

JAN FISCHER

**A user-oriented, comprehensive system
for the 6 DoF recognition of arbitrary rigid
household objects**



Herausgeber:

Univ.-Prof. Dr.-Ing. Thomas Bauernhansl

Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl

Univ.-Prof. a. D. Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h.c. mult. Engelbert Westkämper

Jan Fischer

**A user-oriented, comprehensive system
for the 6 DoF recognition of arbitrary rigid
household objects**

Kontaktadresse:

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon 07 11 9 70-00, Telefax 07 11 9 70-13 99
info@ipa.fraunhofer.de, www.ipa.fraunhofer.de

STUTTGARTER BEITRÄGE ZUR PRODUKTIONSFORSCHUNG**Herausgeber:**

Univ.-Prof. Dr.-Ing. Thomas Bauernhansl
Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl
Univ.-Prof. a. D. Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h.c. mult. Engelbert Westkämper

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, Stuttgart
Institut für Industrielle Fertigung und Fabrikbetrieb (IFF) der Universität Stuttgart
Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW)
der Universität Stuttgart

Titelbild: © Jan Fischer

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über www.dnb.de abrufbar.

ISSN: 2195-2892

ISBN (Print): 978-3-8396-0891-3

D 93

Zugl.: Stuttgart, Univ., Diss., 2015

Druck: Mediendienstleistungen des Fraunhofer-Informationszentrum Raum und Bau IRB, Stuttgart
Für den Druck des Buches wurde chlor- und säurefreies Papier verwendet.

© by **FRAUNHOFER VERLAG**, 2015

Fraunhofer-Informationszentrum Raum und Bau IRB
Postfach 80 04 69, 70504 Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon 07 11 9 70-25 00
Telefax 07 11 9 70-25 08
E-Mail verlag@fraunhofer.de
URL <http://verlag.fraunhofer.de>

Alle Rechte vorbehalten

Dieses Werk ist einschließlich aller seiner Teile urheberrechtlich geschützt. Jede Verwertung, die über die engen Grenzen des Urheberrechtsgesetzes hinausgeht, ist ohne schriftliche Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Speicherung in elektronischen Systemen.

Die Wiedergabe von Warenbezeichnungen und Handelsnamen in diesem Buch berechtigt nicht zu der Annahme, dass solche Bezeichnungen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und deshalb von jedermann benutzt werden dürften. Soweit in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z.B. DIN, VDI) Bezug genommen oder aus ihnen zitiert worden ist, kann der Verlag keine Gewähr für Richtigkeit, Vollständigkeit oder Aktualität übernehmen.

GELEITWORT DER HERAUSGEBER

Produktionswissenschaftliche Forschungsfragen entstehen in der Regel im Anwendungszusammenhang, die Produktionsforschung ist also weitgehend erfahrungsbasiert. Der wissenschaftliche Anspruch der „Stuttgarter Beiträge zur Produktionsforschung“ liegt unter anderem darin, Dissertation für Dissertation ein übergreifendes ganzheitliches Theoriegebäude der Produktion zu erstellen.

Die Herausgeber dieser Dissertations-Reihe leiten gemeinsam das Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA und jeweils ein Institut der Fakultät für Konstruktions-, Produktions- und Fahrzeugtechnik an der Universität Stuttgart.

Die von ihnen betreuten Dissertationen sind der marktorientierten Nachhaltigkeit verpflichtet, ihr Ansatz ist systemisch und interdisziplinär. Die Autoren bearbeiten anspruchsvolle Forschungsfragen im Spannungsfeld zwischen theoretischen Grundlagen und industrieller Anwendung.

Die „Stuttgarter Beiträge zur Produktionsforschung“ ersetzt die Reihen „IPA-IAO Forschung und Praxis“ (Hrsg. H.J. Warnecke / H.-J. Bullinger / E. Westkämper / D. Spath) bzw. ISW Forschung und Praxis (Hrsg. G. Stute / G. Pritschow / A. Verl). In den vergangenen Jahrzehnten sind darin über 800 Dissertationen erschienen.

Der Strukturwandel in den Industrien unseres Landes muss auch in der Forschung in einen globalen Zusammenhang gestellt werden. Der reine Fokus auf Erkenntnisgewinn ist zu eindimensional. Die „Stuttgarter Beiträge zur Produktionsforschung“ zielen also darauf ab, mittelfristig Lösungen für den Markt anzubieten. Daher konzentrieren sich die Stuttgarter produktionstechnischen Institute auf das Thema ganzheitliche Produktion in den Kernindustrien Deutschlands. Die leitende Forschungsfrage der Arbeiten ist: Wie können wir nachhaltig mit einem hohen Wertschöpfungsanteil in Deutschland für einen globalen Markt produzieren?

Wir wünschen den Autoren, dass ihre „Stuttgarter Beiträge zur Produktionsforschung“ in der breiten Fachwelt als substanziell wahrgenommen werden und so die Produktionsforschung weltweit voranbringen.

Alexander Verl

Thomas Bauernhansl

Engelbert Westkämper

A user-oriented, comprehensive system for the 6 DoF recognition of arbitrary rigid household objects

Von der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik
der Universität Stuttgart

zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)

genehmigte Abhandlung

Vorgelegt von

Dipl.-Inform. Jan Fischer

aus Lichtenstein

Hauptberichter: Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl
Mitberichter: Univ.-Prof. Dr.-Ing. Heinz Wörn

Tag der mündlichen Prüfung: 2. Februar 2015

Institut für Steuerungstechnik der Werkzeugmaschinen
und Fertigungseinrichtungen der Universität Stuttgart

2015

Vorwort des Verfassers

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Produktionstechnik und Automatisierung der Fraunhofer Gesellschaft in Stuttgart. Mein besonderer Dank gilt Herrn Professor Dr.-Ing. Dr. h.c. Alexander Verl für die Unterstützung meiner wissenschaftlichen Tätigkeit und die Übernahme des Hauptberichts. Herrn Professor Dr.-Ing. Heinz Wörn danke ich für die Übernahme des Mitberichts zu dieser Arbeit.

Weiterhin danke ich Herrn Martin Hägele, Leiter der Abteilung Roboter- und Assistenzsysteme, der mich am Fraunhofer IPA eingestellt hat und somit den Grundstein für die Erstellung dieser Arbeit legte. Ein besonderer Dank gilt weiterhin Frau Dr. Birgit Graf, Gruppenleiterin Haushalts- und Assistenzrobotik, für die Möglichkeit der Bearbeitung inhaltlich geeigneter Projekte sowie der sorgfältige Durchsicht meiner Arbeit.

Für die angeregten und konstruktiven Diskussionen sowie der fachlichen Prüfung dieser Arbeit danke ich Herrn Richard Bormann. Luzia Schuhmacher möchte ich für die abschließende Korrektur meiner Arbeit danken. Des Weiteren danke ich den von mir betreuten Studenten, die in Form von Praktika und Abschlussarbeiten im Laufe der letzten Jahre zum Gelingen dieser Arbeit beigetragen haben. Weiterhin gilt mein Dank meinen Kollegen Georg Arbeiter, Christian Connette und Manuel Drust für die gute Zusammenarbeit, für zahlreiche motivierende Gespräche und kritische Fragen. Diese haben maßgeblich dazu beigetragen, die vorliegende Arbeit zu formen und zu ihrem Abschluss zu bringen.

Abschließend möchte ich meiner Frau Christina danken, die während der Erstellung dieser Arbeit, vor allem aber in den letzten Jahren vor Vervollständigung der Arbeit, viel Verständnis für die meist langen Wochenenden und Abende am Schreibtisch gezeigt hat.

Zusammenfassung

Ziel der Arbeit ist die Entwicklung eines intuitiv nutzbaren, umfassenden Systems zur Wahrnehmung von typischen Haushaltsgegenständen in Lage und Position. Im Rahmen dieser Arbeit wird der zugrundeliegende Wahrnehmungsprozess in die drei Teilgebiete Datenaufnahme, Objektmodellierung und Objekterkennung untergliedert. Mit dem Ziel, den Wahrnehmungsprozess in seiner Gesamtheit zu optimieren, werden spezifische Entwicklungen in den einzelnen Teilgebieten vorgestellt und evaluiert.

Als Grundlage der Wahrnehmung dienen korrespondierende Bilddaten von Farbkameras und 2.5-D Tiefendaten einer Tiefenbildkamera. Das Teilgebiet der Datenaufnahme wird oftmals nicht genauer untersucht, da im Allgemeinen angenommen wird, dass Kameradaten durch entsprechende Kamerasysteme unmittelbar zur Verfügung stehen. Jedoch ist es möglich, durch Verfahren der Sensordatenfusion, verschiedene Kamerasysteme zu kombinieren, um eine Verbesserung der Kameradaten z.B. hinsichtlich räumlicher Abdeckung und Genauigkeit der Tiefendaten zu erzielen. Im Rahmen dieser Arbeit wird ein Verfahren entwickelt, um die Daten eines Stereokamerasystems mit den Daten einer Tiefenbildkamera zu kombinieren. Dabei werden die Daten der Einzelsysteme durch Aufstellung einer gemeinsamen Kostenfunktion mittels *Belief Propagation* kombiniert. Es wird gezeigt, dass dadurch die räumliche Abdeckung und Genauigkeit der Tiefendaten im Vergleich zu den genutzten Einzelsystemen gesteigert werden kann.

Im Bereich der Objektmodellierung stellt diese Arbeit eine Methode vor, die es ermöglicht, Objekte intuitiv mittels eines Robotersystems einzulernen. Dabei wird das Objekt im Greifer des Roboters platziert, worauf dieser das Objekt autonom modelliert. Relevante Daten zum Greifen und Erkennen des Objektes werden hierbei berechnet und in einem Objektmodell abgespeichert. Zusätzlich wird das Einlernen von Objekten mittels eines Drehtellers sowie mittels eines Schachbretts vorgestellt, um das Erstellen von Objektmodellen auch ohne Robotersystem zu ermöglichen. Eine grundlegende Arbeit zur Modellierung von Objekten wurde bereits in [AFV10] publiziert. Im Rahmen dieser Arbeit wird dieses Verfahren unter Benutzung des *Bundle Adjustment* Algorithmus weiter entwickelt.

Des Weiteren entwickelt diese Arbeit zwei neue binäre Deskriptoren zur Modellierung und Erkennung von texturierten sowie texturlosen Objekten. Diese ermöglichen durch die Benutzung von einfachen Bit-Operationen das schnelle Berechnen von deskriptiven Merkmalen. In Bezug auf die Erkennung von texturierten Objekten wird ein neuer Deskriptor

vorgelegt. Dieser basiert auf den Entwicklungen von Rublee et al. [RRKB11], besitzt jedoch neben der Invarianz gegenüber Änderungen in der Orientierung und Beleuchtung auch Invarianz gegenüber Änderungen in der Skala. Teile dieser Arbeiten wurden bereits in [FABV12] publiziert. In Bezug auf die Erkennung von texturlosen Objekten stellt diese Arbeit einen histogramm-basierten Deskriptor vor. Dieser berechnet mittels binären Operationen 2-D und 3-D Informationen, welche gemeinsam zur Beschreibung von Objektmerkmalen verwendet werden. Teile dieser Arbeiten wurden in [FBAV13] veröffentlicht.

Im Teilgebiet der Objekterkennung werden die vorgestellten Deskriptoren zur Erkennung unterschiedlicher Objekte verwendet. Um die Erkennungsrate zu verbessern, werden im Verlauf der Datenassoziation räumliche Beschränkungen eingeführt. Dadurch wird die räumliche Ausdehnung eines Objektes explizit beachtet. Um texturlose Objekte zu erkennen, wird ein adaptives *Sliding Window* Verfahren entwickelt, welches die Größe des Suchfensters basierend auf den gemessenen Tiefendaten sowie der bekannten Größe des Objektes dynamisch bestimmt.

Die vorgestellten Algorithmen sind eingebettet in ein modulares Softwaresystem, welches auf dem Serviceroboter Care-O-bot[®] 3 sowie auf separaten Einzelsystemen lauffähig ist. Die Einzelkomponenten werden unter Verwendung von Standard-Datensätzen sowie selbst erstellten Aufnahmen von typischen Haushaltsgegenständen separat evaluiert.

Abstract

The objective of this thesis is to develop a model-based object recognition system for 6 DoF localization of typical rigid household objects that explicitly enables an intuitive teaching of new objects. When considering the perceptual process of object recognition in its entirety, it may be divided into the three main areas: data acquisition, object modeling and object localization. The different areas are examined individually and distinct contributions to each of them are presented and evaluated.

The originating conditions for the recognition process system are one-shot images of range and color data. Considering data acquisition, it is most often taken for granted that a sensor delivers directly 2.5D data or color information. However, when combining different sensor modalities, it is possible to exceed the data quality of a single sensor. The thesis follows this idea and presents a novel sensor fusion technique for data acquisition that combines the 2.5D input data from a stereo and a range imaging system.

Regarding object modeling, the thesis presents a method for dense object modeling directly on the robot using its manipulator and camera system. Additionally, two stand-alone training setups are introduced which avoid the explicit need of a robotic manipulator for object modeling. One is using a turn table to rotate the object in front of the camera system and the other one is using a chessboard where the camera is manually moved around a stationary object. Initial work conducted within the scope of this thesis and published in [AFV10] proposes a fastSLAM-based in-gripper object modeling approach which is able to cope with multi-occurrences of similar textures on the object's surface. This approach is further developed and the information filter is replaced by a *Bundle Adjustment* algorithm that enables a faster registration of the individual object views.

This thesis proposes two novel binary descriptors for textured and texture-less object modeling that enable the usage of rapid bit operations to accelerate the descriptor computations. When addressing textured objects, recent fast-to-compute descriptors achieving remarkable recognition rates have been presented. This thesis proposes a scale invariant extension of the binary feature descriptor ORB [RRKB11], which is fast to compute while still being as descriptive as SURF. The presented results have been published in [FABV12]. In order to distinctly describe texture-less objects, a global histogram-based descriptor is presented, that aggregates 2D and 3D gradient information from a local binary descriptor. Compared to the current state-of-the-art, the descriptor exhibits scale and rotation invari-

ance. Additionally, the underlying binary descriptor is computed faster than competing methods by the use of *dynamic programming*. The presented results have been published in [FBAV13].

In order to increase the robustness of texture-less object recognition, data association is subject to a spatial constraint to take account for the spatial expansion of an object. The thesis proposes an adaptive sliding window approach to build up a probability map for prominent object locations. Based on a non-maximum suppression algorithm, the dominant object locations are selected. The presented approach has been published in [FBAV13].

The different components have been integrated in a software framework for 6 DoF object recognition that has been implemented on the service robot Care-O-bot[®] 3 using the middleware ROS. The software components for data acquisition, object modeling and object recognition are evaluated individually using standard datasets and typical real world household objects like plates, bottles or cups.

Contents

Abbreviations and Symbols	xiii
List of Figures	xv
List of Tables	xxi
1. Introduction	1
1.1. Motivation	1
1.2. Problem description	2
1.3. Contributions	4
1.4. Requirements	5
1.5. Outline	8
2. State of the art	9
2.1. Data acquisition	9
2.1.1. Stereo vision	10
2.1.2. Time of flight	14
2.1.3. Low-cost structured light	15
2.1.4. Sensor fusion	16
2.2. Object modeling	18
2.2.1. Feature descriptors	19
2.2.2. Data association	22
2.2.3. Object modeling procedures	24
2.2.4. Simultaneous localization and mapping (SLAM)	25
2.3. Object recognition	26
2.3.1. 3D-3D pose estimation	27
2.3.2. Robust estimators	27
2.3.3. Object recognition frameworks	28
3. System design	30
3.1. Sensor fusion for data acquisition	30
3.2. Object modeling	31

3.3. Object recognition	33
3.4. Hardware setup	34
3.5. Software design	38
4. Sensor Fusion	40
4.1. Calibration	42
4.2. Pre-processing	44
4.3. Projection	45
4.4. Pixelwise matching costs	46
4.5. Cost aggregation	47
4.6. Post-processing	49
4.7. Evaluation	49
5. Object Modeling	55
5.1. Data acquisition	58
5.2. Modeling of textured objects	60
5.2.1. Local feature descriptor	60
5.2.2. Data association	63
5.2.3. Model-based object modeling	64
5.2.4. Post-processing	68
5.2.5. Evaluation	69
5.3. Modeling of texture-less objects	79
5.3.1. Local feature descriptor	81
5.3.2. Dynamic programming for rapid descriptor computation	85
5.3.3. Appearance-based object modeling	86
5.3.4. Evaluation	90
6. Object Recognition	91
6.1. Data acquisition	93
6.2. Recognition of textured objects	93
6.2.1. Pre-processing	93
6.2.2. Data association and clustering	94
6.2.3. Pose estimation	96
6.2.4. Evaluation	96
6.3. Recognition of texture-less objects	105
6.3.1. Adaptive sliding window for data association	106
6.3.2. Voting map for pose estimation	107
6.3.3. Evaluation	108

7. Conclusion and Outlook	117
A. Glossary	121
Bibliography	123

Abbreviations and Symbols

Abbreviations

nD	n -dimensional
BoW	Bag-of-words
CAD	Computer aided design
CCD	Charge coupled device
CMOS	Complementary metal oxide semiconductor
DoF	Degree of freedom
IR	Infrared
k-d tree	k-dimensional tree
LIDAR	Light detection and ranging
LMA	Levenberg-Marquardt algorithm
Lo-RanSaC	Local optimized RanSaC
LSH	Local sensitivity hashing
PnP	Perspective-n-Point
ProSaC	Progressive Sample Consensus
RanSaC	Random Sample Consensus
RGB-D	Red, green, blue, and distance
ROI	Region of interest
ROS	Robot Operating System
SBA	Sparse bundel adjustment
SIMD	Single instruction, multiple data
SLAM	Simultaneous Localization and Mapping
SSE	Streaming SIMD Extensions
ToF	Time of flight

Symbols

$\alpha, \beta \in \mathbb{R}$	Angle of view [rad]
$\mathbf{a}_j \in \mathbb{R}^6$	Pose of camera j
$\mathbf{b}_i \in \mathbb{R}^3$	Object-centric position of feature point i
\mathbf{M}	3×3 intrinsics matrix
\mathbf{M}_{rect}	4×4 stereo rectification matrix

$\mathbf{o} \in \mathbb{Z} \times \mathbb{Z}$	Principle point ([px], [px])
$\mathbf{P}, \mathbf{P}' \in \mathbb{R}^3$	3D points in physical space
$\mathbf{p}, \mathbf{p}' \in \mathbb{Z} \times \mathbb{Z}$	2D points on the image plane ([px], [px])
$\mathbf{r} \in \mathbb{R} \times \mathbb{R}$	2D direction vector in image space ([px], [px])
\mathbf{R}	3×3 rotation matrix
\mathbf{T}	1×3 translation vector
$\mathbf{v} = (v_0, v_1, \dots, v_k)$	k-dimensional descriptor, $v_i \in \mathbb{R}$
$\mathbf{x}_{ij} \in \mathbb{R}^3$	Observation of feature point i on the camera image j
A, B	Sets of descriptors
$B \in \mathbb{R}$	Length of baseline [mm]
$C \in \mathbb{R}^6$	Origin of the camera coordinate system
$D \in \mathbb{N}^{w \times h}$	Disparity image
$d \in \mathbb{R}$	Distance/disparity [cm]/[px]
$f \in \mathbb{N}$	Focal length [px]
I	2D image
$k: \mathbb{R} \rightarrow \mathbb{R}$	Kernel function
$m \in \mathbb{N}$	Number of different camera images from an object
$n \in \mathbb{N}$	Number of 3D feature points
N_p	Set of neighbors relative to p
$O, O' \in \mathbb{R}^6$	Origin of an arbitrary coordinate system
$t \in \mathbb{R}$	Threshold
$u, v \in \mathbb{Z}$	Image coordinates along the image axis [px]
$W \in \mathbb{R}^6$	Origin of the world coordinate system
$w, h \in \mathbb{N}$	Width and height of a 2D image [px]
x, y	Feature points represented by a 2D image point \mathbf{p} and a k-dimensional image descriptor \mathbf{v}
$z \in \mathbb{R}$	Distance [cm]

List of Figures

1.1. Schematic overview of the three core parts for visual perception. Data acquisition provides the sensory information by a camera system (Figure 1.1(a)). Object modeling is devoted to the representation of the object. It extracts 3D and 2D cues from the individual images and creates an object model to uniquely describe the object's shape and texture (Figure 1.1(b)). Object recognition and localization determines the 6 DoF pose for each detected object by matching the 3D and 2D cues between the scene image and the model (Figure 1.1(c)).	3
2.1. 2.5D range image from a single recording: The frontal view on the physical scene in Figure 2.1(a). The missing 3D data, that could not be captured by the camera, is clearly visible when changing the viewpoint as shown in Figure 2.1(b).	10
2.2. Figure 2.2(a) shows a stereo camera setup from the mobile service robot Care-O-bot [®] 3 using two color cameras mounted on the outer sides of the sensor head. Figure 2.2(b) illustrates the principle of triangulation using a standard pinhole camera model.	11
2.3. Figure 2.3(a) shows a ToF camera setup using a SwissRanger [™] SR3000 camera mounted on the sensor head of a robot. Figure 2.3(b) illustrates the measurement principle. ToF cameras emit modulated near infrared light to illuminate a scene. The reflection of the modulated light from the surface of the scene point P is collected by an infrared camera in a CMOS matrix. In order to compute the distance to the reflecting surface, the returning signal is compared to the camera's source modulation to compute the phase shift. The distance to the reflecting surface is given by its linear dependency to the phase shift.	14
2.4. Range imaging sensors based on the ToF principle manufactured by MESA Imaging and PMD[vision] [®]	15
2.5. Structured light camera setup using the Kinect [™] camera (Figure 2.5(a)) and its measurement principle (Figure 2.5(b)).	16
2.6. The principle of an integral image.	19

2.7.	Partition of a 2-dimensional feature space using the feature descriptors defined by $M = \{(1, 2), (2, 3), (3, 6), (4, 5), (7, 8), (8, 9)\}$	23
3.1.	Hardware design of the service robot Care-O-bot [®] 3. The first two images show the robot's cover. It is flexible in the range of the torso to allow the robot to express itself by gestures. The other images show the robot's hardware design below the cover.	34
3.2.	Stand-alone hardware setups without using a robotic platform.	35
3.3.	Sensor setups for the service robot Care-O-bot [®] 3 with different active range cameras.	35
3.4.	Characteristics of the stereo vision setup using a focal length $f = 1450$ pixel and a baseline of $B = 119$ mm. The vertical black lines indicate the desired measurement range.	36
3.5.	Top view on the sensor setup using the ToF camera SwissRanger [™] 4000 for active range sensing. A indicates the sensors' common field of view.	37
3.6.	Software architecture for object recognition and localization using a stereo and active range camera.	38
4.1.	Schematic overview of the individual processing steps for sensor fusion.	41
4.2.	Visualization of intrinsic and extrinsic camera parameters, which are determined through calibration.	42
4.3.	Pre-processing the raw range data delivered by a ToF sensor using wavefront propagation. The scene image has artificially been rotated by 90° for a better presentation of the filtering effects.	45
4.4.	Projection of filtered ToF data onto the high resolution greyscale image.	46
4.5.	Disparity images of stereo vision and the sensor fusion algorithm.	50
4.6.	Images of a planar surface: the original color image (first column), disparity from passive stereo vision (second column), disparity from the Microsoft [®] Kinect [™] sensor (third column), and disparity from the proposed sensor combination algorithm (forth column).	51
4.7.	The first column shows the color images, the second column disparity images from stereo vision and the third column disparity from the proposed sensor fusion algorithm.	52
4.8.	The Middlebury stereo dataset: The first column shows the original color images, the second column disparity images from stereo vision and the third column disparity from the proposed sensor combination algorithm. Beginning with the first row, the datasets are labeled with the names <i>Venus</i> , <i>Teddy</i> , and <i>Cones</i>	54

5.1.	Schematic overview of the individual processing steps for object modeling.	56
5.2.	(a) Service robot recording object images from different viewpoints. (b) One of the single object views recorded by the RGB-D camera for object model creation. (c) Extracted feature points. (d) 3D object model with extracted surfaces using sparse bundle adjustment and Delaunay triangulation.	57
5.3.	Data acquisition with different hardware setups.	58
5.4.	Inferring odometry data from the in-gripper coordinate systems for different object views. The transformations $\mathbf{R}_G^i, i \in \{1, 2, 3\}$ are given by the calibration of the robotic manipulator to the camera system. Therefore, the coordinates of the recorded 3D points $\mathbf{P}_G = (x, y, z)$ within the in-gripper coordinate system \mathbf{G} are transferred into the common coordinate system \mathbf{O}_0 by $\mathbf{P}_{O_0} = \mathbf{M}_i^0 \mathbf{M}_G^i \mathbf{P}_G$, where \mathbf{M}_a^b is a standard 4×4 transformation matrix consisting of the 3×3 rotation \mathbf{R}_a^b and the 3×1 translation vector \mathbf{T}_a^b	59
5.5.	Computing a single binary BRIEF descriptor entry by comparing the sum of intensity values from the local regions around \mathbf{p}'_0 and \mathbf{p}''_0 within the patch centered at \mathbf{p} . The patch size of the affected image region is $m \times m$ pixel and the local regions are computed using a kernel size of $n \times n$ with $n < m$ and $n, m \in \mathbb{Z}$. In the standard implementation $m = 48$ and $n = 9$. The overall descriptor is computed by repeating the comparison for different pairs of \mathbf{p}'_j and \mathbf{p}''_j	61
5.6.	Computing a STAR feature point by comparing the sum of intensity values from the areas denoted by A_1 and A_2	62
5.7.	Multi-occurrences of feature point p on a single object. A single feature point is associated to two different locations on the object image.	64
5.8.	Visualization of the Pseudo-Hubert kernel function k from (5.13) for different values of the kernel parameter c	68
5.9.	Dataset from the Oxford's Visual Geometry Group.	70
5.10.	The boat image samples from the Oxford's Visual Geometry Group dataset. Image transformations represent changes in scale and rotation.	71
5.11.	Comparing the matching accuracy of the sORB feature descriptor depending on varying image distortions. The number at the end of the feature names denotes the number of bits that have been used to represent the descriptor.	74
5.12.	The boat image samples from the Oxford's Visual Geometry Group dataset. Image transformations represent changes in rotation.	75
5.13.	Comparing the matching accuracy of the sORB feature descriptor with respect to varying scales and orientations.	75

5.14. Schematic setup of the simulation environment to evaluate the performance of the sparse bundle adjustment algorithm. The simulated object with coordinate system G and data points $\hat{\mathbf{b}}_i$ is placed in the center. It is recorded from m different camera configurations O_j in a circular arrangement around the object. Gaussian noise is added to the ideal data points $\hat{\mathbf{b}}_i$ and to the camera poses $\hat{\mathbf{a}}_j$ in order to mimic the measurement noise appearing in a real setup.	77
5.15. Error rate plot.	79
5.16. Object models reconstructed using sparse bundle adjustment.	80
5.17. Layout of the feature descriptor to compute the binary descriptor values for the feature point p located at the black cross in the center of the colored squares. The dashed lines indicate corresponding pixel positions p_r^i and p_g^i from the red and green square, at which a function $f(p_g^i, p_r^i)$ is evaluated. The value of f is thresholded to compute a binary value $\tau(f(p_g^i, p_r^i)) \in \{0, 1\}$	81
5.18. Output of the proposed method for 2D gradient computation using discrete angles of 0° , 45° , 90° and 135° . The different gradient orientations are encoded with different colors on the right image.	83
5.19. Illustration of the principles for 3D gradient estimation.	84
5.20. Output of the proposed method for 3D gradient computation using discrete angles ranging from 0° to 315° in steps of 45°	86
5.21. Computation of the binary descriptor values for the pixel location at the black cross in the center of the two squares using dynamic programming. Considering Figure 5.21(a), all descriptors to the left of the current pixel have already been evaluated. For Figure 5.21(b), additionally the descriptors of the top row have been computed.	87
5.22. Recorded training sequence for the car object. The ground truth pose of the object is given by the checkerboard. A virtual box in 3D space is placed around the car to separate the object data (native color values within the red rectangle) from the background.	87
5.23. One local histogram of gradient orientations is computed for each of the four areas surrounded by the green rectangles and one local histogram for the area surrounded by the red rectangle. All rectangles are of the same size. The main 3D gradient orientation from the histogram describing the red area is used for descriptor alignment in order to achieve rotation invariance.	89
6.1. Schematic overview of the individual processing steps for object recognition.	92
6.2. Image pre-processing for object recognition using distance-based filtering and wavefront propagation on a disparity image from stereo vision.	94

6.3.	Data association of feature point x_1 from the scene with feature points y_1, y_2, y_3 , and y_4 from the object model. The associations are classified based on their matching distance. Strong matches with a high similarity are collected within the set $A = \{(x_1, y_3), (x_1, y_4)\}$, e.g. as y_3, y_4 , and x_1 are all associated to sharp angles of the green shape, associations with a larger descriptor distance are placed in the set $B = \{(x_1, y_1), (x_1, y_2)\}$	95
6.4.	ProSaC sampling of feature point correspondences. The object model i to be recognized is illustrated on the right. On the left, the gray dots denote feature points not corresponding to the object model and the red dots denote feature points that could be matched with feature points from the object model i and therefore constitute elements of A_i or B_i . Let $x_1 \in \bar{A}_i$, then all correspondences from A_i or B_i that are related to a feature point of the local neighborhood from x_i (depicted by the bright red dots) are subject to ProSaC sampling.	97
6.5.	25 object models from the IPA dataset for textured object recognition.	98
6.6.	Excerpt from the training set for object recognition.	99
6.7.	Excerpt from the testing set for object recognition showing feature points (blue dots), the ground truth pose (coordinate system), and the recognized object pose (bounding box).	99
6.8.	Evaluation of the marker's rotational accuracy.	101
6.9.	Evaluation of the marker's translational accuracy.	101
6.10.	Rotational accuracy for textured object recognition at a distance of 0.7 m.	102
6.11.	Rotational accuracy for textured object recognition at a distance of 1.7 m.	102
6.12.	Rotational accuracy for textured object recognition at a distance of 0.7 m and 1.7 m.	103
6.13.	Translational accuracy for textured object recognition at a distance of 0.7 m.	103
6.14.	Translational accuracy for textured object recognition at a distance of 1.7 m.	104
6.15.	Translational accuracy for textured object recognition at a distance of 0.7 m and 1.7 m.	104
6.16.	Recognition results with colored boxes indicating the detected pose of an object and the object type.	106
6.17.	Principle of the adaptive sliding window approach.	107
6.18.	Recognition of the <i>Car</i> object using the proposed voting map representation for pose estimation.	108
6.19.	Images taken from the testing dataset for texture-less object recognition. The ground truth pose of the object is given by the checkerboard in order to evaluate the recognition rate.	108
6.20.	Close-up view on the recall-precision plot for different values of t_{2D} from (5.18).	109

6.21. Distribution of L2 distances (first column) and recall-precision plots (second column) for different objects.	110
6.22. 25 object models from the IPA testing dataset for texture-less object recognition.	111
6.23. Rotational accuracy for texture-less object recognition at a distance of 0.7 m.	112
6.24. Rotational accuracy for texture-less object recognition at a distance of 1.7 m.	112
6.25. Rotational accuracy for texture-less object recognition at a distance of 0.7 m and 1.7 m.	113
6.26. Translational accuracy for texture-less object recognition at a distance of 0.7 m.	113
6.27. Translational accuracy for texture-less object recognition at a distance of 1.7 m.	113
6.28. Translational accuracy for texture-less object recognition at a distance of 0.7 m and 1.7 m.	114
6.29. Recognition results shown by colored projections of the best fitting object view onto the scene image	116

List of Tables

1.1. System requirements.	5
1.2. Requirements for data acquisition.	6
1.3. Requirements for object modeling.	7
1.4. Requirements for object recognition.	8
3.1. Requirement matching for data acquisition.	31
3.2. Requirement matching for object modeling.	32
3.3. Requirement matching for object recognition.	33
4.1. Disparity densities in relation to different wavefront propagation ranges for the ToF measurements.	50
4.2. Average disparity error and density on the plane dataset.	52
4.3. Average disparity error on the Middlebury stereo dataset.	53
5.2. Computation time and throughput on a 640 x 480 image for descriptor computation.	76
5.4. Computation time on a 640 x 480 image for 2D and 3D gradient estimation of the original approach by Hintersoisser et al. and the proposed approach using dynamic programming.	90
6.1. Average Euclidean and angular deviation of the error in position and orientation when using SURF and sORB for textured object recognition.	105
6.2. Comparing true positives and false positives for SURF and sORB on the IPA dataset consisting of 1800 different images at a distance of 0.7 m and 1.7 m.	105
6.3. Average Euclidean and angular deviation of the error in position and orientation for texture-less object recognition.	114
6.4. True positive and false negative rate on the IPA dataset consisting of 1800 different images at a distance of 0.7 m and 1.7 m.	115
6.5. Absolute and relative number of true positives with respect to a varying degree of occlusion for texture-less object recognition.	115
7.1. Validation of the system requirements.	117
7.2. Validation of the requirements for data acquisition.	118

7.3. Validation of the requirements for object modeling.	119
7.4. Validation of the requirements for object recognition.	120

1. Introduction

It has always been a desire of robotics research to develop a sophisticated personal robot acting as a household companion. The vision aims at providing assistance at daily housework tasks like tidying up, cleaning or cooking. Given the context of an aging population in Europe and the limited amount of trained care personnel, this vision has also become a promising approach to face the expected deterioration of elderly care by retaining the independence of elderly people through daily assistance. This makes the vision of a household companion not only economically, but also socially reasonable.

Despite the obvious need, commercially available robotic systems are still highly specialized to specific tasks like vacuum cleaning or lawn mowing. These systems possess only limited capabilities, which are restricted to fixed routines. Not equipped with any higher level reasoning capabilities for independent decision making, unexpected and unknown situations force them to abort their operation to request user interaction. Therefore, current systems are not able to significantly ease the daily life of their human operators.

Robotics research has realized these shortcomings and agrees that the only feasible solution provides *cognitive robotics*, which devotes its research towards the improvement of cognitive abilities. This endows the robot to learn and reason in the presence of unknown situations in order to react autonomously in an intelligent manner.

1.1. Motivation

Perception is the core issue of cognitive robotics. It enables the robot to comprehend its environment and provides a symbolic representation for the cognitive processes. Humans possess a sophisticated perception system, which enables them to robustly determine the presence of a specific object within their environment. During their early infancy, children already develop higher level generalization capabilities to cope with variations in an object's appearance like scale, background, illumination or object pose. Robotics research has the ambition to understand and mimic these superior human capabilities, enabling the recognition of arbitrary objects in arbitrary contexts.

While the recognition of arbitrary objects in well defined industrial environments is considered to be solved e.g. the inspection of parts on a conveyor belt, modern algorithms are still unable to robustly cope with unknown dynamic situations. This statement is

also supported by a recent study commissioned by the German Ministry of Education and Research (BMBF) about the profitability of new service robotic applications and their means for robotic development [HBB⁺11]. It denotes perception as the central and most important technology when thinking about commercializing service robots. Based on the evaluation of eleven use cases from different domains ranging from robotic milking to health care assistance, perception proves to be the most widely used component. However, the study criticizes that the availability of a robust perception system for unknown dynamic environments is still far from being reality.

The difficulties researchers experience in developing robust recognition systems are resulting from the wide range of varying conditions that must be coped with. A perfect recognition system should be able to distinguish between an arbitrary number of objects in reasonable time. This requires a scalable system design whose computation time and memory space should be bounded by a linear function depending on the number of objects. Perception must succeed under variable indoor and outdoor conditions, potentially dealing with dirty environments. In the case of visual perception, this induces the need to be invariant against image transformations like translation, orientation or scaling. Other challenges are originating from undefined lighting conditions or partly occluded objects and varying backgrounds. Especially with growing scene complexity i.e. having too much or too few structure, recognition rates are usually decreasing. Furthermore, for a precise object manipulation, the localization of an object must be accurate in its position and its orientation resulting in 6 degrees of freedom (DoF). Depending on the application, accuracy may range from 1 cm up to below 1 mm for position estimates and up to below 1° for orientation estimates. Finally, the dimensions of an object may vary dramatically, when thinking of tiny objects like screws or pens to large objects like airplanes or complete buildings.

1.2. Problem description

Visual perception can generally be divided into three main parts: data acquisition, object modeling, and the actual object recognition and localization (Figure 1.1). Each part of its own embraces a separate research area and may be considered independently from the others.

Visual perception always starts with some sort of *data acquisition* (Figure 1.1(a)), which is devoted to the creation of a digital representation from the real world. In this process, particular attention is paid to the selection of an appropriate sensor system. This is crucial for the quality of the digital representation, as different sensors have different properties e.g. a LIDAR scanner is very accurate but usually cost intensive and slow compared to ToF sensors that have a lower resolution and less accuracy but operate at 30 Hz. By the combination of different sensor modalities, it is possible to exploit their advantages and

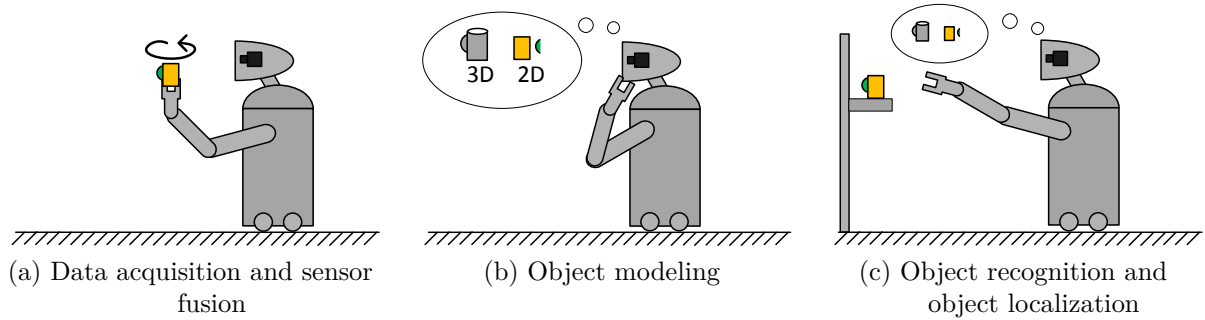


Figure 1.1.: Schematic overview of the three core parts for visual perception. Data acquisition provides the sensory information by a camera system (Figure 1.1(a)). Object modeling is devoted to the representation of the object. It extracts 3D and 2D cues from the individual images and creates an object model to uniquely describe the object’s shape and texture (Figure 1.1(b)). Object recognition and localization determines the 6 DoF pose for each detected object by matching the 3D and 2D cues between the scene image and the model (Figure 1.1(c)).

create a common superior vision setup that outperforms the capabilities of the individual sensors. The algorithmic process of combining the data from the different sensors into a single output is known as *sensor fusion*.

Object modeling creates an abstract representation of the object, representing its shape and texture (Figure 1.1(b)). It is computed before the actual recognition process begins e.g. by placing the relevant object in a robot’s gripper. Distinct object information is extracted from varying object views by repositioning the gripper in front of the camera system. The 2D and 3D image cues from the object’s surface are identified and represented by features. A distinction is usually drawn between appearance-based and model-based methods. Appearance-based methods take a series of images under variations in lighting, viewing direction, and scale to represent the characteristic views or aspects of an object. The problem of object recognition is then defined by finding the view that best matches the given image data. Model-based methods explicitly specify a 3D model of the object’s shape, from which any desired projections may be derived for object recognition.

Object recognition (Figure 1.1(c)) determines the existence of an object in an unknown scene, whereas *object localization* determines the object’s precise 6 DoF pose i.e. its position and orientation. To solve the object localization problem, correspondences between points of the object model and the scene must be established. This is accomplished by matching prominent image parts based on which the object location in the scene is inferred.

1.3. Contributions

The objective of this thesis is to develop a perception system for 6 DoF localization of typical rigid household objects that enables an intuitive teaching of new objects. The system operates on single images and is designed to handle oclusions and multi-occurrences of objects under varying lighting conditions. This thesis contributes to all three parts of the perceptual process as follows:

Robust object modeling demands accurate and dense 3D data in order to create precise 3D object models for object recognition and grasping. To improve existing procedures for 3D data acquisition, this thesis proposes a novel method for sensor combination on a stereo and a range camera system. By calibrating the two sensor systems to each other, valid measurements from the range sensor are converted to disparity guesses within the stereo system. The disparity guesses from the range data constrain the correspondence search results from the stereo matching algorithm. It is shown, that the proposed method effectively enhances the results from stereo vision, especially in structureless areas where stereo correspondence search fails.

The contributions to object modeling are divided in the modeling of textured and texture-less objects. Concerning textured objects, the thesis proposes a scale invariant feature descriptor termed sORB for rapid object recognition, which is evaluated against the performance of state of the art feature descriptors. The descriptors are applied for 3D object modeling using sparse bundle adjustment. Concerning texture-less object recognition, the thesis proposes a novel global histogram-based descriptor which is based on the aggregation of local point feature descriptors to capture the distribution of 2D and 3D gradient directions. Compared to existing approaches, a rapid computation of the descriptor is enabled by the use of dynamic programming and integral images.

The contributions to object recognition are strongly related to the concepts proposed for object modeling. Here, the proposed feature descriptors are recognized within an unknown scene and matched with the object models. For textured object recognition, the thesis presents a multi-hypothesis sampling of matching descriptors based on a nearest-neighbor radius search using ProSaC sampling, which is spatially constrained. In order to recognize texture-less objects, the thesis proposes an adaptive sliding window approach that rescales the sliding window dimensions based on the measured range value to achieve scale invariance.

The proposed methods are implemented and evaluated using the camera system of the service robot Care-O-bot[®] 3. Results of the proposed methods have been published in [FSV10, AFV10, FAV11, FRWV11, FABV12, FBAV13].

1.4. Requirements

The requirements for the proposed object recognition system are derived from the demands to perceive objects in a typical household environment. The requirement definitions are restricted to indoor applications. The key requirements are divided into separate sections, individually addressing one of the three stages data acquisition, object modeling and object recognition as well as the overall system design.

The system design of the proposed perception framework is required to fit to the hardware and software requirements of a robotic household companion. This includes an implementation of appropriate software interfaces for seamless integration and the design of an appropriate sensor layout. It also limits the available hardware to a standard industrial computer with limited memory resources. Additionally, the system layout is required to enable a stand-alone application of the perception framework and its individual parts, in order to allow its application also in other domains. This directly leads to the requirement of a modular software architecture, where the components are decomposable without any strict interdependencies between the modules for data acquisition, object modeling and object recognition. In order to guarantee reliability and robustness, the recognition system is required to cope with arbitrary object poses, occlusion and varying lighting conditions. The mentioned requirements are summarized in Table 1.1.

ID	Criteria	Requirement
R1	Target platform	Service robot and stand-alone vision system
R2	Environment conditions	Indoor, including arbitrary object poses, occlusion and varying lighting conditions
R3	Software architecture	Modular software architecture that enables an independent application of the data acquisition, object modeling and object recognition modules

Table 1.1.: System requirements.

Data acquisition is required to provide range and color information for object modeling and object recognition. Ideally, the range data is available for each image pixel. However, this density is usually not reachable due to reflections or occlusions, which prevent an inference of range information depending on the applied sensor technology. Additionally, when using a multi camera setup, range can only be inferred within the field of view of all cameras. Therefore, within the scope of this thesis it is required that for at least 90% of all image pixel within the field of view of the camera setup a valid range and color value is computed. The maximally allowed measurement distance between the camera system and an object to be recognized is of minor importance when using a mobile robot, as

it is capable of moving towards the object in order to decrease the measurement range. However, this does no longer apply when using a stand-alone vision system. Therefore, a desired measurement range of 70 cm to 170 cm is specified as a further requirement. It is based on the considerations, that on the one hand the used active range sensors of the vision system do not allow the acquisition of range values closer than 70 cm and on the other hand, that for most applications the position of the camera system can be manually chosen not to exceed 170 cm. In order to accurately recognize the pose of an object, it is required that the average measurement error within the measurement range does not exceed 2 cm. Furthermore, the computation of the 3D range and 2D color information must attain at least 5 Hz at VGA resolution images. This value has proven to be acceptable considering the much larger time needed for the actual object recognition.

Considering the hardware setup of the sensor system, it is required to be mountable on a mobile robot e.g. on the service robot Care-O-bot[®] 3. This requires a compact hardware layout, not exceeding 15 cm × 7 cm × 10 cm (width×length×height). Finally, the software should not be fixed to a camera type of a specific manufacturer. A seamless integration of new sensors is required in order to take advantage of future developments in the camera technology e.g. leading to a decrease in pricing or an increase in accuracy. A summary of the requirements for data acquisition is given by Table 1.2.

ID	Criteria	Requirement
R5	Output	Providing range and color information
R6	Density	Retrieving 3D data for at least 90% of all image pixel within the field of view of the camera setup
R7	Measurement range	70 cm to 170 cm
R8	Accuracy	Achieving an average measurement error below 2 cm within the measurement range
R9	Computation time	5 Hz at VGA resolution
R10	Sensor layout	Compact and mountable on a mobile robot, not exceeding 15 cm × 7 cm × 10 cm (width×length×height)

Table 1.2.: Requirements for data acquisition.

Object modeling is required to create dense 3D object models enabling the computation of suitable grasps. Sparse models with missing 3D data would lead to deformed object shapes leading to the inference of incorrect grasps. The accuracy of the model shape is required to be below an average measurement error of 3 mm. This guarantees that even delicate grasps may be applied to manipulate the object. A formal restriction on the object dimensions is defined by a maximal bounding box of 10 cm × 10 cm × 30 cm (width×length×height) that encloses the object. This limitation is mainly based on the observation that most

objects within a household environment do not exceed these dimensions in order to be easily manipulatable by humans. A lower bound on the object dimensions is given by the resolution of the used camera sensors. For the presented setup it is given by a minimal enclosing bounding box of $1\text{ cm} \times 1\text{ cm} \times 2\text{ cm}$. The large variety of objects in a household environment requires the system to be easily adaptable by teaching the system new objects. Therefore, a major requirement is usability even by non-expert users. Table 1.3 summarizes the requirements concerning object modeling.

ID	Criteria	Requirement
R11	Output	Dense 3D model enabling the computation of suitable grasps
R12	Accuracy	Model accuracy below an average error of 3 mm
R13	Object dimensions	Not larger than $10\text{ cm} \times 10\text{ cm} \times 30\text{ cm}$ and not smaller than $1\text{ cm} \times 1\text{ cm} \times 2\text{ cm}$ (width \times length \times height)
R14	Object type	Rigid objects
R15	Usability	Easy introduction of new objects by non-expert users or the robot itself

Table 1.3.: Requirements for object modeling.

Object recognition is required to compute the 6 DoF pose of an object, which is necessary to manipulate it with the robotic gripper. Within the scope of this thesis, the number and type of objects to be recognized encompasses at least 25 textured and 25 texture-less objects. Furthermore, within the scope of this thesis, all objects are required to be rigid and not deformable. The required recognition range is given by the required measurement range of the sensor system, which is 70 cm to 170 cm. The recognition time is required to not exceed 5 s in order to allow a fast execution of the perception task. When grasping an object, the robot is close to the object and usually does not exceed a distance of 0.7 m. Also the mechanical design of common robotic grippers is still rather bulky. Therefore, it is sufficient to achieve a recognition accuracy that does not exceed an average angular deviation of 6° and an Euclidean deviation of 9 mm at a distance of 0.7 m. This suffices the robot to apply common power grasps in order to pick objects. For precision grasps, a more precise localization of the object is desirable. The requirements for object recognition are summarized by Table 1.4.

ID	Criteria	Requirement
R16	Output	6 DoF pose of the object
R17	Object types	Rigid textured and texture-less objects
R18	Number of objects	Recognition of at least 25 textured and 25 texture-less objects
R19	Recognition range	70 cm to 170 cm
R20	Recognition time	Processing of a scene image in less than 5 s
R21	Recognition accuracy	Average angular deviation below 6° and Euclidean deviation below 9 mm at a distance of 0.7 m to the object

Table 1.4.: Requirements for object recognition.

1.5. Outline

The organization of the document follows the subdividing of the perceptual process into data acquisition, object training and object localization. Section 2 outlines the state of the art in the relevant research areas, from which the developments within this thesis are derived at the end of each chapter. An overview of the system design is given by Section 3. It outlines the variations in the hardware design of the sensor setup and gives an overview of the software components and their interconnections. Section 4, Section 5, and Section 6 describe the contributions related to data acquisition, object training and object localization in detail. Each section concludes with an evaluation of the proposed algorithms or methods. A summary of the presented results and an outlook on further work is given by Section 7.

2. State of the art

The outline of the state of the art section follows the structuring of perception from Section 1.2, by subdividing it in the three core parts of data acquisition, object modeling and object recognition and localization. The state of the art in data acquisition is given by Section 2.1, in which current camera sensor technologies and methods relating to their combination are presented. Section 2.2 describes established object modeling techniques and gives a summary of current feature descriptors. Finally, Section 2.3 presents existing work in the field of object recognition and localization.

2.1. Data acquisition

Camera sensors create a digital representation of the physical scene they are perceiving. On the basis of their data, mathematical models are developed to enable the autonomous recognition and interpretation of the environment.

Charge coupled device (CCD) and complementary metal oxide semiconductor (CMOS) are the two main sensor technologies to convert photons into electric charge and from there into digital signals. The main difference lies in the transfer of the per-pixel charge from the individual pixel to the sensor output in order to read their values and create the corresponding voltage. For CCD sensors, there is usually a limited number of output nodes e.g. for each image row. The charge of all pixel is shifted to the output nodes where a digital converter reads out the voltage and converts it to a digital value. CMOS sensors have an individual output for each pixel, resulting in a much faster sensor output. This increases the complexity, but also leads to an increase in frame rate. A common problem of CMOS sensors is the fixed-noise-pattern, which is decreasing the uniformity of the image. The effect is originating from the huge number of individual charge-to-voltage converters having their own characteristics. By elaborating the fixed pattern which is caused by this effect, it may be corrected using appropriate filters. Traditionally, CMOS sensors allow for a higher level of integration and lower power consumption. However, the higher integration level also causes higher chip prices and, due to a smaller photon-receptive area, a lower sensitivity. Nowadays, there is no significant difference in performance between the two sensor types. Current developments have made almost invalid the mentioned shortcomings

of both sensor types and a clear vote for any of the two sensor types is no longer possible [Car02].

In terms of the data dimensionality, a distinction is made between 2D and 2.5D sensors. 2D cameras denote common color cameras, that project the 3D data from the physical scene onto their 2D sensor chip to create a color image. 2.5D sensors refer to cameras that additionally infer a range value for each pixel relating to the distance of the perceived physical scene to the camera internal coordinate system. The data is not considered 3D, because of the observer centric perspective of the camera sensor i.e. the full 3D structure of the physical scene may only be observed when moving the camera to capture further perspectives like side views and integrate them into a common digital 3D representation. This property is also visualized by Figure 2.1.

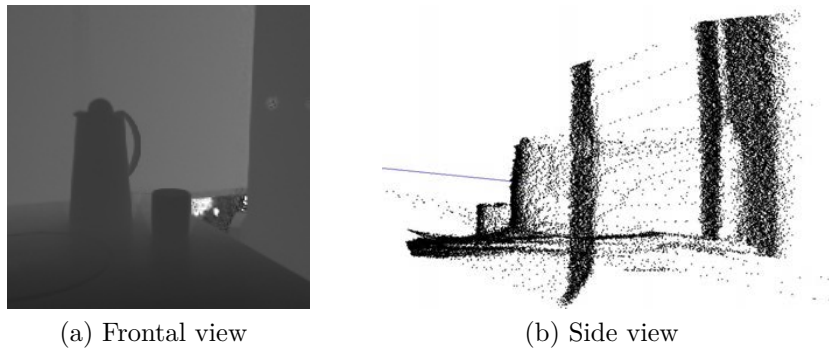


Figure 2.1.: 2.5D range image from a single recording: The frontal view on the physical scene in Figure 2.1(a). The missing 3D data, that could not be captured by the camera, is clearly visible when changing the viewpoint as shown in Figure 2.1(b).

The family of 2.5D sensor technologies is summarized under the common term *range imaging*. The most prominent among them are stereo vision, time-of-flight or structured light techniques.

2.1.1. Stereo vision

A stereo vision system is generally equipped with two color cameras that are horizontally displaced relative to each other. This displacement enables the perception of points from two different perspectives and provides the basis for depth computations using triangulation.

Triangulation relates to the fact that a point in 3D space is not measured directly, but rather indirectly by its projections onto 2D image planes of the recording camera sensors. Depth is inferred based on point correspondences across the image pairs and the knowledge of the cameras' relative position to each other (extrinsic parameters) and their projection rules (intrinsic parameters). On textured scenes, stereo vision is able to provide high resolution point clouds. However, in the absence of features, the system cannot establish

point correspondences across the image pairs and fails to measure depth. Stereo vision is also prone to occlusions, due to the different viewing angles of the two cameras and low frequency distortions, like repetitive patterns. This often disturbs the feature association, leading to false depth measurements. An illustration of a stereo setup on the mobile service robot Care-O-bot[®] 3 and its measurement principle is given by Figure 2.2.

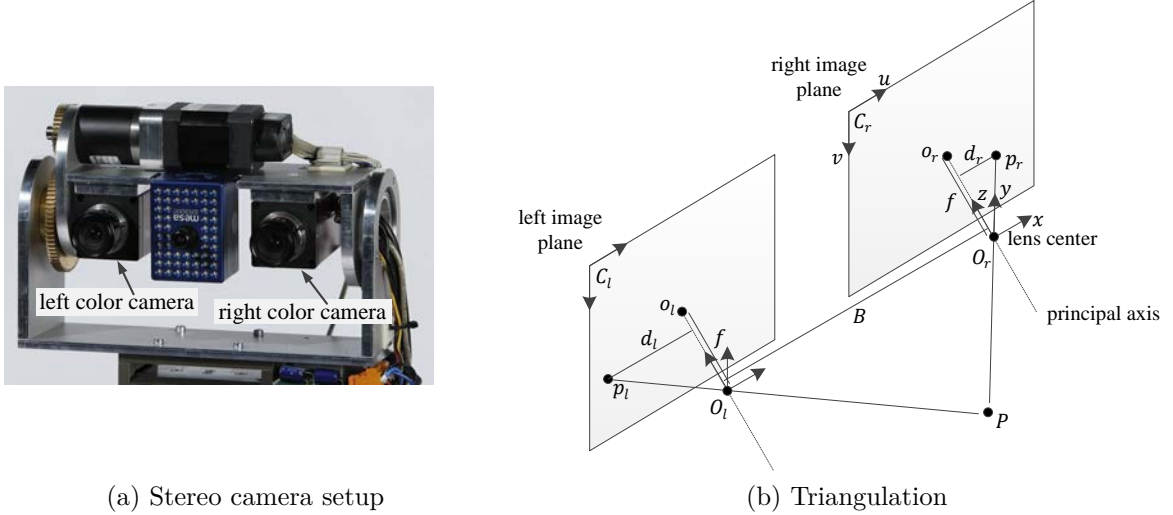


Figure 2.2.: Figure 2.2(a) shows a stereo camera setup from the mobile service robot Care-O-bot[®] 3 using two color cameras mounted on the outer sides of the sensor head. Figure 2.2(b) illustrates the principle of triangulation using a standard pinhole camera model.

Triangulation infers depth from the projection of a point P onto the left and right 2D image plane of the color cameras. It is assumed, that the image planes are rectified, meaning that they are coplanar and the perspective projections $\mathbf{p}_r = (u_r, v_r, f)$ and $\mathbf{p}_l = (u_l, v_l, f)$ of $\mathbf{P} \in \mathbb{R}^3$ are row-aligned. Let B be the length of the baseline connecting the two lens centers and f be the focal length, which denote the distance from the image plane to the lens center. The displacement of \mathbf{p}_r and \mathbf{p}_l is denoted disparity d and is given by $d = u_l - u_r = \frac{Bf}{z}$ from which the distance z to \mathbf{P} may be inferred according to (2.1) [FP03].

$$z = \frac{Bf}{d} \quad (2.1)$$

The derivation of (2.1) is as follows: consider Figure 2.2(b). Let $\mathbf{P} = (x_r, y_r, z_r)$ denote a scene point and $\mathbf{p}_r = (u_r, v_r, f)$ denote its perspective projection on the right image plane, both expressed relative to the coordinate system centered at O_r . Then, $\overline{O_r \mathbf{p}_r} = \lambda \overline{O_r \mathbf{P}}$ and therefore $u_r = \lambda x_r$, $v_r = \lambda y_r$ and $f = \lambda z_r$. Using the equation for f in u_r and v_r yields $u_r = f \frac{x_r}{z_r}$ and $v_r = f \frac{y_r}{z_r}$. The same equations hold for $\mathbf{P} = (x_l, y_l, z_l)$ and $\mathbf{p}_l = (u_l, v_l, f)$ expressed relative to the coordinate system centered at O_l , leading to $u_l = f \frac{x_l}{z_l}$. However, the scene point \mathbf{P} expressed relative to the coordinate system at O_l has the coordinates

$\mathbf{P} = (x_l, y_l, z_l)$ with $z_l = z_r$ and $x_l = x_r - B$. Therefore, $d = u_r - u_l = f_{z_r}^{x_r} - f_{z_l}^{x_l} = f_{z_r}^{x_r} - f_{z_r}^{x_r - B} = \frac{Bf}{z_r}$ \square

An excellent comparison of state of the art stereo algorithms is given on the Middlebury Stereo Page [SS12], which is based on the taxonomy approach by Scharstein and Szeliski [SSZ02]. It evaluates over 120 different stereo vision algorithms based on their 3D reconstruction performance on a predefined dataset of stereo images with associated ground truth data. Within the taxonomy of Scharstein and Szeliski, four basic steps have been identified that are common to most stereo algorithms: Matching cost computation, cost aggregation, optimization and disparity computation, and disparity refinement.

Each stereo vision algorithm starts with the *matching cost computation*, which evaluates a discrete cost function for each pixel based on individual disparity guesses. Prominent approaches for matching cost functions include absolute or squared intensity dissimilarity measures [BT98], robust M-estimators, or approaches originating from the information theory like Mutual Information [Hir08].

Cost aggregation considers the pixel-wise matching cost of local neighborhoods and aggregates it into a single cost value. Common procedures for cost aggregation are based on accumulating costs of fixed sized windows [BG05, KSK06], others use an additional weighting function for cost accumulation. More advanced procedures perform an online adaption of the window size to compensate for occlusions and discontinuities [CSY06].

Concerning *optimization and disparity computation*, a distinction is made between local and global stereo algorithms. Local algorithms perform disparity selection using a Winner-Takes-All strategy, meaning that the disparity with minimal costs is selected for each pixel without explicitly considering global smoothness constraints. Local smoothness constraints are applied by aggregating matching costs over a pixel's local neighborhood in order to minimize an energy function. The energy function consists usually of a data term and a smoothness term. The data term captures the calculated matching costs. The smoothness term explicitly enforces piecewise smoothness assumptions and penalizes disparity variations for close-by pixel. Optimizing the energy function over the complete 2D image space is a NP-hard problem. Therefore, local methods minimize the energy function following individual 1D directions e.g. along image rows. The 1D optimization can be performed in polynomial time using dynamic programming approaches. However, it possesses the disadvantage of generating streaking effects towards the direction of optimization on the resulting disparity image.

Global algorithms perform disparity selection using global reasoning to minimize the energy function over the complete 2D image space. Common approaches to cope with the computational complexity are Graph Cuts [KZ01], Belief Propagation [KSK06, BRCV12] or Simulated Annealing [MMP87]. State of the art stereo matching algorithms achieve the most accurate and dense depth maps only, when using global optimization algorithms,

needing up to a minute or longer of computation time. Local correlation based methods are fast enough for real time applications at the cost of less accuracy and sparse depth maps [FP03]. A summary and detailed description of state of the art optimization strategies is given in [SZS⁺08, Cas10]

Hirschmüller [Hir08] proposed a semi-global approach for minimizing the energy function. Instead of performing global optimization over the complete 2D image space, it is possible to minimize the energy function following several individual 1D directions e.g. along image rows, columns and along different angles. Hirschmüller aggregates matching costs from 16 surrounding 1D directions around each pixel into a common cost function. The directions relative to pixel $\mathbf{p} = (u, v)$ are pointing towards its eight directly surrounding pixel $\{(u - 1, v), (u - 1, v - 1), (u, v - 1), (u + 1, v - 1), (u + 1, v), (u + 1, v + 1), (u, v + 1), (u - 1, v + 1)\}$ and the eight pixel given by $\{(u - 2, v - 1), (u - 1, v - 2), (u + 1, v - 2), (u + 2, v - 1), (u + 2, v + 1), (u + 1, v + 2), (u - 1, v + 2), (u - 2, v + 1)\}$. This avoids the generation of artifacts compared to optimizing along a single direction and makes a compromise between the advantages of global and local methods.

Disparity refinement deals with the removal of outliers originating from wrong disparities and the reduction of quantification errors, originating from the usage of discrete disparity values. Common methods for quantification error reduction apply sub-pixel interpolation over matching cost measures from neighboring disparity values [YYDN07, MS12]. For outlier removal, usually median filters, morphological operators, or bilateral filters [TM98, Yan12] are used. Other methods apply bidirectional matching, that calculates disparity estimates using each of the two images as a reference image for disparity estimation. Outliers are detected by checking the obtained disparity guesses from both images for consistency. More advanced methods apply RANSAC for robust plane fitting to remove outliers on segmented image areas. The underlying assumption is that disparities of each segment vary only smoothly [WZ08, SWP⁺12].

Stereo vision has the advantage that it measures range very precisely in a certain range interval, which is defined by the horizontal displacement of the two camera sensors. Often, the average accuracy of range measurements is below 1 mm. However, the algorithmic efforts in order to compute the range values are rather demanding and often achieve a reasonable computation time only, when using hardware acceleration. A fast computation of accurate range values is only reachable for textured scenes, which enable the association of corresponding pixel between the two camera images using fast-to-compute algorithms. Uniform image areas that cannot be associated require computationally demanding optimization strategies, leading to inappropriate run times.

2.1.2. Time of flight

Time of flight (ToF) cameras are one-shot range sensors that use active illumination in the near infrared range to infer distance information. Figure 2.3 shows a ToF sensor manufactured by MESA Imaging and explains its measurement principle.

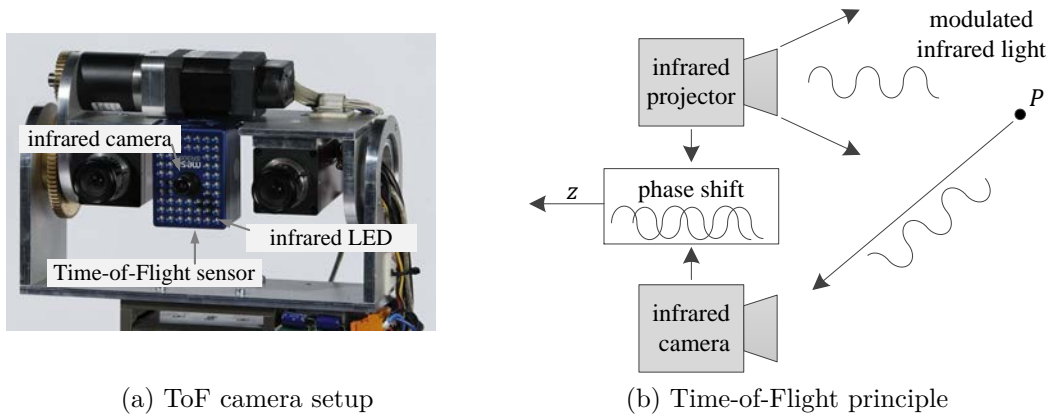


Figure 2.3.: Figure 2.3(a) shows a ToF camera setup using a SwissRangerTM SR3000 camera mounted on the sensor head of a robot. Figure 2.3(b) illustrates the measurement principle. ToF cameras emit modulated near infrared light to illuminate a scene. The reflection of the modulated light from the surface of the scene point P is collected by an infrared camera in a CMOS matrix. In order to compute the distance to the reflecting surface, the returning signal is compared to the camera's source modulation to compute the phase shift. The distance to the reflecting surface is given by its linear dependency to the phase shift.

A Time-of-Flight sensor is able to operate at about 15 to 30Hz. It creates dense point clouds, however, with a limited spatial resolution. As the measurement principle assumes perfectly sinusoidal signals, which are not achievable in reality, the measured distance is subject to noise, which amounts to roughly 1% of the measured distance. Also the measurement principle is biased as a function of object albedo, resulting in poor performance on textured scenes. A prominent example is the distance measurement originating from the reflections of a checkerboard. Here, the black squares seem to be closer to the camera than the white squares.

In addition to the measured range values, ToF cameras provide a low-quality intensity image showing a gray image of the physical scene as well as an amplitude image representing the signal strength of the returning signal. The measurement range of ToF cameras is variably adjustable from 0.1 to 5 m. Due to the measurement principle, which is based on comparing the phase shift between emitted and received signal, ToF sensors have a non-ambiguity range. The phase shift cannot take arbitrarily large values as its range is limited by the wave length of the modulated infrared signal emitted by the camera. Figure 2.4 gives

an overview of existing ToF sensors manufactured by MESA Imaging and PMD[vision][®].



Figure 2.4.: Range imaging sensors based on the ToF principle manufactured by MESA Imaging and PMD[vision][®].

2.1.3. Low-cost structured light

Range imaging sensors, which are based on the principle of structured light, project a known pattern e.g. dots or stripes onto a scene. The reflection of the pattern is captured by a camera sensor and range is inferred based on its visible deformation. Figure 2.5(a) shows a low-cost structured light camera setup using the Kinect[™] camera. It consists of a color camera, an infrared projector and an infrared camera. Figure 2.5(b) illustrates the measurement principle. The inference of depth follows the triangulation principle from stereo as explained by Figure 2.2(b) with the exception that one camera is replaced by an infrared projector, which emits a pseudo-random dot-pattern. Instead of observing the scene image, the dot-pattern is reflected by the scene and observed by an infrared camera. Local parts of the observed patterns are extracted and matched with the emitted pattern. Based on the measured disparity between the observed and the emitted pattern, depth is inferred using triangulation. The color camera is not used for depth computation, but augments the range values with color information.

Repeatability tests for the Kinect[™] camera show [MTD12], that the accuracy of the range measurements amounts to 0.5% with respect to the measured distance. The camera operates at about 30Hz and has a measurement range from 1.2 to 3.5 m. The measurement range may be adjusted by varying the structure of the projected pattern. Compared to a ToF sensor, the main advantage of the Kinect[™] is its low cost. However, the Kinect[™] works only for indoor environments whereas ToF sensors are able to operate in outdoor environments as well. Also, object boundaries are reflected more accurately with a ToF sensor and shadow effects, due to the multi camera setup of the Kinect[™], do not occur.

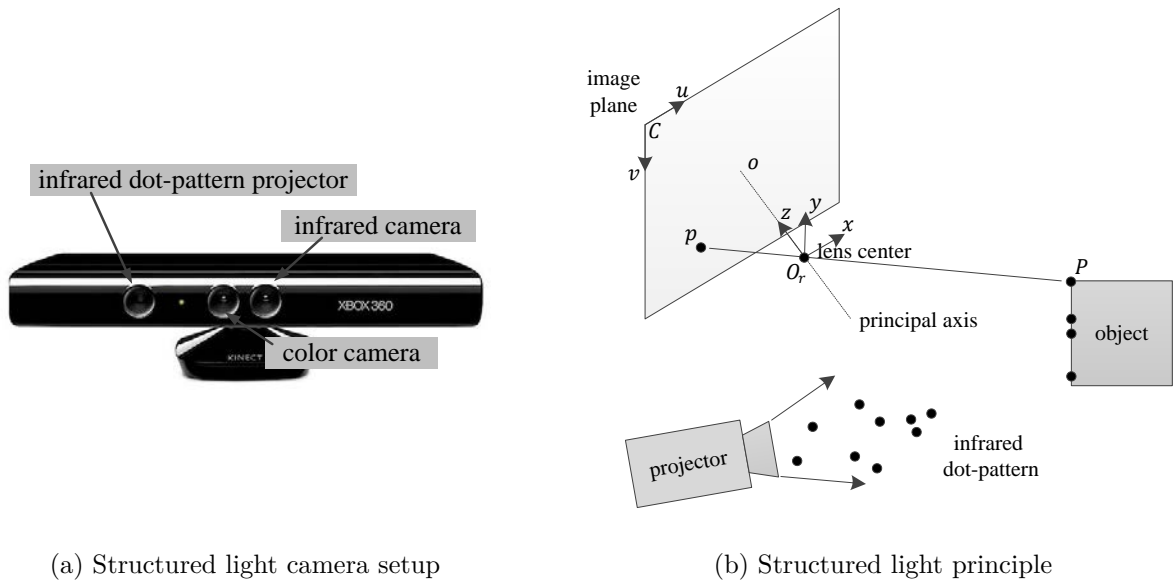


Figure 2.5.: Structured light camera setup using the KinectTM camera (Figure 2.5(a)) and its measurement principle (Figure 2.5(b)).

2.1.4. Sensor fusion

Sensor fusion aims at combining the data from different sensors to create information that exceeds the quality of each individual sensor alone. In terms of quality, one usually relates to accuracy, completeness or confidence. Within the scope of this thesis, specifically the combination of a stereo camera setup with a range sensor either based on the ToF or structured light principle is considered.

Several approaches have been proposed to combine the advantages of stereo vision and ToF range sensors. In accordance to the proposed taxonomy of Scharstein and Szeliski [SSZ02], they are roughly categorized in global and local methods. Global methods make explicit smoothness assumptions and solve an optimization problem to minimize a global energy function. Local methods are based on pixelwise matching costs within a certain neighborhood. By selecting the disparity with minimal cost, effectively a Winner-Takes-All optimization is performed.

Among others, global methods for sensor combination have been reported by Zhu et al. [ZWYD08, ZWGY10, ZWY⁺11] and Hahne et al. [HA08]. Zhu et al. calculate in [ZWYD08] depth probability distributions for both sensor modalities, which are combined using Markov Random Fields (MAP-MRF). The ToF sensor is calibrated using a 4D look-up table to map the measured intensity and 3D coordinates to the ground truth distance given by the stereo camera pair. To calculate the most likely disparity given the sensor measurements, the data term of a global optimization algorithm for stereo vision is multiplied by a term describing the Euclidean distance between the corresponding 3D coordinates from the stereo and the ToF sensor. In [ZWGY10], the fusion technology is extended to the temporal

domain by using a dynamic Markov Random Field to infer depth from both spatial and temporal neighbors. Following their preceding work, the authors calculate the most likely disparity given the sensor measurements by augmenting the data term of a global optimization algorithm for stereo vision with a term describing the Euclidean distance between the corresponding 3D coordinates from stereo and ToF. To achieve temporal dependencies between different frames taken at different time stamps, they propose a layered structure, where each layer itself represents a MRF and connections between the layers describe temporal dependencies. An additional smoothness term of the global optimization function is introduced to model temporal smoothness. In [ZWY⁺11], Zhu et al. replace the ToF data term with a simpler linear function and incorporate the reliability of the ToF measurements in dependence of object reflectance and vignetting into the energy function. Hahne et al. use a graph cut approach in [HA08] to initialize the domain of a volumetric grid with the depth information from the low resolution ToF camera to cut down computation time and to increase accuracy of the depth estimation. Initially, the two color cameras and the ToF sensor are calibrated to each other to get their extrinsic and intrinsic parameters. To compute the desired depth map, the voxels of a $400 \times 300 \times 100$ grid are associated with their corresponding depth values from the ToF sensor. Those voxels, which are not present in the ToF image, are not considered during the graph cut procedure. During graph cut, the data term is extended by a term describing the difference between the assigned distance of the voxel and the measured distance from the ToF sensor. Additionally, the smoothing term is extended to incorporate discontinuities in the ToF depth image. The surface, which is minimizing the energy function, is calculated using a standard graph cut algorithm. The required computation time amounts to a few minutes.

Local sensor combination approaches have been proposed by Gudmundsson et al. [GAL08], Hahne et al. [KKHA09], Bartczak and Koch [BBP⁺09], Yang et al. [YTCA10], and Nai et al. [NLM⁺12]. Gudmundsson et al. [GAL08] perform sensor fusion by calculating disparity estimates for stereo vision from the 3D ToF sensor. This constrains the stereo algorithm on a per pixel basis, resulting in more accurate disparity maps. All cameras are stereo calibrated to each other by down-scaling the color images to the resolution of the ToF sensor. The two color sensors are stereo calibrated to each other using their original image size. For each pixel of the ToF sensor, the corresponding left and right pixel of the color images are determined using the calculated homographies from stereo calibration. After up-scaling the color image to its original size, it yields an initial disparity estimate for the stereo camera pair used as a per pixel constraint for stereo matching. Hahne et al. combine in [KKHA09] a ToF sensor and a stereo rig by using data from the ToF sensor to limit the disparity search range for stereo on demand. The proposed method is real-time capable and applies adaptive disparity search ranges on the stereo images based on the measured distance from the ToF sensor. This increases the reliability of range values from stereo at

depth discontinuities. The dominating cause for wrong depth values are surfaces with bad reflection properties, like dark areas. This information is available through the amplitude information of the sensor. Therefore, Hahne et al. apply a 3×3 median filter to preprocess the amplitude image and threshold the amplitude values to reject unreliable depth values. Rejected values are interpolated with reliable, neighboring range values and further improved by applying the information from the stereo rig. To generate a dense depth estimate from stereo, the preprocessed ToF values are used to generate a piecewise bilinear mesh. The mesh is transformed relative to the stereo rig's coordinate system and the intersection of a viewing ray from the stereo rig with the surface defines an initial disparity guess and a range according to the associated confidence value. Regions with a confidence value below a specified threshold are further processed using a standard correlation based stereo algorithm on the stereo images. For pixel with a valid confidence value, the disparity guess from the ToF sensor is directly used to calculate the corresponding range for the stereo image. Bartczak and Koch introduce in [BBP⁺09] a cost function for each pixel of a high resolution color image, where the minimum of the function's per pixel value corresponds to the locally optimal depth. At first, the low resolution ToF image is warped to fit the size of the high resolution stereo depth map by creating a triangle mesh from the 3D ToF data and re-projecting the mesh on the stereo image. Wrong 3D information originating from the different viewpoints and depth discontinuities are detected by comparing a triangle's normal with the stereo rig's line of view. Triangles with normals close to 90 degree are removed. Then, the authors define a cost function for each pixel over its local neighborhood that incorporates the squared distance between the depth measured by the ToF sensor and the proposed depth from stereo as well as the color consistency of the left and right stereo image patch given the proposed distance. The depth value with the smallest cost is selected for each pixel. Yang et al. present in [YTCA10] a GPU based hierarchical up-sampling method for the low resolution ToF image. By creating confidence estimates for the stereo and ToF data, they design a cost function to populate the 3D volume created by a confidence guided plane-sweeping stereo matching algorithm. Nai et al. propose two GPU-based sensor fusion approaches for ToF and stereo data in [NLM⁺12] using fidelity measures, derived from ToF intensity values and depth gradients and combined with horizontal image gradients and stereo consistency measures from the stereo camera system.

2.2. Object modeling

Object models store relevant 2D texture and 3D shape information. This information is necessary for the computation of distinct 2D and 3D cues, which are represented by feature descriptors. These descriptors are matched against unseen feature descriptors from unknown scenes, which enables the recognition of the object. In order to reduce the com-

putation time, feature descriptors are usually not computed densely for each point, but sparsely using feature detectors to determine salient locations on the object.

2.2.1. Feature descriptors

The development of robust feature descriptors lies at the core of every object recognition system. Viola and Jones presented in [VJ01] a method for rapid object detection in 2D images using Haar features. Haar features are based on the simple principle to subtract the sums of intensity values from adjacent rectangular regions. In order to achieve a rapid computation of the Haar feature, Viola and Jones proposed a novel image representation called *integral image* that enables the computation of the sum of intensity values from a rectangular region with only four additive operations. An integral image is computed based on the original image by assigning each pixel the sum of intensity values over all pixel to its left and to its top in Figure 2.6 e.g. p_1 holds the sum of all intensity values from rectangle A , p_2 from $A+B$, p_3 from $A+C$, and p_4 from $A+B+C+D$. Therefore, the sum of intensity

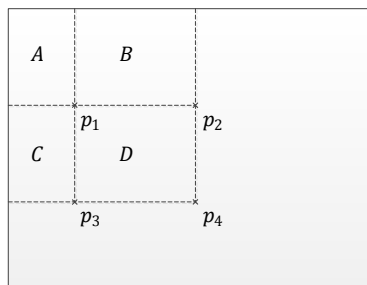


Figure 2.6.: The principle of an integral image.

values for the rectangle enclosed by the pixel p_1 , p_2 , p_3 , and p_4 is given by $p_4 - p_2 - p_3 + p_1$. The computation of the integral image itself is possible in linear time by reusing the already computed values from preceding pixel.

Probably, the most prominent and widely used 2D point feature detector and descriptor is SIFT, which has been presented by Lowe et al. [Low03]. The authors describe a robust point feature descriptor that uses a normalized histogram over gradient directions to describe the local neighborhood of the feature point. The descriptor is invariant against small variations in scale and rotation. The main limitation of SIFT poses its computational complexity, which makes its application without hardware acceleration often infeasible. Mikolajczyk and Schmid [MS05] were the first to introduce a log polar grid instead of SIFT's regular grid for gradient-histogram sampling. The resulting descriptor GLOH (Gradient location-orientation histogram) outperforms SIFT in terms of distinctiveness and robustness. Bay et al. present with SURF [BTG08] an accelerated version of SIFT by using several approximations for feature point detection and description. For interest point detection, the Hessian matrix is computed at each image point p across several scales. The

Hessian captures the local curvature at p , which enables the detection of local feature points at corners or dot-like structures by computing and examining its determinant. A box filter in combination with integral images [VJ01] approximates and accelerates the computation of the determinant of the Hessian matrix. For descriptor computation, the local neighborhood of a feature point is divided into 4×4 equally sized subregions. The feature descriptor captures the sum of the local gradient distribution in the x and y direction for each of the 16 subregions. Tola et al. [TLF08] use a log polar grid with Gaussian weights from [WB07]. They accelerate the histogram binning by applying Gaussian convolutions. The resulting descriptor is termed *DAISY*, due to the flower like arrangement of the local image regions for histogram creation. They compare *DAISY* with other popular descriptors like *SIFT* and *SURF*. *DAISY* and *SIFT* perform equally well and outperform the others. However, *DAISY* is much faster to compute than *SIFT*, which even allows for its descriptors to be calculated densely for every image pixel. The speed-up of *DAISY* in relation to *SIFT* is due to the replacement of the weighted sums from *SIFT* by fast-to-compute Gaussian convolutions. However, even with the improvements in computation time of *DAISY* and *SURF*, these feature descriptors are still not suitable to be used in applications, that demand fast reaction times without the usage of hardware acceleration.

Recent developments in feature point descriptors extensively elaborate possibilities to simplify the descriptor calculation. Popular methods reduce the descriptor computations to pure intensity comparisons or represent the descriptor with binary strings, which enables the application of faster matching algorithms [Sha06, OCLF10, TRD09]. One prominent example is *BRIEF* from Calonder et al. [CLSF10], which randomly compares intensity values of image point pairs within the local neighborhood of a feature point. However, the significant reduction of the computational complexity of the proposed feature descriptors is traded for its poor invariance against visual transformations like scaling or rotation. *BRIEF* has been extended by Rublee et al. [RRKB11] to create *ORB*, a rotational invariant version of *BRIEF*. *ORB* uses the *FAST* interest point detector [RPD10] to locate stable feature points on the image. It simply computes the difference between the intensity value at the current pixel p with the intensity value of its surrounding pixel. If more than k surrounding pixel are considerably brighter or darker than the central pixel p , it is labeled as a corner. Rublee et al. extend the *FAST* corner detector to varying scales using image pyramids and employ an orientation measure using image moments to reorder the descriptor values, accordingly.

Similar to 2D point features, a variety of 3D point features describing the local shape of an object around an interest point have been proposed. Spin images [JH99] describe the surface shape of an object. The algorithm computes oriented 3D surface points by capturing their object-centered 3D position and by assigning them a 3D normal vector based on their local surface shape. With respect to each oriented surface point, the cylindrical 2D coordinates of

neighboring 3D points are captured within a 2D accumulator, whereas the two dimensions denote radius and elevation. The contribution of each surface point is distributed across neighboring bins of the accumulator using bilinear interpolation in order to increase the robustness against noisy 3D measurements. Finally, the spin image descriptor is given by the entries of the 2D accumulator. Spin images have been extended in [FHK⁺04] by introducing 3D shape contexts. The authors make use of a sphere as a local support region around each oriented 3D point with its north pole oriented towards the point’s normal direction. Instead of using a 2D accumulator, the authors introduce a third dimension to capture the azimuth with respect to the oriented 3D surface point. Additionally, the radial dimension of the accumulator is partitioned logarithmically. Experiments show, that the introduction of the third dimension outperforms the 2D spin image approach in terms of recognition rate. Further improvements, that take on the idea of 3D shape contexts, have been proposed by Zhong in [Zho09]. The author introduce intrinsic shape signatures, which replace the normal computation for the local reference frame estimation by an eigenanalysis and uses the eigenvectors to define the local coordinate system at an oriented 3D point. The 3D accumulator is defined by a sphere-shaped support region around each oriented 3D point and a discrete spherical grid of increasing resolution. Experiments show, that the recognition rate of intrinsic shape signatures outperforms the recognition rate achievable with 3D shape contexts. Tombari et al. [TSDS10] have conducted thorough experiments on the stability of the reference frames given by the oriented 3D points. The authors propose a further improvement of the shape descriptor by increasing the stability of the reference frames using a variant of the eigenvector computation for estimating the normal vectors. This is achieved by assigning a smaller weight to distant neighboring points, when computing the covariance matrix for eigenvector computation. Furthermore, instead of capturing the 3D point coordinate distribution within the local support region, the authors propose to capture the distribution of the normal directions and term their descriptor *Signature of Histograms of Orientations* (SHOT).

The presented feature descriptors are local point features, as the scope of the descriptors is limited to a local neighborhood around the feature point. They are predominately used when facing objects with rich texture. In the presence of texture-less objects, global feature descriptors are applied to extend the scope of the feature descriptor to the complete object. Hinterstoisser et al. [HCI⁺12, HCI⁺11, HLI⁺10] explicitly address the recognition of untextured 3D objects without salient feature points by applying a dense descriptor calculation for each image pixel and performing a brute force feature matching approach using a sliding window. The feature descriptor is based on 2D image gradients and 3D normal vectors and is invariant against small transformations. Muja et al. present with BiGGPy in [MRBL11] a variant of the descriptor proposed by Hinterstoisser. They introduce pyramid levels that aggregate the information of neighboring pixel ranging from small 2×2 pixel-neighborhoods

at the lowest level up to the area of the complete object at the highest level. Feature matching is performed beginning with the highest level and continued to the lower levels until a specific matching score is reached. This reduces the number of time-consuming descriptor comparisons as the pyramid's higher levels aggregate the information of many individual features at lower levels and a mismatch prevents the need to continue comparing with all successive pyramid levels.

A common approach in order to derive a global descriptor from a set of local descriptors is termed *Bag-of-Words* (BoW), which is traditionally used for object classification [TCF09, KPVG10, BFG⁺11]. BoW extracts local features from a given image region and clusters them according to their closest distance to a predefined set of *codewords*. The codewords denote a set of defined features that have been determined from an arbitrary image by clustering its image features and extracting the cluster centers. The number of image features, which have been assigned to a codeword, is counted and represented by a histogram. The histogram is then used as a descriptor to represent the given image region.

2.2.2. Data association

Data association tracks the appearance of a feature point across different object images based on its descriptor value. Due to the change in viewpoint and the inherent noise within the sensor data, feature point matching follows the *nearest neighbor rule*. Instead of seeking an exact matching, nearest neighbor matching associates feature points with a descriptor distance not larger than a predefined maximal threshold d_{max} .

Distances within the k -dimensional feature space are usually defined based on the *Euclidean distance* d as given by (2.2) for the two descriptor values $\mathbf{v} = (v_0, v_1, \dots, v_k)$ and $\mathbf{v}' = (v'_0, v'_1, \dots, v'_k)$.

$$d(\mathbf{v}, \mathbf{v}') = \sqrt{\sum_{m=1}^k (v_m - v'_m)^2} \quad (2.2)$$

Without any computational optimizations, the performance for searching n feature point correspondences among m possible candidates would not scale better than $O(nm)$. A naive implementation would simply compare the descriptor values of each candidate with each feature point following the nearest neighbor rule. A *k-d tree* (k-dimensional tree) [Ben75] organizes the known feature descriptors by partitioning their k -dimensional feature space into a binary tree. By elaborating the structure of the binary tree, the effort for nearest neighbor matching is reduced to $O(n \log(m))$, when assuming randomly distributed data. The space partitioning of a k-d tree is based on the construction of axis-aligned splitting planes. Starting with the first dimension i.e. the first descriptor entry of all feature

descriptors, the median of all values is computed and the splitting plane is set up along it. The descriptor corresponding to the median constitutes the new node element and all other descriptors are assigned to its left or right child node based on the partition given by the splitting plane i.e. descriptor values with smaller values than the median are assigned to the left child node and all others to the right child node. The procedure continues recursively with the children of the node element using the following descriptor dimensions to set-up the splitting-planes until all elements have been partitioned. When all dimensions of the descriptor have been traversed during the partition, the separation continues again with the first dimension and proceeds with the following until all descriptors are separated from each other. The construction of a 2-d tree is visualized by Figure 2.7.

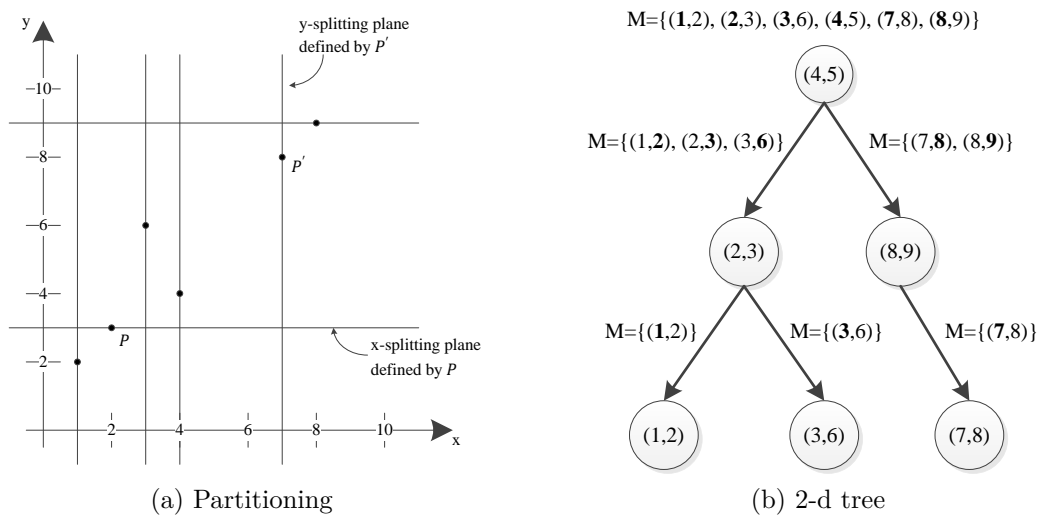


Figure 2.7.: Partition of a 2-dimensional feature space using the feature descriptors defined by $M = \{(1, 2), (2, 3), (3, 6), (4, 5), (7, 8), (8, 9)\}$.

Searching for the nearest neighbor to a query descriptor within the k-d tree reduces to a comparison with the node elements to decide for the direction of traversal until a leaf node is reached. From there on, the distance of the leaf node to the query descriptor serves as a reference distance and the tree is traversed backwards up to the root node. For each node that is passed on the way to the root, it is checked if its distance to the query descriptor is smaller than the reference distance. If it is smaller, the distance to the current node becomes the new reference distance. Additionally, it is checked if the distance to its leaf nodes on the non-traversed side could result in a smaller distance than the reference distance. This is done by intersecting the splitting plane with a sphere around the reference point, whose radius corresponds to the reference distance. If an intersection is detected, the outlined procedure continues from its beginning with the leaf nodes of the current reference node. Otherwise, the traversal to the root node continues. The intersection of the splitting plane with the described sphere can be computed efficiently by comparing the descriptor

values at the position corresponding to the splitting dimension of the node representing the splitting plane e.g. v_m and the node representing the query point e.g. v'_m in relation to the current reference distance d_{ref} e.g. by $|v_m - v'_m| < d_{ref}$.

Optimizations of the original k-d tree algorithm include the introduction of multiple randomized k-d trees [SAH08]. Instead of creating a single search tree, multiple trees with different split dimensions are constructed and searched concurrently. Additionally, the cyclic order, in which the partitioning dimensions are traversed, is replaced with a random selection of the split dimension among the n dimensions with largest variance.

For many applications, approximate nearest neighbor search methods are used, that guarantee the retrieval of the closest matching candidate only with a predefined probability [ML09]. One common strategy is to limit the amount of examined leaf nodes to a fixed number at which the closest matching candidate is returned.

In case of binary data, local sensitivity hashing (LSH) [PJA10] is an adequate method to represent the high-dimensional binary descriptors within a fast to search data structure. LSH creates a random concatenation of hash functions to map similar data to the same hash value. Similar to multiple randomized k-d trees, a sequence of $k \in \mathbb{N}$ hash functions $h_i: \{0, 1\}^d \rightarrow \mathbb{N}, i \in \{1, \dots, k\}$ is applied to the input data, whose output is concatenated to a single hash function $h^j(\mathbf{v}) = [h_1(\mathbf{v}), \dots, h_k(\mathbf{v})], j \in \mathbb{N}$. The procedure may be repeated in order to generate $l \in \mathbb{N}$ different hash functions h^1, \dots, h^l . Then, a nearest neighbor query returns l corresponding hash buckets, from which all associated data points represent potential matching candidates.

Among all potential matching candidates, the closest matching is inferred by elaborating the binary descriptor structure using the *Hamming distance* measure, given by (2.3).

$$d(\mathbf{v}, \mathbf{v}') = \sum_{m=1}^k \text{xor}(v_m - v'_m) \quad (2.3)$$

The Hamming distance computes the amount of differing bit values, when pairwise comparing each bit position of the binary descriptors \mathbf{v} and \mathbf{v}' . Both, the Euclidean distance and the Hamming distance are metrics. However, the computation of the Hamming distance is significantly faster due to its implementation using a hardware accelerated xor operation instead of computing a sum of squares.

2.2.3. Object modeling procedures

The general approach for object modeling is to place an object on a platform, which is either static and the camera system is moved around the object or the platform itself is loose and moves the object in front of a static camera system.

Krainin et al. [KHRF11] use a robot arm to grasp and move the object in front of an RGB-D camera system. The robot arm is jointly tracked with the object by the camera system. A Kalman filter in combination with an ICP variant and RanSAC for feature matching is implemented to generate pose estimates for the object’s feature points and the manipulator configuration. The error function of the ICP variant is extended to consider error terms for sparse and dense feature matching quality, 3D surface and point matching quality as well as prior knowledge. Krainin et al. explicitly address the problem of occlusion induced by the robot’s gripper during the training procedure by introducing an information gain based variant of the next best view algorithm. A highly accurate object modeling technique is presented by Casper et al. [KXD12]. It consists of a 3D digitizer, a loose platform for positioning the object, a stereo camera pair, and three fluorescent lamps to illuminate the platform. The novelty of this system lies in the presented hardware setup which has been accurately calibrated to avoid the need for extensive algorithmic optimizations in order to improve the registration process of the single object views. The presented system setup is stationary and rather large. Therefore, it is not intended to be used outside a laboratory environment.

Wüthrich et al. present in [WPR⁺12] a probabilistic registration procedure, where the object is placed on a table and successively moved by the robot to capture new images. The single object views are combined using a Bayesian framework that incorporates visual and non-visual information to infer the object movement. The visual information considers observed surface patches as well as silhouette information. The non-visual information holds the trajectory of the manipulator when moving the object. Hinterstoisser et al. [HCI⁺12, HCI⁺11, HLI⁺10] place the object at a reference position on a checkerboard. By manually rotating the checkerboard, different object views are recorded and saved for further processing. The object’s pose is inferred by tracking the checkerboard pattern. The information from the individual object images is not combined into a single 3D model, but saved independently for each image.

2.2.4. Simultaneous localization and mapping (SLAM)

In order to enable the localization and handling of objects, it is necessary to map the individual object images into a geometrically consistent 3D model. This process is closely related to the field of 3D environment modeling, called SLAM (*Simultaneous Localization and Mapping*). In SLAM, a robot moves autonomously through an unknown environment while the SLAM algorithm aims at constructing a consistent spatial map out of its sensor measurements. In the case of 3D object modeling, the environment is reduced to the object itself and the robot is represented by the camera system that records different images of the object. The main challenge of SLAM is to cope with measurement errors originating from the inherent noise in the sensor measurements. Therefore, statistical methods are applied

to aggregate the sensor information. In general, it is distinguished between filtering and optimization based methods.

Filtering techniques do not directly store past information, but aggregate the information of each time step into a single probability distribution. A prominent example of a filtering technique is FastSLAM presented by Thrun and Montemerlo et al. [MTRW03]. They propose to use a particle filter to represent the probability distribution over the robot movement and to use an extended Kalman filter to represent the constructed 3D map with its uncertainty by Gaussian distributions.

Optimization based methods, on the other hand, retain the information of dedicated time stamps by establishing so called keyframes and by performing optimization over all keyframes to infer a 3D map. Compared to filtering techniques, optimization based methods are computationally more expensive as the optimization is computed from scratch for every update. A prominent example for optimization is Bundle Adjustment [LA04] which uses the Levenberg-Marquardt minimization algorithm [Ken44] to optimize the map and the robot position over the keyframes. Engels et al. apply in [ESN06] the Bundle Adjustment algorithm for real-time camera tracking. They show that it effectively increases the accuracy of the position estimate and prevents the accumulation of errors over time. Strasdat et al. compare the two SLAM variants in [SMD10a]. The authors conclude that a representation of the SLAM problem using filtering lacks scalability in the number of tracked feature points due to the joint probability distributions representing the state space. The optimization approach keeps track of past poses by keeping a small set of keyframes over time. Therefore, poses and features related to the keyframes are not marginalized out and their original values are kept. This results in more efficient computations depending on the number of features compared to the filtering approach. The authors further state that increasing the number of features results in better accuracy than increasing the number of frames. Therefore, they propose to use filtering whenever the computational power is limited and bundle adjustment elsewhere.

2.3. Object recognition

Object recognition determines the existence and location of objects within the current environment by matching the 2D and 3D feature points from an object model with the feature points from the scene. It provides the basis for object localization, which computes an object's orientation and position in 3D space based on a set of corresponding features.

2.3.1. 3D-3D pose estimation

Literature distinguishes between one-shot and global techniques for object recognition and object localization. One-shot methods use a pairwise registration algorithm to register exactly two distinct 3D point sets e.g. one originating from the single camera view and the other originating from the object model, whereas global techniques are used to register several 3D point sets simultaneously.

The problem of pairwise registration is termed 3D-3D pose estimation [FP03]. Its solution is determined by solving a least squares problem, which minimizes the weighted sum of distances between corresponding feature point pairs from the two 3D point sets. The problem is formally given by (2.4).

$$\min_{\mathbf{R}, \mathbf{T}} \sum_{i=1}^n w_i \|\mathbf{y}_i - (\mathbf{R}\mathbf{x}_i + \mathbf{T})\|^2 \quad (2.4)$$

It determines the optimal 3×3 rotation matrix \mathbf{R} and 3×1 translation vector \mathbf{T} specifying the object's pose. $(\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^3 \times \mathbb{R}^3$ denotes the spatial coordinates of a 3D feature point pair and w_i a weight vector. The weight vector w_i might be used to indicate the reliability or importance of a point pair $(\mathbf{x}_i, \mathbf{y}_i)$ or simply assume a value of $w_i = 1, \forall i \in \mathbb{N}$, when such information is not available. The minimization problem is simplified by computing the centroid of the point sets $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$ and shifting the points according to their centroids resulting in (2.5).

$$\begin{aligned} \min_{\mathbf{R}, \mathbf{T}} \sum_{i=1}^n w_i \|\bar{\mathbf{y}}_i - \mathbf{R}\bar{\mathbf{x}}_i\|^2, \\ \bar{\mathbf{x}}_i = \mathbf{x}_i - \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i, \quad \bar{\mathbf{y}}_i = \mathbf{y}_i - \frac{1}{n} \sum_{i=1}^n \mathbf{y}_i. \end{aligned} \quad (2.5)$$

The translation vector \mathbf{T} may then be inferred, once the optimization problem has been solved for \mathbf{R} . A closed form solution for solving (2.5) has been presented by Horn [Hor87], which uses a unit quaternion representation to show that the eigenvector with the largest positive eigenvalue of a symmetric 4×4 matrix corresponds to the optimal rotation matrix \mathbf{R} . Similar methods have been presented in [HHN88], where orthonormal matrices instead of unit quaternions are used to represent the rotations \mathbf{R} , and [HJL⁺89], where a singular value decomposition is used to solve for the optimal rotation matrix \mathbf{R} .

2.3.2. Robust estimators

A common problem of pose estimation is the reliable detection of corresponding point pairs. The problem arises from noisy sensor measurements or ambiguous image data, resulting

in altered descriptor values and mismatches. Therefore, robust methods are necessary to detect and ignore these mismatches.

A standard method for outlier removal has become *Random Sample Consensus* (RanSaC) [FB81]. It randomly selects a subset of $j \geq 3$, $j \in \mathbb{N}$ feature point pairs from all correspondences, based on which the pose estimation problem is solved. The quality of the resulting rigid transformation is evaluated by aligning the remaining feature point pairs according to the computed transformation. All feature point pairs that are less than a fixed threshold away from each other constitute inliers and their average mean square error is used to determine the quality of the transformation. This hypothesize-and-test procedure is repeated for a fixed number of times or until the mean square error is smaller than a predefined threshold.

Experiments have shown, that even when selecting a subset of j correct correspondences the resulting transformation may still not be optimal relative to the set of all inliers. Therefore, an extension to RanSaC called *local optimized RanSaC* (Lo-RanSaC) has been proposed by Chum et al. in [CMO⁺04]. It extends RanSaC by a final pose estimation step to improve the determined transformation based on the set of all inliers.

Chum et al. further improved RanSaC in [CM05] and named their method *Progressive Sample Consensus* (ProSaC). The authors replaced the randomly drawn samples for hypothesis generation with samples that are drawn using a semi-random approach. Initially, all correspondences are ordered according to their descriptor matching distance. By exploiting the ordered structure, beginning with the samples having the smallest descriptor distances, samples are drawn from a progressively increasing subset of all correspondences. The ProSaC algorithm terminates when either the non-randomness probability or the maximality probability are exceeded. The non-randomness probability describes the event that all inliers supporting the current pose estimate are by chance considered as correct samples, while actually being incorrect correspondences. The maximality probability describes the event, that a fixed number of correct associations exists, but is still not being found. It is shown, that the proposed approach is computationally superior compared to the native RanSaC approach as valid transformations are detected faster.

2.3.3. Object recognition frameworks

Recent developments have led to several frameworks for object recognition. Detry et al. [DPP09] propose a probabilistic framework for 3D object representation based on a Markov-tree hierarchy. The framework encodes the spatial relation between 3D features in different hierarchies of object parts with increasing granularity using probability distribution over their relative position. Within the Markov random field, features are encoded as hidden variables and the spatial relation between them are node potentials. Recognition is performed by using probabilistic inference techniques to propagate the evidence from the

noisy sensor measurements through the Markov-tree. Common disadvantages of probabilistic frameworks for object recognition are their large storage requirements and the time consuming inference procedure.

A framework for full pose estimation has been presented by Collet et al. [TCS10, CTS11]. They model objects using point features extracted from multiple object views to create a 3D feature model of an object. The main contribution is a novel algorithm termed *Iterative Clustering Estimation* (ICE). ICE jointly solves the correspondence and pose estimation problem by repeatedly clustering a set of features into features belonging to the same object and performing RanSaC-based pose estimation on the individual clusters. However, the system relies on rich textured objects in order to enable recognition. In order to achieve low latency, feature computation and matching are implemented on a GPU.

Grundmann et al. [GES⁺10, GFW11] take object models of 3D SIFT feature points originating from the KIT object model database [KXD12]. They apply Bayesian filtering techniques to split up object recognition into a sequential estimation process of the object's pose. Thereby, the probabilistic filtering is performed on a sequence of multiple sensor measurements taken at consecutive time steps. Grundmann et al. propose a Gaussian observation density that incorporates the noisy location of model points and the noisy location of detected feature points. The observation density is later used to determine a weight for importance sampling in a particle set of possible object poses. However, due to the large computational burden of the probabilistic computations and the SIFT descriptor, the system latency of the proposed approach on a single core system is rather high.

Muja et al. present in [MRBL11] an architecture for object recognition called ReIn. The architecture is implemented on top of ROS [QCG⁺09]. Its major components are interfaces for attention operators that enable the selection of a region of interest (ROI), interfaces for feature detectors that operate on the ROI and pose estimators to compute the location of objects in correspondence with the results from feature matching.

Hinterstoisser et al. [HCI⁺12, HCI⁺11, HLI⁺10] propose a template matching approach for object recognition, that enables fast object modeling by simple template storage and recognition by bit comparisons. In order to speed up the brute-force template matching during recognition a novel binary descriptor for object representation is introduced together with a novel similarity measure based on bit comparisons. The similarity measure is designed to be robust against small shifts and deformations by exactly aligning the gradient orientations from the template descriptor with the input image. To further accelerate the recognition process, Hinterstoisser et al. implemented their method using cache optimizations and SSE (Streaming SIMD Extensions) parallelization.

3. System design

The design of the object recognition framework is tailored, but not limited, to the software and hardware design of the service robot Care-O-bot[®] 3. It serves as a demonstrator for the proposed hardware setup and the developed algorithms. This chapter is divided into separate sections that derive the system design and research need by comparing the state of the art with the requirements from Section 1.4. It concludes by describing the design of the robot and its camera system, and a section describing the software architecture and its implementation into the Robot Operating System ROS.

3.1. Sensor fusion for data acquisition

This section matches the camera sensor technologies from Section 2.1 with the requirements R5 to R10 from Table 1.2. Most sensor technologies are able to meet R5 by directly delivering range and color information. Even though not all time of flight and structured light sensors are able to deliver color information, recent models are usually shipped with a build in color camera to overcome this shortcoming. When comparing the density of the 3D range images, passive stereo is typically not able to meet the required density from R6. Passive stereo relies on texture to infer range data and simply fails when no texture is present. The required measurement range from R7 is covered by all considered sensor types. However, in terms of accuracy only stereo vision and global sensor fusion technologies are able to meet requirement R8. The accuracy of stereo vision relies on a proper arrangement of the two color cameras as will be shown in Section 3.4. Global sensor fusion techniques benefit from the global scope of their optimization algorithms. However, state of the art global optimization algorithms typically need significantly more computation time than all other methods and fail to meet R10. The comparison of the different sensor techniques in relation to the requirements R5 - R10 are summarized by Table 3.1.

ID	Criteria	Passive stereo	Time of flight	Low-cost structured light	Sensor fusion (local/global)
R5	Output	x	(x)	(x)	x/x
R6	Density	-	x	x	x/x
R7	Measurement range	x	x	x	x/x
R8	Accuracy	x	-	-	-/x
R9	Computation time	(x)	x	x	x/-
R10	Sensor layout	x	x	x	x/x

Table 3.1.: Requirement matching for data acquisition.

Based on the considerations from Table 3.1, a sensor setup composed of a stereo vision system and one active range imaging sensor has been selected to be most suitable in the context of this thesis. The selection of the two sensor modalities is based on the idea to combine sparse, but precise, measurements from a stereo vision setup with the dense, but coarse, measurements from the active range sensor. Especially, when creating 3D object models, accuracy and density are crucial in order to capture the precise 3D shape of the object. This enhances the recognition performance and enables a more precise computation of possible grasps to handle the object.

The contribution of this thesis targets the drawbacks of existing global and local methods for sensor fusion. Global methods for sensor combination outperform local approaches in terms of accuracy, while local methods exhibit significantly better performance in terms of computation time. However, none of the proposed methods elaborates the advantages of semi-global methods that reside in between local and global methods to achieve both, acceptable accuracy and computation time. Hirschmüller [Hir08] proposed a semi-global matching algorithm for stereo processing using a predefined set of cost paths across the target depth map in order to restrict optimization instead of globally optimizing over all possible paths. This thesis extends the proposed method of Hirschmüller with the information of the ToF sensor to improve the accuracy and density of the stereo algorithm under acceptable timing conditions. The results of the proposed procedure have been published in [FAV11].

3.2. Object modeling

Referring to the requirements for object modeling from Table 1.3, most object recognition frameworks are able to model typical household objects meeting requirements R13 and R14. However, the creation of dense 3D object models suitable for grasping is often not an integral component of those frameworks. Among the presented frameworks only Grundmann et al.

meets R11 and R12 by integrating the KIT object model database, that explicitly enables the computation of accurate grasps. However, the KIT object modeling setup is rather large and stationary, which violates the usability requirement R15. Table 3.2 associates the requirements for object modeling with the presented frameworks, showing that usability and the creation of dense object models suitable for grasping are still open issues to be addressed.

ID	Criteria	[DPP09]	[CTS11]	[GFW11]	[MRBL11]	[HCI+12]
R11	Output	-	-	x	-	-
R12	Accuracy	-	-	x	-	-
R13	Object dimensions	x	x	x	x	x
R14	Object type	x	x	x	x	x
R15	Usability	-	-	-	-	-

Table 3.2.: Requirement matching for object modeling.

This thesis addresses requirements R11 - R15 by presenting a method for dense object modeling directly on the robot using its manipulator and camera system as well as by introducing two additional training setups, one using a turn table and one using a chessboard. This avoids the explicit need for a robotic manipulator for object modeling. Initial work conducted within the scope of this thesis and published in [AFV10] proposes a fastSLAM-based in-gripper object modeling approach, which is able to cope with multi-occurrences of similar textures on the object’s surface. This approach is further developed and the information filter is replaced by a bundle adjustment algorithm that enables a faster registration of the individual object views while still meeting requirement R12.

The thesis proposes two binary descriptors for textured and texture-less object modeling that enable the usage of rapid bit operations to accelerate the descriptor computations. When addressing textured object recognition, recent fast-to-compute descriptors achieving remarkable recognition rates have been presented e.g. ORB by [RRKB11]. However, the descriptor still lacks invariance against scale changes. This thesis proposes a scale invariant extension of the binary feature descriptor ORB, which is fast to compute while still being as descriptive as SURF. The presented results have been published in [FABV12].

In order to distinctly describe texture-less objects, a global histogram-based descriptor is presented that aggregates 2D and 3D gradient information from a local binary descriptor. Compared to the current state of the art, the descriptor exhibits scale and rotation invariance. Additionally, the underlying binary descriptor is computed faster than competing methods by the use of dynamic programming. The presented results have been published in [FBAV13].

3.3. Object recognition

ID	Criteria	[DPP09]	[CTS11]	[GFW11]	[MRBL11]	[HCI ⁺ 12]
R16	Output	x	x	x	x	x
R17	Object types	x	-	-	x	x
R18	Number of objects	x	-	-	x	x
R19	Recognition range	x	x	x	x	x
R20	Recognition time	-	x	-	x	x
R21	Recognition accuracy	-	x	x	(-)	(-)

Table 3.3.: Requirement matching for object recognition.

Comparing the requirements from Table 1.4 with the framework presented by Detry et al. [DPP09], it is clear, that it fails to meet requirement R20. When applying three Markov hierarchies, recognition takes approximately 10 s per object. Additionally, the presented framework does not achieve the required accuracy from R21 for any of the presented experimental setups. The MOPED framework introduced by Collet et al. [CTS11] is able to process a scene image with a resolution of 1600×1200 in about 2.1 s. It is stated to achieve an average recognition accuracy at distances ranging from 0.4 m to 1.2 m of 5 mm in translation and 5.69° in rotation. However, it fails in recognizing texture-less objects due to its usage of local point features, therefore violating requirement R17. Similarly, the framework presented by Grundmann et al. is not able to recognize texture-less objects and additionally violates requirement R20 with an average recognition time of more than 5 s. The framework presented by Hinterstoisser et al. [HCI⁺12] is able to recognize rich textured objects as well as texture-less objects and meets requirements R16 - R20. However, R21 is not directly addressed in the evaluations as accuracy increases with the number of templates which are used for recognizing an object. Similarly, Muja et al. [MRBL11] are able to meet requirements R16 - R20, but do not report any results concerning rotational or translational accuracy. Additionally, Hinterstoisser et al. and Muja et al. are not able to meet all requirements for object modeling, as shown in Section 3.2.

The ReIn architecture [MRBL11] provides a framework that is able to incorporate a multitude of different object recognition algorithms to tackle the detection of both object types. This thesis is in the line with the idea of ReIn and proposes a novel architecture with a concrete implementation for the recognition of textured and texture-less objects. Compared to ReIn, it follows different strategies for feature description and pose estimation. The recognition of textured objects is based on a scale invariant extension of the feature descriptor ORB [RRKB11]. In order to increase the robustness of recognition, data

association is subject to a spatial constraint to take account for the spatial expansion of an object. Results of the proposed approach have been presented in [FABV12].

In order to recognize texture-less objects, the thesis proposes an adaptive sliding window approach to build up a probability map for prominent object locations. Based on a non-maximum suppression algorithm, the dominant object locations are selected. The presented approach has been published in [FBAV13].

3.4. Hardware setup

Figure 3.1 shows the service robot Care-O-bot[®] 3 [RCF⁺09]. It serves as demonstrator for the presented object recognition framework and conforms with requirement R1 from Table 1.1. Viewed from the bottom up, the robot is equipped with a mobile base composed of four omnidirectional drive wheels, which enable the robot to move forwards, backwards and sideways. Three laser scanners monitor the local environment to ensure that its movements are collision free. The control software is running on a computer rack placed in the center of the torso. The upper torso is flexible and equipped with 4 DoF. A turnable touch screen panel allows for an intuitive and convenient user interaction and a 7 DoF redundant manipulator with a three finger gripper enables the execution of manipulation tasks.



Figure 3.1.: Hardware design of the service robot Care-O-bot[®] 3. The first two images show the robot's cover. It is flexible in the range of the torso to allow the robot to express itself by gestures. The other images show the robot's hardware design below the cover.

Additionally, two alternative demonstrators for stand-alone vision systems are proposed in accordance to requirement R1 from Table 1.1. Instead of using a robotic manipulator, the stand-alone system from Figure 3.2(a) consists of a turn table in order to rotate an object in front of the camera system. An even simpler setup is illustrated by Figure 3.2(b), where the object is stationary and the camera system is moved around it.



Figure 3.2.: Stand-alone hardware setups without using a robotic platform.

The camera system is the same for all demonstrators. It consists of two standard color cameras and a one-shot range camera (Figure 3.3). One-shot range sensors, in contrast to laser scanners with a pan tilt unit, create the 2.5D data for each pixel of a scene at the same time. A major advantage of the active range sensor compared to a state of the art stereo vision system is its capability to provide depth data for each image pixel in real time. However, state of the art stereo systems still significantly outperform current one-shot range sensors in terms of spatial resolution and accuracy. Therefore, the usage of a single range sensor alone is not applicable for many vision tasks and a combination of both systems' advantages is desirable. The development of one-shot range sensors is still a highly active field of research and new products are regularly presented. These developments are also reflected in the different range sensors that have been deployed in the camera setups of Care-O-bot[®] 3, as shown in Figure 3.3.

The layout of the sensor setup is determined by the baseline of the stereo camera system, the aim to maximize the overlap of the cameras' field of view, and the width of the sensor head which limits the baseline of the stereo camera system to a maximal value of 130 mm.

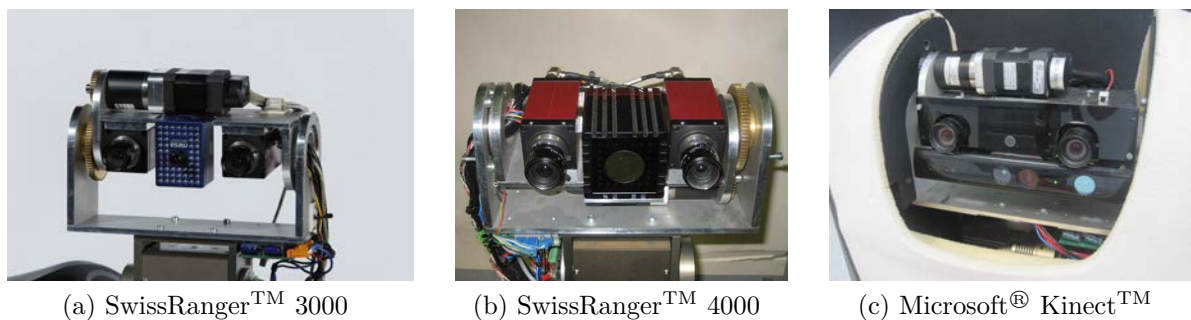


Figure 3.3.: Sensor setups for the service robot Care-O-bot[®] 3 with different active range cameras.

The baseline influences the range and accuracy of the distance measurements z from stereo vision, as expressed by (2.1). Typically, stereo algorithms consider only a small subset of disparities, e.g. ranging from 16 pixel to 64 pixel, in order to limit the search space for the correspondence problem. However, this also restricts the range of measurable distances z to some interval. According to R7 from Section 1.4, measurable distances for object recognition must cover close ranges, starting from 700 mm in order to recognize objects and to create detailed object models, as well as ranges up to 1700 mm in order to search for objects.

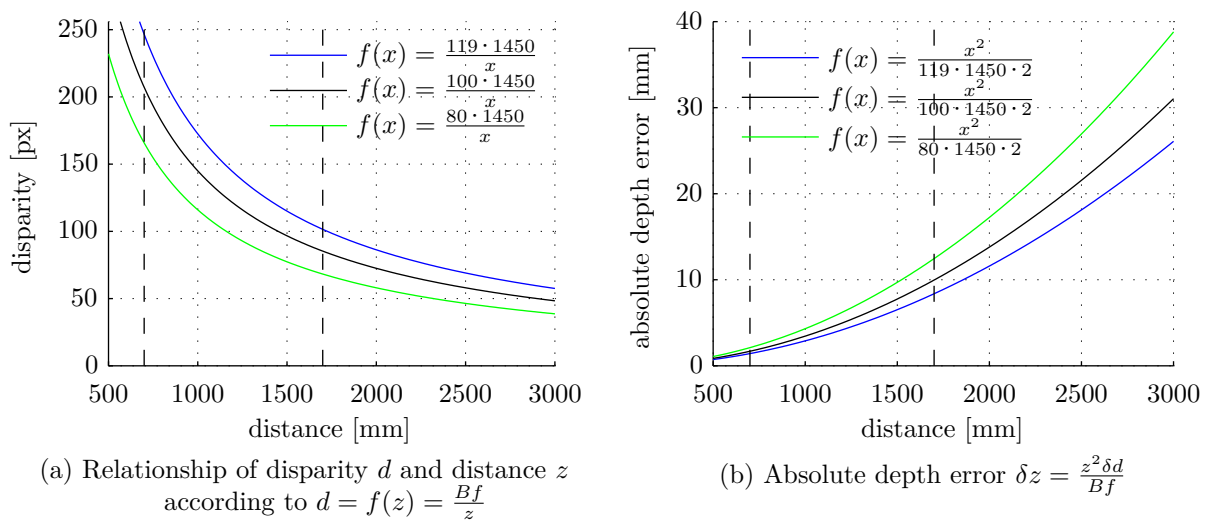


Figure 3.4.: Characteristics of the stereo vision setup using a focal length $f = 1450$ pixel and a baseline of $B = 119$ mm. The vertical black lines indicate the desired measurement range.

With regard to the desired measurement range and the available space in the sensor head, a baseline of 119 mm has been selected for the stereo vision setup. Figure 3.4(a) shows the relationship between disparity d and distance z for the selected baseline length in comparison to baselines of 100 mm and 80 mm. The focal length of 1450 pixel has been determined through stereo calibration using [Bou08] and the black lines indicate the desired measurement range. In general, a larger baseline more accurately resolves the measured distances. However, by increasing the size of the baseline, the overlap of the two stereo cameras is decreasing as well, reducing the common field of view. The selected baseline of 119 mm is a compromise between overlap and accuracy within the given measurement range. For this setup, a distance of 700 mm corresponds to a disparity of 246.5 pixel and a distance of 1700 mm corresponds to a disparity of 101.5 pixel resulting in a total of 145 different disparities in order to express the distance values within the measurement range. Smaller baselines would result in a smaller range of disparities, larger baselines would further

decrease the overlap. In order to measure the accuracy of the distance measurements, the absolute depth error of the proposed setup is computed according to [CC92] in (3.1).

$$\delta z = \frac{z^2 \delta d}{Bf} \quad (3.1)$$

Given that $z = \frac{Bf}{d}$ as explained by (2.1), the absolute depth error is given by the partial derivatives of z with respect to the focal length f , the baseline B , and the disparity d . However, the inaccuracies in f and B are negligible in reality. Therefore, the absolute error is simplified to $\delta z = \frac{Bf\delta d}{d^2}$, the partial derivative with respect to the disparity d . Figure 3.4(b) shows the evaluation of the absolute depth error for the selected baseline length in comparison to baselines of 100 mm and 80 mm. Usually, stereo algorithms interpolate disparities to $\frac{1}{2}$ pixel and smaller. Therefore, an additional factor of $\frac{1}{2}$ has been included in the equation. The absolute depth error is below 1.5 mm for distances up to 700 mm. For 1500 mm, it increase to 6.5 mm and for 1700 mm it is approximately 8.4 mm. As expected, the error increases for smaller baselines.

The active ToF range sensor has been positioned in between the two color cameras in order to maximize the common field of view as indicated by Figure 3.5. The field of view is determined by the angle of views α and β of the stereo and ToF sensor system. The housing

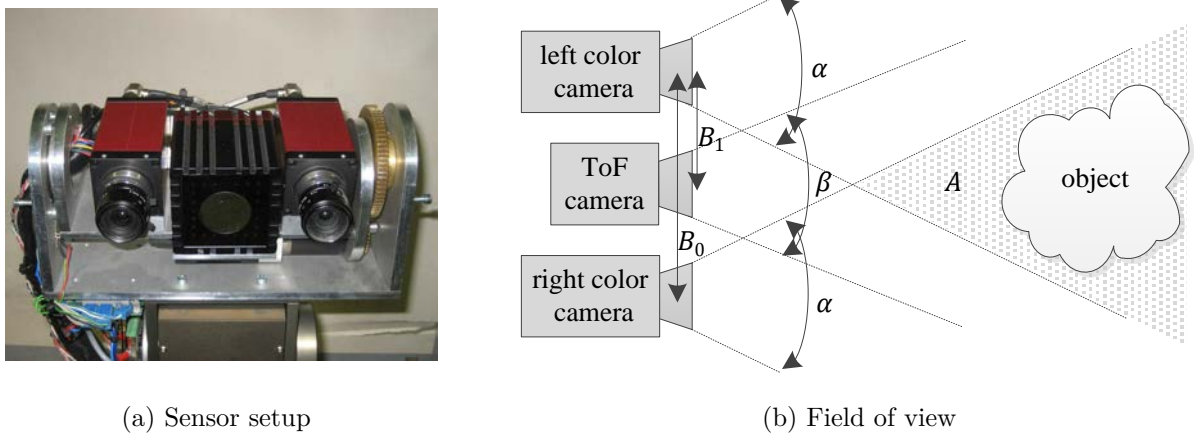


Figure 3.5.: Top view on the sensor setup using the ToF camera SwissRangerTM 4000 for active range sensing. A indicates the sensors' common field of view.

dimensions of the KinectTM camera are larger compared to a ToF camera. Therefore, it is not possible to position it in between the two stereo cameras. Instead, the KinectTM is mounted on top of the color cameras, with the infrared sensor vertically aligned to the left color camera.

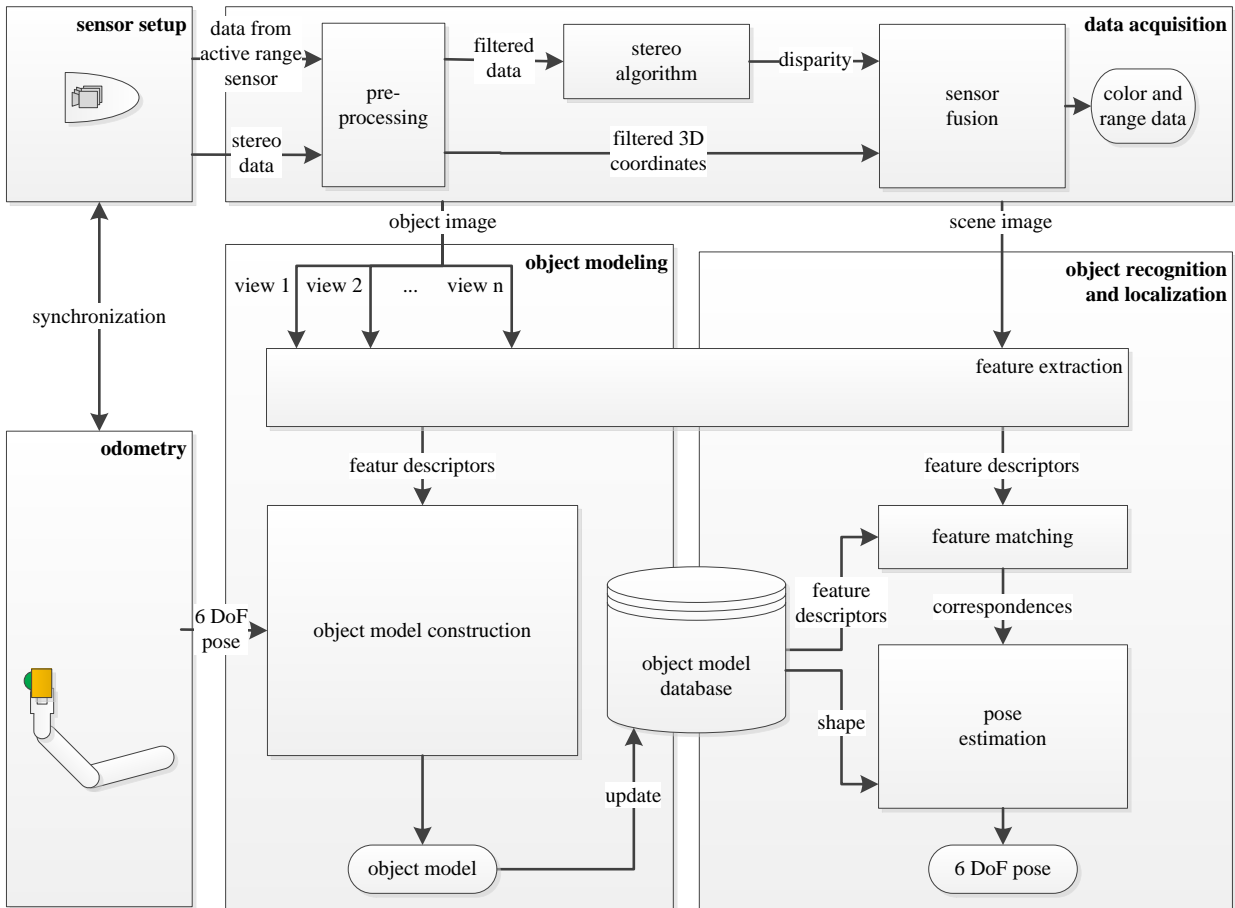


Figure 3.6.: Software architecture for object recognition and localization using a stereo and active range camera.

3.5. Software design

The software design has been developed to fit to requirement R3 from Table 1.1. It is integrated into the robot operating system ROS [QCG⁺09], which is running on the service robot Care-O-bot[®] 3. ROS is an open source meta-operating system protected by the BSD license. Similar to an operating system for PCs, it provides functionalities like hardware abstraction, package management, inter-process message passing, hardware low-level control, and many more. However, ROS also shares characteristics from a middleware by supporting distributed computing systems via its messaging system. Furthermore, ROS provides tools to support the development of code e.g. to simplify its configuration, building, running, or debugging. A major advantage of using ROS is its strong community dedicated to robotics, that drives its development and provides a huge amount of open source libraries for various robotic tasks.

An overview of the software design of the object recognition framework developed in this thesis is given by Figure 3.6. It is structured according to the three main parts of object recognition as outlined by Section 1.2 which are data acquisition, object modeling,

and object recognition and localization. A library for feature extraction and a database component complement the design. All modules are encapsulated into libraries that provide defined interfaces for interaction. Sensory components include a stereo camera system and an active range sensor. Odometry data is considered only for object modeling, for which the pose of the arm during image retrieval is transmitted. Each library is independently usable from all other components and ROS specific function calls are separated from the core functionalities in order to assure the generality and portability of the software.

Beginning with *data acquisition*, the sensory data from the stereo and active range imaging system are combined into a single image. Depending on the current task, the image data either shows the object from a specific viewpoint for object modeling, or it shows an arbitrary scene from which the locations of known objects have to be inferred. The image data is passed to the *feature extraction* library, which extracts 2D and 3D feature points and descriptors either with a global or local scope. Concerning *object modeling*, the feature descriptors are added to the partial object model which has already been created from preceding object views. Once the last view has been integrated, the complete object model is stored in the database. *Object recognition* loads the object models from the database and compares the feature descriptors from the scene image with the descriptors from the model to establish point-based correspondences. Pose estimation uses the shape information from the object model to filter the feature point correspondences. Based on the remaining correspondences, it computes a best fitting pose by solving an optimization problem.

4. Sensor Fusion

Within the context of this thesis, sensor fusion refers to the combination of sensory data from an active range sensor and two color cameras used for stereoscopic vision. The combination of the different sensor modalities has the objective to create information that exceeds the quality of each individual source. In the present sensor setup, this relates to the creation of accurate 2.5D point clouds with associated color information even in unstructured image areas. Even though the type of the active range sensor is not relevant for the proposed procedure, the descriptions of this section are tailored to the usage of a ToF sensor without loss of generality. A detailed description on the characteristics of both sensor modalities is given in Section 2.1.

Figure 4.1 gives an overview of the complete procedure which is presented in more detail within the following sections. Beginning with the sensor setup, the stereo system delivers synchronized color images from its two color cameras, whereas the ToF sensor delivers 2.5D data and a corresponding amplitude image directly. The amplitude image is representing the signal strength of the ToF data, which is utilized for filtering in connection with a succeeding wavefront propagation filtering step. The stereo images are undistorted and rectified before a standard stereo algorithm delivers disparity guesses with corresponding reliability indicators represented as costs. Together with the resulting depth information, the color information of each rectified pixel is saved and returned at the end of the procedure. The range data from the ToF sensor is projected onto the color image of the stereo rig in order to express it within the same coordinate system as the range data from stereo. After another filtering step which is based on the disparity guesses from stereo, the ToF range data is transformed into corresponding disparity values with associated cost values in order to indicate their reliability. Having disparity cost values from stereo and ToF, a cost aggregation step accumulates the information to return a single disparity image after a final post-processing step.

The main novelty of the presented algorithm for sensor fusion is the usage of a semi-global cost aggregation step based on the principles of *Belief Propagation*, that combines stereo and ToF data. It is shown that the proposed procedure results in more accurate disparity images and increases the density of the delivered range data compared to the usage of stereo vision or a single active range sensor alone. The work and results of this section have already been published in [FAV11].

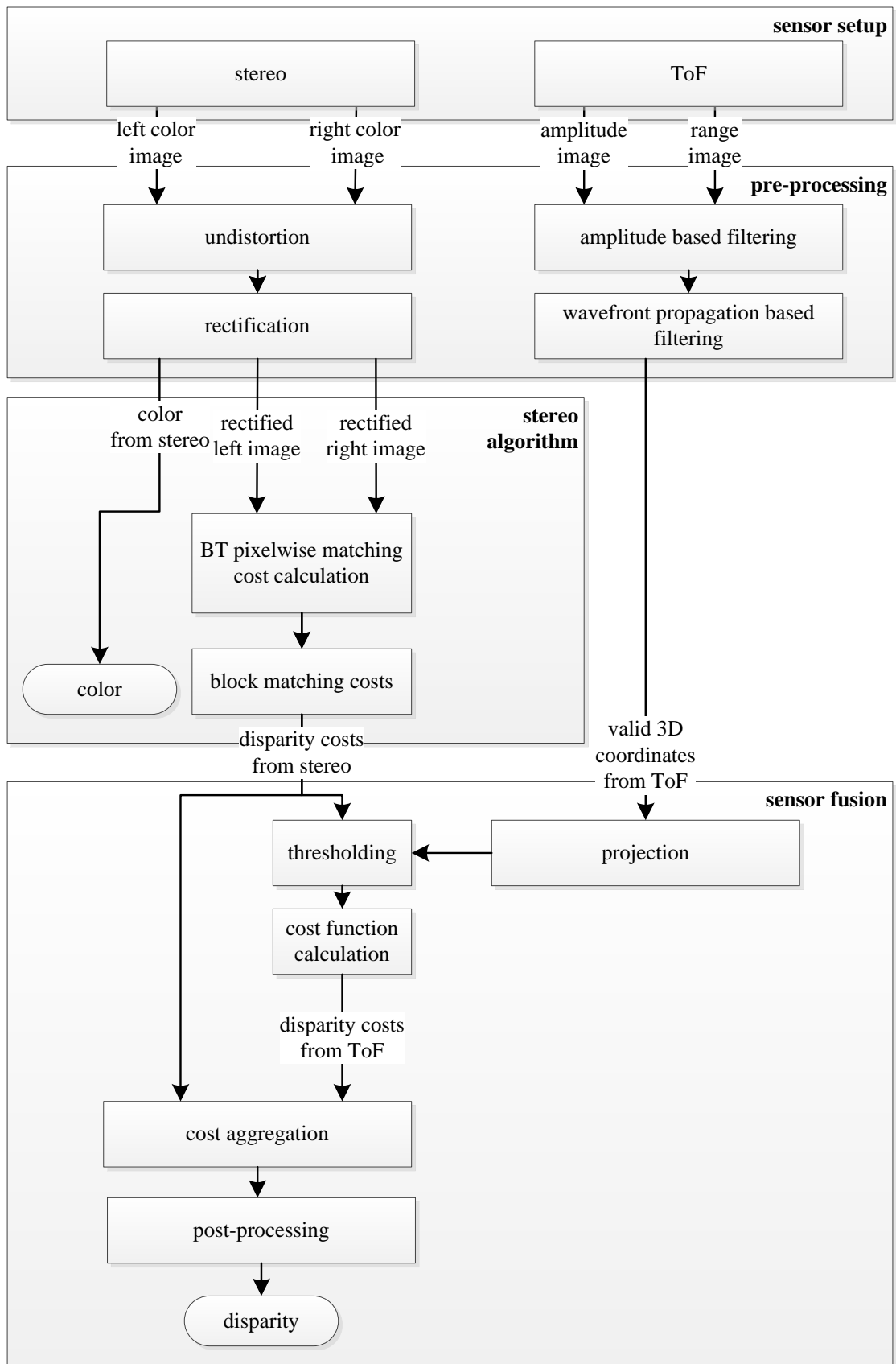


Figure 4.1.: Schematic overview of the individual processing steps for sensor fusion.

4.1. Calibration

Calibration determines the cameras' extrinsic and intrinsic parameters as illustrated by Figure 4.2. *Extrinsic parameters* describe the relative position of the cameras with respect to the world coordinate frame \mathbf{W} by a 3×3 rotation matrix \mathbf{R} and a 1×3 translation vector \mathbf{T} . The cameras' *intrinsic parameters* determine the perspective projection of 3D points onto the image plane and encompass the focal length \mathbf{f} , the coordinates of the principle point \mathbf{o} , skew coefficient θ , and distortion parameters. Assuming a perfect pinhole camera model, the principle point $\mathbf{o} = (c_u, c_v)$ is given by intersecting the principle axis with the image plane, which does usually not coincide with its center. The focal length $\mathbf{f} = (f_x, f_y)$ is determined by the distance from the lens center to the image plane measured in pixel. As physical pixel are usually rectangular, \mathbf{f} is expressed in dependence of the pixel size in x - and y -direction. The skew coefficient θ holds the angle between the coordinate axes of the coordinate system \mathbf{C} , whose axis might be slightly skewed compared to an orthonormal coordinate system. The distortion parameters describe the *radial distortion* arising from geometrical distortions of the optics and *tangential distortions* due to the lens and the image plane not being exactly parallel in real cameras. Without any distortion, a straight line L to be projected onto a straight line on the image plane. Distortion, however, causes a straight line L to be mapped onto a curved line l . A camera's intrinsic parameters are saved in the 3×3 *intrinsic matrix* \mathbf{M} .

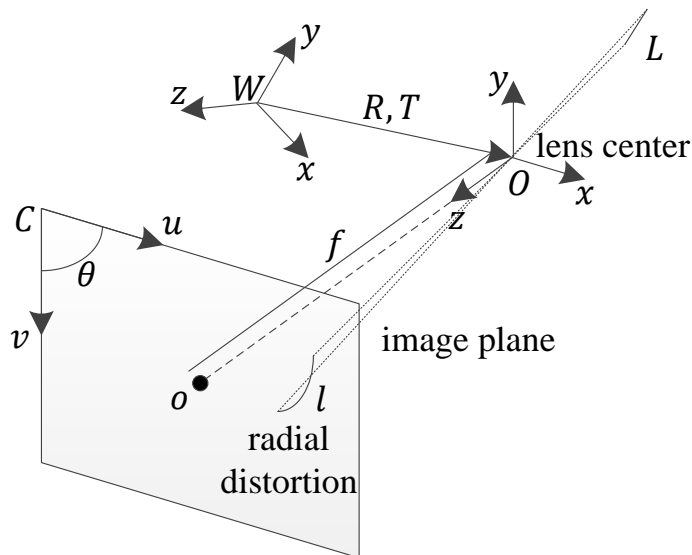


Figure 4.2.: Visualization of intrinsic and extrinsic camera parameters, which are determined through calibration.

For the intrinsic and extrinsic calibration of the three camera sensors, the method of Zhang [Zha00] is applied by using Bouguet's Matlab calibration toolbox [Bou08]. The calibration of ToF sensors is problematic and tends to be inaccurate due to the low resolution and noisy intensity data. Therefore, this thesis follows the ideas of Lindner and Kolb [LK06]

by applying an upscaling method with bi-linear interpolation before estimating the intrinsic parameters based on the intensity image. Lindner and Kolb show, that this effectively increases the calibration quality. When using an active range sensor with a higher image resolution, this step may be omitted. The extrinsic calibration results are used to express 3D data from the ToF sensor relative to the coordinate system of the stereo rig, hence, enabling an association of measured range from ToF data to measured disparity from stereo vision.

The stereo rig is initialized with the extrinsic and intrinsic color camera parameters. Using again Bouguet’s Matlab calibration toolbox, the stereo rectified 4×4 *projection matrix* \mathbf{M}_{rect} is computed. Rectification simplifies the correspondence problem from stereo vision by transferring the left and right image plane into a coplanar, row-aligned arrangement parallel to the baseline. This limits the search space for correspondences on the left and right images to horizontal scanlines. The stereo rectified projection matrix \mathbf{M}_{rect} enables an association of 3D coordinates from the physical scene with 2D coordinates from the rectified color image and substitutes the intrinsics matrix for the stereo cameras.

Based on the extrinsic and intrinsic camera parameters, it is possible to compute a simple mapping of the range information recorded by the ToF sensor onto the color information originating from e.g. the left color camera. As the pose of the cameras’ coordinate system with respect to the world frame is known, the pose of the ToF coordinate system with respect to the left color camera coordinate system may be induced as a 3D translation vector \mathbf{T}_t^l and a 3×3 rotation matrix \mathbf{R}_t^l . Therefore, given a 3D point \mathbf{P}_t relative to the ToF sensor, its corresponding 3D coordinates relative to the left 2D color camera are computed according to (4.1).

$$\mathbf{P}_l = \mathbf{R}_t^l \mathbf{P}_t + \mathbf{T}_t^l \quad (4.1)$$

To compute the projective transform from the 3D coordinates $\mathbf{P}_l \in \mathbb{R}^3$ in meters to the corresponding 2D image coordinate $\mathbf{p}_l \in \mathbb{R}^2$ in pixel on the image plane of the left color sensor, \mathbf{P}_l is multiplied with the 3×3 intrinsics matrix \mathbf{M} according to (4.2).

$$\mathbf{p}'_l = \mathbf{M} \mathbf{P}_l \quad (4.2)$$

$\mathbf{p}'_l = (u, v, w)$ is given in *homogeneous coordinates*, therefore it is divided by w to recover its image coordinates $\mathbf{p}_l = (\frac{u}{w}, \frac{v}{w})$ in pixel. The procedure is repeated for each pixel of the ToF camera until all points are mapped to their corresponding color information from the color cameras. In order to take advantage of the high resolution color image, the low resolution range image from the ToF camera is upscaled by a factor of 3 using bi-linear interpolation prior to the sensor fusion process. The result is an image with each pixel holding 3D coordinates from the ToF sensor and color information from the color camera. By upscaling the image of the 3D range image, more color information is preserved during

sensor fusion. This relates to the fact that each interpolated range value is assigned a color value from the native color image.

The outlined procedure is the simplest method for mapping 3D range data to 2D color images in order to obtain a 3D point cloud with color information. However, the procedure has the major drawback that it does only consider the information of a single color camera and that it does not incorporate the accurate 3D information from the stereo camera rig. Therefore, further processing steps dealing with the integration of the stereo camera rig are presented in the following sections.

4.2. Pre-processing

Pre-processing filters the range image of the ToF sensor to remove speckles originating from noisy range values. Figure 4.3 gives an impression of the filtering effects. It shows the raw 3D values recorded by a ToF camera and its filtered counterpart. The most prominent noise originates from tear-off edges on object borders and noise from objects that exceed the non-ambiguity range. Common filtering techniques apply median filters or fixed amplitude thresholding [KKHA09] to remove noisy range measurements. Amplitude filtering removes most of the noise originating from measurements outside the non-ambiguity range. However, neither amplitude filtering nor median filtering effectively remove tear-off edges. Therefore, the thesis proposes the application of wavefront propagation with prior amplitude thresholding for ToF data filtering.

Initially, amplitude thresholding enables the filtering of weak signals originating from far objects outside the non-ambiguity range of the ToF sensor or from objects with bad reflection properties like transparent or shiny objects. A large amplitude corresponds to a strong signal, which increases the reliability of the range measurement. A low amplitude, however, corresponds to a weak signal and therefore the confidence in the corresponding range measurement is low. By comparing the amplitude of the reflected infrared signal against a predefined threshold, measurements with a low amplitude value are discarded.

Then, wavefront propagation iteratively expands the neighborhood of each remaining pixel. Given a pixel $\mathbf{p} = (u, v)$ that passed the amplitude thresholding, its 4 direct neighbors $\{(u - 1, v), (u + 1, v), (u, v - 1), (u, v + 1)\}$ are examined. If the distance measured along the z axis of the range camera system exceed a given threshold t_z , the neighboring pixel is discarded. Otherwise, the neighboring pixel is added to the set of valid neighbors N_p and the expansion continues with its 4 neighbors, respectively. Once no more neighbors are added to N_p , the size of N_p is compared against a speckle threshold t_s . When $\|N_p\| < t_s$ the pixel's range value is labeled as invalid. Otherwise, it is considered as a valid range value, because neighboring pixel support its measurement as only range values of pixel with a sufficient number of close-by (in terms of depth differences) neighbors will survive. This

directly corresponds to the smoothness assumption made for global stereo vision, which assumes that for most image parts range values vary only smoothly in local neighborhoods.

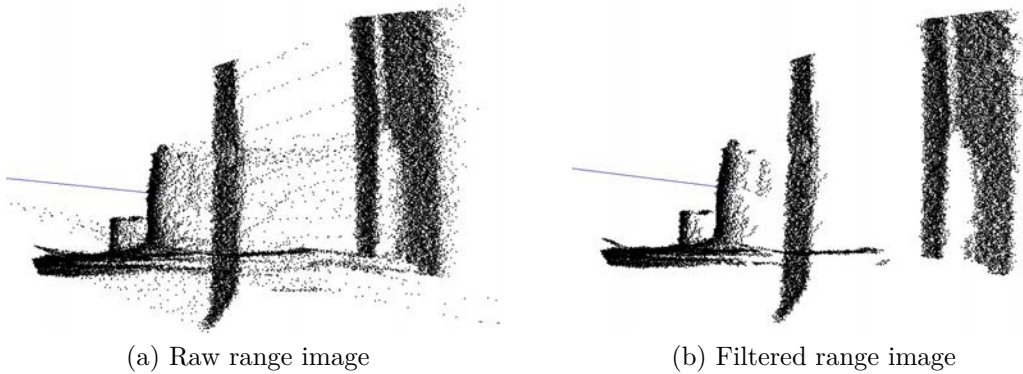


Figure 4.3.: Pre-processing the raw range data delivered by a ToF sensor using wavefront propagation. The scene image has artificially been rotated by 90° for a better presentation of the filtering effects.

4.3. Projection

After pre-processing, the valid 3D coordinates from the ToF camera are projected onto the image plane of the left color camera, as shown in Figure 4.4(b). Let \mathbf{R}_t^l be the extrinsic rotation matrix and \mathbf{T}_t^l the extrinsic translation vector that relates the rectified left color camera coordinate system with the coordinate system of the ToF sensor. Then, (4.1) expresses the 3D ToF measurement $\mathbf{P}_t = (x, y, z)$ with respect to the left color camera. Compared to (4.2), the projection matrix \mathbf{M}_{rect} for the rectified stereo camera system replaces the intrinsic matrix \mathbf{M} . The corresponding 2D image coordinates $\mathbf{p}_l = (\frac{u}{w}, \frac{v}{w})$ within the left color image are inferred from the homogeneous coordinates of $\mathbf{p}'_l = (u, v, w)$ according to (4.3) and their corresponding disparity d is inferred by applying (2.1).

$$\mathbf{p}'_l = \mathbf{M}_{rect} \mathbf{P}_t \quad (4.3)$$

The projected 3D ToF data covers only a small part of the high resolution color image. In order to increase the influence of the projected ToF measurements on the following semi-global cost aggregation step (Section 4.5), they are propagated to their individual neighborhood using wavefront propagation. To reduce computation time, the propagated range values are not interpolated, but copied as they are. To cope with the copied data later on, the uncertainty of each propagated value is increased proportional to the distance from its origin (Section 4.4).

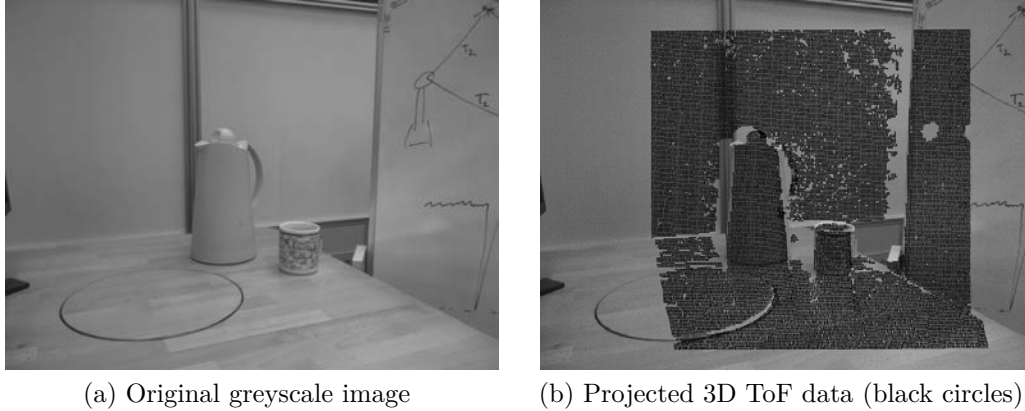


Figure 4.4.: Projection of filtered ToF data onto the high resolution greyscale image.

4.4. Pixelwise matching costs

The projected and propagated ToF data serves as an initial disparity guess for each affected pixel of the color image. However, due to occlusion originating from the different viewing angles of the color and the ToF cameras, wrong ToF range estimates may still have been assigned to corresponding pixel of the color image. Therefore, a novel method is proposed to remove these outliers as follows. Pixelwise stereo matching costs are calculated on the rectified color image pair using the proposed disparity d from the ToF sensor. For the calculation of matching costs, the pixel dissimilarity measure of Birchfield and Tomasi [BT98] is applied. It determines the difference of intensities in the range of half a pixel in both horizontal directions. To improve robustness and to lower the probability of ambiguous matchings, block matching is used to accumulate neighboring dissimilarity measures. The matching costs are accumulated for a squared $n \times n$ region around each pixel resulting in total pixelwise matching costs $C_{BM}(\mathbf{p}, d)$ of pixel $\mathbf{p} = (u, v)$ and disparity d . The block matching based cost measure is given by (4.4).

$$C_{BM}(\mathbf{p}, d) = \sum_{i=-n}^n \sum_{j=-n}^n C_{BT}(u-i, v-j, d) \quad (4.4)$$

$C_{BT}(\mathbf{p}, d)$ returns the dissimilarity measure of Birchfield and Tomasi at pixel coordinate $\mathbf{p} = (u, v)$ and disparity d . For each projected and propagated ToF range measurement, the corresponding pixelwise block matching costs $C_{BM}(\mathbf{p}, d)$ for the proposed disparity d are compared against a fixed threshold $t_{BT} \in \mathbb{R}$. If $C_{BM}(\mathbf{p}, d)$ exceeds t_{BT} , the ToF based disparity guess does not fit to the stereo data and the pixel's disparity value is rejected.

Finally, a discrete cost function is calculated for each pixel. In absence of valid ToF measurements, a pixel's cost function represents directly the stereo block matching costs, which have already been calculated to reject the invalid ToF measurements by (4.4). However, in

the presence of a valid ToF measurement for a pixel \mathbf{p} , the cost function approximates a reversed Gaussian distribution according to (4.5)

$$\begin{aligned} \mathcal{N}_r: \mathbb{R} &\rightarrow \mathbb{R} \\ x &\mapsto \mathcal{N}_r(x; \mu, \sigma^2) = 1 - \mathcal{N}(\mu, \sigma^2) \end{aligned} \quad (4.5)$$

with $\mu \in \mathbb{R}$ corresponding to the ToF-based disparity d and $\sigma \in \mathbb{R}$ corresponding to an expected 1% measurement noise in proportion to the measured distance. The fact, that range values from propagated ToF data have solely been copied and not interpolated from a close-by pixel, is taken into account by increasing the expected measurement noise in proportion to its distance to the measurement's origin in pixel coordinates. To accelerate computations and to incorporate the possibility that ToF based disparity guesses may still be wrong, the cost function becomes constant as disparity differences become larger than 2σ as given in (4.6).

$$C_{ToF}(\mathbf{p}, d) = \begin{cases} k \mathcal{N}_r((\mu - d); \mu, \sigma^2), & \text{if } |d - \mu| < 2\sigma \\ k, & \text{otherwise} \end{cases} \quad (4.6)$$

The factor $k \in \mathbb{R}$ from (4.6) corresponds to the maximal costs induced by the cost function $C_{ToF}(\mathbf{p}, d)$. The final pixelwise cost function is given by summarizing the individual cost functions (4.4) and (4.6) into (4.7).

$$C(\mathbf{p}, d) = \begin{cases} C_{ToF}(\mathbf{p}, d), & C_{BM}(\mathbf{p}, d) < t_{BT} \\ C_{BM}(\mathbf{p}, d), & \text{otherwise} \end{cases} \quad (4.7)$$

4.5. Cost aggregation

Cost aggregation is based on the work of Hirschmüller [Hir08]. To avoid ambiguous matching costs for different disparities, Hirschmüller connects the pixelwise matching costs with smoothness constraints using the energy function $E(D)$ given by (4.8).

$$\begin{aligned} E(D) = \sum_{\mathbf{p} \in I} &\left(C(\mathbf{p}, d_p) + \sum_{\mathbf{q} \in N_p} P_1 T(|d_p - d_q| = 1) \right. \\ &\left. + \sum_{\mathbf{q} \in N_p} P_2 T(|d_p - d_q| > 1) \right) \end{aligned} \quad (4.8)$$

$E(D)$ is evaluated over a disparity image $D \in \mathbb{N}^{w \times h}$, holding for each pixel of an image I with height h and width w a corresponding disparity value d . $C(\mathbf{p}, d_p)$ denotes the pixelwise matching costs of pixel \mathbf{p} at disparity d_p , which has been adapted compared to the original

approach according to (4.7). The function $T: \{\text{true}, \text{false}\} \rightarrow \{0, 1\}$ evaluates to 1 or 0 if its argument evaluates to true or false, respectively. The remaining terms apply smoothness constraints on the neighborhood N_p of \mathbf{p} with the second term penalizing the disparity cost of \mathbf{p} by an additional cost of $P_1 > 1$, when neighboring pixel have a disparity difference of 1 to \mathbf{p} . The third term applies a larger penalty $P_2 > P_1$ for all neighboring pixel having a disparity difference larger than 1.

In order to select the most appropriate disparities for all pixel, $E(D)$ must be minimized over all possible combinations of disparities for all image pixel. However, performing global optimization over the complete 2D image space is an NP-complete problem, which prevents the fast computation of exact solutions. Therefore, this thesis extends the semi-global approach of Hirschmüller for stereo vision to the area of sensor fusion by following 16 individual 1D directions around each pixel in order to minimize the energy function. The directions relative to pixel $\mathbf{p} = (u, v)$ are pointing towards its eight directly surrounding pixel $\{(u-1, v), (u-1, v-1), (u, v-1), (u+1, v-1), (u+1, v), (u+1, v+1), (u, v+1), (u-1, v+1)\}$ and the eight pixel given by $\{(u-2, v-1), (u-1, v-2), (u+1, v-2), (u+2, v-1), (u+2, v+1), (u+1, v+2), (u-1, v+2), (u-2, v+1)\}$.

The definition of an individual cost path is given by (4.9), where $\mathbf{r} = (u_r, v_r)$, $r = \{1, 2, \dots, 16\}$ represents one of the traversed directions of the individual cost path and i traverses over all disparities except d , $d+1$ and $d-1$.

$$\begin{aligned}
 L_r(\mathbf{p}, d) = & C(\mathbf{p}, d) + \min \left(L_r(\mathbf{p} - \mathbf{r}, d), \right. \\
 & L_r(\mathbf{p} - \mathbf{r}, d-1) + P_1, \\
 & L_r(\mathbf{p} - \mathbf{r}, d+1) + P_1, \\
 & \left. \min_{i < d-1, i > d+1} L_r(\mathbf{p} - \mathbf{r}, i) + P_2 \right) - \min_k L_r(\mathbf{p} - \mathbf{r}, k)
 \end{aligned} \tag{4.9}$$

The first minimization term adds the minimal cost of the preceding pixel from the current optimization direction r , when selecting disparity d for the current pixel. The costs of the preceding pixel are penalized depending on its disparity difference to the currently selected disparity d as explained for (4.8). In order to keep the value of $L_r(\mathbf{p}, d)$ not constantly growing while traversing the cost path, the minimum path cost of the preceding pixel is subtracted at the end of the equation. The overall costs for a specific disparity d , which are computed by aggregating the costs from all 16 cost paths, are given by (4.10).

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_r(\mathbf{p}, d) \tag{4.10}$$

By selecting the disparities that minimize (4.10) for each pixel \mathbf{p} , a combined optimization of disparity costs originating from ToF data and stereo data is achieved.

The algorithm evaluates (4.10) by traversing the target image from top to bottom and from left to right. However, in order to achieve the requested computation time from requirement R9, only the 5 direct upper individual 1D optimization paths are aggregated to solve $E(D)$.

4.6. Post-processing

The resulting disparity image is post-processed according to the paper of Hirschmüller and methods from standard stereo vision algorithms. At first, the uniqueness of the disparity is tested by comparing it against all other possible disparity values. The disparity is rejected, if other disparities have similar cost values. Then, a consistency check between the left and right color images based on the selected disparity is performed to ensure that left-right image pixel pairs are unique. To reduce quantization errors originating from the discrete disparity values, the determined disparity d with minimal cost $S(\mathbf{p}, d)$ is interpolated over the cost values of the neighboring disparity costs $S(\mathbf{p}, d - 1)$ and $S(\mathbf{p}, d + 1)$ using a quadric function equation. This results in the interpolated disparity d' from (4.11).

$$d' = d - \frac{S(\mathbf{p}, d + 1) - S(\mathbf{p}, d - 1)}{2(S(\mathbf{p}, d + 1) + S(\mathbf{p}, d - 1) - 2S(\mathbf{p}, d))} \quad (4.11)$$

4.7. Evaluation

Evaluations of the proposed sensor fusion algorithm have been conducted in terms of density, computation time and accuracy. An important factor for the resulting quality of the disparity image is the proportion of projected ToF range measurements relative to the stereo image resolution. Usually, the stereo images on the robot have a much higher resolution e.g. 1388×1038 pixel than the ToF sensor e.g. 176×144 pixel. On the extreme, when no ToF data is available, the algorithm will simply behave like a common stereo algorithm using semi-global optimization. To alter the proportion of available ToF data relative to the available stereo information, the neighborhood size used for wavefront propagation as described by Section 4.3 is varied. In order to determine a best fitting value, the performance of the proposed algorithm has been evaluated against different values for the neighborhood size. A visual impression of the effects of the neighborhood size on the resulting disparity image is given by Figure 4.5.

The disparity map created without ToF information is shown in Figure 4.5(b). It clearly exhibits the typical drawback of stereo vision when facing unstructured areas. The white, texture-less walls and the wooden table have not been assigned any valid disparity value. The propagation and usage of each pixel's ToF measurement to its 3×3 neighborhood results in a visible improvement of the disparity density (Figure 4.5(d)). Now, the number

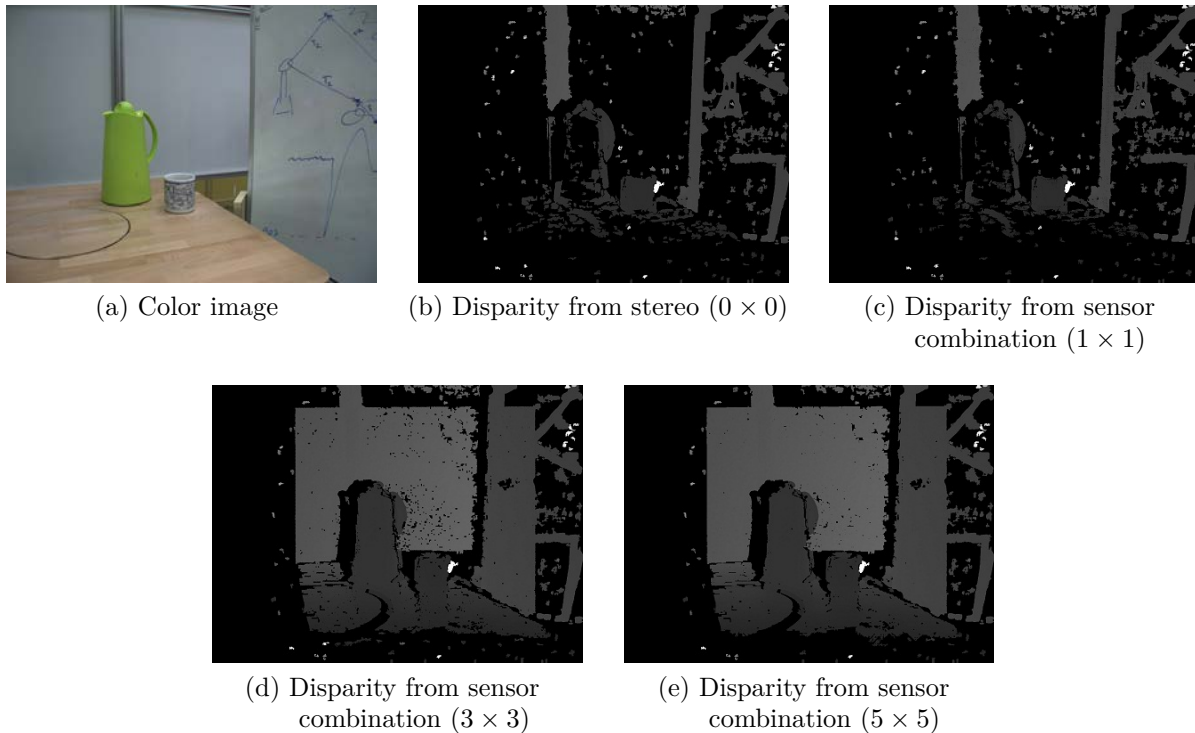


Figure 4.5.: Disparity images of stereo vision and the sensor fusion algorithm.

of projected and propagated ToF data is almost equal to the number of pixel that rely on stereo based disparity guesses only. Within the optimization procedure of the cost function, the ToF values are now able to create unique disparity responses for most of the unstructured areas.

The disparity map becomes even denser when larger propagation ranges are selected as seen in Figure 4.5(e). However, increasing the neighborhood range further also overwrites effectively the disparity information from stereo vision, which is more accurate for structured areas. It even creates blocky effects due to the propagation of the unmodified original ToF values to its neighboring pixel.

The relative amount of pixel holding valid disparity information in proportion to the selected neighborhood size for wavefront propagation is given by Table 4.1 for the scene shown by Figure 4.5. The density values have been computed for the complete image and not only for the overlapping viewing space of all three cameras. A neighborhood range of

Neighborhood range	0×0	1×1	3×3	5×5
Disparity density	17.46%	16.37%	50.61%	51.77%

Table 4.1.: Disparity densities in relation to different wavefront propagation ranges for the ToF measurements.

0×0 denotes that no ToF information has been used to compute the disparity image. As outlined above, an increasing neighborhood size results in a denser disparity image, while

simultaneously the accuracy of the range values decreases. For the following evaluations a 5×5 neighborhood range proved to be the propagation size which results in the best score.

The proposed algorithm has been evaluated on an Intel[®] Core[™] i7-2860QM with 2.5 GHz. In order to accelerate the computations, the implementation has been optimized using SIMD support to execute basic operations in parallel. With color images of a resolution of 450×375 pixel and by evaluating the cost function for 112 different disparities, the algorithm is able to run on average with 10 Hz. In order to further decrease computation time, the number of disparities may be limited to a smaller range depending on the region of interest on the target scene.

The accuracy of the proposed algorithm has been evaluated on a real world dataset as well as the standard Middlebury stereo dataset [SS12]. In order to evaluate the accuracy and density of the sensor fusion algorithm on a real world scene, a planar surface has been positioned in front of the camera system at a distance of 1.7 m. The parametric equation of the planar surface is estimated using the robust estimator RanSaC. Then, the average distance and standard deviation of the computed range values to the plane together with their density is computed for the Microsoft[®] Kinect[™], the stereo, and the sensor fusion sensor system. Figure 4.6 shows the dataset of the plane recordings for each camera system.



Figure 4.6.: Images of a planar surface: the original color image (first column), disparity from passive stereo vision (second column), disparity from the Microsoft[®] Kinect[™] sensor (third column), and disparity from the proposed sensor combination algorithm (forth column).

The evaluation results of the plane dataset are given by Table 4.2. Among all camera systems, passive stereo vision delivers the worst density with 65.5%. This is intuitively clear when inspecting the color image from Figure 4.6. Only at the textured areas stereo vision is able to compute valid range values. However, compared to the active range sensor that yields a density of 99.3%, these values are on average significantly more accurate. Sensor fusion constitutes a tradeoff between the superior accuracy of stereo vision and the superior

density of the KinectTM sensor. It achieves a point cloud density on the plane area of 93.1% while keeping an average accuracy of 0.017 m.

Method	μ	σ	Density
Stereo	0.015 m	0.027 m	65.5%
Microsoft [®] Kinect TM	0.025 m	0.017 m	99.3%
Sensor Fusion	0.017 m	0.012 m	93.1%

Table 4.2.: Average disparity error and density on the plane dataset.

Figure 4.7 shows the application of the sensor fusion algorithm on different real world scenes that have been recorded with the camera setup presented in Section 3.4. The resulting disparity images have been computed with and without using the available ToF data.



Figure 4.7.: The first column shows the color images, the second column disparity images from stereo vision and the third column disparity from the proposed sensor fusion algorithm.

The standard Middlebury stereo dataset provides stereo image pairs and ground truth disparity images, however, it does not provide 2.5D ToF data. In order to apply the sensor

fusion algorithm on the dataset, the ToF data has been simulated using the ground truth disparity data according to (4.12).

$$d = \hat{d} + \mathcal{N}(0, 1) \quad (4.12)$$

Let \hat{d} be the disparity given by the ground truth data, then the disparity corresponding to the ToF data is simulated by applying additive white Gaussian noise with a variance of 1 corresponding roughly to the measurement noise induced by a ToF sensor. Additionally, 1% of all disparity values are replaced by a random value using a uniform distribution over all possible disparities to simulate false range measurements.

Figure 4.8 shows the test images from the Middlebury stereo dataset and their corresponding disparity images using the proposed sensor fusion algorithm without using ToF measurements, in the following termed stereo, and the proposed method combining ToF and stereo data for disparity computation.

The accuracy of the disparity image is measured by comparing it with the ground truth disparity image and by computing the average disparity error. The results of the evaluation are given by Table 4.3. It compares the accuracy resulting from the stereo algorithm against the accuracy resulting from the proposed sensor fusion method.

Method	Venus	Teddy	Cones
Stereo	0.38	3.3	2.94
Sensor Fusion	0.16	0.98	0.9

Table 4.3.: Average disparity error on the Middlebury stereo dataset.

The evaluation results show that the integration of the ToF data into the semi-global optimization process increases the accuracy of the disparity image. In case of the Venus dataset, the average error is reduced by 62% compared to the stereo vision algorithm. For the Teddy and Cones dataset, the average error has even been reduced by over 70%. When comparing the disparity images from Figure 4.8, it is visible, that especially in areas with sharp disparity jumps, e.g. the outline of the cones, sensor fusion is able to preserve the edges where stereo vision fails.

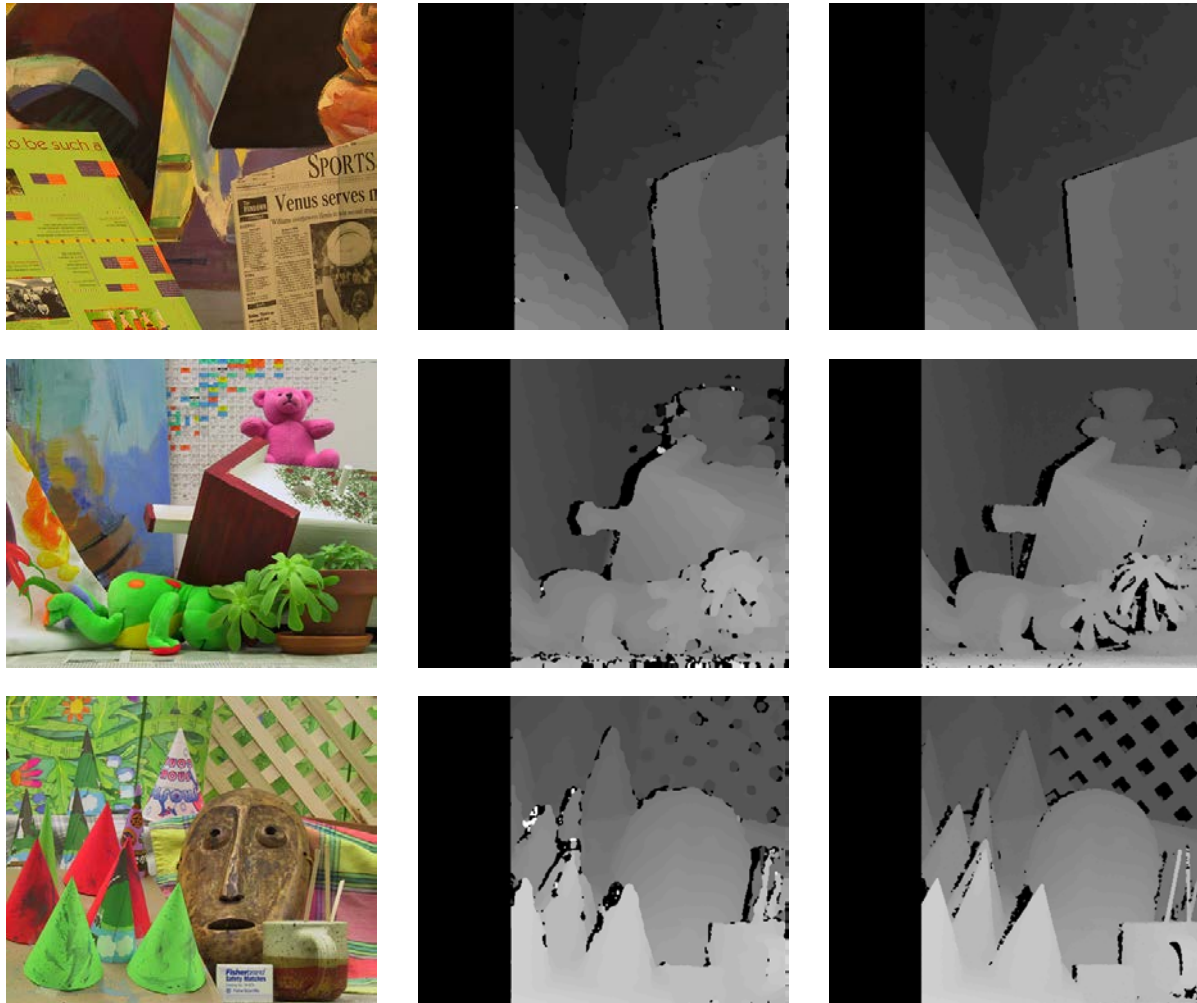


Figure 4.8.: The Middlebury stereo dataset: The first column shows the original color images, the second column disparity images from stereo vision and the third column disparity from the proposed sensor combination algorithm. Beginning with the first row, the datasets are labeled with the names *Venus*, *Teddy*, and *Cones*.

5. Object Modeling

Object modeling creates a-priori knowledge from an object's appearance and shape, which is encoded by features in an abstract object representation. Within this context, *appearance* describes 2D features like texture or edges from the color image, whereas *shape* relates to an object's 3D information originating from its surface. The robust identification and representation of these 2D and 3D features are central to the success of object recognition. In general, a distinction is made between local point features for textured object modeling and global features for texture-less object modeling.

A major advantage of point features is their locality that enables the descriptor to accurately describe an object even under partial occlusion. Point features are fixed to a distinct 2D or 3D position on the object and therefore rely on 2D texture or 3D structure in order to enable their stable localization. By matching the distinct 2D and 3D feature positions across different viewpoints, object modeling combines the local features into a single 3D object model to capture their spatial alignment. While point features achieve impressive recognition results for a large number of objects, texture-less objects are often not suited for point feature based recognition. This is due to their lack of distinct 2D texture, which prevents a stable localization of local feature points.

Here, global features are used to describe the entire object by a single descriptor. Instead of being fixed to a specific position, global features are fixed to a specific object appearance i.e. a specific pose of the object. A common approach to compute global descriptors is to define classes of local descriptors and to capture their frequency of occurrence for the current object appearance within a histogram. The global descriptors are not combined into a single 3D object model, but rather saved in pairs of a global descriptor and its underlying object pose. Compared to local descriptors, the main disadvantage of this approach is its diminished robustness against occlusion as it directly influences the shape of the global descriptor.

Figure 5.1 gives an overview of the complete procedure which will be presented in the following sections. This thesis proposes a simple and intuitive data acquisition method by manually placing an object into the robot's gripper. The robot autonomously moves the object while recording object images from varying viewpoints. In this way, a sequence of object images is recorded which are segmented from the background based on the known pose of the robot's gripper. When using the vision system outside a robotic platform, the

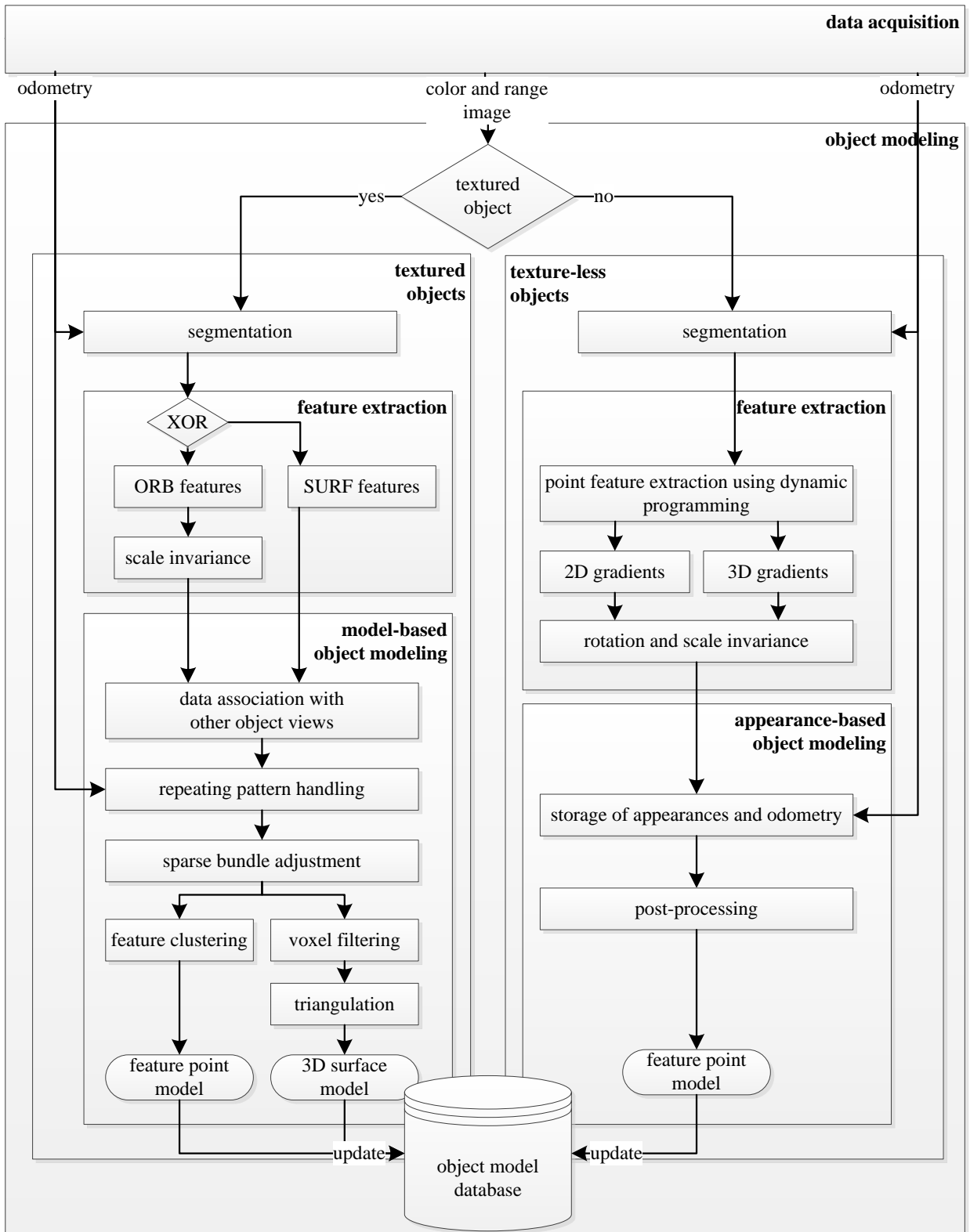


Figure 5.1.: Schematic overview of the individual processing steps for object modeling.

system design enables data acquisition either by using a turn table or by manually moving the camera around an object, while inferring the pose of the object using image tags.

Given a textured object, local point features are computed either based on SURF features or a scale invariant extension of the ORB feature point descriptor. Object modeling is *model-based*. It associates the feature points from all object views with each other and combines the information with the data from the odometry to register the object images into a single 3D model using sparse bundle adjustment. The feature points appearing across several object views are clustered using mean shift and represented with a common descriptor within the feature point model. In order to compute a 3D surface model, voxel filtering removes noisy 3D data points before a Delaunay triangulation algorithm computes a closed surface out of the individual 3D points. The surface model is not necessary for recognition, but it is vital for grasping in order to infer suitable grasp configurations. Figure 5.2 shows the creation of a 3D object model from an exemplary object.

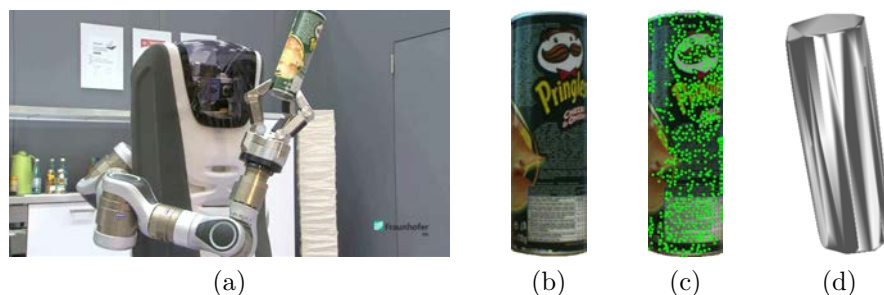


Figure 5.2.: (a) Service robot recording object images from different viewpoints. (b) One of the single object views recorded by the RGB-D camera for object model creation. (c) Extracted feature points. (d) 3D object model with extracted surfaces using sparse bundle adjustment and Delaunay triangulation.

In the presence of texture-less objects, global features representing the distribution of 2D and 3D gradients and exhibiting rotation and scale invariance are computed. Here, the proposed procedure is *appearance-based* and solely associates the global descriptors with the pose originating from the gripper’s odometry or the recognition of the image tag. The feature point model is build up with these individual image-odometry pairs. The combination of the individual images into a single model is based solely on the odometry information, as the necessary point-to-point associations used for bundle adjustment are not given when using global features.

This thesis proposes two distinct modeling concepts depending on an object’s texture. The first approach is model-based and presents an extension for scale-invariance of the binary ORB point feature descriptor [RRKB11] for textured object modeling (Section 5.2). The second approach is appearance-based and targets the modeling of texture-less objects by introducing a fast to compute global feature descriptor using dynamic programming

(Section 5.3). The presented results describing the modeling and recognition of textured objects have already been published in [FABV12]. The proposed approach for texture-less object modeling has been published in [FBAV13].

5.1. Data acquisition

Data acquisition captures object images from different perspectives. The main focus of this thesis lies on the integration of the data acquisition procedure on the robot system as shown by Figure 5.3(a). Additionally, two alternative approaches are proposed to meet requirement R1 from Section 1.4, which demands the application of the proposed system also on a stand-alone vision system.



Figure 5.3.: Data acquisition with different hardware setups.

Data acquisition by the use of the robot system requires the object to be placed within the robot's gripper. While looking with the camera system towards its gripper, the robot autonomously rotates and translates its manipulator between the image recordings to generate different viewpoints. Due to the internal calibration of the robot's manipulator to its camera system, the pose of the object relative to the camera system is approximately given by the pose of the gripper's internal coordinate system. This information is stored as odometry information together with each recorded image. Figure 5.4 illustrates the gripper's coordinate system and its transformations relative to the camera coordinate system.

In the present setup for robot based data acquisition, the camera is at a fixed position and the object is moved in front of it. Therefore, the camera coordinate systems $\mathbf{O}_i, i \in \{1, 2, 3\}$ should be at the same position and the position of the object should move. However, Figure 5.4 illustrates the reversed situation i.e. the camera coordinate systems move and the object is static. This does not change the mathematical considerations, but it transfers the problem

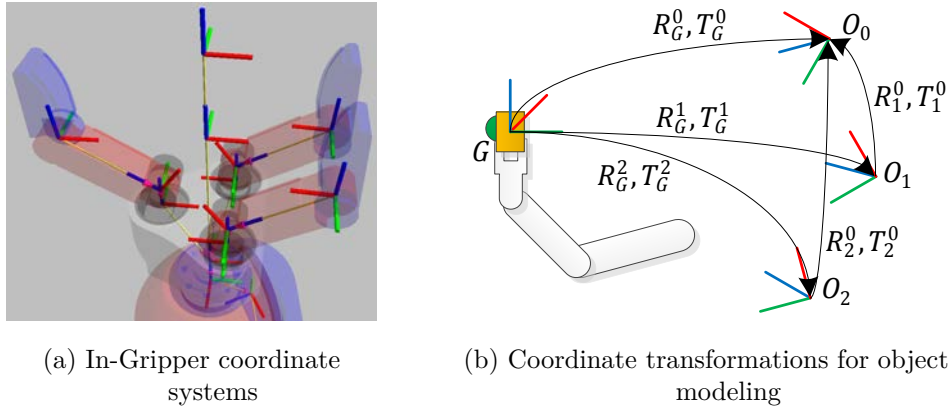


Figure 5.4.: Inferring odometry data from the in-gripper coordinate systems for different object views. The transformations $\mathbf{R}_G^i, i \in \{1, 2, 3\}$ are given by the calibration of the robotic manipulator to the camera system. Therefore, the coordinates of the recorded 3D points $\mathbf{P}_G = (x, y, z)$ within the in-gripper coordinate system \mathbf{G} are transferred into the common coordinate system \mathbf{O}_0 by $\mathbf{P}_{O_0} = \mathbf{M}_i^0 \mathbf{M}_G^i \mathbf{P}_G$, where \mathbf{M}_a^b is a standard 4×4 transformation matrix consisting of the 3×3 rotation \mathbf{R}_a^b and the 3×1 translation vector \mathbf{T}_a^b .

of object modeling into a standard environment modeling problem, where the camera system is moved to record a static environment. The presented approach has been implemented on the mobile service robot Care-O-bot[®] 3 using the ROS software environment.

An alternative approach is to place the object on a turn table to record the object images as illustrated by Figure 5.3(b). The turn table rotates in between the image recordings by the camera system which is placed in front of it. In order to infer the odometry data, that describes the motion of the object in between the image recording, the camera is calibrated relative to the turn table. The illustrated setup has been developed within the scope of this thesis and it has been presented at several trade fairs for demonstration.

A much simpler approach constitutes the usage of a known pattern like a chessboard to infer the pose of the object relative to the camera system for each recording. This approach is illustrated by Figure 5.3(c). Compared to the usage of the robot system or the turn table, the camera is now moving around the object and not vice versa. Thereby, the object is placed at a known position relative to the pattern and the user manually moves the camera around the object.

All three methods result in a set of color and range images with corresponding odometry information representing the pose of the camera system relative to the object. This information provides the basis in order to aggregate the image data into a common object model for textured and texture-less object recognition.

5.2. Modeling of textured objects

The texture of an object, like imprints, letters or numbers, enables the usage of local point features which are affixed to distinct 2D shapes like corners or circular structures. By the means of feature descriptors, the local neighborhood of the feature points is characterized by an n -dimensional vector e.g. by capturing the distribution over 2D image gradients. Due to their distinct description and their precise localization on the 2D image of the object, feature points serve as reference points for the registration of the individual object images into a common 3D model.

The proposed method for object modeling makes use of local point features. The system is evaluated using the well known SURF feature descriptor as well as a newly developed scale invariant extension of the local point feature ORB, termed sORB. As described in Section 2.2, SURF is based on the well known SIFT descriptor. It replaces several time consuming computations with fast to compute approximations while keeping the recognition performance on a similar magnitude like SIFT. ORB is a binary descriptor that compares intensity values of individual pixel with each other. It is an extension of BRIEF that adds rotation invariance to the feature point descriptors. However, a major drawback of ORB is its missing scale invariance. In the following, the thesis elaborates the functionality of BRIEF and ORB in order to present a scale invariant extension of the descriptor even when not having depth data available. Thereby, the SURF feature descriptor serves as a benchmark against which the proposed scale invariant ORB descriptor is evaluated.

5.2.1. Local feature descriptor

BRIEF feature points consist of a binary descriptor, where each bit is calculated according to (5.1).

$$\tau(\mathbf{p}; \mathbf{p}'_j, \mathbf{p}''_j) := \begin{cases} 1, & \text{if } i(\mathbf{p}'_j) < i(\mathbf{p}''_j) \\ 0, & \text{otherwise} \end{cases} \quad (5.1)$$

Here, $i(\mathbf{p}'_j), j \in \mathbb{N}$ constitutes the sum of intensity values over a fixed sized $n \times n, n \in \mathbb{N}$ kernel at the 2D image position \mathbf{p}'_j expressed relative to the 2D reference position $\mathbf{p} = (u, v)$. $\tau(\mathbf{p}; \mathbf{p}'_j, \mathbf{p}''_j)$ simply compares the sum of intensity values around \mathbf{p}'_j with the sum of intensity values around \mathbf{p}''_j and encodes the results with a binary value. Figure 5.5 illustrates the descriptor computations.

The overall descriptor is composed by comparing different pixel positions with each other and writing the binary results into the descriptor f_k according to (5.2), where $k \in \mathbb{N}$ relates

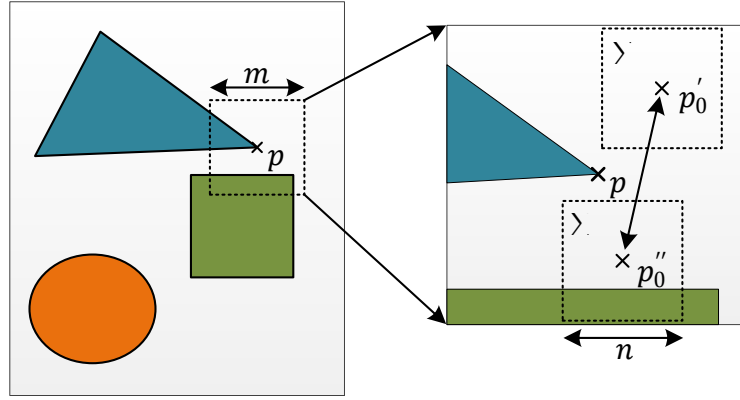


Figure 5.5.: Computing a single binary BRIEF descriptor entry by comparing the sum of intensity values from the local regions around \mathbf{p}'_0 and \mathbf{p}''_0 within the patch centered at \mathbf{p} . The patch size of the affected image region is $m \times m$ pixel and the local regions are computed using a kernel size of $n \times n$ with $n < m$ and $n, m \in \mathbb{Z}$. In the standard implementation $m = 48$ and $n = 9$. The overall descriptor is computed by repeating the comparison for different pairs of \mathbf{p}'_j and \mathbf{p}''_j .

to the size of the descriptor in bytes.

$$f_k := \sum_{1 \leq j \leq 8k} 2^{j-1} \tau(\mathbf{p}; \mathbf{p}'_j, \mathbf{p}''_j) \quad (5.2)$$

A major advantage of BRIEF is its straightforward descriptor calculation that enables its fast computation and matching. The computation of the binary descriptor is based on a simple pixel comparison and thresholding function. Feature point matching may be performed efficiently using local sensitivity hashing.

ORB extends the BRIEF descriptor to be invariant against rotations around the optical axis of the camera system. By computing the intensity centroid of the image patch according to (5.3),

$$m_{jk} = \sum_{u,v} u^j v^k i(u, v) \quad (5.3)$$

$$\mathbf{p}_i = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

the orientation of the patch is simply given by the angle of the vector from the patch center \mathbf{p} to the intensity centroid \mathbf{p}_i (5.4).

$$\theta = \text{atan2}(m_{01}, m_{10}) \quad (5.4)$$

Orientation invariance is achieved by rotating the coordinates of pixel comparisons according to the determined orientation of the patch. In order to improve the orientation accuracy,

the image moments from (5.3) are computed within a circular region around \mathbf{p} with radius $r = \frac{m}{2}, r \in \mathbb{R}$ instead of directly using the rectangular region of the patch. This ensures that the covered region for moment computation does not change when rotating the image patch.

Even when using ORB, the descriptor does not exhibit scale invariance. Therefore, the thesis proposes to use the STAR feature point detector [AKB⁺08] for scale estimation in combination with the ORB feature point descriptor. The STAR detector computes the intensity difference between a star-shaped inner region and its star-shaped outer boundary around the current image point \mathbf{p} . In Figure 5.6 e.g., the intensity values of the two areas A_1 and A_2 at the feature scale $m > 0, m \in \mathbb{Z}$ are compared against each other by computing $|i(A_1) - i(A_2)|$.

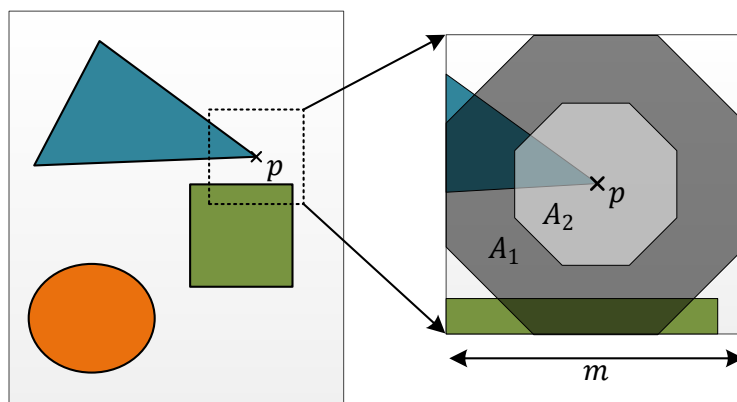


Figure 5.6.: Computing a STAR feature point by comparing the sum of intensity values from the areas denoted by A_1 and A_2 .

In order to efficiently process the star-shaped regions, each of them is split up into a squared and a slanted region whose intensity sum is computed using integral images. A feature point is asserted, when the difference of intensity values exceeds a given threshold. The procedure is repeated for different sizes of the star-shaped regions in order to discover feature points at different scales. Thus, scale is natively given by the size of the regions at which the feature point has been detected. This information can directly be applied to alter the patch size of the ORB descriptor to achieve scale invariance. The scale invariant ORB feature point is termed sORB.

Given the scale $s > 0, s \in \mathbb{R}$ from the STAR feature detector, the native ORB orientation assignment is dynamically adapted by modifying the radius $r = \frac{m}{2}$ of its circular patch. According to the scale s , the patch size m is adapted to $m = 48 \frac{s}{\hat{s}}$, where $\hat{s} \in \mathbb{Z}$ denotes the reference scale at which the patch size is fixed to 48 pixel, the standard value for the BRIEF descriptor (Figure 5.5). Additionally, the kernel size n of the descriptor is adapted to $n = 9 \frac{s}{\hat{s}}$, where the kernel size is set to 9×9 pixel at the reference scale \hat{s} . This adjusts the area of the intensity comparisons to the modified patch size. Following (5.2) the

Table 5.1.: Association of BRIEF patch sizes and kernel sizes in relation to the detected STAR scale of the feature point.

	STAR scale s	9	13	17	23	25	33
\hat{s}							
9	Patch size m	48	69	90	122	133	176
	Kernel size n	9	13	17	25	27	35
13	Patch size m	33	48	62	84	92	121
	Kernel size n	7	9	11	17	17	23
17	Patch size m	25	36	48	64	70	93
	Kernel size n	5	7	9	13	13	17
25	Patch size m	17	24	32	44	48	63
	Kernel size n	3	5	7	9	9	11
33	Patch size m	13	18	24	33	36	48
	Kernel size n	3	3	5	7	7	9

resized coordinates $\frac{s}{\hat{s}}\mathbf{p}'_j$ and $\frac{s}{\hat{s}}\mathbf{p}''_j$ for the intensity comparison, the scale invariant descriptor computation is then given by (5.5).

$$f_k := \sum_{1 \leq j \leq 8k} 2^{j-1} \tau(\mathbf{p}; \frac{s}{\hat{s}}\mathbf{p}'_j, \frac{s}{\hat{s}}\mathbf{p}''_j) \quad (5.5)$$

An evaluation of the proposed scale invariant BRIEF descriptor is given in Section 5.2.5. Crucial for the performance of the sORB descriptor is the value for the reference scale \hat{s} . Evaluations have been conducted for different values of \hat{s} based on the set of test images used within the evaluation section. Table 5.1 shows the determined BRIEF patch sizes and kernel sizes in accordance to the detected STAR feature point scales for varying \hat{s} . The first column specifies the value of \hat{s} . The first row gives the value of the detected STAR scale s . The following rows give the adapted kernel and patch sizes in correspondence to the changing STAR scale for different \hat{s} .

The proposed procedure could also be applied in accordance to the existing distance value of the feature point from the RGB-D camera device. However, it has been decided to use only the STAR scale in order to keep the possibility of applying the proposed procedure even when not having 2.5D data available.

5.2.2. Data association

Data association tracks the appearance of a feature point across different object images based on its descriptor value. This is important for object modeling in order to assemble the single object views to a 3D object model. The matching of a query descriptor with a set of known descriptors is established using computationally efficient techniques like k-d trees or LSH as outlined in Section 2.2. For the proposed feature descriptors, one global k-d tree

in case of the SURF features and one global LSH hash table in case of the sORB descriptor is used to store all feature descriptors extracted from all object views over the set of all known objects. Alternatively, it would be possible to maintain a single k-d tree or hash table per object. However, this would increase the computation time as all data structures would have to be parsed sequentially instead of elaborating the logarithmic performance e.g. of a single k-d tree when searching over all objects.

Usually, the visual features of typical household objects occur more than once on an object's surface e.g. the features of a manufacturer's label, which is printed on several sides as outlined by Figure 5.7. This poses a problem to data association as it is most likely,



Figure 5.7.: Multi-occurrences of feature point p on a single object. A single feature point is associated to two different locations on the object image.

that feature points occurring on different positions on the object are associated with each other and therefore might not be uniquely identified. To solve this problem, a radius based nearest neighbor search strategy is proposed. Instead of associating just a single descriptor to a query point, all descriptors within a maximal distance to the query point are marked as potential matching candidates. The final decision on which feature corresponds best is postponed until the measurement update, where the odometry data from the motion of the object e.g. by the robot's gripper is evaluated. The measurement update considers all correspondence candidates and selects the corresponding feature point with the largest measurement probability i.e. the feature point matching that best resembles the motion of the robot's gripper that caused the feature point displacement. All other candidates are rejected.

5.2.3. Model-based object modeling

3D object model reconstruction from the individual camera images is formulated as a non-linear least-squares optimization problem. It is based on the observations of feature point correspondences across the different viewpoints and the odometry data describing the camera's movement. Feature points that have already been assigned to the 3D object model are termed *object-centered*. Their appearances on the individual camera images are termed

observations. Using this terminology, the solution of the optimization problem amounts to minimizing the discrepancy between the individual observations of the feature points and the coordinates of their object-centered counterparts.

The solution of the optimization problem is computed using *sparse bundle adjustment* (SBA) by jointly refining the object-centered feature coordinates and the poses of the different viewpoints. SBA formulates the reconstruction problem in compliance to the *Levenberg-Marquardt algorithm* (LMA), which is used for solving the non-linear least-squares problem by local linearization and iteratively moving towards the (locally) optimal solution. As for all iterative minimization algorithms, it is vital to provide the LMA with a good initial guess for the solution, which must be close to the actual solution. This is achieved by using the odometry information, e.g. from the motion of the gripper as explained in Section 5.1, in order to establish an initial transformation between the different object images.

The *Levenberg-Marquardt algorithm* (LMA) is an iterative method, that interpolates between gradient descent and the Gauss-Newton method for solving non-linear least squares problems. It locates the minimum of a sum of squared function values by modifying its behavior depending on the rate of convergences towards the locally optimal solution. When convergence is fast, the algorithm behaves like the Gauss-Newton method. However, when convergence slows down the behavior of the LMA adapts to the gradient descent algorithm. Sparse bundle adjustment applies the LMA to solve the reconstruction problem of object modeling as explained in the following.

Let n be the number of 3D feature points from the object surface that are visible within m different camera viewpoints. The aim of object modeling is to reconstruct the optimal camera movement and the optimal 3D feature point position that best explain the observations. The optimization is performed with respect to the parameter set $\mathbf{p} = \{\mathbf{a}_1, \dots, \mathbf{a}_m, \mathbf{b}_1, \dots, \mathbf{b}_n\}$, where $\mathbf{a}_j \in \mathbb{R}^6$ denotes the parameters of the pose of camera j with respect to the world coordinate system and $\mathbf{b}_i \in \mathbb{R}^3$ denotes the parameters of the 3D position for feature point i . The nm observations of the n 3D feature points are stored within an observation vector $\mathbf{o} = \{\mathbf{x}_{11}, \dots, \mathbf{x}_{1m}, \mathbf{x}_{21}, \dots, \mathbf{x}_{2m}, \dots, \mathbf{x}_{n1}, \dots, \mathbf{x}_{nm}\}^\top$, where $\mathbf{x}_{ij} \in \mathbb{R}^3$ denotes the observation of feature point i from the camera image j . With the non-linear function $f: \mathbb{R}^9 \rightarrow \mathbb{R}^3, (\mathbf{a}_j, \mathbf{b}_i) \mapsto f(\mathbf{a}_j, \mathbf{b}_i) = \mathbf{R}_j \mathbf{b}_i + \mathbf{T}_j$, that maps the reconstructed 3D feature point position \mathbf{b}_i into the camera coordinate system j using the rotation matrix \mathbf{R}_j and the translation vector \mathbf{T}_j encoded by $\mathbf{a}_j = (x, y, z, \alpha, \beta, \gamma)^\top$, the minimization problem is defined by (5.6).

$$\min_{\mathbf{p}} \sum_{i=1}^n \sum_{j=1}^m r_{ij}^2, \quad r_{ij} = \|(f(\mathbf{a}_j, \mathbf{b}_i) - \mathbf{x}_{ij})\| \quad (5.6)$$

It is desired to find the parameter vector \mathbf{p} that minimizes the residuals r_{ij} given by the squared distance between the observation \mathbf{x}_{ij} and the reconstructed 3D point \mathbf{b}_i , which

has been mapped to the same coordinate system as \mathbf{x}_{ij} . Let $S(\mathbf{p}) = \sum_{i=1}^n \sum_{j=1}^m r_{ij}^2$, then the minimum of (5.6) is given when $\frac{\partial S}{\partial \mathbf{p}} = 0$, yielding to (5.7).

$$\frac{\partial S}{\partial \mathbf{a}_j} = 2 \sum_{i=1}^n \sum_{j=1}^m r_{ij} \frac{\partial r_{ij}}{\partial \mathbf{a}_j} = 0, \quad \frac{\partial S}{\partial \mathbf{b}_i} = 2 \sum_{i=1}^n \sum_{j=1}^m r_{ij} \frac{\partial r_{ij}}{\partial \mathbf{b}_i} = 0 \quad (5.7)$$

Due to the non-linearity of f , there is no closed-form solution for the partial derivatives from (5.7). Instead, a good initial parameter estimate \mathbf{p}^0 must be provided to the LM-algorithm, which refines the parameter vector \mathbf{p} iteratively by $\mathbf{p} \approx \mathbf{p}^{k+1} = \mathbf{p}^k + \delta$. $\delta = (\delta \mathbf{a}_1, \dots, \delta \mathbf{a}_m, \delta \mathbf{b}_1, \dots, \delta \mathbf{b}_n)$ is a shift vector that steers the parameter vector \mathbf{p} towards a local minimum of (5.6) and k denotes the level of iteration. f is linearized around $\mathbf{p}_{ij}^k = (\mathbf{a}_j^k, \mathbf{b}_i^k)$ using a first-order Taylor expansion about \mathbf{p}_{ij}^k according to (5.8).

$$\begin{aligned} f(\mathbf{a}_j, \mathbf{b}_i) &\approx f(\mathbf{a}_j^k, \mathbf{b}_i^k) + \sum_{j'=1}^m \frac{\partial f(\mathbf{a}_j^k, \mathbf{b}_i^k)}{\partial \mathbf{a}_{j'}} (\mathbf{a}_j - \mathbf{a}_j^k) + \sum_{i'=1}^n \frac{\partial f(\mathbf{a}_j^k, \mathbf{b}_i^k)}{\partial \mathbf{b}_{i'}} (\mathbf{b}_i - \mathbf{b}_i^k) \\ &= f(\mathbf{a}_j^k, \mathbf{b}_i^k) + \sum_{j'=1}^m J_{kj'} \delta \mathbf{a}_j + \sum_{i'=1}^n J_{k(m+i')} \delta \mathbf{b}_i \end{aligned} \quad (5.8)$$

Here, \mathbf{J} denotes the $(nm) \times (n+m)$ Jacobian matrix of $\mathbf{F} = (f(\mathbf{a}_1, \mathbf{b}_1), \dots, f(\mathbf{a}_1, \mathbf{b}_n), f(\mathbf{a}_2, \mathbf{b}_1), \dots, f(\mathbf{a}_2, \mathbf{b}_n), \dots, f(\mathbf{a}_m, \mathbf{b}_1), \dots, f(\mathbf{a}_m, \mathbf{b}_n))^\top$ with respect to \mathbf{p} and $J_{kj'} \delta \mathbf{a}_j = 0, \forall j' \neq j, J_{k(m+i')} \delta \mathbf{b}_i = 0, \forall i' \neq i$ with $k = nj + i$. Therefore, (5.6) may be approximated using matrix notation by (5.9)

$$\min_{\mathbf{p}} \sum_{i=1}^n \sum_{j=1}^m \|f(\mathbf{a}_j^k, \mathbf{b}_i^k) + \mathbf{J}_k \delta - \mathbf{x}_{ij}\|^2 = \min_{\mathbf{p}} \|\mathbf{F}^k + \mathbf{J} \delta - \mathbf{o}\|^2 \quad (5.9)$$

Now, it is possible to solve (5.9) by setting its derivative with respect to δ to 0, resulting in the *normal equations* given by (5.10).

$$\frac{\partial S(\mathbf{p}^k + \delta)}{\partial \delta} = 2(\mathbf{F}^k + \mathbf{J} \delta - \mathbf{o})^\top \mathbf{J} = 0 \Leftrightarrow \mathbf{J}^\top \mathbf{J} \delta = \mathbf{J}^\top (\mathbf{o} - \mathbf{F}^k) \quad (5.10)$$

The LM-algorithm introduces a damping factor $\lambda \in \mathbb{R}$ to the normal equations, leading to (5.11).

$$(\mathbf{J}^\top \mathbf{J} + \lambda \text{diag}(\mathbf{J}^\top \mathbf{J})) \delta = \mathbf{J}^\top (\mathbf{o} - \mathbf{F}^k) \quad (5.11)$$

Let $\text{diag}(\mathbf{J}^\top \mathbf{J})$ denote a diagonal matrix whose diagonal elements take the value of the corresponding elements from $\mathbf{J}^\top \mathbf{J}$. A small value of λ has less effect on the adjusted normal equations (5.11) and approximates the behavior of the Gauss-Newton algorithm. Solving the adjusted normal equations for larger values of λ mimics the behavior of the steepest descent algorithm. The damping factor is decreasing for each iteration, as long as a reduction of S is given. As soon as the convergence towards the local minimum slows down, the damping factor is increased to steer the direction of convergence towards the direction of the steepest descent. The optimization algorithm terminates, when S has dropped below a predefined threshold or a maximal number of iterations has been reached.

When dealing with feature point observations, it is inevitable, that similar descriptor values cause a small percentage of the returned feature point associations to be incorrect. The LM optimization algorithm is able to cope with this inaccuracy by incorporating a block diagonal covariance matrix Σ_o for the observation vector \mathbf{o} into the normal equations from (5.11) resulting in the *weighted normal equations* (5.12).

$$(\mathbf{J}^\top \Sigma_o^{-1} \mathbf{J} + \lambda \text{diag}(\mathbf{J}^\top \Sigma_o^{-1} \mathbf{J})) \delta = \mathbf{J}^\top \Sigma_o^{-1} (\mathbf{o} - \mathbf{F}^k) \quad (5.12)$$

Instead of minimizing the squared Euclidean distance r_{ij} , the introduction of the Σ_o leads to a minimization of its squared Mahalanobis distance. The weight for each observation encoded in the covariance matrix is computed based on the distance of the observed and predicted distance of a feature point. The larger the discrepancy, the more likely it is that the measurement originates from a false feature association. Therefore, its influence to the minimization problem should decrease. This relationship is encoded by the monotonic decreasing kernel function $k: \mathbb{R} \rightarrow \mathbb{R}$ from (5.13) that computes the weight of each observation as a function of its discrepancy to its predicted value.

$$k: \mathbb{R} \rightarrow \mathbb{R} \\ r_{ij} \mapsto k(r_{ij}; c) = \frac{\sqrt{|2c^2 \left(\sqrt{1 + \left(\frac{r_{ij}}{c}\right)^2} - 1 \right)|}}{r_{ij}} \quad (5.13)$$

The kernel parameter c is crucial for the shape of k . A larger value of c results in a slower reduction of the weight value from small r_{ij} to larger r_{ij} . For small values of c , small changes in the lower values of r_{ij} lead to larger variations in the corresponding weight values. Figure 5.8 illustrates the explained dependencies of k on different values of c . A detailed evaluation on the effect of varying values for the kernel parameter c on the accuracy of the resulting model is given in Section 5.2.5.

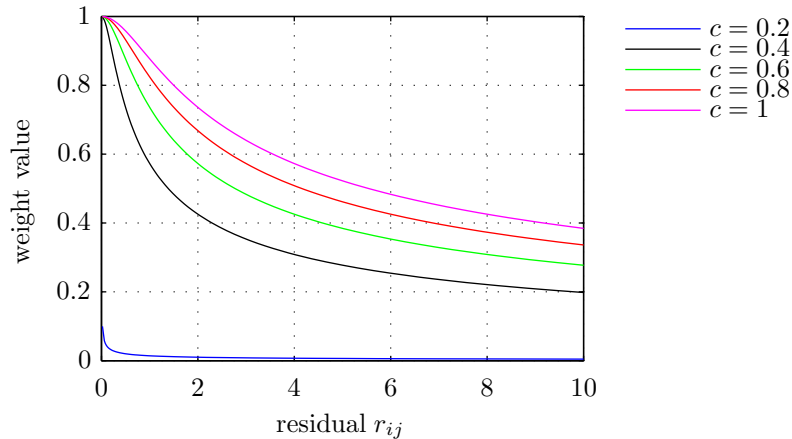


Figure 5.8.: Visualization of the Pseudo-Hubert kernel function k from (5.13) for different values of the kernel parameter c .

The sparseness of sparse bundle adjustment relates to the sparse nature of the Jacobian \mathbf{J} and the diagonal shape of Σ_o . This sparseness is elaborated to efficiently compute the shift vector δ as presented in [LA04].

5.2.4. Post-processing

The sparse 3D feature point model provides the basis for object recognition and localization. Without post-processing, the feature model consists of about 20000 feature points per object. The large number of descriptors mainly originates from their repeated appearance on different object views. In order to reduce the memory requirements and to decrease the computation time for feature point matching, feature point descriptors mapped to the same 3D coordinates are clustered using mean-shift [CM02] with a fixed bandwidth and only the cluster centers are stored within the object model. The individual feature descriptors are discarded.

Mean-shift is a non-parametric algorithm for detecting the modes of an arbitrarily shaped dataset. It follows the strategy of gradient descent methods by iteratively computing a mean-shift vector that moves a data point \mathbf{x} towards the center of density of its current neighborhood. The computation of the mean shift vector \mathbf{m}_x is based on the gradient of a multivariate kernel density estimator, which is applied on the local neighborhood N_x of \mathbf{x} leading to (5.14).

$$\mathbf{m}_x = \sum_{\mathbf{q} \in N_x} \frac{\mathbf{q} k(\|\mathbf{q} - \mathbf{x}\|)}{k(\|\mathbf{q} - \mathbf{x}\|)} - \mathbf{x} \quad (5.14)$$

Again, $k: \mathbb{R} \rightarrow \mathbb{R}$ denotes a kernel function, which has the property that it is symmetric, non-negative, and the integral over its domain amounts to 1. A popular example for k constitutes the Gaussian function with $k(u) = \frac{1}{2\pi} e^{-\frac{u^2}{2}}$. For every iteration of the mean-

shift algorithm, (5.14) is evaluated and the data point \mathbf{x} is shifted by \mathbf{m}_x . The mean-shift vector is a composition of the coordinates from nearby data points in N_x , weighted in proportion to their distance to \mathbf{x} . The smaller the distance from a neighboring data point \mathbf{q} to \mathbf{x} , the more influence has its position on the mean-shift vector. The procedure is repeated until a maximal number of iterations has been reached or the mean-shift vector has converged to a stationary point that constitutes a local mode of the data set.

When clustering feature descriptors, each descriptor constitutes a data point, which is shifted in the direction of its mean-shift vector until the algorithm terminates. After the mean-shift iterations the descriptors converging to the same local mode are aggregated into a common cluster and the descriptor value of the mode is stored as their representative. In order to apply mean-shift on the binary sORB feature descriptors the descriptor values are first transform to floating point values, then the mean-shift algorithm is applied and finally the mean-shift clusters are converted to binary values again by a simple rounding operation to the nearest binary value. This effectively reduces the number of feature points by approximately 75%.

To create a dense 3D object model for grasp calculation e.g. using OpenRAVE [Dia10], the optimized odometry data \mathbf{a}_j is applied not only on the feature points, but on all 3D points from all object views within the calculated bounding box of the object. The 3D data is filtered using a voxel filter and post-processed using a Delaunay Triangulation algorithm for surface reconstruction.

5.2.5. Evaluation

This section investigates the accuracy of the created object model for varying kernel parameters k from (5.13) using a simulation environment to create ground truth information for the true feature point locations, against which the object model is evaluated. In order to evaluate the matching rate of the proposed sORB descriptor, a standard test dataset is used to compare the performance of SIFT, SURF, BRIEF and ORB against the proposed sORB descriptor.

Evaluation of the local feature descriptor

The proposed scale invariant sORB descriptor is benchmarked against the well-known descriptors SIFT, SURF, BRIEF and ORB on the same dataset as proposed by Calonder et al. in [CLSF10]. The test data originates from the Oxford's Visual Geometry Group¹ and consists of original and transformed image pairs from different scenes as illustrated in Figure 5.9 and Figure 5.10. The transformations cover changes in scale, blur, JPEG compression, and illumination. Additionally, two different scene types are given, one showing

¹<http://www.robots.ox.ac.uk/~vgg/research/affine/>

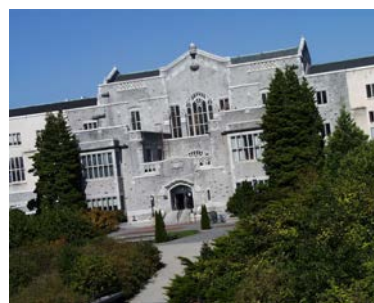
repeated patterns of different forms like a wall of bricks, the other showing homogeneous regions of natural scenes e.g. a harbor scene. For each test set, homographies between the reference image and its transformed images are given in order to provide ground truth matches between feature points originating from the reference image and feature points originating from its transforms.



(a) Boat image (scale and rotation)



(b) Graffiti image (viewpoint)



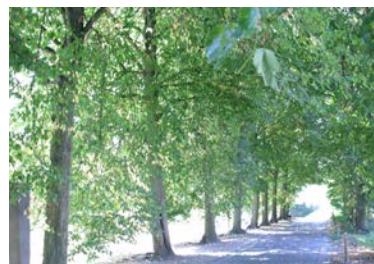
(c) UBC image (JPEG)



(d) Bikes image (blur)



(e) Wall image (viewpoint)



(f) Trees image (blur)



(g) Leuven image (light)



(h) Bark image (scale and rotation)

Figure 5.9.: Dataset from the Oxford's Visual Geometry Group.

The evaluation of the matching rate is jointly considering the performance of the detector and the descriptor. SIFT and SURF have been already proposed together with a feature detector which is used for evaluation. In the case of BRIEF and ORB, the performance is evaluated in conjunction with the STAR detector. Initially, the interest point detector generates a set of keypoints on the reference image and its transforms. As not all image parts are visible on all transformed images, invalid keypoints must be removed using the provided homographies. Then, a pair-wise evaluation of reference and transformed image is conducted by computing the descriptor values and performing a brute force nearest neighbor

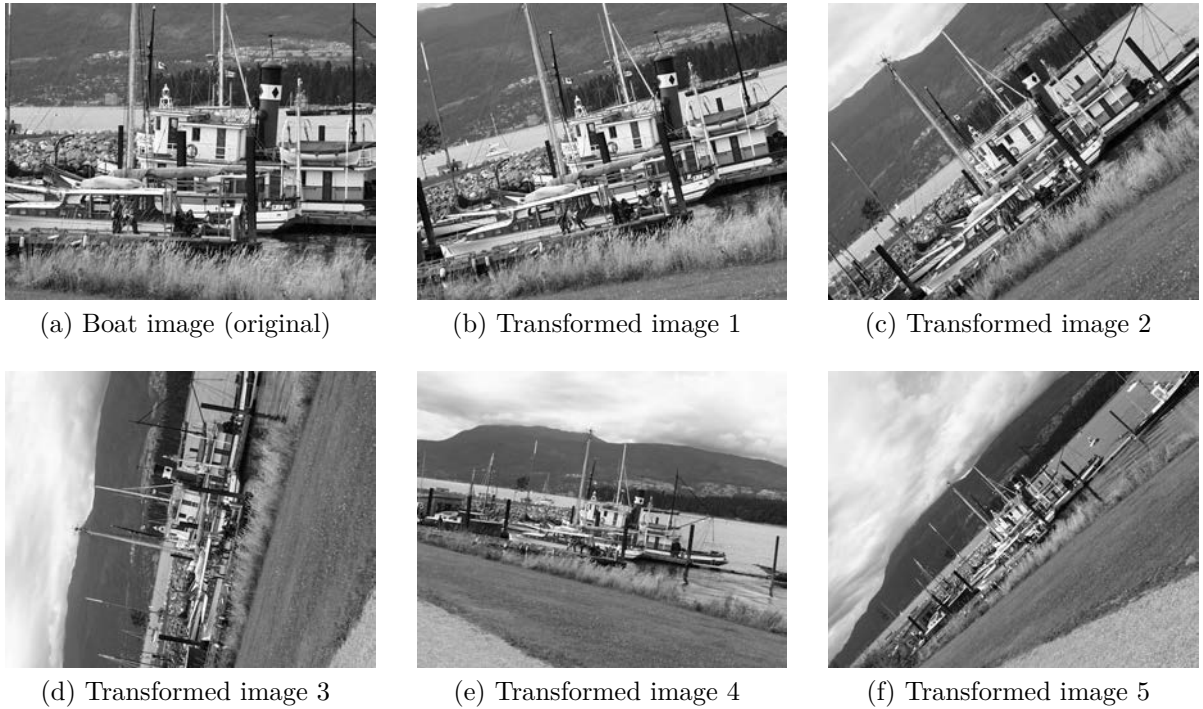


Figure 5.10.: The boat image samples from the Oxford’s Visual Geometry Group dataset. Image transformations represent changes in scale and rotation.

search using the FLANN library [ML09]. A match is considered correct if its corresponding feature point coordinate is corresponding to the ground truth homography. The matching accuracy is calculated for each image pair as the amount of correct matches over the number of all valid keypoints.

Figure 5.11 shows the evaluation results over all image sequences for the SURF, SIFT, ORB, and the proposed sORB descriptor. All binary descriptors use 32 bytes for the representation of the pixel comparisons. On the current dataset, a reference scale $\hat{s} = 17$ from Table 5.1 has given the best recognition rates. Additionally, the evaluations have been performed on a BRIEF variant, termed sBRIEF. Here, the proposed scale invariant extension has been applied to the BRIEF descriptor, which does not exhibit rotation invariance. It is evaluated to illustrate the effect that additional invariance always comes at the cost of a decrease in the recognition of cases where the specific invariance is not needed. This has been pointed out by Calonder et al. in [CLSF10].

Figures 5.11(a) and 5.11(h) show the evaluation results for the Boat and Bark image sequences, which have been specifically designed to exhibit large variances in scale and rotation. In general, sORB outperforms all other descriptors on the Bark image with a matching rate above 75% and outperforms the SIFT and SURF descriptors on the image sequences 1, 2, 3, and 5 from the Boat images. The sBRIEF descriptor clearly fails to reliably match the keypoint descriptors when larger rotations occur e.g. for the Boat dataset 2, 3, and 5. As expected, the ORB descriptor is not able to match keypoints when

large scale changes occur. This is visible for the Bark dataset in image 3 where the scale changed by a factor larger than 2 and the matching rate dropped to a value of 10%.

Variations in viewpoint are addressed by the Graffiti and Wall images from Figure 5.11(b) and 5.11(e). The Wall image sequences do not exhibit larger variations in rotation. Therefore, the sBRIEF descriptor outperforms all other image descriptors, which confirms the fact that the additional unnecessary rotational invariance decreases the matching rate of the other descriptors for this image sequence. The sORB descriptor exhibits approximately the same matching rate like SURF. It slightly performs worse on the Wall images, whereas it slightly outperforms SURF on the Graffiti images. The SIFT descriptor has on average a matching rate that is 10 percentage points higher than SURF or sORB. This is due to its more exact algorithmic implementation, which avoids any time-saving approximations as they are applied to SURF or sORB.

Different levels of blur are addressed by the Bikes and Tree images from Figure 5.11(d) and 5.11(g). The SIFT descriptor proves to be very sensitive to blur, as the distortion complicates the exact measurement of the gradient directions. Therefore, SIFT performs worst on the Bikes and Tree image sequences. sORB and SURF approximate the gradient information over larger areas, therefore the blurry gradients have only minor influence on the deformation of the descriptor values. sORB clearly outperforms the SURF descriptor on the Bikes image and performs approximately on the same level like SURF on the Trees image sequence. The Bikes and Trees image sequence do not exhibit any rotational and scale distortions. Therefore, sBRIEF and ORB benefit from their missing invariance and are able to outperform all other descriptors.

Varying lighting conditions have been captured with the Leuven image sequence from Figure 5.11(f). The sORB descriptor exhibits a matching rate above 77% and performs on average approximately 10 percentage points above SIFT and SURF. This is due to the fact that uniform lighting changes do not change the values of the pixel comparisons and therefore have only a minor influence on the descriptor. Also the pixel comparisons prove to be more stable than the 2D gradient estimations performed for SIFT and SURF. The SURF descriptor performs slightly better than the SIFT descriptor. As for the Bikes and Trees images, the Leuven image sequence does not exhibit any rotational or scale distortions. This enables the ORB and the sBRIEF descriptor to outperform all others.

Different JPEG compression rates do not occur within real world scenes, but are evaluated for the sake of completeness with the UBC image sequence from Figure 5.11(c). The SIFT descriptor proves to be very sensible to the different JPEG compression rates due to their heavy influence on the SIFT gradient estimation. The SURF descriptor performs on average approximately 20 percentage points above the SIFT descriptor as its gradient approximations can handle the image artifacts created by the JPEG compression significantly better than SIFT. sORB still outperforms SIFT and SURF. Its pixel comparisons

for descriptor computation are averaged over a larger image region and therefore the influence of the JPEG compression rate can be kept to minimum. Again, ORB and sBRIEF outperform all others due to the missing rotational and scale distortions within the Leuven image sequence.

In contrast to the Boat and Bark image sequences from Figure 5.11(a) and 5.11(h), Figure 5.13(a) and 5.13(b) evaluate the rotational and scale invariance independently. As the Oxford test set does not provide images with these characteristics, the image sequences for rotation and scale invariance have been created manually (Figure 5.12).

The results for rotation invariance are shown by Figure 5.13(a). The tests have been conducted by taking the Leuven image from Figure 5.9 and by artificially rotating the image around its center in steps of 10° . The x-axis gives the current angle and the y-axis the corresponding matching rate. The evaluations have been conducted in the same way as for the image sequences from the Oxford dataset. It is evident, that the sBRIEF descriptor does not exhibit any rotational invariance as it rapidly drops to a matching rate below 5% for rotations of more than 40° . The SURF descriptor performs best, when facing rotations of multiples of 90° and drops to a value of approximately 60% matching rate in between. This is due to its usage of Haar wavelet filters for orientation assignment which are aligned in discrete angles of 0° and 90° . The SIFT descriptor does not use such approximations and therefore achieves a matching rate constantly above 80%. The proposed sORB descriptor even outperforms SIFT on the test set. Its moment-based orientation assignment does adequately capture the main orientation even under larger rotations. The ORB descriptor benefits on the current dataset from its missing invariance against scale changes resulting in a matching rate that exceeds all others.

Results concerning scale invariance are shown by Figure 5.13(b). The test images have been created by artificially scaling the Leuven image from Figure 5.9 by a factor ranging from 0.5 to 2 in steps of 0.1. The ORB descriptor cannot cope with scale changes. Therefore, its matching rate drops with increasing scale changes down to a value below 5% for a scale factor of 2. SIFT, SURF and sORB exhibit a similar performance for different scales. This is mainly due to the similar performance of the interest point detectors which determine the scale of the current feature point. The sBRIEF descriptor benefits from its missing rotation invariance resulting in a significantly better performance compared to all other descriptors.

The computation time of the descriptors has been evaluated by averaging the time for feature extraction and descriptor computation over 100 runs on the Bark reference image. The results are summarized by Table 5.2. It is distinguished between the computation time, that denotes the time needed for computing all features on the 640×480 test image, and the throughput, that divides the computation time by the number of computed features. It is not surprising that the combination of SIFT detector and descriptor exhibits the longest computation time. The computation of a single feature point and its descriptor takes about

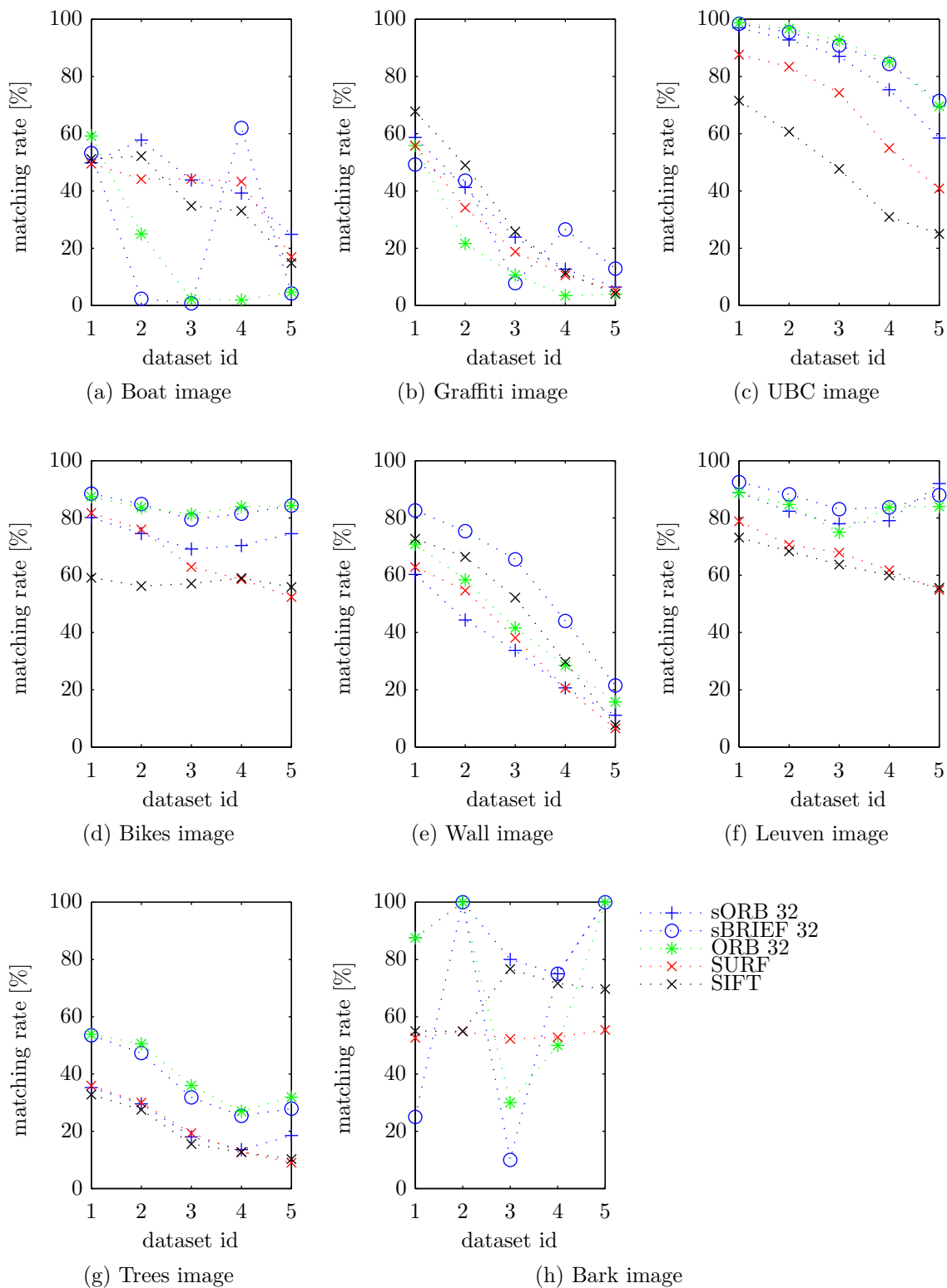


Figure 5.11.: Comparing the matching accuracy of the sORB feature descriptor depending on varying image distortions. The number at the end of the feature names denotes the number of bits that have been used to represent the descriptor.

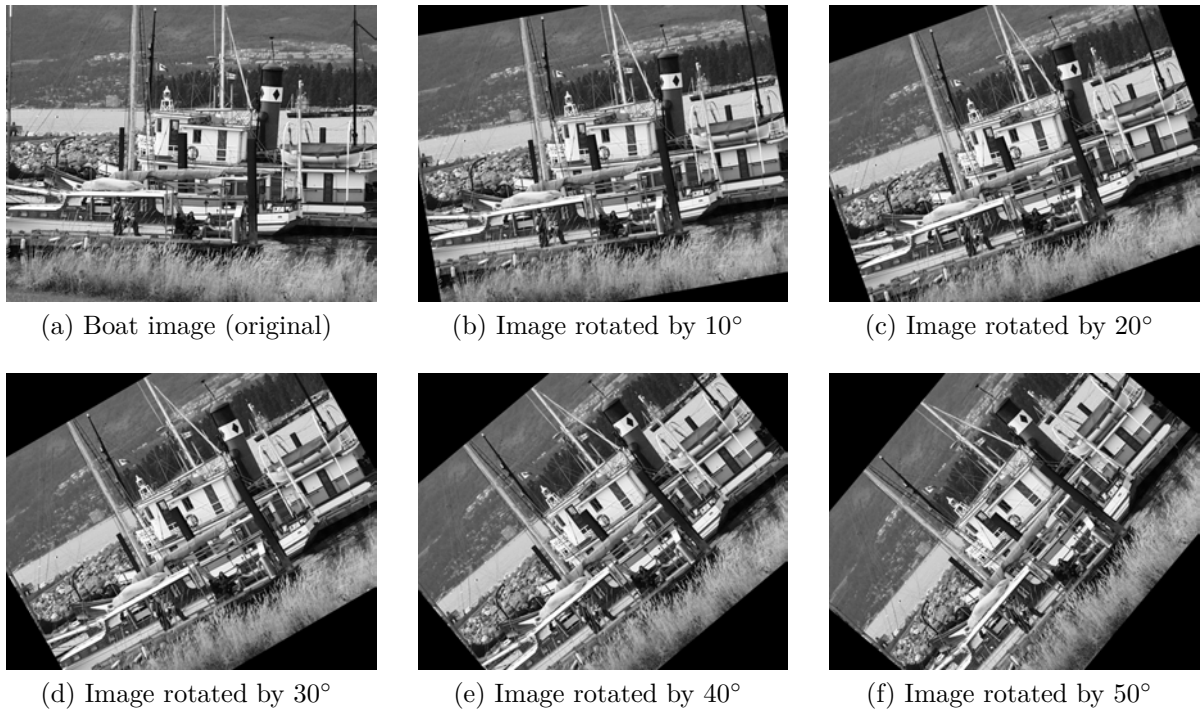


Figure 5.12.: The boat image samples from the Oxford's Visual Geometry Group dataset. Image transformations represent changes in rotation.

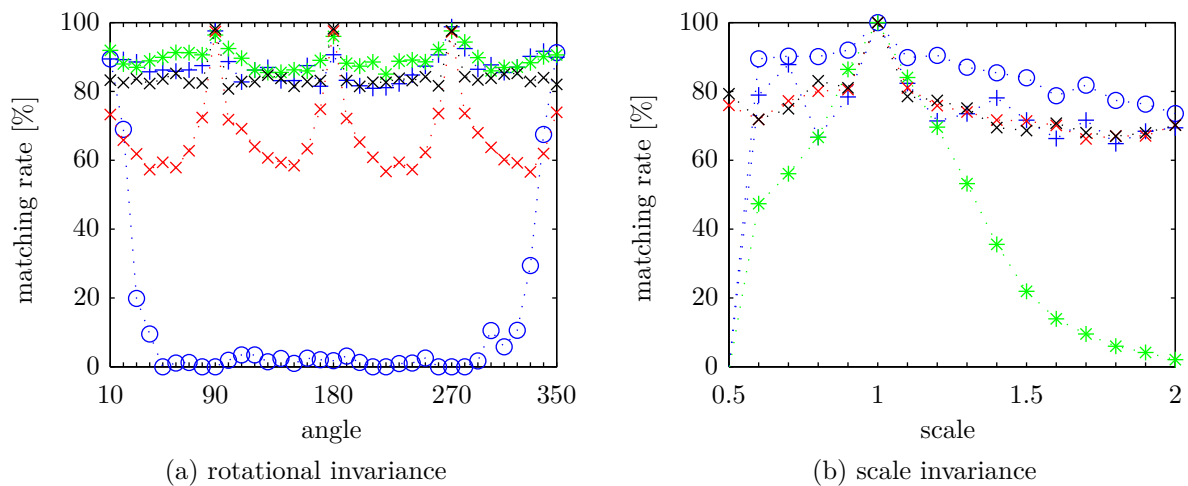


Figure 5.13.: Comparing the matching accuracy of the sORB feature descriptor with respect to varying scales and orientations.

4 times longer than the proposed sORB approach. For sORB, the computation time is only minimally increased compared to the ORB and sBRIEF descriptor, which indicates that the additional invariance can be implemented by causing just a small overhead. Surprisingly, SURF outperforms all other descriptors in terms of throughput. However, this is not due to its fast to compute approximations, but rather due to its heavy use of SSE and multi-processor hardware acceleration for its parallel implementation resulting in a decrease of computation time by a factor between 2 and 8. As the implementation of the sORB descriptor has not yet been optimized for hardware acceleration, the performance of the SURF descriptor on a comparable implementation would definitely be slower compared to the proposed sORB approach. This is also confirmed by the evaluations for the ORB descriptor in [RRKB11].

Descriptor	Computation time	Throughput
sORB	13.4 ms	0.068 $\frac{\text{ms}}{\text{descriptor}}$
sBRIEF	12.6 ms	0.064 $\frac{\text{ms}}{\text{descriptor}}$
ORB	12.9 ms	0.066 $\frac{\text{ms}}{\text{descriptor}}$
SURF	74.3 ms	0.047 $\frac{\text{ms}}{\text{descriptor}}$
SIFT	1071.4 ms	0.255 $\frac{\text{ms}}{\text{descriptor}}$

Table 5.2.: Computation time and throughput on a 640 x 480 image for descriptor computation.

These evaluations show that the proposed sORB descriptor generally outperforms ORB, sBRIEF, SURF, and SIFT on the Oxford standard image data set in terms of matching rate. Additionally, the performance of sORB on artificially rotated and scaled images has been evaluated, which confirms the results presented on the Oxford dataset. Concerning the computation time, sORB heavily benefits from the usage of fast to compute pixel comparisons. Therefore, when using comparable implementations without hardware acceleration, sORB is computed significantly faster than SIFT or SURF.

Evaluation of sparse bundle adjustment

Sparse bundle adjustment has been implemented based on the bundle adjustment library from Strasdat et al. [SMD⁺10b]. It is evaluated with respect to the reconstruction error, the kernel parameter c from (5.13) and the expected false data association rate. The evaluation is performed on simulated data in order to create a defined set of n 3D object points $\{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \dots, \hat{\mathbf{b}}_n\}$, $\hat{\mathbf{b}}_i \in \mathbb{R}^3$ and m different camera viewpoints $\{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \dots, \hat{\mathbf{a}}_m\}$, $\hat{\mathbf{a}}_j \in \mathbb{R}^6$. Data simulation is necessary in order to enable an evaluation of the reconstruction results

by comparing it with the artificially generated ground truth data. Figure 5.14 visualizes the setup of the simulated data.

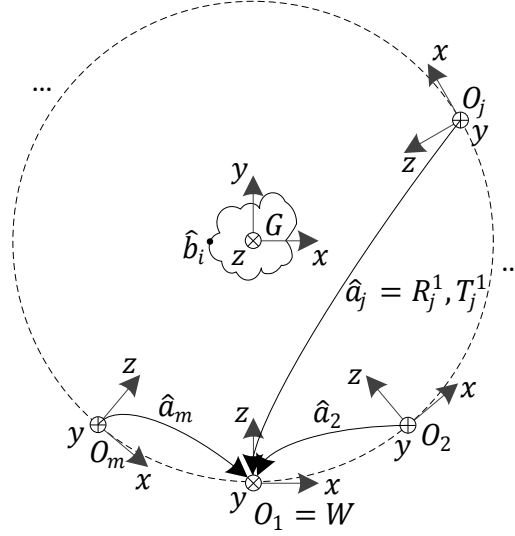


Figure 5.14.: Schematic setup of the simulation environment to evaluate the performance of the sparse bundle adjustment algorithm. The simulated object with coordinate system G and data points $\hat{\mathbf{b}}_i$ is placed in the center. It is recorded from m different camera configurations O_j in a circular arrangement around the object. Gaussian noise is added to the ideal data points $\hat{\mathbf{b}}_i$ and to the camera poses $\hat{\mathbf{a}}_j$ in order to mimic the measurement noise appearing in a real setup.

All cameras are positioned in the x - y plane of the object coordinate system G in circular arrangement around the object. The angular spacing between the cameras is 10° . The x - and z -axis of the camera coordinate systems O_j are lying within the x - y plane of the object coordinate system G , with the z -axis pointing towards the object center. The y -axis of each camera coordinate system is pointing towards the plane's normal vector.

Based on the ideal 3D object points $\hat{\mathbf{b}}_i \in \mathbb{R}^3$ and camera pose vectors $\hat{\mathbf{a}}_j \in \mathbb{R}^6$, the simulation applies Gaussian noise according to (5.15) to take account for the measurement noise appearing in the real setup.

$$\begin{aligned}
 \mathbf{b}_i^0 &= \hat{\mathbf{b}}_i + \mathcal{N}(0, \Sigma_{\mathbf{B}}), \quad \Sigma_{\mathbf{B}} = \text{diag}(\sigma_x, \sigma_y, \sigma_z) \\
 \mathbf{a}_j^0 &= \hat{\mathbf{a}}_j + \mathcal{N}(0, \Sigma_{\mathbf{A}}), \quad \Sigma_{\mathbf{A}} = \text{diag}(\sigma'_x, \sigma'_y, \sigma'_z, \sigma_\alpha, \sigma_\beta, \sigma_\gamma) \\
 \mathbf{x}_{ij} &= \hat{\mathbf{R}}_j \hat{\mathbf{b}}_i + \hat{\mathbf{T}}_j + \mathcal{N}(0, \Sigma_{\mathbf{B}})
 \end{aligned} \tag{5.15}$$

The parameter vectors \mathbf{a}_j^0 and \mathbf{b}_i^0 serve as an initial guess for the iterative LMA optimization. In order to simulate incorrect data associations, a defined percentage e_o of observed data points \mathbf{x}_{ij} visible from camera j and corresponding to the object point $\hat{\mathbf{b}}_i$ is randomly

associated to an object point $\hat{\mathbf{b}}_k, k \neq i$. The reconstruction results are evaluated with respect to the Euclidean distance of the reconstructed object model points \mathbf{b}_i to the ground truth data $\hat{\mathbf{b}}_i$ according to (5.16).

$$d = \frac{1}{n} \sum_{i=1}^n \|\mathbf{b}_i - \hat{\mathbf{b}}_i\| \quad (5.16)$$

The simulated object model consists of 200 feature points $\hat{\mathbf{b}}_i$, which are arranged in the shape of a cone. There are 36 different camera perspectives, whereas a feature point $\hat{\mathbf{b}}_i$ is measured as an observation $\mathbf{x}_{i(j-2)}, \mathbf{x}_{i(j-1)}, \dots, \mathbf{x}_{i(j+2)}$ in exactly 5 successive perspectives. All tests have been repeated 100 times. An overview of the simulation parameters and their values is given by Table 5.3.

Table 5.3.: The parameter set for evaluating the sparse bundle algorithm.

Value	Description
$i = 200, i \in \mathbb{N}$	number of simulated feature points b_i
$j = 36, j \in \mathbb{N}$	number of simulated viewpoints a_j
$l = 50, l \in \mathbb{N}$	maximal number of LMA iterations
$\sigma_x = \sigma_y = \sigma_z = 10^{-2}$ m	variance of observations in x - y - and z -directions
$\sigma'_x = \sigma'_y = \sigma'_z = 10^{-4}$ m	variance of camera pose in x - y - and z -directions
$\sigma_\alpha = \sigma_\beta = \sigma_\gamma = 10^{-3}$ rad	angular variance of camera pose
$0 \leq e_o \leq 0.25, e_o \in \mathbb{R}$	invalid data association rate
$10^{-7} \leq c \leq 10^{-3}, c \in \mathbb{R}$	kernel parameter range
$n = 100, n \in \mathbb{N}$	repetitions per parameter set

Figure 5.15 gives the mean reconstruction error according to (5.16) for different kernel parameters c in relation to different values of the incorrect data association rate e_o ranging from 0% to 25%. The dashed line indicates the initial mean error given by the measurement noise from (5.15).

The figure shows clearly, that the choice of a very small kernel parameter below 10^{-6} does not lead to any improvement compared to the initial reconstruction error. This is due to the fact, that a very small kernel parameter does no longer distinguish between feature point observations that fit well to the camera viewpoints and observations that originate from incorrect data associations. Even correct observations will have a residual r_{ij} that is too large for the kernel function to return a significant larger weight value compared to the incorrect observations. Therefore, approximately the same weight value is assigned to all observations and the optimization algorithm is not guided towards an optimal solution.

When inspecting the reconstruction error for $10^{-6} \leq c \leq 10^{-4}$, an improvement of the mean error compared to the initial error is visible. The mean error is reduced all the more by the LMA optimization, the lower the percentage of incorrect data associations is. This is intuitively clear, as the probability that wrong associations disturb the optimization process

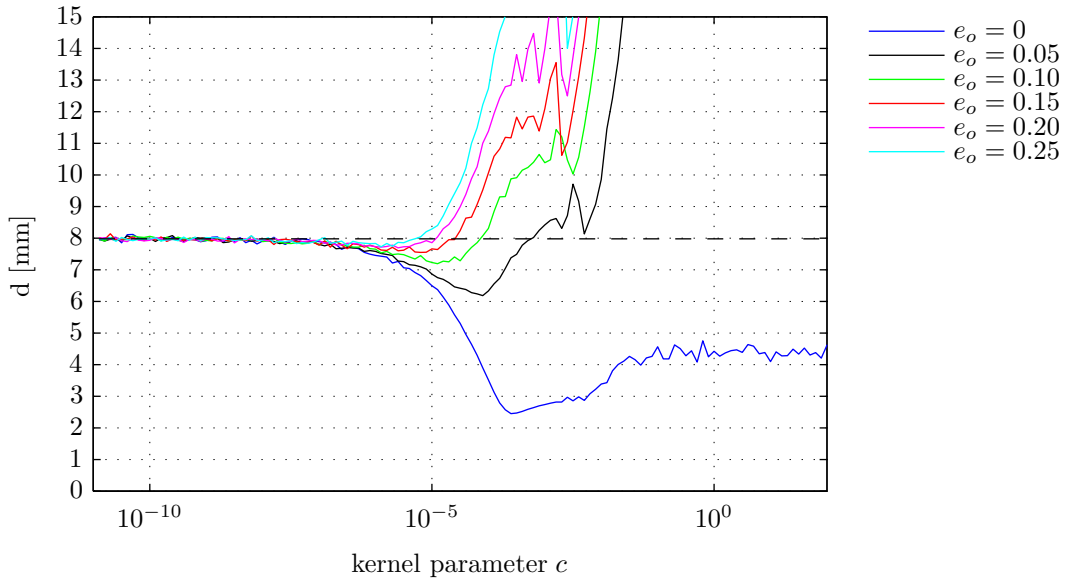


Figure 5.15.: Error rate plot.

is lowered with a decreasing e_o . It is also visible, that an optimal choice does always depend on the present incorrect data association rate. For a lower value of e_o , a larger value for the kernel parameter c should be chosen. This gives more weight on observations with a small residual as the probability of a wrongly associated observation is low. On the other hand, for larger e_o a smaller value of c should be chosen.

A kernel parameter, which is larger than 10^{-4} does no longer improve the reconstruction results and should therefore be avoided. It even lets the LMA optimization diverge resulting in an average mean error that is larger than the initial mean error for values of $e_o > 0.05$.

Figure 5.16 shows exemplary point clouds of object models that have been reconstructed using the proposed sparse bundle adjustment algorithm with a kernel value of $c = 10^{-4}$ and sORB feature points.

5.3. Modeling of texture-less objects

This section presents a novel method for the rapid and dense computation of local 2D image and 3D depth cues on the basis of which a global histogram-based descriptor for the distinct description of object models without rich texture is computed. The representation of the local point feature is using a binary descriptor. This per-pixel information is aggregated into a histogram over the object related image area. The histogram is reordered and normalized in order to achieve rotation and scale invariance.



Figure 5.16.: Object models reconstructed using sparse bundle adjustment.

5.3.1. Local feature descriptor

The local point feature descriptor captures quantized 2D gradient information from the color image and quantized 3D gradient information from the range image to describe an object area. However, compared to other approaches [HCI⁺12, HCI⁺11, HLI⁺10], the dense computation of the descriptor values for each image pixel is based on the principles of dynamic programming to break down the estimation of 2D and 3D gradients into simple comparisons that are solved only once and reused afterwards to compute the remaining descriptors. This results in a significant speed up in descriptor computation.

Figure 5.17 shows the layout of the binary point feature descriptor. In general, a real-valued function $f(x, y)$ is evaluated on different pixel-pairs as indicated by the dashed lines and the results are thresholded to compute binary values which are concatenated to a local descriptor. The layout of the feature descriptor has been chosen to compare pixel

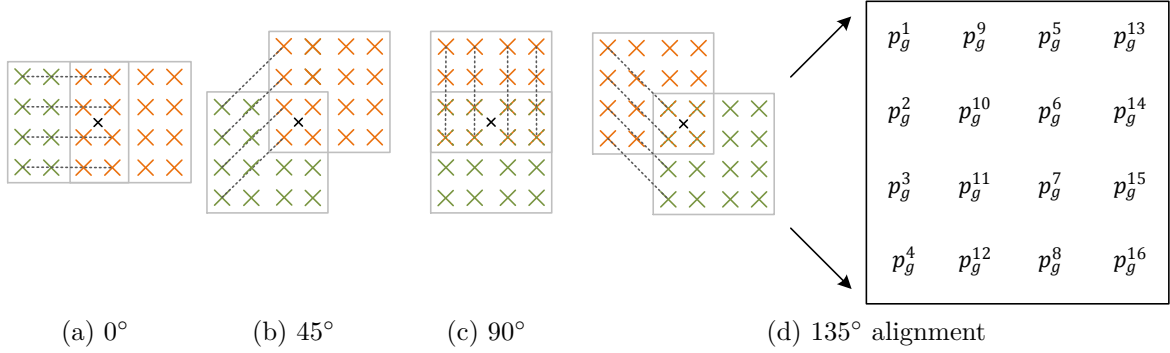


Figure 5.17.: Layout of the feature descriptor to compute the binary descriptor values for the feature point p located at the black cross in the center of the colored squares. The dashed lines indicate corresponding pixel positions p_r^i and p_g^i from the red and green square, at which a function $f(p_g^i, p_r^i)$ is evaluated. The value of f is thresholded to compute a binary value $\tau(f(p_g^i, p_r^i)) \in \{0, 1\}$.

of opposing squared regions around the feature point p as indicated by the green and red squares in Figure 5.17. The arrangement of the opposing squares is rotated around the feature point p through an angle of 45° in anticlockwise direction (Figure 5.17(a) - 5.17(d)). This results in a total of 4×2 different opposing squared regions that contribute to the value of the binary descriptor. The computation of the descriptor value is the same for each alignment of the opposing rectangles, therefore the following sections refer to Figure 5.17(d) without loss of generality, if not stated otherwise.

Within each of the squared regions 16 equally spaced pixel positions $p_s^i, i \in \{1..16\}, s \in \{r, g\}$ are selected, where s refers to the red or green square in Figure 5.17(d). For each of the 16 pixel positions p_g^i a counterpart p_r^i is attributed as indicated by the dashed lines resulting in 16 pixel-pairs per alignment. The RGB-D values corresponding to the pixel-pairs are jointly evaluated to compute the necessary information for the later 2D and 3D

gradient estimation and the results are binarized, which allows the application of the Hamming operator (Section 2.2.2) in order to speed up further computations. The number of 16 pixel-pairs for each distinct alignment of the square regions has been chosen to increase the robustness of the proposed descriptor. It reduces the effect of noisy measurements and avoids the need for a preceding smoothing operation on the image data due to an averaging of the evaluation results among all 16 pixel-pairs, which is described in the following sections.

2D gradient estimation

2D gradients are computed on the RGB channels of the color image. The gradient direction is omitted to avoid the influence of bright or dark object-backgrounds on the 2D gradient direction and the gradients are computed on each channel individually. The algorithm computes quantized gradient orientations ranging from 0° to 135° in steps of 45° . Initially, the 2D gradient magnitudes for the orientation corresponding to the current arrangement of the opposing rectangles (0° , 45° , 90° or 135°) are computed in (5.17) by computing the maximal intensity difference over all color channels at all 16 pixel-pairs.

$$f_{2D}: \mathbb{N}^3 \times \mathbb{N}^3 \rightarrow \mathbb{N} \quad (5.17)$$

$$\left(\mathbf{c}(p_g^i), \mathbf{c}(p_r^i) \right) \mapsto f_{2D} \left(\mathbf{c}(p_g^i), \mathbf{c}(p_r^i) \right) = \max_{j \in \{\mathbf{R}, \mathbf{G}, \mathbf{B}\}} |c_j(p_g^i) - c_j(p_r^i)|$$

The three color channels are denoted by $j \in \{R, G, B\}$ and $c_j(p)$ returns the intensity value for color channel j at the pixel p . Then, the result of f_{2D} is binarized as given in (5.18).

$$\tau_{2D}: \mathbb{N} \rightarrow \{0, 1\} \quad (5.18)$$

$$f_{2D}(a, b) \mapsto \tau_{2D}(f_{2D}(a, b)) = \begin{cases} 1, & \text{if } f_{2D}(a, b) < t_{2D} \\ 0, & \text{otherwise} \end{cases}$$

τ_{2D} compares the value of f_{2D} against a predefined fixed threshold $t_{2D} \in \{0, 1, \dots, 255\}$ expressed by a fixed intensity value. An optimal value of $t_{2D} = 70$ has experimentally been determined according to Section 5.3.4. Finally, the binary values computed from all 16 pixel-pairs in 5.18 are concatenated to a 16-bit descriptor d_{2D}^k .

In order to assign a single 2D gradient orientation to the feature point, the discretized 16-bit descriptors d_{2D}^k are evaluated in (5.19) - (5.21) for all four alignments $k \in \{0, 1, 2, 3\}$ of the opposing square regions from Figure 5.17 .

$$x = \sum_{k=0}^3 \left(\text{Ham} \left(d_{2D}^k, \mathbf{0} \right) \cos \left(\frac{45\pi}{180} k \right) \right) \quad (5.19)$$

$$y = \sum_{k=0}^3 \left(\text{Ham} \left(d_{2D}^k, \mathbf{0} \right) \sin \left(\frac{45\pi}{180} k \right) \right) \quad (5.20)$$

$$\text{ori}_{2D}(p) = \begin{cases} 0^\circ, & -\frac{\pi}{8} < \text{atan2}(y, x) \leq \frac{\pi}{8} \\ 45^\circ, & \frac{\pi}{8} < \text{atan2}(y, x) \leq \frac{3\pi}{8} \\ 90^\circ, & \frac{3\pi}{8} < \text{atan2}(y, x) \leq \frac{5\pi}{8} \\ 135^\circ, & \frac{5\pi}{8} < \text{atan2}(y, x) < \frac{7\pi}{8} \end{cases} \quad (5.21)$$

$\text{Ham}(h_1, h_2)$ computes the Hamming distance between the binary values of h_1 and h_2 . The values of the sin, cos and atan2 functions have been implemented using a look-up table in order to speed up computations. Finally, the discretized gradient orientation closest to the value of $\text{ori}_{2D}(p)$ is selected and saved as the 2D gradient orientation at feature point p . A visual impression of the computed and quantized 2D gradient orientation is given by Figure 5.18.

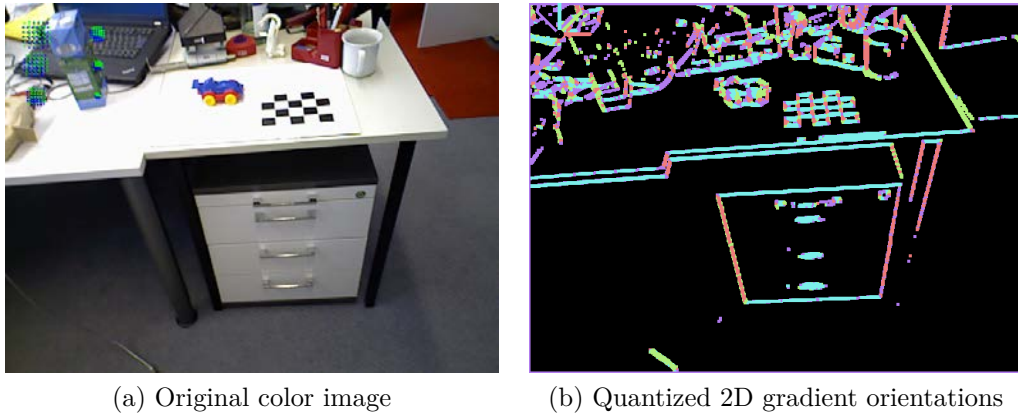


Figure 5.18.: Output of the proposed method for 2D gradient computation using discrete angles of 0° , 45° , 90° and 135° . The different gradient orientations are encoded with different colors on the right image.

3D gradient estimation

The idea of the 3D gradient estimation is to estimate the gradient of the tangent plane going through the feature point p . Therefore, the function f_{3D} for 3D normal estimation measures the gradient on the range image at the current pixel-pair as given in (5.22).

$$f_{3D}: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \quad (5.22)$$

$$(z_g^i, z_r^i) \mapsto f_{3D}(z_g^i, z_r^i) = z_g^i - z_r^i$$

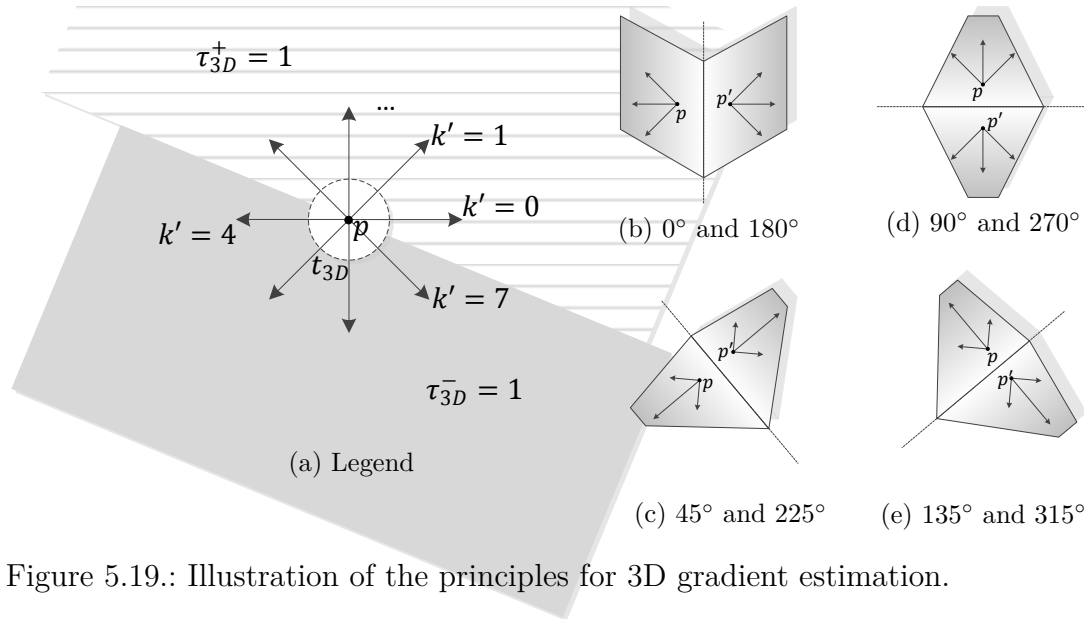


Figure 5.19.: Illustration of the principles for 3D gradient estimation.

$z_g^i = z(p_g^i)$ denotes the measured range value at the pixel coordinate p_g^i . f_{3D} estimates the 3D gradient orientation at the current feature point for eight discretization angles ranging from 0° to 315° in steps of 45° . The binarization of f_{3D} is given in (5.23) and (5.24).

$$\tau_{3D}^+ : \mathbb{R} \rightarrow \{0, 1\}$$

$$f_{3D}(z_g^i, z_r^i) \mapsto \tau_{3D}^+(f_{3D}(z_g^i, z_r^i)) = \begin{cases} 1, & \text{if } f_{3D}(z_g^i, z_r^i) > t_{3D} \\ 0, & \text{otherwise} \end{cases} \quad (5.23)$$

$$\tau_{3D}^- : \mathbb{R} \rightarrow \{0, 1\}$$

$$f_{3D}(z_g^i, z_r^i) \mapsto \tau_{3D}^-(f_{3D}(z_g^i, z_r^i)) = \begin{cases} 1, & \text{if } f_{3D}(z_g^i, z_r^i) < -t_{3D} \\ 0, & \text{otherwise} \end{cases} \quad (5.24)$$

The value of f_{3D} is compared against two thresholds t_{3D} and $-t_{3D}$ expressed in meters in order to preserve the sign of the gradient directions. Compared to t_{2D} in (5.18), t_{3D} is not fixed, but modified proportional to the measured distance at the current feature point location in order to compensate for the lower spatial resolution of the input data at increasing distances. A value of $t_{3D} = 0.002$ m at a measured distance of 0.5 m has experimentally been determined to yield best results. Finally, the binary values from (5.23) and (5.24), which have been computed for all 16 pixel-pairs of all eight orientations, are concatenated to 16-bit descriptors $d_{3D}^{k'}$, $k' \in \{0, 1, \dots, 7\}$. Here, $d_{3D}^{k'}$, $k' \in \{0, 1, 2, 3\}$ holds the evaluation results from τ_{3D}^+ and $d_{3D}^{k'}$, $k' \in \{4, 5, 6, 7\}$ holds the evaluation results from τ_{3D}^- .

The computations from (5.22), (5.23), and (5.24) are illustrated by Figure 5.19. The orientation of the vectors corresponds to the layout of the feature descriptor from Figure

5.17, e.g. the vectors pointing towards $k = 0$ and $k = 4$ correspond to the results when evaluating the pixel-pairs illustrated in Figure 5.17(a) with the simplification that just one pixel-pair per orientation is visualized as a vector. In general, when (5.23) evaluates to 1 the vectors are oriented towards $k \in \{0, 1, 2, 3\}$, otherwise, when (5.24) evaluates to 1 the vectors are oriented towards $k \in \{4, 5, 6, 7\}$. The magnitude of the vectors correspond to the value of (5.22) e.g. a large vector magnitude corresponds to a significant range difference of the corresponding pixel-pair. The threshold t_{3D} is indicated by the dashed circle in 5.19(a). Then, the orientation of a plane is expressed by inspecting the magnitudes of each vector as illustrated by Figure 5.19(b)-5.19(e). However, instead of storing the real valued vector magnitudes directly, they are encoded for each orientation by the amount of pixel-pairs evaluating to 1 using (5.25) - (5.27).

$$x = \sum_{k'=0}^7 \left(\text{Ham} \left(d_{3D}^{k'}, \mathbf{0} \right) \cos \left(\frac{45\pi}{180} k' \right) \right) \quad (5.25)$$

$$y = \sum_{k'=0}^7 \left(\text{Ham} \left(d_{3D}^{k'}, \mathbf{0} \right) \sin \left(\frac{45\pi}{180} k' \right) \right) \quad (5.26)$$

$$\text{ori}_{3D}(p) = \begin{cases} 0^\circ, & -\frac{\pi}{8} \leq \text{atan2}(y, x) \leq \frac{\pi}{8} \\ 45^\circ, & \frac{\pi}{8} < \text{atan2}(y, x) \leq \frac{3\pi}{8} \\ 90^\circ, & \frac{3\pi}{8} < \text{atan2}(y, x) \leq \frac{5\pi}{8} \\ 135^\circ, & \frac{5\pi}{8} < \text{atan2}(y, x) < \frac{7\pi}{8} \\ 180^\circ, & \frac{7\pi}{8} \leq \text{atan2}(y, x) \leq \pi \\ 180^\circ, & -\pi \leq \text{atan2}(y, x) \leq -\frac{7\pi}{8} \\ 225^\circ, & -\frac{7\pi}{8} < \text{atan2}(y, x) \leq -\frac{5\pi}{8} \\ 270^\circ, & -\frac{5\pi}{8} < \text{atan2}(y, x) \leq -\frac{3\pi}{8} \\ 315^\circ, & -\frac{3\pi}{8} < \text{atan2}(y, x) < -\frac{\pi}{8} \end{cases} \quad (5.27)$$

A visual impression of the computed and quantized 3D gradient orientations is given by Figure 5.20.

5.3.2. Dynamic programming for rapid descriptor computation

Due to the dense calculation of the descriptor for each image pixel, it is of utmost importance to enable a fast computation of the feature descriptors. Therefore, the descriptor layout of the pixel comparisons has been designed explicitly for the application of the dynamic

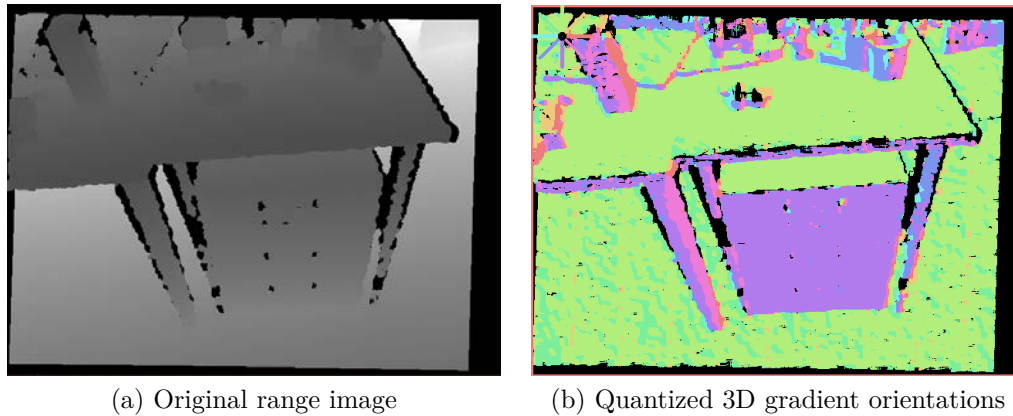


Figure 5.20.: Output of the proposed method for 3D gradient computation using discrete angles ranging from 0° to 315° in steps of 45° .

programming paradigm. Figure 5.21 illustrates the principle of reusing already computed pixel-pair evaluations for the computation of the current descriptor. The illustration is limited to the evaluation of a single alignment (here, the 135° alignment from Figure 5.17(d)), but the principles may be applied in the same way to all other alignments.

Initially, the descriptor computation starts with the top left pixel and proceeds by moving from left to right and from top to bottom. Only the first descriptor values must be calculated from scratch. All successive descriptors may reuse the results from their predecessors reducing the number of pixel-pair evaluations for all four alignments for the descriptors of the first row from 4×16 to 4×4 (Figure 5.21(a)). Once the descriptors of the first row have been computed, an additional reduction in the number of evaluations from 4×4 to 4×1 (Figure 5.21(b)) is achieved. A further speed-up of the descriptor computations results from the binarization as explained in (5.18), (5.23) and (5.24). It limits the amount of data that needs to be copied and enables the usage of fast descriptor assignments by simple bit-shifts.

5.3.3. Appearance-based object modeling

The setup for object training is visualized in Figure 5.22. In order to teach a new object to the system, the object is placed on a predefined position at a checkerboard. Then, the camera is moved around the object to record images from different viewpoints. The training procedure detects the pose of the checkerboard and infers the position of the object relative to it. By the use of a virtual 3D box with a predefined size depending on the object's dimensions, the image data corresponding to the object is determined and the background is removed. Finally, local point feature descriptors are computed according to section 5.3.1 for all remaining pixel.

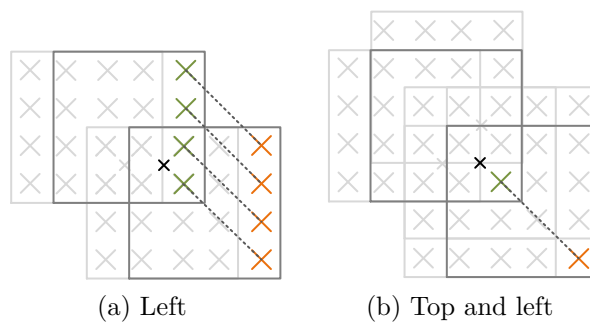


Figure 5.21.: Computation of the binary descriptor values for the pixel location at the black cross in the center of the two squares using dynamic programming. Considering Figure 5.21(a), all descriptors to the left of the current pixel have already been evaluated. For Figure 5.21(b), additionally the descriptors of the top row have been computed.

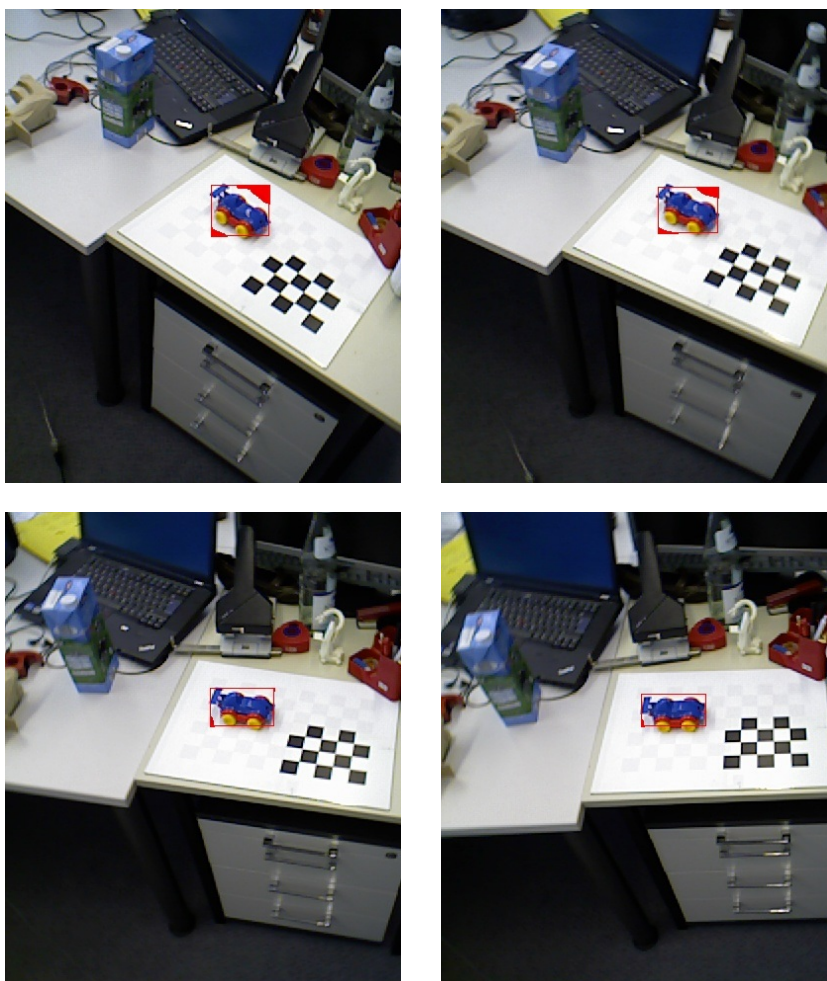


Figure 5.22.: Recorded training sequence for the car object. The ground truth pose of the object is given by the checkerboard. A virtual box in 3D space is placed around the car to separate the object data (native color values within the red rectangle) from the background.

Even though it would have been possible to follow the approach of Hinterstoisser et al. by applying brute force template matching for object description and recognition on the 2D and 3D image cues, the proposed method applies a global descriptor for object description that exhibits scale and rotation invariance. This approach has the advantage, that the number of templates necessary for a robust object description is significantly reduced. The proposed procedure for object training and recognition is slower compared to the approach of Hinterstoisser. However, in the present implementation no optimizations concerning SSE accelerations or cache optimizations have been conducted.

The basic idea of the global descriptor is to capture the local 2D and 3D gradient distribution of the object. Therefore, the training data is divided into smaller spatial regions according to Figure 5.23 and the global descriptor is computed by counting the frequency of 3D and 2D gradient directions over these regions. The final object descriptor is constructed by concatenating the histogram values from all five spatial regions.

In order to speed up the computation of the histograms, integral images are used as explained in Section 2.2.1. Therefore, for each of the resulting gradient directions from (5.21) and (5.27) a separate gradient map is created, where each pixel of the gradient map corresponds to a pixel in the original image. This results in four maps for the 2D gradients, eight maps for the 3D gradients, and two maps representing undefined 2D or 3D gradients which do not exhibit any significant direction. The significance of a gradient direction is determined by comparing the magnitude of the 2D vector (x, y) from (5.19) and (5.20) for the 2D gradients and the 2D vector (x, y) from (5.25) and (5.26) for the 3D gradients against a predefined threshold. Each gradient map is of the same size as the source image and holds a value of 1 at pixel position \mathbf{p} if the corresponding 2D or 3D gradient direction is dominant among all other 2D or 3D gradients. This results in exactly 2 gradient map entries for each pixel of the source image.

By the use of one integral image for each gradient map, it is feasible to compute the frequency of a single gradient direction for a rectangular area with only 4 operations. Once the gradient maps have been established, they are converted to integral images. The computation of a histogram over local descriptors in the area covered by the sliding window is then performed by simply computing the number of histogram entries for each gradient map using their integral image representations. This results in a total of 14×4 operations. The individual results are concatenated to create the final histogram of gradients.

Rotation invariance

Invariance against rotation is achieved by computing the most dominant 3D gradient orientation and reordering the values of the histograms accordingly. Assuming that the gradient with the strongest magnitude has an angle of 45° , then the first histogram entry will be the gradient frequency of direction 45° , followed by direction 90° , 135° and so on, until

the gradient frequency 0° is appended to the end. The dominant 3D gradient orientation is computed based on the image area indicated by the red rectangle in Figure 5.23. The rectangle is centered on the object and of smaller size than its outline in order to minimize the influence of the background on the main orientation assignment.

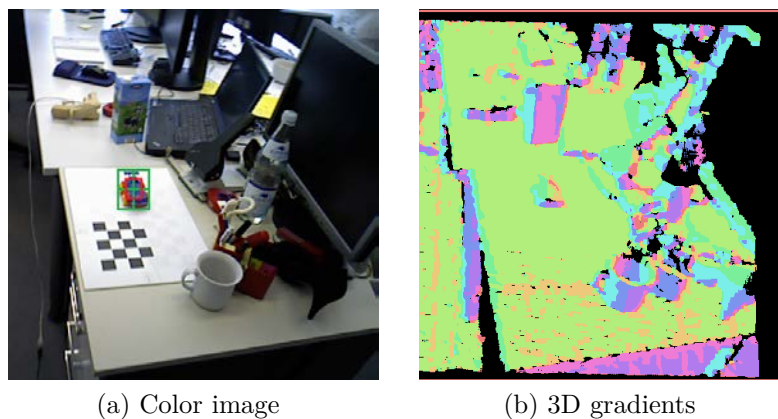


Figure 5.23.: One local histogram of gradient orientations is computed for each of the four areas surrounded by the green rectangles and one local histogram for the area surrounded by the red rectangle. All rectangles are of the same size. The main 3D gradient orientation from the histogram describing the red area is used for descriptor alignment in order to achieve rotation invariance.

Scale invariance

In order to achieve scale invariance, the histogram entries are normalized. Let $\mathbf{v}^i = (v_1^i, v_2^i, \dots, v_k^i)$ be the original k -dimensional descriptor entries from histogram $i \in \{1, 2, \dots, 5\}$, then the normalized descriptor $\bar{\mathbf{v}}^i = (\bar{v}_1^i, \bar{v}_2^i, \dots, \bar{v}_k^i)$ is given by

$$\bar{v}_j^i = \frac{v_j^i}{\sum_{l=1..k} v_l^i} \quad (5.28)$$

Additionally, the dimensions of the training rectangles (Figure 5.23) are stored together with the measured range value at the center of the object. Compared to a pure histogram normalization, the proposed approach preserves the object dimensions as the search window for a later recognition is resized in order to fit to the stored dimensions of the training rectangle e.g. assume that the dimension of the training rectangles is 10×20 pixel at a distance of 1 m, then the search window for recognition will be resized to 20×40 pixel at a distance of 2 m or to 5×10 pixel at a distance of 0.5 m.

5.3.4. Evaluation

The target hardware platform is an Intel[®] Core[™] i7-2860QM with 2.5 GHz and 8 GB RAM. The computation time for 2D and 3D gradient estimation is compared to the approach of Hinterstoisser et al. for textureless object recognition as it was given in the OpenCV [Bra00] repository from 09/2012. Table 5.4 shows that the proposed approach performs faster while still exhibiting a similar recognition performance as shown in Section 6.3.3 even without the usage of any SSE or cache optimizations.

	Original approach	Proposed approach
2D gradient estimation	29 $\frac{\text{ms}}{\text{frame}}$	27 $\frac{\text{ms}}{\text{frame}}$
3D gradient estimation	39 $\frac{\text{ms}}{\text{frame}}$	36 $\frac{\text{ms}}{\text{frame}}$

Table 5.4.: Computation time on a 640 x 480 image for 2D and 3D gradient estimation of the original approach by Hinterstoisser et al. and the proposed approach using dynamic programming.

An evaluation of the matching rate is conducted on an object level in Section 6.3.3. A feature point based evaluation, as it has been performed for the sORB descriptor in Section 5.2.5 does not apply to the proposed descriptor due to its global nature.

6. Object Recognition

Object recognition determines the presence or absence of an object within a scene. It is based on the information extracted during object modeling to associate unknown sensor stimuli with known object instances. Object localization follows object recognition to determine accurate location information, i.e. an object's position and orientation, for each object within the scene. It is necessary to enable a robotic system to position its gripper relative to the object's pose in order to grasp it.

Figure 6.1 gives an overview of the complete procedure which is presented in more detail within the following sections. At the beginning, many processing steps for object recognition are identical or similar to object modeling. Initially, sensor fusion from Section 4 delivers RGB-D data from an arbitrary scene, where objects are to be searched for. The data is segmented based on a fixed maximal distance value $d > 0$, so that pixel at a distance greater than d are ignored. Depending on the object type, the further processing steps are distinguished between textured and texture-less objects.

When searching for textured objects, sORB or SURF feature descriptors are extracted from the scene image as explained by Section 5.2.1. The descriptors are associated with all object models from the database using a k-d-tree data structure. Matching descriptor pairs are sorted by their matching distance and divided into two groups corresponding to close and far matchings. Then, samples of closely spaced descriptor pairs, which are corresponding to the same object instance, are subject to pose estimation. By the use of *Progressive Sample Consensus* (ProSaC), a variant of *Random Sample Consensus* RanSaC, for pose estimation, groups of three matching descriptor pairs are repeatedly drawn in the sequence of their matching distance, based on which the pose of the object is estimated. Finally, from all pose estimates, the one that fits best to the scene is selected.

When searching for texture-less objects, 2D and 3D gradients are extracted from the scene image according to Section 5.3. The feature descriptors are associated with descriptors from the object model database. Due to the appearance-based object recognition approach, each descriptor within the database describes a complete object view, which is connected to a corresponding pose. Therefore, pose information is directly given by the closest matching descriptor pair.

This Section presents a novel algorithm for data association that explicitly addresses the handling of multiple occurring features like brand names on an object's surface. The

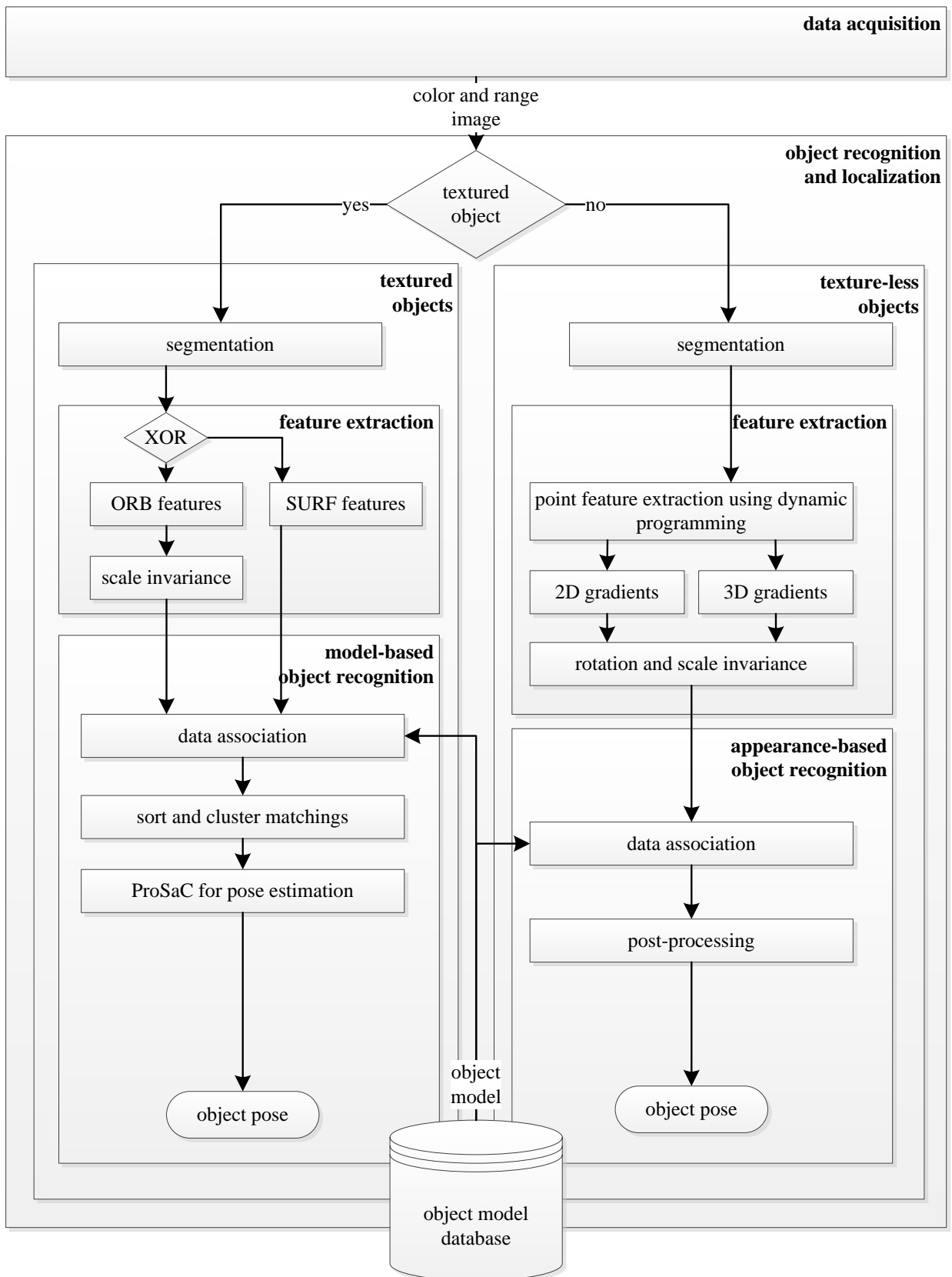


Figure 6.1.: Schematic overview of the individual processing steps for object recognition.

recognition of textured objects is based on the proposed sORB feature descriptor, the recognition of texture-less objects makes use of an adaptive sliding window approach using 2D and 3D gradients as described by Section 5.3. The presented results for the recognition of textured objects have been published together with the results for object modeling in [FABV12]. The proposed approach for texture-less object recognition and modeling has been published in [FBAV13].

6.1. Data acquisition

The sensor setup for object recognition is the same as for object modeling. However, in contrast to object modeling that works on a sequence of RGB-D images, only a single object image is processed. Consecutive object images are processed independently of one another. This object recognition approach is termed *single shot* object recognition.

On the one hand, single shot object recognition has the advantage that it outputs a recognition result for each image. This makes the approach faster compared to approaches that incorporate the temporal dependence of consecutive images. On the other hand, single shot object recognition does not explicitly exclude the incorporation of temporal dependencies. It rather constitutes a necessary first step on top of which temporal dependencies may be established, e.g. by accumulating the results of single shot object detectors as votes over a defined temporal domain.

6.2. Recognition of textured objects

In general, texture simplifies the recognition process as it enables the definition of local features, which are located at a fixed position on the object and the scene. Using point-to-point association of local features from the scene with features saved in the object models as explained in Section 5.2, it is possible to apply standard methods for 3D-3D pose estimation according to Section 2.3.

6.2.1. Pre-processing

Experiments have shown, that typical household objects like cups or bottles, which are smaller than $0.3\text{ m} \times 0.3\text{ m} \times 0.3\text{ m}$, are no longer recognized when they are further away than 3 m from the camera system. Due to the limited camera capabilities, the increased image noise distorts the feature point descriptors, which prevents a reliable matching. Therefore, pre-processing applies a simple distance based segmentation onto the sensor data to mask pixel exceeding a distance of 3 m. This saves computation time when locating feature points

and avoids the computation of unnecessary feature descriptors. It also reduces the search space for data association, which speeds up feature point matching.

In order to decrease the influence of outliers on the range data, the wavefront propagation procedure as described by Section 4.2 is applied on the range image after the range-based filtering. Figure 6.2 gives a visual impression on the filtering effects for a disparity image using stereo vision.

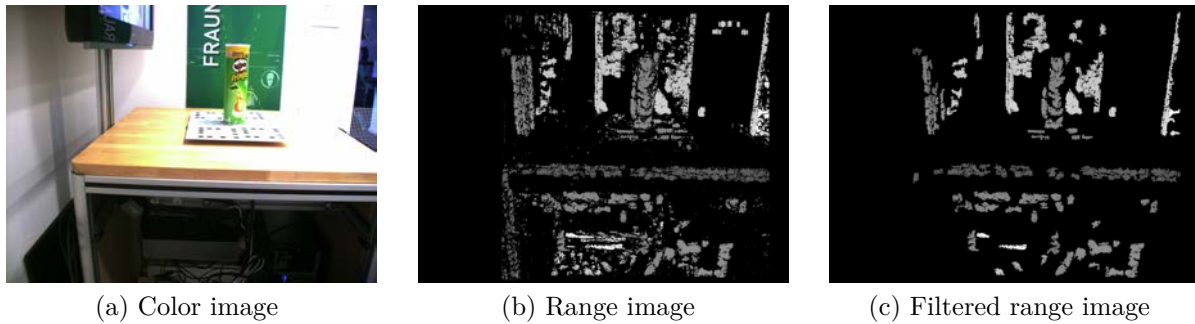


Figure 6.2.: Image pre-processing for object recognition using distance-based filtering and wavefront propagation on a disparity image from stereo vision.

6.2.2. Data association and clustering

Feature point matching is performed by comparing the Euclidean distance of feature descriptors from the scene image with all feature descriptors from all object models using a k-d-tree data structure. The principles of a k-d-tree are detailed in Section 2.2.2. A widely used method to determine the validity of feature point correspondences is to consider the closest two matches of an object model with the query descriptor from the scene. Let $d_1 \in \mathbb{R}$ denote the distance between the query point and its closest matching and $d_2 \in \mathbb{R}, d_1 \leq d_2$ be the distance between the query point and its second closest matching from the object model. Then the distances d_1 and d_2 are compared. Only when the distance ratio $\frac{d_1}{d_2}$ is smaller than a fixed threshold e.g. 0.8, the validity of the correspondence is accepted. The argumentation is based on the assumption that feature descriptor matches originating from noise are not unique.

However, this method has the disadvantage that it discards valid feature point matches as well e.g. when similar brand names and logos, resulting in similar descriptors, appear on different objects. Therefore, a method addressing this problem is proposed as follows. Given a query descriptor from the scene image, all correspondences from all models with a predefined maximal distance $t_B \in \mathbb{R}$ are saved and classified according to a second distance threshold $t_A \in \mathbb{R}$, with $t_A \ll t_B$. The procedure is repeated for all feature points of the scene. This gives us two sets A and B of feature point correspondences, one that holds matches with distances smaller than t_A (strong matches) and one that holds matches

with a descriptor distance between t_A and t_B (weak matches). The sets A and B are split up according to the object instance membership of the matchings, resulting in two sets of feature point correspondences A_i and B_i for each object instance $i \in \mathbb{N}, 1 \leq i \leq |\text{object models}|$. The procedure is visualized by Figure 6.3.

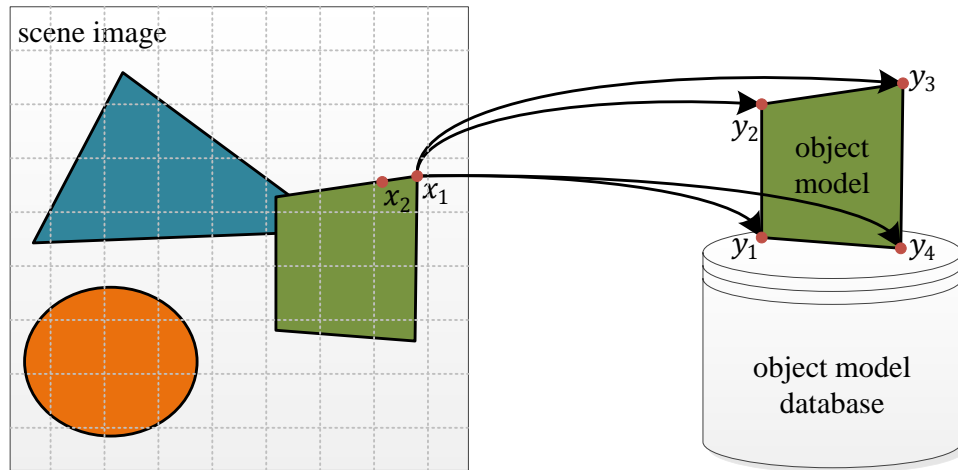


Figure 6.3.: Data association of feature point x_1 from the scene with feature points $y_1, y_2, y_3,$ and y_4 from the object model. The associations are classified based on their matching distance. Strong matches with a high similarity are collected within the set $A = \{(x_1, y_3), (x_1, y_4)\}$, e.g. as $y_3, y_4,$ and x_1 are all associated to sharp angles of the green shape, associations with a larger descriptor distance are placed in the set $B = \{(x_1, y_1), (x_1, y_2)\}$.

It must be mentioned, that it is explicitly allowed to have multi-associations of a single feature point from the scene to several matching feature points from different object instances. This is related to the fact, that there may be multiple objects of the same type within an image. Furthermore, it is not assumed, that a specific object is searched, but rather it is the goal to recognize any objects within the scene. Therefore, to localize all known object instances, the sets A_i and B_i are successively parsed in order to determine if the matches lead to a correct object localization or originate from noise instead.

Initially, for each object i , the set A_i is sorted by the matching distance of the correspondences. Then, a 2D grid is superimposed on the scene image with an edge size of k pixel to perform a spatial down-sampling of the set A_i . Out of all correspondences with 2D spatial coordinates covered by the same grid cell, only the one with the closest matching distance is inserted into the set \bar{A}_i . In order to determine the set \bar{A}_i , A_i is processed sequentially, beginning with the closest matching. One after another, the matches are inserted into the grid cells according to their 2D spatial coordinates of the feature point on the scene image. Therefore, the correspondences with closest matching distance are occupying the grid cells at first and any correspondence falling into an occupied grid cell later on must have a larger matching distance and can be rejected without any computation intensive comparisons.

The idea of the set \bar{A}_i is to get feature point correspondences with small matching distances that are equally scattered across the image. These feature points will guide the search for object instances within the current scene image as explained in Section 6.2.3. To return to the example from Figure 6.3, the two feature points from x_1 and x_2 would occupy the same grid cell. However, the correspondences from x_1 would ideally have smaller descriptor distances than the correspondences from x_2 , e.g. because of the inaccurate alignment of x_2 along an edge compared to the exact alignment of x_1 at a corner. Therefore, only the correspondences of x_1 would be inserted into \bar{A}_i .

6.2.3. Pose estimation

Pose estimation is based on ProSaC as described in Section 2.3.2. Based on the idea that an object model has a fixed size and therefore its feature points within the scene image will be located close to each other, not all correspondences from the scene image are considered for the semi-random sampling procedure of ProSaC. Instead, a small set of spatially close correspondences is considered based on the subset \bar{A}_i from Section 6.2.2. For each matching feature point pair $(x_j, y_j) \in \bar{A}_i$, where x_j corresponds to a feature point on the scene image and y_j corresponds to a matching feature point from the object model i , other correspondences $(x_k, y_k), k \in \mathbb{N}$ for object model i close to x_j are extracted from A_i and B_i and stored within the set C_{ij} . Only the elements of C_{ij} are subject to the ProSaC sampling for repeated and guided drawing of three sample points to fit the complete object model into the scene image. Figure 6.4 illustrates the proposed procedure.

Once ProSaC has returned a localization estimate that satisfies the non-randomness condition or ProSaC has stopped without any valid localization estimate, the procedure is repeated for the next point of \bar{A}_i . A 6D-aligned bounding box is obtained for each successful object localization. In case two bounding box estimates intersect, only the one with the larger ProSaC non-randomness probability is kept. The other pose estimate is discarded.

6.2.4. Evaluation

The proposed method for textured object recognition has been tested on a dataset consisting of 25 different objects (Figure 6.5). The dataset holds for each object a set of training images (Figure 6.6) and a set of testing images (Figure 6.7). Training images have been recorded by placing the object on a turn table in front of the camera system, which rotates the object by steps of 10° . A marker board is placed underneath the object to allow the inference of its pose and to enable the 3D object model construction according to Section 5.2.

The testing dataset is recorded in a manner similar to the training dataset. A reference object, which is to be detected, is placed on top of a marker board and the recognition of the markers serves as the ground truth information for the object's pose. Compared to

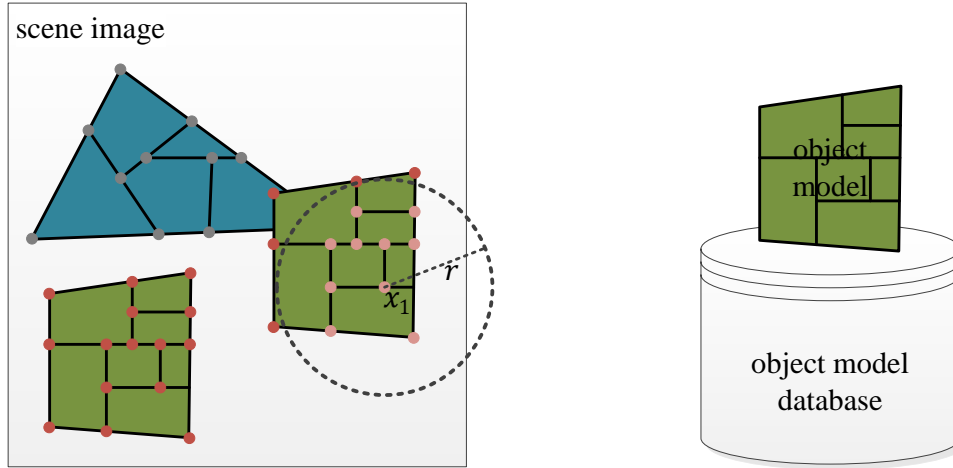


Figure 6.4.: ProSaC sampling of feature point correspondences. The object model i to be recognized is illustrated on the right. On the left, the gray dots denote feature points not corresponding to the object model and the red dots denote feature points that could be matched with feature points from the object model i and therefore constitute elements of A_i or B_i . Let $x_1 \in \bar{A}_i$, then all correspondences from A_i or B_i that are related to a feature point of the local neighborhood from x_i (depicted by the bright red dots) are subject to ProSaC sampling.

object training, several other known and unknown objects are placed around the reference object in order to complicate recognition (Figure 6.7). By rotating the turn table by steps of 10° , 36 testing images showing the object in different orientations at a fixed distance are recorded. The procedure is executed for a close range of 0.7 m and a far range of 1.7 m. Therefore, the testing dataset for a single object consists of 2×36 images, resulting in a total of 1800 images over all 25 objects.

Evaluations are conducted with respect to the false positive rate, the true positive rate, the number of false negatives, the accuracy and the computation time. False positives denote recognition results, which are not reflected in the ground truth data or recognition results with a pose estimate that exceeds either a distance of 100 mm or exhibits an angle of more than 90° compared to the ground truth pose of an object. True positives denote recognition results that correspond to the ground truth data and that are within the bounds of 100 mm and 90° . False negatives are missed recognitions.

The target hardware platform is an Intel[®] Core[™] i7-2860QM with 2.5 GHz and 8 GB RAM. The computation time for object recognition has been measured for all 1800 images and results in an average value of 2.894 s when searching for a single object.

Accuracy is evaluated by comparing the rotational and the translational error independently. The translational error $e_t \in \mathbb{R}$ is given by (6.1), which is computed by the L_2 -norm between the translation vector $\hat{\mathbf{T}} \in \mathbb{R}^3$ from the object's ground truth pose $\hat{\mathbf{a}} = (\hat{\mathbf{R}}, \hat{\mathbf{T}})$ with



Figure 6.5.: 25 object models from the IPA dataset for textured object recognition.

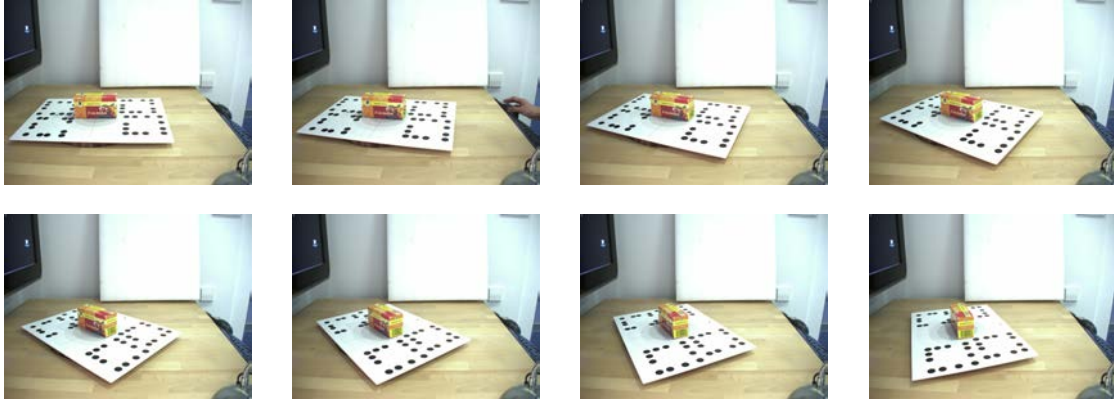


Figure 6.6.: Excerpt from the training set for object recognition.

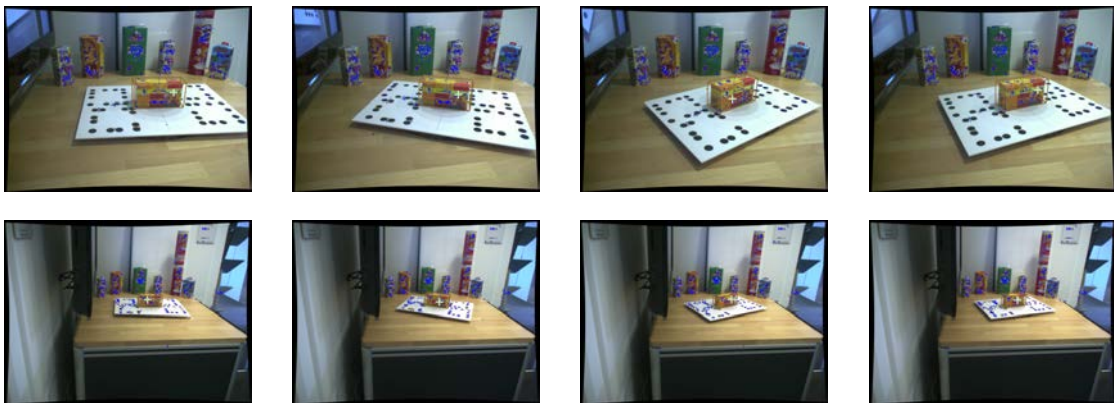


Figure 6.7.: Excerpt from the testing set for object recognition showing feature points (blue dots), the ground truth pose (coordinate system), and the recognized object pose (bounding box).

respect to the camera system and the translation vector $\mathbf{T} \in \mathbb{R}^3$ from the pose $\mathbf{a} = (\mathbf{R}, \mathbf{T})$ returned by the recognition result.

$$e_t = \|\hat{\mathbf{T}} - \mathbf{T}\| \quad (6.1)$$

By expressing the rotation with a 3×3 rotation matrix, the rotational error $e_r \in \mathbb{R}$ is measured according to (6.2). It computes the sum of the squares of the pair-wise matrix element differences between the ground truth rotation $\hat{\mathbf{R}} \in \mathbb{R}^{3 \times 3}$ and the rotation $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ returned by the recognition results using the Frobenius norm $\|\cdot\|_F$.

$$\begin{aligned} e_r &= \|\hat{\mathbf{R}} - \mathbf{R}\|_F^2 = \|\hat{\mathbf{R}}\|_F^2 - 2\text{tr}(\hat{\mathbf{R}}^T \mathbf{R}) + \|\mathbf{R}\|_F^2 \\ &= \text{tr}(\hat{\mathbf{R}}^T \hat{\mathbf{R}}) - 2(1 + 2 \cos(\theta)) + \text{tr}(\mathbf{R}^T \mathbf{R}) \\ &= \text{tr}(\mathbf{I}) - 2(1 + 2 \cos(\theta)) + \text{tr}(\mathbf{I}) \\ &= 3 - 2(1 + 2 \cos(\theta)) + 3 = 6 - 2(1 + 2 \cos(\theta)) \end{aligned} \quad (6.2)$$

Here, θ denotes the minimal angle of rotation between $\hat{\mathbf{R}}^T$ and \mathbf{R} , when expressing $\hat{\mathbf{R}}^T \mathbf{R}$ in axis-angle notation. The derivation is based on the fact, that for orthogonal matrices $\|\mathbf{R}\|_F^2 = \text{tr}(\mathbf{R}^T \mathbf{R})$ and $\text{tr}(\mathbf{R}) = 1 + 2 \cos(\theta')$. Instead of directly specifying $\|\cdot\|_F$ for each testing set, the more intuitive angle θ is evaluated.

Using the recognition of the marker in order to induce the ground truth pose of the object is justified by the fact, that markers are especially designed to be well recognized even under larger viewpoint changes. However, the accuracy measurements for object recognition are only valid up to the inaccuracies induced by the marker recognition. Therefore, at first an evaluation determining the accuracy of the marker recognition in rotation and translation has been conducted.

The applied marker pattern is visualized by Figure 6.6. The layout is based on the approach from Bergamasco et al. [BAT12], where a rectangular dot pattern is proposed. A single dot is located at each corner of the rectangle and on the line connecting two adjacent corners, another two dots are placed. The recognition and identification of a marker is based on the recognition of the dots and their relative position to each other. By elaborating the invariance of the cross-ratio under projective transformations, markers are distinguished by shifting the dots in between the pattern's corners to different positions.

The accuracy of the marker has been determined by artificially creating perspective projections of the marker pattern and by comparing the recognition results with the known transformation. The simulated test data has been generated for different camera distances ranging from 60 cm to 200 cm. For each distance 180 different viewpoints have been generated and evaluated. Furthermore, the tests have been conducted for two different Gaussian

blur kernels 5×5 and 15×15 pixel which have been applied to the artificially rendered images. The metrics to determine the rotational and translation error follow (6.2) and (6.1).

Figure 6.8 shows the rotational error of the pattern recognition with error bars indicating the standard deviation of the measurements. It is clearly visible, that the angular error does not exceed 2° , even for a larger blur kernel of 15×15 pixel. Figure 6.9 shows the translational error, again, with error bars indicating the standard deviation of the measurements. For all distances, the translational error stays below an average value of 0.1 mm. Even, when considering the standard deviation, this value is not exceeded.

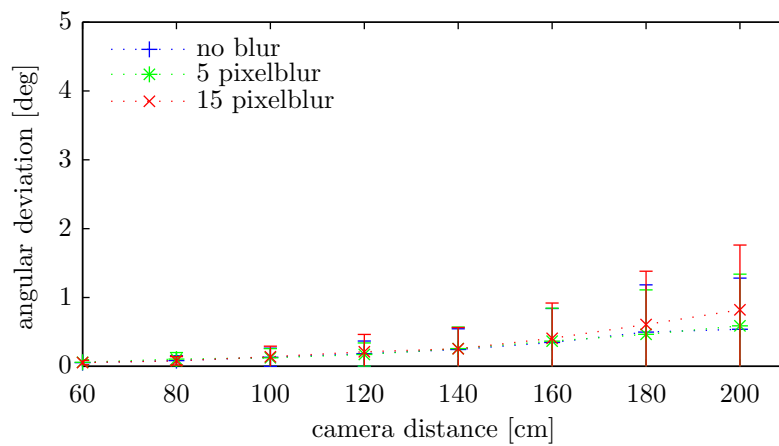


Figure 6.8.: Evaluation of the marker's rotational accuracy.

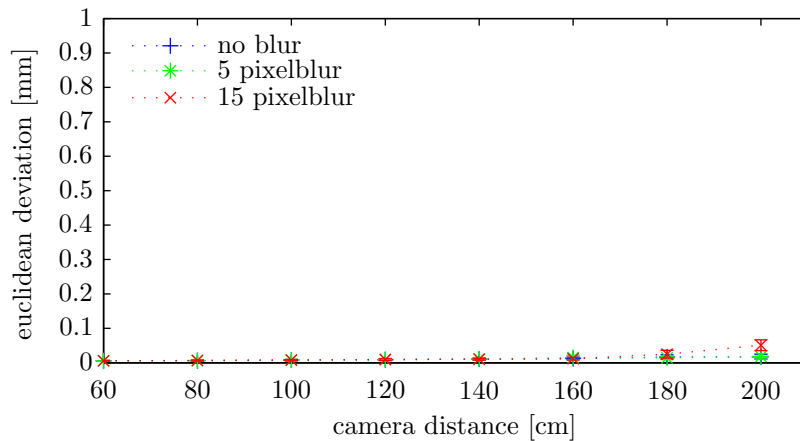


Figure 6.9.: Evaluation of the marker's translational accuracy.

The rotational accuracy for textured object recognition is given by Figure 6.10 for a distance of 0.7 m, by Figure 6.11 for a distance of 1.7 m, and by Figure 6.12 that summarizes the results at the two distances. At a distance of 0.7 m, the average rotational error of sORB is 2.24° , which is about 9% less compared to the average angular deviation of SURF that amounts to 2.46° .

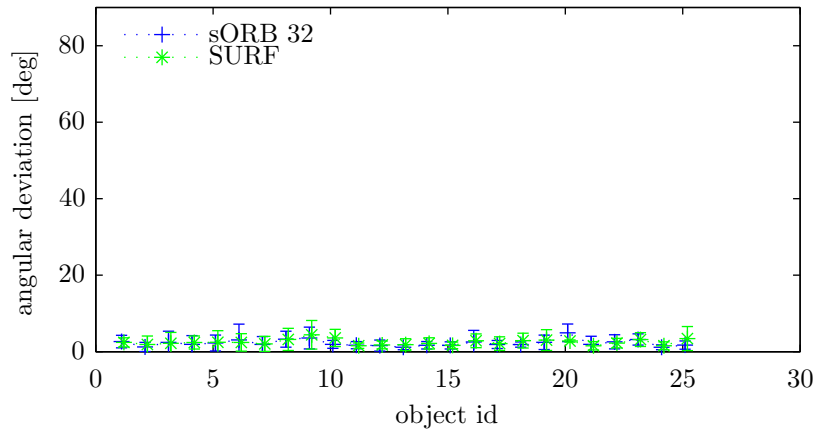


Figure 6.10.: Rotational accuracy for textured object recognition at a distance of 0.7 m.

At a distance of 1.7 m, the performance of both descriptors drops to an average angular deviation of 4.44° for the sORB descriptor and a value of 4.43° for the SURF descriptor. This is intuitively clear, as the training data has been recorded at a distance of about 1.0 m and both descriptors rely on their ability to cope with scale changes in order to successfully recognize and associate the feature points. The slightly worse performance of the sORB descriptor, however, is due to the fact, that only sORB is able to recognize object 18 and object 19 at a distance of 1.7 m. The recognition accuracy of object 18, however, is rather poor compared to all other objects and therefore decreases the overall recognition accuracy of sORB. When using the SURF descriptor, it is not possible to recognize these objects and therefore they also do not influence the accuracy results of SURF.

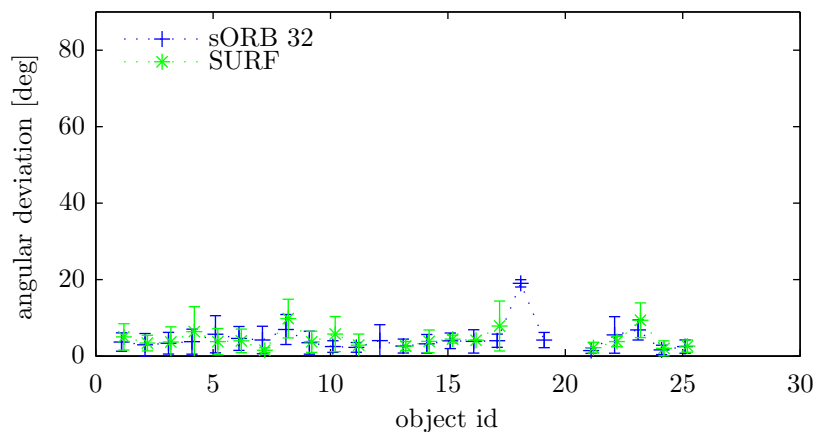


Figure 6.11.: Rotational accuracy for textured object recognition at a distance of 1.7 m.

When summarizing the performance for both distances over the complete dataset, sORB achieves an average angular deviation of 2.68° and SURF an average angular deviation of 2.8° .

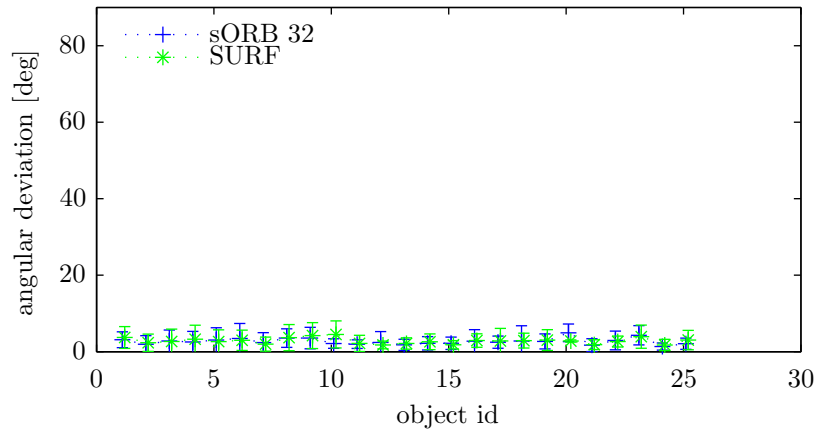


Figure 6.12.: Rotational accuracy for textured object recognition at a distance of 0.7 m and 1.7 m.

The translational accuracy for textured object recognition is given by Figure 6.13 for a distance of 0.7 m, by Figure 6.14 for a distance of 1.7 m, and by Figure 6.15 that summarizes the results at the two distances. At a distance of 0.7 m, the average Euclidean deviation for sORB is 2.58 mm, which is about 8.5% less compared to the average Euclidean deviation of SURF with 2.82 mm.

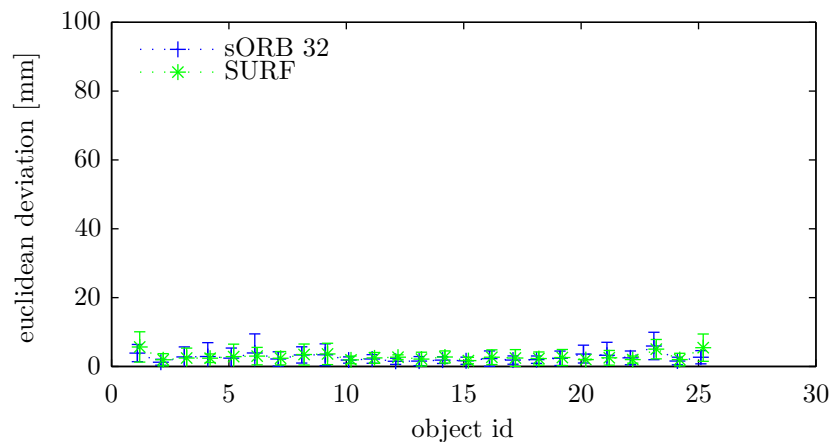


Figure 6.13.: Translational accuracy for textured object recognition at a distance of 0.7 m.

The translational error increases, when moving the object 1.7 m away from the camera. On average, sORB achieves an accuracy of 4.83 mm, whereas SURF slightly performs better with an average accuracy of 4.67 mm. Similar to the rotational error, the reason for the slightly worse accuracy of sORB is founded in its higher descriptive abilities as shown in Section 5.2.5. Only sORB is able to detect object 18 and object 19, from which predominantly object 18 negatively influences the accuracy of sORB.

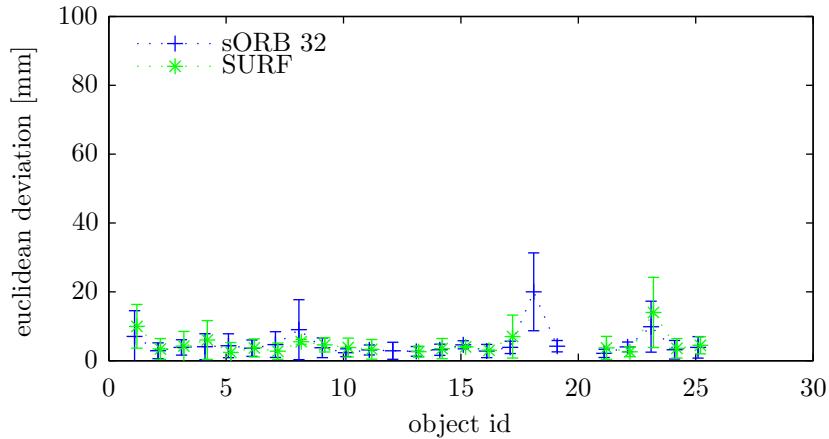


Figure 6.14.: Translational accuracy for textured object recognition at a distance of 1.7 m.

When accumulating the scores over both distances, sORB achieves an average accuracy in measuring an object’s position of 3.02 mm, whereas the SURF descriptor achieves an average accuracy of 3.17 mm. The average angular and Euclidean deviation results for different distances are summarized by Table 6.1.

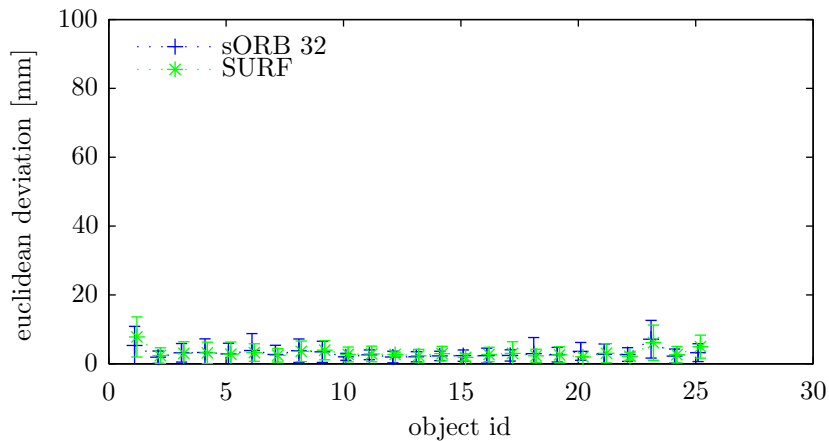


Figure 6.15.: Translational accuracy for textured object recognition at a distance of 0.7 m and 1.7 m.

The number of true and false positives is given by Table 6.2 for a distance of 0.7 m and 1.7 m. At a distance of 0.7 m, sORB is able to recognize 783 out of 900 objects, which amounts to a true positive rate of 87%. This is more than 7 percentage points better compared to SURF and expresses the higher descriptive power of sORB, which is also visible in its ability to recognize object 18 and object 19. It is evident, that the number of true positives drops when increasing the distance to the object to 1.7 m. Here, sORB still outperforms SURF by 4 percentage points, however, both descriptors are no longer able to achieve a true positive rate larger than 44%.

	sORB	SURF
Avg. angular deviation (0.7 m)	2.24°	2.46°
Avg. angular deviation (1.7 m)	4.44°	4.43°
Avg. angular deviation (0.7 m and 1.7 m)	2.68°	2.8°
Avg. Euclidean deviation (0.7 m)	2.58 mm	2.82 mm
Avg. Euclidean deviation (1.7 m)	4.83 mm	4.67 mm
Avg. Euclidean deviation (0.7 m and 1.7 m)	3.02 mm	3.17 mm

Table 6.1.: Average Euclidean and angular deviation of the error in position and orientation when using SURF and sORB for textured object recognition.

Besides the true positive rate, it is even more important to keep the number of false positives to a minimum in order to prevent the robot from grasping wrong or non-existing objects. Here, sORB is able to achieve a false positive rate below 1.6% for both distances, whereas the false positive rate of SURF reaches a value of 6% at a distance of 0.7 m.

	sORB	SURF
True positives (0.7 m)	783 (87.00%)	717 (79.67%)
True positives (1.7 m)	389 (43.22%)	352 (39.11%)
True positives (0.7 m and 1.70 m)	1172 (65.11%)	1069 (59.39%)
False positives (0.7 m)	15 (1.60%)	54 (6.00%)
False positives (1.7 m)	12 (1.34%)	16 (1.78%)
False positives (0.7 m and 1.70 m)	27 (1.50%)	70 (3.89%)

Table 6.2.: Comparing true positives and false positives for SURF and sORB on the IPA dataset consisting of 1800 different images at a distance of 0.7 m and 1.7 m.

Invariance against occlusion is naturally given by using local point feature matching for object recognition. On the extreme, it is sufficient to have three visible feature points that can be matched with the object model in order to determine an object’s pose. However, evaluations on occlusion would strongly depend on which part of an object is visible or covered. When an object exhibits only a view feature points, covering those would prevent its detection immediately. On the other hand, as long as those feature points are visible the object would be successfully recognized. Therefore, it has been decided to not evaluate the invariance against occlusion empirically, as its results would be of little significance. Figure 6.16 gives an example of the resulting recognition quality.

6.3. Recognition of texture-less objects

Point features fail to capture the shape and appearance of an object with less or any texture. Therefore, this thesis uses a global histogram-based descriptor for the distinct description of texture-less objects as proposed in Section 5.3. For object recognition, the global descriptions of the texture-less object models are compared against global descriptors



Figure 6.16.: Recognition results with colored boxes indicating the detected pose of an object and the object type.

from the scene image using a novel adaptive sliding window approach. The matching results are captured and aggregated within a voting map in order to select the best fitting object locations.

6.3.1. Adaptive sliding window for data association

In general, the sliding window approach shifts a window of fixed size row-by-row across the scene image and considers only the image area which is covered by the window for descriptor computation. The descriptor given by the sliding window is compared against all descriptors of all models and a matching with a descriptor distance closer than a predefined threshold is considered valid. Without resizing the sliding window, invariance to varying scales is not given. An object covers more image area the closer its distance is to the camera. Conversely, the image area covered by an object is decreasing with an increasing distance to the camera. Therefore, it is common practice to repeatedly shift the sliding window across the scene image while altering its dimensions after each run. This, however, directly increases computation time which should be avoided. The thesis proposes an adaptive sliding window approach that requires only a single run of the sliding window across the scene image.

The basic idea of the proposed approach is to take advantage of the range data given by the camera setup. With the knowledge of an object's dimensions and the knowledge of the range value for the currently considered image pixel at the center of the sliding window, the sliding window's dimensions may be adapted from shift to shift, accordingly e.g. when the area covered by the sliding window moves further away from the camera, the dimensions of the sliding window will decrease (Figure 6.17(b)). Vice versa, when the sliding window gets closer to the camera, the dimensions of the sliding window will increase (Figure 6.17(a)).

The distance of the sliding window to the camera is given by the range value d_p of the pixel p located at the center of the sliding window. The width and height of the sliding

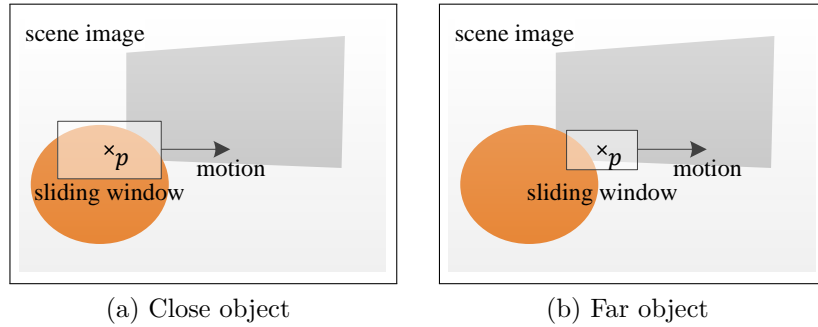


Figure 6.17.: Principle of the adaptive sliding window approach.

window is then multiplied by the scale factor $s(d_p)$ defined by (6.3), where \hat{d}_p denotes the reference distance at which the training data has been recorded.

$$s: \mathbb{R} \rightarrow \mathbb{R}$$

$$d_p \mapsto s(d_p) = \frac{\hat{d}_p}{d} \quad (6.3)$$

The sliding window approach is well suited for the proposed descriptor, as its rectangular shape directly enables the application of integral images for the global descriptor computation as explained in Section 5.3.3. This enables a fast computation of the global descriptor value by computing the histogram of local descriptors over the area covered by the sliding window with only four operations.

6.3.2. Voting map for pose estimation

Without limiting the generality of the foregoing, it is assumed that only a single object is searched for. Then, the sliding window approach results in one closest descriptor distance d_v for each image pixel \mathbf{p} , which corresponds to the best matching object appearance. In order to decide for a matching object and its pose, all descriptor distances which are smaller than a predefined threshold are stored into a data structure termed *voting map*. A visual impression of the voting map is given by Figure 6.18(b) which corresponds to the scene image shown by Figure 6.18(a).

The voting map is of the same size as the scene image. In order to decide whether an object is present within the scene image, a non-minimum suppression algorithm is applied on the voting map to extract the local minimum from the voting map. The local minimum corresponds to the minimal descriptor distance and it is marked as a valid object location, when not being larger than a predefined threshold. The non-minimum suppression step is implemented using dilation and erosion operators from morphological image processing.

The decision on the best fitting object pose is computed by taking the 3×3 neighborhood of the valid object locations and aggregating the pose information into a mean pose vector.

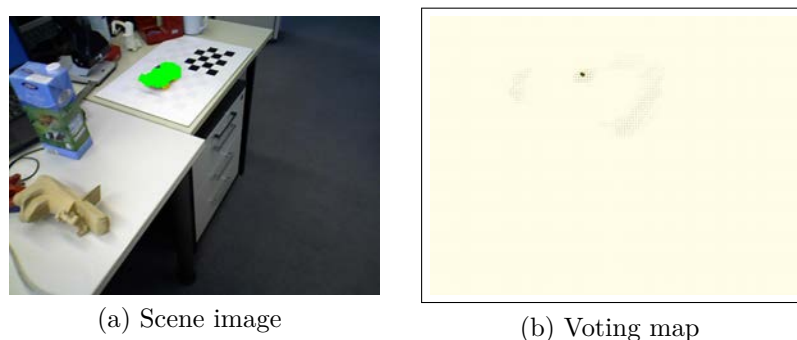


Figure 6.18.: Recognition of the *Car* object using the proposed voting map representation for pose estimation.

6.3.3. Evaluation

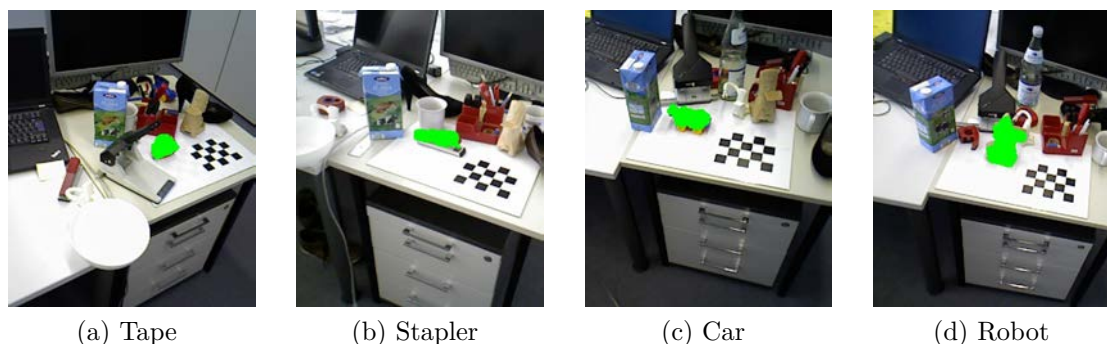


Figure 6.19.: Images taken from the testing dataset for texture-less object recognition. The ground truth pose of the object is given by the checkerboard in order to evaluate the recognition rate.

Threshold parameters, recall-precision plots and the distribution of L2 descriptor distances with respect to texture-less object recognition have been determined based on a dataset of four different objects. Each dataset consists of 200 training images and 1000 testing images recorded from strongly varying viewpoints. The ground truth position of the object is given by a checkerboard that is placed at a fixed position relative to the object. Similar to the evaluations from Section 6.2.4, false positives denote recognition results which are not reflected in the ground truth data or recognition results with a pose estimate that exceeds either a distance of 100 mm or exhibits an angle of more than 90° compared to the ground truth pose of an object. True positives denote recognition results that correspond to the ground truth data and that are within the bounds of 100 mm and 90° . False negatives are missed recognitions. Images from the training and testing dataset are shown by Figure 5.22 and Figure 6.19.

The definition of the true positive rate ρ_{tpr} , which is also termed *recall*, and the *precision* ρ_{pr} is given in (6.4).

$$\rho_{\text{tpr}} = \frac{h_{\text{tp}}}{h_{\text{tp}} + h_{\text{fn}}} \quad \rho_{\text{pr}} = \frac{h_{\text{tp}}}{h_{\text{tp}} + h_{\text{fp}}} \quad (6.4)$$

Here, we denote the number of true positives h_{tp} , the number of false positives h_{fp} and the number of false negatives h_{fn} . The recall expresses the relative number of correctly recognized objects in relation to the total number of all recognizable objects e.g. a recall of 0.82 means that 82% of all objects have been recognized. The precision expresses the relative number of objects that have been correctly recognized among all recognition results e.g. a precision of 0.91 means that only 91% of all recognized objects are actually correct. Both values are summarized in the recall-precision plot that ideally achieves a recall and a precision of 1.

At first an optimal parameter of t_{2D} from (5.18) has been determined by computing and evaluating the recall-precision curves for different values of t_{2D} on the *Car* dataset. Figure 6.20 gives the results for values ranging from 6 to 70 pixel. It clearly shows, that the

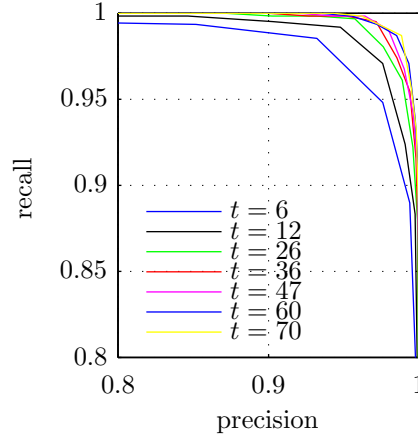


Figure 6.20.: Close-up view on the recall-precision plot for different values of t_{2D} from (5.18).

performance gradually increases with increasing values of t_{2D} until an optimal value of 70 is reached. The performance does not further improve for larger values of t_{2D} , therefore $t_{2D} = 70$ has been used for all following evaluations. The same evaluation has been conducted for t_{3D} from (5.23) and (5.24) based on which a value of $t_{3D} = 0.002$ m at a distance of 0.5 m has been selected.

Furthermore, the distribution of L2 descriptor distances for positive and negative samples from the testing dataset compared to their closest matching descriptor from the training dataset is evaluated for the different objects. Results are given in the left column of Figure 6.21, whereas the right column shows the resulting recall-precision plot for varying L2

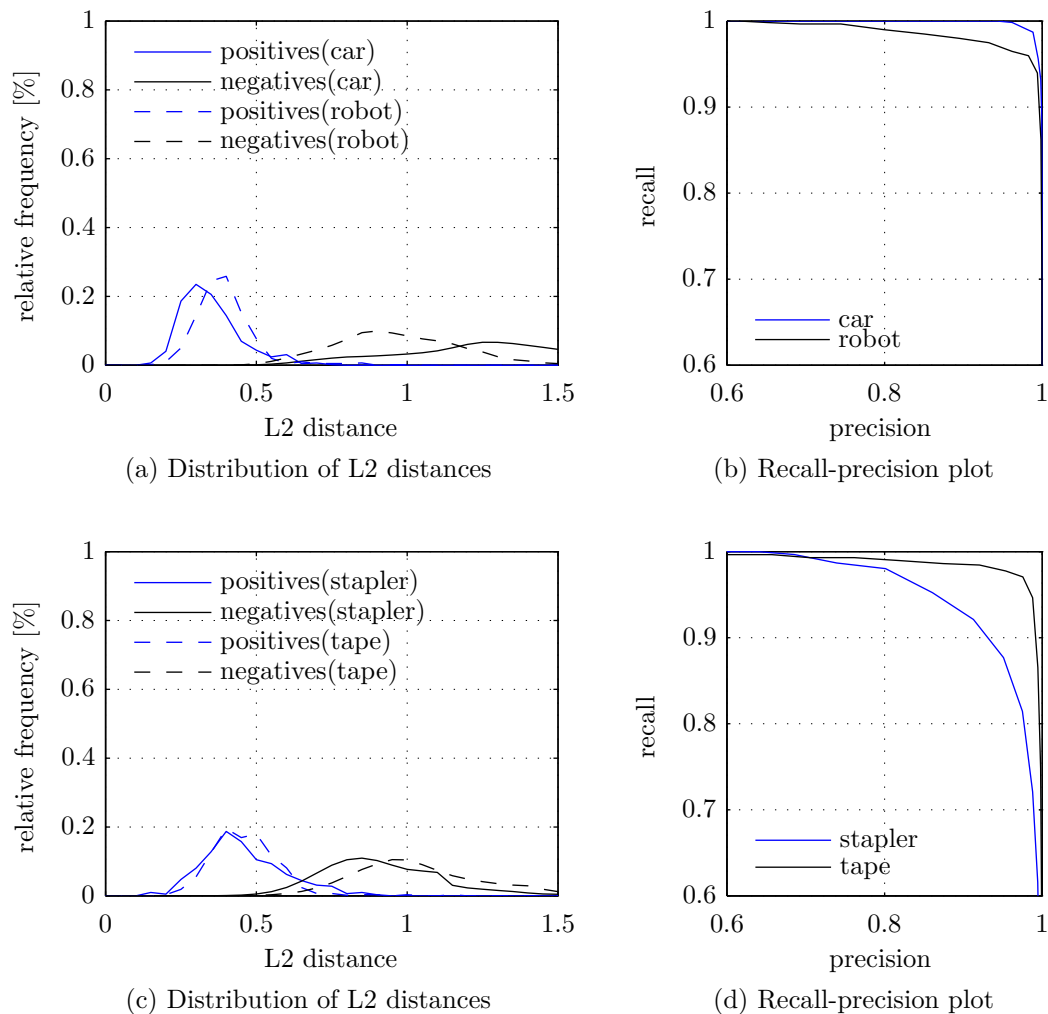


Figure 6.21.: Distribution of L2 distances (first column) and recall-precision plots (second column) for different objects.

matching thresholds. The distribution of the L2 distances is approximately bell-shaped and clearly shows that a separation of positive and negative samples based on thresholding on the L2 distance is feasible. Referring to the recall-precision plot the car, robot and tape object reach a precision of more than 99% while having a recall rate above 95%, when choosing an appropriate L2 distance threshold. This means, that we can recognize 95% of all objects when accepting a false positive rate of 1%. Due to its more common shape, the performance of the stapler object is slightly worse compared to the other objects. However, it still achieves a recall rate of 74% for a precision of 99%.

Similar to the evaluation of the textured object recognition (Section 6.2.4), the translational and rotational accuracy of the proposed method for texture-less object recognition has been tested on a dataset consisting of 25 different objects (Figure 6.22) using the dot-pattern marker board for ground truth inference. The rotational accuracy for texture-less



Figure 6.22.: 25 object models from the IPA testing dataset for texture-less object recognition.

object recognition is given by Figure 6.23 for a distance of 0.7 m, by Figure 6.24 for a distance of 1.7 m, and by Figure 6.25 that summarizes the results at the two distances.

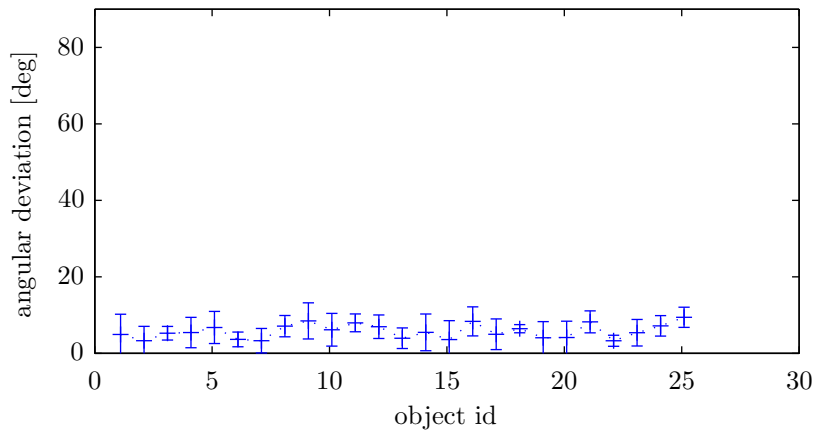


Figure 6.23.: Rotational accuracy for texture-less object recognition at a distance of 0.7 m.

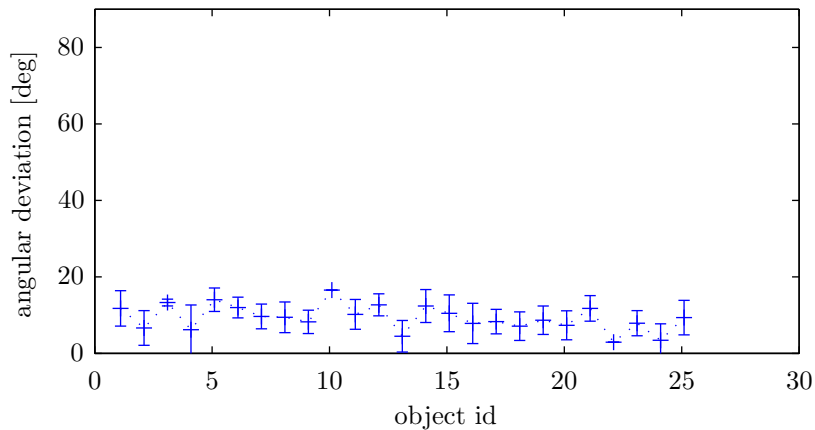


Figure 6.24.: Rotational accuracy for texture-less object recognition at a distance of 1.7 m.

At a distance of 0.7 m, the average rotational error of texture-less object recognition is 5.74° . At a distance of 1.7 m, the performance of the descriptors drops to an average angular deviation of 9.76° . This is intuitively clear, as the projection of the object on the camera plane becomes smaller when increasing the distance between camera and object. Therefore, also the descriptor has less data to capture the object, which decreases its descriptive power. When summarizing the performance for both distances over the complete dataset, the texture-less object recognition achieves an average angular deviation of 6.94° .

The translational accuracy for texture-less object recognition is given by Figure 6.26 for a distance of 0.7 m, by Figure 6.27 for a distance of 1.7 m, and by Figure 6.28 that summarizes the results at the two distances.

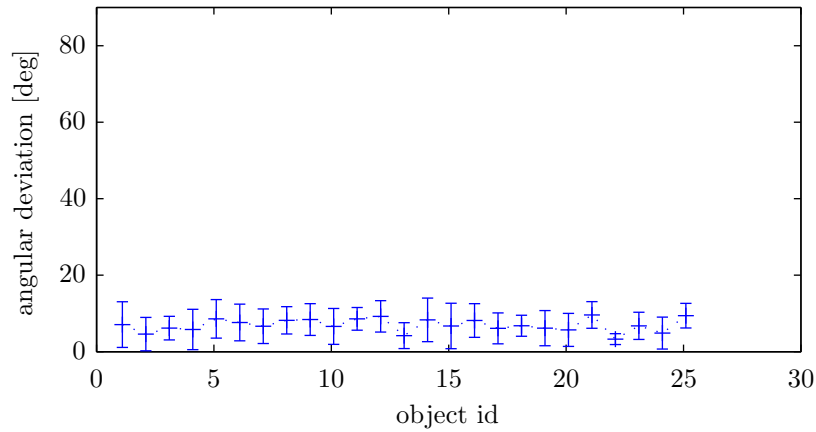


Figure 6.25.: Rotational accuracy for texture-less object recognition at a distance of 0.7 m and 1.7 m.

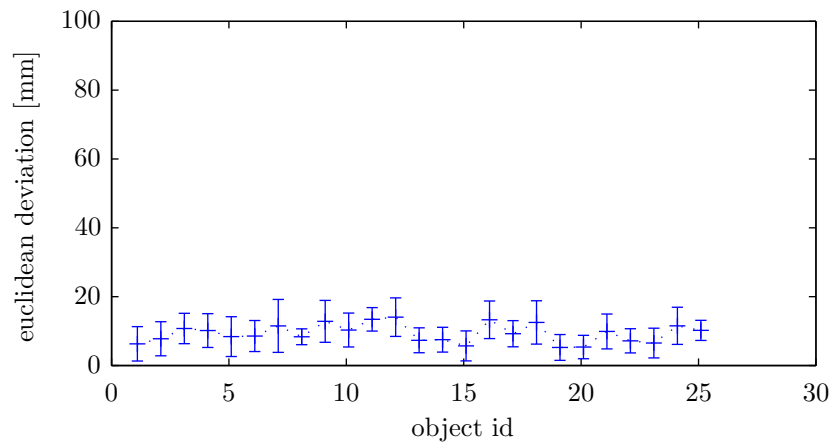


Figure 6.26.: Translational accuracy for texture-less object recognition at a distance of 0.7 m.

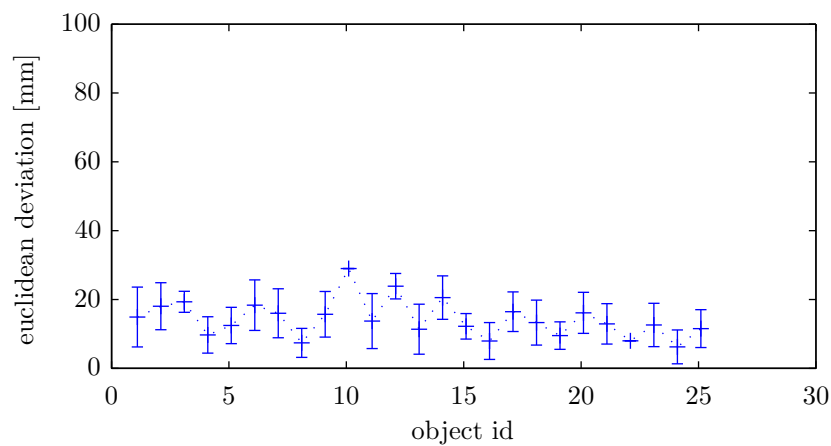


Figure 6.27.: Translational accuracy for texture-less object recognition at a distance of 1.7 m.

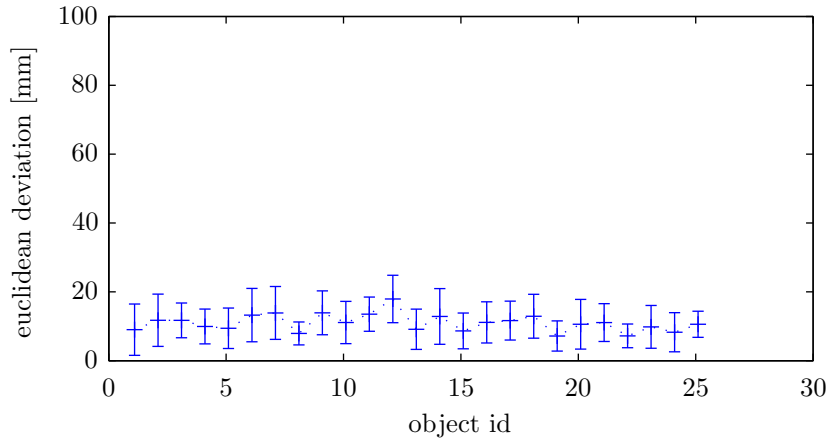


Figure 6.28.: Translational accuracy for texture-less object recognition at a distance of 0.7 m and 1.7 m.

At a distance of 0.7 m, the average Euclidean deviation for texture-less object recognition is 9.39 mm. The translational error increases, when moving the object further away from the camera. On average, recognition achieves an accuracy of 14.73 mm at a distance of 1.7 m. When accumulating the scores over both distances, texture-less object recognition achieves an average Euclidean deviation of 10.96 mm.

The average angular and Euclidean deviation results for different distances are summarized by Table 6.3.

Avg. angular deviation (0.7 m)	5.74°
Avg. angular deviation (1.7 m)	9.76°
Avg. angular deviation (0.7 m and 1.7 m)	6.94°
Avg. Euclidean deviation (0.7 m)	9.39 mm
Avg. Euclidean deviation (1.7 m)	14.73 mm
Avg. Euclidean deviation (0.7 m and 1.7 m)	10.96 mm

Table 6.3.: Average Euclidean and angular deviation of the error in position and orientation for texture-less object recognition.

The number of true positives and false negatives on the IPA dataset is given by Table 6.4 for a distance of 0.7 m and 1.7 m. At a distance of 0.7 m, the texture-less object recognition is able to recognize 707 out of 900 objects, which amounts to a true positive rate of 78.56%. The number of true positives drops to a value of 50.44% when increasing the distance to the object to 1.7 m.

The design of the descriptor for texture-less object recognition is based on the global appearance of an object. Therefore, invariance against partial occlusion is limited as the descriptor rapidly changes its shape when the object is no more completely visible. Table 6.5 shows how the recognition performance drops with an increasing degree of occlusion. Up to 30% occlusion the number of true positives stays at an acceptable value of 80%. At

True positives (0.7 m)	707 (78.56%)
True positives (1.7 m)	454 (50.44%)
True positives (0.7 m and 1.7 m)	1161 (64.50%)
False negatives (0.7 m)	193 (21.44%)
False negatives (1.7 m)	446 (49.56%)
False negatives (0.7 m and 1.7 m)	639 (35.50%)

Table 6.4.: True positive and false negative rate on the IPA dataset consisting of 1800 different images at a distance of 0.7 m and 1.7 m.

40% occlusion, however, the relative amount of true positives significantly decreases to a value of 18.52% and by further increasing the occlusion, the object cannot be recognized any longer. The tests have been conducted on 216 newly recorded test images from object 1 from the IPA dataset by artificially coloring the specified percentage of image pixel from the test object in black.

occlusion	true positives(absolute/relative)
0%	216 (100%)
10%	216 (100%)
20%	215 (99.53%)
30%	175 (81.02%)
40%	40 (18.52%)
50%	0 (0%)

Table 6.5.: Absolute and relative number of true positives with respect to a varying degree of occlusion for texture-less object recognition.

The target hardware platform is an Intel[®] Core[™] i7-2860QM with 2.5 GHz and 8 GB RAM. The computation time for texture-less object recognition has been measured for all 1800 images and results in an average value of 5.325 s when searching for a single object. The most time is spend to match the large amount of descriptors when applying the adaptive sliding window to search for the object. Figure 6.29 gives an example of the resulting recognition quality.



Figure 6.29.: Recognition results shown by colored projections of the best fitting object view onto the scene image

7. Conclusion and Outlook

This thesis has broken down the problem of object recognition into the three steps of (1) data acquisition and sensor fusion, (2) object modeling, and (3) object recognition and localization. For each step, individual contributions have been presented and evaluated according to the requirements listed in Section 1.4.

The system design described in Section 3.4 explicitly enables the integration of the proposed perception framework with the hardware of a service robot as well as its usage in a stand-alone vision system. It has been implemented on the service robot Care-O-bot[®] 3 and on different stand-alone hardware setups using a turn table or marker patterns for object modeling. The proposed feature descriptors and the software algorithms for pose estimation have been evaluated with respect to arbitrary object poses, occlusion (Section 6.2.4 and Section 6.3.3) and varying lighting conditions (Section 5.2.5). All software components are embedded in a modular, decomposable software architecture that enables the independent application of the individual components for data acquisition, object modeling and object recognition. Table 7.1 summarizes the results in correspondence to the requirements.

ID	Criteria	Requirement	Fulfilled
R1	Target platform	Service robot and stand-alone vision system	Yes
R2	Environment conditions	Indoor, including arbitrary object poses, occlusion and varying lighting conditions	Yes
R3	Software architecture	Modular software architecture that enables an independent application of the data acquisition, object modeling and object recognition modules	Yes

Table 7.1.: Validation of the system requirements.

Concerning data acquisition and sensor fusion, the 3D information from a range camera and a stereo camera system have been combined in a common coordinate system to create a denser disparity map with higher resolution and accuracy than could be obtained with a single sensor source alone. Evaluations have been conducted on a real world data set and on the standard Middlebury stereo data (Section 4.7). It has been shown, that sensor

fusion improves the results from stereo vision especially in structureless areas where stereo vision is prone to errors. The algorithm achieves a pixel density of 93.1% and an average accuracy of 0.017 m at distances ranging from 0.7 m to 1.70 m. The computation time on a standard PC setup for images in VGA resolution is 10 Hz. The results are summarized by Table 7.2.

ID	Criteria	Requirement	Fulfilled
R5	Output	Providing range and color information	Yes
R6	Density	Retrieving 3D data for at least 90% of all image pixel within the field of view of the camera setup	Yes
R7	Measurement range	70 cm to 170 cm	Yes
R8	Accuracy	Achieving an average measurement error below 2 cm within the measurement range	Yes
R9	Computation time	5 Hz at VGA resolution	Yes
R10	Sensor layout	Compact and mountable on a mobile robot	Yes

Table 7.2.: Validation of the requirements for data acquisition.

In the area of object modeling, the thesis distinguishes between textured and texture-less objects. Concerning textured objects, it proposes a scale invariant extension of the binary feature descriptor ORB, termed sORB. The performance of the sORB descriptor is evaluated against the well known SIFT and SURF descriptors as well as the original ORB descriptor. Based on the dataset from the Oxford’s Visual Geometry Group, it is shown, that the sORB descriptor outperforms the mentioned descriptors while being computed faster. The feature point is used to create dense object models of typical household objects by combining single object images from different viewpoints using bundle adjustment. The performance of bundle adjustment in terms of the reconstruction accuracy is below 3 mm, which has been evaluated using a simulation environment (Section 5.2.5). Concerning texture-less objects, the thesis proposed a global object descriptor based on histograms of 2D and 3D gradients. The feature descriptor is explicitly designed to make use of dynamic programming which enables its fast computation. The single object images, which are recorded during object modeling, are combined into a common object model by directly using the inferred odometry data from the rotating marker board or the robot’s gripper movement. Therefore, R12 is only partially valid for texture-less object modeling, as it solely depends on the accuracy of the measured odometry data. One focus has been put on usability of the modeling procedure. Therefore, three methods have been proposed to model the object either autonomously by the robot, autonomously by a stand-alone turn table, or semi-autonomously by manually moving the camera around the object.

ID	Criteria	Requirement	Fulfilled
R11	Output	Dense 3D model enabling the computation of suitable grasps	Yes
R12	Accuracy	Model accuracy below an overall average error of 3 mm compared to the ground truth data	(Yes)
R13	Object dimensions	Not larger than 10 cm × 10 cm × 30 cm and not smaller than 1 cm × 1 cm × 2 cm (width×length×height)	Yes
R14	Object type	Rigid objects	Yes
R15	Usability	Easy introduction of new objects by non-expert users or the robot itself	Yes

Table 7.3.: Validation of the requirements for object modeling.

Object recognition and localization operates on individual images to localize the 6 DoF pose of rigid objects in a range of 70 cm to 170 cm. Similar to object modeling, it is distinguished between the recognition of textured and texture-less objects. Concerning textured objects, recognition is model-based by fitting 3D object models into the scene image. The fitting is based on the association of feature points from the object models with feature points from the scene image. The performance of the proposed method has been evaluated based on a set of 25 objects (Section 6.2.4). On average, the proposed method achieves an angular accuracy of 2.68° and Euclidean accuracy of 3.02 mm at a distance of 70 cm to 170 cm, and the computation time for a single recognition amounts to 2.894 s.

Concerning the recognition of texture-less objects, the global descriptor used for object modeling is matched against individual image parts using an adaptive sliding window approach. Similar to the recognition of textured objects, the accuracy of the proposed method is evaluated on a set of 25 objects (Section 6.3.3). On average, the algorithm achieves an angular accuracy of 5.74° and an Euclidean accuracy of 9.39 mm, and the computation time for a single recognition amounts to 5.325 s.

Considering future work, one could address the combination of the two methods for the recognition of textured and texture-less objects into a common probabilistic framework. At the moment, the user has to decide for either of the two methods during the modeling phase. However, for objects consisting of partially textured and texture-less areas a combination of both methods would be beneficial compared to the application of a single method alone. Therefore, a promising approach could be the assignment of belief values to each of the recognition methods' pose estimates in order to combine them to a single pose. Another approach could tackle the combination of both methods on a feature point level, e.g. by using the local and global descriptors together to model the object.

ID	Criteria	Requirement	Fulfilled
R16	Output	6 DoF pose of the object	Yes
R17	Object types	Rigid textured and texture-less objects	Yes
R18	Number of objects	Recognition of 25 textured and 25 texture-less objects	Yes
R19	Recognition range	70 cm to 170 cm	Yes
R20	Recognition time	Processing of a scene image in less than 5 s	Yes
R21	Recognition accuracy	Average angular deviation below 6° and Euclidean deviation below 9 mm at a distance of 0.7 m to the object	Yes

Table 7.4.: Validation of the requirements for object recognition.

Future work could also address the combination of several camera images in order to track the object across different views. This would be beneficial in order to improve the localization of an object as recognition could be performed on different viewpoints instead of a single one alone e.g. most objects consist of unique parts that simplify recognition and improve localization when being seen. Additionally, the localization could be refined across the objects or, when being applied on a robot, the camera could be actively guided to move to the most promising viewpoint in order to enable recognition.

A. Glossary

Correspondence problem Refers to the problem of matching a set of features (e.g. from an object model) with another set of features to find corresponding pairs. The matching distance is computed based on the feature descriptors.

Data association Associating sensor measurements to a set of possible entities (e.g. feature points or objects).

Disparity Disparity is related to stereo vision and originates from the horizontal displacement of the two cameras. It describes the relative displacement of the two projections of a 3D point on two image planes and it is measured in pixel.

Extrinsic parameters Extrinsic parameters describe the position and orientation of the individual camera systems within a common coordinate system. Related to stereo vision, extrinsic parameters describe the relative position of the two color cameras to each another.

Feature descriptor An algorithm to distinctly describe the local neighborhood of \rightarrow feature points.

Feature detector An algorithm to locate feature points in 2D or 3D space.

Feature point A distinct well-defined point in 2D or 3D space that has a local neighborhood which is rich in information content to enable its later recognition. The location of the feature point is, to some extent, stable under various influences like perspective transformations or illumination and brightness variations.

Intrinsic parameters Intrinsic parameters characterize the camera system and describe the mapping from a point in 3D camera coordinates to its projection to 2D image coordinates measured in pixel. Intrinsic parameters describe the focal length, radial and tangential distortion, the principle point and the skew coefficient.

LIDAR An optical sensing technology to measure distance information by using active illumination.

Object detection Determines the presence or absence of an instance from a certain class (e.g. bottle) within the stimulus (e.g. 2.5D image data).

Object localization Follows object detection or recognition to determine accurate location information for each object, i.e. object O is at position (x, y) within the image.

Object model An abstract representation of an object's properties like shape and texture.

Object recognition Determines the presence or absence of a specific object (e.g. a bottle from a specific company) within the stimulus (e.g. 2.5D image data).

One-Shot-Procedure The term relates to hardware or algorithms that operate on data recorded at a single point in time.

Pose estimation problem Refers to the problem of estimating a pose that is consistent with the solutions of the \rightarrow correspondence problem.

Range imaging Refers to a family of imaging techniques providing not only color information, but also spatial distance information for each image pixel.

RGB-D camera An RGB-D camera is capable of producing red, green, and blue (RGB) color information as well as distance (D) information for each pixel.

RGB-D image Each image pixel is assigned red, green, and blue (RGB) color information as well as distance (D) information.

SLAM A technique to build up a consistent map of an unknown environment by jointly estimating a robot's position and a mapping of its environment into a common coordinate system.

Rotation invariance A feature descriptor is called rotation invariant, if it is invariant to rotations around the camera's optical axis. Such a behavior is desirable in the case of object recognition, as the relative pose of objects to the camera is unknown.

Scale invariance A feature descriptor is called scale invariant if it is invariant to translations along the camera's optical axis. Such a behavior is desirable in the case of object recognition as the relative pose of objects to the camera is unknown.

Stereo vision The term stereo vision denotes the perception of depth from digital images by perceiving the environment from two distinct points. Depth is inferred using \rightarrow triangulation by taking advantage of the camera sensors' spatial displacement.

Triangulation Triangulation relates to the fact, that a point in 3D space is not measured directly, but rather indirectly by its projections on the 2D image planes of two or more recording camera sensors. Common sensor setups include two color cameras that are horizontally displaced. Depth is inferred based on point correspondences across the image pairs and the knowledge of the cameras' relative position to each other (\rightarrow extrinsic parameters) and their projection rules (\rightarrow intrinsic parameters).

Bibliography

- [AFV10] Arbeiter, Georg, Fischer, Jan, and Alexander Verl, 2010. 3D perception and modeling for manipulation on Care-O-bot[®] 3. In: *IEEE International Conference on Robotics and Automation (ICRA)*. May 3-8, 2010. Anchorage, AK. Piscataway: IEEE Press, 5 pp.
- [AKB⁺08] Agrawal, Motilal, Konolige, Kurt, Morten Blas, et al., 2008. CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching. In: *Lecture Notes in Computer Science*, 5305, Berlin, Heidelberg: Springer, ISBN 978-3-540-88692-1, pp. 102–115
- [BAT12] Bergamasco, Filippo, Albarelli, Andrea, and Andrea Torsello, 2012. Pi-Tag: a fast image-space marker design based on projective invariants. *Machine Vision and Applications*, **24**(6), pp. 1295–1310, ISSN 0932-8092
- [BBP⁺09] Bebis, George, Boyle, Richard, Bahram Parvin, et al., 2009. Dense Depth Maps from Low Resolution Time-of-Flight Depth and High Resolution Color Views. In: *Advances in Visual Computing*, 5876, Berlin, Heidelberg: Springer, ISBN 978-3-642-10519-7, pp. 228–239
- [Ben75] Bentley, Jon Louis, 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, **18**(9), pp. 509–517, ISSN 0001-0782
- [BFG⁺11] Browatzki, Björn, Fischer, Jan, Birgit Graf, et al., 2011. Going into depth: Evaluating 2D and 3D cues for object classification on a new, large-scale object dataset. In: *IEEE International Conference on Computer Vision Workshops (ICCV)*. November 6-13, 2011. Barcelona, Spain. Piscataway: IEEE Press, pp. 1189–1195
- [BG05] Bleyer, Michael and Gelautz, Margrit, 2005. A layered stereo matching algorithm using image segmentation and global visibility constraints. *ISPRS Journal of Photogrammetry and Remote Sensing*, **59**(3), pp. 128–150

- [Bou08] Bouguet, Jean Yves, 2008, Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/, accessed: December 30th, 2013
- [Bra00] Bradski, Gary, 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, **25**(11), pp. 120–126
- [BRCV12] Barzigar, Nafise, Roozgard, Aminmohammad, Samuel Cheng, et al., 2012. SCoBeP: Dense image registration using sparse coding and belief propagation. *Journal of Visual Communication and Image Representation*, **24**(2), pp. 137–147, ISSN 1047-3203
- [BT98] Birchfield, Stan and Tomasi, Carlo, 1998. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, **35**(3), pp. 1405–1573
- [BTG08] Bay, Herbert, Tuytelaars, Tinne, and Luc Van Gool, 2008. Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, **110**(3), pp. 346–359
- [Car02] Carlson, Bradley, 2002. Comparison of modern CCD and CMOS image sensor technologies and systems for low resolution imaging. In: *IEEE Sensors*. June 12-14, 2002. Holtsville, NY. Piscataway: IEEE Press, ISBN 0-7803-7454-1, pp. 171–176
- [Cas10] Cassisa, Cyril, 2010. Local vs global energy minimization methods: Application to stereo matching. In: *IEEE International Conference on Progress in Informatics and Computing (PIC)*. December 10-12, Shanghai. Piscataway: IEEE Press, pp. 678 –683
- [CC92] Chang, Chienchung and Chatterjee, Shankar, 1992. Quantization error analysis in stereo vision. In: *Conference on Signals, Systems and Computers*. October 26-28, 1992. Pacific Grove, CA. Piscataway: IEEE Press, pp. 1037–1041
- [CLSF10] Calonder, Michael, Lepetit, Vincent, Christoph Strecha, et al., 2010. BRIEF: Binary Robust Independent Elementary Features. In: *Proceedings of the European conference on Computer vision (ECCV)*. September 5-11, 2010. Heronissos, Greece. Berlin, Heidelberg: Springer, pp. 778–792
- [CM02] Comaniciu, D. and Meer, P., 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(5), pp. 603–619, ISSN 0162-8828

-
- [CM05] Chum, Ondřej and Matas, Jiří, 2005. Matching with PROSAC - Progressive Sample Consensus. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. June 20-25, 2005. San Diego, CA. Piscataway: IEEE Press, ISBN 0-7695-2372-2, pp. 220–226
- [CMO⁺04] Chum, Ondřej, Matas, Jiří, Štěpán Obdržálek, et al., 2004. Enhancing RANSAC by Generalized Model Optimization. In: *Proc. of the Asian Conference on Computer Vision (ACCV)*. January 2004. Seoul, Korea South. Berlin, Heidelberg: Springer, ISBN 89-954842-0-9, pp. 812–817
- [CSY06] Cheng, Lei, Selzer, Jason, and Yee-Hong Yang, 2006. Region-Tree Based Stereo Using Dynamic Programming Optimization. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. June 17-22, 2006. New York, NY, USA. Piscataway: IEEE Press, ISBN 0-7695-2597-0, pp. 2378–2385
- [CTS11] Collet, Romea, Torres, Manuel Martinez, and Siddhartha Srinivasa, 2011. The MOPED framework: Object recognition and pose estimation for manipulation. *International Journal of Robotics Research*, **30**(1), pp. 1284–1306
- [Dia10] Diankov, Rosen, 2010. *Automated Construction of Robotic Manipulation Programs*. Pittsburgh, PA, Carnegie Mellon University, Robotics Institute, Ph.D. thesis
- [DPP09] Detry, Renaud, Pugeault, Nicolas, and Justus Piater, 2009. A Probabilistic Framework for 3D Visual Object Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**(10), pp. 1790–1803, ISSN 0162-8828
- [ESN06] Engels, Chris, Stewénus, Henrik, and David Nistér, 2006. Bundle Adjustment Rules. In: *Photogrammetric Computer Vision (PCV)*, September 20-22, 2006. Bonn, Germany, 6 pp.
- [FABV12] Fischer, Jan, Arbeiter, Georg, Richard Bormann, et al., 2012. A framework for object training and 6 DoF pose estimation. In: *German Conference on Robotics (ROBOTIK)*. May 21-22, 2012. Munich, Germany. Berlin, Offenbach: VDE Verlag, ISBN 978-3-8007-3418-4, 6 pp.
- [FAV11] Fischer, Jan, Arbeiter, Georg, and Alexander Verl, 2011. Combination of Time-of-Flight depth and stereo using semiglobal optimization. In: *International Conference on Intelligent Robots and Systems (IROS)*. May 9-13, 2001. Shanghai, China. Piscataway: IEEE Press, ISBN 978-1-61284-386-5, pp. 3548–3553

- [FB81] Fischler, Martin A. and Bolles, Robert C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, **24**(6), p. 381–395, ISSN 0001-0782
- [FBAV13] Fischer, Jan, Bormann, Richard, Georg Arbeiter, et al., 2013. A feature descriptor for texture-less object representation using 2D and 3D cues from RGB-D data. In: *IEEE International Conference on Robotics and Automation (ICRA)*. May 6-10, 2013. Karlsruhe, Germany. Piscataway: IEEE Press, 6 pp.
- [FHK⁺04] Frome, Andrea, Huber, Daniel, Ravi Kolluri, et al., 2004. Recognizing Objects in Range Data Using Regional Point Descriptors. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. May 11-14, 2004, Prague, Czech Republic. Berlin, Heidelberg: Springer, pp. 224–237
- [FP03] Forsyth, David and Ponce, Jean, 2003. *Computer vision: a modern approach*. Upper Saddle River, London: Prentice Hall, ISBN 978-0-13-608592-8
- [FRWV11] Fischer, Jan, Ruppel, Alexander, Florian Weißhardt, et al., 2011. A Rotation Invariant Feature Descriptor O-DAISY and its FPGA Implementation. In: *International Conference on Intelligent Robots and Systems (IROS)*. September 25-30, 2011. San Francisco, USA. Piscataway: IEEE Press, pp. 2365–2370
- [FSV10] Fischer, Jan, Seitz, Daniel, and Alexander Verl, 2010. Face detection using 3-D time-of-flight and colour cameras. In: *Proceedings for the joint conference of 41st International Symposium on Robotics (ISR) and the 6th German Conference on Robotics*. June 7-9, 2010. Munich, Germany. Berlin, Offenbach: VDE Verlag, pp. 112–116
- [GAL08] Gudmundsson, Sigurjon Arni, Aanaes, Henrik, and Rasmus Larsen, 2008. Fusion of stereo vision and Time-Of-Flight imaging for improved 3D estimation. *International Journal of Intelligent Systems Technologies and Applications*, **5**(3/4), p. 425, ISSN 1740-8865
- [GES⁺10] Grundmann, Thilo, Eidenberger, Robert, Martin Schneider, et al., 2010. Robust high precision 6D pose determination in complex environments for robotic manipulation. In: *Proc. Workshop Best Practice in 3D Perception and Modeling for Mobile Manipulation at IEEE International Conference on Robotics and Automation (ICRA)*. May 3-8, 2010. Anchorage, Alaska. Piscataway: IEEE Press, 7 pp.

-
- [GFW11] Grundmann, Thilo, Feiten, Wichert, and Georg v. Wichert, 2011. A Gaussian measurement model for local interest point based 6 DOF pose estimation. In: *IEEE International Conference on Robotics and Automation (ICRA)*. May 9-13, 2011. Shanghai, China. Piscataway: IEEE Press, pp. 2085–2090
- [HA08] Hahne, Uwe and Alexa, Marc, 2008. Combining Time-Of-Flight depth and stereo images without accurate extrinsic calibration. *International Journal of Intelligent Systems Technologies and Applications*, **5**(3/4), p. 325, ISSN 1740-8865
- [HBB⁺11] Hägele, Martin, Bengel, Matthias, Nikolaus Blümlein, et al., 2011. *Wirtschaftlichkeitsanalysen neuartiger Servicerobotik - Anwendungen und ihre Bedeutung für die Robotik-Entwicklung*. Stuttgart: Fraunhofer IPA
- [HCI⁺11] Hinterstoisser, Stefan, Cagniart, Cedric, Slobodan Ilic, et al., 2011. Multi-modal Templates for Real-Time Detection of Texture-less Objects in Heavily Cluttered Scenes. In: *IEEE International Conference on Computer Vision (ICCV)*. November 6-13, 2011. Barcelona, Spain. Piscataway: IEEE Press, ISBN 978-1-4577-1101-5, pp. 858–865
- [HCI⁺12] Hinterstoisser, Stefan, Cagniart, Cedric, Slobodan Ilic, et al., 2012. Gradient Response Maps for Real-Time Detection of Textureless Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(5), pp. 876–888, ISSN 0162-8828
- [HHN88] Horn, Berthold K. P., Hilden, Hugh. M., and Shariar Negahdaripour, 1988. Closed-Form Solution of Absolute Orientation using Orthonormal Matrices. *Journal of the Optical Society of America*, **5**(7), pp. 1127–1135
- [Hir08] Hirschmüller, Heiko, 2008. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(2), pp. 328–341, ISSN 0162-8828
- [HJL⁺89] Haralick, Robert M., Joo, Hyonam, Chung-Nan Lee, et al., 1989. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, **19**(6), pp. 1426–1446, ISSN 0018-9472
- [HLI⁺10] Hinterstoisser, Stefan, Lepetit, Vincent, Slobodan Ilic, et al., 2010. Dominant Orientation Templates for Real-Time Detection of Texture-Less Objects. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. June 13-18, 2010. San Francisco, CA. Piscataway: IEEE Press, ISBN 978-1-4244-6984-0, pp. 2257–2264

- [Hor87] Horn, Berthold K. P., 1987. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, **4**(4), pp. 629–642
- [JH99] Johnson, Andrew E. and Hebert, Martial, 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**(5), pp. 433–449, ISSN 0162-8828
- [Ken44] Kenneth, Levenberg, 1944. A Method for the Solution of Certain Non-Linear Problems in Least Squares. *Quarterly of Applied Mathematics*, **2**(2), pp. 164–168
- [KHRF11] Krainin, Michael, Henry, Peter, Xiaofeng Ren, et al., 2011. Manipulator and object tracking for in-hand 3D object modeling. *The International Journal of Robotics Research*, **30**(11), pp. 1311–1327, ISSN 0278-3649
- [KKHA09] Kolb, Andreas, Koch, Reinhard, Uwe Hahne, et al., 2009. Depth Imaging by Combining Time-of-Flight and On-Demand Stereo. In: *Lecture Notes in Computer Science*, 5742, Berlin, Heidelberg: Springer, ISBN 978-3-642-03777-1, pp. 70–83
- [KPVG10] Knopp, Jan, Prasad, Mukta, and Luc Van Gool, 2010. Orientation invariant 3D object classification using hough transform based methods. In: *Proceedings of the ACM workshop on 3D object retrieval*. October 25-29, 2010. Firenze, Italy. New York: ACM, ISBN 978-1-4503-0160-2, pp. 15–20
- [KSK06] Klaus, Andreas, Sormann, Mario, and Konrad Karner, 2006. Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure. In: *International Conference on Pattern Recognition (ICPR)*. August 20-24, 2006. Hong Kong. Piscataway: IEEE Press, ISBN 0-7695-2521-0, pp. 15–18
- [KXD12] Kasper, Alexander, Xue, Zhixing, and Rüdiger Dillmann, 2012. The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research*, **31**(8), pp. 927–934, ISSN 0278-3649
- [KZ01] Kolmogorov, Vladimir and Zabih, Ramin, 2001. Computing Visual Correspondence with Occlusions via Graph Cuts. In: *International Conference on Computer Vision (ICCV)*. July 7-14, 2001. Vancouver, BC. Piscataway: IEEE Press, ISBN 0-7695-1143-0, pp. 508–515
- [LA04] Lourakis, Manolis I. A. and Argyros, Antonis A., 2004. *The design and implementation of a generic sparse bundle adjustment software package based on*

- the levenberg-marquardt algorithm*, 340. Heraklion, Greece: Foundation for Research and Technology - Hellas (FORTH)
- [LK06] Lindner, Marvin and Kolb, Andreas, 2006. Lateral and depth calibration of pmd-distance sensors. In: *Proceedings of the Second international conference on Advances in Visual Computing (ISVC)*. November 6-8, 2006. Lake Tahoe, USA. Berlin, Heidelberg: Springer, pp. 524–533
- [Low03] Lowe, David G., 2003. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, **60**(2), pp. 91–110
- [ML09] Muja, Marius and Lowe, David G., 2009. Fast approximate nearest neighbors with automatic algorithm configuration. In: *International Conference on Computer Vision Theory and Applications (VISAPP)*. February 5-8, 2009. Lisboa, Portugal. Berlin, Heidelberg: Springer, pp. 331–340
- [MMP87] Marroquin, J. L., Mitter, S. K., and T. A. Poggio, 1987. Probabilistic Solution of Ill-Posed Problems in Computational Vision. *Journal of the American Statistical Association (ASAJ)*, **82**(397), pp. 76–89
- [MRBL11] Muja, Marius, Rusu, Radu Bogdan, Gary Bradski, et al., 2011. REIN - A Fast, Robust, Scalable REcognition INfrastructure. In: *IEEE International Conference on Robotics and Automation (ICRA)*. May 9-13, 2001. Shanghai, China. Piscataway: IEEE Press, ISBN 978-1-61284-386-5, 8 pp.
- [MS05] Mikolajczyk, Krystian and Schmid, Cordelia, 2005. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(10), pp. 1615–1630, ISSN 0162-8828
- [MS12] Manap, Nurulfajar and Soraghan, John, 2012. Disparity refinement based on depth image layers separation for stereo matching algorithms. *Journal of Telecommunication Electronic and Computer Engineering*, **4**(1), pp. 51–64
- [MTD12] Molnár, B., Toth, C. K., and A. Detrekői, 2012. Accuracy Test of Microsoft Kinect for Human Morphologic Measurements. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, August 25th - September 1st, 2012. Melbourne, Australia, pp. 543–547
- [MTRW03] Montemerlo, Michael, Thrun, Sebastian, Daphne Roller, et al., 2003. Fast-SLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: *Proceedings of the 18th international joint conference on Artificial intelligence*. San Francisco, CA, USA. Burlington: Morgan Kaufmann, pp. 1151–1156

- [NLM⁺12] Nair, Rahul, Lenzen, Frank, Stephan Meister, et al., 2012. High Accuracy TOF and Stereo Sensor Fusion at Interactive Rates. In: *Proceedings of the European conference on Computer vision (ECCV)*. October 7-13, 2012. Florence, Italy. Berlin, Heidelberg: Springer, ISBN 978-3-642-33867-0, pp. 1–11
- [OCLF10] Ozuysal, Mustafa, Calonder, Michael, Vincent Lepetit, et al., 2010. Fast Keypoint Recognition Using Random Ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(3), pp. 448–461, ISSN 0162-8828
- [PJA10] Paulevé, Loïc, Jégou, Hervé, and Laurent Amsaleg, 2010. Locality sensitive hashing: A comparison of hash function types and querying mechanisms. *Pattern Recognition Letters*, **31**(11), pp. 1348–1358, ISSN 0167-8655
- [QCG⁺09] Quigley, Morgan, Conley, Ken, Brian P. Gerkey, et al., 2009. ROS: an open-source Robot Operating System. In: *ICRA Workshop on Open Source Software*. May 12-17, 2009. Kobe, Japan. Piscataway: IEEE Press, 6 pp.
- [RCF⁺09] Reiser, Ulrich, Connette, Christian, Jan Fischer, et al., 2009. Care-O-bot 3 - Creating a product vision for service robot applications by integrating design and technology. In: *International Conference on Intelligent Robots and Systems (IROS)*. October 11-15, 2009. St. Louis, USA. Piscataway: IEEE Press, ISBN 978-1-4244-3803-7, pp. 1992–1998
- [RPD10] Rosten, Erwin, Porter, Reid, and Tom Drummond, 2010. Faster and Better: A Machine Learning Approach to Corner Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(1), pp. 105 –119, ISSN 0162-8828
- [RRKB11] Rublee, Ethan, Rabaud, Vincent, Kurt Konolige, et al., 2011. ORB: An Efficient Alternative to SIFT or SURF. In: *International Conference on Computer Vision (ICCV)*. November 6-13, 2011. Barcelona, Spain. Piscataway: IEEE Press, ISBN 978-1-4577-1101-5, pp. 2564–2571
- [SAH08] Silpa-Anan, Chanop and Hartley, Richard, 2008. Optimised KD-trees for fast image descriptor matching. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 23-28, 2008. Anchorage, AK. Piscataway: IEEE Press, ISBN 978-1-4244-2242-5, pp. 1–8
- [Sha06] Shakhnarovich, Greg, 2006. *Learning Task-specific Similarity*. Cambridge, Massachusetts, Massachusetts Institute of Technology (MIT), Ph.D. thesis
- [SMD10a] Strasdat, Hauke, Montiel, J. M. M., and Andrew J Davison, 2010. Real-time monocular SLAM: Why filter? In: *IEEE International Conference on Robotics*

- and Automation (ICRA)*. May 3-8, 2010. Anchorage, AK. Piscataway: IEEE Press, ISBN 978-1-4244-5038-1, pp. 2657–2664
- [SMD⁺10b] Strasdat, Hauke, Montiel, J. M. M., Andrew J. Davison, et al., 2010. Scale Drift-Aware Large Scale Monocular SLAM. In: *Robotics: Science and Systems*. June 27-30, 2010. Zaragoza, Spain. Cambridge, Massachusetts: The MIT Press, ISBN 978-0-262-51681-5, 8 pp.
- [SS12] Scharstein, Daniel and Szeliski, Richard, 2012, Middlebury Stereo Vision Page. <http://vision.middlebury.edu/stereo/>, accessed: October 10th, 2013
- [SSZ02] Scharstein, Daniel, Szeliski, Richard, and Ramin Zabih, 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: *IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV)*. December 9-10, 2001. Kauai, HI. Piscataway: IEEE Press, ISBN 0-7695-1327-1, pp. 131–140
- [SWP⁺12] Shi, Chenbo, Wang, Guijin, Xiaokang Pei, et al., 2012. Stereo Matching Using Local Plane Fitting in Confidence-Based Support Window. *IEICE Transactions on Information and Systems*, **E95.D(2)**, pp. 699–702
- [SZS⁺08] Szeliski, Richard, Zabih, Ramin, Daniel Scharstein, et al., 2008. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30(6)**, pp. 1068–1080, ISSN 0162-8828
- [TCF09] Toldo, Roberto, Castellani, Umberto, and Andrea Fusiello, 2009. A Bag of Words Approach for 3D Object Categorization. In: *Proceedings of the 4th International Conference on Computer Vision/Computer Graphics Collaboration Techniques*. May 4-6, 2009. Rocquencourt, France. Berlin, Heidelberg: Springer, ISBN 978-3-642-01810-7, pp. 116–127
- [TCS10] Torres, Manuel Martinez, Collet, Alvaro, and Siddhartha Srinivasa, 2010. MOPED: A Scalable and Low Latency Object Recognition and Pose Estimation System. In: *IEEE International Conference on Robotics and Automation (ICRA)*. May 3-8, 2010. Anchorage, AK. Piscataway: IEEE Press
- [TLF08] Tola, Engin, Lepetit, Vincent, and Pascal Fua, 2008. A fast local descriptor for dense matching. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. June 23-28, 2008. Anchorage, AK. Piscataway: IEEE Press, ISBN 978-1-4244-2242-5, pp. 1–8

- [TM98] Tomasi, C. and Manduchi, R., 1998. Bilateral filtering for gray and color images. In: *International Conference on Computer Vision (ICCV)*. January 4-7, 1998. Bombay. Piscataway: IEEE Press, pp. 839–846
- [TRD09] Taylor, Simon, Rosten, Edward, and Tom Drummond, 2009. Robust feature matching in 2.3 μ s. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. June 20-25, 2009. Miami, FL. Piscataway: IEEE Press, ISBN 978-1-4244-3994-2, pp. 15–22
- [TSDS10] Tombari, Federico, Salti, Samuele, and Luigi Di Stefano, 2010. Unique signatures of histograms for local surface description. In: *Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III*. September 5-11, 2010. Hersonissos, Greece. Berlin, Heidelberg: Springer, ISBN 978-3-642-15557-4, pp. 356–369
- [VJ01] Viola, Paul and Jones, Michael, 2001. Rapid object detection using a boosted cascade of simple features. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. December 8-14, 2001. Kauai, HI. Piscataway: IEEE Press, pp. 511–518
- [WB07] Winder, Simon A. J. and Brown, Matthew, 2007. Learning Local Image Descriptors. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. June 18-23, 2007. Minneapolis, Minnesota, USA. Piscataway: IEEE Press, ISBN 1-4244-1179-3, pp. 1–8
- [WPR⁺12] Wüthrich, Manuel, Pastor, Peter, Ludovic Righetti, et al., 2012. Probabilistic depth image registration incorporating nonvisual information. In: *IEEE International Conference on Robotics and Automation (ICRA)*. May 14-18, 2012. St. Paul, MN, USA. Piscataway: IEEE Press, pp. 3637–3644
- [WZ08] Wang, Zeng-Fu and Zheng, Zhi-Gang, 2008. A region based stereo matching algorithm using cooperative optimization. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 23-28, 2008. Anchorage, AK. Piscataway: IEEE Press, ISBN 978-1-4244-2242-5, pp. 1–8
- [Yan12] Yang, Qingxiong, 2012. Recursive Bilateral Filtering. In: *European Conference on Computer Vision (ECCV)*. October 7-13, 2012. Firenze, Italy. Berlin, Heidelberg: Springer, ISBN 978-3-642-33717-8, pp. 399–413
- [YTCA10] Yang, Qingxiong, Tan, Kar-Han, W. Bruce Culbertson, et al., 2010. Fusion of active and passive sensors for fast 3D capture. In: *IEEE International Workshop on Multimedia Signal Processing (MMSP)*. October 4-6, 2010. Saint-Malo, France. Piscataway: IEEE Press, ISBN 978-1-4244-8110-1, pp. 69–74

-
- [YYDN07] Yang, Qingxiong, Yang, Ruigang, James Davis, et al., 2007. Spatial-Depth Super Resolution for Range Images. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. June 17-22, 2007. Minneapolis, MN. Piscataway: IEEE Press, ISBN 1-4244-1179-3, pp. 1–8
- [Zha00] Zhang, Zhengyou, 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(11), pp. 1330–1334, ISSN 0162-8828
- [Zho09] Zhong, Yu, 2009. Intrinsic shape signatures: A shape descriptor for 3D object recognition. In: *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. September 27th - October 4th, 2009. Kyoto. Piscataway: IEEE Press, ISBN 978-1-4244-4442-7, pp. 689–696
- [ZWGY10] Zhu, Jiejie, Wang, Liang, Jizhou Gao, et al., 2010. Spatial-Temporal Fusion for High Accuracy Depth Maps Using Dynamic MRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(5), pp. 899–909, ISSN 0162-8828
- [ZWY⁺11] Zhu, Jiejie, Wang, Liang, Ruigang Yang, et al., 2011. Reliability Fusion of Time-of-Flight Depth and Stereo Geometry for High Quality Depth Maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**(7), pp. 1400–1414, ISSN 0162-8828
- [ZWYD08] Zhu, Jiejie, Wang, Liang, Ruigang Yang, et al., 2008. Fusion of time-of-flight depth and stereo for high accuracy depth maps. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. June 23-28, 2008. Anchorage, AK. Piscataway: IEEE Press, ISBN 978-1-4244-2242-5, pp. 1–8

Perception is the core issue of cognitive robotics. It enables the robot to comprehend its environment and provides a symbolic representation for the cognitive processes. Humans possess a sophisticated perception system, which enables them to robustly determine the presence of a specific object within their environment. Robotics research has the ambition to understand and mimic these superior human capabilities, enabling the recognition of arbitrary objects in arbitrary contexts. The objective of this thesis is to develop a perception system for the 6 DoF localization of typical rigid household objects, that enables an intuitive teaching of new objects. The system operates on single images and is designed to handle occlusions and multi-occurrences of objects under varying lighting conditions.

ISBN 978-3-8396-0891-3



FRAUNHOFER VERLAG