detectable, Figure 5.34 shows results with different handle positions. The range image for this experiment is acquired with the sensor cell.

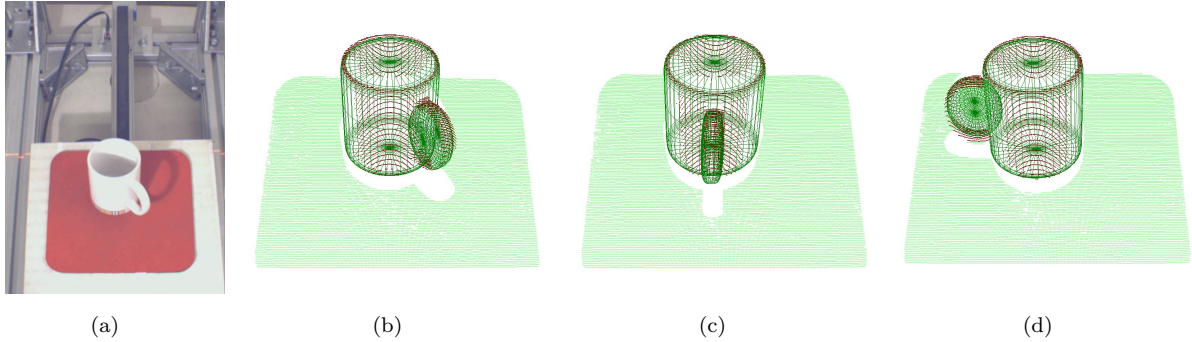


Figure 5.34: Detection of a mug with different handle orientations, modeled with two Superquadrics.

The second experiment demonstrates the detection of the mug (Figure 5.35(a)) in a real scene shown in Figure 5.32(a), captured with the mobile scanning head. The mug is also partly occluded but the detection needed 27.3 seconds because of finding the small handle in the scene ($m = 100!$). However note, the sparse data available from the mug (especially of the handle) and the robust detection of it in 6 DoF suitable for robotic grasping tasks. Beside that, Salganicoff et al. showed that for a grasp point planning the Superquadrics' size, shape and pose parameter are sufficient [111]. Detecting the cup in 5 degrees of freedom (only the cup-body) speeds up the processing time significantly (3.1 seconds). Table 5.7 summarizes the model and algorithm parameters of all mug detection experiments.

Fig.	part	a_1	a_2	a_3	ε_1	ε_2	k_x	k_y	m	k	α_k	d_k	time
-	Mug-Body	38mm	38mm	48mm	0.1	1.0	0.0	0.0	-	-	-	-	-
-	Mug-Handle	30mm	23mm	8mm	0.5	1.0	0.0	0.0	-	-	-	-	-
-	Mug	-	-	-	-	-	-	-	-	-	90°	50mm	-
5.34	solid Mug	-	-	-	-	-	-	-	112	2	-	-	5.3s
5.35(a)	occluded Mug	-	-	-	-	-	-	-	20	2	-	-	27.3s

Table 5.7: Summary of the algorithm and Superquadric parameter of the object detection results in Figure 5.34 and 5.35(a).

The next experiment investigates the object detection of a rubber-mallet in the scene of Figure 5.32(a) and in a cluttered scene (Figure 5.36(a)) in terms of detection probability using a composite object model exploiting the Superquadric relationships. Figure 5.35(b)

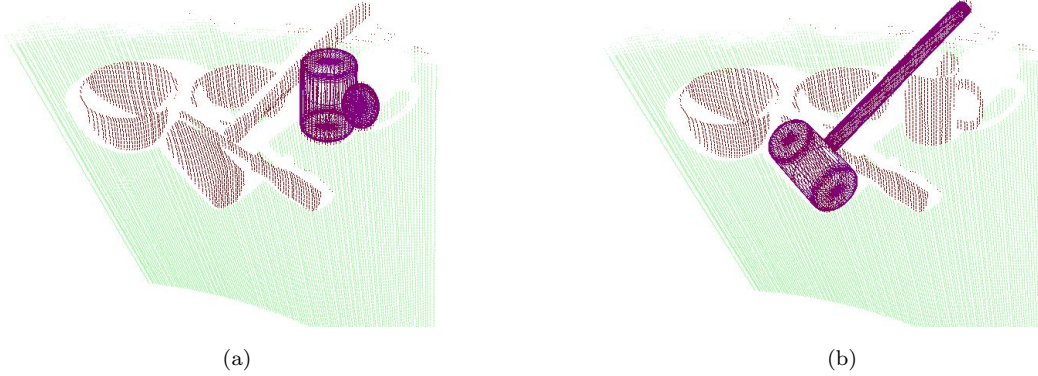


Figure 5.35: Two detection examples of the scene in Figure 5.32(a) – (a) a coffee mug and (b) a rubber-mallet.

illustrates the correctly detected rubber-mallet in a cluttered tool scene (range image size: 31101 points). Trying to detect the rubber-mallet with one Superquadric in this scene with many shape ambiguities does not end in a robust result. To evaluate the performance we examined 100 experiments on the same scene describing the rubber-mallet first with the head model only, second with the shaft only and finally with both Superquadric models exploiting their relationship. The results are summarized in Tab. 5.8. Note, that trying to detect the mallet with a single Superquadric causes a false detection in every second trial. Only when exploiting the geometric relationship the true positive detections of the rubber-mallet raise to 76%. The remainder are false detections due to the similarity of objects. In the scene in Figure 5.35(b) without similarly shaped object parts (such as the shaft of the metal hammer and the spray tin in Fig. 5.36) the detection rate of the rubber-mallet is 84%. The reason for not obtaining 100% is that the number of hypotheses m is bound from 10 to 100 due to computational efficiency. It still remains a random process and if m is increased the detection results converge towards 100% (see Figure 5.20(a)).

Fig.	part	a_1	a_2	a_3	ε_1	ε_2	k_x	k_y	m	k	detect	α_k	d_k	time
5.36(b)	Head	30mm	30mm	56mm	0.1	1.0	0.0	0.0	10	1	48%	-	-	2.3s
5.36(b)	Shaft	12mm	10mm	120mm	0.1	1.0	0.0	0.0	26	1	55%	-	-	7.1s
5.36(b)	Mallet	-	-	-	-	-	-	-	36	2	76%	90°	170mm	10.2s
5.35(b)	Mallet	-	-	-	-	-	-	-	77	2	84%	90°	170mm	20.3s

Table 5.8: Performance evaluation of object detection in Figure 5.36 and summary of the algorithm and Superquadric parameter.

The last experiment investigates the detection of a model consisting of three linked Superquadrics. As previously mentioned the detection has its lack if several similar objects are in the scene. Therefore, the relationships between the model components can be exploited to get a true positive detection result. To create a difficult detection task only paper roles with the same shape and dimensions are used to model an arbitrary object, which can be seen in Figure 5.37(a). To create occlusions additional four paper roles are

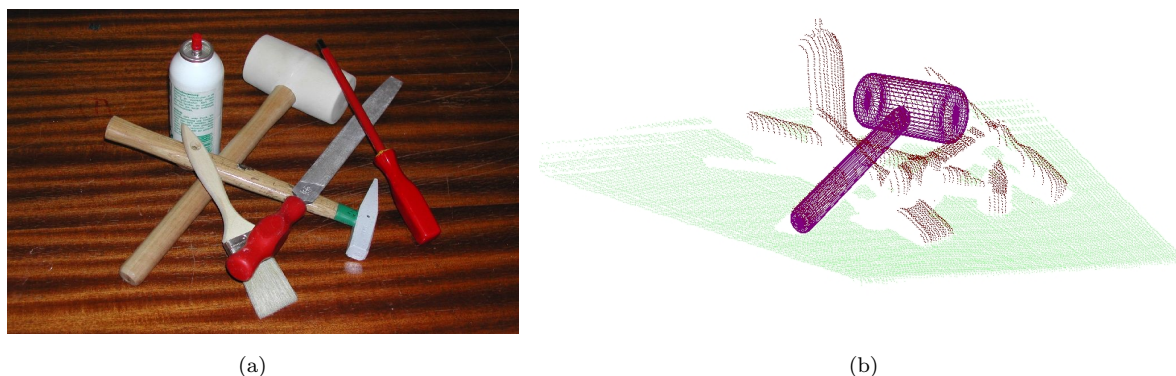


Figure 5.36: Detection result of a rubber-mallet in a cluttered tool scene of similar objects exploiting the relation of the two modeled Superquadrics.

added to scene (Figure 5.37(b)). Only exhibiting the relations the random chosen object can be detected robustly with a true positive rate of 82% within an average time of 10.3 seconds. Figure 5.37(c) shows a true positive result, but note, that the position of the cylinders can be shifted along their axis, as described in Figure 5.26. Finally the model parameter and detection results are summarized in Table 5.9.

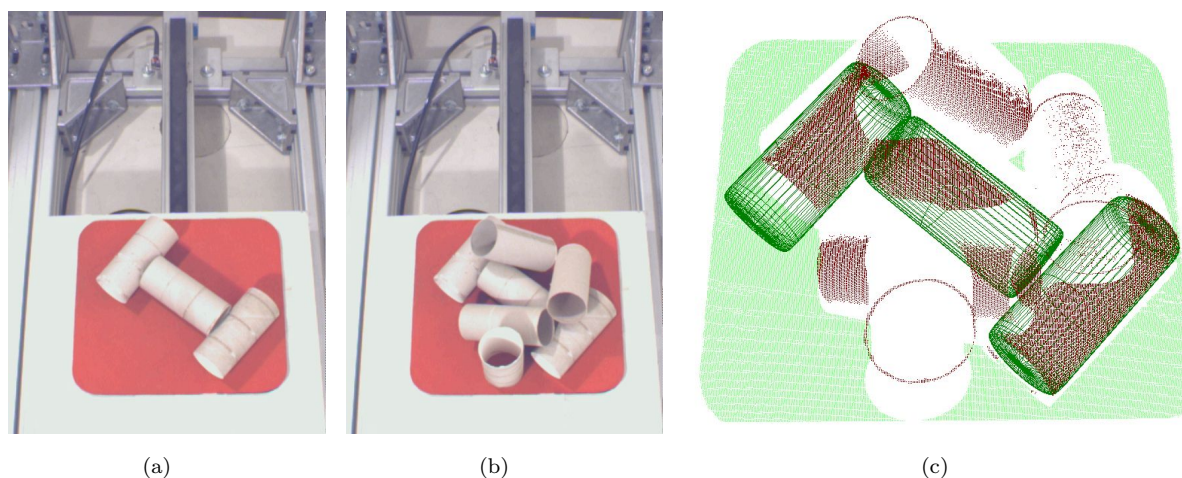


Figure 5.37: Detection of an arbitrarily shaped object consisting of three similar paper-roles. (a) Model to be detected. (b) Occluded and cluttered scene. (c) Detection result.

part	a_1	a_2	a_3	ε_1	ε_2	k_x	k_y	m	k	$\alpha_{k_{12}}$	$d_{k_{12}}$	$\alpha_{k_{13}}$	$d_{k_{13}}$	$\alpha_{k_{23}}$	$d_{k_{23}}$	time
Role	21mm	21mm	48mm	0.1	1.0	0.0	0.0	10	1	-	-	-	-	-	-	-
Model	-	-	-	-	-	-	-	30	3	90°	69mm	90°	138mm	90°	69mm	10.3s

Table 5.9: Performance evaluation of the object detection in Figure 5.37 and summary of the algorithm and Superquadric parameter.

5.2.4 Bore Hole versus Model-Based Detection

As the timing bars in Figure 4.27 indicated, a previous range image segmentation for a following feature detection is a time consuming procedure. The power of the model-based approach, introduced in this chapter, is that a previous segmentation is not needed. Hence, the bore hole detection approach of Chapter 4 is faced against the performance of the Superquadric-based detection approach. Therefore, the computational cost as well as the detection accuracy of both methods are evaluated and compared. In the following the least-squares fitting (LSF) is related to the Superquadric fit and the random samples fitting is related to the cylinder fitting.

To apply the object detection method a model of the bore is needed. Through CAD information it is available in the FibreScope's framework, where the length of the Superquadric is set to 40mm . Figure 5.38 shows two bores with a diameter of 11mm (one cast part of Figure 4.1(a) and a slanted bore) and the detection result using a Superquadric model. To obtain comparable results of the two methods the same range image must be provided. Hence, the range image section of the bore detection method is used for the model-based object detection (red marked raw data points in Figure 5.38).

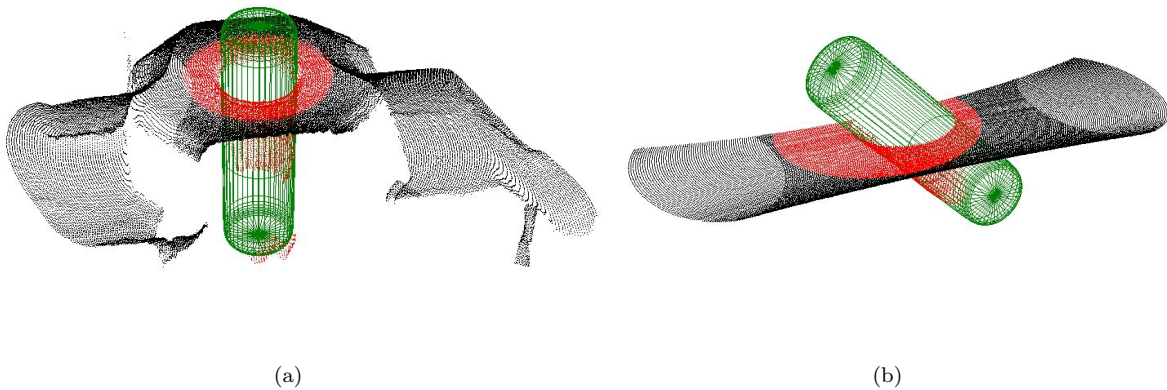


Figure 5.38: Method comparison with two examples. (a) Bore #1: Cast part of Figure 4.1(a). (b) Bore #2: Slanted bore hole. Both bores have a diameter of 11mm and the raw data points which are processed are marked red.

First the accuracy of the bore entrance pose is evaluated by executing each method 100 times. The results are presented in Figure 5.39 and 5.40. The relative pose deviation in x - and y -direction of the world coordinate system is shown in overview and in zoomed detail. The z -direction denotes the normal vector of the bore entrance plane. The experiments are carried out using full resolution for both methods. For the Superquadric detection 10 hypothesis with each 10 RANSAC trials are performed. Ten Levenberg-Marquardt iterations are used for each Superquadric fit.

Interpreting the results it can be seen that the poses of the Superquadric detection (LSF) are significantly wider spread than the detected poses achieved with the sequential cylinder fit (RSF). The evaluation results are summarized in Table 5.10. The averaged

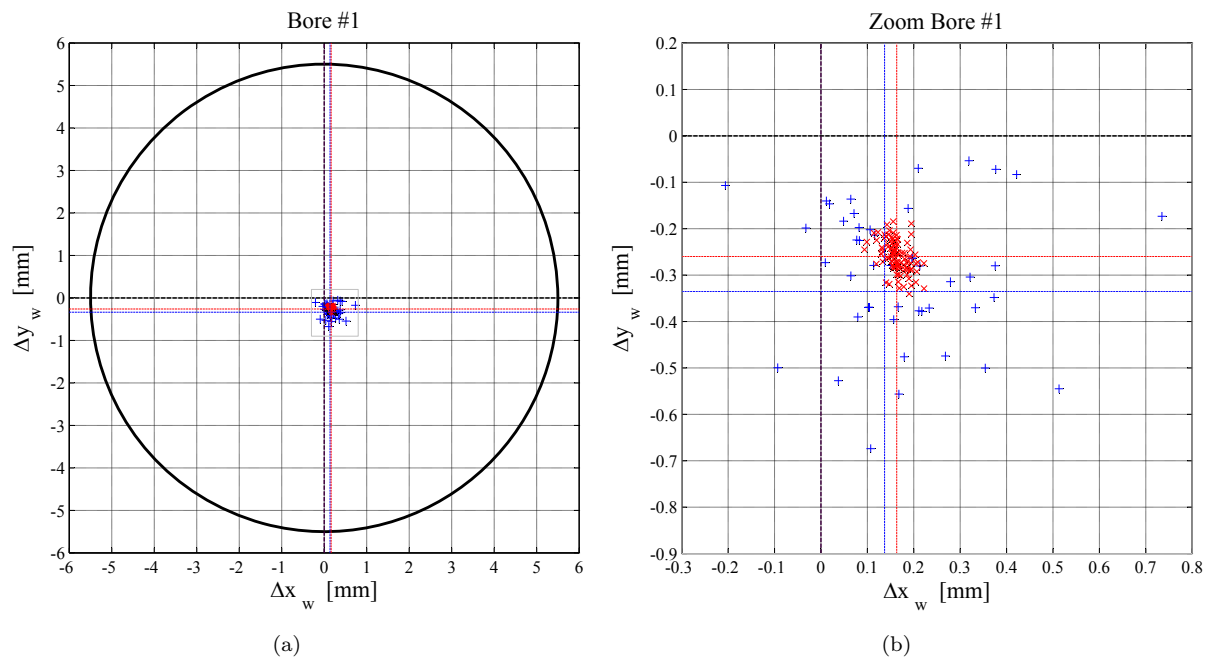


Figure 5.39: Bore detection results of of bore #1 using 100 attempts of the LSF (blue +) and the RSF (red ×) method.

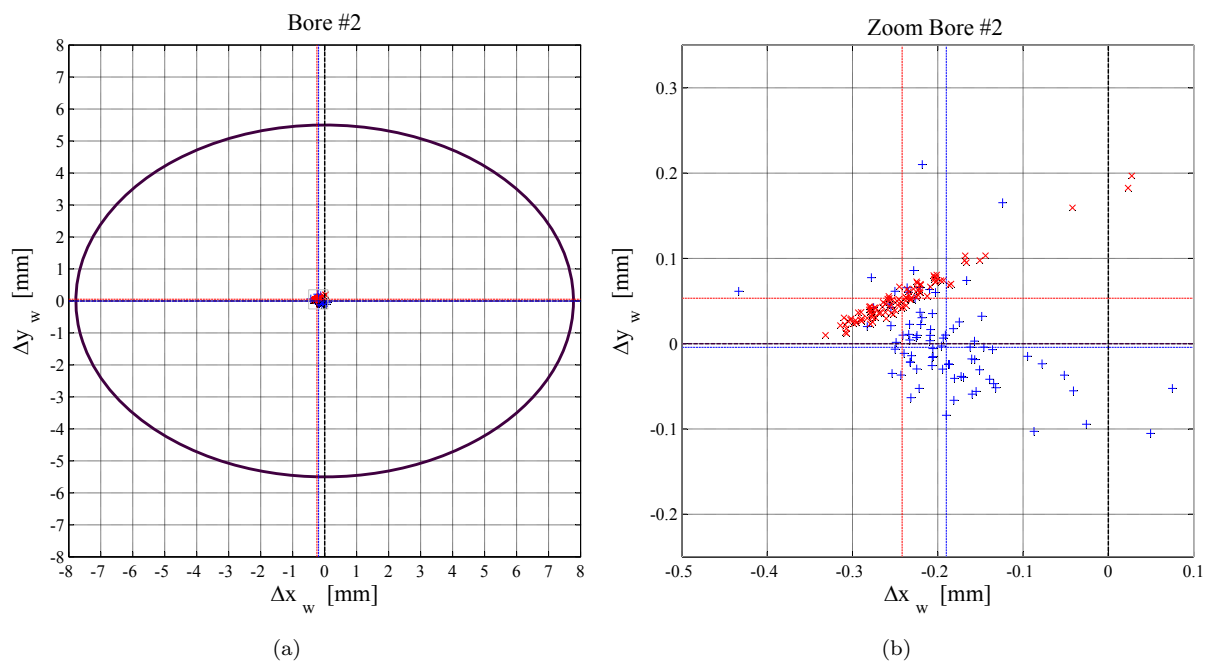


Figure 5.40: Bore detection results of of bore #2 using 100 attempts of the LSF (blue +) and the RSF (red ×) method.

centers of the LSF and the RSF are as well presented as the standard deviation and the

maximum of the distance

$$d = \sqrt{(\bar{x} - x_p)^2 + (\bar{y} - y_p)^2}, \quad (5.12)$$

with \mathbf{x}_p the detection pose. The average processing time is also shown in Table 5.10.

The interesting observation is that the RSF achieves accurate results but needs in both examples more processing time due to the segmentation process. Note, that the RSF needs a second cylinder fit because of radial notches inside the bore of the cast part. Because of a close scanning distance in the case of the slanted bore the scan consists of high dense raw data and the effective range data to be processed is rather large. But note, that the processing times for the two bores can be reduced without a loss of detection accuracy on sub-sampled data. Using 20% of the raw data points the processing time for bore #1 is $819ms$ and for bore #2 $1.198s$.

Summarizing the results of the method comparison, one can say, that both approaches are efficient methods for model-based detection, whereas the RSF, tuned for cylinder fitting, achieves more accurate results. The high computational cost of the segmentation can be reduced using sub-sampled data. The detection approach using Superquadrics is also well suited to detect bore holes, but with the lack of a reduced accuracy due to the more general approach.

Fit Method		$\Delta\bar{x}$	$\Delta\bar{y}$	std(d)	max(d)	t
Bore #1	LSF	$0.137mm$	$-0.336mm$	$0.108mm$	$0.620mm$	$2.363s$
	RSF	$0.163mm$	$-0.260mm$	$0.020mm$	$0.091mm$	$2.879s$
Bore #2	LSF	$-0.190mm$	$-0.004mm$	$0.053mm$	$0.270mm$	$2.627s$
	RSF	$-0.242mm$	$-0.053mm$	$0.048mm$	$0.305mm$	$5.707s$

Table 5.10: Summarized results of the evaluated comparison of bore #1 and #2.

Chapter 6

Applications

This chapter discusses the direct outcome of the research work giving examples of applications. First starting with the application of a bore inspection task the second application presents a concept of fusing shape and appearance information utilized for object handling operations.

The bore hole detection was part of the project FibreScope, which prototype has proven the feasibility of the system concept with long-term demonstrations on two industrial fairs, the AUSTROTEC from September 30th to October 2nd 2003 in Austria and the AUTOMATICA from June 15th to 18th 2004 in Germany. Figure 6.1 gives a rough overview of the system components and shows the presented prototype on the AUTOMATICA fair. The technological know how is now part of the company Profactor, whose goal is the com-

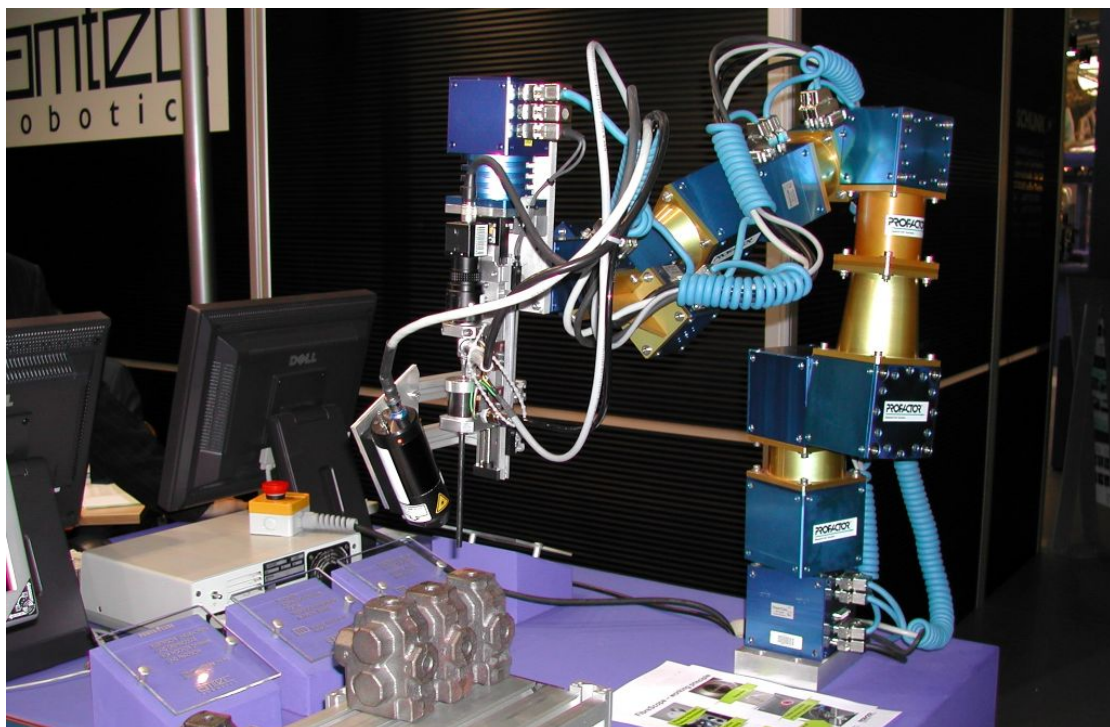


Figure 6.1: Prototype of the bore inspection system presented at an industrial fair.

mercialization of new technologies suited for automated industrial solutions. Depending on customer requests, fully automated robotic quality assurance tasks coping lot size one standards, are from now on in the product range to be offered.

Combining visual shape-capturing and vision-based object manipulation without intermediate manual interaction steps is important for autonomous robotic systems. The outcome of the Superquadric-based object detection approach is the concept of a vision system that guides the manipulation of convex shaped objects. Robotic applications such as visual servoing or grasping tasks are the goal. The main contribution is the closing of the gap between shape-capturing, detecting and tracking the object, integrating the individual vision steps in a fully automatic way. The approach is to show the object once to the robot vision system. It is scanned by a laser range sensor that derives a volumetric object description for further detection and tracking. Performing the detection in a totally different environment (e.g. in a home environment on potential object places) is possible and results in the object pose, which is the starting pose for the subsequent tracker. This monocular tracker uses the 3D-pose as well as the 3D-object model delivered during the shape-capturing step for continuously updating the pose of the object. Appearance information for the tracker (cues in any form, i.e., interest points in the system proposed) is not derived until now, i.e., from the actual scene – decoupling the illumination and environmental conditions of the shape-capturing and the manipulation steps.

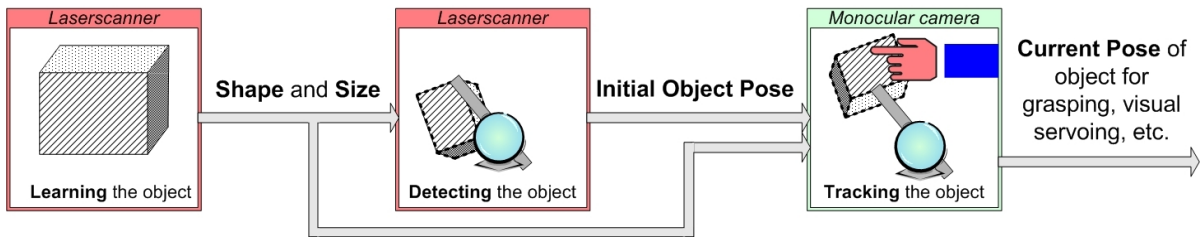


Figure 6.2: Concept of our perceptual system: The fully automatic sequence starts with the object capturing where the size and shape parameters are gained that are used for subsequent object detection and tracking in an occluded and cluttered scene.

The proposed approach of the concept, described in detail in [112], is outlined in Figure 6.2. At the beginning of the fully automatic sequence the shape of the object is learned by capturing a range image and fitting a single Superquadric. To verify this concept we focused on simple objects where a part decomposition is not necessary for an entire model description. After the learning procedure a range image of the scene of interest can be acquired. Applying the object detection method, introduced in Chapter 5, the object can be located in the occluded and cluttered environment. This detected pose can further be used as initial starting solution for a subsequent tracking of the object in which any kind of model-based tracker may be used. In a first implementation we used interest points (SIFT, [75]) on the object’s surface along with the pose of the previous frame – or the starting pose from the detection described respectively above – to compute the current 3D pose (Lu et al., [76]).

Three experiments have been conducted to verify the proposed approach. In Figure 6.3 an Uncle Ben's box is handled. Note the dynamic occlusions of the mallet-shaft while tracking. Figure 6.4 shows the learning, detection and tracking of a tin can while Figure 6.5 presents the automatic task handling of a video tape. All these experiments verified the feasibility of the concept. Further work will focus on the system performance and on the ability to learn, detect and track more complex models.

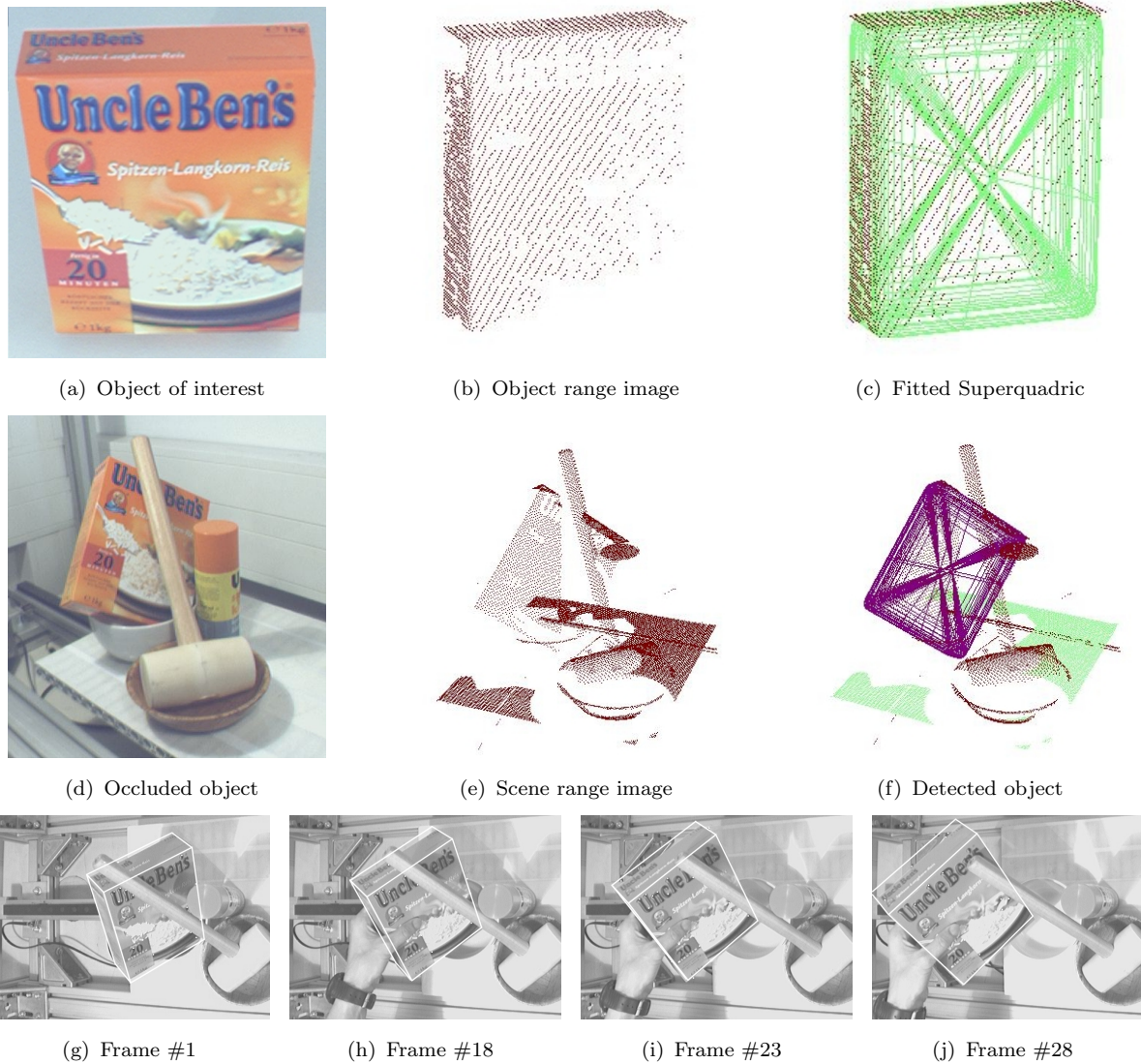


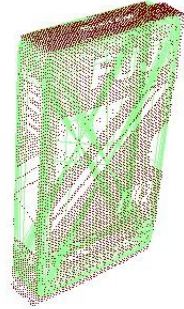
Figure 6.3: Experiment 1: Fig. (a) to (c): Capturing the object's parameters; Fig. (d) to (f): Detection of the object in the scene; Fig. (g) to (j): Some tracking frames.



Figure 6.4: Experiment 2: Handling of a cylinder. Capturing model (first row), Detecting (second row) and Tracking (last row). The reprojected pose is depicted as mesh-grid.



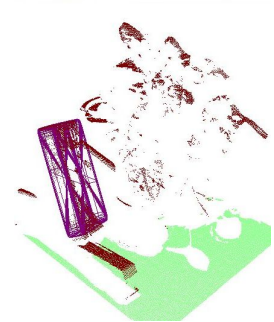
(a) Object of interest: a video tape.



(b) Range image with fitted model.



(c) Ocluded object in a cluttered scene.



(d) Range image with the detected tape.



(e) Starting pose for the tracker.



(f) Tracking sequence showing frame #15.

Figure 6.5: Experimental 3: Fig. (a) and (b): Learning the object in a scene with no disturbances; Fig. (c) and (d): Detection of the object in the cluttered scene; Fig. (e): Starting pose of the object in the scene for the tracker; the pose is reprojected and depicted as white lines; Fig. (f): A frame of the tracking sequence. Note the white dots marking the interest points.

Chapter 7

Conclusion

This thesis presented an efficient part feature and object detection approach by fitting geometric models to range image data scanned from one view. The cylinder- and the Superquadric models are used to introduce a low-level feature detection and a high-level object detection approach. In both cases the focus of research work was to achieve robust and reliable detection results suited for industrial and robotics applications. The second request on the presented methods was the fast processing to obtain the ability to operate real time.

The presented work in Chapter 4 describes a method that detects a class of features, namely bore holes, in range images suitable for sensor guided robotics. Bore holes can have different edges or chamfers, surface properties, and notches. Although range sensing can deliver only a small fraction of the bore hole surface, detection is robust. Robustness is achieved by exploiting expected bore hole dimension and location from CAD data, accurate range data from a compact robot-held sensor, a normal vector estimation coping with many boundary conditions and a criteria to select good points for estimating the cylinder radius. Fast processing in less than 1s is achieved with sampling data points and a sequential fit of the bore hole cylinder.

For the industrial implementation of this method we used a highly flexible approach with a robot to move an endoscope in 6 DoF to inspect each bore around the work piece on a feed system. To compensate large uncertainties in real 3D we used a laser range sensor, which is mounted on the robot arm, to achieve a sensor-based detection of the position and orientation of the bore. To teach the system new work pieces we chose a rapid, ergonomic and simple way of programming the system via a CAD editing tool, which enables the user to efficiently generate the robot program within a few minutes. This includes a collision avoidance check of the robot movement. The endoscope is connected with a CCD camera to capture pictures from the bore surface for a subsequent automated quality control. However if the detection and localization of the bore hole fails the endoscope is protected with a clearance check. In case of an endoscope collision while inspection process the endoscope is retired immediately.

The bore detection approach can be extended to detect elliptical or conic bores because the Gaussian image supports a segmentation of these shapes. This will expand the

application to other inspection tasks. Example given, a motor block consist of a lot of conical, elliptical or curved bores that need to be inspected regarding quality assurance.

To sum up Chapter 5, a 3D object detection approach has been introduced using a geometric model description approximated with Superquadrics. Fast and robust results are achieved combining the RANSAC algorithm with a hierarchical sub-scale search. A key for robust detection is the introduction of two new criteria of fit and a ranked voting in the hypothesis selection step. For compound objects two more criteria have been proposed and evaluated using the same voting procedure. Experiments confirmed the rapid and reliable detection of every day objects.

Learning the model parameter in a more natural way, as proposed in the previous chapter, applied to convex shaped objects is still a challenging task, which should be tackled in the future. Showing a cognitive system the object which it should detect, the system should be able to extract the relevant Superquadric parameter. This will enable the system to learn new tasks and to improve the human-robot interface. First steps are being taken to achieve this goal to realize an autonomous robot system. Acquiring a color range image of the object (compare the sensor cell in Section 5.2.1) enables a geometric decomposition [98] and final Superquadric fit to describe the object. First tests have shown the feasibility of this approach but there is more work needed to achieve robust extraction of the model parameters.

The future goal of our research group is a totally independent mobile robot platform which processes naturally given human commands and behaves autonomously to solve the given task. The command “*James, please bring me the coffee mug*” should give enough information to execute the job. In this context efficient part feature and object detection methods, as presented in this thesis, are an important step towards visual perception of a cognitive robotic system.

Glossary

3D Three Dimensional.

CAD Computer Aided Design.

CCD Charge Coupled Device – camera’s sensor chip technology.

CORBA Component Object Request Broker Architecture – representing the architecture of a middleware framework.

CSS Circular Scanning Sensor – laser scanner used in the project FibreScope.

DoF Degrees of Freedom – in most cases the degrees of freedom are related to translation and rotation in an Euclidean 3D space.

FibreScope Flexible Inspection of Bores with a Robotic Endoscope.

FlexPaint Efficient Low Volume High Variant Robotized Painting.

GUI Graphical User Interface.

IVP Integrated Vision Products – Swedish company associated with SICK, producing laser range sensors.

LASER Light Amplification by Stimulated Emission of Radiation.

LSF Least-Squares Fitting – in this thesis defined for Superquadric fitting.

OLP Off-Line Programming – this term is used in robot programming.

MAPP Matrix Array Picture Processor.

MCU Motion Control Unit.

Origin Define the coordinate frame center.

OST Oxford Sensors Technology – produces the Circular Scanning Sensor in the FibreScope Project.

PCA Principal Component Analysis – image processing methodology [92].

POSE Position and Orientation – if nothing else is noted the pose is defined with 6 DoF (3 for the position and 3 for the orientation in a 3D space).

Real Time Definition given in the German industry standards, DIN 44300: *The operating mode of a computer system in which the programs for the processing of data arriving from the outside are permanently ready, so that their results will be available within predetermined periods of time; the arrival times of the data can be randomly distributed or be already a priori determined depending on the different applications.*

RANSAC **R**andom **S**ample and **C**onsensus – algorithm proposed by [40].

RSF **R**andom **S**ample **F**itting – in this thesis defined for the novel sequential cylinder fit approach.

Range Data Three dimensional point cloud measured from a laser range scanner.

SME **S**mall and **M**edium-sized **E**nterprises – this term is used related to supplier in research projects funded by the European Community.

STL **S**tereo **L**ithography - file format for a 3D mesh representation.

Superquadric Mathematical geometric model with a closed surface and a quadric characteristic, described with two shape and three size parameter [3].

TCP **T**ool **C**enter **P**oint – reference point of a robot arm.

VRML **V**irtual **R**eality **M**odeling **L**anguage – file specification used for the data exchange in the project FlexPaint.

VTK **V**isualization **T**ool **K**it – freely available open source software.
(<http://public.kitware.com/vtk>)

Bibliography

- [1] M. Attene, B. Falcidieno, and M. Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *Springer; The Visual Computer: International Journal of Computer Graphics*, 22(3):181–193, 2006.
- [2] C.C. Barcenas and P.M. Griffin. Geometric tolerance verification using superquadrics. *Kluwer; IIE Transactions*, 33(12):1109–1119, 2001.
- [3] A.H. Barr. Superquadrics and angle preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.
- [4] A.H. Barr. Global and local deformations of solid primitives. *IEEE Computer Graphics and Applications*, 18(3):21–30, 1984.
- [5] T. Berndorfer, C. Eitzinger, A. Brenner, and W. Van Dyck. Multivariate approach to obtain real time behaviour of image processing applications. In *Proceedings of the SPIE: Process Imaging for Automatic Control*, volume 4188, pages 10–16, 2001.
- [6] M. Bertero, T.A. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, 1988.
- [7] P.J. Besl and R.C. Jain. Three-dimensional object recognition. *ACM; Computing Survey*, 17(1):75–145, 1985.
- [8] I. Biederman. Recognition-by-components: A theory of human image understanding. *APA Journal; Psychological Review*, 94(2):115–147, 1987.
- [9] G. Biegelbauer, A. Pichler, M. Vincze, C. Nielsen, H. Andersen, and K. Häusler. The inverse approach of FlexPaint: Automatic generation of robot painting motions for unknown parts. *IEEE Robotics & Automation Magazine*, 12(3):24–34, 2005.
- [10] G. Biegelbauer and M. Vincze. Fast and robust bore detection in range image data for industrial automation. In *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization & Transmission*, pages 526–533, 2004.
- [11] G. Biegelbauer and M. Vincze. Robust and fully automated robotic bore inspection for high variant parts. In *Proceedings of the IEEE Workshop on Robotic Sensing*, pages 9–13, 2004.

BIBLIOGRAPHY

- [12] G. Biegelbauer and M. Vincze. 3D vision-guided bore inspection system. In *Proceedings of the IEEE International Conference on Computer Vision Systems*, pages 22–22, 2006.
- [13] G. Biegelbauer, M. Vincze, H. Nöhmayer, and C. Eberst. Sensor based robotics for fully automated inspection of bores at low volume high variant parts. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 5, pages 4852–4857, 2004.
- [14] T. Binford. Visual perception by a computer. In *Proceedings of the IEEE Conference on Systems and Control*, pages 116–123, 1971.
- [15] M.M. Blane, Z. Lei, H. Civi, and D.B. Cooper. The 3L algorithm for fitting implicit polynomial curves and surfaces to data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):298–313, 2004.
- [16] W. Blaschke. Untersuchungen über die Geometrie der Speere in der Euklidischen Ebene. *Monatshefte für Mathematik und Physik*, 21:3–60, 1910.
- [17] A. Blomdell, G. Bolmsjo, T. Brogardh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K. Nilsson, M. Olsson, T. Olsson, A. Robertsson, and Jianjun Wang. Extending an industrial robot controller: Implementation and applications of a fast open sensor interface. *IEEE Robotics & Automation Magazine*, 12(3):85–94, 2005.
- [18] J. Böhm. *Modellbasierte Segmentierung und Objekterkennung aus Distanzbildern*. PhD Thesis, Faculty of Aerospace Engineering and Geodesy, University of Stuttgart, 2005.
- [19] F. Bookstein. Fitting conic sections to scattered data. *Academic Press; Computer Vision, Graphics and Image Processing*, 9:56–71, 1979.
- [20] V. Brajovic. Fast row-parallel CMOS range image sensor. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 945–949, 2004.
- [21] C. Brechbühler, G. Gerig, and O. Kübler. Surface parametrization and shape description. In *Proceedings of the SPIE: Visualization in Biomedical Computing*, volume 1808, pages 80–89, 1992.
- [22] C. Brenner, J. Bohm, and J. Guhring. Photogrammetric calibration and accuracy evaluation of a cross-pattern stripe projector. In *Proceedings of the SPIE: Conference on Videometrics*, volume 3641, pages 164–172, 1999.
- [23] R. Brooks. Model-based 3D interpretation of 2D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):140–150, 1983.
- [24] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

BIBLIOGRAPHY

- [25] V. Cantoni and L. Lombardi. Hierarchical architectures for computer vision. In *Proceedings of the IEEE Euromicro Workshop on Parallel and Distributed Processing*, pages 392–398, 1995.
- [26] T. Chaperon and F. Goulette. Extracting cylinders in full 3D data using a random sampling method and the Gaussian image. In *Proceedings of the 6th International Fall Workshop: Vision, Modeling, and Visualization*, pages 35–42, 2001.
- [27] L. Chevalier, F. Jaillet, and A. Baskurt. Segmentation and superquadric modeling of 3D objects. *Journal of WSCG; ISSN 1213-6972*, 11(1), 2002.
- [28] I. Cohen and L.D. Cohen. A hyperquadric model for 2-D and 3-D data fitting. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 2, pages 403–405, 1994.
- [29] S. Coquillart. A control-point-based sweeping technique. *IEEE Computer Graphics and Applications*, 7(11):36–45, 1987.
- [30] D. Dion, Jr., D. Laurendeau, and R. Bergevin. Generalized cylinders extraction in a range image. In *Proceedings of the IEEE International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 141–147, 1997.
- [31] M. de Berg, O. Schwarzkopf, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, ISBN: 3-540-65620-0, 2000.
- [32] M.P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, ISBN: 0132125897, 1976.
- [33] R.O. Duda and P.E. Hart. The use of the Hough transform to detect lines und curves in pictures. *Comm. Assoc. Comp. Machine*, 15:11–15, 1972.
- [34] C. Eitzinger. Oberflächeninspektion komplexer Werkstücke. In *Qualitätssicherung durch Messtechnik und Bildverarbeitung*, 1999.
- [35] P. Faber and R.B. Fischer. Euclidean fitting revisited. In *Proceedings of the Internatinoal Workshop on Visual Forms*, pages 165–175, 2001.
- [36] P. Faber and R.B. Fischer. Pros and cons of Euclidean fitting. In *Proceedings of the Annual German Symposium for Pattern Recognition*, pages 414–420, 2001.
- [37] J.T. Feddema and C.Q. Little. Rapid world modeling: Fitting range data to geometric primitives. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 2807–2812, 1997.
- [38] V. Ferrari, T. Tuytelaars, and L. van Gool. Simultaneous object recognition and segmentation by image exploration. In *Proceedings of the European Conference on Computer Vision*, pages 40–54, 2004.

BIBLIOGRAPHY

- [39] F.P. Ferrie, J. Lagarde, and P. Whaite. Recovery of volumetric object descriptions from laser rangefinder images. In *Proceedings of the European Conference on Computer Vision*, pages 387–396, 1990.
- [40] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 26(6):381–395, 1981.
- [41] E. Freund, D. Rokossa, and J. Rossmann. Process-oriented approach to an efficient offline programming of industrial robots. In *Proceedings of the IEEE IECON Conference*, pages 208–213, 1998.
- [42] W. Furong and Y. Baozong. Least-squares fitting for deformable superquadric models based on orthogonal distance. *Elsevier, Pattern Recognition Letters*, 25(33):933–941, 2004.
- [43] F. Goulette. Automatic CAD modeling of industrial pipes from range images. In *Proceedings of the IEEE International Conference on Recent Advances on 3-D Digital Imaging and Modeling*, pages 229–233, 1997.
- [44] A. Gupta, L. Bogoni, and R. Bajcsy. Quantitative and qualitative measures for the evaluation of the superquadric models. In *Proceedings of the IEEE Workshop on Interpretation of 3D Scenes*, volume 12, pages 162–196, 1989.
- [45] E. Hameiri and I. Shimshoni. Estimating the principal curvatures and the Darboux frame from real 3-D range data. *IEEE Transactions on Systems, Man and Cybernetics*, 33(4):626–637, 2003.
- [46] S. Han, D.B. Goldgof, and K.W. Bowyer. Using hyperquadrics for shape recovery from range data. In *Proceedings of the IEEE Conference on Computer Vision*, pages 492–496, 1993.
- [47] A.J. Hanson. Smoothly deformable shapes with convex polyhedral bounds. *Elsevier; Computer Vision, Graphics and Image Processing*, 44:191–210, 1988.
- [48] J. Haverinen and J. Röning. A 3-D scanner capturing range and color. Design and calibration. In *Proceedings of Vision Interfaces*, pages 378–381, 2000.
- [49] J. Heikkilä. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1066–1077, 2000.
- [50] A. Helzer, M. Barzohar, and D. Malah. Stable fitting of 2D curves and 3D surfaces by implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(210):1283–1294, 2004.
- [51] P. Hertling, L. Hog, L. Larsen, J.W. Perram, and H.G. Petersen. Task curve planning for painting robots – Part I: Process modeling and calibration. *IEEE Transactions on Robotics and Automation*, 12(2):324–330, 1996.

- [52] D.D. Hoffman and W.A. Richards. Parts of recognition. *Cognition*, 18:65–96, 1984.
- [53] A. Hoover, G. Jean-Babtiste, X. Jiang, P.J. Flynn, H. Bunke, D.B. Goldgof, D.W. Eggert, A. Fitzgibbon, and R.B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):1–17, 1996.
- [54] P.V.C. Hough. Method and means for recognizing complex patterns. *US Patent 3069654*, 1962.
- [55] Q. Iqbal and J.K. Aggarwall. Applying perceptual grouping to content-based image retrieval: Building images. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 42–48. ISBN: 0-7695-0662-3, 1999.
- [56] L. Jacob, D. Kollhoff, R. Gergs, A. Graf, and K.-H. Franke. System zur Inspektion von langen Bohrungen in Aluminiumguss. In *Proceedings of the 44th International Scientific Colloquium, Technical University of Ilmenau*, 1999.
- [57] A. Jaklic, A. Leonardis, and F. Solina. *Segmentation and Recovery of Superquadrics*. Kluwer Academic Publishers, ISBN: 0-7923-6601-8, 2000.
- [58] J.E. Dennis, Jr. and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, ISBN: 0-89871-364-1, 1983.
- [59] X. Jiang and H. Bunke. Edge detection in range images based on scan line approximation. *Elsevier; Computer Vision and Image Understanding*, 73(2):183–199, 1999.
- [60] O. Jokinen. Self-calibration of a light striping system by matching multiple 3-D profile maps. In *Proceedings of the IEEE Conference on 3-D Digital Imaging and Modeling*, pages 180–190, 1999.
- [61] D. Katsoulas. Reliable recovery of piled box-like objects via parabolically deformable superquadrics. In *Proceedings of the IEEE 9th International Conference on Computer Vision*, volume 2, pages 931–938, 2003.
- [62] D. Keren, D. Cooper, and J. Subrahmonia. Describing complicated objects by implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):38–53, 1994.
- [63] D. Keren and C. Gotsman. Fitting curves and surfaces with constrained implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1):31–41, 1999.
- [64] S.I. Kim and S.J. Ahn. Extraction of geometric primitives from point cloud data. In *Proceedings of the International Conference on Control, Automation and Systems*, 2005.

BIBLIOGRAPHY

- [65] A. Kochan. NOMAD welds small batches of large structures. *Emerald; Industrial Robot: An International Journal*, 31(1):26–29, 2004.
- [66] D. Kragic, M. Bjorkman, H.I. Christensen, and J.-O. Eklundh. Vision for robotic object manipulation in domestic settings. *Elsevier; Robotics and Autonomous Systems*, 52(1):85–100, 2005.
- [67] J. Krivic and F. Solina. Part-level object recognition using superquadrics. *Elsevier; Computer Vision and Image Understanding*, 95(1):105–126, 2004.
- [68] S. Kumar, S. Han, D. Goldgof, and K. Bowyer. On recovering hyperquadrics from range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1079–1083, 1995.
- [69] K.S. Kwok, C.S. Louks, and B.J. Driessen. Rapid 3-D digitizing and tool path generation for complex shapes. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 2789–2794, 1998.
- [70] C. Lawson and R. Hanson. *Solving Least Squares Problems*. New York: Prentice-Hall, ISBN: 0-89871-356-0, 1974.
- [71] S.H. Lee, H.K. Hong, and J.S. Choi. Assembly part recognition using part-based superquadric model. In *Proceedings of the IEEE TENCN*, volume 4, pages 479–482, 1999.
- [72] A. Leonardis and A. Jaklic. Superquadrics for segmenting and modeling range data. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(11):1289–1295, 1997.
- [73] K. Levenberg. A method for the solution of certain problems in least squares. *Brown University; Quarterly of Applied Mathematics*, 2:164–168, 1944.
- [74] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Elsevier; Artificial Intelligence*, 31(3):355–395, 1987.
- [75] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Springer; International Journal of Computer Vision*, 60(2):91–110, 2004.
- [76] C.P. Lu and G.D. Hager. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000.
- [77] G. Lukacs, R. Martin, and D. Marshall. Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 671–686, 1998.
- [78] D. Malin. *Aufnahme von Tiefenbildern nach dem Lichtschnittverfahren mit überlagelter Farbinformation*. Master Thesis, Vienna University of Technology, 2005.

BIBLIOGRAPHY

- [79] D. Marquardt. An algorithm for least squares estimation of nonlinear parameters. *SIAM; Journal of the Society of Industrial and Applied Mathematics*, 11:431–441, 1963.
- [80] D. Marshall, G. Lukacs, and R. Martin. Robust segmentation of primitives from range data in the presence of geometric degeneracy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):304–314, 2001.
- [81] J. Matas and O. Chum. Randomized RANSAC with $T_{d,d}$ test. *Elsevier, Image and Vision Computing*, 22(10):837–842, 2004.
- [82] N.L. Max and E.D. Getzoff. Spherical harmonic molecular surfaces. *IEEE Computer Graphics and Applications*, 8(4):42–50, 1988.
- [83] A.M. McIvor. Calibration of a laser stripe profiler. In *Proceedings of the IEEE Conference on 3-D Digital Imaging and Modeling*, pages 92–98, 1999.
- [84] R. Mohan and R. Nevatia. Using perceptual organization to extract 3D structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1121–1139, 1989.
- [85] J.J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. *Heidelberg: Springer Verlag; Numerical Analysis, Lecture Notes in Mathematics*, pages 105–116, 1995.
- [86] D.R. Myatt, P.H.S. Torr, S.J. Nasuto, J.M. Bishop, and R. Craddock. NAPSAC: High noise, high dimensional robust estimation - its in the bag. In *Proceedings of the 13th British Machine Vision Conference*, volume 2, pages 458–467, 2002.
- [87] Q.L. Nguyen and M.D. Levine. Representing 3-D objects in range images using geons. *Elsevier; Computer Vision and Image Understanding*, 63(1):158–168, 1996.
- [88] D. Nister. Preemptive RANSAC for live structure and motion estimation. In *Proceedings of the IEEE International Conference Computer Vision*, volume 1, pages 199–206, 2003.
- [89] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, ISBN: 0-387-98793-2, 1999.
- [90] M. Noessing. *Tiefenbildaufnahme nach dem Lichtschnittverfahren mittels Dreh/Schwenkeinheit*. Master Thesis, Vienna University of Technology, 2005.
- [91] M. Ohuchi and T. Saito. Three-dimensional shape modeling with extended hyperquadrics. In *Proceedings of the IEEE International Conference on 3-D Digital Imaging and Modeling*, pages 262–269, 2001.
- [92] E. Oja, L. Xu, and C.Y. Suen. Modified hebian learning for curve and surface fitting. *Elsevier; Neural Networks*, 5(3):441–457, 1992.

BIBLIOGRAPHY

- [93] A. Okabe, B. Boots, and K. Sugihara. *Spatial Tessellations: Concept and Applications of Voronoi Diagrams*. New York: Wiley, ISBN: 0471-93430-5, 1992.
- [94] M.M. Olsen and H.G.A. Petersen. A new method for estimating parameters of a dynamic robot model. *IEEE Transactions on Robotics and Automation*, 17(1):95–100, 2001.
- [95] B. Parhami. Voting algorithms. *IEEE Transactions on Reliability*, 43(4):617–629, 1994.
- [96] M. Peternell. Developable surface fitting to point clouds. *Elsevier, Computer Aided Geometric Design*, 21:785–803, 2004.
- [97] M. Peternell, H. Pottmann, and T. Steiner. Hough transform and Laguerre geometry for the recognition and reconstruction of special 3D shapes. Technical Report TR No.100, Institute of Geometric Modeling and Industrial Geometry, Vienna University of Technology, 2003.
- [98] A. Pichler, R.B. Fisher, and M. Vincze. Decomposition of range images using markov random fields. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 24–27, 2004.
- [99] A. Pichler, M. Vincze, H. Andersen, O. Madsen, and K. Häusler. A method for automatic spray painting of unknown parts. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 444–449, 2002.
- [100] H. Plach, C. Eitzinger, T. Berndorfer, and W. Van Dyck. Survey of multivariate calibration methods for pattern classification. In *Proceedings of the SPIE: Optomechatronic Systems III*, volume 4902, pages 521–527, 2002.
- [101] J. Ponce, D. Chelberg, and W.B. Mann. Invariant properties of straight homogeneous generalized cylinders and their contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(9):951–966, 1989.
- [102] H. Pottmann and J. Wallner. *Computational Line Geometry*. Springer, ISBN: 3-540-42058-4, 2001.
- [103] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, ISBN: 0-521-35465-X, 1988.
- [104] N.S. Raja and A.K. Jain. Recognizing geons from superquadrics fitted to range data. *Elsevier; Image and Vision Computing*, 10(3):179–190, 1992.
- [105] K. Rao and R. Nevatia. Computing volume descriptions from sparse 3-D data. *Springer; International Journal of Computer Vision*, 2(1):33–50, 1988.
- [106] I.D. Reid. Projective calibration of a laser-stripe range finder. *Elsevier; Image and Vision Computing*, 14(9):659–666, 1996.

- [107] D.W. Ritchie. *Parametric Protein Shape Recognition*. PhD Thesis, Departments of Computing Science and Molecular & Cell Biology, University of Aberdeen, 1998.
- [108] D.W. Ritchie. High order analytic translation matrix elements for real space six-dimensional polar fourier correlations. *Blackwell Publishing; Journal of Applied Crystallography*, 38:808–818, 2005.
- [109] C. Robertson and R.B. Fisher. Empirical calibration method for adding colour to range images. In *Proceedings of the 1st International Symposium on 3D Data Processing, Visualization & Transmission*, pages 558–561, 2002.
- [110] C. Robertson, R.B. Fisher, N. Werghi, and A.P. Ashbrook. Fitting of constrained feature models to poor 3D data. In *Springer LNCS, Adaptive Computing Design Manufacture*, pages 149–160, 2000.
- [111] M. Salganicoff, L.H. Lyle, and R. Bajcsy. Active learning for vision-based robot grasping. *Kluwer; Machine Learning*, 23(2):251–278, 1996.
- [112] M.J. Schlemmer, G. Biegelbauer, and M. Vincze. An integration concept for vision-based object handling: Shape-capture, detection and tracking. In *International Workshop on Intelligent Computing in Pattern Analysis / Synthesis*, volume LNCS 4153, pages 215–224, 2006.
- [113] T.W. Sederberg, D.C. Anderson, and R.N. Goldman. Implicit representation of parametric curves and surfaces. *Academic Press; Computer Vision, Graphics and Image Processing*, 28:72–84, 1984.
- [114] X. Sheng and M. Krömker. Surface reconstruction and extrapolation from multiple range images for automatic turbine blades repair. In *Proceedings of the IEEE IECON Conference*, volume 3, pages 1315–1320, 1998.
- [115] T.F. Shipley and P.J. Kellman. *Advances in Psychology: Form Fragments to Objects*, volume 130. Elsevier Science B.V., ISBN: 0-444-50506-7, 2001.
- [116] F. Solina and R. Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):131–147, 1990.
- [117] L. Staib and J. Duncan. Model based deformable surface finding for medical images. *IEEE Transactions on Medical Imaging*, 15(5):720–731, 1996.
- [118] W. Stöcher and G. Biegelbauer. Automated simultaneous calibration of a multi-view laser stripe profiler. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 4435–4440, 2005.
- [119] K.H. Strobl, W. Sepp, E. Wahl, T. Bodenmüller, M. Suppa, J.F. Seara, and G. Hirzinger. The DLR multisensory hand-guided device: The laser stripe profiler. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1927–1932, 2004.

BIBLIOGRAPHY

- [120] G. Kos T. Varady, P. Benko. Reverse engineering regular objects: simple segmentation and surface fitting. *World Scientific; International Journal on Shape Modeling*, 4:127–141, 1998.
- [121] Y. Takagi and G. Medioni. Volumetric description of dip solder joints from range data. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pages 146–151, 1998.
- [122] L. Tao, U. Castellani, and V. Murino. Robust 3d segmentation for underwater acoustic images. In *Proceedings of the 2ⁿ International Symposium on 3D Data Processing, Visualization & Transmission*, pages 813–819, 2004.
- [123] G. Taubin. Estimation of planar curves, surfaces and non-planar space curves defined by implicit equations, with application to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, 1991.
- [124] G. Taubin. An improved algorithm for algebraic curve and surface fitting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 658–665, 1993.
- [125] G. Taylor and L. Kleeman. Grasping unknown objects with a humanoid robot. In *Proceedings of the Australian Conference on Robotic and Automation*, pages 191–196, 2002.
- [126] G. Taylor and L. Kleeman. Robust range data segmentation using geometric primitives for robotic applications. In *Proceedings of the 9th International Conference on Signal and Image Processing*, pages 467–472, 2003.
- [127] G. Taylor and L. Kleeman. Integration of robust visual perception and control for a domestic humanoid robot. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 1, pages 1010–1015, 2004.
- [128] G. Taylor, L. Kleeman, and A. Wernersson. Robust colour range sensing for robotic applications using a stereoscopic light stripe scanner. In *Proceedings of the IEEE International Conference on Robots and Systems*, volume 1, pages 86–81, 2002.
- [129] B. Tiddeman, N. Duffy, G. Rabey, and J. Lokier. Laser-video scanner calibration without the use of a frame store. *IEE Proceedings - Vision, Image Signal Processing*, 145(4):244–248, 1998.
- [130] B. Tordoff and D.W. Murray. Guided sampling and consensus for motion estimation. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 86–92, 2002.
- [131] P.H.S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Elsevier, Computer Vision and Image Understanding*, 78:138–156, 2000.

BIBLIOGRAPHY

- [132] E. Trucco, R.B. Fisher, and A.W. Fitzgibbon. Direct calibration and data consistency in 3-D laser scanning. In *Proceedings of the 5th British Machine Vision Conference*, pages 489–498, 1994.
- [133] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, ISBN: 0-13-261108-2, 1998.
- [134] R.Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, 1986.
- [135] W.C. Tse and Y.H. Chen. A robotic system for rapid prototyping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 1815–1820, 1997.
- [136] M. Vincze, G. Biegelbauer, and A. Pichler. Painting parts automatically at lot size one. In *Proceedings of the IEEE Workshop on Robotic Sensing*, pages 35–40, 2004.
- [137] M. Vincze, A. Pichler, and G. Biegelbauer. Detection of classes of features for automated robot programming. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 151–156, 2003.
- [138] X. Weiwei, L. Weibin, and Y. Baozong. Superquadric-based geons recognition utilizing support vector machines. In *Proceedings of the IEEE International Conference on Signal Processing*, volume 2, pages 1264–1267, 2004.
- [139] S.B. Yacoub and C. Menard. Robust axis determination for rotational symmetric objects out of range data. In *Proceedings of the 21st Workshop of the Austrian Association for Pattern Recognition*, pages 197–201. ISBN: 3-7029-0436-0, 1997.
- [140] N. Yokoya and M.D. Levine. Range image segmentation based on differential geometry: A hybrid approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):643–649, 1989.
- [141] M. Zerrouh and R. Nevatia. Part-based 3D descriptions of complex objects from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):835–848, 1999.
- [142] Y. Zhang. Experimental comparison of superquadric fitting objective functions. *Elsevier; Pattern Recognition Letters*, 24(14):2185–2193, 2002.
- [143] Y. Zhang, A. Koschan, and M. Abidi. Superquadrics based 3D object representation of automotive parts utilizing part decomposition. In *Proceedings of the SPIE: International Conference on Quality Control by Artificial Vision*, volume 5132, pages 241–251, 2003.
- [144] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

BIBLIOGRAPHY

- [145] L. Zhou and C. Kambhamettu. Representing and recognizing complete set of geons using extended superquadrics. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 3, pages 713–718, 2002.
- [146] M. Zillich and E. Al-Ani. Camcalb: A user-friendly camera calibration software. In *Proceedings of the 28th Workshop of the Austrian Association for Pattern Recognition*, pages 111–115. ISBN: 3-85403-179-3, 2004.

Curriculum Vitae

Personal Information

Full Name: Georg Biegelbauer
Academic Degree: Dipl.-Ing.
Address: Erdbergstraße 52-60/1/5, 1030 Vienna, Austria
Date and place of birth: January 24th, 1973, Vienna, Austria
Citizenship: Austrian
Marital status: Unmarried
Children: None
Languages: German (mother tongue), English
Military Service: Oct. 1992 - Oct. 1993

Education

1979-1983: Elementary school, Groß-Enzersdorf.
1983-1987: Grammar school, Gänserndorf.
1987-1992: Technical collage, Vienna.
Jun. 1992: Matura, final examination passed with distinction.
1993-2001: Studies of Electrical Engineering at the Vienna University of Technology.
Nov. 2001: Graduation in Electrical Engineering at the Vienna University of Technology.

Work Experience

1989-1992: 5 month practical, electrician, Eskimo Iglo GmbH.
1993-1999: 5 month practical, logistics, ÖBB.
since Jan. 2002: Research assistant, Automation and Control Institute.
Oct. 2003 - Jul. 2004: Research employee, Profactor GmbH.

Project Work

Jan. 2001 - Oct. 2002: Research project FlexPaint (GROWTH GRD1-1999-10693).
Jan. 2002 - Jul. 2004: Research project FibreScope (CRAFT CR 1999-70631).
Oct. 2003 - Jul. 2004: Research project Lot Size One, Profactor GmbH.
since Oct. 2005: Research project REDUX, EADS (FFG, 810568/1553 SCK/SAI).

Publications

- G. Biegelbauer.** *Realisierung der Tiefenbildaufnahme nach dem Lichtschnittverfahren für ein vollautomatisches Lackiersystem.* Master Thesis, Vienna University of Technology, 2001.
- M. Vincze, A. Pichler, G. Biegelbauer, K. Häusler, H. Andersen, O. Madsen, and M. Kristiansen.** Automatic Robotic Spray Painting of Low Volume High Variant Parts. *Proceedings of the 33rd International Symposium on Robotics*, pages 012, 2002.
- M. Vincze, A. Pichler, and G. Biegelbauer.** Detection of Classes of Features for Automated Robot Programming. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 151–156, 2003.
- G. Biegelbauer, H. Nöhmeyer, C. Eberst, and M. Vincze.** Sensor Based Robotics for fully Automated Inspection of Bores at Low Volume High Variant Parts. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4852–4857, 2004.
- G. Biegelbauer and M. Vincze.** Robust and fully Automated Robotic Bore Inspection for High Variant Parts. *Proceedings of the IEEE International Workshop on Robot Sensing*, pages 9–13, 2004.
- M. Vincze, G. Biegelbauer, and A. Pichler.** Painting Parts Automatically at Lot Size One. *Proceedings of the IEEE International Workshop on Robot Sensing*, pages 35–40, 2004.
- G. Biegelbauer and M. Vincze.** Fast and Robust Bore Detection in Range Image Data for Industrial Automation. *Proceedings of the IEEE International Symposium on 3D Data Processing, Visualization & Transmission*, pages 526–533, 2004.
- W. Stöcher and G. Biegelbauer.** Automated Simultaneous Calibration of a Multi-View Laser Stripe Profiler. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4424–4429, 2005.
- R. Vogl, M. Vincze and G. Biegelbauer.** Finding Tables for Home Service Tasks and Safe Mobile Robot Navigation. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3025–3040, 2005.
- G. Biegelbauer, A. Pichler, M. Vincze, C. Nielsen, H. Andersen, and K. Häusler.** The Inverse Approach of FlexPaint: Automatic Generation of Robot Painting Motions for Unknown Parts. *IEEE Robotics & Automation Magazine - Special Issue: "Industrial Robotics Applications & Industry-Academia Cooperation in Europe, New Trends and Perspectives"*, 12(3):24–34, 2005.

BIBLIOGRAPHY

- G. Biegelbauer and M. Vincze.** Fast and Robust 3D Object Detection Using a Simplified Superquadric Model Description. *Proceedings of the 7th Conference on Optical 3-D Measurements Techniques*, pages 222–230, 2005.
- G. Biegelbauer and M. Vincze.** 3D Vision-Guided Bore Inspection System. *Proceedings of the IEEE International Conference on Computer Vision Systems*, pages 22–22, 2006.
- M.J. Schlemmer, G. Biegelbauer, and M. Vincze.** An Integration Concept for Vision-Based Object Handling: Shape-Capture, Detection and Tracking. *Proceedings of the International Workshop on Intelligent Computing in Pattern Analysis / Synthesis*, LNCS 4153, pages 215–224, 2006.