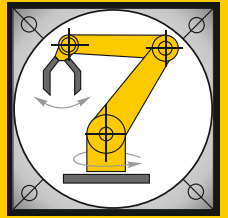


Institut für Informatik
Lehrstuhl für Robotik und Telematik
Prof. Dr. K. Schilling
Prof. Dr. A. Nüchter



Würzburger Forschungsberichte
in Robotik und Telematik

Uni Wuerzburg Research Notes
in Robotics and Telematics

Julius-Maximilians-

**UNIVERSITÄT
WÜRZBURG**

Dissertation an der Universität
Würzburg im Rahmen der GSST

Christian Pfitzner

Visual Human Body
Weight Estimation
with Focus on
Clinical Applications

Band 18

Die Schriftenreihe

wird vom Lehrstuhl für Informatik VII: Robotik und Telematik der Universität Würzburg herausgegeben und präsentiert innovative Forschung aus den Bereichen der Robotik und der Telematik.

Die Kombination fortgeschrittener Informationsverarbeitungsmethoden mit Verfahren der Regelungstechnik eröffnet hier interessante Forschungs- und Anwendungsperspektiven. Es werden dabei folgende interdisziplinäre Aufgabenschwerpunkte bearbeitet:

- Robotik und Mechatronik: Kombination von Informatik, Elektronik, Mechanik, Sensorik, Regelungs- und Steuerungstechnik, um Roboter adaptiv und flexibel ihrer Arbeitsumgebung anzupassen.
- Telematik: Integration von Telekommunikation, Informatik und Steuerungstechnik, um Dienstleistungen an entfernten Standorten zu erbringen.

Anwendungsschwerpunkte sind u.a. mobile Roboter, Tele-Robotik, Raumfahrtssysteme und Medizin-Robotik.

Lehrstuhl Informatik VII
Robotik und Telematik
Am Hubland
D-97074 Würzburg

Tel.: +49 (0) 931 - 31 - 86678
Fax: +49 (0) 931 - 31 - 86679

schi@informatik.uni-wuerzburg.de
<http://www7.informatik.uni-wuerzburg.de>

Dieses Dokument wird bereitgestellt
durch den Online-Publikationsservice
der Universität Würzburg.

Universitätsbibliothek Würzburg
Am Hubland
D-97074 Würzburg

Tel.: +49 (0) 931 - 31 - 85906

opus@bibliothek.uni-wuerzburg.de
<https://opus.bibliothek.uni-wuerzburg.de>

ISSN: 1868-7474 (online)
ISSN: 1868-7466 (print)
ISBN: 978-3-945459-27-0 (online)

Zitation dieser Publikation

PFITZNER, C. (2019). Visual Human Body Weight Estimation with Focus on Clinical Applications. Schriftenreihe Würzburger Forschungsberichte in Robotik und Telematik, Band 18. Würzburg: Universität Würzburg.
URN: urn:nbn:de:bvb:20-opus-174842

Dissertation an der Universität Würzburg
im Rahmen der GSST

Doctoral thesis
for the doctoral degree
Doctor rerum naturalium (Dr. rer. nat.)

Visual Human Body Weight Estimation
with Focus on Medical Applications

*Optische Körpergewichtsschätzung
für medizinische Anwendungen*



Submitted by
Christian Pfitzner
from
Schwarzenbruck, Germany
Würzburg, 2018

Submitted on / *Eingereicht am*: August 06th, 2018

Members of thesis committee / *Mitglieder des Promotionskomitees*

Chairperson / Vorsitz: Prof. Dr. Andreas Hotho

1. Reviewer and Examiner / 1. Gutachter und Prüfer: Prof. Dr. Andreas Nüchter
2. Reviewer and Examiner / 2. Gutachter und Prüfer: Prof. Dr. Stefan May
3. Examiner / 3. Prüfer: Prof. Dr. Klaus Schilling

Day of thesis defense / *Tag des Promotionskolloquiums*: December 20th, 2018

Affidavit

I hereby confirm that my thesis entitled *Visual Human Body Weight Estimation with Focus on Medical Applications* is the result of my own work. I did not receive any help or support from commercial consultants. All sources and / or materials applied are listed and specified in the thesis.

Furthermore, I confirm that this thesis has not yet been submitted as part of another examination process neither in identical nor in similar form.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, die Dissertation *Optische Körpergewichtsschätzung für medizinische Anwendungen* eigenständig, d.h. insbesondere selbständig und ohne Hilfe eines kommerziellen Promotionsberaters, angefertigt und keine anderen als die von mir angegebenen Quellen und Hilfsmittel verwendet zu haben.

Ich erkläre außerdem, dass die Dissertation weder in gleicher noch in ähnlicher Form bereits in einem anderen Prüfungsverfahren vorgelegen hat.

”Any sufficiently advanced technology is indistinguishable from magic.”

Sir Arthur Charles Clarke (1917-2008),
British Science Fiction Author, Inventor, and Futurist

Acknowledgment

At the beginning of this thesis, I want to thank some special people, helping me to fulfill my work and this thesis. I am very grateful for my first reviewer and supervisor Prof. Dr. Andreas Nüchter, who gave me the opportunity to do my Ph.D. at the Julius-Maximilians-University, Würzburg, Germany. His knowledge and guidance in the field of 3D point cloud processing helped me to improve the presented algorithms. Additionally, I want to thank my second supervisor Prof. Dr. Klaus Schilling for his insightful comments during discussions and his review. I truly appreciate the organization and participation in the thesis defense by Prof. Dr. Andreas Hotho and Dr. Dorit Borrmann. I would also like to thank the organizers of the Graduate School for Science and Technology, as well as the employees and students in the Advanced seminar of the Department of Computer Science VII – Robotics and Telematics.

A very special thanks goes to my supervisor and reviewer Prof. Dr. Stefan May: Since the beginning of my Master's degree, he helped me to improve myself every day. He showed me that the world of engineering is not limited to the programmable logic controller and that there are more and interesting things in science, like mobile robotics or 3D perception. I really enjoyed my time at the Technische Hochschule Nürnberg, Georg Simon Ohm (THN), studying, working and doing various things besides my Ph.D., like the RoboCup Rescue. Therefore, I want to thank all members of the Team AutonOHM for helping me within the last four years to complete my Ph.D.. I am sure, I will miss the weekly meetings and the intense preparation phase of RoboCup.

Without funding, the here presented results would not have been possible: I'd like to thank the BayWISS research training group Digitization for the scholarship, as well as the funding of an open access publication. The German Academic Exchange Service (DAAD) funded my contribution to the IFAC in Toulouse, France, in 2017. Most of the publications were funded by the Federal Ministry of Education and Research in Germany with the project funding reference number 03FH040PX3. Within this funding, I want to say thank you to the physicians of the University Hospital Erlangen, Germany, Dr. Lorenz Breuer and Dr. Martin Köhrmann, for their work collecting data from real patients in the emergency room. Without these data sets and the application of this thesis, my work would only be half that worth. Both were my contact persons to improve the system *Libra3D*, the user interface, and the structure of the database which is formed based on their requirements. I also enjoyed the joint work with Dr. Joel Braun and Franz Dirauf from the Siemens Healthcare GmbH.

Within this thesis, we submitted a patent. Therefore, I'd like to thank Dr. Rolf Kapust, the inventor consultant of the THN. Without his help, I would have never taken the step of registering the ideas presented in this thesis as a patent. I enjoyed the discussions in the context of patent and the support in collaboration with the German Patent and Trade Mark Office.

I'd like to thank Johannes Ziegler, Johanna Gleichauf, Dr. Lorenz Breuer, and Kirsten Ludwig for proofreading. Without you, there would be much more errors in this thesis.

Several students helped me while working on my Ph.D. due to their work in project, bachelor or master theses. These students are Eduard Roth, Alexey Akhrymenka, Anna Federle, Anna Wegleiter, Christian Regnat, Christian Espig, Marion Kaiser, Deniz Neufeld, and Johann Delchmann. They did remarkable work with their motivation and enthusiasm and helped me to improve the whole system with their contribution.

Finally, I want to thank my friends, my family and especially my wife Kirsten Ludwig: Without her love and unconditional support, this thesis would have been a lot more work, and less fun.

Abstract

It is the aim of this thesis to present a visual body weight estimation, which is suitable for medical applications. A typical scenario where the estimation of the body weight is essential, is the emergency treatment of stroke patients: In case of an ischemic stroke, the patient has to receive a body weight adapted drug, to solve a blood clot in a vessel. The accuracy of the estimated weight influences the outcome of the therapy directly. However, the treatment has to start as early as possible after the arrival at a trauma room, to provide sufficient treatment. Weighing a patient takes time, and the patient has to be moved. Furthermore, patients are often not able to communicate a value for their body weight due to their stroke symptoms. Therefore, it is state of the art that physicians guess the body weight. A patient receiving a too low dose has an increased risk that the blood clot does not dissolve and brain tissue is permanently damaged. Today, about one-third gets an insufficient dosage. In contrast to that, an overdose can cause bleedings and further complications. Physicians are aware of this issue, but a reliable alternative is missing.

The thesis presents state-of-the-art principles and devices for the measurement and estimation of body weight in the context of medical applications. While scales are common and available at a hospital, the process of weighing takes too long and can hardly be integrated into the process of stroke treatment. Sensor systems and algorithms are presented in the section for related work and provide an overview of different approaches. The here presented system – called *Libra3D* – consists of a computer installed in a real trauma room, as well as visual sensors integrated into the ceiling. For the estimation of the body weight, the patient is on a stretcher which is placed in the field of view of the sensors. The three sensors – two RGB-D and a thermal camera – are calibrated intrinsically and extrinsically. Also, algorithms for sensor fusion are presented to align the data from all sensors which is the base for a reliable segmentation of the patient.

A combination of state-of-the-art image and point cloud algorithms is used to localize the patient on the stretcher. The challenges in the scenario with the patient on the bed is the dynamic environment, including other people or medical devices in the field of view. After the successful segmentation, a set of hand-crafted features is extracted from the patient’s point cloud. These features rely on geometric and statistical values and provide a robust input to a subsequent machine learning approach. The final estimation is done with a previously trained artificial neural network.

The experiment section offers different configurations of the previously extracted feature vector. Additionally, the here presented approach is compared to state-of-the-art methods; the patient’s own assessment, the physician’s guess, and an anthropometric estimation. Besides the patient’s own estimation, *Libra3D* outperforms all state-of-the-art estimation methods: 95 percent of all patients are estimated with a relative error of less than 10 percent to ground truth body weight. It takes only a minimal amount of time for the measurement, and the approach can easily be integrated into the treatment of stroke patients, while physicians are not hindered. Furthermore, the section for experiments demonstrates two additional applications: The extracted features can also be used to estimate the body weight of people standing, or even walking in front of a 3D camera. Also, it is possible to determine or classify the BMI of a subject on a stretcher. A potential application for this approach is the reduction of the radiation dose of patients being exposed to X-rays during a CT examination. During the time of this thesis, several data sets

were recorded. These data sets contain the ground truth body weight, as well as the data from the sensors. They are available for the collaboration in the field of body weight estimation for medical applications.

Zusammenfassung

Diese Arbeit zeigt eine optische Körpergewichtsschätzung, welche für medizinische Anwendungen geeignet ist. Ein gängiges Szenario, in dem eine Gewichtsschätzung benötigt wird, ist die Notfallbehandlung von Schlaganfallpatienten: Falls ein ischämischer Schlaganfall vorliegt, erhält der Patient ein auf das Körpergewicht abgestimmtes Medikament, um einen Thrombus in einem Gefäß aufzulösen. Die Genauigkeit der Gewichtsschätzung hat direkten Einfluss auf den Erfolg der Behandlung. Hinzu kommt, dass die Behandlung so schnell wie möglich nach der Ankunft im Krankenhaus erfolgen muss, um eine erfolgreiche Behandlung zu garantieren. Das Wiegen eines Patienten ist zeitaufwändig und der Patient müsste hierfür bewegt werden. Des Weiteren können viele Patienten aufgrund des Schlaganfalls nicht ihr eigenes Gewicht mitteilen. Daher ist es heutzutage üblich, dass Ärzte das Gewicht schätzen. Erhält ein Patient eine zu geringe Dosis, steigt das Risiko, dass sich der Thrombus nicht auflöst und das Gehirngewebe dauerhaft geschädigt bleibt. Eine Überdosis kann dagegen zu Blutungen und weiteren Komplikationen führen. Ein Drittel der Patienten erhält heutzutage eine unzureichende Dosis. Ärzte sind sich dessen bewusst, aber derzeit gibt es kein alternatives Vorgehen.

Diese Arbeit präsentiert Elemente und Geräte zur Messung und Schätzung des Körpergewichts, die im medizinischen Umfeld verwendet werden. Zwar sind Waagen im Krankenhaus üblich, aufgrund des engen Zeitfensters für die Behandlung können sie aber nur schlecht in den Behandlungsablauf von Schlaganfallpatienten integriert werden. Der Abschnitt zum Stand der Technik zeigt verschiedene Sensorsysteme und Algorithmen. Das hier gezeigte System – genannt *Libra3D* – besteht aus einem Computer im Behandlungsraum, sowie den in der Decke integrierten optischen Sensoren. Für die Gewichtsschätzung befindet sich der Patient auf einer Liege im Blickfeld der Sensoren. Die drei Sensoren – zwei RGB-D- und einer Wärmebildkamera – sind intrinsisch und extrinsisch kalibriert.

Des Weiteren werden Algorithmen zur Sensorfusion vorgestellt, welche die Daten für eine erfolgreiche Segmentierung des Patienten zusammenführen. Eine Kombination aus verschiedenen gängigen Bildverarbeitungs- und Punktwolken-Algorithmen lokalisiert den Patienten auf der Liege. Die Herausforderung in diesem Szenario mit dem Patienten auf dem Bett sind ständige Veränderungen, darunter auch andere Personen oder medizinische Geräte im Sichtfeld. Nach der erfolgreichen Segmentierung werden Merkmale von der Punktwolke des Patienten extrahiert. Diese Merkmale beruhen auf geometrischen und statistischen Eigenschaften und bieten robuste Werte für das nachfolgende maschinelle Lernverfahren. Die Schätzung des Gewichts basiert letztlich auf einem zuvor trainierten künstlichen neuronalen Netz.

Das Kapitel zu den Experimenten zeigt verschiedene Kombinationen von Werten aus dem Merkmalsvektor. Zusätzlich wird der Ansatz mit Methoden aus dem Stand der Technik verglichen: der Schätzung des Patienten, des Arztes, und einer anthropometrischen Schätzung. Bis auf die eigene Schätzung des Patienten übertrifft *Libra3D* hierbei alle anderen Methoden: 95 Prozent aller Schätzungen weisen einen relativen Fehler von weniger als 10 Prozent zum realen Körpergewicht auf. Dabei benötigt das System wenig Zeit für eine Messung und kann einfach in den Behandlungsablauf von Schlaganfallpatienten integriert werden, ohne Ärzte zu behindern. Des Weiteren zeigt der Abschnitt für Experimente zwei weitere Anwendungen: Die extrahierten Merkmale können dazu verwendet werden das Gewicht von stehenden und auch laufenden Personen zu schätzen, die sich vor einer 3D-Kamera befinden. Darüber hinaus ist es

auch möglich den BMI von Patienten auf einer Liege zu bestimmen. Diese kann die Strahlenexposition bei CT-Untersuchungen beispielsweise verringern. Während dieser Dissertation sind einige Datensätze entstanden. Sie enthalten das reale Gewicht, sowie die dazugehörigen Sensordaten. Die Datensätze sind für die Zusammenarbeit im Bereich der Körpergewichtsschätzung für medizinische Anwendungen verfügbar.

Contents

List of Figures	xix
List of Tables	xxi
List of Listings	xxiii
List of Abbreviations	xxiv
List of Symbols	xxvi
1 Introduction	1
1.1 The Human Brain	1
1.2 Stroke	2
1.2.1 Diagnosis	4
1.2.2 Treatment of Ischemic Stroke with tPA	5
1.3 Body Weight Adapted Dosing	7
1.4 Objectives and Contributions	7
1.5 List of Publications	9
1.6 Structure of this Thesis	10
2 Related Work	13
2.1 Traditional Weight Measurement	13
2.2 Weight Estimation Methods	16
2.2.1 Weight Estimation Devices	16
2.2.2 Weight Estimation Based on Anthropometric Features	17
2.2.3 Weight Estimation with Optical Sensors	19
2.2.4 Weight Estimation in Video Streams	22
2.3 Summary	23
3 Conceptual Design and Sensors	25
3.1 Environment	25
3.2 Diagnostics and Treatment	27
3.3 Handling of the Weight Estimation with Libra3D	27
3.4 Data Management	32

3.5	Applied Sensors for contact-less Estimation	34
3.5.1	Monocular Camera	35
3.5.2	Time-of-Flight Camera	36
3.5.3	Structured Light Sensor	39
3.5.4	Thermal Camera	41
3.6	Representations of Sensor Data	42
3.7	Sensor Calibration	43
3.7.1	Intrinsic Calibration	44
3.7.2	Extrinsic Calibration	47
3.7.3	Multimodal Calibration Target	50
3.7.4	Noise Model Calibration for Depth Sensors	51
3.7.5	Syncing of multiple Sensor Streams	53
3.8	Sensor Fusion	54
3.9	Summary	55
4	Segmentation of Humans from the Environment	57
4.1	Scenario for Segmentation	58
4.2	Bounding Box Filter	59
4.3	Thermal Filter	63
4.4	Color Filter	64
4.5	Normal Filter	66
4.6	Plane Filter	68
4.7	Thermal Plane Distance Filter	73
4.8	Background Subtraction	74
4.9	Segmentation based on Edges	75
4.10	Removing Distortions in the Segmentation	77
4.11	Distinguish between several People in the Scene	81
4.12	Timing Analysis for Segmentation	82
4.13	Summary	84
5	Feature Extraction for Body Weight Estimation	87
5.1	Feature Extraction	88
5.1.1	Geometric Features	88
5.1.2	Features from Eigenvalues	93
5.1.3	Statistic Features	95
5.1.4	Contour Features	96
5.1.5	Features from Personal Data	98
5.1.6	Thermal Features	99
5.2	Feature Extraction for Standing and Walking People	101
5.3	Changes in Posture	102
5.4	Correlation between the Features and Body Weight	104
5.5	Comparison of Subject Extrema	105
5.6	Timing Analysis for Body Weight Estimation	108
5.7	Summary	109

6	Supervised Learning	113
6.1	Model of a Single Neuron	114
6.2	Net Architecture	116
6.3	Forward Propagation	117
6.4	Learning	119
6.5	Issues in Training of Neural Networks	121
6.5.1	Scaling of Feature Values	121
6.5.2	Overfitting	122
6.5.3	Number of Neurons in the Hidden Layer	122
6.6	Summary	124
7	Experiments and Results	125
7.1	Data for Testing and Validation	125
7.2	Setting for Validation	128
7.3	Evaluation from different Feature Assemblies	130
7.4	Sensor Modalities	135
7.5	Comparison against Related Work	138
7.6	Extension for Standing People	141
7.6.1	Weight Estimation from Standing People	141
7.6.2	Weight Estimation from Walking Subjects	143
7.7	Extension for BMI Estimation for CT Dose Reduction	145
7.8	Summary	148
8	Conclusion and Future Work	149
	References	153
A	Appendix	I
A.1	Software structure	II
A.2	Finding the Nearest Neighbors	III
A.3	Computation of Normals	IV
A.4	Principal Component Analysis	IV

List of Figures

1.1	Ischemic and hemorrhagic stroke	2
1.2	Radiography of an ischemic stroke from Computer Tomography (CT) and Magnet Resonance Imaging (MRI) with Diffusion Weighted Imaging (DWI) imaging . . .	5
1.3	Medical imaging devices to diagnose strokes.	6
2.1	Different principles in weight measurement.	14
2.2	Different devices for weighing in the hospital environment.	15
2.3	Weight approximation devices	17
2.4	Various methods to estimate the weight of livestock	19
2.5	Various approaches to obtaining anthropometric data via a 3D sensor.	20
2.6	Related work from Nguyen et al. [145]	22
2.7	Extraction of features presented by Arigbabu et al. [10]	23
3.1	Clinical integration of sensors into a trauma room	26
3.2	Graphical User Interface (GUI) with data handling for a new patient	28
3.3	GUI for offline processing with a database	28
3.4	Process in weight acquisition for patients in the trauma room	30
3.5	Process of data acquisition in the program Libra3D	31
3.6	Structure of the My Structured Query Language (MySQL) database applied for Libra3D.	33
3.7	Sensors tested for body weight estimation within this thesis.	34
3.8	Time of Flight (ToF) principle based on a sinusoidal modulated input signal. . .	37
3.9	Removing jumping edge error from scene recorded with a ToF sensor.	38
3.10	Scene from Kinect One camera	39
3.11	Depth measurement via active triangulation	39
3.12	Scene from a Kinect camera	40
3.13	Person recorded with a thermal camera in different false-color representations . .	42
3.14	Process of sensor calibration.	44
3.15	Rigidly mounted Kinect, Kinect One, and Optris Pi400	45
3.16	The pinhole camera model	46
3.17	Different kind of distortions.	47
3.18	Transformation tree for the system's sensors.	48
3.19	Calibration of the color image and calibration pattern visible in thermal image .	51
3.20	Comparison of the sensor's noise between the Kinect and the Kinect One. . . .	52

3.21	Depth image and point cloud from a scene, filtered with a bilateral filter	53
3.22	Synchronization of multiple sensors based on optical flow	54
3.23	Visualization of the sensor fusion	55
4.1	Result from segmentation	58
4.2	Reducing the point cloud's size by bounding box filter	60
4.3	Bounding box for pre-filtering and oriented minimum bounding box around the stretcher with patient.	61
4.4	Minimum Bounding Box Filter	62
4.5	Filtering based on oriented minimal bounding box	63
4.6	Applied thermal filter to segment a patient from the stretcher	65
4.7	Segmentation based on a point's color	66
4.8	Applied normal filter with different configurations	67
4.9	Different definitions of a plane.	68
4.10	Progress for RANSAC with applied line model	71
4.11	Applied RANSAC algorithm to estimate the inliers of the stretchers surface . . .	72
4.12	Removing points from the scene based on the thermal plane distance filter	73
4.13	Removing the background from a scene	76
4.14	Edges extracted from different sensor streams	77
4.15	Morphological operations to remove distortions from segmentation	79
4.16	Filtering the scene with a Euclidean cluster	80
4.17	Filtering people close to the patient	82
4.18	Sequence in segmentation for a walking subject	84
4.19	Sequence in segmentation for a subject on a medical stretcher	85
5.1	Process of body weight estimation	87
5.2	Different approaches for triangle mesh	88
5.3	Schematic of line-plane intersection	89
5.4	Triangle mesh with front surface triangles and back triangles used for volume estimation	90
5.5	Eigenvalues from a set of points visualized as an ellipse.	94
5.6	Kurtosis for different distributions	96
5.7	Contour and convex hull from the segmented subject	98
5.8	A fuzzy logic approach for the age of a subject	99
5.9	Thermal features	100
5.10	Sequence of someone walking towards the camera.	103
5.11	Schematic for experiment with people walking towards the sensor.	104
5.12	Poses of 14 people walking towards the camera	104
5.13	Correlation ρ between extracted feature vector against the ground truth body weight m	106
5.14	Comparison of of strongly differing subjects with light and heavy, small and tall, male and female, as well as young and old	107
5.15	Pregnant woman	108

6.1	The structure of a single neuron	114
6.2	Common activation functions for a single artificial neuron [80]	115
6.3	Network architectures for regression and classification	116
6.4	Comparison of two networks having different learning rates	120
6.5	Standardization of the feature vector $\mathbf{f} = (f_1, \dots, f_{12})$	121
6.6	Function regression with three different fittings	123
6.7	Relation between bias and variance	123
6.8	Training of networks with increasing amount of units in the hidden layer	124
7.1	Scenes from three different data sets	126
7.2	Distribution of different features from the Hospital with Thermal - Data Set (HT-DS) data set recorded in the trauma room	128
7.3	Results from experiments with different feature groups	133
7.4	Comparison of experiments with a cumulative plot	135
7.5	Weight estimation performed with the Kinect One.	136
7.6	Resized point cloud	137
7.7	Results from subsampling the size of the point clouds	137
7.8	Comparison of <i>Libra3D</i> with related work	139
7.9	Data from the W8-300 data set published by Nguyen et al. [145].	142
7.10	Results from the experiment with people standing in front of the camera.	143
7.11	Results from the experiment with people walking towards the camera	144
7.12	Comparison of body weight estimation with walking and lying subjects.	146
7.13	Regression of Body Mass Index (BMI)	147
8.1	Child from data set	151
A.1	Nearest neighbors in an ordered point cloud (a) and in an unordered point cloud (b).	III
A.2	Different methods to estimate normals for a set of points	V

List of Tables

1.1	Stroke chain of survival	4
2.1	Results for contact-less human body weight estimation from related work	24
3.1	Property table of the used sensors	35
4.1	Timing for different segmentation approaches.	83
4.2	Timing for segmentation for lying subject and a person walking towards the camera.	83
5.1	Statistical values of geometric features	93
5.2	Statistical values from features based on eigenvalues.	95
5.3	Statistical values for the features based on statistics.	96
5.4	Statistical values for contour features.	98
5.5	Statistical values for the age of the subjects. 57 percent of the subjects male and 43 percent are female subjects.	99
5.6	Statistical values for the minimum, maximum and average temperature of the subject, as well as the ambient temperature.	101
5.7	Changes in features with different poses	105
5.8	Comparison of features from subjects with strongly differing characteristics. . .	108
5.9	Tested hardware including time measurements for the estimation	109
5.10	List of features for body weight estimation	111
7.1	Comparison of the applied data sets	127
7.2	Survey about the following experiments with different configurations for the forwarded features.	130
7.3	List of features used for experiments and statistical results	134
7.4	Evaluation of the difference between a subsets for training and testing.	134
7.5	Comparison of weight estimation between Kinect and Kinect One sensor.	136
7.6	Statistics from subsampled data.	138
7.7	Results from comparison with related work	140
7.8	Results from experiments for standing and walking people	145
7.9	Results from the estimation of BMI.	146
7.10	Confusion matrix for BMI classification with three classes.	148
7.11	Confusion matrix for BMI classification with five classes.	148

List of Listings

1	Process of extrinsic calibration.	51
2	Random Sample Consensus (RANSAC) algorithm for the estimation of plane inliers.	71
3	Canny Edge algorithm [38].	78
4	Algorithm for Euclidean clustering [180].	81
5	Euclidean clustering and filtering to improve the outcome of weight estimation based on a set of images.	145

List of Abbreviations

ANN	Artificial Neural Network
BMI	Body Mass Index
CAD	Computer-Aided Diagnosis
CT	Computer Tomography
DNT	Door-to-Needle Time
DOG	Difference of Gaussian
DWI	Diffusion Weighted Imaging
EMS	Emergency Medical Services
E-DS	Event - Data Set
ED	Emergency Department
FDA	U.S. Food and Drug Administration
FOV	Field Of View
FPS	Frames Per Second
GPU	Graphical Processing Unit
GUI	Graphical User Interface
H-DS	Hospital Data Set
HT-DS	Hospital with Thermal - Data Set
HSV	Hue, Saturation and Value
IR	Infrared
ID	Identifier
kNN	k Nearest Neighbors
IV	Intravenous
ICP	Iterative Closest Point
LIDAR	Light Detection and Ranging
LED	Light-emitting Diode
MAE	Mean Absolute Error
MSE	Mean Square Error
MRI	Magnet Resonance Imaging
MySQL	My Structured Query Language

NIHSS	National Institutes of Health Stroke Scale
NHAMCS	National Hospital Ambulatory Medical Care Survey
NaN	Not a Number
OpenCV	Open Computer Vision
PCA	Principal Components Analysis
PCL	Point Cloud Library
PCD	Point Cloud Data
RANSAC	Random Sample Consensus
ROS	Robot Operating System
SSH	Secure Shell
RGB	Red Green Blue
RGB-D	Red Green Blue Depth
ROI	Range of Interest
tPA	Tissue(-type) Plasminogen Activator
ToF	Time of Flight
THN	Technische Hochschule Nürnberg Georg Simon Ohm (Engl. Nuremberg Institute of Technology)
WHO	World Health Organization
W-DS	Walking - Data Set
USB	Universal Serial Bus

List of Symbols

Typographical Convention

x	arbitrary scalar value
$X_{ N }$	arbitrary set with n elements
\mathbf{x}_n	arbitrary vector with n elements
$\mathbf{X}_{n \times m}$	arbitrary matrix with n rows and m columns
$\bar{x}(\mathbf{x})$	mean value value of a set values calculated by $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ where $x_i \in \mathbf{x}$
\tilde{x}	estimated value
\hat{x}	measured value

Constants

g	gravity with a value of $9.81 \frac{\text{m}}{\text{s}^2}$
π	the ratio of a circle's circumference to its diameter with a value of approx. 3.14159
c	speed of light with a value of $299,792,458 \frac{\text{m}}{\text{s}}$

Common

α, β, γ	arbitrary angles
λ	eigenvalue
ρ	density
a	acceleration
d	distance
f	frequency
F	force
i, j, k	indices
l	length
m	body weight
\mathbb{N}	set of natural numbers
\mathbb{R}	set of real numbers

r	radius
s	surface area of an object
T	temperature
t	time
v	volume of an object
\mathbf{v}	eigenvector
w, h, d	width, height, depth

Statistics

$\sigma(\mathbf{x})$	standard deviation of a set of values \mathbf{x}
ϵ	relative error
ρ	correlation coefficient
e	absolute error
p	probability
$x_{\max}(\mathbf{x})$	maximum value in a set of values \mathbf{x}
$x_{\min}(\mathbf{x})$	minimum value in a set of values \mathbf{x}
\mathbf{x}_{std}	standardized set of values based on a set of values \mathbf{x}

Sensor Processing

$\delta_{\mathbf{t}}$	model für tangential distortion
ϵ	thermal emission
λ	wavelength
ϕ	phase shift
ρ	thermal reflectivity
Σ	covariance matrix
τ	thermal transmission
${}^A\xi_B$	relative pose from the coordinate frame $\{B\}$ w.r.t. to a frame $\{A\}$
$\{A\}$	coordinate frame of a sensor, here known as sensor A
a	area
C	a contour is defined as set of pixels \mathbf{q} from an image \mathbf{I}
D	data set, e.g., a set of depth images \mathbf{D}
\mathbf{d}	distortion vector $\mathbf{d} = (k_1, k_2, \dots, p_1, p_2, \dots)$, containing the radial (k_1, k_2, \dots) and tangential coefficients (p_1, p_2, \dots)
E	edge in an image marked with pixels $\mathbf{q} \in \mathbb{R}^2$
f	focal length
$\mathbf{H}_{4 \times 4}$	homography matrix \mathbb{R}^4

h	Hessian plane model defined by $\mathbf{h} = (\mathbf{n} \ d)^T = (n_x \ n_y \ n_z \ d)^T$
I	image containing pixels $\mathbf{q}(u, v) = (r \ g \ b)^T$ in \mathbb{R}^3 containing color for red r , green g , and blue b
l	line $\in \mathbb{R}^3$
M	set of triangles $M = \{T_1, T_2, \dots, T_N\}$ forming a triangle mesh $\in \mathbb{R}^3$
\mathcal{N}_k	set of k-nearest neighbors of a point
n	normal of a point defined by $\mathbf{n} = (n_x \ n_y \ n_z)^T \in \mathbb{R}^3$
\mathcal{P}	set of point grouped as point cloud $\mathcal{P} = \{\mathbf{p}_0, \mathbf{p}_1, \dots\} \in \mathbb{R}^3$
p	arbitrary point in $\mathbf{p} = (x \ y \ z)^T \in \mathbb{R}^3$
$\mathbf{P}_{3 \times 4}$	projection matrix for the pinhole camera model $\mathbf{P}_{3 \times 3} = \begin{pmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}$ where γ is the skewness of the image axis, f_x and f_y are the focal length, and u_0 and v_0 define the center of an image
q	arbitrary pixel in $\mathbf{q} = (u \ v)^T \in \mathbb{R}^2$
$\mathbf{R}_{3 \times 3}$	rotation matrix in \mathbb{R}^3
s	scaling value
T	a triangle defined by three points $T = \{\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k\} \in \mathbb{R}^3$
$\mathbf{T}_{4 \times 4}$	homogeneous transformation matrix $\mathbf{T} = \begin{pmatrix} \mathbf{R}_{3 \times 3} & \mathbf{t}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{pmatrix} \in \mathbb{R}^3$, containing rotation \mathbf{R} and translation \mathbf{t}
$\mathbf{t}_{3 \times 1}$	translation vector $\mathbf{t} = (t_x \ t_y \ t_z)^T \in \mathbb{R}^3$

Neural Network

α	constant factor for the learning rate of a network
η	learning rate for backpropagation
δ	responsibility of a neuron
a	output of an artificial neuron without activation function
b	bias of an artificial neuron
E	error of the networks output, depending on the output of a network \mathbf{u} and the target values \mathbf{t}
$g(\cdot)$	activation function of an artificial neuron
\mathbf{t}	vector containing target values $\mathbf{t} = (t_1, \dots, t_n)$
\mathbf{u}	output of a neural network
w	weight of an artificial neuron

Chapter 1

Introduction

Although in some tasks computers outperform the human brain, for example in mathematics, the brain can easily outperform a computer in tasks like perception. While we know precisely which part of a computer is responsible for a certain task, this is not clear in the case of the brain. Therefore, researchers are looking for a way to map the brain and its functions [208].

1.1 The Human Brain

Before the moment of birth, the brain controls and monitors all other organs in the human body. These organs can be seen as the actors of a machine. Without the brain, unconscious tasks like breathing would not be possible. The lungs would not know how to act after birth, and therefore the body could not receive the necessary oxygen. The heart is also triggered by the brain to beat in a given way. If higher blood flow is needed, the brain signals the heart to increase its pumping rate. Conversely, it gives the signal to slow down when someone is resting. Conscious motions augment these ongoing controls. To grasp a glass of water, several muscles must be monitored at the same time. Within the control loop, feedback is gathered by the sense of touch in the tips of the fingers. Thus, grasping small and fragile objects is possible without breaking them. Speaking a single word is achieved by controlling 70 muscles, including those of the lips and tongue, even for simple words [4].

The brain receives many external signals, and all signals from the senses are processed to enable reactions. Vision, hearing, smell, taste, and touch are used to perceive the environment. Furthermore, the brain is responsible for emotions. Being stressed, happy, frustrated or angry are signs of ongoing processes caused by the environment. On the one hand, emotions might be shown with facial expressions, mimicry or sounds [60]. On the other hand, they might also exist only in the awareness of the person experiencing them.

The brain is not perfect from the beginning: it never stops learning and it remembers external influences. Walking, speaking or learning to interpret someone's feelings by their facial expression are skills achieved by learning. The experiences remembered by the brain affect personality and likes and dislikes. One person might be interested in creative tasks, such as art and music, while someone else might be more interested in sports or research.

The brain needs certain nutrients, especially in fetal or early postnatal life [219]. The nutrients should contain carbohydrate, protein, fat, minerals and vitamins to achieve performance. The brain needs one-fifth of the oxygen demand in the human body [175]. The blood flow via the carotid artery supplies the brain with oxygen while removing carbon dioxide as a product of metabolism. A continuous blood flow is required to provide the brain with oxygen. If this blood flow is impaired, the brain loses its energy supply. Therefore, it cannot fulfill its tasks.

1.2 Stroke

A stroke is the sudden death of brain cells in a localized area due to inadequate blood flow. Annually, more than 15 million people around the world suffer a stroke. Approximately one third of these people die, and one third remain permanently disabled [136].

Stroke can have different causes. The two most common types of stroke are ischemic stroke and intracranial hemorrhage. About 87 percent of strokes are ischemic strokes with an occlusion of brain vessels caused either by local thrombosis or embolism (see Figure 1.1a) [41]. Early therapeutic strategies aimed to dissolve the underlying clot and to restore sufficient blood flow to the affected parts of the brain. This included systemic intravenous thrombolysis and interventional approaches with mechanical thrombectomy. In the case of intracranial hemorrhage (see Figure 1.1b), a vessel within the brain ruptures. As well as the resulting critical shortage of blood supply, the accumulation of blood within the inflexible skull may create excess intracranial pressure [95]. Therapeutic strategies focus on prevention of further hemorrhage and control of intracranial pressure. Depending on the cause, size and localization of the hemorrhage, this involves either medical or surgical approaches [201].

Initially, some stroke patients may not even realize that their symptoms are related to an acute stroke since they or their relatives do not know the typical stroke symptoms. In contrast to a cardiac infarction, stroke symptoms are usually not accompanied by pain. Temporary symptoms, e.g., in the case of a transient ischemic attack, might be underestimated or not taken seriously at first. In addition, some patients might not be able to communicate their symptoms or to call for help because of speech disorders such as aphasia. Therefore, many stroke patients receive very late or insufficient treatment for acute stroke. About 50 percent of all emergency calls were made within the first hour after the onset of stroke symptoms [140]. Furthermore, the National Hospital Ambulatory Medical Care Survey (NHAMCS) demonstrated that only 53 percent of

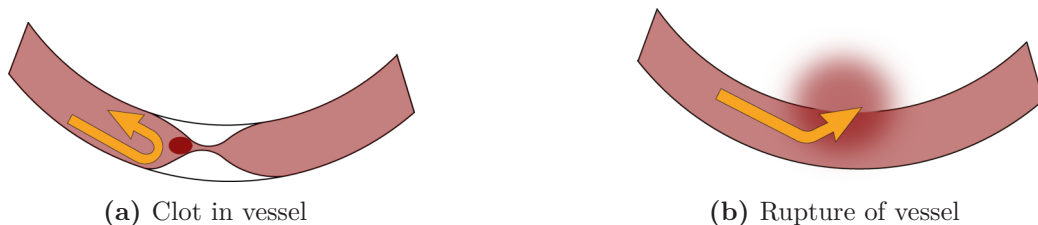


Figure 1.1: Ischemic and hemorrhagic stroke: In the case of an ischemic stroke, a blood clot in a vessel interrupts the blood flow (a). In the event of a hemorrhagic stroke the vessel is ruptured, causing bleeding in the brain (b). Both scenarios result in a reduced supply of oxygen to the surrounding brain tissue.

all stroke patients use Emergency Medical Services (EMS), although the treatment of patients brought by EMS starts earlier and leads to better chances of recovery [137]. In contrast to a cardiac infarction, people involved can give limited aid to the person having a stroke. Even an emergency physician can only give limited help outside a hospital setting with adequate imaging modalities, e.g., Computer Tomography (CT) or Magnet Resonance Imaging (MRI) scanners.

Typical stroke symptoms are sudden hemiparesis, one-sided palsies or numbness in the face or one leg or arm. Further symptoms include sudden confusion, trouble speaking or difficulty understanding speech, sudden vision disorder in one or both eyes, double vision, sudden difficulty in walking, dizziness, loss of balance or lack of coordination. In rare cases, sudden severe headache with no known cause can indicate an acute stroke. Each of these symptoms can occur on its own, although a combination of them is possible and common. Particular symptoms can indicate the brain area where the stroke is located, e.g., a stroke in Broca's area, which is responsible for speaking, may lead to severe speech disorders. In addition, knowledge of grammar can be lost, and patients may have problems forming correct sentences. In contrast to that, damage in Wernicke's area can lead to a loss of understanding of speech. Affected people might also have problems assigning noises to their sources, e.g., a car driving by [215, 193].

Typical risk factors for ischemic stroke are age above 60 years, high blood pressure, and other cardiovascular diseases. In addition, lifestyle is one of the highest risk factors. Heavy alcohol usage and smoking increase the risk of having a stroke. Nutrition containing a large amount of fat can lead to high cholesterol and obesity, both primary risk factors for stroke. The incidence of stroke is declining in many industrial and developing countries because of better health care or regulation of smoking and alcohol consumption. However, the number of strokes increases with an aging population, higher life expectancy and increasing comorbidities in this age group. The average age for suffering a stroke is 72 years with a standard deviation of around 12.1 years [6].

In the case of an ischemic stroke, doctors focus on dissolving the vessel-occluding blood clot as fast as possible. In this regard, it is crucial to optimize emergency procedures, including pre-hospital and in-hospital procedures. In this context, the Door-to-Needle Time (DNT) is the most important benchmark parameter for in-hospital processes. It starts when the patient arrives at the hospital, passing the entrance, and it stops when the patient receives treatment to dissolve the blood clot via Intravenous (IV) lysis. Within this time, physicians take an anamnesis, perform a short neurological examination, take blood samples and perform brain imaging via CT or MRI to check whether the patient has had a stroke or whether another disease is causing the symptoms. To reduce the DNT, hospitals set up specialized stroke teams for treatment. When an ambulance transports a patient with stroke symptoms, pre-notification of the receiving hospital staff is a top priority in order to improve the DNT, as shown by Learmonth et al. [116]. The stroke team can then prepare for the newly arriving patient according to the recorded symptoms and the patient's medical history. A medical imaging device – CT or MRI – can be pre-reserved for this patient, so that the team does not lose valuable time for diagnostics. International guidelines recommend a DNT of less than 60 minutes [66, 103]. However, this recommendation often cannot be achieved because of a delay in diagnosis or treatment. Bray et al. [29] conclude that larger hospitals with specialized physicians can provide faster – and for this reason better – stroke treatment. Larger hospitals treat more patients over the year, and so the physicians have more routines for stroke treatment [195, 194]. To standardize the treatment of strokes,

and therefore improve the outcome, Jauch et al. [95] note eight essential steps for the correct treatment of stroke patients. Table 1.1 presents these eight steps.

1.2.1 Diagnosis

Relatives or physicians can mistake the symptoms of a stroke for other issues. Therefore, an attending physician must check the anamnesis. One example of a mimicking disease is hypoglycemia. For a patient with a history of diabetes, the blood glucose might be an indicator [95]. Another example is Wernicke’s encephalopathy. Due to the lack of vitamin B1, paralysis of motion or speech can occur. This disease often occurs in patients having a history of alcohol abuse [95]. Mandatory tests for the diagnosis of a stroke are given by the ”Guidelines for the Early Management of Patients With Acute Ischemic Stroke”, published by the American Stroke Association [95, 170]. Not all test results may be available by the time treatment starts. Nevertheless, the medical imaging, the level of blood glucose and the level of oxygen saturation are mandatory for starting treatment [95].

Doctors can only distinguish the different types of stroke by medical imaging. Figure 1.2 shows a CT and an MRI scan of an ischemic stroke. In the non-contrast CT image, the ischemic stroke can hardly be seen. In contrast to that, the MRI with Diffusion Weighted Imaging (DWI) shows a clear loss of gray matter in the right hemisphere. Kang et al. [100] showed in a small study that CT-based screening has an average DNT of 67.5 ± 22.5 minutes. In contrast, MRI-based screening leads to an average DNT of 86.8 ± 21.5 minutes. However, the outcome of screening did not vary significantly. Additionally, CT scanners are more common than MRI devices. Due to the strong magnetic field produced by the MRI, the device is not easy to access and some patients have to be excluded from it. Patients with pacemakers, aneurysm clips or other ferromagnetic material in their bodies cannot be safely imaged via MRI [207].

Radiologists distinguish between two states of damaged brain tissue: the umbra (infarct core) is dead brain tissue which cannot recover from the damage. The brain cells in this area are necrotic and perform no function for the body. The adjoining damaged brain tissue is called the penumbra, and here, tissue can potentially recover from damage if blood flow and nutrition are resumed in time; otherwise, the penumbra dies, and this brain tissue is also lost.

Table 1.1: Stroke chain of survival by Jauch et al. [95].

detection	patient or bystander recognition of stroke signs and symptoms
dispatch	immediate activation of emergency call and priority EMS dispatch
delivery	prompt triage and transport to most appropriate stroke hospital and pre-hospital notification
door	immediate ED triage to high-acuity area
data	prompt ED evaluation, stroke team activation, laboratory studies and brain imaging
decision	diagnosis and determination of most appropriate therapy, discussion with patient and family
drug	administration of appropriate drugs or other interventions
disposition	timely admission to stroke unit, intensive care unit or transfer

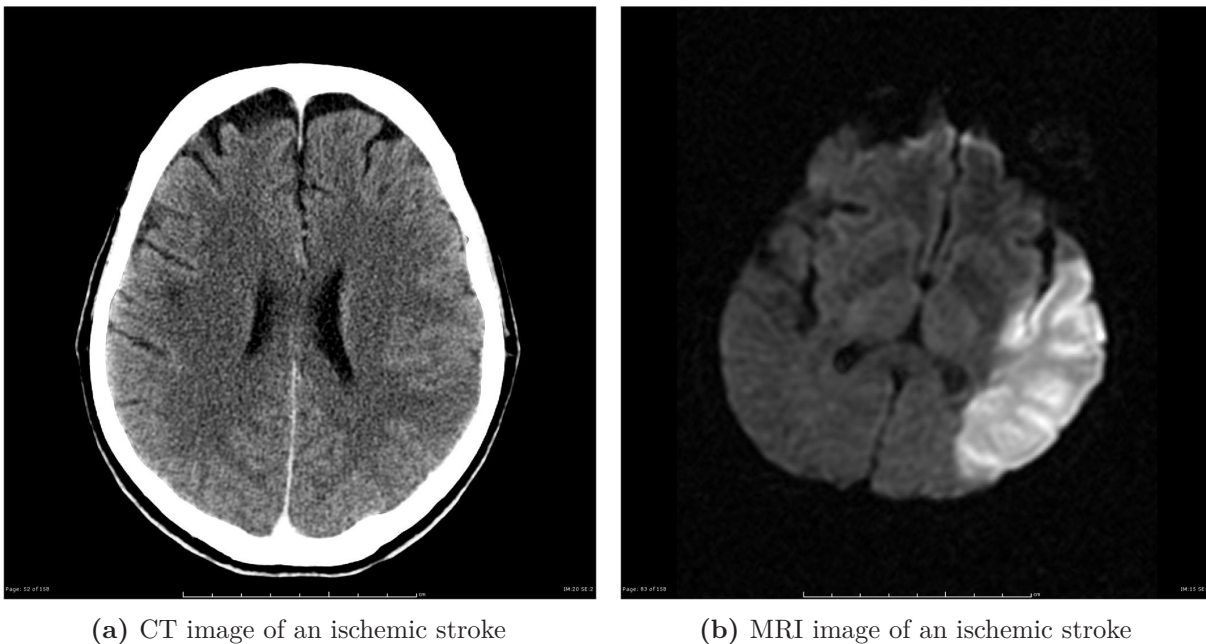


Figure 1.2: Radiography of an ischemic stroke from CT and MRI with DWI imaging: While the ischemic stroke is hardly visible in the CT image, the infarct is clearly visible in the MRI image. Source (a,b): Case courtesy of A. Prof. Frank Gaillard, Radiopaedia.org, rID: 21849 [174].

To clinically quantify different degrees of stroke severity, several stroke measurements exist. The National Institutes of Health Stroke Scale (NIHSS) is the most common scale for gauging the level of stroke severity [111]. The symptoms of the patient are described in eleven categories, giving a certain score in each category. The protocol is repeated multiple times within the first days at a stroke unit, to check if the stroke symptoms improve but also to detect possible neurological deterioration. Furthermore, assessing stroke severity in this way allows prediction of stroke outcome. By adding the patient's medical history to the NIHSS score, even mortality can be predicted [195].

1.2.2 Treatment of Ischemic Stroke with tPA

Approved in 1996 by the U.S. Food and Drug Administration (FDA), Tissue(-type) Plasminogen Activator (tPA) is still today's state-of-the-art medication for IV thrombolysis of acute ischemic stroke [232]. The enzyme tPA improves the conversion of plasminogen to plasmin within the coagulation system. Plasmin degrades many blood plasma proteins, including fibrin (fibrinolysis) and is therefore responsible for the breakdown of a blood clot. The time window for treatment with tPA is narrow. The probability of a good outcome and the numbers of patients that can potentially benefit from the treatment decrease with time. Initially, CT-based tPA treatment was approved only within three hours of stroke symptom onset. Based on study results, systemic IV thrombolysis is now approved within the first 4.5 hours after symptom onset [76]. Ideally, the dose is given as soon as possible, e.g., in the first hour after symptom onset [183]. This narrow



(a) CT scanner, Source: [198]



(b) MRI scanner, Source: [127]



(c) Mobile stroke unit in an ambulance, Source: [200]

Figure 1.3: Medical imaging devices to diagnose strokes: The fastest way to determine if a patient has an ischemic or hemorrhagic stroke is via a CT scan (a). In contrast, diagnosis based on MRI (b) is time-consuming and not suitable for every patient due to the magnetic field. Nevertheless, the quality of the acquired images is superior compared to CT. To reduce the time between the first appearance of the symptoms of a stroke and the confirmed diagnosis, small head CT scanners are integrated into special ambulances (c).

time window is often called the *golden hour*. In an extended time window, adverse effects such as internal bleeding may dominate if the blood clot is dissolved and blood flow is restored, but the brain tissue is already irretrievably destroyed. In addition, an intracranial hemorrhage can appear. Furthermore, the attending physician needs to know the pre-existing conditions of the patient, if possible. Recent surgery within the last weeks, internal bleeding or the intake of a blood thinner all lead to exclusion from treatment with tPA. If tPA is given to a patient for whom the drug is not suitable, it can cause internal bleeding. Nevertheless, it is the treating physician's task to decide in each case which treatment is most appropriate for a particular patient. Since the release of tPA, several guidebooks for the treatment have been released [95, 2]. Every year the American Heart Association releases the statistics for stroke [142, 18], providing the current trends in the origin of strokes and guidelines for treatment. For carefully selected ischemic stroke patients, (for example, with evidence of a proximal intracranial vessel occlusion, small infarct core and large penumbra volumes), mechanical thrombectomy provides an additional treatment opportunity [99, 7]. A catheter is used to position a stent to break up the clot mechanically and to retrieve it from the vessel. Thrombectomy can also be considered if patients have to be excluded from treatment with systemic tPA.

Several approaches exist to improve success in stroke treatment and health care has improved over the last decades. Better medical imaging using MRI and CT enhances the detection of strokes. Therefore, physicians can calculate the risks of different treatments and medications. Figure 1.3a and Figure 1.3b show the two common medical imaging devices. New drugs are under investigation to dissolve blood clots even faster with fewer adverse effects. Highly specialized stroke units have been established in hospitals since 1990 in Germany, to accelerate the treatment of stroke patients. These stroke units are led by experts with high access to monitoring and medical imaging equipment. Today, there are even mobile stroke units to accelerate the treatment further. In an ambulance equipped with a small head CT scanner, physicians can start the treatment on the way to the hospital, as shown in Figure 1.3c. If a doctor is not sure about

treatment, several physicians can be consulted using telemedicine, sharing necessary patient data and vital signs [200].

Studies show that faster treatment might achieve the biggest gain in stroke treatment [95, 118]. The quotation *time is brain* refers to the rapid loss of brain tissue without treatment [182]. An average patient loses 1.9 million neurons for each minute the stroke remains untreated. The process of care can improve the outcome of stroke significantly, as shown by Langhorne et al. [114].

1.3 Body Weight Adapted Dosing

One of the biggest challenges for a physician in applying tPA to a patient is determining the correct dose. The dosage must be adapted to the patient's body weight, with 0.9 mg/kg. A maximum dose of 90 mg is set for patients heavier than 100 kg. Common weighing devices, such as scales, are hard to integrate into the process of treatment in an emergency room. Furthermore, many stroke patients cannot stand up for weighing due to their symptoms. In addition, severe injuries or motor symptoms prohibit easy weighing procedures for many patients. In a registry with 27,910 stroke patients, only 14.6 percent were weighed [56]. Furthermore, asking the patient for his or her own body weight can be problematic. In clinical routine, many emergency patients are unable to communicate information on their body weight because of their symptoms, for example, decreased consciousness or neurological disorders. They may simply not know their body weight [31]. Elderly patients might suffer from dementia and therefore are not able to provide a reliable value for their body weight. Diseases like ischemic stroke entail a very narrow time window for treatment and do not allow for the time needed to weigh each patient in an emergency situation. Therefore, visual estimation of the patient's body weight by the attending physician in the emergency room has become a common routine. This approach carries the risk of estimation errors [42, 46, 135, 130, 63] and may result in dosing errors, which have been shown to occur in weight-based emergency medication [37, 68]. A physician can guess a patient's weight more accurately if they are similar regarding age, gender, and constitutional type. Furthermore, it is better to average the body weight estimation, with several doctors making a suggestion independently. Nurses are often better at estimating weight than doctors [31].

Less complicated and more precise methods for evaluating body weight are required for emergency patients, to minimize potential dosing errors.

1.4 Objectives and Contributions

The contributions of this thesis are presented to conclude the introduction. Within the thesis, the emphasized questions will be answered concerning the topic of visual body weight estimation.

In 2012 the Technische Hochschule Nürnberg Georg Simon Ohm (Engl. Nuremberg Institute of Technology) (THN) was asked by Siemens AG in a small mission-oriented research study to demonstrate whether body weight estimation for emergency patients can be achieved with a 3D camera. A single Microsoft Kinect camera was used, mounted over the patient lying on a medical stretcher. Early experiments, working with background subtraction and a familiar environment, achieved adequate results. However, the scenario was very limited, with a fixed position for the

camera as well as a fixed position and background for the patient lying on the stretcher. In addition, the approach was only tested for around 10 people, mostly THN students lying on a stretcher. Therefore, the first question is as follows.

- *Does a low-cost consumer camera achieve enough accuracy for body weight estimation?*

The Microsoft Kinect was released in December 2011 as a consumer 3D camera achieving similar accuracy as state-of-the-art sensors, with a price of around 100 euros. After the release, publications about 3D perception, registration, and scene reconstruction increased significantly. Having such a low-cost sensor for the application of body weight estimation would ease the clinical integration, although the sensor was not certified for clinical usage at that time. Other comparable sensors with a certified standard for the industry can have a price several times higher than the Kinect. In addition, some of them, such as ToF cameras, are not suited for medical applications because their high emission of light means that they do not meet the eye-safety requirements for the usage outside of a laboratory [157].

As already mentioned, it is still state of the art for physicians to estimate the body weight of a patient visually. With an error rate of around one third, physicians are aware of the fact that visual body weight estimation is not sufficient. To date, a reliable and fast approach for weight estimation in stroke patients has been missing.

- *Is a visual body weight estimation more reliable than state-of-the-art methods?*

The clinical usage indicates the value of this thesis for real patients from the emergency room. This data set reflects a wide variety of different patients, body weights, shapes, ages and more. The direct collaboration with the Department of Neurology at the University Hospital Erlangen and with Siemens Healthcare GmbH is ideal for such a research project, bringing the theoretical results into real health care applications.

During this research, when experiments showed increased performance over the first two years, people at conferences often asked if the presented method would work for standing or even walking people. Weighing standing people can easily be done on standard spring scales. However, the automatic weighing of several people in a short time could offer a benefit in some applications. Since 2017, the Finnish airline *Finnair* has weighed passengers to obtain the total weight of an airplane for take-off. While the weight of baggage is measured on scales, the weight of the passengers is only roughly estimated using standardized weights [172]. Precise knowledge about the weight allows the possibility of optimizing fuel requirements and therefore operating costs [62]. In 1985, a McDonnell Douglas DC-8 jetliner crashed with 256 people on board. One reason for the crash might have been the underestimation of on-board weight, as was mentioned in the occurrence report [26]. Furthermore, the motivation for a visual weight system is increased since objects that the subject is wearing or carrying, e.g., a backpack, can be filtered out for weight estimation.

The body weight can further be used to improve the outcome in identification [3]. While most soft biometrics can be altered quickly with little effort, for example, dyeing the hair or putting on heavy makeup, the body weight cannot be changed immediately. In addition, the shape of the body is visible from a distance, in contrast to small soft biometrics such as the color of the eyes [49]. Therefore, the contribution in the clinical setting with the patient lying on a stretcher is extended towards a more generalized approach.

- *How reliable is the estimation of body weight for standing or walking people?*

In summer 2017, THN was contacted by the radiology unit of a local municipal hospital. One of the employees had read about the system in the newspaper and was interested in whether the system could also be used to estimate the BMI of lying subjects. It is common to forward the BMI of a patient to the CT scanner to reduce the radiation dose to a minimum. Patients with a low BMI can be scanned using less radiation to get an adequate image, while a higher BMI demands a higher radiation dose. Receiving a high dose of radiation from a CT scan in childhood can increase the risk of leukemia [156]. Conceptually, few adaptations would be required to integrate the system into CT scanning, compared to the stroke scenario.

- *Can the approach estimate the BMI of a patient on a stretcher?*

The final contribution to this thesis consists of the recorded data sets. These data sets allow other researchers to contribute to the field of visual body weight estimation. The data sets contain the recorded sensor data, the ground-truth body weight and some soft biometrics.

The approach developed here provides a novel method for body weight estimation with the focus on clinical application, by combining state-of-the-art algorithms from image processing and machine learning. The funded project – called *Libra3D* – started in October 2014 and ended in September 2016. Project partners were THN, the University Hospital Erlangen, Germany and Siemens Healthcare GmbH, Kemnath, Germany. Most of the contributions listed here were published as conference papers during the project.

1.5 List of Publications

The following papers were published during the last four years of this study. The content of these papers is included in this thesis and marked with references. If necessary, the papers are cited within the text.

Journal Articles

- Christian Pfitzner, Stefan May and Andreas Nüchter: Body Weight Estimation for Dose-Finding and Health Monitoring of Lying, Standing and Walking Patients Based on RGB-D Data, *Sensors* 2018, 18, 1311. [161]

Conference Proceedings

- Christian Pfitzner, Stefan May and Andreas Nüchter: Evaluation of Features from RGB-D Data for Human Body Weight Estimation, In Proceedings of the 20th World Congress of the International Federation of Automatic Control, 9-14 July 2017, Toulouse, France, 2017. [162]
- Christian Pfitzner, Stefan May and Andreas Nüchter: Neural Network-based Visual Body Weight Estimation for Drug Dosage Finding. In Proceedings of the SPIE Medical Imaging Conference on Image Processing, San Diego, CA, USA, March 2016. [164]

- Christian Pfitzner, Stefan May, Christian Merkl, Lorenz Breuer, Martin Köhrmann, Joel Braun, Franz Dirauf and Andreas Nüchter: Libra3D: Body Weight Estimation for Emergency Patients in Clinical Environment with a 3D Structured Light Sensor, In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, May 2015. [163]

Attended Ph.D. Forum

- Christian Pfitzner. Robotic Vision in Medical Applications: Visual Weight Estimation for Emergency Patients. Workshop Proposal to the Ph.D. Forum at the International Conference on Robotics and Automation (ICRA '15), Seattle, WA, USA, May 2015. [160]

Patents

- Christian Pfitzner, Stefan May and Christian Merkl: Vorrichtung und Verfahren zur optischen Erfassung eines Gewichtes einer Person, Deutsches Patent. Submission date: February 29th 2016. [158].

Data Sets

- Pfitzner, Christian. RGB-D(-T) Data sets for Body Weight Estimation of Stroke Patients from the Libra3D Project. Open Science Framework, June 20th 2017. [159]

1.6 Structure of this Thesis

For the reader's convenience, each chapter has a summary at the end of each section. The thesis is structured as follows.

- *Chapter 2* discusses related work on body weight measurement. Considering common body weight measurement devices and their principles, this chapter also gives an insight into body weight estimation with a focus on its applicability in emergency situations. Furthermore, the state of the art presented here is structured by body weight estimation for lying, standing, and walking subjects.
- *Chapter 3* shows the clinical environment and details of experiments are described. In addition, the sensor concept is discussed and examined with regard to its characteristics and the benefits of a visual body weight estimation system. Various characteristics of the applied sensors are compared and sensor fusion of three sensors is explained.
- *Chapter 4* illustrates the segmentation of the patient on the stretcher from the environment. Filters and common segmentation approaches based on the fused point cloud are presented.
- *Chapter 5* shows the extraction of selected features and the calculation for body weight estimation. The features are extracted from the segmented point cloud of the patient.

-
- *Chapter 6* presents the estimation of the body weight based on the previously extracted feature vector and an Artificial Neural Network (ANN). The structure of the applied ANN is also presented, as well as the process of learning with improved generalization based on recorded data sets.
 - *Chapter 7* presents the results for body weight estimation, based on different data sets recorded from people lying on a stretcher in a trauma room. Furthermore, the results of *Libra3D* are compared to the accuracy of weight estimation by physicians, the patient's self-estimation and an established anthropometric method based on body height, waist circumference, and hip circumference. Additional experiments with people standing and walking in front of an RGB-D camera demonstrate the algorithm's versatility. In addition, the sensor's characteristics are tested for accuracy of estimation by reducing the size of the point cloud.
 - Finally, *Chapter 8* concludes with a discussion of future work and possible improvements. Future applications based on the algorithm presented here are illustrated.

Chapter 2

Related Work

The unit kilogram is the last SI unit which is not defined by a natural constant. Since 1889, a cylinder made of a platinum-iridium alloy has represented this SI unit. The cylinder should have the same weight as water filling a rectangular volume of $0.1 \times 0.1 \times 0.1$ cubic meters with a temperature of 4°C , where water has the highest density [110]. Based on this cylinder, several copies were manufactured to calibrate sets of scales. However, the cylinder which is stored in the International Bureau of Weights and Measures (French: Bureau International des Poids et Mesures), protected under three glass bells, changed weight over time [51]. Comparing the first kilogram with its copies, either it became lighter or the copies became heavier. Damage, abrasion or oxidation can change the weight. Research today is pursuing a less problematic definition of the kilogram. One method could be via the Avogadro experiment, where the kilogram can be defined by a fixed number of atoms in an isotope [12]. Another possibility could be the watt balance, where the weight is related to the electrical power [202, 167]. At the General Conference on Weights and Measures in 2018, researchers will try to define the kilogram based on a new method; probably one of the two presented approaches.

This chapter describes state-of-the-art weight measuring devices and methods for weight estimation applicable for medical use with a focus on stroke patients.

2.1 Traditional Weight Measurement

Measuring the weight of an object can be achieved via various physical principles. The most common method is to use scales, although these can operate via different principles. One of the oldest principles is the direct comparison with a given weight. This type of mechanical balance can only show a difference in weight with respect to a given reference weight. Figure 2.1a shows such a balance. The object to be weighed is placed on one of the scale pans, which will then move downwards due to gravity. On the empty pan, weights are added until both scale pans are level. Often, an additional tongue is applied to the scale to improve accuracy and comfort for the user when checking whether both pans are level. To obtain the weight of the object, all the weights on the other pan must be summed. The accuracy of such a scale is limited due to friction and the discrete subdivision of the reference weights, for example, 1 g, 10 g or 1 kg.

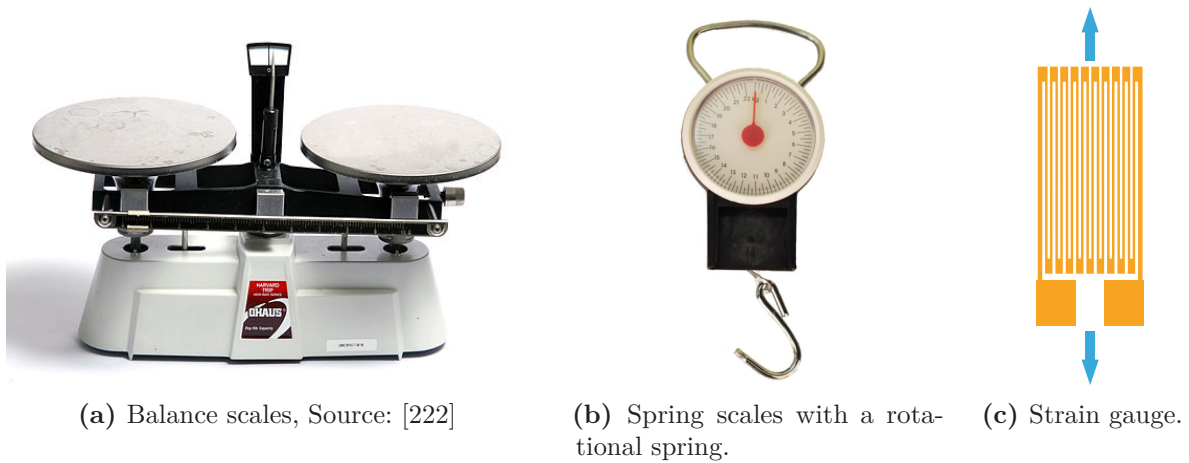


Figure 2.1: Different principles in weight measurement: The balance scales indicate whether the objects on both pans have the same weight with the tongue in the middle of the scale (a). To obtain the weight of an object, the other pan is filled with known reference weights. Another approach to measuring weight is the use of spring scales (b). Often, modern scales use strain gauges (c): under tension or pressure, a strain gauge changes its length and therefore its electrical resistance. This change is measurable using a microcontroller with an analog-digital converter.

Another mechanical approach is weight measurement using a spring. With a linear spring extension rate and an applied force, the spring extension corresponds linearly with the weight [48]. Spring scales can work on different principles, for example, pull springs, compression springs or torsional springs. Pull and compression springs work because the weight force accelerates an object with $g = 9.81 \frac{\text{m}}{\text{s}^2}$ towards the ground, with force $F = m \cdot g$. A torsional spring acts with the help of the spring pressure, which is defined by $F_s = F_{\text{max}} \cdot \frac{\alpha}{\alpha_{\text{max}}}$, where F_{max} is the maximum spring pressure, α is the angle of rotation and α_{max} is the maximum angle. Figure 2.1b shows a scale with a pull spring. Spring scales easily reach an accuracy of 0.1 kg. The accuracy of all mechanical scales is limited because of friction or non-linear spring extension rates [122].

Today's most common method of weight measurement is using electronic scales. Resistance strain gauges react to changes of pressure and therefore change their resistance value. Figure 2.1c shows a schematic of a strain gauge. The conductive trails on a strain gauge contract under pressure and expand when the pressure is released. The limitation in accuracy is determined by the linear characteristics of the resistance strain gauge and the manufacturing process used. A set of electronic scales has the advantage that the weight can be filtered in time, e.g., an averaging window can be applied over time to minimize the impact of someone stepping on the scales. Further, the weight can be forwarded to a display or processed for transmission to another system. Based on the principle of strain gauges, several modern possible weighing devices for people exist. Standing scales represent one of the most common types of scales. Figure 2.2a shows a set of standing scales used for medical observations.

With the focus on stroke patients, this method has the disadvantage that the patient must stand independently for successful weight rating. As an alternative to standing scales, chair scales exist, developed for seniors to be weighed while sitting. The weighing process is eased for

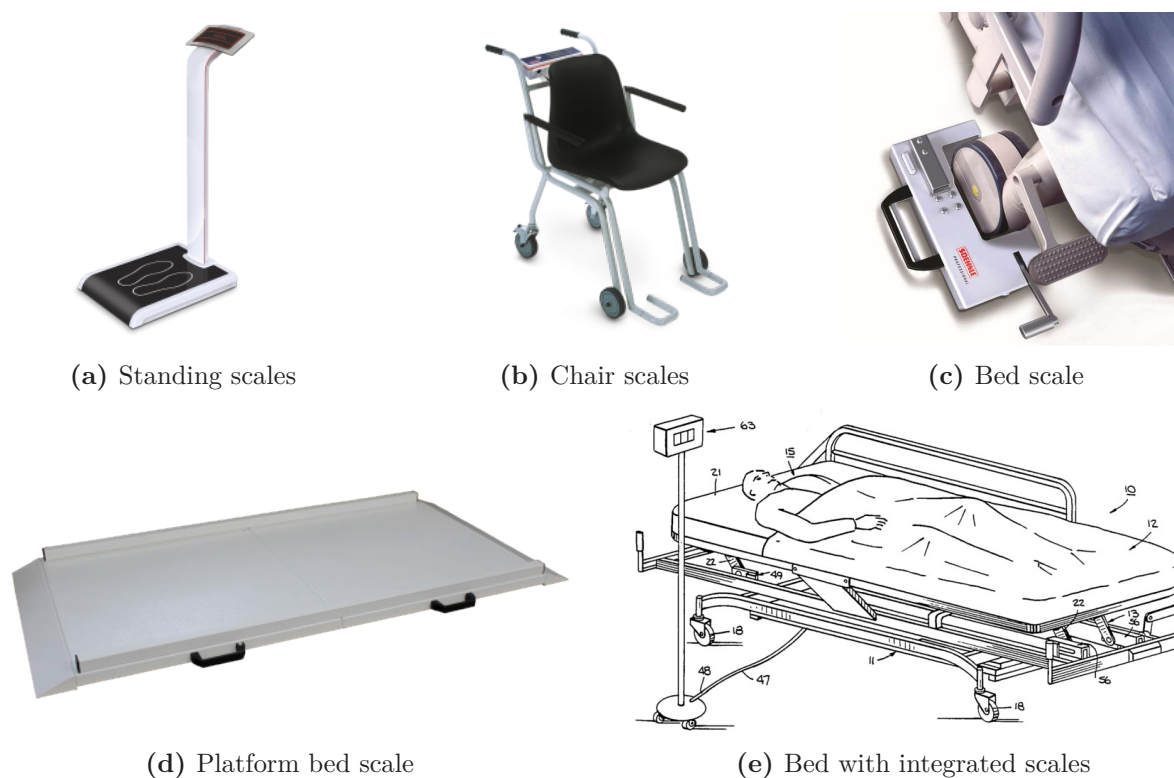


Figure 2.2: Different devices for weighing in the hospital environment: The most common way to obtain the body weight is with a set of standing scales if the patient is able to stand (a). If not, chair scales (b) are an alternative to balancing. If the patient is already lying in bed and cannot be moved, bed scales are the best solution to obtain the body weight (c-e). Source (a-d): [196]; Source (d): [204]

immobile patients. However, unconscious persons cannot be weighed with these types of scales. Figure 2.2b shows a set of chair scales for clinical usage.

The weights of patients reclining on a stretcher or patient table can be acquired in principle via different kinds of force sensing. Scales situated on the floor which can weigh the whole bed with the patient on it are available for use in hospitals. These scales measure the vertical force of gravity. Bed scales exist with different constructions. Fixed bed scales are integrated into the ground forming a rectangular weighing area. Mobile rectangular bed scales also exist, see Figure 2.2d. These can be set up by nursing staff, as they have a small ramp for moving the bed into the weighing area. The setup time for these is one to two minutes, which can be crucial if time is short. A third category consists of four mobile units which are attached to the bed. By pressing a lever, the bed is raised a few millimeters, and the scale value on each wheel is summed to obtain the total weight. Figure 2.2c illustrates such a set of scales with a weighing unit for each wheel [197]. Unfortunately, this type does not work with all medical stretchers. If there are large wheels, the weighing devices might not fit, or the stretcher may have an additional fixed fifth wheel in the middle under the bed, to ease steering. There is a high uncertainty in the patient's weight determination with all types of bed scales, as the exact tare for all stretchers with their individual accessories is usually not known precisely. Additional parts of the bed might

be attached, such as handlebars, infusion bags or other medical monitoring devices – all of these would increase the error in measurement if a default weight for a bed is used. The only adequate way to minimize errors for bed scales is to measure the empty bed first, set the tare weight and then place the patient on the bed. However, this procedure is time-consuming, and the patient has to be moved.

Swersey [204] considers a set of scales which is integrated into the bed, weighing only the reclining area including the mattress and the stretcher. The effect of different attachments is therefore minimized. Figure 2.2e shows a schematic of the bed with the integrated scales.

To reduce the effect of the different tare weights of beds, weighing bedspreads consisting of several strain gauges arranged in a grid pattern are applied. The bedspread is placed between the mattress of the stretcher and the patient. The impact of a potentially incorrect tare weight is therefore minimized, and the weighing takes place closer to the subject [152]. Attachments mounted on the bed do not disturb the weight measurement.

For stretchers used in medical imaging applications, there are some alternative methods in use or proposed for patient weight acquisition, mainly utilized to improve the imaging control parameters. In some cases, the motor current or the hydraulic pressure in the lifting device is measured. This is proportional to the vertical force of the table itself and the patient on it. Due to friction and other disturbing effects, the weight determination is usually very coarse [21]. An alternative method is to accelerate the tabletop with the patient on it, using a predetermined force F over a short distance in a horizontal direction. From the resulting velocity progression Δv , the mass of the patient's body can be calculated using the equation of motion $F = m \cdot a$ [110].

2.2 Weight Estimation Methods

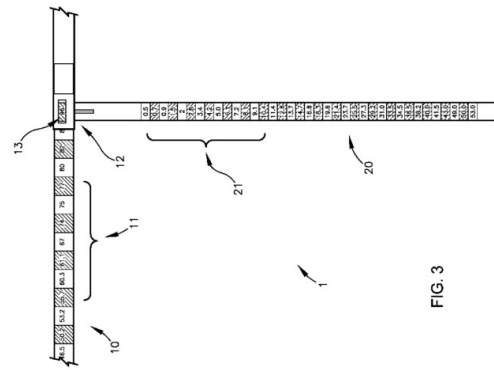
This section focuses on approximations and estimations of the weight. In case the measurement is hardly possible, an estimation can be sufficient for some scenarios.

2.2.1 Weight Estimation Devices

To approximate body weight in a medical scenario, hardware devices exist. These devices help to measure anthropometric features, e.g., the patient's body height, waist or hip circumference. Common forms are rulers or measuring tapes. The Broselow tape is such a measuring device and was developed in 1985 by James Broselow and Robert Luten to approximate the body weight of children for dosing and emergency treatment. It provides nine different weight groups for children younger than 12 years having a weight lower than 36 kg [33]. A colored scale on the measuring tape relates to various medical sets, prepared for the treatment of the different weight groups in case of an emergency, with the focus on saving time [32]. Further, the color assignment to different doses helps to minimize errors in dosing. Children are more sensitive to incorrect dosages because their organs are not fully developed. Additionally, an overdose can occur more easily for children and the concentration of the active substance in the medicine is the same for children and adults. To overdose an adult, often several syringes are necessary. In contrast, it might only take one syringe to overdose a child. Since the development of the Broselow tape, several studies have illustrated that it is reliable for use by first-aid personnel [9, 5]. However, some studies demonstrate that the Broselow tape is not reliable for all ethnic groups.



(a) Broselow tape, Source: [223]



(b) Estimation ruler for children, Source: [1]

FIG. 3

Figure 2.3: Weight approximation devices

Applying the tape to those groups can lead to an overestimation or underestimation, as shown by Ramarajan et al. [176] who applied the tape to Indian children. For other groups, however, it can provide an adequate body weight estimation [94]. Figure 2.3a shows the measuring tape for performing a weight estimation for a child.

Another example of an estimation device is the pediatric weight-estimation ruler [1]. With its help, the person's girth and length are measured. The device combines both lengths and shows an estimated body weight in a window. Figure 2.3b shows the ruler for weight approximation in children.

2.2.2 Weight Estimation Based on Anthropometric Features

Obviously, there exist correlations for the body weight based on features of the human body. Someone who is tall will probably be heavier than someone who is small; someone with thin wrists is also more likely to be lighter than someone with thick wrists. This section describes state-of-the-art methods which are based on measurements of the human body.

Estimating body weight m from body volume v demands knowledge about body density ρ . To measure human body density there are various commonly used methods, such as hydrodensitometry or air-displacement plethysmography [52], but they are time-consuming and therefore not very suitable for clinical practice.

Popa et al. [168] demonstrated the measurement of body density ρ using the bioelectric impedance rating. The body density is different from patient to patient, and also depends on gender and age, as shown in medical studies by Durnin and Womersley [58]. Based on 481 patients, the highest body density was measured as $1,082 \text{ kg/m}^3$, while the lowest was 968 kg/m^3 . Males have a slightly higher body density, and older persons have a slightly lower density. Furthermore, Wang et al. showed differences in the percentage of body fat and density for various ethnic groups [217].

Sendroy and Collison [186] deployed the correlation between body weight m , length l , surface area s and volume v in a study with over 700 patients in 1966:

$$m = \left(\frac{v - c}{s - a} \right)^b \cdot l,$$

where the parameters a , b and c represent empirical setting options for different ages, genders and ethnic groups [186].

Several equations exist to calculate body surface area [27, 71, 141]. For some medical scenarios, such as the treatment of cancer patients with chemotherapy or the treatment of fire victims, knowledge of the area of the body surface is essential for good treatment. Often, these equations relate the body height h and the body weight m . Therefore, the equations can be solved for the body weight m . The equation proposed by Bois [27] was determined empirically with only nine patients in 1916. The equation in terms of body mass is given by

$$m = 0.425 \sqrt{\frac{s}{0.20247 \cdot l^{0.725}}} ,$$

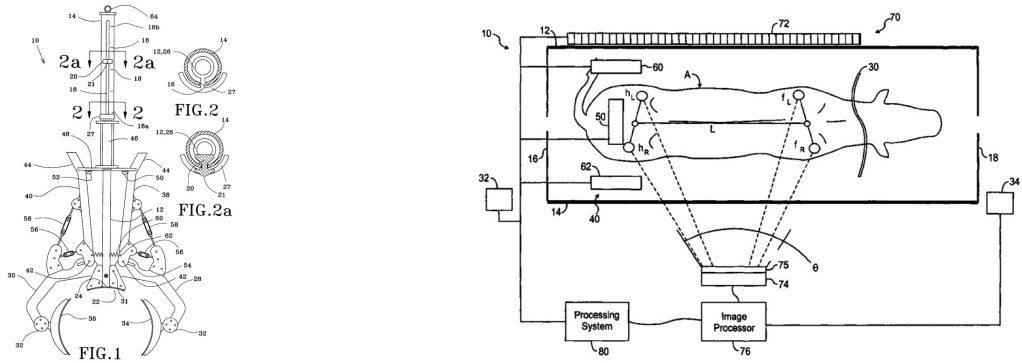
using empirically estimated coefficients as well as the body's surface area s and length l . The equations of Gehan and George [71], Haycock et al. [81] and Mosteller [141] are extensions to the formula developed by Bois [27]. The authors of these equations generated the equations empirically but used different data sets.

Today's physicians can use various types of anthropometric estimation methods to determine the body weight of a patient in emergency scenarios. Lorenz et al. [124] developed a formula for weight approximation with a focus on stroke patients which has up to almost 94 percent accuracy for a ± 10 percent range. They used body height l , waist circumference l_w and hip circumference l_h – all in centimeters. The coefficients for the equation differ with respect to the patient's gender. The weight is approximated by

$$m = \begin{cases} -137.432 + l \cdot 0.60035 + l_w \cdot 0.785 + l_h \cdot 0.392 & \text{for males} \\ -110.924 + l \cdot 0.4053 + l_w \cdot 0.325 + l_h \cdot 0.836 & \text{for females} \end{cases} . \quad (2.1)$$

Measuring the circumference of the human body may lead to unnecessary movements, which could cause a decline in cases of fractures or internal injuries. Additionally, the measurement takes time and is therefore hard to apply in an emergency situation. Although the equation is easy to calculate on a computer, it still takes some time to type every measured feature and the coefficients into a calculation program. This can cause an additional delay in treatment and there is a risk of typographical errors, e.g., a transposed digit, which could result in incorrect dosing.

Breuer et al. [31] tested in their publication the quality of visual estimation of body weight by physicians and nursing staff, as well as the previously described method proposed by Lorenz et al. [124]. In experiments, they only reached an accuracy of 80 percent for the estimated weight within ± 10 percent of ground-truth body weight. However, visual estimation from the physicians was even worse, with an accuracy of 65 percent. The body weight as provided by the patient was sufficient for treatment in 80 percent of all cases. However, only half of the subjects were able to provide their own body weight, due to symptoms of stroke, such as speech disorders or unconsciousness.



(a) Mechanical weight estimation device for live-stock, Source: Morissette [139]

(b) Visual weight estimation of a cow, Source: John [97]

Figure 2.4: Various methods to estimate the weight of livestock. Morissette [139] presents a mechanical caliper to approximate the weight of pigs. A contact-less method is proposed by John [97]. He presents an algorithm which extracts features from a cow to obtain an estimate of its weight. The features are extracted using a camera above the animal.

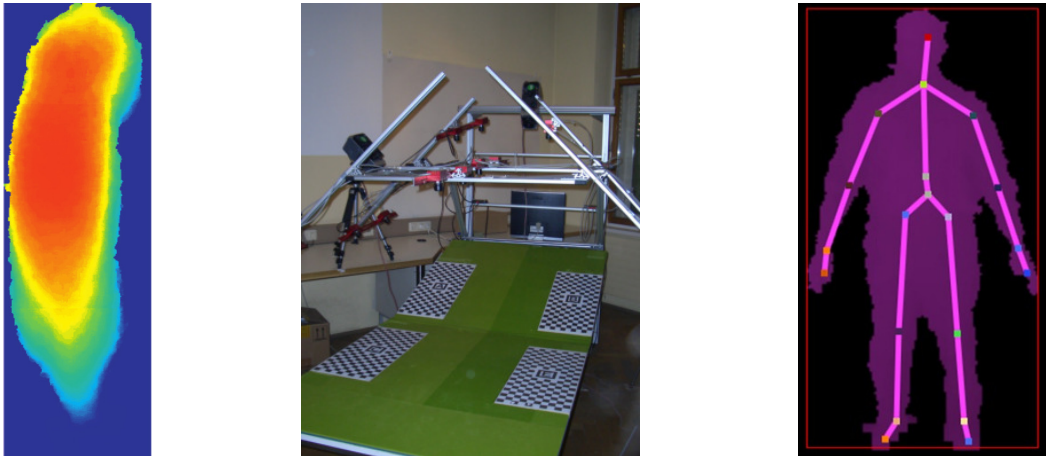
Gascho et al. [70] provide in their paper a method for estimating body weight during a CT examination. Based on a dose modulation, a correlation with body weight is determined via linear regression. The approach was tested with 329 decedents. This approach has the benefit that the body weight estimation is done during CT examination, a procedure which is in most cases necessary for the treatment of acute ischemic stroke.

2.2.3 Weight Estimation with Optical Sensors

Besides weight estimation of humans in a clinical environment, there are several publications concerning the estimation of livestock weights, based on mechanical devices – for example as proposed by Morissette [139] and shown in Figure 2.4a – or on visual systems. Weighing livestock is necessary to monitor health and readiness for the market but has several challenges. To weigh a single animal, it has to be separated from the rest of the herd, which can be stressful for the animal and for the stockman. The benefit of an optical system is that the animals can be visually segmented from each other while walking or standing in their natural habit.

Kongsro [108] presents an approach to estimating the weights of pigs for slaughter. Based on a depth image for a structured light sensor from above a single pig, the body weight is estimated using a statistical relationship with a previously measured reference database. In experiments with 142 pigs of three different breeds, they achieved an accuracy with an error of less than 5 percent. Figure 2.5a shows a depth image of a pig, recorded from above.

Another approach to obtaining the weight of livestock is presented by John [97]. Here, the estimation is focused on cows. Four markers on the backs of the animals are localized based on a monocular camera. Additional ultrasonic sensors measure the thickness of the cow's back, as well as its height. Figure 2.4b illustrates the approach using the extracted four markers, the visual sensor and the ultrasonic sensor. The patent does not provide any results from experiments.



(a) Setup for visual body weight estimation of a pig by Kongsro [108].

(b) Setup for visual body weight estimation with 16 stereo cameras by Pirker et al. [165].

(c) Body weight estimation by Cook et al. [44] based on a structured light sensor.

Figure 2.5: Various approaches to obtaining anthropometric data via a 3D sensor.

In addition, various approaches exist for performing an estimation of human body weight based on visual and contactless methods. The applications of these methods are often related to the treatment of emergency patients with medication dosing.

Kocabey et al. [106] present an approach for estimating the BMI of a person, based on a color image showing the face. The authors collected 4,206 faces with corresponding genders and BMI information, from social media pictures. The method uses the deep learning approach (VGGNet), presented by Simonyan and Zisserman [192], to train a model. The system performance is close to the estimations made by humans. It can estimate 56.2 percent of a given data set with an error of less than 5.5 percent with respect to the ground-truth BMI. The pre-trained model is only available for academic research, due to its potential abuse in social media.

The estimation of the BMI is also presented in the work by Nahavandi et al. [143]. Based on 100,000 computer-generated 3D mannequins, the body surface area is used as a feature for the estimation. The approach uses a deep residual convolutional neural network model [82]. The proposed system estimates the BMI with an accuracy of 95 percent.

Pirker et al. [165] employed 16 stereo cameras around a stretcher to estimate the volume of a subject. Additional projectors are needed for complete illumination. It is complemented with a parametric human model of the back of the body. Composed images are filtered for noise reduction, and finally, the volume is calculated with the help of cross sections along the body. Because of the large number of cameras around the patient's bed, physicians would be constricted while giving treatment. Figure 2.5b shows a stretcher with markers for calibration and the frame for the sensors.

Another approach is presented by Robinson and Parkinson [178]: Here, anthropometric features are extracted from a scene's point cloud. The raw sensor data are from an Red Green Blue Depth (RGB-D) sensor with a person standing in front of it. This approach also demonstrated that the features from the point cloud could lead to a bias due to an uncalibrated

sensor or to noise. Further, even thin clothes can confuse the extraction of features such as the circumference of a body part, e.g., the waist or the hip.

Cook et al. [44] presented a framework based on a structured light sensor for radiation dose estimation in CT examinations [44]. In preliminary experiments, they showed results for five persons standing in front of a structured light sensor. The measured volumes of the patients differ for various positions of their arms, and therefore a fixed posture is required for the measurement. Figure 2.5c shows the depth image, including the extracted skeleton markers, necessary for the body weight estimation.

With the help of skeleton tracking, Velardo and Dugelay [212] demonstrated a computer vision system to prove the health of a person with the support of a structured light sensor. The system estimates anthropometric features such as the circumferences of arms, legs and the body. In addition, an operator adds the age of the patient to improve the outcome of the body weight estimation. The authors provide a trained statistical model from a medical database containing anthropometric measurements from more than 28,000 subjects, as well as the ground-truth body weights. This approach has the benefit of a large sample size for training. However, the estimation of the anthropometric features based on the RGB-D data is difficult, due to sensor noise. The authors further presented their approach as suitable for body mass estimation in space with no available gravity [213].

Based on a single image from an RGB-D sensor, Nguyen et al. [145] demonstrated a method to estimate a person's body weight using a side-view feature, as shown in Figure 2.6. A support vector regression model is applied to the extracted feature vector. The authors divide their experiments and data sets by gender. For females, their approach achieved an average error of 4.62 kg, while the error for males was higher at 5.59 kg. Additionally, they compared the visual guess by surrounding people with their approach, demonstrating that the human visual guess gives a worse estimation. Further, the authors made the RGB-D data set from the experiments public, including the ground-truth body weights. The experiments in this thesis also use this data set. Figure 2.6 shows the sensor together with the subject, as well as the extracted side-view feature.

Some of the content presented here has also been published in advance of this thesis. In the first conference paper, Pfitzner et al. [163] demonstrated a body weight estimation based on a volume reconstruction. The volume estimation was eased because the subjects were lying on a medical stretcher, modeled as a flat surface. With the help of a fixed value of density, the body weight could be estimated. The focus was on clinical usage, especially the treatment of stroke patients with a body weight-adapted dose for tPA. Results from experiments with around 100 patients showed that this approach is more suitable for medical dosing, compared to a physician's visual guess. In an extension, Pfitzner et al. [164] presented an approach relying on more than just the volume. A set of 10 features is extracted from the person's point cloud. These features are forwarded to an artificial neural network, which was trained with a previously recorded data set. Results from the experiment reached an accuracy of nearly 90 percent for a relative error of ± 10 percent of the ground-truth body weight.

With another optimization, Pfitzner et al. [162] reached a good accuracy by extracting 23 features, extending the geometric features from the previous approach [164]. Features from contours and from a thermal camera, as well as the ground-truth gender, improved the outcome of the body weight estimation. These features are further analyzed for their correlation to

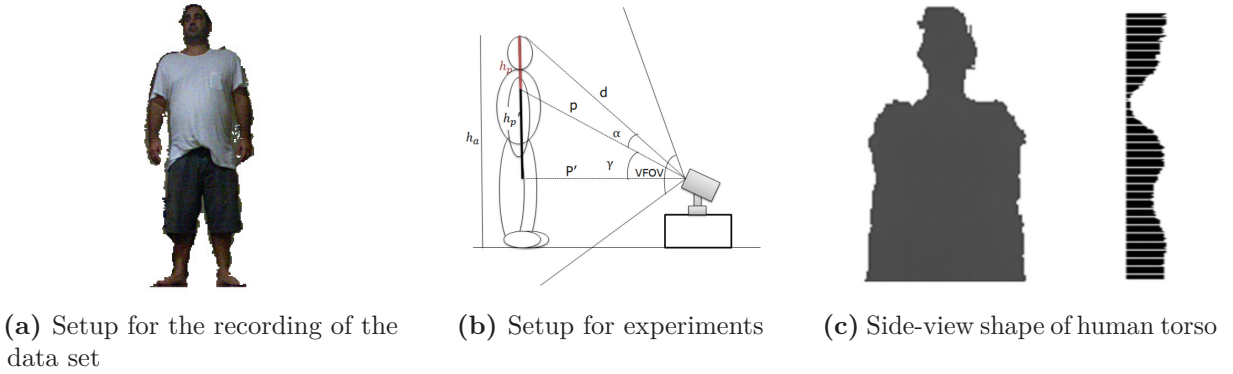


Figure 2.6: Related work from Nguyen et al. [145]: The data set used for the experiments consists of 299 subjects (a). The data set is published and is also used in this thesis for experiments. All subjects are standing in front of a Kinect camera, which is placed on the floor, pointing upwards to the subject (b). The proposed side-view feature provides a 200-dimensional vector on the basis of an extracted depth image (c). Source: Nguyen et al. [145]

body weight. Different feature groups are evaluated for body weight estimation via experiments. Finally, the approach achieved an accuracy of 94.8 percent when applying data from the Microsoft Kinect camera and 95.3 percent when applying data from the Kinect One camera. The data from the feature extraction are published for the data set used in the hospital, to encourage joint work in the field of body weight estimation for medical purposes. All three papers had a clear focus on medical dosing of stroke patients and together they illustrate the progress of this study over time.

2.2.4 Weight Estimation in Video Streams

Similarly to body weight estimation at a single moment, it is also possible to estimate body weight using a sensor data sequence. Labati et al. [112] developed a body weight estimation technique suitable for walking persons. The focus was on a contactless and low-cost method. It is based on frame sequences from two cameras, which are mounted perpendicularly to obtain a frontal and a side view of the walking person. The feature vector consists of the height of the person, measured in pixels, an approximation of the body volume, an approximation of the body shape and the walking direction. The extracted features are forwarded to an Artificial Neural Network (ANN) to obtain the body weight. Experiments were performed with 20 subjects, walking in eight different directions. A maximum absolute mean error was recorded of less than 2.4 kg.

Arigbabu et al. [10] demonstrated the extraction of soft biometrics, e.g., body height and weight, based on video frames from a single monocular camera. With a homogeneous background, a person's silhouette can easily be extracted with state-of-the-art image processing techniques such as background subtraction. The silhouette is converted into a binary mask, where 13 features are extracted, depending on the pixel density in segmented regions. Figure 2.7 illustrates the feature extraction from the person's silhouette.

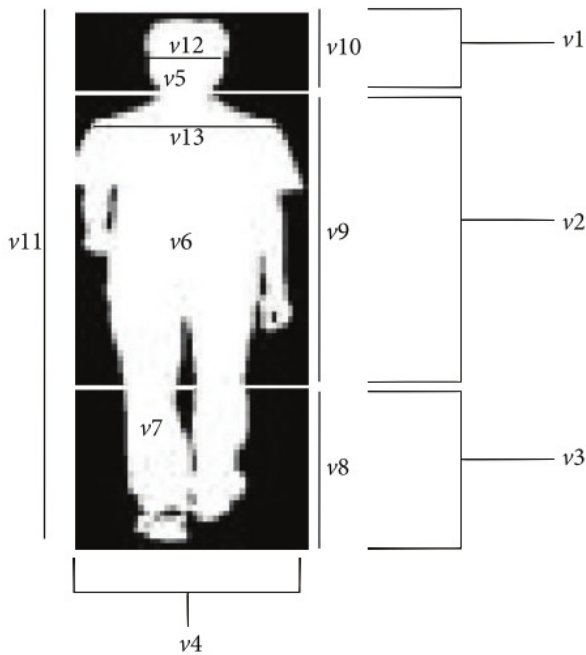


Figure 2.7: Extraction of features presented by Arigbabu et al. [10]: Based on a binary mask of the segmented subject in the FOV of the camera, a feature vector is calculated. Features 1 to 3 in the graphic describe pixel density divided by the length of the head, the torso and the lower legs. Other features, such as numbers 12 and 13, measure the width of the head or the shoulders.

The feature vector is finally forwarded to an ANN to estimate the body weight. In experiments with 80 subjects and a total of 1,120 video frames, a mean absolute error of 4.66 kg was achieved, with a standard deviation of 3.48 kg, for body weight estimation. The update rate of the extraction of all the described soft biometrics was about 1 Hz. The approach is compared to the previously presented method proposed by Labati et al. [112] and Velardo and Dugelay [212].

Pfitzner et al. [161] presented an extension of the body weight estimation in medical scenarios [163, 164, 162]. Based on almost the same feature set, estimations for standing and walking subjects were achieved. The camera was placed in a hallway with the subjects walking towards the camera. The results are compared with the approach presented by Nguyen et al. [145] and are also presented in detail in the section on experiments.

2.3 Summary

This section illustrated that today, several weighing devices and approximation approaches exist. State-of-the-art scales are superior to all estimation methods but sometimes cannot be used, or can only be used with considerable effort. Therefore, estimation methods are sometimes necessary to obtain a weight value. Particularly in medical applications, estimation is often sufficient for dosing of drugs in cases of emergency. The approach for body weight estimation discussed here will be hardly as reliable and accurate as a primitive spring scale. However, such a weight approximation based on data from a camera system, has the benefit that it is contact-less and – with the focus on stroke patients – the patient does not have to get on a set of standing scales or need to be moved to be placed on a chair or on bed scales. It is the aim of the project

Libra3D and of this study to achieve a method which outperforms the estimation accuracy of physicians, with a focus on stroke patients. Table 2.1 compares the presented related work for visual body weight estimation as a summary, provided the data from experiments are available.

Table 2.1: Results for contact-less human body weight estimation from related work in alphabetical order. The results are not directly comparable due to different evaluation metrics, e.g., Mean Absolute Error (MAE) or an error bound. The first section of this table presents related work for a single frame, while the second part illustrates estimation based on a video stream.

Method	Sensor	Constraints	Results
Cook et al. [44]	RGB-D structured light	sample size six subjects	only volume estimation
Kocabey et al. [106]	RGB		BMI estimation
Nahavandi et al. [143]	RGB	simulation	BMI estimation
Pirker et al. [165]	8 stereo cameras	scene has to be known	only volume estimation
Pfitzner et al. [163]	RGB-D structured light	person is lying on a flat surface	79.1 % within relative error of 10 %
Pfitzner et al. [164]	RGB-D structured light	person is lying on a flat surface	89.6 % within relative error of 10 %
Pfitzner et al. [162]	RGB-D structured light or ToF	person is lying on a flat surface	95.3 % within relative error of 10 %
Nguyen et al. [145]	RGB-D structured light		5.2 kg MAE
Velardo and Dugelay [212]	RGB-D structured light	sample size 15 subjects	2.7 kg for a single subject
Arigbabu et al. [10]	RGB		4.66 kg MAE
Labati et al. [112]	2 RGB	sample size 20 subjects	2.3 kg std error
Pfitzner et al. [161]	RGB-D		3.30 kg MAE for walking subjects and 4.31 kg MAE for standing subjects

Chapter 3

Conceptual Design and Sensors

This chapter includes a conceptual design for clinical integration, as well as discussion about applied sensors for data acquisition. While the project *Libra3D* started with the funding in October 2013, the first prototype was deployed in the neurology of the University Hospital Erlangen, Germany, in February 2014. The system was optimized over time: Sensors were added and compared, new data was acquired, and therefore different data sets were recorded. This section presents the final design of the system and the characteristics of the applied sensors, as well as the sensor fusion.

3.1 Environment

An Emergency Department (ED) often has several trauma rooms for the treatment of patients arriving at the hospital. The purpose of such a trauma room is the fast diagnosis and immediate treatment, especially for patients being in a critical and life-threatening situation. The room is used for all patients, independent of the patient's issue. The trauma rooms in the hospital in Erlangen have a size of 4×6 meters, containing a medical stretcher for the patient, a table, a drug cabinet and medical devices for the treatment close to the patient. The stretcher is placed in the middle of the room, close to one wall where the medical devices, e.g., ventilator and electrocardiogram are stored. On three sides, the stretcher is free to access, so several physicians can treat a patient at the same time if needed. Figure 3.1a shows the trauma room with a patient on the stretcher. The stretcher is equipped with raisable handlebars for patients with seizures to prevent them from falling. Additionally, the backrest of the stretcher can be adjusted. For cardiovascular disease, it is common that the patient is sitting in a more upward position to ease breathing.

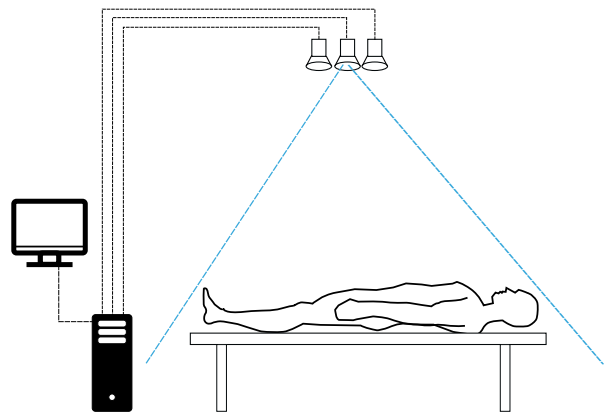
With the arrival at the emergency room, the patient's information is recorded as far as possible and if not done before. Every treated patient is identified with a unique barcode printed on a wristband. This code is placed on every document concerning the patient's anamnesis or therapy. Therefore the physicians have to enter the number under the barcode or scan it, to provide explicit identification. This guarantees ideal conditions for the treatment and provides a unique identification for the habitation in the hospital. A label displaying the name of the patient, the barcode and the patient's Identifier (ID) is placed on all drugs the patient receives.



(a) Trauma room with sensors integrated into the ceiling



(b) Yellow markers on the floor to ease the positioning of the stretcher in the FOV



(c) Schematic of the sensor system and its connections to a computer

Figure 3.1: Clinical integration of sensors into a trauma room: Within the scenario of the trauma room in which physicians mostly treat emergency patients the sensor system is integrated into the ceiling (a). Besides the sensors in the ceiling, the system consists of a computer system – including a keyboard and a mouse for interaction and a monitor for visualization and a barcode scanner to identify patients with their ID (b). USB cables achieve the connection between the sensors and the computer.

This ID ensures that every patient receives the medication, which was meant for him or her. Therefore, a hospital can minimize issues with patient confusion, thus reducing complications during the treatment.

To integrate the visual body weight system into the trauma room, the environment should be changed as little as possible, keeping the procedure for the treatment the same as without the system. However, some changes are necessary: First, the sensors are mounted in the ceiling. The cables for the data from the sensors are installed in the cable conduit. Some of the sensors need an additional power supply, a power socket is set up behind the suspended ceiling. Finally, the computer for processing and a monitor for visualization are placed on the desk in the trauma

room. Figure 3.1c illustrates the setup in the trauma room with the sensors in the ceiling, the computer and the patient on the stretcher.

In any case, the stretcher has to be placed inside a marked rectangle on the floor. This rectangle is added to the room for the here presented application with yellow adhesive tape to provide a visual feedback if the stretcher is placed correctly.

It is important that a visual body weight estimation approach is not distracted by objects in the environment. Therefore, objects close to the patient who might affect the estimation are segmented. Such objects are, for example, a heart monitor or a saline bag, see Figure 3.1.

3.2 Diagnostics and Treatment

Patients are prioritized when coming into the emergency room via a triage system, e.g., the emergency severity index [225]. The index starts at one, for patients being in a life-threatening situation, going up to a value of five, for patients who can wait some time. Typically, stroke patients receive a severity index of one or two, depending on the symptoms of a stroke. A patient not being able to breathe on his or her own and being unconsciousness will be given an index of one and is therefore treated with the highest priority.

Often, physicians work as a team of several people, especially in a life-threatening situation like a stroke. First, a doctor asks the patient for his or her symptoms. With this impression, the responsible physician will investigate the different symptoms, e.g., touching the patient for pain, or ask about medical history, to conclude to a possible diagnosis. In most cases, blood from the patient is analyzed. If necessary, the patient is brought to a medical imaging device, either X-ray, CT or MRI scanners. In case the physician suspects a stroke, the medical imaging is mandatory to differ between an ischemic or a hemorrhagic stroke. The treatment of these types of stroke are entirely different, and applying the wrong treatment would lead to a worsening of the patient's condition, as explained in the introduction of this thesis. In case of an acute ischemic stroke, the patient can be treated with tPA, to solve the blood clot in one of the brain vessels. The patient's body weight is now needed because the lyse is adapted to the blood volume, which correlates to the body weight.

3.3 Handling of the Weight Estimation with Libra3D

The sensor system and the computer are always ready for data acquisition. Only for service, the computer will be shut down, but then the nursing staff and the physicians are informed. The program starts up automatically after the computer is powered. In case the program is not running, it can be started with a button on the desktop. The started program initially shows the live stream as a point cloud on the left side, as shown in Figure 3.2 on page 28. On the right side, the attending physician can add data of the patient or the measurement, either before the measurement or afterward. Although it is quite common that the medical stretcher is placed at the same position in the trauma room, additional markers are placed on the floor which indicates the FOV, to guarantee that the stretcher is placed correctly without looking at a live stream. This ensures that no time is lost to position the stretcher with the patient in the FOV.

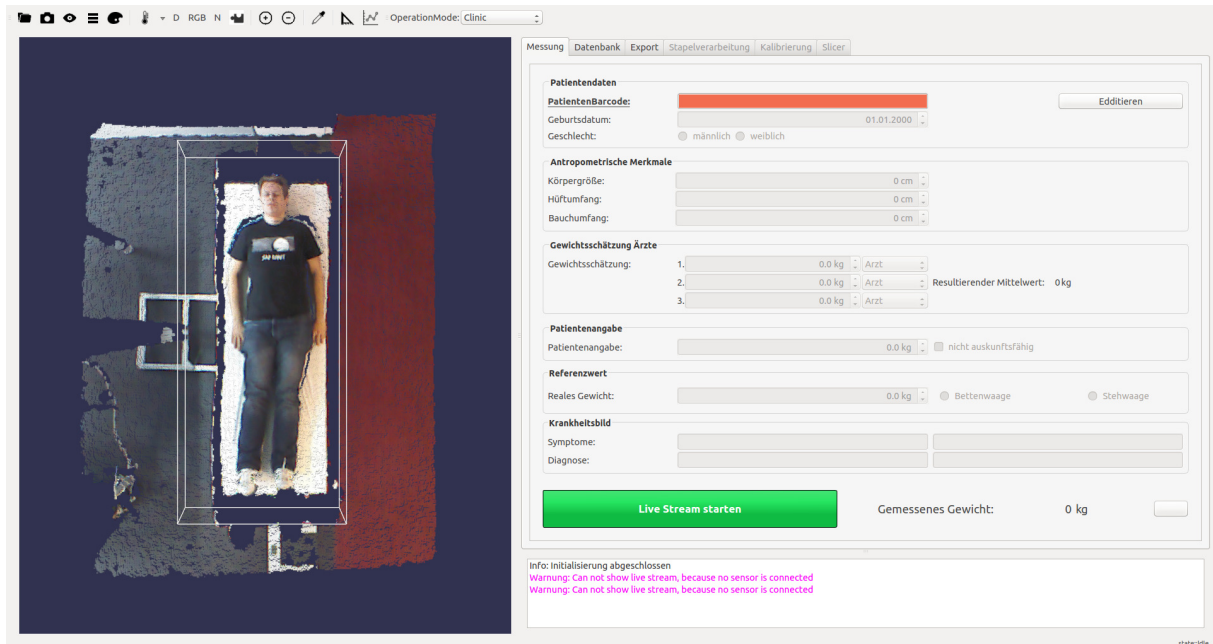


Figure 3.2: GUI with data handling for a new patient: For body weight estimation in the trauma room the medical staff has a live stream of the sensors available. By pressing a button, a single frame of each camera is taken to process the body weight estimation. Afterward, the physicians can add additional information – e.g., ground truth body weight, height, gender – to the data set to improve the outcome of sub-sequentially estimations.

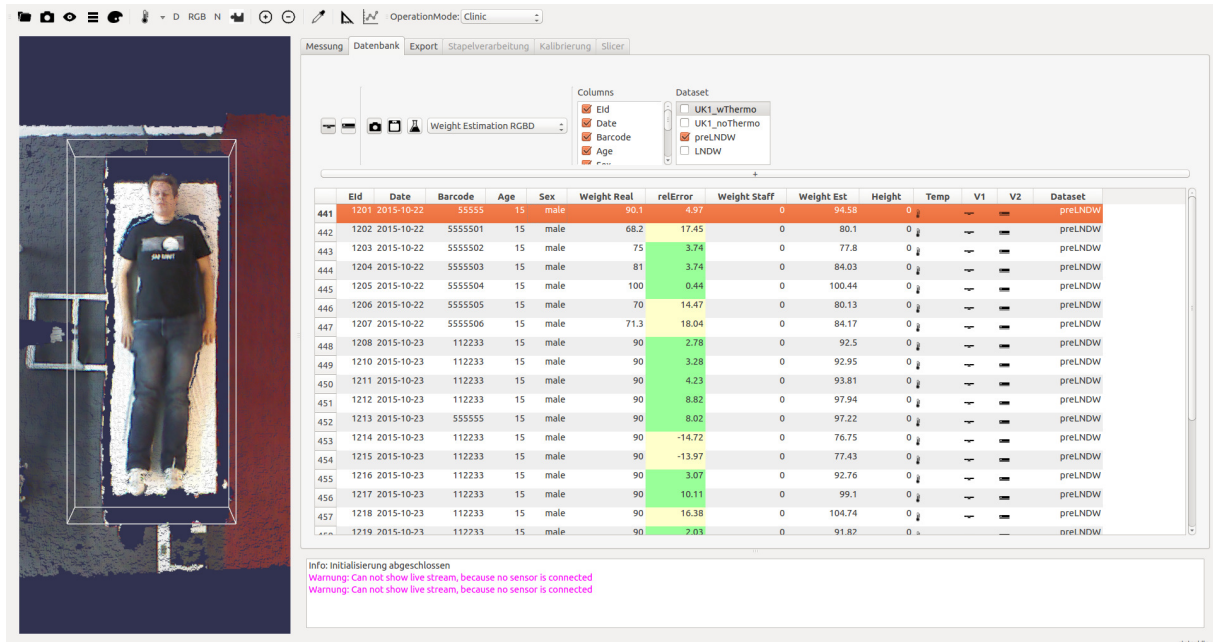


Figure 3.3: GUI for offline processing with the database: Additionally, the medical staff can display previously taken data sets from the table. Filters can be applied to search for a specific data set.

Patients have to lie flat on the stretcher, as illustrated in Figure 3.1c. A doctor can start the measurement by pressing the button *Start Measurement* (see Figure 3.2). The moment the button is pressed, the patient should be clearly visible in the FOV. Physicians close to the patient should take a step back for the data acquisition. However, it is no problem if they are within the FOV or within the area spanned by the markers on the floor – as shown in a subsequent section about segmentation. After this trigger, the system takes the next frames, processes them, and returns a value for the body weight estimation on the display. Within less than a second, the treating physician gets the estimated weight on a monitor. Being in a stage of development, all physicians decide if they trust the recommendation from the weight estimation, or their own perception. Figure 3.4 illustrates the process in weight estimation with the *Libra3D* system. First, the attending physician has to decide on an initial medical examination, whether the patient is in a critical state. If yes, then the physician has to check, whether the treatment relies on the body weight. If so, then the algorithm will try to estimate the body weight. If the result of the estimation looks reliable to the physician, the treatment can start with this value. If not, then the physicians start the treatment with an own visual suggestion. When the patient is not in a critical state, the physician, the patient and the algorithm give a value successively.

On a monitor attached to the computer, the medical staff sees a live image of the sensor's data, as shown in Figure 3.2. Free of choice of all available sensors, different views, data can be applied, like from the color, depth or thermal data. It is most convenient for physicians to observe the color image to check if the patient on the stretcher is completely within the FOV.

While developing *Libra3D*, the Graphical User Interface (GUI) is adapted to the requirements of the physicians. The GUI has several views: Figure 3.2 shows the starting surface, which is responsible for data acquisition for the body weight estimation. On the left side, the physician sees the live stream from one of the sensors. Different representation can be selected, either Red Green Blue (RGB), thermal view or a fused image with an overlay of thermal and color information.

The live stream ensures that the patient is positioned correctly in the FOV. Although the range for the patient is displayed with the markers on the floor, the physicians often prefer the additional indicator via the live view on the monitor, as shown in Figure 3.2.

In a second view, medical staff can retrieve existing measurements from the database. The physicians can have a look at the recorded data, or modify the patient's data in case of incorrect insertions. With filters, a physician can look for a specific case, based on several data, e.g., date, real body weight or gender. Figure 3.3 shows this view. On the left view, the point cloud of the marked data set is displayed. Users can rotate the displayed point cloud or can zoom in. The GUI further provides a virtual measuring tape, so distances between selected points from the cloud can be extracted. The mouse cursor can set measuring spots, so the Cartesian coordinates are displayed or the temperature of the marked point.

The system can be connected to the local network or the Internet: This has the benefit that recorded data can be saved automatically to a remote server to prevent data loss. Furthermore, a remote connection on the computer in the hospital, e.g., via Secure Shell (SSH), can be established to maintain the system. With every start of the program, a logging file is generated. Every interaction of the user is stored, as well as warnings and errors, including a time stamp. Furthermore, the results of all performed measurements are stored in the log file, so misbehavior can be reconstructed. The physicians do not recognize the logging of the data. It is rather a

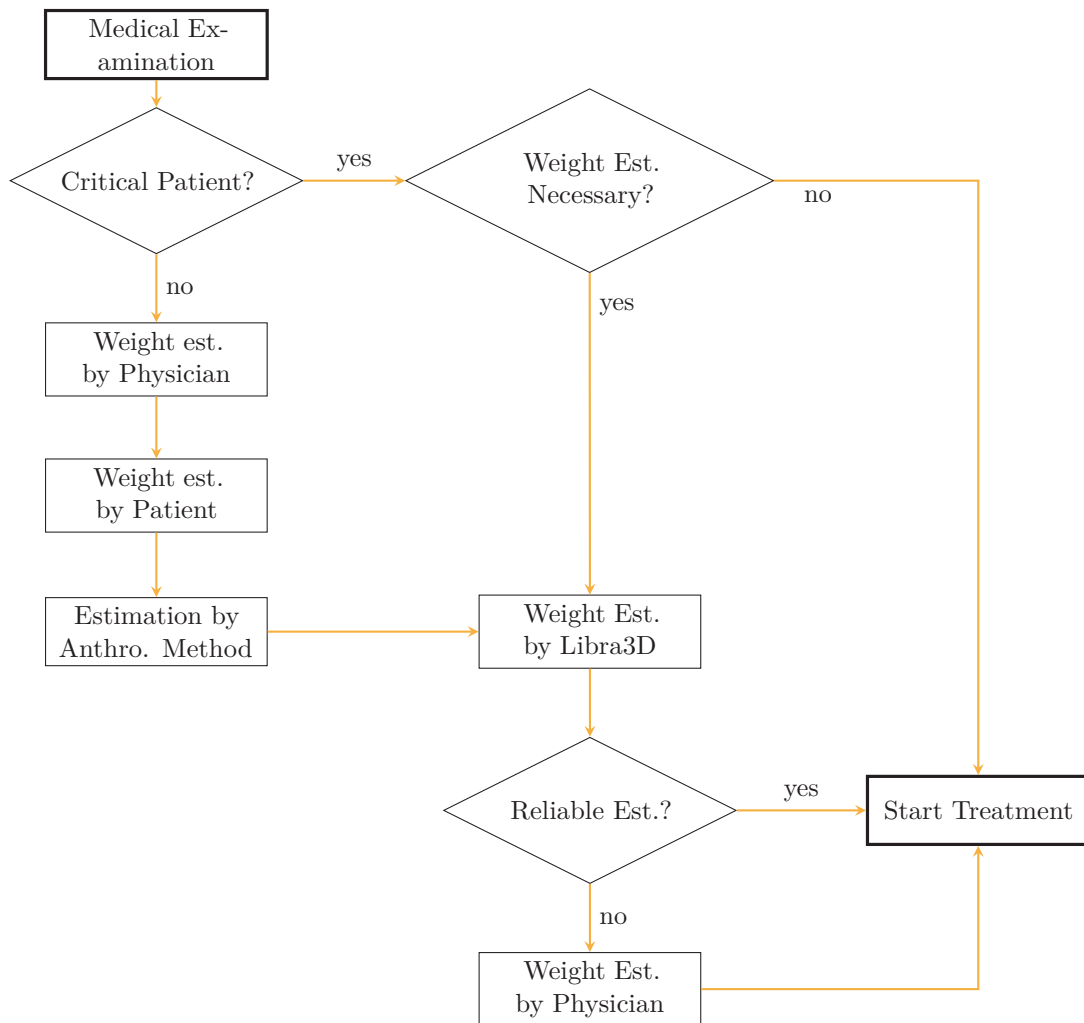


Figure 3.4: Process in weight acquisition for patients in the trauma room: For the study in body weight estimation, the order in applying different methods is crucial. First, the weight of the patient is estimated by each attending physician independently. Next, the patient should provide his or her body weight if possible. Finally, anthropometric features are taken from the patient for a state-of-the-art estimation method. The graphic does not show the ground truth measurement, which is done after the treatment of the patient at the infirmary.

method to locate errors in the algorithm or the user interface. The log file does not contain the data provided by the sensor, which is stored in an additional database.

The GUI of *Libra3D* offers three different operating modes: The first one is called *Clinic*, providing all necessary functions for the usage in the hospital with the focus on stroke patients. A second mode, called *Expert* mode, provides more possibilities for configurations. This mode is used during development in the laboratory and provides the configuration to change the calibration of the sensor setup or the trained model for weight estimation. The difference between those two modes ensures that physicians are not confused by too many settings. Both modes

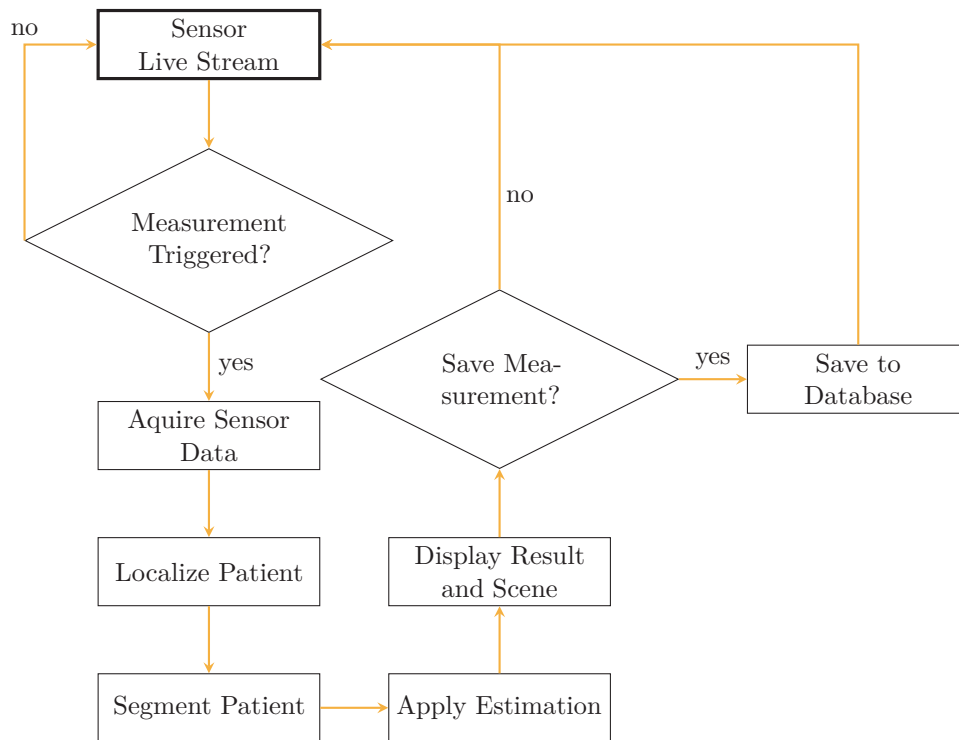


Figure 3.5: Process of data acquisition in the program Libra3D: When the system is started, the GUI shows a live stream of the sensors. When the measurement is started, the data from the sensors is forwarded for processing. Finally, the user can select if the measurement should be saved or discarded. After a measurement, the system returns to the live stream.

give access to the database, to check past estimations, or to export the data for post-processing in a statistical software.

A third mode, called *Event* mode, is used for public presentations and demonstrations. The initial input screen is reduced, so no hospital common parameters like the ID or the estimation of the physicians can be entered. A user can also decide, whether the data from the sensors should be saved to the database, in case a subject does not want to provide his or her data for research. After the visual body weight estimation, the ground truth body weight can be added afterward. To ensure maximum privacy for former subjects, the database can not be accessed in this mode, in case the computer with the system installed is unattended. A password is required to switch between the modes.

Figure 3.5 illustrates the process for a single measurement as a flow chart: While the system waits for the user to start a measurement, the current stream of the sensors is displayed in the GUI. After the estimation, the system returns to display the live data.

3.4 Data Management

With every measurement, the data set is saved to the computer. Afterward, the recorded data can be used to improve the outcome of future estimations. Therefore, a My Structured Query Language (MySQL) database is integrated into the computer for data management [184]. This type of database can be accessed via various frameworks. Furthermore, it can be accessed remotely and provides additional features to backup the recorded data in case of a corrupted hard disk. The database is structured in tables, representing objects, which can be accessed by the algorithm. The database is integrated via the QDjango framework, a Qt-based extension for the MySQL wrapper framework Django [24, 57]. Figure 3.6 shows the structure of the database with its tables. Please note that the diagram does only contain elements necessary for the process of weight estimation, while minor elements are not in this visualization, like the primary keys.

- **Patient:** For every patient applied to the system, a unique *Patient* item is defined in the database. The patient item can be found based on a unique ID and the barcode the patient receives with admission in the hospital. With the additional entry *comment*, patients can be marked, for example for exclusion from data set due to various reasons.
- **MasterData:** The item *MasterData* stores a patient's basic personal data, like the name, birth date, and gender. For every patient, a unique master data is generated. This entry is designed to not being modified afterward. The ground truth body weight is not integrated into the *MasterData* element, as it can change over time. For every *Patient* element, exactly one *MasterData* element exists.
- **Measure:** The *Measure* item is generated with every recorded measurement. The measurement can be excluded from a data set with an entry *isValid*. In case the measurement might not contain a valid patient, or the measurement is started by mistake, the recorded measurement can be excluded from offline processing and optimization. A single patient can have multiple measurements. Furthermore, a time stamp is added to the table to identify a measurement or sort them by time. One patient can have several measurements but must have at least one, because the *Patient* and the *Measure* element are created at the same time.
- **MeasureBlob:** In the *MeasureBlob* item, data from the sensors is stored. For each sensor, a corresponding entry exists. If possible, the saved data is loss-free png compressed to reduce the size of the database. Only the thermal data is saved in raw. Additionally, the *MeasureBlob* item also includes the saved calibration data for extrinsic and intrinsic calibration to perform sensor fusion afterward.
- **HealthRecord:** The *HealthRecord* table is designed to store data from the patient, which is obtained by the optimization of the body weight estimation. Therefore, anthropometric measurements like the circumference of the hip or the waist are stored, as well as the patient's and the physician's estimation. For every patient, a single *HealthRecord* table exists.
- **Estimation:** An *Estimation* table stores the result of the estimation based on an applied algorithm. Additionally, the table stores entries of the date and the time of the estimation,

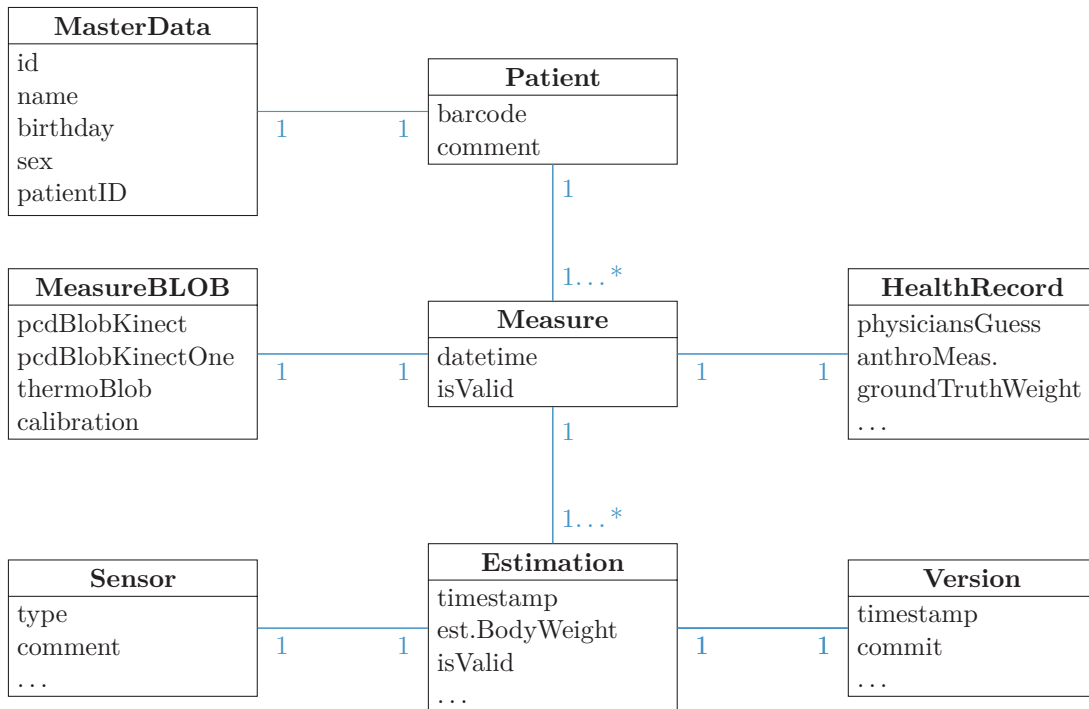


Figure 3.6: Structure of the MySQL database applied for Libra3D: The objects *MasterData* and *Patient* are generated if the system does not know the patient, e.g., by the identification based on the barcode. When a measurement is triggered, a new *Measure* object is generated, as well as a *MeasureBlob* and *HealthRecord* object. A single measurement can be estimated with different approaches, so multiple *Estimation* objects can exist for a single measurement.

as well if the estimation is taken in the trauma room scenario, in contrast to an estimation which was taken in the laboratory. A single measurement can have multiple estimations when different methods or algorithms are applied for testing and optimization.

- **Version:** The *Version* table stores different versions of the applied algorithm since the system was continuously developed. Therefore, an estimation can be traced back to a specific applied version of the algorithm.
- **Sensor:** The *Sensor* item marks the sensor data, which is used for body weight estimation. The algorithms can be applied to different 3D cameras, with or without thermal data.

An interaction with the hospital patient database is possible but was not achieved due to concerns about data security issues. Having such an interface to the hospital patient database could have the effort of a fully integrated patient treatment: Body weight indicators like height, age, sex or previously measured body weight could be incorporated automatically into a body weight estimation.

3.5 Applied Sensors for contact-less Estimation

The key to perception is based on the chosen sensor, its characteristics, and the quality of data. There exist several visual sensors to achieve contact-less measurements for human body weight estimation. The final system consists of multiple cameras.

The sensors used for this thesis and experiments are explained for convenience in the following section. Each applied sensor should fulfill two requirements: First, the patient should be entirely in the FOV of a sensor. If a patient is completely visible depends on the size of the FOV, as well as the sensor's distance towards the patient. Second, it is expected that a sensor system should provide depth information to estimate the weight of a subject in the FOV. In contrast to the estimation with 2D sensors, the estimation in 3D provides more information about the subject. Therefore the maximum range of a depth sensor should exceed the distance to the patient. Depending on the measurement principle, the range is limited to a maximum, but also to a minimum distance.

Besides a single sensor integrated into the ceiling, a set of sensors is possible. Two or more sensors in different poses attached to the ceiling have the benefit of having a better side elevation of the patient due to the larger angle of view. In contrast to that, a multi-view system has to deal with more occlusion of the patient due to people or objects between the patient and the sensors. Calibration of all sensors is a higher effort compared to a single view system. Therefore, a system from a single point of view is applied for the estimation of body weight.

Figure 3.7 shows all sensors used for this thesis and in experiments. Additional, Table 3.1 presents the specifications of these sensors for convenience. First, the Microsoft Kinect camera was installed in one of the trauma rooms of the University Hospital Erlangen in February 2014, see Figure 3.7a. This camera was applied because it became one of the most common 3D sensors in the robotics and perception community, since its release in 2011. The sensor provides color as well as depth information for every pixel and is therefore called RGB-D sensor. Later, in autumn 2014, a thermal camera was added to the system in the trauma room, see Figure 3.7c. With this sensor, the segmentation should be improved. Finally, in 2015 the system was completed with a Microsoft Kinect One, as shown in Figure 3.7b, the successor of the Microsoft Kinect camera – also an RGB-D sensor but based on a different principle to obtain depth measurements.

For a contact-less visual perception of a subject, other sensors would be possible: The most common and obvious way for the perception is a monocular camera. Here, a great variety of



Figure 3.7: Sensors tested for body weight estimation within this thesis.

Table 3.1: Property table of the used sensors: The three sensors are selected for the body weight estimation because of their similar FOV, which provides a total view of the patient on the stretcher. For the 3D sensors, the measurement range is sufficient. The frame rate of at least 30 Hz is acceptable, while the thermal camera provides a frame rate of 80 Hz. The 3D sensors are cheap compared to the thermal camera, having a price of 3,000 EUR.

Model	Kinect [218]	Kinect One [218]	Optris PI400 [91]
Principle	structured light	ToF	IR thermal camera
Resolution	640 × 480 pixels	512 × 424 pixels	382 × 288 pixels
Range	0.8 - 4.0 m	0.4 - 8 m	–
FOV	57° × 43°	70° × 60°	62° × 49°
Frame rate	30 FPS	30 FPS	80 FPS
Dimensions	73 × 283 × 73 mm	249 × 66 × 67 mm	46 × 56 × 90 mm
Weight	564 g	1,400 g	320 g
Power consumption	12 W	32 W	<2.5 W via USB
Interface	USB 2.0	USB 3.0	USB 2.0
Price	100 Euro	200 Euro	3,000 Euro

state-of-the-art filter and segmentation algorithms exist. However, a monocular camera alone can not obtain a depth image from a scene. Two monocular cameras mounted together with a known transformation between the sensor form a stereo camera: Based on triangulation, depth can be perceived [191]. Nonetheless, also a stereo camera can only get the depth of a scene if the surface provides enough structure for the triangulation. Another sensor which could be used in the scenario for the weight estimation is a Terahertz scanner: These scanners are sensitive to wavelengths shorter than 1 mm and provide the opportunity to look through clothes – therefore they are common at airports to check passengers for unauthorized items. However, the range and the distance for measurements is limited [154]. In contrast to common RGB-D cameras, terahertz scanners have a bigger housing, which increases the expenditure for integration in a trauma room. Also, the sensor is more expensive, compared to the RGB-D sensors like the Kinect.

The upcoming section illustrates the different principles of the applied sensors, their characteristics in sensor data, and their issues.

3.5.1 Monocular Camera

Monocular cameras are the most common types of optical sensors. Until the beginning of the 21st century, only a few digital cameras were available, and cameras used a photosensitive photographic film. Today’s cameras replace the photographic film with a digital sensor. Monocular cameras can be subdivided by the type of wavelength they can recognize. Having a monochrome sensor, every pixel delivers an intensity value. On the other side, having a color camera, each pixel delivers three values, red, blue and green. This pattern is known as the Bayer pattern [45]. The intensity values of such a sensor are fused to create a color RGB image. To provide a realistic and natural image, processors inside of such a monocular camera apply a color correction based on non-linear curves, while processing. Exceptions are high-quality industrial cameras or digital

single-lens reflex cameras, which can also provide a raw sensor data for post-processing on a computer.

3.5.2 Time-of-Flight Camera

The first commercially introduced Time of Flight (ToF) camera was developed by CSEM's Swiss Ranger in 2006, providing an increased robustness of depth measurements [133]. A ToF camera is based on time measurement of an emitted light source and reflected by an object. The distance of a given point can be calculated by the time t the light travels with the help of the speed of light c .

$$d = \frac{t \cdot c}{2} \quad .$$

As the speed of light has a value of $c = 299,792,458 \frac{\text{m}}{\text{s}}$, it is hardly possible to measure the distance precisely. Therefore, the measurement of the distance between a light source and an object is based on the phase shift: A modulated light source emits a light pulse towards an object. The frequency for modulation is known and a phase shift can be measured from the reflected signal. The measurement principle is applied in parallel to a set of pixels, delivering a depth image \mathbf{I}_D of a scenery in the FOV. The phase shift for an arbitrary pixel ϕ_i is identified with a set of the amplitude s of a signal with four equally spaced samples $s_i(\tau_0)$, $s_i(\tau_1)$ and $s_i(\tau_2)$, $s_i(\tau_3)$, repeated after the modulated emitted signal reaches a multiple of $\pi/2$. The phase shift is calculated by

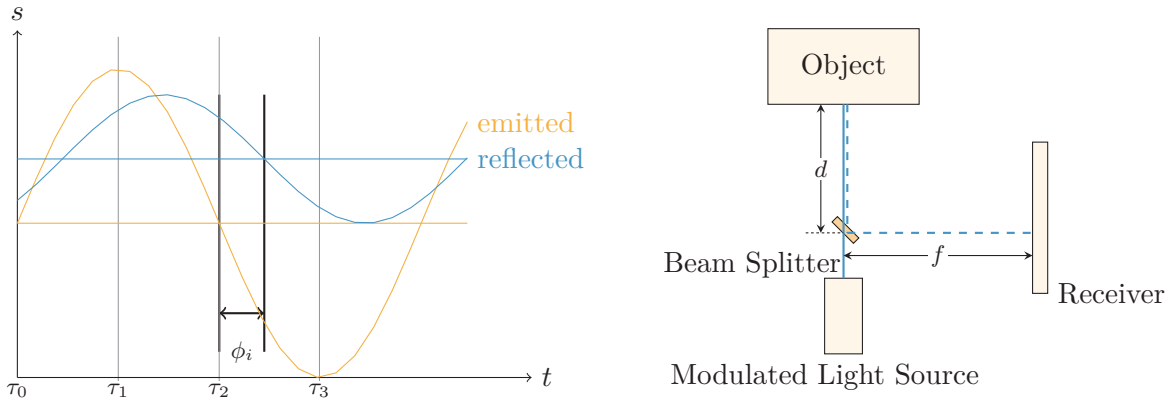
$$\phi_i = \arctan \left(\frac{s_i(\tau_0) - s_i(\tau_2)}{s_i(\tau_1) - s_i(\tau_3)} \right) \quad . \quad (3.1)$$

The distance between the camera and an object d_i for a pixel can finally be calculated based on the wavelength of the modulated signal λ_m and the phase shift by

$$d_i = \frac{\lambda_m}{2} \cdot \frac{\phi_i}{2\pi} \quad .$$

Figure 3.8 illustrates the ToF principle, as well as the structure of the sensor: A modulated light source is sent towards an object. A sensor detects the shift in phase of the modulated signal and therefore the distance d can be calculated.

The maximum distance, a ToF sensor can provide, depends on the frequency of the modulated light source. Due to the 2π -modulo, the ToF principle is limited in measuring a maximum distance by $d_{\max} = \frac{\lambda_m}{2}$ with $\phi = 2\pi$. A ToF sensor having a modulation frequency of 25 MHz has a wavelength of around $\lambda_m = \frac{c}{f_m} = \frac{c}{25 \text{ MHz}} = 12$ meters and is therefore limited to measure distances up to 6 meters. Measuring a distance of an object, which is eight meters away from the sensor will lead to the same phase shift, as an object, which is two meters away from the sensor. In conclusion, a lower frequency provides a bigger maximum range of the distance measurement, while a higher frequency can provide a distance measurement with more accuracy. For some ToF cameras the modulation frequency can be changed, to adapt to the necessary measurement range. With a bigger distance measured, the illuminating power of the modulated light source has to increase because the amplitude of the reflecting signal decreases proportionally to the inverse square of the distance [177, 132]. A ToF camera has the advantage that the distance



(a) Depth measurement via four equally spaced samples. The phase shift of the emitted and the reflected signal is calculated by equation (3.1).

(b) Principle of the depth measurement via modulated light source: The object's distance d is determined by the phase shift between the reference and the measurement signal. Source: [132]

Figure 3.8: ToF principle based on a sinusoidal modulated input signal.

measurement can be done quite fast, compared to a stereo or structured light sensor. ToF with high frame rates of 160 Hz exist and are suitable for real-time applications [90].

However, the ToF principle has to deal with some systematic errors, users should be aware of: Multiple way reflections occur at concave objects like corners of a room or object, placed on the ground. The measured distance of a pixel appears to be closer to the sensor than ground truth. Corners and hollows appear to be rounded off and they appear smoother when the light is traveling several paths at once, as shown in Figure 3.9. Additionally, the modulated light source can be outshined by other intense light sources, providing waves in the same wavelength. Also, problems are to be expected if multiple ToF cameras are used at the same time, facing the same FOV: When the cameras run on the same modulation frequency, interferences occur and distort the distance measurement of the sensor.

Another characteristic for ToF sensors are jumping edges: Measuring the distance of a pixel facing an edge in the real world, jumping edges can occur. Several approaches exist to filter or correct this error. During pre-processing a filter is applied, calculating the angle between neighboring pixels and filtering them out by a fixed threshold [132].

Figure 3.9 illustrates the result of the applied equation to remove jumping edge errors. In contrast to the removal of the errors, Poppinga et al. [169] demonstrate an approach to correct the jumping edge error. The approach is based on over 100 depth images and is computational complex.

The ToF camera can only deliver an intensity image, based on the reflectivity and the distance to an object. To obtain a depth image with color, a monocular camera is needed, which has to be calibrated towards the ToF sensor. Figure 3.10 demonstrates the sensor modalities of a ToF camera as a point cloud. May [132] illustrates approaches for segmentation and registration with ToF cameras, while Fuchs and Hirzinger [69] demonstrate the extrinsic and depth calibration of it.

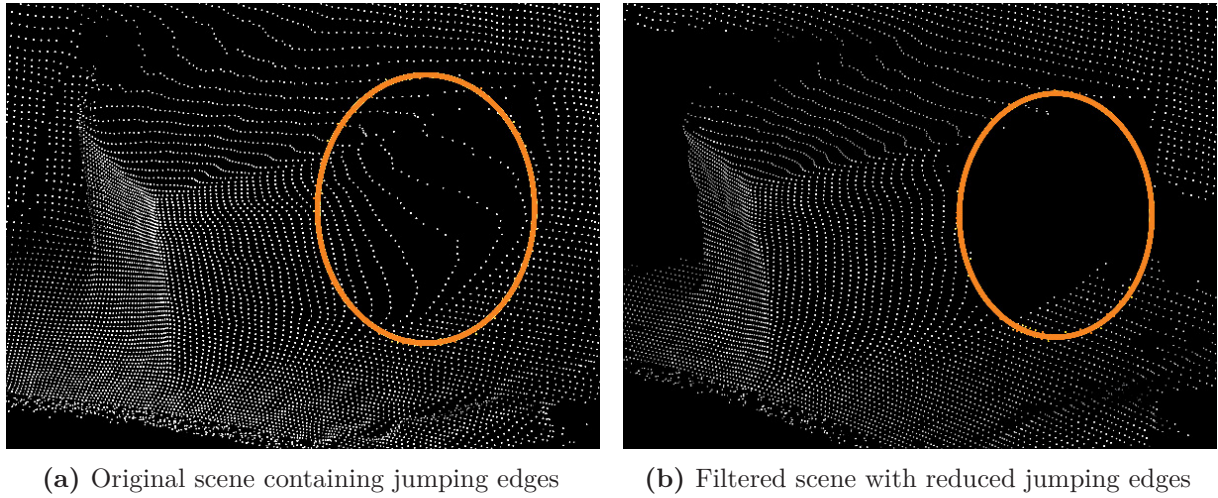


Figure 3.9: Removing jumping edge error from scene recorded with a ToF sensor: The scene shows a cube in front of a wall. The jumping edge error occurs between an edge close to the wall, resulting in several incorrect points (a). This issue can be fixed when the method presented by May [132] is applied, which removes the incorrect points by comparing them to their neighbors (b). Additionally to the jumping edge error, the edge close to the ground appears rounded off, due to multiple-way reflections.

For the here presented approach, a ToF camera is suited: The background light is limited, and sunlight cannot shine directly into the window-less trauma room. Furthermore, only a single ToF camera is intended to be used for data acquisition, so that no interference can occur.

Microsoft Kinect One

The Microsoft Kinect One is a ToF sensor. It was released as the successor of the Microsoft Kinect for the newer game console Microsoft Xbox One. Compared to the Kinect, the Kinect One provides a more detailed depth image, with higher resolution and less sensor noise. The aim of this improvement was the better recognition of the player's pose, for example, with this new sensor it was possible to recognize the user's finger, which allows more control for gaming. For gaming the sensor is placed in front of a television, facing away from the screen towards the user standing in place. The device has an Infrared (IR) sensor and a Light-emitting Diode (LED) light source to obtain the ToF principle. The camera already provides a fused stream of the sensor's data and is therefore intrinsically and extrinsically calibrated. Lachat et al. [113] present the characteristics of a Microsoft Kinect One camera and compare precision and accuracy towards ground truth. Moreover, Yang et al. [226] present an accuracy model for different areas in the FOV of the sensor and how the accuracy based on a trilateration method can improve accuracy for multiple Kinects simultaneously. The data from the Kinect sensor can be received via the OpenNI framework [151]. The framework is already integrated into the used library for point cloud processing, the Point Cloud Library (PCL), and gives access to all available sensor streams.

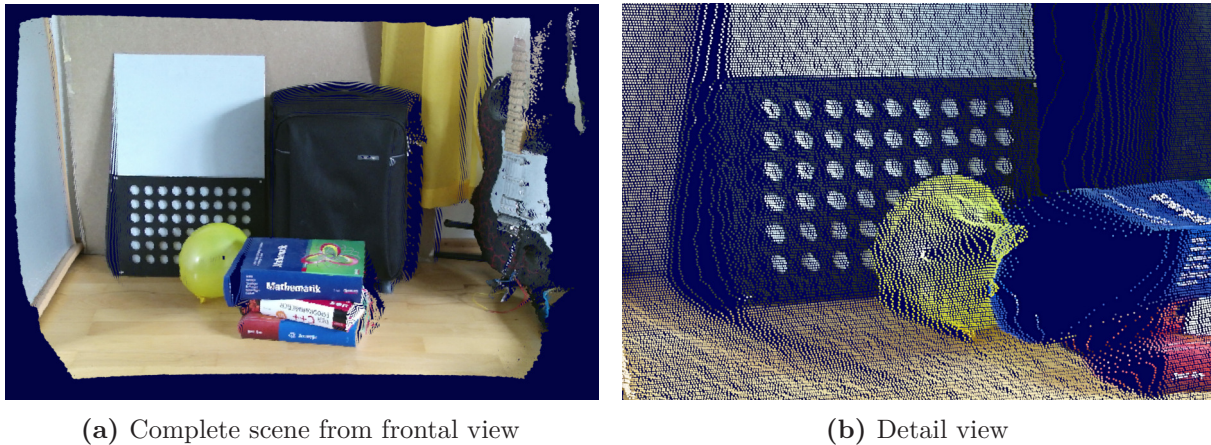


Figure 3.10: Scene from Kinect One camera: The scene contains several objects, like a balloon, a board with a circle pattern used for calibration, and a pile of books (a). As previously shown in Figure 3.9 jumping edges can occur. Here they are visible in the detail image, for example, at the edge of the calibration board. As the surface of the balloon is shiny, the resulting point cloud in this area is distorted (b).

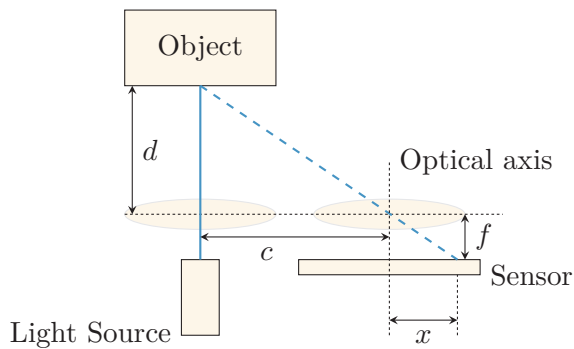


Figure 3.11: Depth measurement via active triangulation: First, a light source emits a single ray of light to the environment, and the object reflects the light. Second, a camera with a lens and a sensor recognizes the reflected light source. In this schematic, the offset to the sensors center x determines the distance to the object d . Due to the known transformation between the source and the sensor c , as well as the focal length f , the depth d can be calculated. Source: May [132]

3.5.3 Structured Light Sensor

Another approach to obtain a depth image is the structured light principle. Such a sensor consists of a projector and a monocular camera. To compute the depth towards an arbitrary pixel, a well-known pattern is emitted into the environment. With the monocular camera – and a fixed and known transformation between the projector and the camera – the structure is seen from a slightly different perspective. Based on active triangulation the depth can be obtained, as illustrated in Figure 3.11 for a simple sensor with a single emitted ray. The distance of the reflecting object d is calculated based on the focal length f , the distance between the light source and the sensor's centroid c , and the position on the sensor's surface the reflected light source hits by

$$d = f \frac{c}{x} .$$

calculated. Figure 3.12 shows the same scene with a Kinect camera, previously recorded by the Kinect One. A structured light sensor has to deal with similar disadvantages: The light source

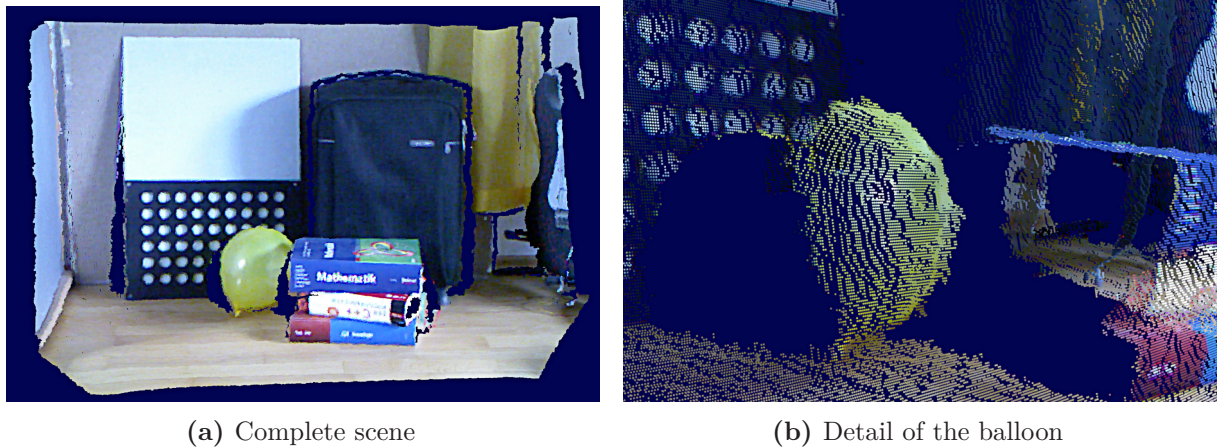


Figure 3.12: Scene from a Kinect camera: The scene is the same as observed by the Kinect One. In contrast to the scene recorded by the ToF camera, the point cloud contains more holes, for example at the edge of the board with the circle pattern. The text on the cover of the books is hardly readable because of the lower resolution (a). In a more detailed look, the sampling steps caused by the sensor are visible on the yellow balloon (b).

for the structured light pattern has to be powerful to be visible in illuminated scenes. Most structured light sensors work only in indoor lighting conditions and have issues to obtain depth measurements outside in bright sunlight. Furthermore, applying several structured light sensors to the same scene with an overlapping FOV can cause interferences for depth acquisition, due to the overlapping of the projected pattern in the scene. This issue can be solved when each camera is shaken slightly, as shown by Butler et al. [35]: They attached to several Kinect cameras a small motor with an offset weight, causing a vibration to the sensor. Now, due to the motion, each sensor sees only its own pattern sharply; while other projected patterns appear blurry and are not recognized for depth acquisition. In the detail image of the scene, steps appear in the point cloud. These steps get wider the further away an object is from the sensor. This step pattern occurs due to a rough sampling of the Kinect for depth. The effect can be reduced with state-of-the-art algorithms, like bilateral filtering [209] or a median filter [89].

Microsoft Kinect

The Microsoft Kinect camera is such a structured light sensor. It was released in November 2011 as an add-on for the Microsoft Xbox to enable the person in front of the sensor to control games with its movements. The sensor was the first of its kind to be used in game industry and a motor in the base is able to pitch the sensor, so it can adapt to the player's pose. Since its release, the Kinect started as a cheap but usable sensor for robotic's perception, having a big community [61]. Besides robotics, the sensor is also used in other applications, demanding for a sensor providing depth and color at the same time, for example, applications in healthcare [166, 150]. The sensor consists of an IR projector, an IR sensitive sensor to obtain the projected pattern and a color sensor. The image of the IR camera and the RGB camera are calibrated intrinsically; all cameras and the IR projector are further calibrated extrinsically by the manufacturer. Many publications

concerning the sensor's characteristic exist: Gonzalez-Jorge et al. [72] compare the Kinect and the Kinect One, concerning its characteristics and sensor's noise. Khoshelham and Elberink [102] illustrate the characteristics of the Kinect sensor for a mapping application. Nguyen et al. [144] analyze the sensor's noise with a focus to improve 3D reconstruction and tracking. They differ between axial noise and lateral noise, so a reconstruction with finer details of small objects is possible. Their work is an extension of the KinectFusion approach, first presented by Izadi et al. [93]. The lateral noise therefore mostly depends on the rotation angle towards a given reference plane θ . The axial noise σ_z depends on the depth of an arbitrary pixel and the angle towards the line of sight. With the equations presented by Nguyen et al. [144] for the lateral and axial noise, a model can be generated, reducing the impact of the noise of the Kinect camera. This approach is not limited to the Kinect camera and is suitable for all depth sensors.

3.5.4 Thermal Camera

The difference between a monocular camera which delivers data of intensities or a colored image, a thermal camera works at a different wavelength. The visible light is in a range between 380 nm – blue – and 780 nm – red. In contrast to that, a thermal camera works at a higher wavelength. Depending on the used sensor, thermal cameras can have a range in wavelength from 1 to 2 μm for Indium Gallium Arsenide sensors, 3 to 5 μm for Indium Antimony sensors, or 8 to 14 μm for Gallium Arsenide sensors. The optimal wavelength depends on the temperature measured in the scene: Short wavelengths close to the visible spectrum can be distracted by visible light, but are sufficient for high temperatures. With a long wavelength sensor, these distractions are minimized [216].

A thermal camera measures the emission of an object. Three principles for the transfer of heat exist: Emission ϵ , transmission τ , and reflectivity ρ . The sum of the three variables is always one, $\epsilon + \rho + \tau = 1$, meaning that an object with high emission close to one must have a small value for transmission and reflectivity [216]. Every body above absolute zero ($T_0 = 0\text{ K}$) emits energy by emission, which can be calculated by the Stefan-Boltzmann law [110].

Based on a calibration, the energy values for all pixels are converted into a temperature value. To get an interpretable image of a scene, the range of temperature over all pixels is set to a minimum and maximum value to amplify the gradient in the image. Figure 3.13 shows a scene with different false-color representations.

Temperature determination based on a contact-less method like a thermal camera can lead to lower measured value than ground truth: The reason for this is due to the different emission coefficient. While glossy or metallic surfaces tend to have a smaller coefficient, raw and dark surfaces have a higher coefficient closer to one. A low emission coefficient results in a reflection, and the thermal camera perceives not the temperature of the object itself, rather than the reflected object. In the here presented application the algorithms use only a relative temperature for segmentation of the human body. Furthermore, the human skin has a high emission coefficient of $\epsilon = 0.98$ and is therefore close to ideal emission coefficient. Lower emission coefficients in the scenario of a trauma room can appear on metal or lacquered surfaces at the medical stretchers or shiny parts of the patient's clothing, e.g., a belt buckle.

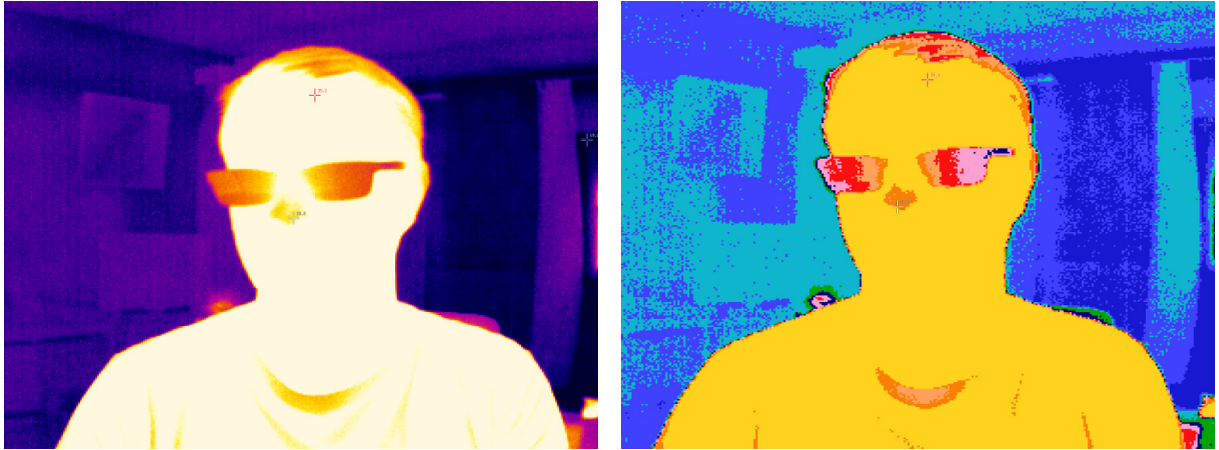


Figure 3.13: Person recorded with a thermal camera in different false-color representations: The scene was recorded with an Optris PI 400. The person in front of the camera can be easily segmented from the background. In both images, the range of the false-color is adapted to the minimum and maximum temperature.

Optris PI 400

For experiments, the thermal camera Optris PI 400 with a resolution of 382×288 pixels was used, attached to a lens of 62-degree FOV. The wavelength of this camera is in a range of 7.5 to 13 μm . Depending on the calibration, the camera can deliver a temperature range of -20°C to 100°C , 0°C to -250°C or 150°C to 900°C . To sense thermal data from people – which have in general a body temperature of about 37°C – the first range was chosen to acquire test data. The calibration file includes a function to improve the temperature measurement close to ground truth and is provided by the manufacturer. To provide absolute temperatures the Optris PI camera applies a thermal emission coefficient of $\epsilon = 0.95$. The image of the sensor is not rectified and contains radial and tangential distortions, which have to be removed by calibration (see section 3.7.1). The PI 400 has the advantage that the resolution of 382×288 pixels is quite high for a thermal camera of this size. Furthermore, the size of the FOV is similar to both 3D sensors, which is also good for sensor fusion. The low latency and the high update rate of 90 Hz is sufficient for sensor fusion and the segmentation can also be done for moving patients.

3.6 Representations of Sensor Data

The applied algorithms for segmentation and feature reconstruction rely on different representations. Three different representations are used in this thesis:

- **Intensity image:** An intensity image is a grid-based structure with pixels $\mathbf{q} = (u v)^T \in \mathbb{R}^2$, having a fixed width w and height h . Most projective sensors generate this representation: A monocular sensor provides values for the intensity of each pixel. Depending on the applied sensor, this intensity can either reflect the monochrome intensity of a black and white camera, a color image, or of a thermal camera with the temperature of a pixel based

on the pinhole camera model. An arbitrary intensity image is symbolized by $\mathbf{I}_{w \times h}$ while the number of pixels is given by $|\mathbf{I}| = w \cdot h$.

- **Depth image:** A range or depth image \mathbf{I}_D is provided by 3D sensors, e.g., 3D cameras or 3D Light Detection and Ranging (LIDAR)s. It has the same composition as the intensity image based on pixels $\mathbf{q} \in \mathbb{R}^2$. Every pixel corresponds to a distance $d(u, v)$ based on the pinhole camera model for a 3D camera. A range image is acquired by the previously introduced sensor principles, stereo triangulation, structured light or ToF. In case a distance cannot be obtained, for example, due to a limited measuring range, a distance value is marked as invalid, zero or to be Not a Number (NaN) value. Range and intensity images both have the benefit that the data is organized, which implies that the data is ordered in a grid and a point's neighbors are easily determined compared to a three-dimensional unorganized structure.
- **Point Cloud:** A Cartesian point cloud can also represent data from 3D sensors. Often, a point cloud is generated from a range image. Compared to the range image, the point cloud is less memory efficient, due to the extension towards three coordinates. Beside the Cartesian coordinates, each point $\mathbf{p} = (x \ y \ z)^T \in \mathbb{R}^3$ can contain additional values from sensor fusion, e.g., the color or the temperature. A point cloud is represented by $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$, while the number of points is given by $|\mathcal{P}| = n$. The size of a projective point cloud is defined by the width and height of the point cloud $|\mathcal{P}| = w \cdot h$. In contrast to image-based representations, point clouds might not be organized. Therefore, the search for neighbors can be expensive in computation, as shown in the appendix A.2 on page III. However, projective 3D sensors commonly provide an organized point cloud, where the neighbors can roughly be selected based on the index of a pixel in a range image.

The range and the intensity image have the benefit that they are smaller compared to a point cloud, due to the discrete pixel values for $u, v \in \mathbb{N}_0^+$, the lower precision of the applied data types and the two-dimensional space with \mathbb{R}^2 . Depending on the applied algorithm, it is necessary to transform from one representation to another. Intensity and range images allow processing and filtering in two dimensions.

3.7 Sensor Calibration

To achieve a correct sensor fusion, all sensors have to be calibrated. Figure 3.14 demonstrates the process of sensor calibration as it is presented in this section: First, all sensors are calibrated intrinsically. Second, the relative transformations between the sensors are estimated based on extrinsic calibration, providing the rotation \mathbf{R} and translation \mathbf{t} . Finally, the sensor signals are synchronized in time. Due to the rigidly mounted sensors (see Figure 3.15), the described calibration process has to be done once in advance to data acquisition for body weight estimation. If the sensors are moved relatively towards each other, the calibration process has to be repeated. It is not necessary to calibrate the sensor system towards the environment; the only constraint for the position of the sensors is that the patient on the stretcher is completely visible in the FOV of all sensors.

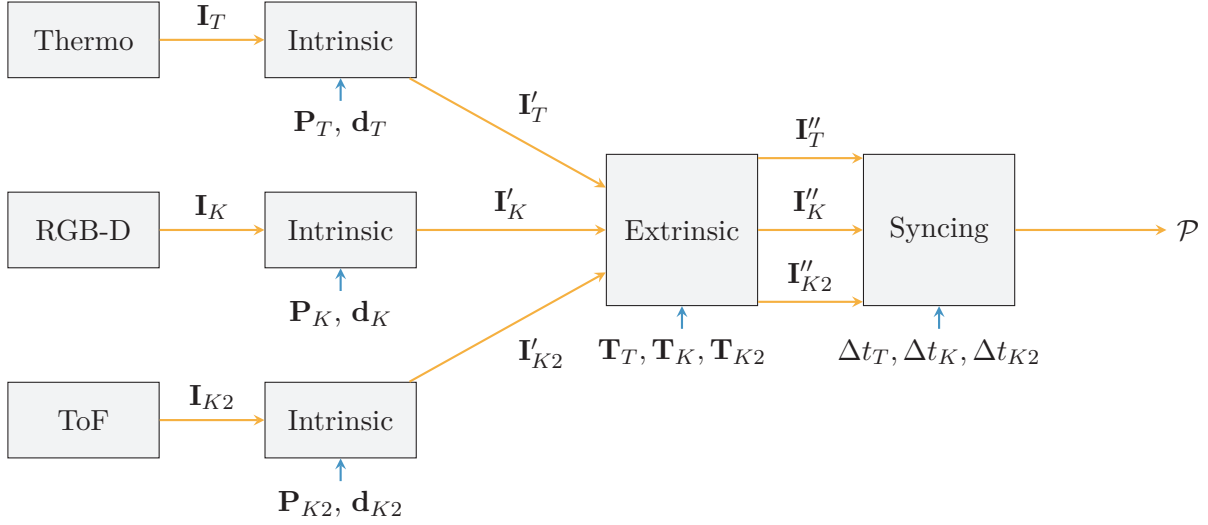


Figure 3.14: Process of sensor calibration: First, all projective sensors are calibrated intrinsically to remove distortions from the image and to obtain the projection matrix \mathbf{P} and the coefficients for distortions $\mathbf{d} = (k_1 \ k_2 \ k_3 \ p_1 \ p_2)$. Second, the sensors are calibrated extrinsically, estimating the transformations between the sensors \mathbf{T} . The calibrated images are noted by \mathbf{I}'_T , \mathbf{I}'_K , and \mathbf{I}'_{K2} . Finally, the data from the sensors are synchronized in time based on $\Delta t_T, \Delta t_K, \Delta t_{K2}$. The synchronized images are noted by $\mathbf{I}''_T, \mathbf{I}''_K, \mathbf{I}''_{K2}$, and \mathbf{I}''_T . After this process of calibration, sensor fusion can be applied and data is converted towards a Cartesian point cloud $\mathcal{P} \in \mathbb{R}^3$.

3.7.1 Intrinsic Calibration

Three-dimensional objects from the real world are projected on a camera's sensor and therefore are transformed into a planar two-dimensional surface. Hence, the information is reduced from \mathbb{R}^3 to \mathbb{R}^2 and the depth information is lost. However, knowing the size of an object in the real world, the distance to an object can be reconstructed.

The pinhole camera model is the elementary model to apply a projection for a camera. Pinhole cameras were the first cameras and consisted of a box, with a photosensitive material on one side of the box, and a tiny hole at the counterpart. Through the hole an image of the scene in front of the box is projected to the photosensitive material, turning the scene upside down. The smaller the hole, the sharper can be the scene projected as an image. Though, the photosensitive material has to be exposed longer to gather a bright image from the scene. Although pinhole cameras are free from spherical or chromatic aberrations, the obtained image is not very sharp and especially in the border area of the image darker and blurry. To enhance sharpness and to lower exposure time, real projective sensors use lenses, instead of a tiny hole. A bigger aperture ensures that more reflected light from a scene can hit the sensor's surface. Exposure time is therefore minimized. Unfortunately, a single lens can cause additional aberrations to an image, e.g., geometric and chromatic aberrations. To minimize the aberrations, different lenses are combined and grouped. This model implies that every object seen by the camera is pictured without any error on the sensor of the camera. Figure 3.16 describes the pinhole model graphically.

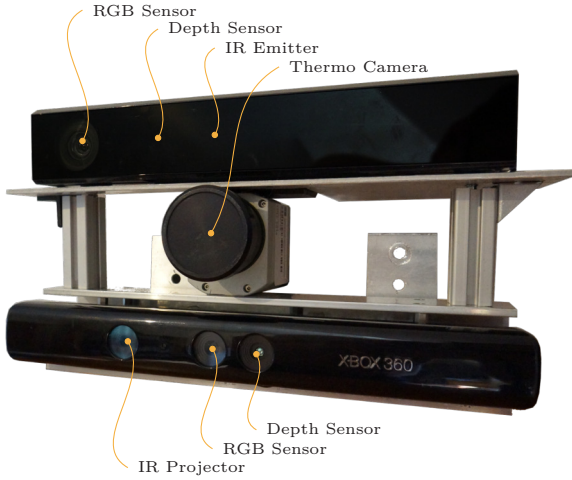


Figure 3.15: Rigidly mounted Kinect, Kinect One, and Optris Pi400: Parts of the housings of the two RGB-D cameras are removed. All sensors are mounted with aluminum profiles to prevent a loss of extrinsic calibration. The top sensor is the Kinect One, while the RGB and depth sensor, as well as the IR emitter are hardly visible. The bottom sensor is the Kinect with an IR projector and an RGB and depth sensor. In the middle of the sensor system, the Optris PI400 is mounted.

The pinhole model is applied to correct these distortions. This proceeding is called intrinsic calibration and is often referred to Zhang [230]. The pinhole model is described with a projection matrix \mathbf{P} . With the intrinsic calibration, the errors in a projection of a lens are corrected and transformed to a pinhole model. Moreover, having an intrinsically calibrated device, the scaling factor between the sensor plane and the real world can be achieved. The pinhole model is applied for all projective sensors, like Kinect, Kinect One, and the thermal imager. The model consists of the focal length (f_x, f_y) and the center of the image has to be known (u_0, v_0) . Additional, skewness γ of the image axis is added in case the image axes are not perpendicular; otherwise, the skewness is set to zero. Based on the intrinsic parameters, an equation can be formed to project a point from the environment $\mathbf{p} = (x \ y \ z)^T \in \mathbb{R}^3$ onto the sensor, transforming it in a two-dimensional image coordinate $\mathbf{q} = (u' \ v')^T \in \mathbb{R}^2$:

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{P}} \cdot \underbrace{\begin{pmatrix} x/z \\ y/z \\ 1 \end{pmatrix}}_{\mathbf{p}}, \quad (3.2)$$

where a different focal length can be applied for the x - and the y -axis [230].

The sensor can be illustrated as a matrix with w columns and h rows. Therefore, the pixel coordinates u and v can only have discrete values with $0 \leq u < w$ and $0 \leq v < h$ where $u, v, w, h \in \mathbb{N}^+$. The focal length f describes the distance between a lens and the focal point. Having a lens with a small focal length, e.g., 8 mm the device has a wider FOV compared to a high focal length like 120 mm. Having a pinhole camera model, a distance for an arbitrary pixel can be calculated by applying the intercept theorem by

$$\frac{f}{z} = \frac{u}{x} = \frac{v}{y} .$$

The image plane is spanned by the axis x and y . The coordinate frame for the image plane is right-handed with the z -axis along the optical axis, as shown in Figure 3.16.

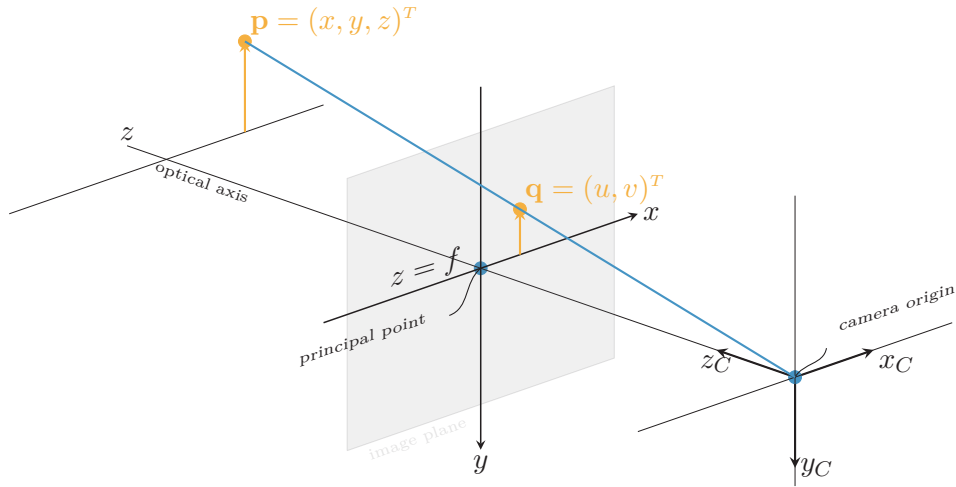


Figure 3.16: The pinhole camera model: A point from the environment $\mathbf{P} \in \mathbb{R}^3$ is projected onto the image plane, which is spanned by the two vectors \mathbf{x} and \mathbf{y} , and is represented by $\mathbf{q} \in \mathbb{R}^2$. Source: [45]

Based on the pinhole camera model, an object can be projected onto a sensor's surface. Unfortunately, camera lenses have non-linear distortions, which have to be corrected to apply the pinhole camera model. Lines in the real world are projected to the sensor's plane curved. These spherical aberrations can be separated in radial and tangential distortions. Radial distortions translate image points along radial lines from the principal point.

The coordinates from the sensor's plane (u, v) are moved via the sensor's center \mathbf{q}_c to (u_0, v_0) . First, the radial distortion is corrected: The radius r is defined as a sensor point's distance from the principal point by

$$r = \sqrt{u^2 + v^2} \quad ,$$

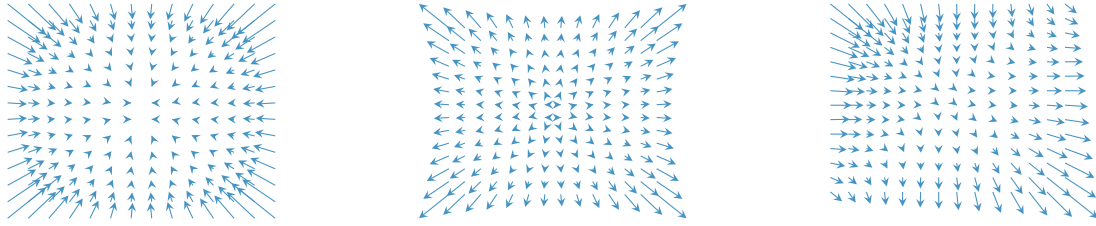
where the image points are transformed to have the principal point as their reference coordinate frame where $u = u' - u_0$ and $v = v' - v_0$.

When the magnification decreases with growing radius, barrel distortions appear. This distortion type often turns up at wide angle or fisheye lenses. Figure 3.17a illustrates such a barrel distortion. Vice versa, pincushion distortions appear when the magnification increases when the radius is growing. This kind of distortion is shown in Figure 3.17b. The correction factor for the radial distortions is calculated by a polynomial

$$1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots \quad ,$$

where the degree of a polynomial to calculate k can vary. For good lenses, a low degree can be good enough for a sufficient calibration. With a growing degree of the polynomial, the computational cost increase for calibration.

Commonly, the radial distortion has a significantly higher impact in aberration compared to the tangential distortions. Tangential distortions occur at right angles to the radii. This



(a) Barrel distortion
where $k_1 = -0.0004$

(b) Pincushion distortion
where $k_1 = 0.0004$

(c) Tangential distortion
where $p_1 = -0.003$, $p_2 = 0.003$

Figure 3.17: Different kind of distortions: A positive radial distortion coefficient results in a pincushion distortion (a), while a negative coefficient will result in a barrel distortion (b). Having a tangential distortion leads to a skewed image (c). The rectification is applied by equation (3.3).

kind of distortion is illustrated in Figure 3.17c. The parameters for the tangential distortion are described by p_1 and p_2 and the distortion $\delta_{\mathbf{t}}$ is modeled by

$$\delta_{\mathbf{t}} = \begin{pmatrix} 2p_1u'v' + p_2(r^2 + 2u'^2) \\ p_1(r^2 + 2y'^2) + 2p_2u'v' \end{pmatrix} .$$

Finally, the image can be rectified based on the distortion parameters $\mathbf{d} = (k_1, k_2, k_3, p_1, p_2)$ by

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u' \\ v' \end{pmatrix} \underbrace{\left(1 + k_1r^2 + k_2r^4 + k_3r^6 + \dots\right)}_{\text{radial distortion}} + \underbrace{\begin{pmatrix} 2p_1u'v' + p_2(r^2 + 2u'^2) \\ p_1(r^2 + 2y'^2) + 2p_2u'v' \end{pmatrix}}_{\text{tangential distortion}} , \quad (3.3)$$

where u and v describe the rectified pixels. Although the Kinect and the Kinect One are pre-calibrated by the manufacturer and tend to have nearly no noticeable distortions, all sensors are calibrated intrinsically. The obtained projection matrix, the radial, and the tangential distortion are used to rectify the images received from the cameras.

3.7.2 Extrinsic Calibration

The relationship between the environment, points in the environment, and the sensors are described by different relations and coordinate frames. Three different coordinate frames are used:

- **World coordinate frame:** The world coordinate frame $\{0\}$ is based on a three-dimensional Cartesian coordinate system and defines the origin of the scene's world. This origin can be a unique landmark in the scene or can be determined by an arbitrary point $\mathbf{p}_0 = \{0\} = (0 \ 0 \ 0)^T \in \mathbb{R}^3$.
- **Camera coordinate frame:** Points within the coordinate frame are relative to the camera sensor's center. The z-axis is usually perpendicular to the image plane. The *camera coordinate frame* is related to the *world coordinate frame* $\{0\}$ by extrinsic parameters – rotation $\mathbf{R}_{3 \times 3}$ and translation $\mathbf{t} = (t_x \ t_y \ t_z)^T$ of the coordinates.

- **Image coordinate frame:** Coordinates $\mathbf{q} = (u_0 \ v_0)^T \in \mathbb{R}^2$ are related towards the image's center \mathbf{q}_c . The origin of this coordinate system is usually in the upper-left of the image. The *image coordinate frame* and the *camera coordinate frame* are related to the perspective projection of the points onto the image plane.

The previously explained intrinsic calibration is the basis for the now following extrinsic calibration. The sensors are mounted rigidly towards each other, so the sensor's frame cannot change in translation or rotation. Extrinsic calibration aims to estimate the poses between all sensors and therefore to determine the pose of an object's coordinate frame with respect to the camera's frame. Figure 3.18 illustrates the extrinsic transformation of each sensor device to each other in the shape of a transformation tree. All sensor frames are right-handed, with the sensor's plane spanned by the x - and y -vector.

Both 3D devices consist of several cameras, each having a pre-calibrated transformation \mathbf{T} . The Microsoft Kinect sensor consists of two sensors: One sensor for color imaging $\{K_c\}$ and one for infrared imaging $\{K_i\}$. Due to the structured light principle, the Kinect is equipped with an infrared projector. To calculate a depth image, its pose $\{K_p\}$ also has to be known. The infrared frame of the Kinect is used as global frame $\{0\}$ and therefore defines the Euclidean origin with $(0 \ 0 \ 0)^T$. The Kinect One works with the ToF principle and has, therefore, an infrared sensor $\{K_{2i}\}$. The relative transformation to the color sensor $\{K_c\}$ is known by factory calibration. The pose of the light source does not affect the calculation of a depth image from a ToF sensor and can be neglected.

The thermal imager consists of one sensor with the frame $\{T\}$. The extrinsic calibration between the sensors uses the global frame $\{0\}$ as a reference. Therefore calibration is done in relation to the infrared frame of the Kinect sensor.

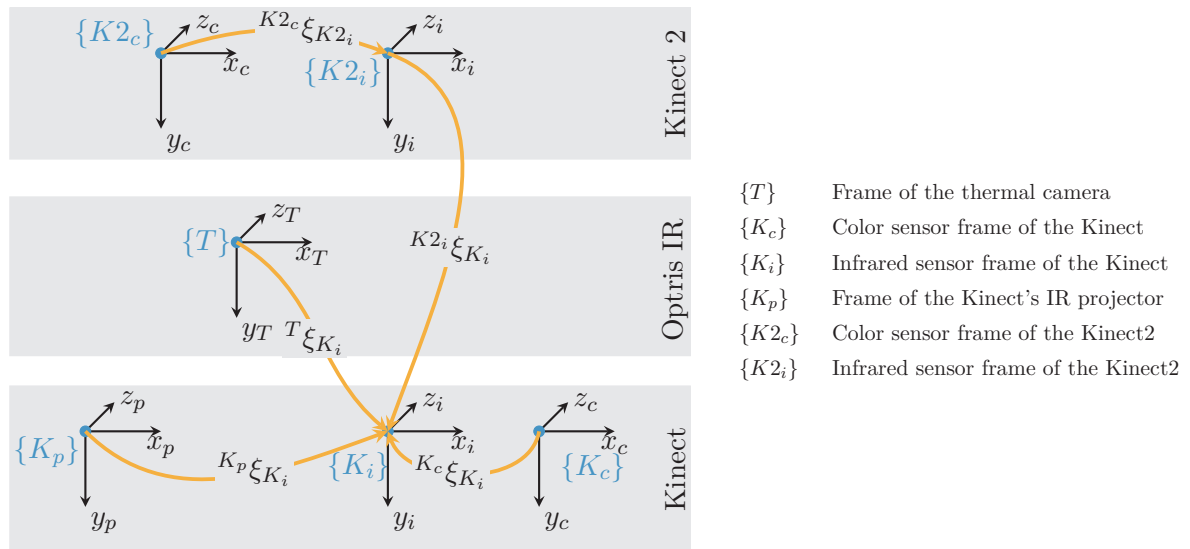


Figure 3.18: Transformation tree for the system's sensors: The infrared frame of the Kinect is used as a reference for the world coordinate origin $\{0\}$. The 3D sensor's own sensor frames are already calibrated by the manufacturer. To obtain the transformation between the Kinect and the Kinect One, the IR sensors from both cameras are taken as reference.

The relative pose of a frame with respect to another coordinate frame is expressed by ${}^A\xi_B$, which defines the relative pose from a coordinate frame $\{B\}$ with respect to a frame $\{A\}$. A point ${}^B\mathbf{p}$ can be transformed into the coordinate frame $\{B\}$ by ${}^A\mathbf{p} = {}^A\xi_B {}^B\mathbf{p}$.

To perform calibration, the geometry of a target has to be well known. Often, for camera calibration, chessboards patterns are used. The corners of such a pattern can be detected reliably, e.g., by the Harris corner detector [79]. The necessary characteristics for a calibration in this scenario is described in the upcoming section on page 50. A calibration target delivers a set of n points $\mathbf{p}_i = (x_i \ y_i \ z_i)^T \in \mathcal{P}$ where $i \in [1, n]$ with respect to the target's coordinate frame. To apply the projection of the target's point onto the sensor's image plane, the intrinsic parameters have to be known.

For extrinsic calibration, the method proposed by Zhang [230] is applied. The here presented derivation is taken from Nüchter [149]: Corresponding 3D and 2D points are first used to calculate a solution based on a homography matrix $\mathbf{H} \in \mathbb{R}^4$

$$s \begin{pmatrix} \mathbf{p}' \\ 1 \end{pmatrix} = \mathbf{H} \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix},$$

where s is a scaling parameter. To estimate the homography matrix at least four points are necessary, e.g., the four outside markers on the calibration pattern. The equation forms an over-specified system of equations. In an initial step, the equation is solved with a low amount of corresponding point pairs to approximate \mathbf{H} . Furthermore, a non-linear optimization algorithm like the Levenberg-Marquardt algorithm is used to improve the outcome of the approximation [138]. The algorithm applies a maximum likelihood criterion with

$$\sum_{i=1}^n \sum_{j=1}^m \|\mathbf{p}'_{ij} - \hat{\mathbf{p}}_{ij}(\mathbf{H}_j)\| \quad ,$$

where \mathbf{p}'_{ij} defines projections of the 3D points in the image and $\hat{\mathbf{p}}$ represents the points which are projected based on the homography matrix \mathbf{H} . Afterward, the optimized homography matrix \mathbf{H} is taken to calculate the projection matrix \mathbf{P} and the extrinsic parameters, the rotation \mathbf{R} and the translation \mathbf{t} ,

$$\sum_{i=1}^n \sum_{j=1}^m \|\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij}(\mathbf{P}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{d})\| \quad ,$$

where \mathbf{p}_{ij} describes the detected coordinates of the j -th corner in the i -th image. Furthermore, $\hat{\mathbf{p}}_{ij}$ defines the projection of an arbitrary point in real world coordinates with the parameters for the intrinsic projection matrix \mathbf{P} , the rotation matrix \mathbf{R}_j , the translation vector \mathbf{t}_j , as well as the radial and tangential distortion parameters $\mathbf{d} = (k_1, k_2, k_3, p_1, p_2)$.

Finally, the extrinsic calibration can be applied to the previously known pinhole model. The projection matrix of the intrinsic calibration \mathbf{P} is multiplied with the transformation matrix \mathbf{T} and the point $\mathbf{p} \in \mathbb{R}^3$ to get the projection onto the image plane by

$$s \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{P}} \cdot \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}}_{\mathbf{R}, \mathbf{t}} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad . \quad (3.4)$$

The here presented approach for calibration is programmed based on the image processing framework OpenCV [28, 50].

3.7.3 Multimodal Calibration Target

The applied calibration pattern for intrinsic calibration – and further also for extrinsic calibration – has to be visible for all sensors in color, depth and thermal stream. For monocular RGB or monochrome cameras commonly a chess or circle pattern is applied. The patterns are recognizable with state-of-the-art algorithms, e.g Harris corner detection for a chessboard patterns [79] or the Hough transform for circle detection [14]. Such a pattern can be printed on a piece of paper or cardboard and represents a strong gradient, easy to detect in the sensor’s data.

Luhmann et al. [125] illustrate different calibration methods for close range application for various types of thermal cameras. An object with a unique pattern has to be visible for the thermal camera. Therefore, Luhmann et al. [125] build a calibration target with an aluminum plate and a pattern of self-adhesive foil on it. The target is placed outside, facing towards the sky. While the foil reflects the temperature of itself, the shiny aluminum plate reflects the cold temperature of space. This leads to a high gradient in temperature, visible for the sensor and the pattern can be recognized with low effort and state-of-the-art image processing algorithms.

Visibility in a thermal image can be achieved with two solutions: On one side the pattern can be heated up, showing a pattern with high gradient. On the other side, a difference in emission would also be possible. Figure 3.19 demonstrates the here applied target for calibration in a color, and a thermal image. It consists of two layers: The back side of the pattern is made from aluminum. It is heated up to a temperature significantly above ambient temperature. Furthermore, the surface of the aluminum is coated white. The front side is made from a wooden plane which has a circle pattern with holes. The wooden plane is manufactured with a laser cutter, providing a highly precise and accurate pattern. The distance between the two surfaces ensures that the difference in temperature is kept: Moreover, it is painted black and has a distance of around 1 cm towards the aluminum plane. The temperature at the front stays close to the ambient temperature, while the plane is heated and slowly loses temperature over time to the environment. This pattern ensures to be visible in all available sensor streams. The size of the calibration target is adapted to the clinical environment, with the sensors mounted rigidly in the ceiling. The target board has a size of 30 × 40 cm and fits in the FOV of all sensors. On the other side, the size of the circles on the target is big enough to be visible from the distance close to the ground of the room.

The algorithm for calibration is integrated into the *Libra3D* application in the hospital. During the calibration, the software provides a rectangular marker in the viewer of the software, in which the target should be moved. When the target is close to the marker and steady, a frame from the sensors is acquired for calibration. Afterward, a new marker is viewed at a different position. The change of the marker in the viewer leads to several changes in perspective and size, the target is visible for the sensors. This routine was developed, that also non-experts – e.g., physicians – are able to calibrate the system on their own. The routine for calibration is described in Listing 1.

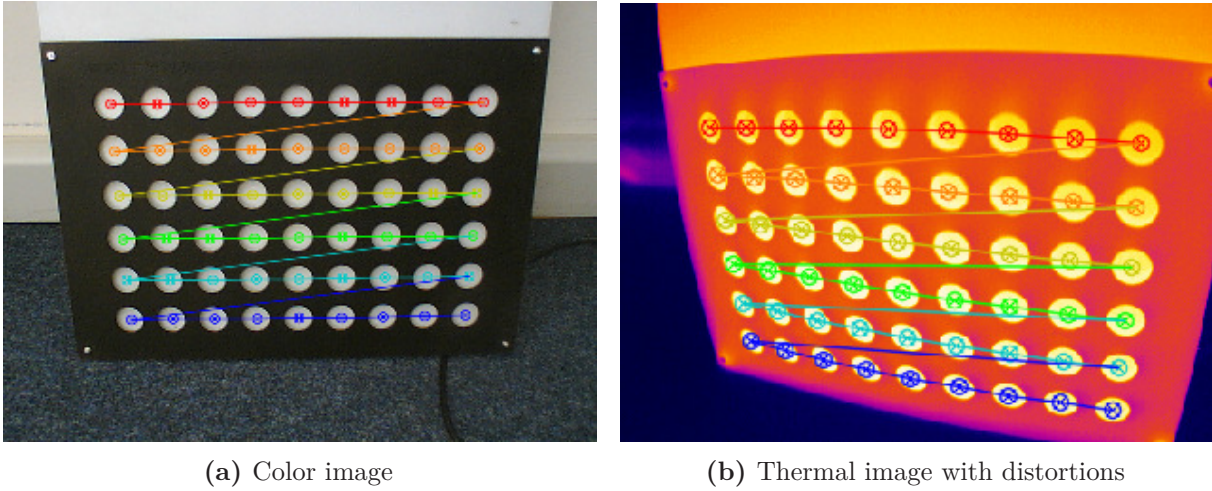


Figure 3.19: Calibration of the color image and calibration pattern visible in thermal image: The colored markers in the scene illustrate the detected circle pattern. While the lines for a row in the color image are nearly straight (a), the lines visible in the thermal image appear curved (b).

Algorithm 1: Process of extrinsic calibration.

1. Search pattern in the current sensor frame. If found, add the image to a set of calibration images. Repeat this, until a sufficient number of n frames is acquired.
 2. Apply intrinsic calibration to get the distortion parameters $\mathbf{d} = (k_1, k_2, k_3, p_1, p_2)$.
 3. Remove distortions from images in the set, by applying equation (3.3).
 4. Apply extrinsic calibration, to get the translation \mathbf{t} and the rotation \mathbf{R} , as shown in equation (3.4).
 5. Save the calibration $(\mathbf{d}, \mathbf{R}, \mathbf{t})$ to the computer and the database.
-

3.7.4 Noise Model Calibration for Depth Sensors

The noise of depth sensors is often not homogeneous, due to their working principle: Small ToF cameras with a single LED can lead to the problem that corners of the image plane are not sufficiently lighted. This effect is known by vignetting [132]. As shown previously for the ToF camera, the measured distance depends on the amplitude of the modulated source as well as the amplitude of the background light.

For a static scene, a set of n continuous depth images $\{\mathbf{I}_{D_1}, \dots, \mathbf{I}_{D_i}, \dots, \mathbf{I}_{D_n}\}$ can be recorded and averaged to a new depth image $\bar{\mathbf{I}}_D$ over a fixed number of n frames with

$$\bar{\mathbf{I}}_D = \frac{1}{n} \sum_{i=1}^n \mathbf{I}_{D_i} \quad .$$

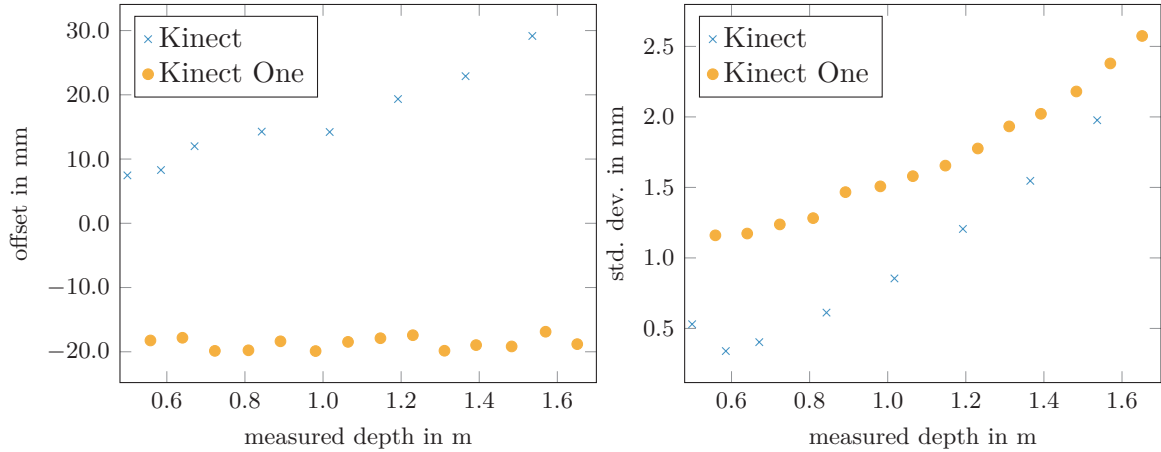


Figure 3.20: Comparison of the sensor’s noise between the Kinect and the Kinect One: While the offset to ground truth distance of the Kinect increases over depth, the offset of the Kinect One stays close to around -20 mm, as shown in the left plot. Moreover, both sensors have an increasing standard deviation over distance, with a square relation, as shown in the right plot. Source: [218]

This procedure is low in computational cost but demands a static scene. If an object in the FOV is moving over the number of frames, distortions are generated. While the patient is lying on a medical stretcher, he or she does not move much, commonly. However, the number of frames taken for averaging should be evaluated in advance. A number of 10 frames means for a sensor grabbing data with 30 Hz, that the subject must not move for 330 ms.

Another method to minimize the impact of the sensor’s noise is a correction model for each pixel. Therefore, the camera is aligned towards a plain surface, measuring the distance towards the plane over several frames, averaging them and comparing it to ground truth.

Wasenmüller and Stricker [218] compare the Kinect and the Kinect One concerning its accuracy and precision, see Figure 3.20: While the accuracy and precision of the Kinect is not affected by the color of the reflecting surface, the distance provided by the Kinect One depends on the surface and changes the offset to ground truth. Moreover, for both sensors the standard deviation of the sensor’s noise increase over the measured depth; both in a similar way. In contrast to that, the offset to the ground truth distance increases over the measured distance for the Kinect, while the Kinect One has a nearly fixed offset of around 18 mm. The authors further illustrate the jumping edge error appearing in the data from the Kinect One. Additionally, they looked for thermal correlations: While the mean distance of the Kinect changes slightly over time when the camera is powered, the Kinect One shows a correlation of the sensor’s temperature to the average distance. As the Kinect One has a fan included, the mean measured distance can change if the fan is spinning. The authors propose to run the sensor for around 25 minutes until a stable mean distance is reached.

To improve the quality of an RGB-D sensor, bilateral filtering should be applied [209]. The filter replaces the intensity of a pixel with an averaged intensity of the neighboring pixels. The intensities of the nearby pixel are weighted with a Gaussian kernel. The benefit of this procedure is that edges are preserved, while the values in areas are smoothed. Figure 3.21 shows the

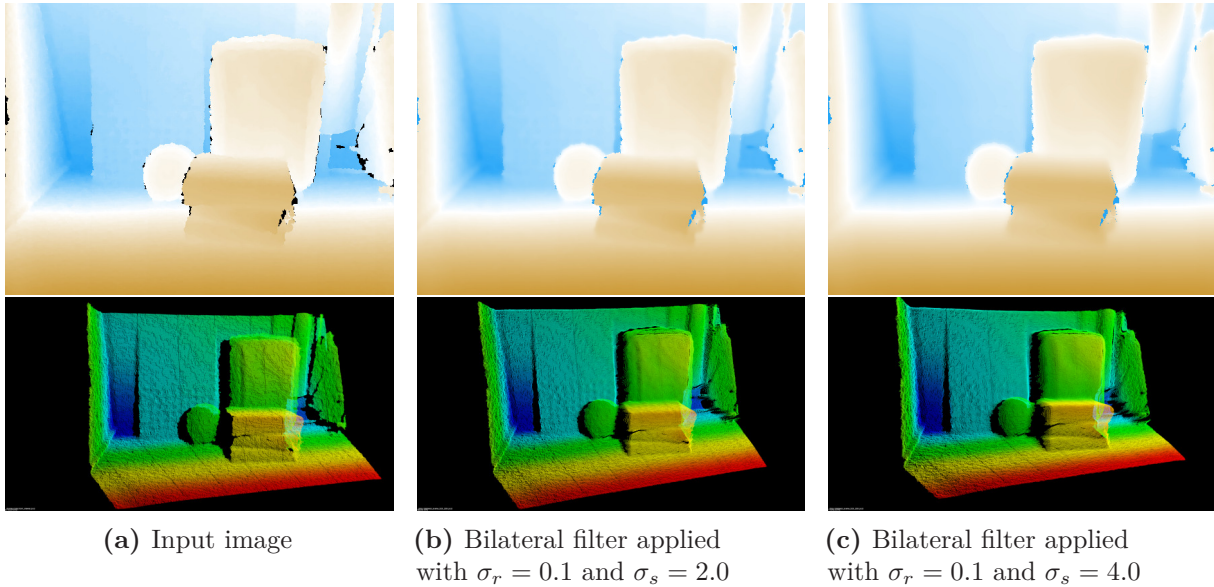


Figure 3.21: Depth image and point cloud from a scene, filtered with a bilateral filter: With the increase of the σ_s value, the surface of the point cloud is smoothed, while the edges are preserved. With a high value, jumping edges can occur. The bilateral filter has to be adapted to the scene.

smoothing of a depth image with a bilateral filter. This filter can be applied on gray, as well as on a depth image. Two parameters control the behavior of the bilateral filter: The standard deviation σ_s weights the pixels used for filtering based on a radial distance. In contrast to that, the standard deviation σ_r weights the pixels used for filtering based on the distance in color. Figure 3.21 illustrates different settings for a scene recorded with the Kinect camera.

3.7.5 Syncing of multiple Sensor Streams

Having a set of multiple sensors is the basis to apply sensor fusion. Due to the data processing in the sensor and the communication via different transmission technology (USB, Ethernet, etc.) the data can be delayed in time. Especially if the scene is dynamic, such a delay can cause errors in sensor fusion, fusing old data from one sensor with up-to-date data from another sensor. One approach is presented by Lussier and Thrun [126], syncing a thermal and an RGB-D without a specific target: A person moving in the field-of-view generates an optical flow in both images. The number of pixels moving in each image is summed up. Temporal minima and maxima can be extracted from a plot. The difference in time syncing can be estimated when the extremes are aligned by a difference in time Δt , as presented in Figure 3.22a. This works best if the person is easy to segment, e.g., no other heat sources and only the person is moving in the scene. Figure 3.22b illustrates the aligned edges in the thermal, depth, and thermal frame. For the medical scenario, most patients on the stretcher do not move much. Therefore, the synchronization in time is not necessary.

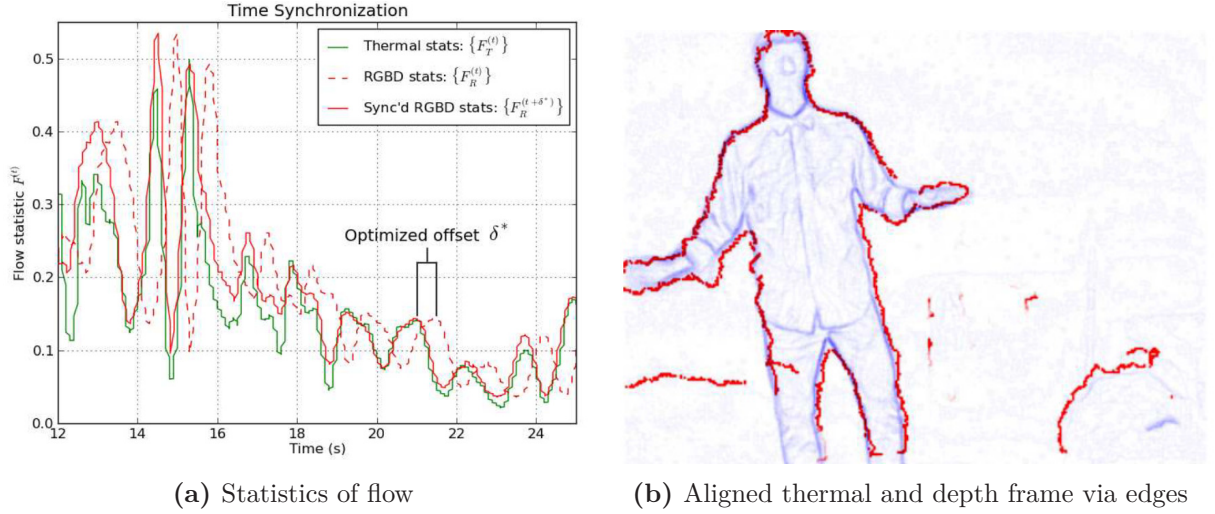


Figure 3.22: Synchronization of multiple sensors based on optical flow: In the experiment provided by Lussier and Thrun [126] a thermal camera and an RGB-D sensor are used. The data from the thermal camera has less delay compared to the depth sensor, which is visible at the peak values in the plot (a). In the presented approach, the extrinsic calibration is applied via edge detection of a subject moving in front of the cameras (b).

3.8 Sensor Fusion

Based on the previously defined intrinsic and extrinsic calibration, as well as the time synchronization, the data from the sensors can be fused. After this processing, the sensor's data can be used to generate a point cloud, containing Cartesian points $\mathbf{p} \in \mathbb{R}^3$. After sensor fusion, all sensor streams are converted towards a point cloud \mathcal{P} where

$$\mathcal{P} = \{\mathbf{p}_i \mid \mathbf{p}_i \in \mathbb{R}^3, i = 1, \dots, |\mathcal{P}|\} \quad .$$

Every point in the cloud is able to hold all data from all sensors: Each of these points can contain a color $\mathbf{p}_i \rightarrow c_{\text{RGB}}$ and additionally thermal information $\mathbf{p}_i \rightarrow T$. Due to the more narrow FOV of the thermal camera, some points do not contain a thermal value and are marked as NaN value. Thermal data is represented by false-color representation. Here an iron palette is applied, giving a dark blue color for low temperatures and a bright yellow color for high temperatures. This palette is emulated from the glowing light of heated iron and is quite common for the visualization of thermal data.

Pixel-wise comparison applies the visualization of sensor fusion of the color and temperature data of the color: The color image from the RGB-D sensor \mathbf{I}_C , as well as the converted false-color representation from the thermal camera \mathbf{I}_T are split into color channels red r , green g , and blue b . Now, for every available pixel, the values of each channel are compared. The channel with the biggest value for an overlying pixel is copied to the fused representation by

$$\mathbf{I}(i) = \begin{cases} \max(\mathbf{I}_T(i) \rightarrow r, \mathbf{I}_C(i) \rightarrow r) \\ \max(\mathbf{I}_T(i) \rightarrow g, \mathbf{I}_C(i) \rightarrow g) \\ \max(\mathbf{I}_T(i) \rightarrow b, \mathbf{I}_C(i) \rightarrow b) \end{cases} \quad \text{where } i = 1, \dots, n \quad ,$$

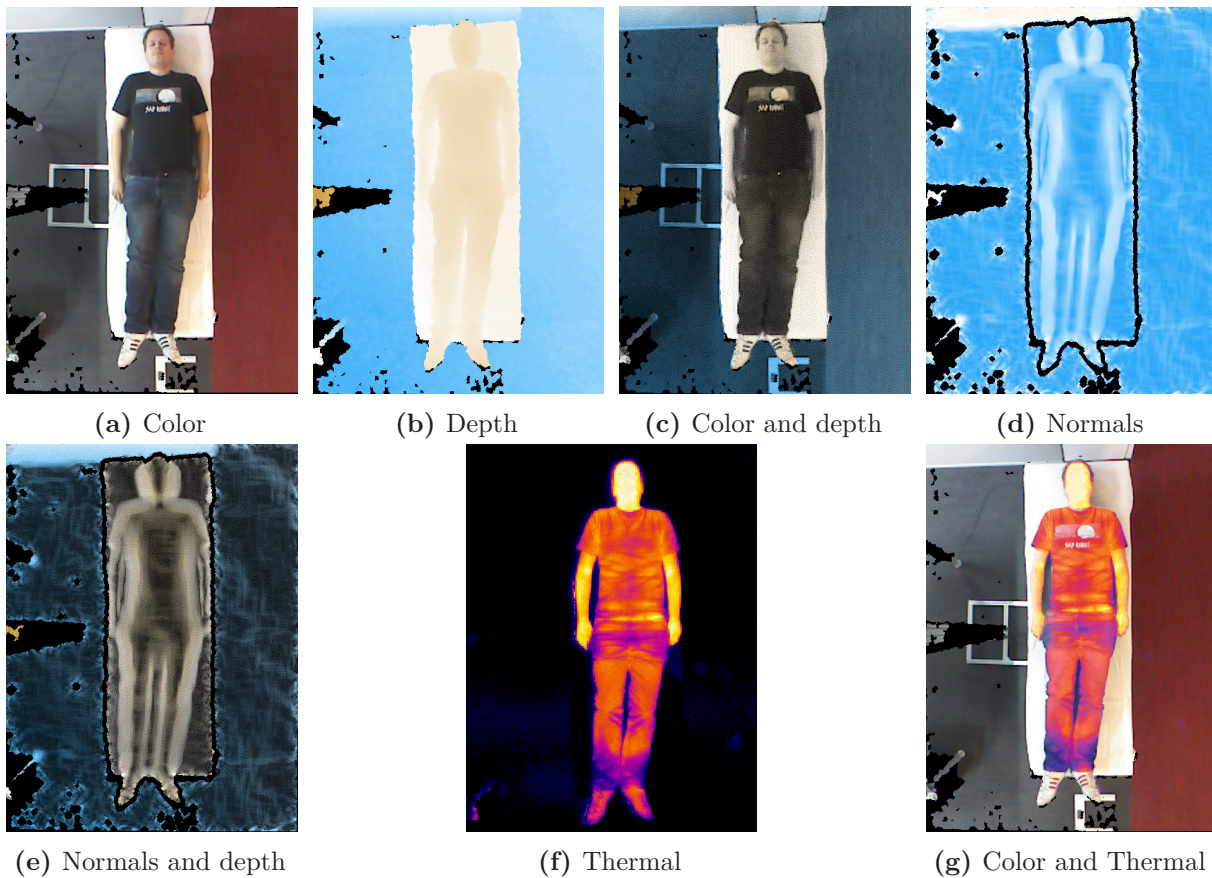


Figure 3.23: Visualization of the sensor fusion: Figure (a) shows the raw color stream from an RGB-D camera. The depth stream can be visualized using a colormap, here drawing blue values for far objects and orange for nearer objects (b). Both streams could be fused in one image multiplying the depth image with the gray-scaled intensities of the color image (c). Calculated normals can also be represented by a color map (d). The stream from the thermal camera can be visualized in several ways: Either it is drawn with false-color representation (f) or can be combined with other streams – here a combination of the color stream, highlighted with temperature (g).

for n pixels. This representation for the combination of color and thermal information is first shown by Vidas et al. [214]. The benefit of such a fusion is, that the texture of the color image is preserved while the color gives an indicator for the temperature. Figure 3.23 illustrates the sensor fusion and its visualizations.

3.9 Summary

The system's conceptual design is adapted to the requirements requested by the physicians of the stroke team of the University Hospital Erlangen. The physicians demand that they are not hindered during treatment. The camera system is hidden under the suspended ceiling in the trauma room. The changes in the room, for example, the markers on the floor and the cable

conduit, are minimal and it can be removed with low effort. The system is easy to use via a GUI and should save data for offline optimization and processing, which is done in a MySQL database. Moreover, this section presented the necessary calibration routine for sensor fusion, which is based on state-of-the-art sensor fusion algorithms. The intrinsic and extrinsic sensor calibration is performed on all visual sensors. Finally, a stream of sensor data is available, fusing color, depth, and thermal data, which is necessary for the upcoming segmentation and body weight estimation.

Chapter 4

Segmentation of Humans from the Environment

Segmentation is one of the most common applications when it comes to image processing [205, 229]. An image – consisting of a set or matrix of pixels – is grouped into different parts, while those parts receive a label. By segmentation, object classification can be established. Segmentation is represented in different ways: Either a set of pixel forms a closed group. Or otherwise, a border through a single set divides this sets into two groups. Such a segment consists of one or more pixels. Whether a group or edge-based representation is chosen depends on the applied algorithm. Segmentation can be applied to different sensor data: The most common way in image processing is the segmentation based on a color image, provided by a monocular camera. Besides from being ideal, segmentation can have two different other states: Having an over-segmentation will result in a high number of segments. Different illumination with bright regions and dark shadows or an object having a highly textured surface will likely result in an over-segmentation with too many objects [205]. On the other side, there can also appear under-segmentation, where too few segments are created. In such a case, the segmentation is not successful and should provide more different segments. To prevent over- or under-segmentation, the right method should be chosen, as well as a proper configuration. Looking for a green ball on grassland will result in an under-segmentation if a segmentation based on color is chosen.

Segmentation of people within images and video streams is essential for various applications. Especially if the segmentation and the detection are needed for safety application, for example, collision avoidance of an autonomous driving system, the algorithms must provide a reliable result of the segmentation and detection within a specified duration of time. An algorithm for the detection of humans often has to deal with various difficulties, like partial occlusions due to overlaps or different perspectives. Furthermore, some approaches illustrate the segmentation and the tracking of several humans at once [185, 231].

With the focus on medical applications, segmentation helps radiologists and physicians to detect issues in CT or MRI images [187]. Segmentation on the basis of medical imaging can improve the diagnosis and can detect tumors or infections earlier. This guidance in medical imaging is often referred to as Computer-Aided Diagnosis (CAD).

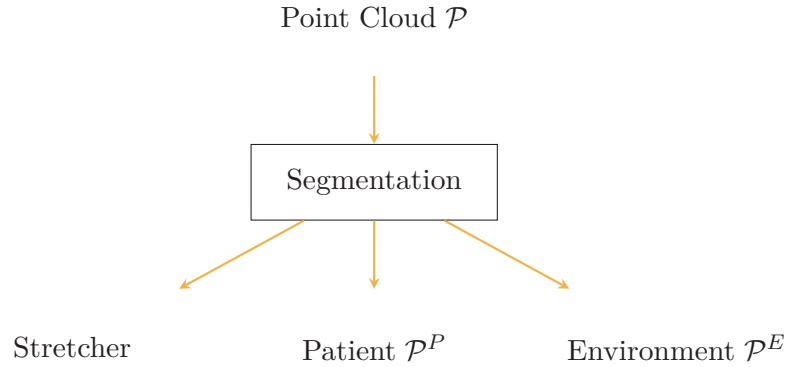


Figure 4.1: Result from segmentation: After the segmentation, a point cloud containing the patient \mathcal{P}^P is segmented from the environment \mathcal{P}^E . Additionally, the plane model for the stretcher is estimated within the segmentation.

Depending on the scenario, it can be hard to segment a person from the scene with a monocular camera. Such a scenario could be a person in front of a white wall, while the person in the FOV of the camera wears white clothes. The illumination and shadows can increase the reliability of the segmentation in such a scenario. However, the segmentation and detection of a human body with a 3D sensor can improve the outcome in such a situation.

With the focus on reliable and precise segmentation, sensor fusion is essential. While the segmentation on a single sensor stream can work in most cases, there are often conditions the segmentation might fail from time to time. Segmentation based on multiple sensor streams increases the performance of segmentation, applying more different methods to distinguish between various objects or people. Therefore, for the segmentation of humans in a mixed and natural environment often the combination of a thermal, depth and color camera is used [153].

For the here applied scenario, it is essential to distinguish between the patient, the environment or other people within the FOV. The result of the upcoming approaches for segmentation are presented with images from the sensor's perspective, while the filtered parts are illustrated by a blue overlay. Parts in the image, which are not removed due to the filtering, are unmodified.

4.1 Scenario for Segmentation

The patients have different body heights, different shapes or also different clothes in shape and color. However, it is the task of the segmentation, so the subject is recognized reliably and with sufficient precision. Furthermore, the body weight estimation system should not rely on a particular type of stretcher as several types of different stretchers are used in most hospitals. Also, most beds can be adjusted in height to ease medical treatment for the physicians. This excludes the precise modeling of such a stretcher to handle the segmentation. In case the model for the stretcher is fixed, and only the position and orientation can be changed, the location of it could be found via the Iterative Closest Point (ICP) algorithm [22].

The data provided by sensor fusion contains the patient, as well as the environment from the sensor’s view. To process the body weight, the first step is to differ which data belongs to the patient and is therefore relevant for body weight estimation, and which part belongs to the environment. Because the patient is lying on a flat surface – the medical stretcher – the segmentation in a depth image is more complicated compared to a person standing with a significant distance to the environment. The points \mathbf{p}_i belonging to the patient are described by $\mathcal{P}^P \subset \mathcal{P}$ while the size of the patient’s data is set by the cardinality of the set $|\mathcal{P}^P|$. Figure 4.1 illustrates the different sets of data after the segmentation which are necessary for the upcoming machine learning approach.

The great variety of different subjects is one problem for segmentation, and hardly any color of the subject’s can be set as given. Another challenge in the segmentation of such a scene is the close distance of the patient to the surface he or she is lying on. That is why the segmentation based on depth data is challenging. Moreover, the segmentation of the stretcher can be distorted when people are standing close to the stretcher, e.g., physicians treating the patient. Nevertheless, the segmentation of static parts in the scene, like the floor, can be done with low afford, because the sensors are not moving and mounted rigidly in relation to the room.

The process of segmentation is structured as follows: First, the amount of data in the scene is reduced by removing known and static objects, like the floor, by a simple distance threshold. In a second step, the stretcher with the patient on it is segmented and localized. Now with the scene only containing the stretcher and the patient, the patient is segmented from the stretcher. The surface of the bed, which is modeled as a plane, is also segmented from the rest of the stretcher. This plane is needed for the estimation of the back surface of the patient and is necessary for the upcoming feature extraction and body weight estimation. Finally, minor distortions are removed from the segmented subject, and the result gains robustness and precision.

The benefit of algorithms working on bit masks is the increased speed in processing. Furthermore, data processing is kept to a minimum. Another benefit of working with binary segments is that Boolean algebra can be applied to the different sets. However, binary mask have an issue, because the size of the binary mask correlates with the size of the point cloud: For every point in the point cloud, an element in the mask has to exist. Even if a point cloud is reduced to one-tenth of its original size, the size of the filter mask would be still the same. To fix this issue, the upcoming implemented filters rely on indices [155]. The set of indices only contains the index of the elements in the point cloud, which are valid after filtering. Therefore, the set of indices reduces the amount of data removed from the scene. A set of indices for a point cloud, which is reduced to one-tenth, also has the size of one-tenth of its indices; while a binary mask containing only valid points has the same amount of elements as in the indices set. Depending on the scenario or the algorithm, indices can be converted to filter masks and vice-versa.

4.2 Bounding Box Filter

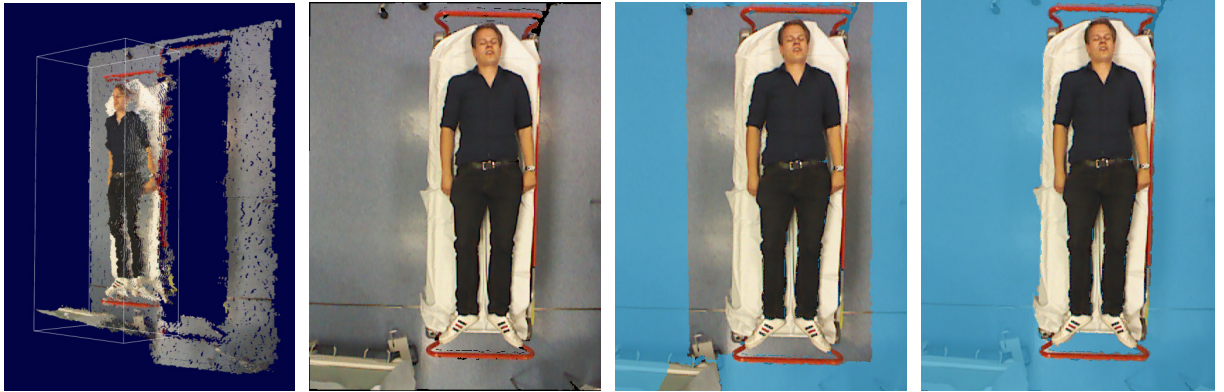
Because the sensors are mounted rigidly to the ceiling and the fixed environment, the easiest way to reduce the size of the point cloud \mathcal{P} is a depth filter. Points in a certain range of depth $d = [d_{\min}, d_{\max}]$ filter out the floor and objects close to the sensor. The range has to be adapted to possible heights of different patients, considering especially the adjustable height

of the medical stretcher. People like physicians close to the patient could still be included in the weight estimation. Therefore a bounding box is applied: The box is defined in width $w = [w_{\min}, w_{\max}]$, height $h = [h_{\min}, h_{\max}]$, and depth $d = [d_{\min}, d_{\max}]$ and includes points to the patient by intersection

$$\begin{aligned} \mathcal{P}^* &= \mathcal{P}_H^* \cap \mathcal{P}_W^* \cap \mathcal{P}_D^* && \text{where} && (4.1) \\ \mathcal{P}_H^* &= \{\mathbf{p}_i \mid h_{\min} < \mathbf{p}_i \rightarrow x < h_{\max}\} \\ \mathcal{P}_W^* &= \{\mathbf{p}_i \mid w_{\min} < \mathbf{p}_i \rightarrow y < w_{\max}\} \\ \mathcal{P}_D^* &= \{\mathbf{p}_i \mid d_{\min} < \mathbf{p}_i \rightarrow z < d_{\max}\} \end{aligned}$$

The output of the filtering for the three dimensions with width \mathcal{P}_W^* , height \mathcal{P}_H^* , and depth \mathcal{P}_D^* can be calculated independently. The bounding box for the experiments in the hospital environment is about 2.5 meters long, 1.2 meters wide and about 0.8 meters high in order to guarantee that the patient fits into this volume. These settings are chosen for the test application in the trauma room. In case of a different environment with a different stretcher, these values have to be adapted. A smaller range guarantees that only the patient is set for weight estimation and people close to the patient are excluded. The bigger the bounding box is set in the range, the easier the patient can be placed inside the FOV with fewer constraints. Attached markers on the floor assist to place the stretcher – and therefore the patient – in the FOV of the sensors. In a sensor live stream, a physician can supervise that the patient is entirely visible to the sensors.

Figure 4.2 illustrates the bounding box filter: First, the scene is visualized as a point cloud in Figure 4.2a. The marker for the bounding box is also shown in the GUI provided for the physicians. The point cloud and the box can be rotated to reach different perspectives of the scene, to ensure that the patient is completely within the bounding box. The whole scene contains 307,200 points, as shown in an above view by Figure 4.2b. The amount of data is reduced by



(a) Scene with a bounding box, visualized as a points wire mesh model

(b) Data size of 307,200 points

(c) Data size of 134,121 points

(d) Data size of 75,442 points

Figure 4.2: Reducing the point cloud's size by bounding box filter: To ease the correct positioning of the stretcher, a wireframe model visualizes the configuration of the bounding box in the scene (a). The bounding box reduces the size of the scene in the medical scenario to around one quarter (b)-(d).

applying the filter for x- and y-direction, see Figure 4.2c. Therefore, the output of the filter includes the stretcher with the subject, as well as the floor of the room. The amount of data in the input point cloud is further reduced to about one third when the bounding box filter is applied to all dimensions, see Figure 4.2d. The reduction of the point cloud at an early stage is essential to speed up upcoming steps in processing.

The previously presented bounding box filter is easy to apply and reduces the point cloud dramatically. However, after this step in filtering, the scene contains the stretcher including the subject, but also parts of the stretcher which are not necessary for weight estimation, like the handlebars of the stretcher. The handlebars of the stretchers in the emergency room are all coated red, but stretchers from other hospital wards can look different and therefore filtering by color cannot always be guaranteed. The bounding box presented in the previous section is aligned fixed to the coordinate's axes. To minimize the data for body weight processing further, another bounding box is generated. In contrast to the presented bounding box, this one is oriented towards the already filtered data and is smaller in its dimensions.

Figure 4.3 illustrates the application of the minimum oriented bounding box filter: The filter has the task to segment the patient including the stretcher. The biggest box marked in the schematic is the previously applied bounding box filter. Afterwards, the orientation of the resulting data is calculated, and a smaller bounding box is adapted to the scene, also containing the patient and the stretcher. Now, the newly oriented bounding box can be reduced in size to remove data from the point cloud aligned with the principal components.

Based on Principal Components Analysis (PCA), the input data is aligned to the coordinate's axis. Figure 4.4 demonstrates the oriented bounding boxes as a schematic. The appendix illustrates the principle of the PCA on page IV.

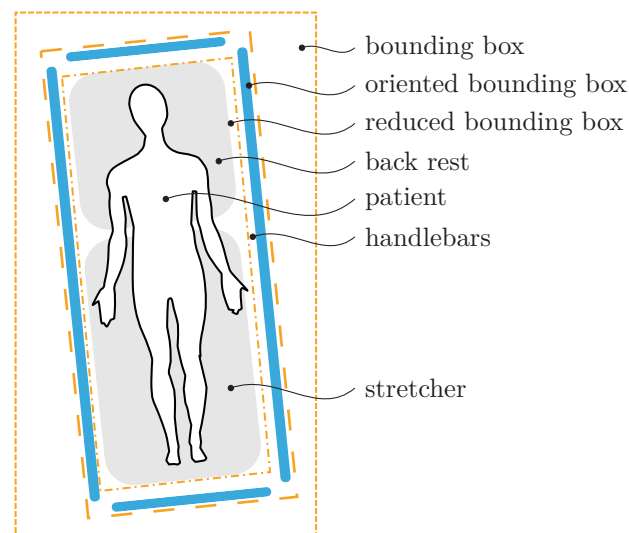


Figure 4.3: Bounding box for pre-filtering and oriented minimum bounding box around the stretcher with patient.

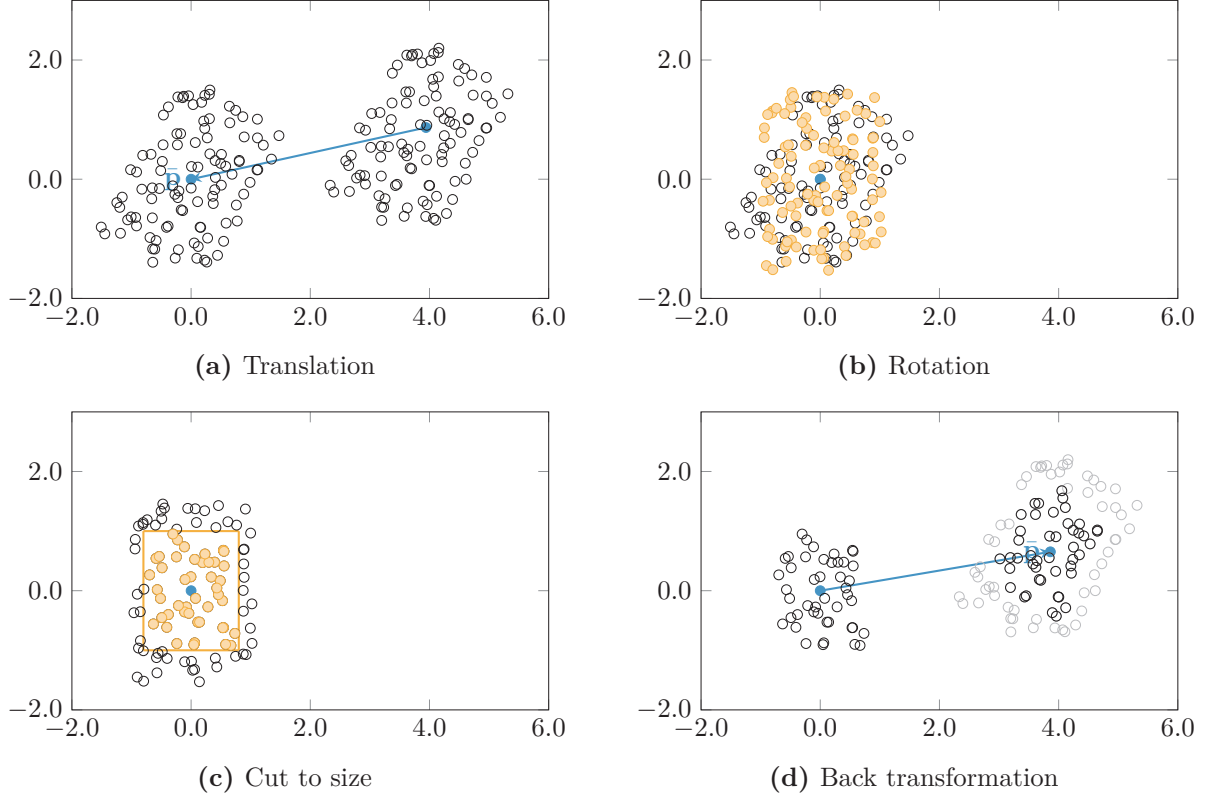


Figure 4.4: Minimum Bounding Box Filter: A set of given points \mathcal{P} should be cut along its principal components (a). Therefore, the set is first moved to the origin of the coordinate system based on its centroid $\bar{\mathbf{p}}$. With the help of the PCA, the set of points can be aligned to the axis of the coordinate system (b). The previously presented bounding box filter is then applied (c). Finally, the set of points is rotated and translated with the help of the inverse transformation, and the final result of the minimum bounding box filter is reached (d).

First, the scene \mathcal{P} is moved via a translation \mathbf{t} towards the origin of the coordinate frame, so the centroid $\bar{\mathbf{p}} = (\bar{x} \ \bar{y} \ \bar{z})^T$ and the scene's origin align. The complete point cloud \mathcal{P} is moved to the centroid by a homogeneous transformation matrix \mathbf{T} by

$$\mathcal{P}' = \{\mathbf{T} \cdot \mathbf{p}_i \mid \mathbf{p}_i \in \mathcal{P}\} \quad \text{where } \mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & \bar{x} \\ 0 & 1 & 0 & \bar{y} \\ 0 & 0 & 1 & \bar{z} \\ 0 & 0 & 0 & 1 \end{pmatrix} .$$

Second, the now shifted scene \mathcal{P}' is rotated via the eigenvalues \mathbf{v}_1 , \mathbf{v}_2 , and \mathbf{v}_3 , calculated from the covariance matrix Σ , so the principal component with the highest value aligns with the x-axis. Moreover, the second principal component aligns with the y-axis, and the third element aligns with the z-axis. The rotation of the point cloud \mathcal{P}' is calculated by

$$\mathcal{P}'' = \{\mathbf{V} \cdot \mathbf{p}_i \mid \mathbf{p}_i \in \mathcal{P}'\} \quad \text{where } \mathbf{V} = (\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3) ,$$

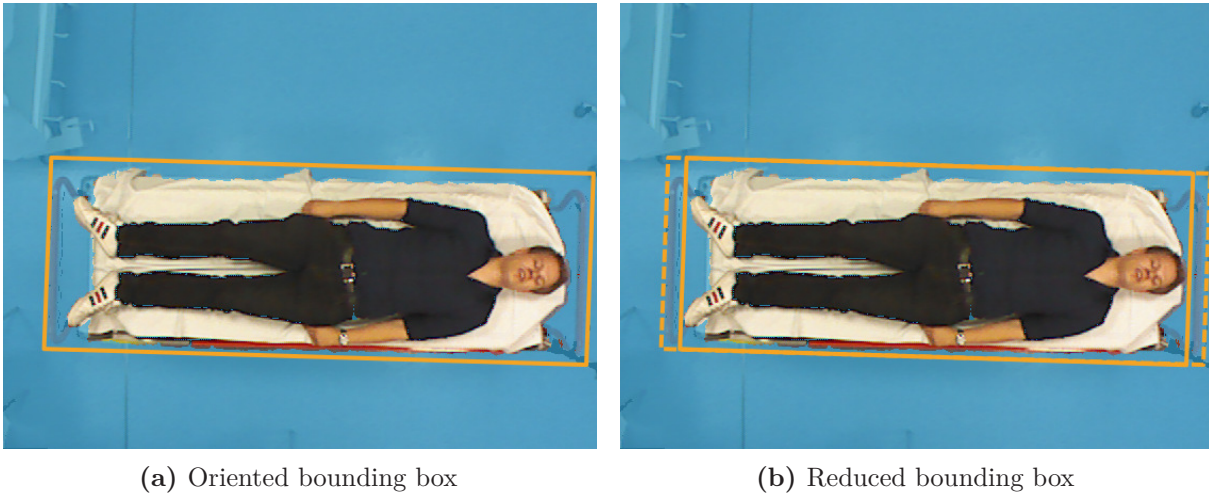


Figure 4.5: Filtering based on oriented minimal bounding box: The filter is applied to resize the scene along its principal components. While the red handlebars on top and bottom the stretcher in (a) are inside of the bounding box, they can easily be removed from it by reducing the length of the box (b).

and is shown in Figure 4.4b. Third, the bounding box filter, as shown in equation (4.1) is applied to the transformed data set. The used filter obtains its coordinate system based on the origin of the centroid of the set of points. Last, the point cloud is transformed back to its initial coordinate frame.

Figure 4.5 shows the result of the filter: First, the oriented bounding box is calculated based on previously extracted data. The box is aligned and oriented to the point cloud’s data (a). Second, the bounding box is reduced in length with the help of a fixed value, but keeping the same angles (b). Therefore, the handlebars at the top and the bottom of the stretcher are removed.

The calculation of the minimum oriented bounding box ensures that only the patient and the stretcher is in the FOV. The excluded points from this filter are helpful to improve the outcome of plane segmentation, to estimate the position and inliers of the Random Sample Consensus (RANSAC) algorithm. To remove the handlebars and the stretcher’s grid on the side of the stretcher (see Figure 4.3), the minimum bounding box filter is applied. To filter this scenario the starting set of points must only contain the stretcher with the patient on it. Other things from the environment have to be removed previously, e.g., the floor or people close to the stretcher.

4.3 Thermal Filter

The used thermal camera provides good data for sensor fusion and segmentation: While the trauma room is air-conditioned and kept at a temperature of around 21 °C, people in the FOV mostly have a higher temperature. The human body has a temperature of around 37 °C on the inside. The thermal imaging camera detects a lower temperature at the surface of the skin. The thermal range $T = [T_{\min}, T_{\max}]$ is set depending on the ambient temperature of the trauma room.

A minimum and a maximum threshold for the temperature define the output of the thermal filter \mathcal{P}^* by

$$\mathcal{P}^* = \{\mathbf{p}_i \mid T_{\min} < \mathbf{p}_i \rightarrow T < T_{\max}\} \quad .$$

Experiments showed that the minimum temperature should be set about 2 to 3 °C higher than ambient temperature.

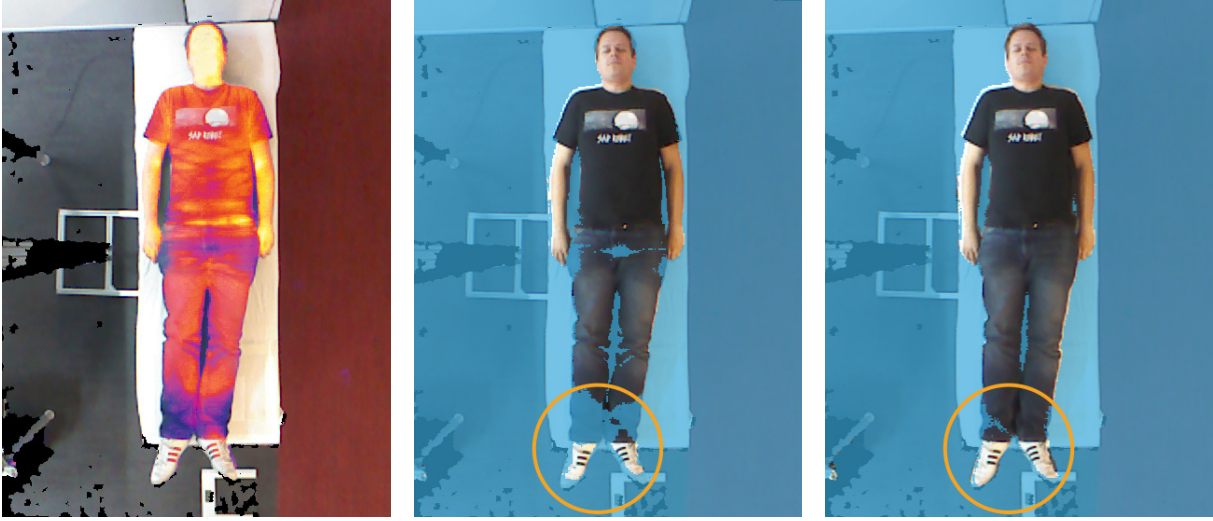
Figure 4.6 demonstrates the result of an applied thermal filter to the scene: The input point cloud contains thermal data in a range of 23.4 to 34.6 °C (a). The ambient temperature in this scene is around 23.5 °C and therefore close to the minimum temperature. The subject is clearly visible due to the gradient in temperature to the environment. Skin visible to the thermal imager has a higher temperature compared to the clothes the subject is wearing. Furthermore, the temperature depends on the thickness of the clothes, as well if clothes are tightly or loosely fitting. The temperature of the shirt, the subject is wearing is higher than the jeans. Especially the shoes and the lower parts of the jeans have a temperature closer to the ambient temperature. While the shoes consist of relatively thick material, the jeans only have few contact to the leg. If the thermal filter is applied to the scene with a lower threshold of 27 °C, parts of the subject are excluded (b). The tip of the shoe is excluded as well as parts of the jeans. To fix this issue in filtering, the threshold temperature is lowered to 25 °C. This leads to a bigger segment, containing nearly every point of the subject (c). Now the difference between the ambient temperature of 23.5 °C is only 1.5 K. Lowering the threshold for filtering further leads to noise in the segmentation, and parts of the stretcher are added to the set of the subject. If the scene does contain objects and devices which can have a higher temperature than the ambient temperature, the maximum threshold should be set. This should ensure that parts of a medical device, close to the patient, do not confuse the filter.

In case a patient is hypothermic because he or she had a stroke and got unconsciousness outside in the winter, the thermal filter can cause issues if working with a fixed temperature. In some cases, the temperature of a subject's hand was below 15 °C. Working in such a scene with a fixed temperature close to the ambient temperature, the hand would be filtered from the scene. For such an event the physicians are supervised to have a close look at the result of the segmentation on the display in the trauma room.

The filtering based on the thermal camera is not essential. However, it can improve the outcome of the segmentation and leads therefore to a more robust weight estimation.

4.4 Color Filter

As mentioned in the introduction of this section, color can be a reliable source for segmentation if the color of an object is predictable. The surface of the bed is always covered with a blank white sheet due to hygiene issues. Because of high hygiene standards in a trauma room, the medical stretcher is cleaned after each patient, which includes changing the cover of the stretcher. The white color of the sheet helps to check for contaminations. For color filtering, the value of the color is converted to the Hue, Saturation and Value (HSV) color space which is more suitable for filtering in this scenario. The color filter is defined by the range of the hue value $c_h = [c_{h_{\min}}, c_{h_{\max}}]$, the saturation $c_s = [c_{s_{\min}}, c_{s_{\max}}]$ as well as the value $c_v = [c_{v_{\min}}, c_{v_{\max}}]$. Additionally, the ambient



(a) Input point cloud containing thermal data in a range of 23.4 to 34.6 °C (b) Thermal filter with setting of $T_{\min} = 27\text{ }^{\circ}\text{C}$, $T_{\max} = 38\text{ }^{\circ}\text{C}$ (c) Thermal filter with setting of $T_{\min} = 25\text{ }^{\circ}\text{C}$, $T_{\max} = 38\text{ }^{\circ}\text{C}$

Figure 4.6: Applied thermal filter to segment a patient from the stretcher: The range of temperature is between 23.4 to 34.6 °C – here in a fused visualization (a). Based on a lower threshold of 27 °C, parts of the subject are removed from the scene (b). Lowering this threshold further can reduce this issue. However, even with a lower threshold of 25 °C, the trousers are partially excluded (c).

light varies little because the trauma room is windowless and the ceiling lighting is homogeneous. Therefore the values for filtering can be set to a fixed range. The filter is applied to the points after the bounding box filter which still includes the medical stretcher. It is applied to differ between the patient \mathcal{P}^P and the stretcher \mathcal{P}^S . The filter for color is applied to the point cloud by

$$\begin{aligned} \mathcal{P}^* &= \mathcal{P}_H^* \cap \mathcal{P}_S^* \cap \mathcal{P}_V^* && \text{where} \\ \mathcal{P}_H^* &= \{\mathbf{p}_i \mid c_{h_{\min}} < \mathbf{p}_i \rightarrow c_h < c_{h_{\max}}\} \\ \mathcal{P}_S^* &= \{\mathbf{p}_i \mid c_{s_{\min}} < \mathbf{p}_i \rightarrow c_s < c_{s_{\max}}\} \\ \mathcal{P}_V^* &= \{\mathbf{p}_i \mid c_{v_{\min}} < \mathbf{p}_i \rightarrow c_v < c_{v_{\max}}\} \end{aligned}$$

Although the color filter is not necessary, it eases and speeds up the segmentation of the patient and the stretcher. The color filter works best if there is a distinct difference in color between the sheet and the person’s clothes. In a worst-case scenario, the patient would be covered with white clothes, so the color filtering would not have any benefit.

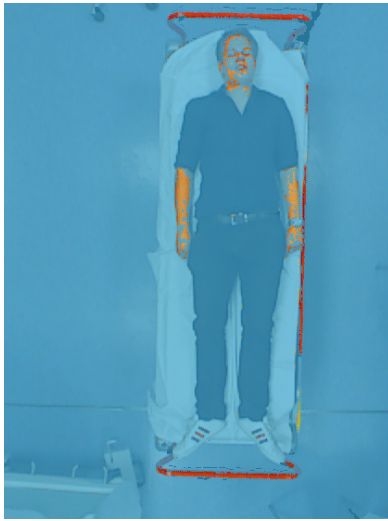
Figure 4.7 illustrates the result of the filter: In a first setting, the handlebars of the bed should be segmented (a). The color red is unique in this scene. The bars are found, but due to a wide range of filtering also the arms of the subject are considered to be part of the handlebars. This issue can be minimized by lowering the threshold of the filter, but then also parts of the handlebars might be excluded due to shadows and lighting conditions. In a second setting, the surface of the stretcher is segmented. Applying the filter with this setting to the whole scene, the surface of the stretcher is found, but is also confused due to the light color of the floor (b).

To improve the outcome of this filter, the bounding box filter is used previously, so the color filter only processes on data containing the subject and the stretcher (c). Here in this example, the surface of the stretcher is found reliable. However, also the white shoes are considered to be part of the stretcher. To estimate the parameters of the stretcher further steps are necessary, which are presented in section 4.6 on page 68.

4.5 Normal Filter

As described in the previous chapter, the noise in depth depends on the perspective of the sensor to the scene. It can increase, the bigger the viewing angle is. The computation of a point cloud's normals is described in the appendix on page IV in detail. Based on the calculated normals of a point cloud, a filter is used to remove points, having a certain angle towards a given reference axis. An example: Medical stretchers often can be adapted in height and also the angle of the backrest can be adjusted. For the localization of such a stretcher, a filter can be used, removing all points, having a higher angle towards the optical axis than the range of the stretcher. The angle α of a normal \mathbf{n} towards the optical axis $\mathbf{a} = (0 \ 0 \ 1)^T$ is defined by [8]

$$\alpha_i(\mathbf{n}) = \arccos\left(\frac{\mathbf{a} \cdot \mathbf{n}}{\|\mathbf{a}\| \cdot \|\mathbf{n}\|}\right) .$$



(a) Filter for red with settings HSV = (0, 255, 255)



(b) Filter for white with settings HSV = (0, 10, 180)



(c) Filter for white based the output of a previously applied bounding box filter with settings HSV = (0, 10, 180)

Figure 4.7: Segmentation based on a point's color: The left image shows the result of a color filter applied to find the red handlebars of the stretcher (a). Figure (b) and (c) illustrate the filtering for the white surface of the stretcher to improve upcoming processing.



(a) Threshold of 40° , normal smoothing size factor of 15 (b) Threshold of 60° , normal smoothing size factor of 15 (c) Threshold of 60° , normal smoothing size factor of 5

Figure 4.8: Applied normal filter with different configurations: The normal filter takes two arguments, first, the angle of the normal towards a given axis – here the z-axis of the sensor system – and second, a smoothing factor. A point is removed if the normal’s angle towards the axis is above the threshold (a). The smoothing factor is applied, to reduce noise in segmentation. The influence of the smoothing factor is presented in (b) and (c).

The normal filter removes points from the cloud by just a simple threshold and is applied by

$$\mathcal{P}^* = \{\mathbf{p}_i \mid \|\alpha_i(\mathbf{p}_i \rightarrow \mathbf{n})\| < \alpha_{\text{th}}\} \quad .$$

Figure 4.8 shows different settings to remove points from a scene based on their normals. Between the three images, the threshold for the angle varies, as well as the factor for previously applied smoothing. The calculation of the normals in this example is based on the method proposed by Holzer et al. [87] and Holz et al. [86]. In Figure 4.8a, the normal filter is configured with a threshold of 40° and a smoothing factor of 15. The z-axis of the depth sensor coordinate frame $\mathbf{a}_z = (0 \ 0 \ 1)^T$ is chosen to be the reference axis for angle calculation. Therefore, points with a normal having a bigger value than 40° to the perpendicular axis are removed from the scene. The output of the filter shows that especially jumping edges between the stretcher and the floor are removed. Also, parts of the body are removed, and a border between the subject and the surface of the stretcher is visible. Thin and round parts, like the red handlebars, are removed nearly completely. Increasing the threshold to 60° and keeping the smoothing factor to the same value leads to a result where fewer points are removed from the scene, as illustrated in Figure 4.8b. Also, the depth edges are removed widely, but more of the subject’s body is kept in the scene. Lowering the normal smoothing factor leads to two effects: The output of the filter provides a smaller edge, for example at the border between the subject and the stretcher’s surface. With the higher smoothing factor of 15, only parts of those edges are removed, while now, with a factor of 5, a closed border between the subject and the stretcher exists. Due to the

lower smoothing factor, sensor noise gains more impact on the result of the filtering. This can be seen in Figure 4.8c, where parts of the floor are removed because of the depth sensor's noise.

The normal filter is used for the upcoming RANSAC algorithm to estimate the inliers of the stretcher. Although this is not necessary, it increases the reliability for the outcome in plane localization and segmentation.

4.6 Plane Filter

For the clinical weight estimation, the subject is always on a medical stretcher. Therefore, the stretcher is always visible in the scene. With a simplified model, the surface of the bed can be modeled as a plane. The estimation works best if the stretcher is completely visible. However, with the subject lying on it, most parts of the stretcher are occluded and not visible to the sensors. Having a high gradient in color or intensity between the subject and the bed eases the segmentation; for example, someone wearing black clothes lying on the white surface. A worst-case scenario would be a segmentation based on an RGB camera when the subject is wearing white clothes.

To segment the patient from the stretcher, the RANSAC algorithm [64] is applied to the previously defined data set after color and normal filtering.

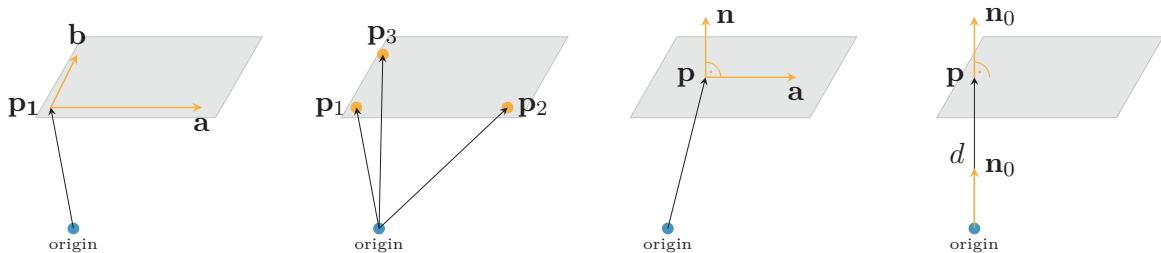
Different equations can describe a plane: First, it can be described by a point and two directional non-collinear vectors \mathbf{a} and \mathbf{b} with

$$\mathbf{p}(a, b) = \mathbf{p}_1 + a \cdot \mathbf{a} + b \cdot \mathbf{b} \quad \text{where } a, b \in \mathbb{R} \text{ and } \mathbf{a} \neq \mathbf{0} \text{ and } \mathbf{b} \neq \mathbf{0} \quad .$$

The parameters a and b are used for scaling, while the two directional vectors span an affine coordinate system. Figure 4.9a shows the modeling of a plane based on two non-collinear vectors.

Second, the plane can be described by three arbitrary points \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 . An additional constraint is that these points must not lie on a straight line. Therefore $(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1) \neq \mathbf{0}$ has to be fulfilled and the plane is defined by

$$\mathbf{p}(a, b) = \mathbf{p}_1 + a(\mathbf{p}_2 - \mathbf{p}_1) + b(\mathbf{p}_3 - \mathbf{p}_1) \quad \text{where } a, b \in \mathbb{R} \quad . \quad (4.2)$$



(a) Plane defined by two vectors (b) A Plane defined by three points (c) Plane defined by normal and point (d) Hesse normal form

Figure 4.9: Different definitions of a plane.

Figure 4.9b illustrates this plane model. Furthermore, a plane can be described by a vector \mathbf{p}_1 and a normal vector \mathbf{n} which is perpendicular to the plane. The equation for a plane is then defined by

$$\mathbf{n} \cdot (\mathbf{a} - \mathbf{p}) = 0 \quad \text{or} \quad \mathbf{n} \cdot \mathbf{a} = \mathbf{n} \cdot \mathbf{p} \quad .$$

Figure 4.9c demonstrates the description of the plane based on this normal form. With the help of this equation, the parameter form of a plane can be defined. With the definition of $\mathbf{a} = (a \ b \ c)^T$, the normal vector \mathbf{n} , and a point on the plane $\mathbf{p} = (x \ y \ z)^T$, the equation (4.2) can be solved to

$$ax + by + c = z \quad . \quad (4.3)$$

A special variant of the previously presented definition based on a normal is the Hesse normal form: Here the perpendicular normal vector \mathbf{n}_0 aligns with a line from the origin. The normal vector \mathbf{n}_0 is normalized to a length of 1. Compared to the previously presented equation, the point \mathbf{p} can be neglect and is replaced just by the distance d so [8]

$$\mathbf{p} \cdot \mathbf{n}_0 = d \quad .$$

The Hesse normal form is efficient to calculate a distance of an arbitrary point \mathbf{p} towards the plane. To calculate the distance of an arbitrary point \mathbf{p}_1 to the plane $d(\mathbf{p}_1)$, the position vector of \mathbf{p}_1 is taken and forwarded to

$$d(\mathbf{p}_1) = \mathbf{p}_1 \cdot \mathbf{n}_0 - d \quad . \quad (4.4)$$

If the distance is zero $d(\mathbf{p}_1) = 0$, the point is part of the plane. Often, the form is written as $\mathbf{h} = (h_x \ h_y \ h_z \ d)$ or $\mathbf{h} = (\mathbf{n}_0 \ d)$ [8].

With the assumption that the available point cloud provides a plane, the least square algorithm is suitable to estimate the parameters, even if the data is noisy. With the help of the parameter form of the plane equation (eq. (4.3)), an error function is defined by [59]

$$e(a, b, c) = \sum_{i=1}^n ((ax_i + by_i + c) - z_i)^2 \quad ,$$

where n is the total number of plane inliers and the sum of the squared errors between the plane parameters and z_i is minimized. Here the error is measured only along the z-axis. The minimum of this nonnegative function can be found by the calculation of the gradient of the error function ∇e by

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \nabla e = 2 \sum_{i=1}^n ((ax_i + by_i + c) - z_i) \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \quad ,$$

and finally, the parameters can be solved by

$$\begin{pmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n y_i & \sum_{i=1}^n 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n x_i z_i \\ \sum_{i=1}^n y_i z_i \\ \sum_{i=1}^n z_i \end{pmatrix} \quad . \quad (4.5)$$

While this solution can lead to an ill-conditioned linear system, an alternative solution can be applied by subtracting the point cloud by the centroid $\mathbf{p} = (\bar{x} \ \bar{y} \ \bar{z})^T$ [59]. The fitted plane with $z - \bar{z} = a(x - \bar{x}) + b(y - \bar{y})$ is solved by

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n (x_i - \bar{x})^2 & \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \\ \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^n (y_i - \bar{y})^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^n (x_i - \bar{x})(z_i - \bar{z}) \\ \sum_{i=1}^n (y_i - \bar{y})(z_i - \bar{z}) \end{pmatrix} .$$

The least square estimation can only provide a reliable output if the data set does not contain outliers. Therefore, outliers have to be detected and excluded from the estimation. The RANSAC algorithm has the advantage to do a robust estimation of inliers for a given model, even a high uncertainty based on outliers is present. On the other side, RANSAC is not optimal and takes a fixed amount of iterations – and therefore time – to do the computation. The outcome of the estimation is influenced by the pre-processing and the number of iterations. It is essential to set the number of iterations to a certain level to ensure a good result for the algorithm. Setting the number of iterations too high is a waste in computational costs. Therefore, the number of iterations k is set by the relation between outliers and inliers. The probability to choose an inlier is given by w , while this is also the expected relation between inliers and the total size of the available data. In addition, the probability that a subset of selected points does only contain inliers is calculated by w^n , where n is the number of points needed to apply the model. For the here searched plane model, three points are necessary $n = 3$. The chances to pick a subset containing outliers are given by $1 - w^n$. For k iterations, the probability of choosing only subsets with outliers is therefore defined by

$$p(k \text{ subsets with outliers}) = (1 - w^n)^k \quad ,$$

which is also the probability, that the RANSAC algorithm fails to pick a subset with inliers in k iterations. Following from this, the probability that the algorithm finds a valid set in k iterations is defined by

$$p(\text{success}) = 1 - (1 - w^n)^k \quad .$$

The equation can be solved for k by the logarithm so finally the number of iterations k is calculated by

$$k = \frac{\log(1 - p(\text{success}))}{\log(1 - w^n)} \quad .$$

Typical the probability is set to $p(\text{success}) \geq 0.99$ to ensure a good and reliable outcome of RANSAC [64]. To ensure a robust outcome of the algorithm, the points forwarded to the RANSAC algorithm are filtered in advance. Only points included by the color filter – the surface of the stretcher in the medical scenario is always covered with a white sheet – and excluded by the thermal filter are forwarded as input.

Based on a Hessian plane model, the algorithm returns after several iterations a subset in data, fitting best to the model. Algorithm 2 provides the procedure of the RANSAC algorithm.

Figure 4.10 illustrates the principle of the RANSAC algorithm in a simplified 2D version, to estimate the inliers of a line. Due to sensor noise in a plane, a distance boundary for the inliers is necessary to find a reliable consensus set. The distance boundary is parametrized with a threshold d_{th} and should be adapted to the sensor's noise σ . If the threshold is set to low, a

Algorithm 2: RANSAC algorithm for the estimation of plane inliers.

1. Select a sufficient number of points $n = 3$ randomly from the scene to apply the model. \mathcal{P}_R is called the random set. The selected points $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3 \in \mathcal{P}_R$ must not be collinear in case of a plane model.
 2. Compute the plane model based on the selected points $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ based on equation (4.2).
 3. Calculate the distance of each point $d(\mathbf{p}, \mathbf{h})$ in set \mathcal{P} towards the calculated model \mathbf{h} based on equation (4.4).
 4. Count the number of points $\mathcal{P}_c = \{\mathbf{p}_i \mid d(\mathbf{p}_i, \mathbf{h}) < d_{\text{th}} \text{ where } \mathbf{p}_i \in \mathcal{P}\}$ those distance to the model is below a set threshold. The threshold's value should be close to the expected noise of the plane's inlier σ . This is the so-called consensus set \mathcal{P}_c . In case the current consensus set is bigger than an old set $|\mathcal{P}_c| \geq |\mathcal{P}_{c_{\text{old}}}|$, save the size of the set and the corresponding plane model. If the number of iteration is not reached, continue with step one.
-

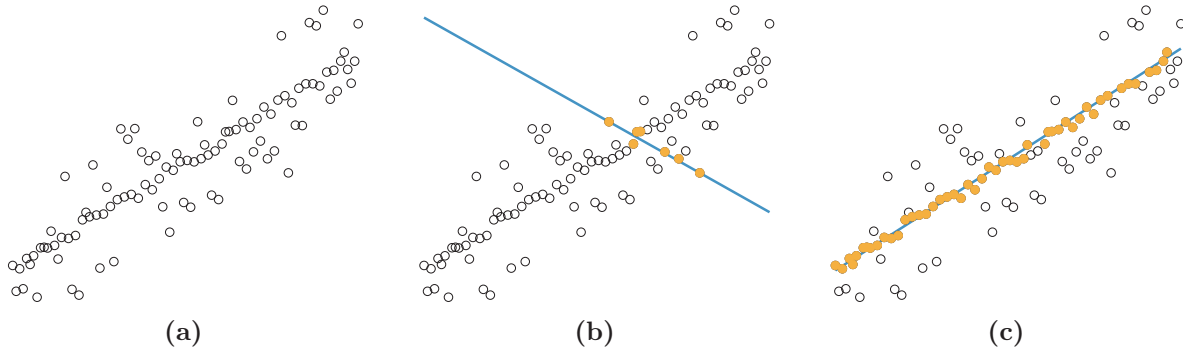


Figure 4.10: Progress for RANSAC with applied line model: The data set consists of a linear function applied with Gaussian noise on the y-axis (a). An arbitrary iteration will find a similar solution as presented in (b) while the inliers of the consensus set are colored. The solution with the most inliers for the consensus set is presented after the maximum number of iterations (c). The final plane is estimated by applying the least-square optimization.

low amount of inliers is found by RANSAC, while a high threshold could lead to the selection of outliers for the consensus set.

Figure 4.11 illustrates the result of the applied RANSAC algorithm to estimate the pose of the stretcher, modeled as a plane. For this experiment, the scene is already filtered with the bounding box filters and additionally with the normal filter. Furthermore, only white points filtered by the color filter are forwarded to the RANSAC algorithm. With this filtered data, the algorithm is started with 200 iterations. Due to the depth sensor's noise, a distance threshold of 5 cm is configured. This ensures that most parts of the stretcher are recognized to be part of the plane model. However, not all white points of the stretcher are marked as inliers and due to the sensors noise some points are removed. Points below the estimated plane are removed from

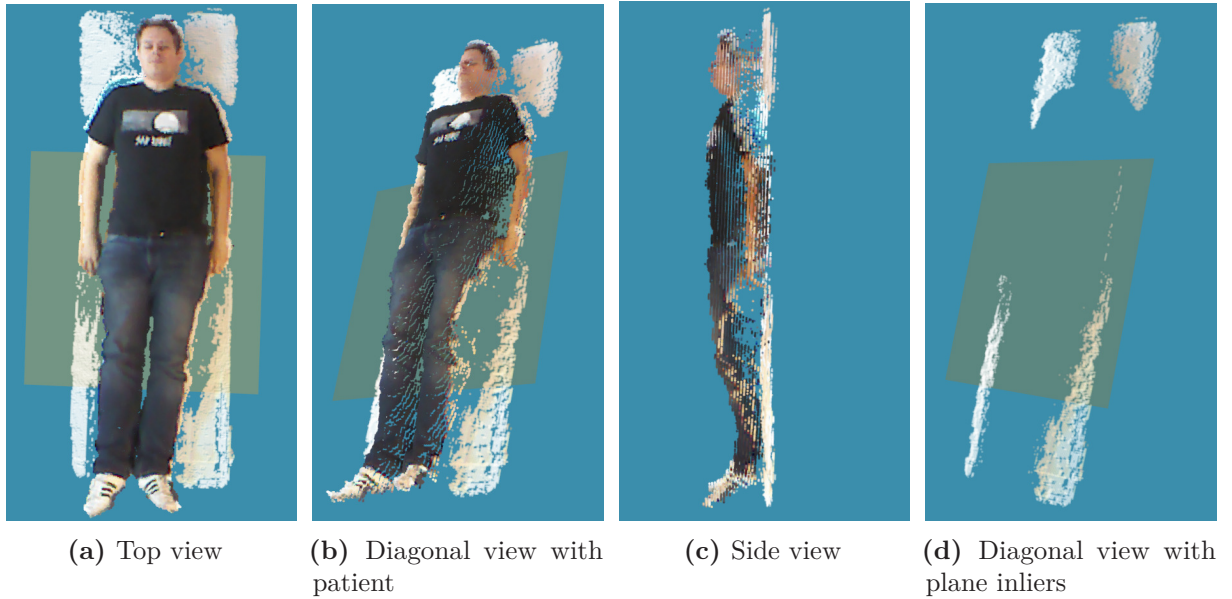


Figure 4.11: Applied RANSAC algorithm to estimate the inliers of the stretchers surface: The first three images (a-c) show the forwarded point cloud. The cloud contains the subjects, as well as the surface of the stretcher due to previously applied filters. The result of RANSAC is shown in (d), providing the inliers of the estimated plane.

the scene. The Figures 4.11a to 4.11d illustrate the scene after the plane estimation, including the patient and the plane’s inliers. Figure 4.11d shows the plane’s inliers as well as the orange square for the plane model. With the correct inliers of the plane, the final plane is calculated based on the least-square algorithm, minimizing the distances of all inliers towards the plane model [179, 65].

In an additional step, the found plane is tested for plausibility: Although the stretchers in trauma rooms might vary slightly in shape, length or width, the plane must have a minimum area spanning behind the patient. If several planes are found by RANSAC, the planes are sorted by their area, starting with the biggest one. To test the area of a plane a rectangular shape is defined. The four points defining the corners $\mathbf{p}_i = (x_i \ y_i \ 0)$ with $i = 0..3$ of the rectangular are taken to calculate the area a_r via Gauss’s area formula [8] by

$$a_r = 0.5|(y_0 - y_2) \cdot (x_3 - x_1) + (y_1 - y_3) \cdot (x_0 - x_2)| \quad .$$

The resulting reference plane is used to segment the patient from the stretcher, while it is also utilized for volumetric reconstruction of the patient, as shown by Pfitzner et al. [163].

Essentially the results in weight estimation of lying people are strongly correlated to the result of the reference plane estimation. Minor errors in the parameters acquired by RANSAC of the reference plane can lead to a big absolute error in weight estimation, due to a wrong estimation of the volume. Especially parts of the stretcher on top and bottom of the reclining area should be visible to the sensor to prevent errors in plane localization.

4.7 Thermal Plane Distance Filter

It might occur that parts of the human body are not extracted correctly. As seen in Figure 4.6, parts of the subject can be excluded, because thick and loose clothes provide a temperature close to the ambient temperature. For example, if a patient is brought to the emergency room in winter, the shoes will have a lower temperature than the ambient temperature. However, for the upcoming feature extraction, it is essential that no parts of the subject's body are excluded due to filtering. The here presented thermal plane distance filter extends the trivial thermal filter by concerning the pose of the points in relation to the back of the stretcher.

First, the thermal plane distance filter depends on the previously estimated plane \mathbf{h} . For all points above the plane the Euclidean distance between the plane $\mathbf{h} = (h_x \ h_y \ h_z \ h_d)$ and an arbitrary point $\mathbf{p}_i = (x \ y \ z)^T$ is calculated by

$$d(\mathbf{p}_i, \mathbf{h}) = \frac{x \cdot h_x + y \cdot h_y + z \cdot h_z + h_d}{\|\mathbf{p}_i\|} .$$

The formula provides a signed result. The value is positive for a point on the same side of the plane as the normal vector is pointing [8].

Second, the filter depends on two thresholds: Based on the first threshold, points are removed from the scene, when its temperature is below, independent of the distance towards the reference plane. This part has now the same behavior as the previously explained thermal filter. Now, a second threshold is introduced, which is set above the first threshold $T_{th1} < T_{th2}$. If the temperature between the first and the second threshold, the removal of the point depends on the distance to the plane. A point being close to the plane but below the thermal threshold is

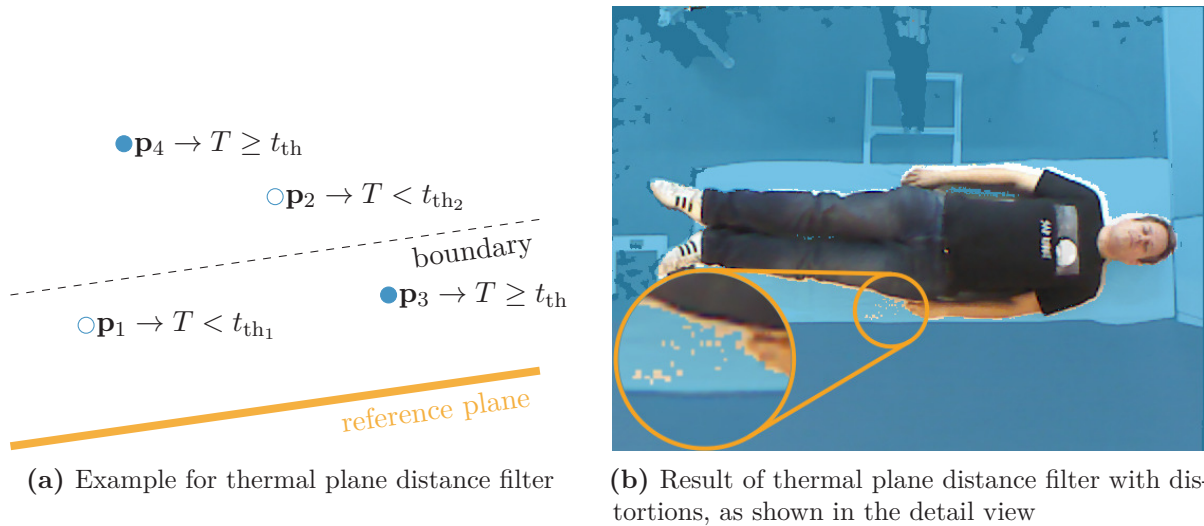


Figure 4.12: Removing points from the scene based on the thermal plane distance filter: The schematic illustrates the principle of the thermal plane distance filter (a). The thermal plane distance filter provides a better segmentation than just the thermal filter. Now, also the shoes and the complete trousers are kept after filtering, compared to the thermal filter. An outlier filter or morphological operations remove the distortions seen in the detail view.

removed; if the point's temperature is above, the point is kept. The output of the thermal plane distance filter, \mathcal{P}^* , is described by

$$\mathcal{P}^* = \{\mathbf{p}_i \mid d(\mathbf{p}_i, \mathbf{h}) > d_{\text{th}}\} \quad .$$

Figure 4.12a illustrates the process of filtering: The schematic consists of four points, having different conditions. Two points are below the boundary \mathbf{p}_1 and \mathbf{p}_3 . Furthermore, one point in the schematic has a temperature below the first threshold \mathbf{p}_1 , while another point has a temperature between the first and the second threshold \mathbf{p}_2 . The point \mathbf{p}_2 is removed, because its temperature is below the first and lower threshold, independent of the distance to the plane. Furthermore, the point \mathbf{p}_1 is filtered because it is close to the reference plane and its temperature is within the defined range between the first and the second threshold. The point \mathbf{p}_3 and \mathbf{p}_4 remain in the scene because they are above the boundary or have a temperature above the upper threshold.

Figure 4.12b shows the result of the filter: In comparison to a simple single threshold, now the subject is segmented correctly from the scene. The lower part of the jeans, which was excluded by the simple thermal filter, remains now in the scene. Due to noise in depth, parts of the bed can be also added in single points to the output of the filter. This issue is fixed by erosion, which is described in detail in an upcoming section on page 77.

4.8 Background Subtraction

One obvious approach to segment an object from a static scene is the background subtraction: A reference frame from the sensor system is needed, in which no person is visible. This approach is only suitable if the sensors are mounted rigidly towards the scene.

Having an RGB-D sensor, the background subtraction can be done either via the color channel or the depth data. In both cases, the values from the current sensor frame are compared pixel-wise towards the previously recorded background frame. Therefore the background segmentation based on the color looks for similar values, transforming the color of a pixel in HSV color space giving a certain threshold. The background subtraction based on depth data is achieved by comparing the distance pixel-wise between the reference frame \mathcal{P}^r and the current scene \mathcal{P} by

$$\mathcal{P}^* = \{\mathbf{p}_i \mid \|\mathbf{p}_i - \mathbf{p}_i^r\| > d_{\text{th}}\} \quad \text{where} \quad \mathbf{p}_i \in \mathcal{P} \quad \text{and} \quad \mathbf{p}_i^r \in \mathcal{P}^r \quad ,$$

where \mathbf{p}_i and \mathbf{p}_i^r are corresponding points. Due to the sensor's noise in depth, which increases over distance [218], the threshold can be adapted with respect to the distance in the reference frame. For near objects, the threshold can be low, while an increasing distance in the reference frame should provide a higher threshold. Having a sensor like the Kinect One, which has an increasing sensor noise of around 1 centimeter per meter. Therefore, an adaptive threshold is calculated based on the sensor's noise model, described by the variance σ^2 of the measured distance of an arbitrary point to the sensor's origin $\|\mathbf{p}_i\|$ by

$$d_{\text{th}_i}(\mathbf{p}_i, d_{\text{th}_{\min}}) = \sigma^2(\|\mathbf{p}_i\|) + d_{\text{th}_{\min}} \quad ,$$

while a minimum threshold of $d_{\text{th}_{\min}}$ should always be kept.

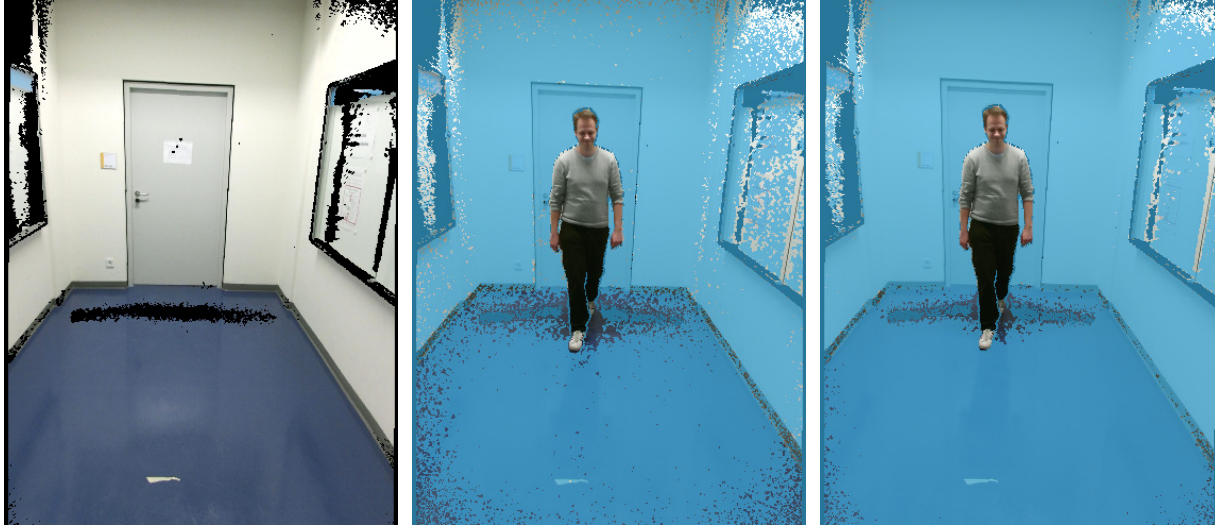
For the approach in the clinical environment, this segmentation approach is not the first choice: Most objects in the FOV are movable. Nevertheless, this approach could be replaced with the previously presented distance or bounding box filter to reduce the amount of data and to speed up processing. The filtering based on a background subtraction can only be applied if the sensor stays in a fixed pose towards the static environment during experiments. Otherwise, a new reference frame has to be recorded.

Figure 4.13 shows different settings for the background subtraction: In the first image, the scene for subtraction is taken. The here shown scene is used for the body weight estimation of people walking towards the camera, which is presented in more detail in the section for experiments. The scene is recorded with a Kinect One camera, while the maximum measured distance is around 4.5 meters. Because this is also the maximum distance which can be measured by the sensor, parts of the point cloud contain holes. Furthermore, this is reinforced because of the steep angle towards the slightly reflecting floor. The second image, Figure 4.13b, shows the result of the applied background subtraction with a distance threshold of 1 cm. Because this threshold value is in the same order as the sensors noise in depth, the noise in the filtering is also visible. Particularly on the ground and the walls, small fragments are visible. Due to the steep viewing angle, the depth noise is here increased. To fix this issue, the threshold is increased to 4 cm: Now the impact of the sensor's noise is reduced, in particular on the ground. Small and single areas can be removed with morphological operations. If the distortions on the floor cannot be removed, an additional plane filter can be applied to remove the ground plane, while the accuracy in filtering can be improved if the ground plane has a particular color – like the floor in this scenario is blue. Otherwise, parts of the shoes could be missing in the filtered scene, depending on the size of the applied threshold towards the estimated ground plane.

4.9 Segmentation based on Edges

Edges are clear features to separate sets of points or pixels based on a gradient. Most often edges are ruptured and cannot separate the sets with an enclosed border. Nevertheless, an edge-based filter can improve the outcome in segmentation. If the color filter cannot be applied due to uncertain conditions of the environment, e.g., there exist several different colored sheets to be placed on the stretchers in the trauma room, the color filter is not the best approach for the segmentation. Although the filter could be adapted to several cases, errors in segmentation could occur due to such an issue. An edge filter can work on several streams: Taking the color stream from one of the sensors, the gradient between dark clothes and the light sheet on the stretcher are reliable for segmentation. Similar, the filter could work on the data from the thermal camera, finding the edge between warm and cold objects.

Several approaches for edge detection exist: First, an edge could be found based on the Canny edge detection algorithm [38]. Here, a maximum suppression is needed to ensure a thickness of the edge of only one pixel. Different filter kernels can be applied. The most common kernels are the Roberts, Prewitt or Sobel kernel [190]. Second, edges can be found based on convolution with the Sobel operator and a Difference of Gaussian (DOG) [129]. To get the DOG, the original image is subtracted from the blurred original image.



(a) Scene for background subtraction (b) Applied Background subtraction with 1 cm threshold (c) Applied Background subtraction with 4 cm threshold

Figure 4.13: Removing the background from a scene: First, a reference shot is saved (a). This frame should only contain the background of the scene. Due to the sensor’s noise, distortions can appear in the result of the filter (b). Therefore, an additional threshold is added, so the output of the filter has fewer distortions (c).

The size of the Gaussian kernel is defined by $(2k + 1) \times (2k + 1)$ where $k \in \mathbb{N}_0^+$ and can therefore only have a size with an uneven dimension. The size of the Gaussian kernel affects the result of the edge detection: Applying a larger kernel will lower the detector’s sensitivity to noise. In addition, the error in localization increases with the size of the kernel. The Sobel operator differs for the x- and y-direction in an image. The edges for the gradient in x-direction \mathbf{I}_{G_x} and the y-direction \mathbf{I}_{G_y} are applied to an image \mathbf{I} by

$$\mathbf{I}_{G_x} = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} * \mathbf{I} \quad \mathbf{I}_{G_y} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * \mathbf{I} \quad .$$

The edge detection, as proposed by Canny [38], is shown by Algorithm 3 Figure 4.14 illustrates the result of applying Canny edge detection to the color, depth and thermal stream. While the stretcher, including the subject, can be found clearly in the edges from the depth image, the subject is visible in the edges from the thermal frame. Moreover, the edges can be fused: Some edges might appear in multiple sensor streams. A dark object in front of a white wall will result in an edge in the color stream, as well as in the depth stream. Although the edge detection in the depth image provides an excellent feature to detect the stretcher with the patient, the bounding box filter with the marked rectangle on the floor provides more reliable results. Furthermore, it is more transparent for the physicians, that the estimation can provide a result with a high error if the stretcher is not in the marked area, compared to a false segmentation based on one of the edges.

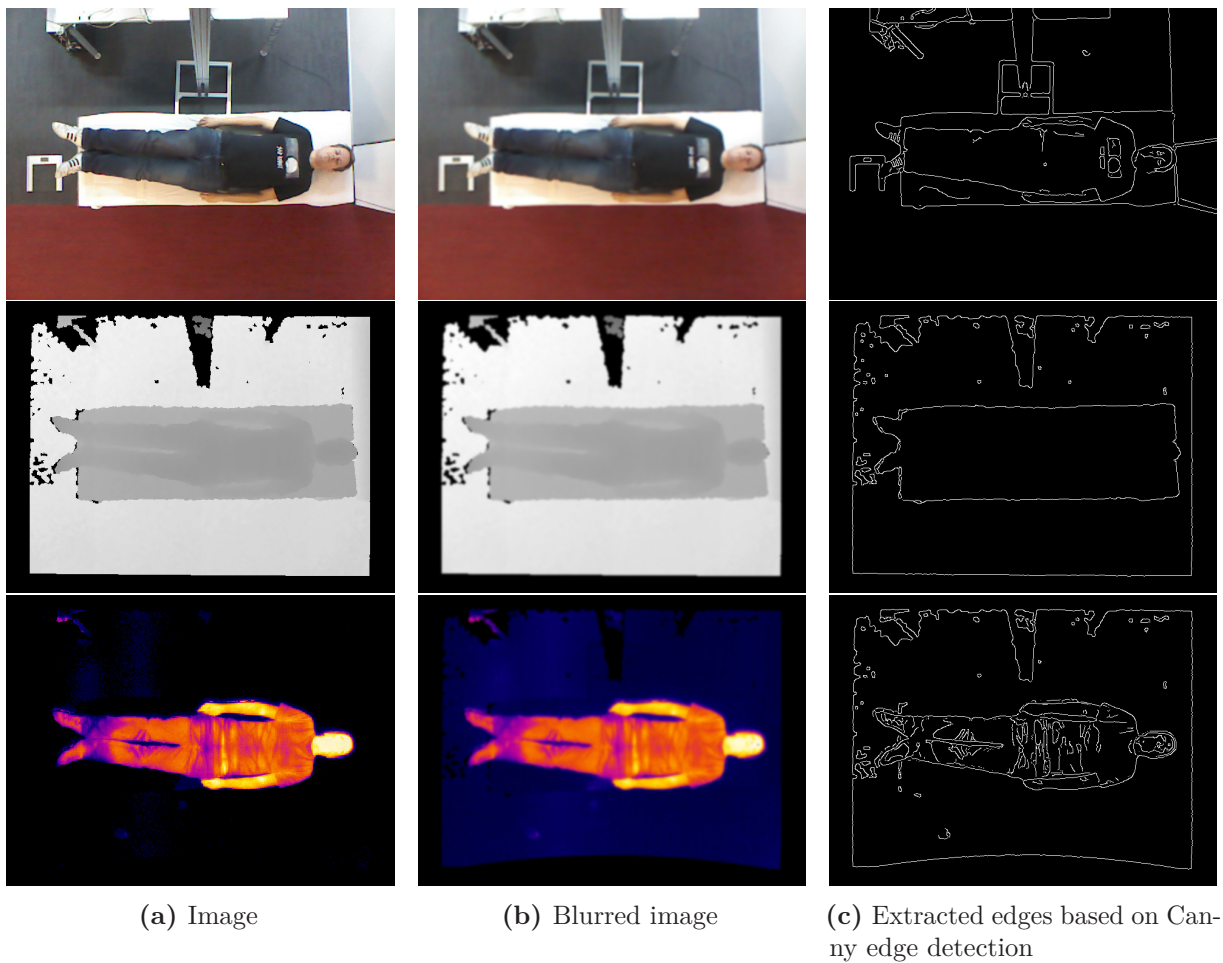


Figure 4.14: Edges extracted from different sensor streams: The raw image from each frame is first blurred and finally subtracted from the original image. The edges here are extracted by the Canny edge detection algorithm.

4.10 Removing Distortions in the Segmentation

Based on the present segmentation of the subject from the environment, minor distortions can appear. Single and small groups of points can be in the resulting point cloud. Such a distortion is shown previously in Figure 4.12b after applying the thermal plane distance filter, where parts of the stretcher are in the resulting point cloud, due to noise. These small errors in segmentation are solved with state-of-the-art algorithms. Morphological operations like erosion and dilation work on 2D images containing only Boolean data. Therefore, a Boolean mask is extracted from the point cloud's indices. Based on the extracted binary image, a kernel operation is applied, either expanding the bit mask (dilation) or shrinking the bit mask (erosion). Erosion and dilation help to fill gaps and holes in the bit mask caused by incorrect temperature measurements and reflections to the environment, due to reflective clothing. Therefore a cross-shaped kernel of the size of 3×3 is applied to the bit mask for all morphological operations. Other kernel shapes

Algorithm 3: Canny Edge algorithm [38].

1. Remove noise in the image by applying a Gaussian filter to smooth the image.
 2. Locate the intensity gradient in the image based on the sum of the gradient for x and y by $\|\mathbf{I}_G\| = \sqrt{\mathbf{I}_{G_x}^2 + \mathbf{I}_{G_y}^2}$.
 3. Apply a non-maximum suppression to thin the edges to a minimum width. All non-maximum values along the gradient are suppressed and set to 0.
 4. Based on two thresholds, detect strong edges E_s and weak edges E_w : If the intensity of the gradient is above the threshold for strong edges $E_{s_{th}}$, classify this edge to be strong $E_s = \{E \mid E > E_{s_{th}}\}$. If the edge is below the threshold for strong edges and above the threshold for weak edges $e_{w_{th}}$, consider the edge to be weak $E_w = \{E \mid E_{w_{th}} < E < E_{s_{th}}\}$. Edges below the weak threshold are removed.
 5. Suppress edges below the lower threshold of the hysteresis. Edges about the upper threshold E_s of the hysteresis are taken as valid edges. Weaker edges E_w below the upper threshold are excluded if they are not connected to a strong edge. Weak edges connected to a strong edge are represented by E'_w . Therefore the resulting edges are defined by $E = E_s \cap E'_w$.
-

are possible, for example, a circle or a square. Furthermore, the size of the kernel can be scaled, while the dimension of the kernel matrix is always an uneven value for the number of columns and rows.

Figure 4.15 shows the principle of the morphological operations. The binary mask is illustrated as a blue overlay over the scene. The first step (Figure 4.15b) increases the binary mask. In contrast to that, the dilate operation decreases the size of the binary mask, see Figure 4.15c.

First, to remove the noise of the filtered results, erosion is applied to the scene. This removes the border between valid and already filtered points from the cloud. Small areas and points with few valid neighbors are removed from the scene in this step. Second, dilation is applied to this shrunk bit mask, increasing the size of the valid items based on the convolution with the selected kernel. The now received result is for big areas similar, compared to the initial version of the point cloud. However, the scene is now cleared from small noises in the bit mask and the border between valid and removed points is smoothed. The combination of erosion and dilation is common and is also called closing. Morphological operations have low computational cost and are fast to apply [205].

Another way to remove outliers from the scene is the statistical outlier filter. During the process of segmentation and filtering, it can occur that single points might stay in the scene due to sensor noise. To remove those points, a statistical outlier removal model is applied. The algorithm needs a point cloud \mathcal{P} , while for each point \mathbf{p}_i the k Nearest Neighbors (kNN) \mathcal{N}_k have to be known. This structure is called k -nearest neighbor tree. Two different types of the

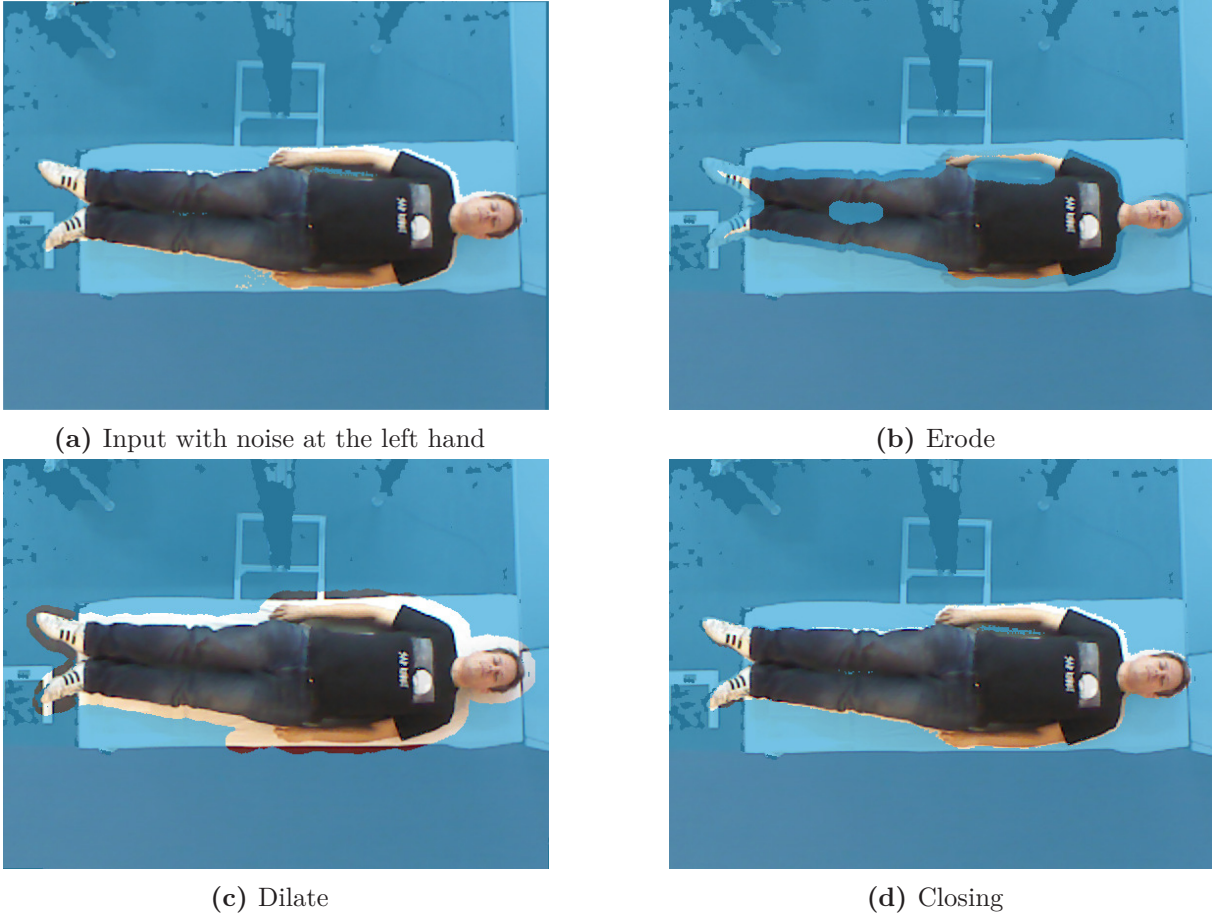
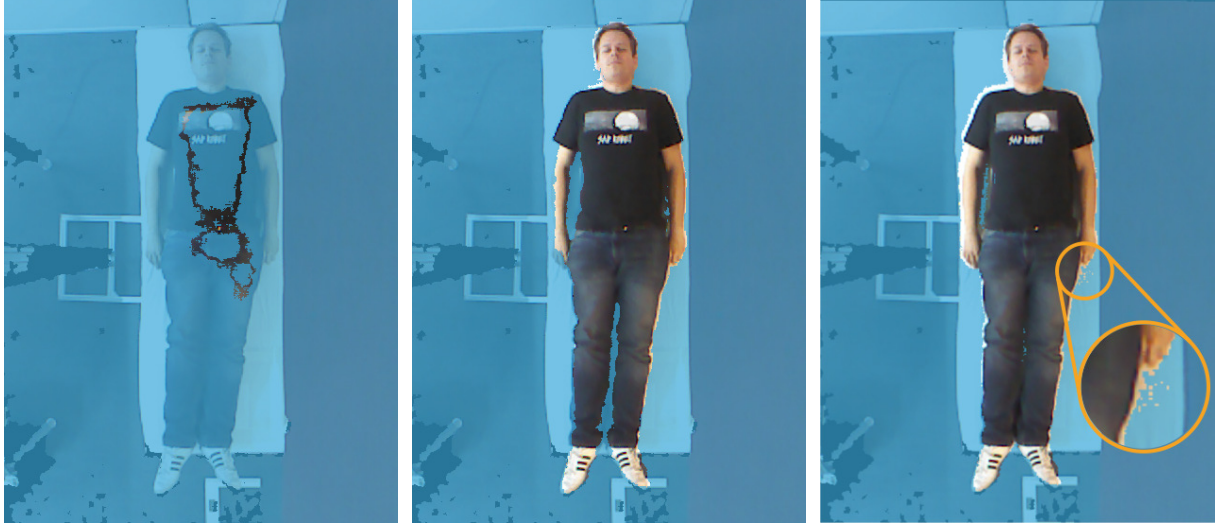


Figure 4.15: Morphological operations to remove distortions from segmentation: For the here presented experiment, the scene after applying the thermal distance plane filter is used (a) which contains noise in the segmentation (left hand). Applying erosion with a cross-shaped kernel shrinks the filter masked (b), while dilation increases the filter mask (c). Combining erosion and dilation removes the noises in the forwarded scene (d). This procedure is called closing.

calculation of the nearest neighbors are presented in the appendix A.2 on page III. For every point $\mathbf{p}_i \in \mathcal{P}$, the average distance to the closest k neighbors \bar{d}_i is calculated by

$$\bar{d}(\mathbf{p}_i) = \frac{1}{k} \sum_{j=1}^k \|\mathbf{p}_i - \mathbf{p}_k\| \quad \text{where } \mathbf{p}_k \in \mathcal{N}(\mathbf{p}_i) \quad ,$$

while the calculation of the k neighbors is expensive in calculation and should be kept to a minimum. Filtering the points just by a fixed threshold would lead to a misbehavior: Because of point clouds from a projective sensor or a LIDAR, the point cloud is more sparse in the distance. Therefore, points further from the sensor's origin would likely be filtered out, although they are no outliers. It is more effective to filter out points based on statistical distribution. The



(a) Cluster tolerance of 1.0 cm (b) Cluster tolerance of 1.5 cm (c) Cluster tolerance of 2.5 cm

Figure 4.16: Filtering the scene with a Euclidean cluster: With a cluster tolerance of 1 cm the clustering fails and provides only a minimal segment of the subject (a). The reason for this result is that the cluster tolerance with this value is in the same range as the sensor’s noise in depth. Increasing the cluster tolerance to 1.5 cm leads to a sufficient removal of the previously detected distortions at the left hand (b). Increasing the tolerance further leads to hardly any noise reduction – visible in the detail view (c).

standard deviation of the previously calculated distances $\sigma_d(\mathbf{p})$ based on the average distance to the closest k -neighbors \bar{d} is calculated by the help of equation 4.10

$$\sigma_d(\mathbf{p}_i) = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - \bar{d})^2} \quad ,$$

where \bar{d} is the average distance to the kNN over the complete point cloud. Together with the average standard deviation of all distances towards their nearest neighbors in the cloud $\sigma_d(\mathcal{P})$ by

$$\sigma_d(\mathcal{P}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (\sigma_d(\mathbf{p}_i) - \bar{\sigma}_d)^2} \quad , \text{ where } \mathbf{p}_i \in \mathcal{P}$$

and a threshold for the standard deviation in distance to the nearest neighbors d_{th} , the filter removes outliers by a given threshold by

$$\mathcal{P}^* = \{\mathbf{p}_i \mid \sigma_d(\mathbf{p}_i) + d_{\text{th}} < \sigma_d(\mathcal{P})\} \quad .$$

The presented filter is described in detail by Rusu [179]. Especially the RANSAC algorithm benefits from removing potential outliers, reducing the number of trials; or improving the probability of finding a good solution with a fixed number of trials. Also, the upcoming feature extraction gains robustness due to the removal of outliers.

Another possibility to remove outliers is based on a Euclidean clustering method. Here the given point cloud \mathcal{P} is split into several clusters, by comparing the distance of a given point to its neighbored point \mathcal{P}_i^k . The algorithm works with a fixed threshold for the distance d_{th} to find the nearest neighbors. The algorithm is applied as presented by Algorithm 4. The implementation

Algorithm 4: Algorithm for Euclidean clustering [180].

1. A Kd-tree is generated [19], based on the forwarded point cloud \mathcal{P} .
 2. All points from the cloud are assigned to a queue \mathcal{P}_Q . Furthermore, an empty list for the clusters \mathcal{P}^c is created.
 3. For every point, $\mathbf{p}_i \in \mathcal{P}$ is added to the queue \mathcal{P}_Q .
 4. For every point $\mathbf{p}_i \in \mathcal{P}_Q$, do the following steps:
 - Neighbors of an arbitrary point \mathbf{p}_i are searched within a given radius $r < d_{\text{th}}$.
 - The algorithm checks every neighbor $\mathbf{p}_i^k \in \mathcal{P}_i^k$ if the point has already been processed; if not, the point is added to the queue \mathcal{P}_Q .
 - In case all points in the set \mathcal{P}_Q have been processed, the queue \mathcal{P}_Q is added to the list of clusters \mathcal{P}^c .
 5. The algorithm terminates if all points \mathbf{p}_i from the cloud \mathcal{P} have been processed and are assigned of most and at least one element of the clusters \mathcal{P}_i^c .
-

of the Euclidean clustering is taken from the PCL [179]. The filter is applied to the scene with the walking subject, providing a better segmentation compared to the morphological filters. Figure 4.16 illustrates the filter with an increasing cluster tolerance.

4.11 Distinguish between several People in the Scene

During treatment, it is common that several persons are in the FOV of the camera system, e.g., the patient, treating physicians, nursing staff or members of the family. To ensure that a body weight estimation is not affected by the presence of those people, the system has to distinguish between the patient lying on the stretcher and other persons. The previously explained bounding box reduces the whole scene to a Range of Interest (ROI) and therefore helps to minimize the influence of people being in the FOV but outside of the set bounding box. Thus, a problem can occur if a person is close to the patient or is bending over the patient, e.g., for better treatment.

If a person is close to the patient, but clearly separable by an edge in the color or depth frame, the discrimination can be achieved via the contour separation: Taking the point cloud after the applied thermal filter as input, the corresponding binary mask is most suitable for separation. In the presence of only a single person, there should only be a single contour; the patient lying on the stretcher. Therefore, if several contours occur, they are ordered by their size. The contour with the biggest area is most likely to be the patient. Nevertheless, this approach might not work sufficiently. If a person is bending over the patient, segmentation could be applied based on



(a) Scene for filtering

(b) Filtered scene

Figure 4.17: Filtering people close to the patient: Two subjects appear in the scene, one lying on the stretcher and one close to the stretcher (a). To remove the subject close to the stretcher, the size of the contours is compared, while only the biggest contour is kept (b).

Euclidean clustering: Here, subsets form the point cloud are generated if there is a gap with a defined size is separating them. To minimize the impact of people standing close to the patient, the staff using the body weight estimation is instructed, that taking a shot with the sensors is best when no one is close to the patient. Figure 4.17 shows a scene where a second subject is standing close to the subject on the stretcher. The person close to the stretcher is removed, because his contour in the binary mask is smaller, compared to the person on the stretcher.

4.12 Timing Analysis for Segmentation

The previously presented approaches for segmentation differ in their complexity and also in its computational cost. Therefore, the computational time is compared to each other. The experiments were performed on a Dell M4800 mobile computer, containing an Intel Core i7-4900MQ CPU with a clock speed of 2.8 GHz, 4 physical cores, and 8 threads. The timing includes only the filter routine, without initialization or configuration of the implemented filter. Table 4.1 shows the result of the timing analysis of the different segmentation approaches. While most filters are applied to the complete point cloud with its full-size, some approaches are applied to already reduced point clouds, for example, the minimum bounding box or the plane filters.

The bounding box filter is applied to the full-size point cloud, reducing in the medical scenario the scene to about one-third of its original size. With 24 ms the filter takes longer, compared to the other approaches. The minimum bounding box filter is applied to this already reduced point cloud. The runtime of this filter correlates with the point clouds size linearly. The color filter takes the longest in this example: It includes the conversion of the color space from RGB to HSV color space. However, it could be optimized and commonly the color filter is not applied to the full scene, but to an already reduced point cloud, e.g., by a bounding box filter. The

Table 4.1: Timing for different segmentation approaches.

Segmentation Approach	Number of Points	Time in ms
Bounding Box Filter	307,200	24.18
Minimal Bounding Box Filter	78,546	11.36
Color Filter	307,200	33.58
RANSAC	23,577	1.99
Plane Filter	69,985	9.79
Thermal Plane Distance Filter	69,985	8.99
Erosion	307,200	0.34
Dilation	307,200	0.34
Opening	307,200	0.67
Background subtraction Filter	217,088	2.40

Table 4.2: Timing for segmentation for lying subject and a person walking towards the camera.

Scenario	Time in ms
Segmentation for medical application	119
Segmentation for walking subjects	121

RANSAC algorithm to find the plane of the stretcher takes less than 2 ms, while both plane filters take around 8 ms. Erosion, dilation and the opening approach to remove the distortions are the fastest filters applied to the full scene. The reason is that all morphological operations are implemented based on the OpenCV library [28], providing already an efficient implementation. The background subtraction, which is used for the weight estimation of standing and walking subjects, takes around 2 ms for a point cloud provided by the Kinect One.

For a successful segmentation and an upcoming feature extraction and weight estimation, the filters have to be aligned. Therefore, the complete process of segmentation is evaluated for the two scenarios with a patient lying on a stretcher, as well as the subject walking towards the camera. The timing is measured for the complete process, including the configuration of the filters. Table 4.2 shows the timings of both scenarios. For the segmentation in the medical scenario with the patient on the stretcher, it takes around 119 ms, averaged over ten segmentations. Although the segmentation for the walking subject is less complex, the processing time is nearly the same, having a value of 121 ms. Due to the missing bounding box filter, and the higher amount of plane inliers, the computational costs are similar.

The segmentation could be improved with a redesign of the code, parallelization, and optimizations, even now parts of the code use optimizations to improve performance, like OpenMP [47]. Additionally, the algorithm could benefit from outsourcing and parallelization of the code on a Graphical Processing Unit (GPU). The computer in the trauma room is already equipped with a GPU which is accessible via the CUDA or OpenCL framework [146, 203].

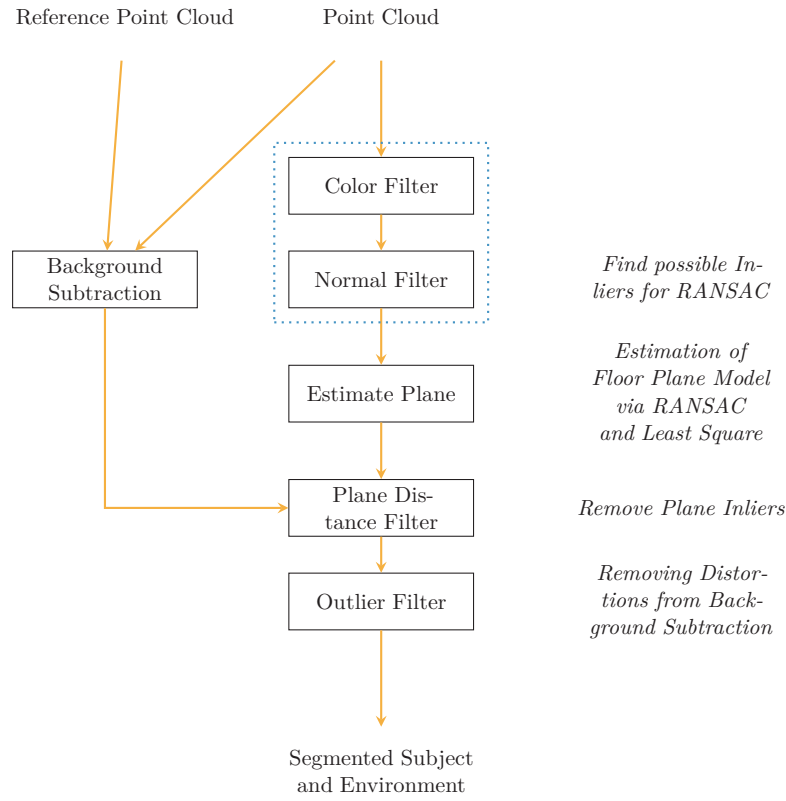


Figure 4.18: Sequence in segmentation for a walking subject: The segmentation based on the background subtraction needs a reference point cloud, recorded without a subject in the FOV. Based on the color and normal filter, possible inliers to detect the ground plane are searched. The floor is removed with the plane distance filter. The final result is additionally filtered to reduce outliers [161].

4.13 Summary

This section illustrated the different approaches to segment a person from the stretcher. The here presented filters are based on state-of-the-art algorithms. Depending on the scenario and the constraints of the environment, segmentation can be eased, e.g., with a fixed color of the surface, the patient is lying on. However, the result of the upcoming estimation of the body weight strongly depends on a successful and accurate segmentation; a combination of the filters is necessary to enhance the outcome of it. Figure 4.18 provides the procedure of the segmentation of a subject standing or walking in front of the camera. Figure 4.19 shows the complete process of segmentation in the scenario with a subject lying on a medical stretcher.

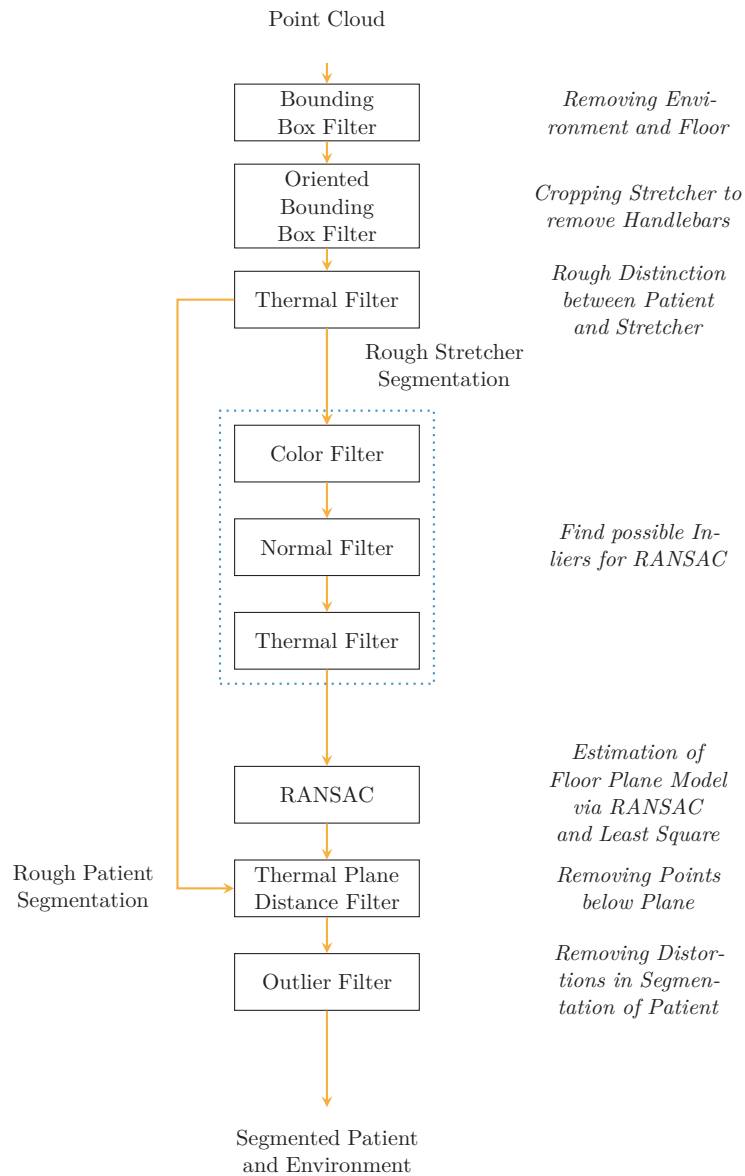


Figure 4.19: Sequence in segmentation for a subject on a medical stretcher: To ensure a robust outcome of the body weight estimation, it is mandatory to differ between the patient, the stretcher, and the surrounding environment. The size of the point cloud is reduced to one fourth based on the bounding box filter, reducing the process time for following algorithms. The thermal filter provides a rough segmentation of the patient and the stretcher. Based on the RANSAC algorithm, the plane of the stretcher is found. Finally, the patient and the stretcher are segmented by the help of the thermal plane distance filter and an outlier filter.

Chapter 5

Feature Extraction for Body Weight Estimation

Based on the segmentation of the scene as shown in the previous chapter, the body weight estimation based on features is presented in this chapter. Figure 5.1 shows the process in feature estimation and body weight estimation: First, on the basis of the extracted point cloud, containing the patient, a feature vector is calculated. Second, the extracted features are used to train an ANN, together with the ground truth body weight. Finally, the output of the trained model provides a value for the body weight on the output.

This research is based on the application of classic machine learning techniques [224], by image processing [205]. All here presented features have the focus being robust and not being affected by the sensor's noise or illumination. Some of the values forwarded to the ANN are synthetic values, generated from computer vision, e.g., the length of an edge or contour. However, some of the feature values are based on real data which is observable for a human, for example, the volume of a human body or the circumference of waist or hip. To improve learning and the approach for body weight estimation, it would be good to compare the estimated volume with the ground truth volume. However, it cannot be practically applied in the medical scenario to measure the volume precisely, although it is possible with hydro-densitometry or air displacement plethysmography [52]. Due to the 3D sensor's noise and further constraints, such observable data always has an error.



Figure 5.1: Process of body weight estimation: Based on the previously segmented point cloud features are calculated. Those features are forwarded to an ANN.

5.1 Feature Extraction

The here presented features rely on geometric approaches. A useful feature is invariant to scale, rotation, translation, and perspective. If a feature is invariant against most of those cases, less training data is needed. For the clinical scenario, the posture of the patients varies only a little: All patients are seen from a frontal view, lying with their back on a stretcher. The size of the stretcher also limits the variety of postures, and most people have their arms aside from their upper body or on their stomach, and their legs stretched or crossed. Also, the perspective does not change that much in the existing environment: The camera is mounted rigidly in the ceiling. Therefore, the perspective can only change a little, depending on the position of the stretcher in the FOV.

5.1.1 Geometric Features

The relation between an object's volume and mass was already found more than 2000 years ago by the Greek Archimedes of Syracuse. As the density of humans does only vary in a specific range – between 968 kg/m^3 and $1,082 \text{ kg/m}^3$ – the volume of a person strongly correlates with the body weight [58]. Therefore, the first feature presented for body weight estimation is the volume v : Assuming that every human has the same density, the body weight m could be easily calculated based on the volume with $m = \rho \cdot v$ with the density ρ . Based on the 3D data provided by one of the depth sensors, the volume of a body can be measured, assuming that the sensor provides ground truth data with no noise. With the previously presented sensor configuration, the patient is only visible from a frontal view from a fixed position. Adding more sensors with different poses to the subject can lead to fewer occlusions, though the back side is never visible and covered by the subject him or herself. To predict the volume of a patient, the visible surface of the stretcher is used to model the back of the patient. Together with the frontal surface, the volume between those two surfaces can be calculated.

In a first step, a triangle mesh M is generated to get the frontal surface, based on the extracted patient's point cloud \mathcal{P}^P . An arbitrary triangle in the mesh $T_i \in M$ consists of the three neighbored points $\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k \in \mathcal{P}^P$. The implementation is taken from the work by Holz and Behnke [85]. The triangulation is only applied to the point cloud containing data from the patient which is integrated into the PCL [155]. The type of meshing can be selected and is illustrated in Figure 5.2. For the developed research, the adaptive triangle mesh was taken,

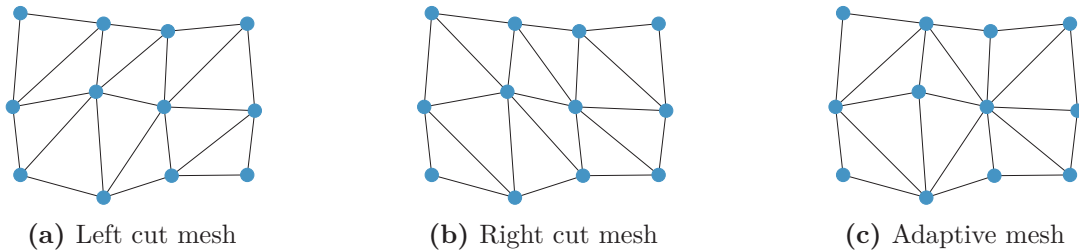


Figure 5.2: Different approaches for triangle mesh: The calculation of the left (a) and right (b) cut is faster compared to adaptive triangulation (c). Source: Holz and Behnke [85]

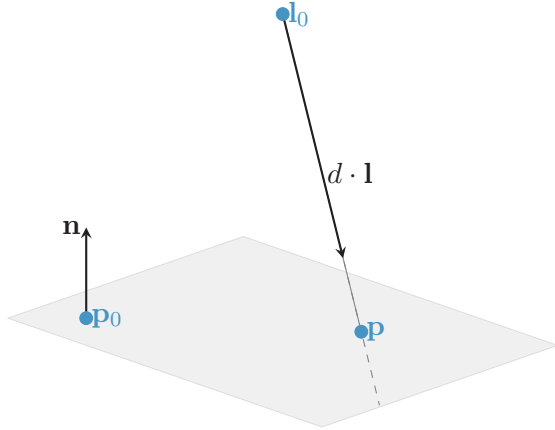


Figure 5.3: Schematic of line-plane intersection: The line \mathbf{l}_i is defined by the sensor's origin $\{0\}$ and a point from the patient's point cloud $\mathbf{p}_i \in \mathcal{P}^P$. The intersection of the line and the previously found plane model \mathbf{h} is now based on equation (5.2).

although the left and right cut are faster to calculate. The adaptive triangle mesh connects points by the shortest distance to a neighbor to generate surface triangles. The set of triangle defines the frontal surface s_f , visible to the sensor's view $\{T_1, T_2, \dots, T_N\} \in M$. In a second step, the back surface of the subject is estimated with the help of the previously extracted plane model \mathbf{h} : For each point on the frontal surface \mathbf{p}_i , a point on the back of the patient \mathbf{p}_i^r is calculated. All of the rear points lie within the previously calculated Hessian plane $\mathbf{p}_i^r \in \mathbf{h}$. The position of a point on the rear is calculated by line-plane intersection [8], which is defined by

$$\mathbf{p} = d \cdot \mathbf{l} + \mathbf{l}_0 \quad , \quad (5.1)$$

where \mathbf{p} is an arbitrary point on the line, \mathbf{l} is a vector in the direction of the line, d is a scalar value, and \mathbf{l}_0 is a point on the line. For the scenario with the patient on the stretcher, an arbitrary point of the patient's frontal surface \mathbf{p}_i defines a line $\mathbf{l}_i \in \mathbb{R}^3$ together with the sensor's origin $\{0\}$. The equation is modified therefore to

$$\mathbf{p}_i = d \cdot \mathbf{l}_i + \{0\} = d \cdot \mathbf{l}_i \quad , \quad (5.2)$$

and simplifies to $\mathbf{p}_i = d \cdot \mathbf{l}_i$ due to the sensor's origin $\{0\} = (0 \ 0 \ 0)^T$. The intersection is found via substituting the equation for the line (equation (5.2)) into the equation for the plane (equation (5.1)), and solved for d :

$$\begin{aligned} 0 &= (d \cdot \mathbf{l} + \mathbf{l}_0 - \mathbf{p}_0) \cdot \mathbf{n} \\ d &= \frac{(\mathbf{p}_0 - \mathbf{l}_0) \cdot \mathbf{n}}{\mathbf{l}_i \cdot \mathbf{n}} \quad , \end{aligned}$$

while an intersection exists if the denominator is unequal to zero $\mathbf{l}_i \cdot \mathbf{n} \neq 0$; otherwise the selected line and the plane are parallel, not having an intersection. With the calculated distance towards the plane of a given point on the line, the intersecting point can be calculated by inserting the value for d in the equation for the line

$$\mathbf{p}_i = d \cdot \mathbf{l}_i + \mathbf{l}_0 = d \cdot \mathbf{l}_i \quad .$$

Figure 5.3 illustrates the general form of the intersection of a plane and a line. Figure 5.4 transforms the general form of the line-plane intersection towards the approach of body weight

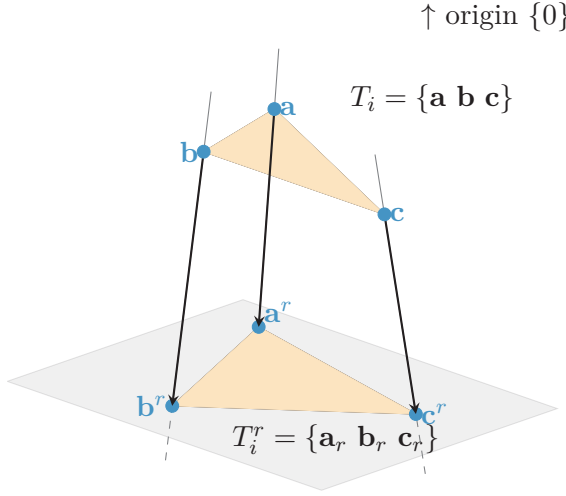


Figure 5.4: Triangle mesh with front surface triangles $T_i = \{\mathbf{a} \ \mathbf{b} \ \mathbf{c}\}$ and back triangles $T_i^r = \{\mathbf{a}_r \ \mathbf{b}_r \ \mathbf{c}_r\}$ used for volume estimation: The front triangles are seen by the sensor directly. The triangles on the back surface are generated by line-plane intersection as shown in Figure 5.3.

estimation. The points on the frontal surface of the subject's point cloud $\mathbf{p}_i \in \mathcal{P}^P$ correspond to the vector \mathbf{l} in the line equation. The intersecting points, which model the back surface of the subject, are noted by \mathbf{p}^r . The line-plane intersection is calculated for every point in the subject's point cloud.

Three neighbored intersecting points define a single triangle on the back $T^r = \{\mathbf{a}^r \ \mathbf{b}^r \ \mathbf{c}^r\}$. Furthermore, the surface of the back is modeled by all triangles lying within the plane model $M^r = \{T_1^r \ T_2^r \ \dots \ T_N^r\}$. Now, three points on the frontal surface and three points of the back surface are forming two tetrahedrons. The goal is now to calculate the volume of those tetrahedrons by subtracting the back tetrahedron and the frontal tetrahedron. The volume of a single arbitrary regular tetrahedron v_T – which means that three edges have the same length – can be calculated with its base area a_0 and its height h , or via the length of a by [107]

$$v_T = \frac{1}{3}a_0h = \frac{\sqrt{2}}{12}a^3 = \frac{a^3}{6\sqrt{2}} \quad .$$

The tetrahedrons generated by the meshing of the point cloud are not regular. In this case, the volume of a tetrahedron v is calculated based on four points $\mathbf{a}, \mathbf{b}, \mathbf{c}$, and \mathbf{d} . Now, the volume of a tetrahedron can be computed by

$$v = \frac{\|(\mathbf{a} - \mathbf{d})((\mathbf{b} - \mathbf{d}) \times (\mathbf{c} - \mathbf{d}))\|}{6} \quad .$$

If the point \mathbf{d} is set to the coordinate's origin $\mathbf{d} = \mathbf{0} = (0 \ 0 \ 0)^T$ the equation simplifies to

$$v = \frac{\|\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})\|}{6} \quad .$$

One arbitrary volume element v_i , established by a frontal triangle T_f and a triangle on the back T_r , can therefore be seen as a difference of two tetrahedrons, while the volume of the

tetrahedron containing the back surface is bigger than the frontal tetrahedron $v_r > v_f$. This difference is calculated by

$$\begin{aligned} v_i &= v_{r_i} - v_{f_i} \quad \text{where} \\ v_{r_i} &= \frac{\|\mathbf{a}_i^r \cdot (\mathbf{b}_i^r \times \mathbf{c}_i^r)\|}{6} \quad \text{and} \\ v_{f_i} &= \frac{\|\mathbf{a}_i \cdot (\mathbf{b}_i \times \mathbf{c}_i)\|}{6} \quad \text{where } \mathbf{a}_i^r, \mathbf{b}_i^r, \mathbf{c}_i^r \in \mathcal{P}^S \text{ and } \mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i \in \mathcal{P}^P, \end{aligned}$$

where \mathcal{P}^S is the set of points generated by the plane-line intersection which are also part of the stretcher. To get the complete volume of a person v , all n volume elements are summed up by

$$v = \frac{1}{6} \sum_{i=1}^n \|\mathbf{a}_i^r \cdot (\mathbf{b}_i^r \times \mathbf{c}_i^r)\| - \|\mathbf{a}_i \cdot (\mathbf{b}_i \times \mathbf{c}_i)\|$$

Of course, this volumetric reconstruction is just a rough estimation, compared to the real volume of a person. Several errors appear: The modeling of the back with a plane is a simplification, while the back of a person is never straight, even when lying. Also, a person breathing changes the volume, but not the weight. For a refined volume, an algorithm for the tracking of the breathing has to be integrated, e.g., by the provided depth information [171], or via the thermal data [88]. Moreover, the patient is lying on a stretcher with a mattress on top. If the patient's back is sinking into the mattress, the volume would be lower than in reality.

The computed volume is invariant against scale, rotation, and translation. Due to sensor noise or the sinking into the mattress, the volume can differ depending on the posture of the patient: A patient lying with a hollow back on the stretcher will have a higher volume value because the camera cannot see the back of the patient. Also, a change in perspective can lead to a change in the value of volume.

For the applied data sets, the value of the volume varies as presented in Table 5.1. Based on the publication from Brožek et al. [34], the volume of the human body is in a range between 71 to 126 liters, while this result is only validated by 20 subjects. However, the visually-measured volume is higher, then the presented value. This can be explained by the sensor's noise in depth measurement. Furthermore, the patient is not visible from all sides, and undercuts can appear behind the patient. Also, the simplified plane model tends to generate a higher volume than ground truth, and the patient is wearing clothes for the measurement, adding additional volume if loose or thick.

In one of the early related publications, see Pfitzner et al. [163], the body weight was estimated by the volume, a fixed density, and a linear regression [163]. In contrast to the upcoming ANN, the model for weight estimation was then only set by linear regression to calculate an average density for all subjects. The results were already better than the estimation provided by the physicians and are presented in detail in the chapter for experiments.

Besides the increase in volume for a heavier person, it is also suspected that the surface of someone increases. Therefore the patient's point cloud is converted to a surface representation. With the help of the previously generated triangle mesh M , the surface s is computed. For an

arbitrary triangle $T_i = \{\mathbf{a}_i \ \mathbf{b}_i \ \mathbf{c}_i\} \in M$ the area is calculated using the Heron's formula with the semiperimeter $l_s = \frac{ab+ac+bc}{2}$ [107]. The surface of a single frontal surface triangle s_f is defined by

$$s_f = \sqrt{l_{s_i}(l_{s_i} - ab_i)(l_{s_i} - ac_i)(l_{s_i} - bc_i)} \quad \text{where} \quad (5.3)$$

$$ab_i = \|\mathbf{a}_i - \mathbf{b}_i\|, \quad ac_i = \|\mathbf{a}_i - \mathbf{c}_i\| \quad \text{and} \quad bc_i = \|\mathbf{b}_i - \mathbf{c}_i\|,$$

while the surface for an arbitrary triangle on the back s_r is calculated with

$$s_r = \sqrt{l_{s_i}^r(l_{s_i}^r - ab_i^r)(l_{s_i}^r - ac_i^r)(l_{s_i}^r - bc_i^r)} \quad \text{where} \quad (5.4)$$

$$ab_i^r = \|\mathbf{a}_i^r - \mathbf{b}_i^r\|, \quad ac_i^r = \|\mathbf{a}_i^r - \mathbf{c}_i^r\| \quad \text{and} \quad bc_i^r = \|\mathbf{b}_i^r - \mathbf{c}_i^r\| \quad .$$

Finally, to get the complete surface s , front and back are summed up for each triangle based on equations (5.3) and (5.4) by

$$s = \sum_{i=1}^n (s_f + s_r) \quad ,$$

where n is the number of triangles on the front or the back. Similar, as predicted for volume, this is only a rough estimation. Also due to the vague back reconstruction with the plane, an error in surface estimation can appear. Moreover, the outcome of the surface value can also depend on the pose of a patient lying on the stretcher: Someone having the arms on his or her stomach will result in a lower surface value, compared to the same person having the arms aside.

Another feature observed to correlate with body weight is the number of points belonging to the patient's point cloud \mathcal{P}^P . Therefore it is defined by the point cloud's cardinal number by

$$f_3 = |\mathcal{P}^P| \quad .$$

Having two persons in the field of view, at the same distance, and one person contains significantly more points than the other person, it is more likely that this person is estimated with a higher body weight. Of course, this is only if the measurement is taken from the same distance and a similar perspective – which is the case in this medical scenario.

Moreover, the density can be calculated, which is defined by the relation of the number of points belonging to the patient $|\mathcal{P}^P|$ towards the complete number of points of the whole point cloud \mathcal{P} . Having the data from a Microsoft Kinect camera, the total number of Cartesian coordinates is 307,200. In contrast to that, data from a Kinect One camera has around 217,088 points. Feature number four is calculated by

$$f_4 = \frac{|\mathcal{P}^P|}{|\mathcal{P}|} \quad .$$

This feature makes the correlation towards the body weight more independent of the sensor's characteristics. Especially the distance between the sensor and the subject is now invariant and does not influence the result of body weight estimation.

The surface is invariant against scale, rotation, and translation. Changes in perspective can lead to deviations of the surface. Furthermore, the posture directly affects the surface: A patient having the arms beside the body will result in a higher surface value compared to someone having his arm crossed on the stomach.

Table 5.1: Statistical values of geometric features: The values are based on two fused data sets, having a total size of 233 subject. The data sets are presented in the upcoming section for experiments.

	Feature	Min	Max	Range	Mean	σ^2
f_1	Volume in liters	59.5	160.6	101.1	99.7	21.7
f_2	Surface in m^2	1.7	3.0	1.3	2.3	2.8×10^{-1}
f_3	Nr. of points	1.6×10^4	4.8×10^4	3.1×10^4	2.9×10^4	8.7×10^3
f_4	Density	7.4×10^{-2}	1.5×10^{-1}	8×10^{-2}	1.1×10^{-1}	1.7×10^{-2}

Table 5.1 shows the distribution of the values for the surface, as well as the density. The average surface in square meter was measured with $1.91 m^2$ on average for male subjects and on average $1.71 m^2$ for female subjects [181]. These values were measured during a trial with 3,613 cancer patients in the United Kingdom to optimize chemotherapy drugs. The rough approximation can explain the difference to these ground truth values from sensor’s noise in depth, as well as that the subject is only visible from a frontal view. Furthermore, the patient is wearing clothes and therefore the average value of $2.31 m^2$ is probably bigger than ground truth.

5.1.2 Features from Eigenvalues

Two subjects can have the same body weight, although they have different body shapes: Someone being thin and tall can have a similar body weight as someone who is small and wide. Via this relation, the visual body weight estimation is performed, by looking at someone and comparing it with the own body weight. Two observations can be declared: The body weight is positively related to the body height $m \sim h$. Also, the width of someone is positively related to the body weight $m \sim w$, while the width itself is a simplification of the circumference of someone. However, to measure the body height and the circumference is challenging, with a 3D sensor due to noise in depth and different postures of a subject.

To use this correlation with machine learning, the eigenvalue and derived equations are chosen because they are more robust to the sensor’s noise and a single outlier does not change the forwarded value for the machine learning approach dramatically. This section describes the extraction of features which are based on eigenvalues λ of the patient’s point cloud \mathcal{P}^P . The eigenvalues are calculated based on PCA. For a detailed explanation of the PCA, see the appendix on page IV: For a point cloud $\mathcal{P} \in \mathbb{R}^3$, three eigenvalues λ_1 , λ_2 and λ_3 exist. The eigenvalues are arranged in descending order $\lambda_1 > \lambda_2 > \lambda_3$.

Figure 5.5 illustrates the eigenvalues of different distributions of an arbitrary point cloud with the size of $n = 100$: For the first set of points, Figure 5.5a, the points are distributed along the diagonal of a xy-coordinate system. This distribution results in two eigenvalues having different values. In the second scene, see Figure 5.5b, the set of points is aligned more homogeneous in the x- and y-direction of a coordinate system. Therefore, the values of the eigenvalues are now more similar. In both scenes, the eigenvalues are visualized as an ellipse or an ellipsoid in \mathbb{R}^3 . The centroids of the point clouds are marked in blue. All three eigenvalues are forwarded to the following machine learning approach, so $f_5 = \lambda_1$, $f_6 = \lambda_2$ and $f_7 = \lambda_3$.

Also, the features from eigenvalues have the benefit that they are invariant to scale, rotation, and translation. Concerning the here presented body weight estimation, the patient's position is not relevant for the outcome of the algorithm. As shown in the correlation analysis on page 106, the eigenvalues have a high correlation with the body weight and therefore provide a useful feature for the following machine learning approach. Moreover, combinations of those eigenvalues represent geometric features, as presented by Linder et al. [121] and Hackel et al. [77]. The first feature forwarded to the machine learning framework for body weight estimation is the sphericity. It describes the roundness of a set of points $\in \mathbb{R}^3$ and is defined by

$$f_8(\lambda) = 3 \cdot \frac{\lambda_3}{\sum_i \lambda_i} .$$

A sphere would have a sphericity with a value of one, based on three equal eigenvalues $f_3(\lambda_1 = \lambda_2 = \lambda_3) = 1$. In case all points are located in a planar shape, the third eigenvalue would be zero, and therefore the sphericity would also be zero [220].

Also, the flatness can be calculated from eigenvalues. It is defined as based on the second and the third eigenvalue, as well as the sum of all three eigenvalues

$$f_9 = 2 \cdot \frac{\lambda_2 - \lambda_3}{\sum_i \lambda_i} .$$

Having a set of points arranged in a plane would lead to a flatness with a value of one. On the other side, a set of points having similar or equal second and third eigenvalues, the value for flatness tends to go towards zero [77].

And last, the linearity can be calculated from eigenvalues. Here the first two eigenvalues are used, together with the sum of all eigenvalues which is defined by

$$f_{10} = \frac{\lambda_1 - \lambda_2}{\sum_i \lambda_i} .$$

A set of points ordered in a line would lead to a flatness value of one. If the first and the second eigenvalue have similar or equal values ($\lambda_1 \approx \lambda_2$), the value for linearity goes towards zero [77].

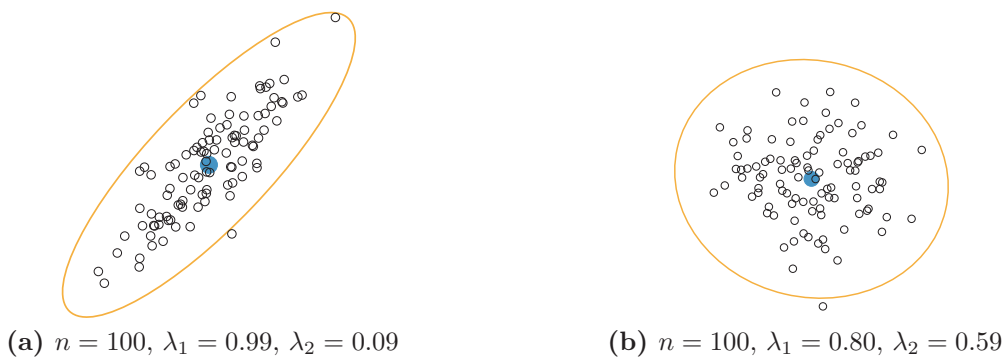


Figure 5.5: Eigenvalues from a set of points visualized as an ellipse: Having a set of $n = 100$ points with distinct eigenvalues – here $\lambda_1 \approx 10 \cdot \lambda_2$ – the ellipse will be drawn as shown in (a). Having similar eigenvalues with $\lambda_1 \approx \lambda_2$ the ellipse will be approximately drawn as a circle. The centroid of each set is illustrated in blue.

Table 5.2: Statistical values from features based on eigenvalues.

	Feature	Min	Max	Range	Mean	σ^2
f_5	1st Eigenvalue	2423	12,246	9,823	6,188	2096
f_6	2nd Eigenvalue	234	1,547	1,313	612	242
f_7	3rd Eigenvalue	34	287	252	104	46
f_8	Sphericity	2.10×10^{-2}	8.35×10^{-2}	6.25×10^{-2}	4.55×10^{-2}	1.17×10^{-2}
f_9	Flatness	8.53×10^{-2}	2.54×10^{-1}	1.69×10^{-1}	1.46×10^{-1}	2.92×10^{-2}
f_{10}	Linearity	6.81×10^{-1}	8.84×10^{-1}	2.04×10^{-1}	8.08×10^{-1}	3.60×10^{-2}

Table 5.2 illustrates the values of the eigenvalues, as well as the features based on eigenvalues.

5.1.3 Statistic Features

Furthermore, features from statistics are added. The first feature mentioned is the standard deviation of the patient's point cloud \mathcal{P}^P , with respect to its centroid $\bar{\mathbf{p}}$. Similar to the features from eigenvalues, the idea for these features is based on Linder et al. [121]. The feature is calculated by

$$f_{11} = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^n (\mathbf{p}_i - \bar{\mathbf{p}})^2} \quad \text{where } \mathbf{p}_i \in \mathcal{P}^P \quad . \quad (5.5)$$

If the set of points is dense and focused on a single point, the sum of the deviation will be low, compared to a point cloud with high variety. This feature is somehow similar with the compactness of a set of points.

Another feature based on statistic is the kurtosis with respect to the centroid $\bar{\mathbf{p}}$. The kurtosis is calculated based on the previously defined standard deviation (equation (5.5)) with respect to the patient's centroid by

$$f_{12} = \frac{\sum_i (\mathbf{p}_i - \bar{\mathbf{p}})^4}{f_{11}} \quad \text{where } \mathbf{p}_i \in \mathcal{P}^P .$$

The kurtosis captures the peakedness of points: For a point cloud with normal distribution, the value for the kurtosis is close to zero. In case more points are closer to the centroid of the point cloud, a leptokurtic distribution arises, having a positive value for the kurtosis. In contrast to that, a point cloud with a platykurtic distribution results in a negative value for the kurtosis.

The last feature from statistics is the average deviation from the patient's point cloud centroid $\bar{\mathbf{p}}$, which is defined by

$$f_{13} = \frac{1}{n} \cdot \sum_{i=1}^N \|\mathbf{p}_i - \bar{\mathbf{p}}\| \quad \text{where } \mathbf{p}_i \in \mathcal{P}^P \quad ,$$

where n is the amount of points in the patient's point cloud. The extracted statistics features f_{11} to f_{13} are invariant against scale, rotation, and translation because the features are related

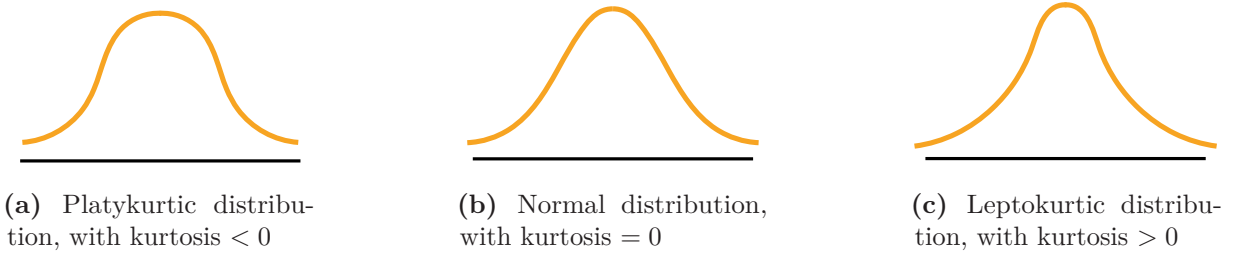


Figure 5.6: Kurtosis for different distributions: The kurtosis of a distribution affects the size of the curvature of a distribution, widening it for a positive kurtosis (a) and narrowing for a negative kurtosis (c).

Table 5.3: Statistical values for the features based on statistics.

	Feature	Min	Max	Range	Mean	σ^2
f_{11}	Compactness	3.97×10^{-1}	5.50×10^{-1}	1.53×10^{-1}	4.84×10^{-1}	3.17×10^{-2}
f_{12}	Kurtosis	2.16×10^3	1.45×10^4	1.24×10^4	6.55×10^3	2.52×10^3
f_{13}	Alt. Compactness	8.67×10^{-1}	9.02×10^{-1}	3.42×10^{-2}	8.85×10^{-1}	6.38×10^{-3}
f_{14}	Distance	2.05	2.28	0.23	2.13	0.05

towards the centroid of the point cloud. However, the features are not invariant against the patient's posture. A change in perspective can also lead to changes in the feature values.

Figure 5.6 shows graphs of a normal distribution with different kurtosis: If the points in the cloud are equally distributed, the kurtosis should have small values. Table 5.3 provides the data of the here presented features, which are calculated on the basis of all data sets.

To increase the robustness to different distances between the camera system and a subject on the stretcher, the distance of the subject is forwarded as an input parameter to the following machine learning approach. For the distance d , the centroid of the patient's point cloud \mathcal{P}^P is used by

$$f_{14} = d = \|\bar{\mathbf{p}}\| \quad \text{where } \mathbf{p}_i \in \mathcal{P}^P \quad .$$

A different distance of the subject can occur due to the adjustable stretcher in the trauma room. Furthermore, the mounting height of the system can vary, depending on the room the system is installed. The recorded data sets which are presented in the experiment section already have different mounting heights and also different stretcher heights.

5.1.4 Contour Features

The contour is a common feature for image processing when it comes to classification. Therefore the contour from the patient's silhouette should be used to improve the outcome of body weight estimation. In contrast to the previously presented feature, the contour is based on a two-dimensional projection towards the sensor plane. A contour describes the border between valid and invalid points in the binary mask. One condition for the feature is, that the person in the FOV is segmented and only one binary blob is visible in the binary mask. To check if an arbitrary point is set as a contour point $\mathbf{p}_c \in C$ the point's four neighbors $\mathcal{N}_4(\mathbf{p}) = \{\mathbf{p}_{n_0}, \dots, \mathbf{p}_{n_3}\}$ have

to be evaluated: The element in the binary mask itself must have a value of zero. If one of the neighbored points is valid, the element is considered to be a contour element c .

Furthermore, the contour C can be extracted based on morphological operations. The contour in a binary mask is the border between the values 0 and 1. From the contour itself, two features can be calculated: First, the length is given by the sum of all points belonging to the contour. Therefore, all pixels belonging to a contour are enumerated to receive the contour's length l_c . This value is added to the feature value with

$$f_{15} = l_c \quad .$$

To obtain the area enclosed by the contour, a_c , the surrounding pixels are counted. This values are added to the feature vector as the 16th element by

$$f_{16} = a_c \quad .$$

The features from the contour's length are invariant against rotation and translation, but not against perspective, scale or posture.

Another feature similar to contours is based on the convex hull. This hull can be calculated by several algorithms [74, 36] providing different complexities. The convex hull itself is defined, so all points of a given set are enclosed via a border. The shape of this border can only be convex, always bending in one direction. In almost the same manner to the contour feature, the length and the area of the convex hull are calculated. Both values are forwarded to the feature vector with f_{17} and f_{18} .

Equivalent to the previously calculated area of the contour it is suspected that the area of the convex hull can be a reliable feature to improve the outcome of the estimation.

Additionally, two features can be derived based on the features from the convex hull and the contour. First, the contour solidity s_c describes the relation between the areas of the contour a_c and the convex hull a_h with

$$f_{19} = \frac{a_c}{a_h} \quad .$$

Having a subject in the field of view, keeping its legs closed or crossed, and the arms close to the side of his or her upper body will result in a low value because the areas of the convex hull and the contour are similar. In contrast to that, a subject with stretched out arms to the side and space between the legs will result in a disproportion of the two values, and the contour will have a more significant value, compared to the convex hull.

The final feature calculated from the contour is the equivalent diameter: Here the area of the contour is used to derive the diameter of a circle, having the same area. This value is related to the area of the contour calculated by

$$f_{20} = \sqrt{\frac{4 \cdot a_c}{\pi}} \quad .$$

Figure 5.7 visualizes the extracted contour and the convex hull: The contour is marked with an orange set of pixels between the valid and invalid points of the subject, see Figure 5.7a. In contrast to that, the convex hull has a shorter distance but fills a bigger surface, as illustrated in Figure 5.7b.

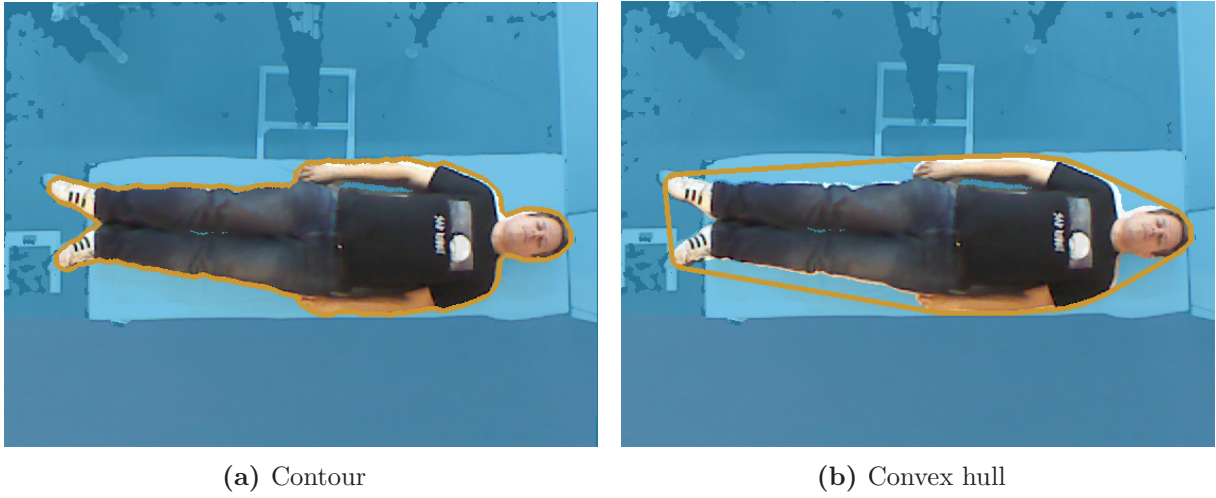


Figure 5.7: Contour and convex hull from the segmented subject: The contour marks the border between the segmented subject and the environment (a). Depending on the posture of the subject, the contour can change in size, for example, if the patient stretches the arms to the side. The convex hull marks an edge around the subject, while this edge can only bend in one direction (b).

Table 5.4: Statistical values for contour features.

	Feature	Min	Max	Range	Mean	σ^2
f_{15}	Contour Length in pixels	8.33×10^2	2.29×10^3	1.45×10^3	1.26×10^3	2.61×10^2
f_{16}	Contour Area in pixels	1.58×10^4	4.88×10^4	3.30×10^4	2.92×10^4	8.75×10^3
f_{17}	Convex Hull Length in pixels	6.10×10^2	1.15×10^3	5.38×10^2	8.64×10^2	1.34×10^2
f_{18}	Convex Hull Area in pixels	1.93×10^4	6.54×10^4	4.62×10^4	3.69×10^4	1.16×10^4
f_{19}	Solidity	2.36×10^1	4.41×10^1	2.05×10^1	3.31×10^1	5.28
f_{20}	Equ. Diameter	1.42×10^2	2.49×10^2	1.07×10^2	1.91×10^2	2.89×10^1

Of course, the contour features can differ sharply with the posture of the patient: Keeping the arms close to the body will result in a much smaller size of the contour, compared to a person with the arms stretched to the side. Although the patients lying on the stretcher were not told to stay in a predefined position, most of them have the arms aside or laid on the stomach. Moreover, with a high set of training data, it is expected that features from the contour improve the outcome of the estimation. However, features from the contour are not invariant for scale, perspective or the patient's posture. Table 5.4 shows the statistic values for the features from the contours.

5.1.5 Features from Personal Data

A patient brought into an emergency room by an ambulance might have an ID in the pocket. Based on the ID, a physician can get the age. Woman and man have different body shapes, and

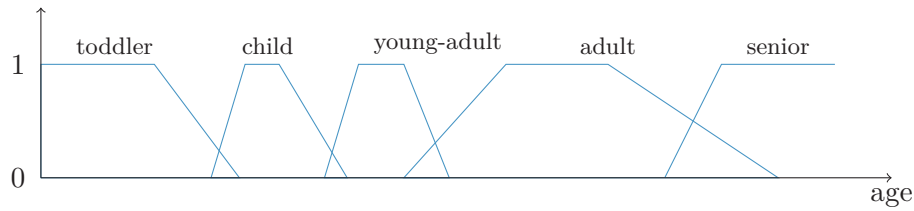


Figure 5.8: A fuzzy logic approach for the age of a subject: A fuzzy logic provides different classes, with linear transitions between each class. The borders between the different classes are overlapping. This approach can be useful if the exact age is not known.

Table 5.5: Statistical values for the age of the subjects. 57 percent of the subjects male and 43 percent are female subjects.

	Feature	Min	Max	Range	Mean	σ^2
f_{21}	Age in years	18	87	51.3	51.3	17.7
f_{22}	Gender	0	1	1	0.43	0.49

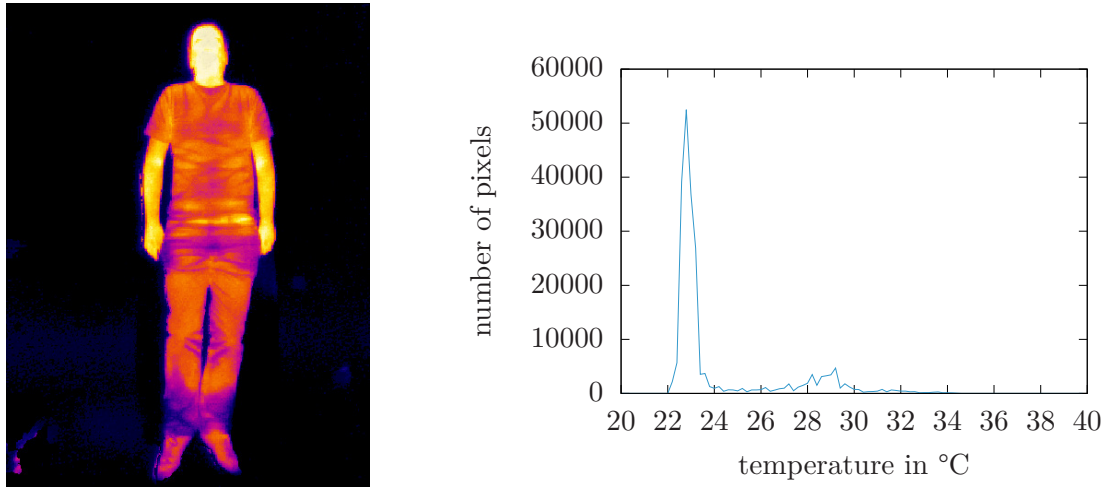
also their body density is different [84]. To forward the gender to the neural network, a binary model is applied, having a value of 0 for female subjects and a value of 1 for male subjects.

For experiments in the upcoming section, the gender is added manually. Although this is not suitable for a fully autonomous weight estimation system, the input to the network is not time-consuming for a physician and can improve the outcome in body weight estimation. However, there exist several different approaches based on template matching from previously extracted faces [73] or deep convolutional neural network to estimate someone's gender [120]. One method mentioned earlier uses similar features as the estimation of the body weight, to perform an estimation of a subject's gender from different perspective view [121].

Another feature to improve the estimation is the age: While aging, the density of fat-free body mass changes [84]. Algorithms to perform an estimation of someone's age exist [120]. However, not always the exact age is available, due to an unconscious patient or the missing identity card. Then the subject's age can be estimated by a treating physician. To minimize an error in estimation, it is further possible to assign the subject to a category, like a toddler, child, adult or senior. These classes are fuzzy without sharp borders: A subject having the ground truth age of 60 years, could be classified as adult, but it is also correct to classify him or her as a senior. Figure 5.8 shows the principle of this fuzzy logic approach for age. The groups for ages are overlapping to model the uncertainty between the classified ages. The slope between two states can be different [228, 105]. Table 5.5 illustrates the distribution of the features from a subject's data. The age was available for 127 subjects, 57 percent of them are male and 43 percent are female subjects.

5.1.6 Thermal Features

Up to this point, the thermal camera is only used to enhance segmentation. To improve the estimation of body weight, the sensor's data should also be considered as a feature. The two



(a) Thermal view of the scene including the subject on the stretcher

(b) Thermal histogram of complete scene from (a)

Figure 5.9: Thermal features: The raw thermal data (a) is transformed into a histogram (b) to process the ambient temperature by finding the peak in the histogram.

features with the highest correlation – the volume and the surface – strongly depends on the clothes someone is wearing: Thick clothes like a sweater or a down jacket result in a higher volume, as well as a bigger surface and would, in consequence, lead to a weight estimation with a greater error.

Figure 5.9 illustrates the data from the thermal camera in the estimation scenario, as well as a histogram: In the thermal image, the subject is visible. While the stretcher and the floor are colored in dark blue, the subject itself is colored in bright colors, indicating that these parts are warmer. When looking at the histogram, a first significant peak can be noticed, representing the temperature of the environment. The position of this peak depends on the ambient temperature. It moves to the right on the x-scale for a warmer ambient temperature, and to the left, if the ambient temperature is lower. The size of the peak illustrates the amount of visible environment. If the distance to the subject is lower, fewer parts of the environment would be visible. Furthermore, if not only the subject is in the FOV of the sensors, this peak can be smaller because less environment with ambient temperature is visible. Features extracted from the patient’s thermal histogram are the minimum temperature t_{\min} , the maximum temperature t_{\max} , as well as the average temperature \bar{t} .

The ambient temperature is calculated from the histogram by looking for the biggest peak. To ensure that outliers are not taken for the ambient temperature, the data for the ambient temperature is filtered for a range, for example, 18°C to 26°C. Otherwise electronic devices which generate waste heat could confuse the finding of the ambient temperature. The approach is only tested for the windowless trauma room: If the room has windows and the sun can shine into the room on the floor, a false ambient temperature could be sensed from the thermal camera.

Table 5.6: Statistical values for the minimum, maximum and average temperature of the subject, as well as the ambient temperature.

	Feature	Min	Max	Range	Mean	σ^2
f_{23}	min. Temperature in °C	15.2	23.0	7.8	20.9	1.20
f_{24}	max. Temperature in °C	31.3	37.0	5.7	34.3	0.90
f_{25}	avg. Temperature in °C	25.0	29.3	4.3	27.6	0.82
f_{26}	amb. Temperature in °C	19.1	26.8	7.7	23.6	1.45

To get around that issue, a simple temperature sensor could be added to the system to sense the ambient temperature.

Another approach to handle the different kinds of clothing could be the date or the current weather the patient is brought to the trauma room for treatment: On a summer day with temperatures above 20°C it is more likely that someone is wearing only thin clothes, like a shirt. In contrast to that, the probability increases that someone is wearing thick clothes on a cold winter day. The additional volume visible to the sensors can be an issue for the weight estimation. However, a patient is usually undressed from thick clothes to ease medical treatment.

Apparently, the thermal features are invariant against scale, translation, and rotation. In case the patient is wearing reflective material, e.g., a belt buckle, a change in perspective can lead to a reflection and therefore a change of the measured temperature for some pixels. Nevertheless, alone standing thermal features are not invariant against a change of the ambient temperature, a machine learning approach should have few problems to learn the context of the thermal features and the ambient temperature.

5.2 Feature Extraction for Standing and Walking People

Besides the body weight estimation for clinical applications, it should also be possible to estimate the body weight of someone standing or even walking in front of the camera. First, the person in the FOV has to be segmented from the background: Based on the acquired depth image and the calculated point cloud, filters from section 4 can be applied. In an initial step, the floor is removed from the scene based on the RANSAC algorithm, including a plane model. If the camera remains on a static spot concerning the scene, the calculation of the plane has to be done once. For a reasonable estimation of the floor's inliers the scene should be filtered in advance, for example, only points below a certain threshold should be forwarded to RANSAC. If the floor has a homogeneous color, it could also be helpful to apply a color filter in advance. To finally segment the person from the scene multiple solutions exist: Filtering for edges in the depth image can be one way to segment the person. Also, background subtraction can be applied, if the scene remains static, except the moving person. It has to be considered that RANSAC might filter parts of the feet. Overall, the segmentation of a person standing in front of a camera does need less computational effort in contrast to the segmentation of a patient lying on a stretcher.

Figure 5.10 illustrates the raw scene as a sequence, as well as the results from segmentation. Markers on the floor highlight the starting and the ending of the walk, so the person is continuously

in the FOV of the sensor. The rear marker is applied to ensure that the sensor can obtain depth information.

The FOV, the person's height, and the maximum distance for 3D data acquisition mark the starting and end markers on the floor, see Figure 5.11. Figure 5.12 illustrates the poses of all people walking towards the camera. Due to the different settings for the experiment, the path people tend to walk differs. Furthermore, the camera did not always have the same orientation towards the floor or was mounted at the same height.

It is expected that the accuracy of body weight estimation reduces due to the higher amount of different poses, while walking, compared to someone lying on a medical stretcher. Moreover, not all previously presented features can be used: Features like the volume – which has the highest correlation with body weight – have a dependency on the obtained reference surface. This reference plane is now missing; therefore such a feature cannot be used. Though, the recording of a sequence of shots can help to improve the outcome, in contrast to the single shot for the clinical approach.

5.3 Changes in Posture

Most values in the here presented feature vector are not variant against a change in posture, as presented in Table 5.10 on page 111. Therefore this section presents the changes in the feature vector, depending on the posture of a subject. A change of the posture is more likely for the weight estimation of standing or walking subjects.

Table 5.7 demonstrates the changes of the feature values with different postures: The first scene shows a subject standing straight with the arms aside. The features are listed and calculated by the previously presented equations. In the second scene, the subject raises both hands a bit. The values for surface and density do not change much. Also, the first eigenvalue nearly stays the same. However, the value for the third eigenvalue changes, due to the arms raised in front of the person. Flatness and sphericity – which correlate with the third eigenvalue – also change significantly. Compared to the third scene, where the subject stands with legs apart, the second eigenvalue changes most. Therefore, also flatness and linearity change. Comparing the first and the fourth scene, the subject crosses the arms: The surface lowers, as well as all features from the contour and the convex hull. The second eigenvalue decreases while the third eigenvalue increases. In the last scene, the subject is wearing a backpack. Comparing the features from this scene with the first scene, most of the features are within the same range. However, there are differences due to slight differences in posture. The body weight estimation can ignore such a thing as a backpack, if not visible to the sensor – in contrast to a common scale.

Concerning all here presented poses the features from contour and convex hull can vary most: A subject having the arms aside can cause a much higher length in contour when there is a small gap between the body and the arm. However, as shown by Pfitzner et al. [162], the length and the area of the contour correlate with the body weight, and therefore it can be useful to enclose such features for body weight estimation. The detailed results of the estimation of walking subjects are presented in the experiment section.

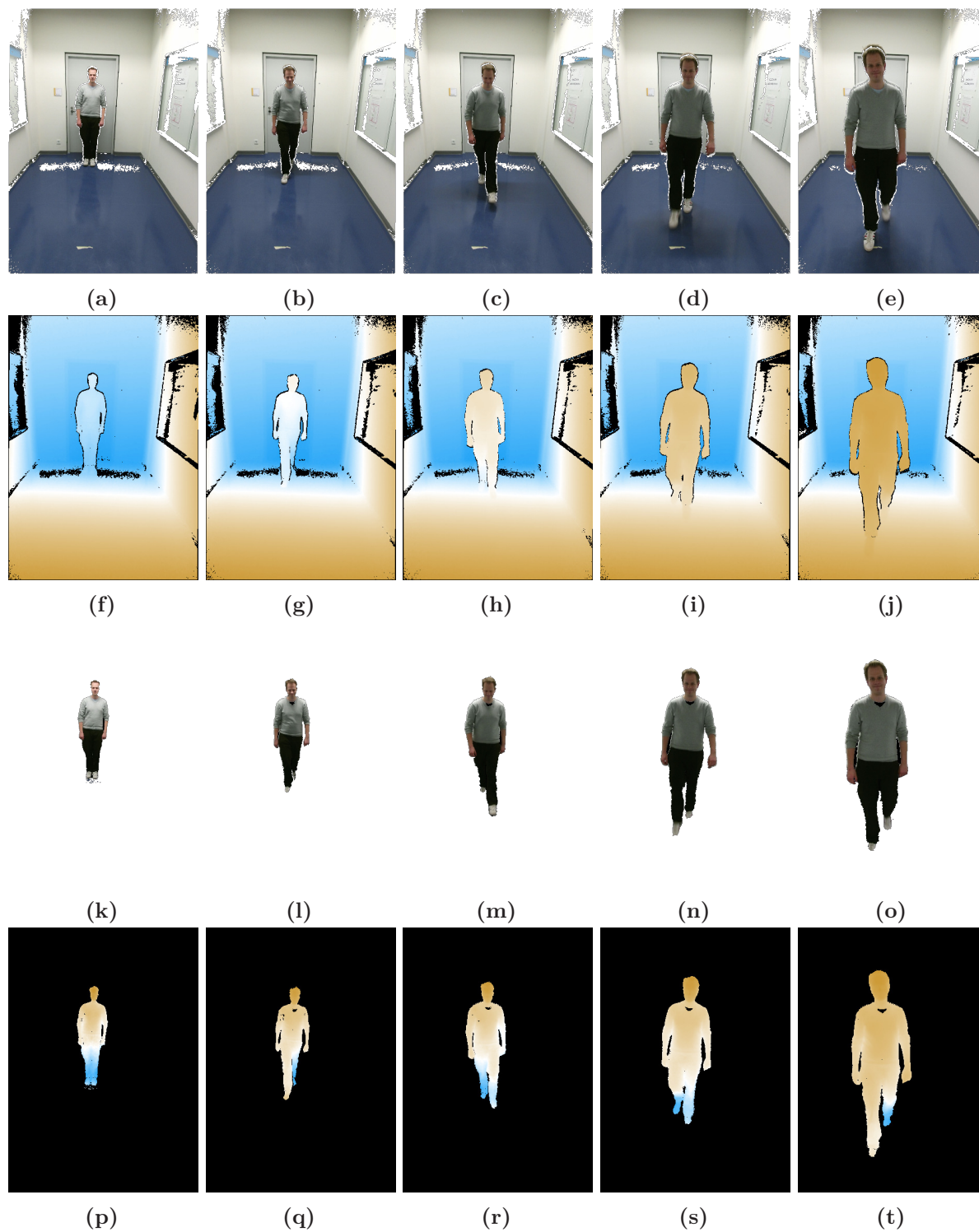


Figure 5.10: Sequence of someone walking towards the camera: (a) - (j) illustrate the raw scene in color and depth, while (f) - (t) show the segmented person. The sequence was recorded over five seconds with a Kinect One camera.

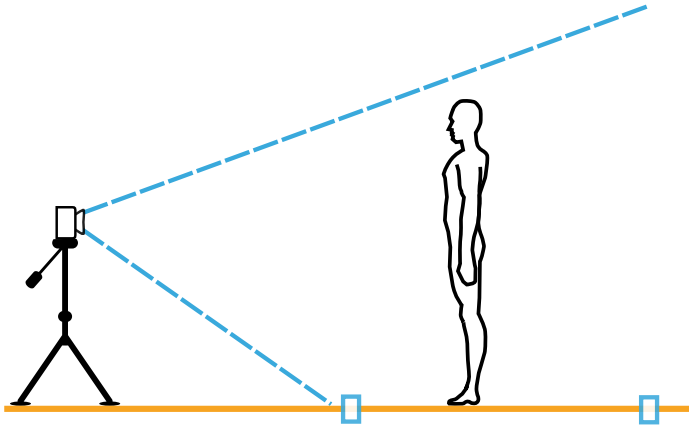


Figure 5.11: Schematic for experiment with people walking towards the sensor: The people stand at the first marker. While walking towards the second marker close to the camera every frame from the sensor is saved for offline processing. The recording is stopped when the second marker is reached.

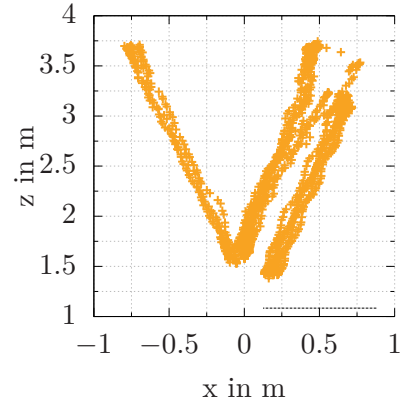


Figure 5.12: Poses of 14 people walking towards the camera: The complete data set consists of several independent experiments. Therefore, the poses of the people walking differ, depending on the orientation of the camera.

5.4 Correlation between the Features and Body Weight

In this section, the extracted features from the previously presented method are compared with each other, depending on their correlation to ground truth body weight. Figure 5.13 illustrates the correlation of all features. For the estimation of the body weight, the correlation for an arbitrary feature towards the ground truth body weight is essential.

The correlation coefficient between two variables $(x_1, y_1), \dots, (x_n, y_n)$ is calculated by

$$\rho(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \cdot \sum_{i=1}^n (y_i - \bar{y})^2}} \quad ,$$

where \bar{x} and \bar{y} are the mean values of both variables. Two variables are not correlating if the correlation coefficient has a value of zero. A value close to one means that both variables have a strong positive correlation, while a negative coefficient is an indicator for a negative correlation [8].


The first column and the first row present the raw correlation values for an arbitrary feature to the ground truth body weight. Especially the first seven features, including the volume, surface, and the eigenvalues have a strong positive correlation to the ground truth body weight. The features from eigenvalues, sphericity, and flatness also have a high positive correlation value, while the linearity has a strong negative correlation. In contrast to that, the features from the thermal camera have a weak value correlation with the weight; as expected.

The correlation analysis is not necessary for the body weight estimation because the ANN adapts to the different correlation between the features and the demanded output of the ANN; the body weight. However, it is essential to understand, which elements in the feature vector provides a useful feature to perform the estimation.

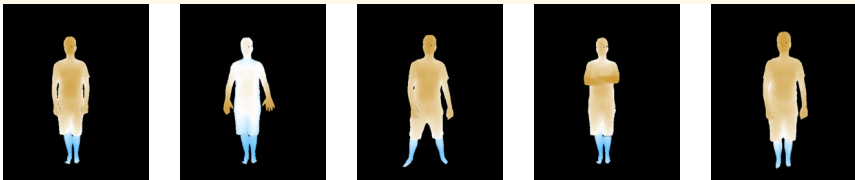
Table 5.7: Changes in features with different poses: Five different scenes illustrate the change in feature values depending on the posture.

Features	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5
Surface	9.5×10^{-1}	9.7×10^{-1}	9.6×10^{-1}	8.6×10^{-1}	9.7×10^{-1}
Density	1.1×10^{-1}	1.1×10^{-1}	1.2×10^{-1}	1.0×10^{-1}	1.3×10^{-1}
1 st eigenvalue	4.7×10^3	4.7×10^3	5.1×10^3	4.5×10^3	5.4×10^3
2 nd eigenvalue	3.9×10^2	4.6×10^2	6.9×10^2	2.5×10^2	5.2×10^2
3 rd eigenvalue	7.1×10^1	3.2×10^2	7.9×10^1	9.4×10^1	7.7×10^1
Sphericity	4.1×10^{-2}	1.8×10^{-1}	4.0×10^{-2}	5.8×10^{-2}	3.8×10^{-2}
Flatness	1.2×10^{-1}	5×10^{-2}	2.0×10^{-1}	6×10^{-2}	1.4×10^{-1}
Linearity	8.3×10^{-1}	7.7×10^{-1}	7.5×10^{-1}	8.8×10^{-1}	8.1×10^{-1}
Compactness	4.6×10^{-1}	4.6×10^{-1}	4.6×10^{-1}	4.7×10^{-1}	4.5×10^{-1}
Kurtosis	5.4×10^3	5.5×10^3	6.2×10^3	5.0×10^3	6.0×10^3
Alt. Compactness	8.6×10^{-1}	8.7×10^{-1}	8.6×10^{-1}	8.6×10^{-1}	8.7×10^{-1}
Distance	1.8	1.8	1.7	1.8	1.6
Contour length	1.0×10^3	1.4×10^3	1.4×10^3	1.1×10^3	1.4×10^3
Contour area	2.5×10^4	2.5×10^4	2.8×10^4	2.1×10^4	1.4×10^3
Convex hull length	8.2×10^2	8.3×10^2	9.3×10^2	8.0×10^2	8.8×10^2
Convex hull area	3.0×10^4	3.5×10^4	4.3×10^4	2.6×10^4	3.7×10^4

color



segmented depth



5.5 Comparison of Subject Extrema

This section should give an impression which feature values change for different subjects. The set of two people is selected manually by similar ground truth body weight, same body height, and same gender, if possible. Figure 5.14 shows real patients from the data set recorded in the trauma room with strongly different characteristics; the faces of the patients are blurred unrecognizable. Table 5.8 on page 108 provides the feature values from the volume, the surface, the density and all three eigenvalues.

Figure 5.14a and 5.14b show two subjects, one being light with 48.6 kg and the other one heavy with 129 kg. Besides the substantial difference in body weight, their body height is similar, with 160 cm and 168 cm. Furthermore, both subjects are female, which gives a good selection for a direct comparison of the difference in extracted features. Table 5.8 shows some of the extracted

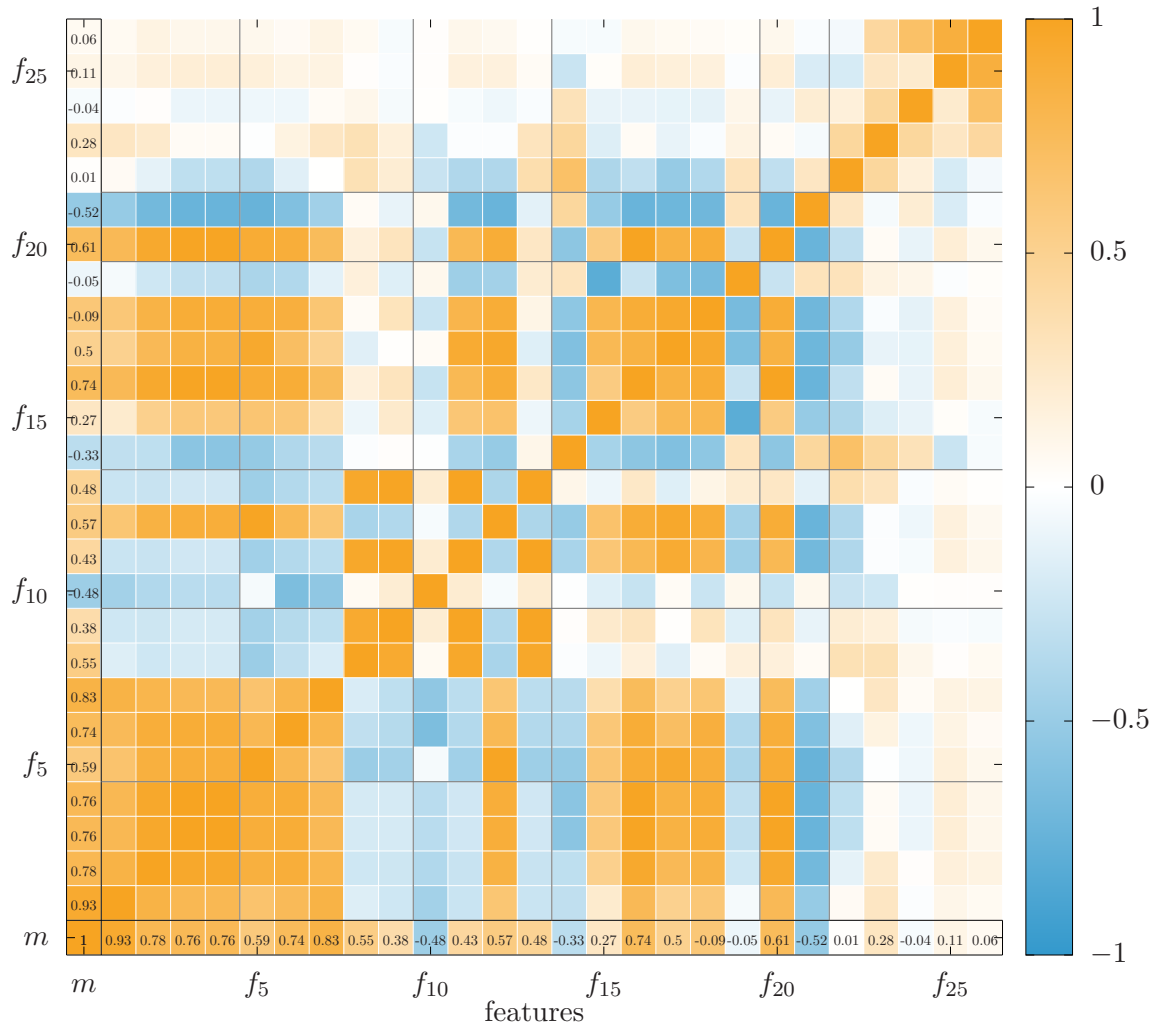


Figure 5.13: Correlation ρ between extracted feature vector against the ground truth body weight m : A feature with a high correlation towards the body weight is a good feature for the body weight estimation based on an ANN. Although, features with low correlation can help to improve the result in the estimation. The values in the first column and row give the precise correlation towards the ground truth body weight.

features for better comparison: Most of the feature values are doubled or even more for the heavy subject, due to a positive correlation between the features, which is presented in section 5.4 on page 104. The only feature which is higher for the lighter subject is the second eigenvalue. Deriving the eigenvalue-based features from the eigenvalues, the lighter subject would have a higher linearity, caused by the lower second eigenvalue.

Figure 5.14c and 5.14d illustrate two subjects, one being small with a body height of 166 cm and the other subject being tall with a body height of 194 cm. Unfortunately, both patients

have a difference of several kilograms in body weight with 84 kg for the small subject, and 91 kg for the tall subject. When looking at the raw feature values, the first four values in the table provide similar values for both subjects. When looking for the eigenvalues, the first eigenvalue is higher for the tall person, which is obvious, because the first eigenvalue represents the height of a subject. For the second eigenvalue, both subjects have similar values. The third eigenvalue is the only values beside the volume which is higher for the smaller subject.

Figure 5.14e and 5.14f show two subjects, with similar body weight – 78.7 kg for the male and 78.5 kg for the female subject. Additionally, the difference in body height is low, with 173 cm for the male and 165 cm for the female subject. Most of the features are in the same range. The volume is higher for the female subject than for the male person, which indicates for the same body weight, that the density is smaller.

Figure 5.14g and 5.14h demonstrate the difference in feature values for a young female, with an age of 18 years, weighing 55 kg, and an old female, with an age of 86 years and a body weight of 56 kg. Furthermore, their body height is similar with 167 cm for the young subject and 165 cm

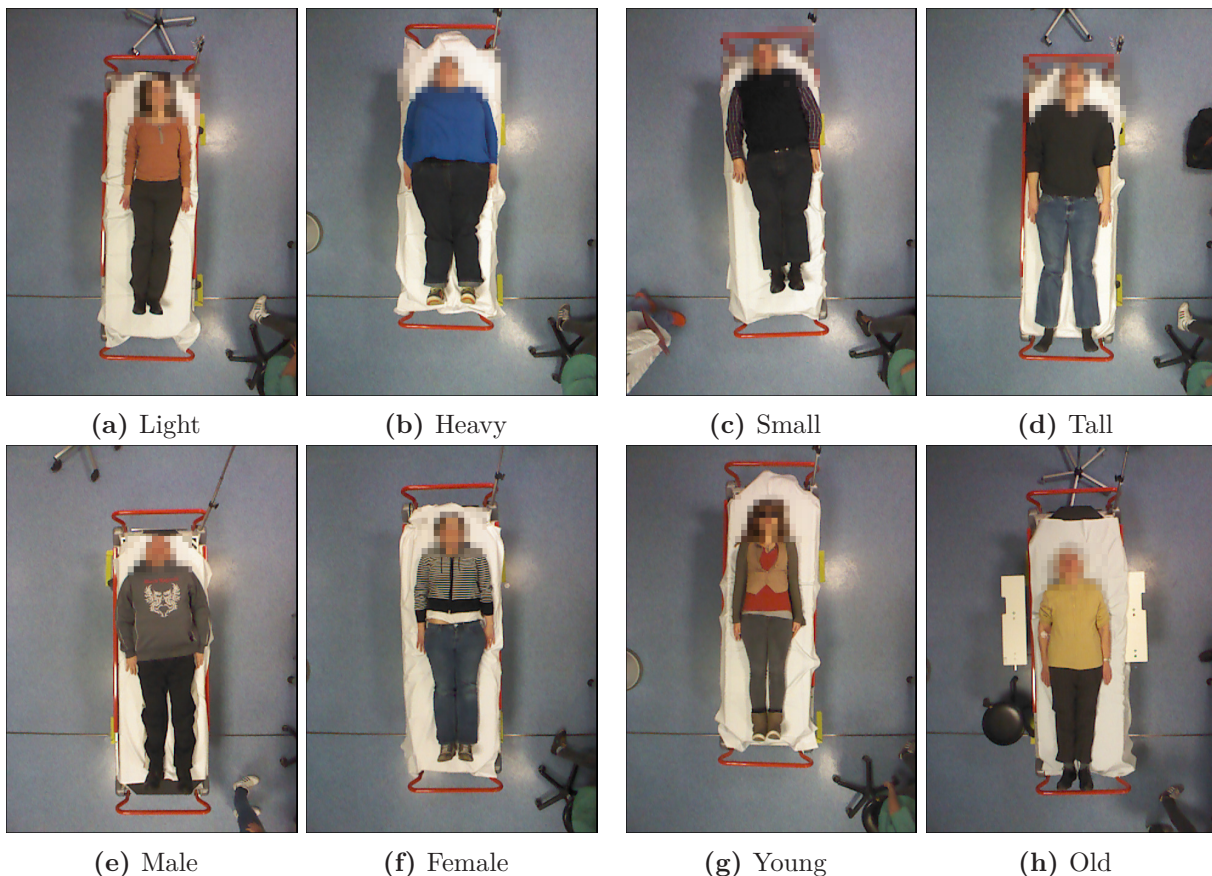
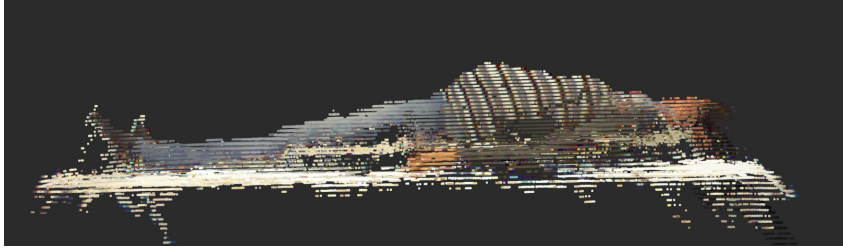


Figure 5.14: Comparison of strongly differing subjects with light and heavy, small and tall, male and female, as well as young and old. The subjects are recorded in the trauma room of the Hospital in Erlangen. The faces are blurred due to anonymization.

Table 5.8: Comparison of features from subjects with strongly differing characteristics.

		Weight in kg	Volume in liters	Surface in m ²	Nr. of Points	λ_1	λ_2	λ_3
weight	thin	48.6 kg	6.1×10^1	1.7	2.6×10^4	4.7×10^3	2.7×10^2	3.5×10^1
	heavy	129 kg	1.6×10^2	2.7	4.3×10^4	7.6×10^3	1.1×10^3	2.5×10^2
size	small	91.7 kg	1.2×10^2	2.4	3.9×10^4	6.6×10^3	9.1×10^2	1.4×10^2
	tall	83.8 kg	1.1×10^2	2.7	4.2×10^4	1.1×10^4	10.0×10^2	1.1×10^2
gender	male	78.7 kg	6.8×10^1	1.9	2.9×10^4	5.5×10^3	3.9×10^2	5.8×10^1
	female	78.5 kg	8.2×10^1	2.0	3.1×10^4	5.6×10^3	4.6×10^2	6.8×10^1
age	young	55 kg	6.8×10^1	1.9	2.9×10^4	5.5×10^3	3.9×10^2	5.8×10^1
	old	56 kg	8.2×10^1	2.0	3.1×10^4	5.6×10^3	4.6×10^2	6.8×10^1

**Figure 5.15:** Pregnant woman: The woman has a ground truth body weight of 91.1 kg and was estimated with a weight of 90.8 kg. She is the only pregnant woman in the data set and was not in the data set for training.

for the old person. The most significant difference can be found in the volume, as well as the second and third eigenvalue. The other features are similar. A different constitutional type or clothing can explain the variance in volume and eigenvalues.

Figure 5.15 emphasizes a unique subject in the available data set: In one of the public events, a pregnant woman was estimated with the *Libra3D* approach. While no similar data was available for the previously applied training, the estimation of the woman was close to the ground truth value.

5.6 Timing Analysis for Body Weight Estimation

As mentioned in the introduction, time is a crucial factor in the treatment of patients in the trauma room, especially stroke patients. It is not essential to speed the algorithm up to a time for a single estimation in the range of milliseconds. However, the result should be available to the physicians within a reasonable time, e.g., less than 5 seconds. Therefore, this sections presents the overall time needed for the estimation on different platforms.

The algorithm – including the sensor fusion, the feature extraction and the forwarding to an artificial neural network is implemented on a conventional desktop computer, which is installed in the trauma room. The computer is equipped with an Intel i7 of the 4th generation, can provide the result in body weight estimation within 300 ms, including the saving of the sensor data. The software does not rely on specialized hardware, like a high-end graphics card, although the

Table 5.9: Tested hardware including time measurements for the estimation: The most significant part of processing time is used to segment the patient from the environment. In contrast to that, the extraction of the features and the processing via an artificial neural network is small. The total time includes visualization and logging during the processing. Changing the parameterization of filters used for segmentation can result in different timing.

	Desktop Computer	Dell M4800 Mobile Computer	Raspberry PI 3	Asus Tinkerboard
Processor	Intel i7-4820K	Intel i7-4900MQ	ARM Cortex-A53	Rockchip RK3288
Nr. of Threads	8	8	4	4
max. Clock	3.90 GHz	3.80 GHz	1.2 GHz	1.8 GHz
TDP	130 W	47 W	< 3.7 W	5 W
Time for Segmentation	239 ms	245 ms	5321 ms	2661 ms
Time for Estimation	22 ms	23 ms	267 ms	212 ms
Total Processing Time	263 ms	270 ms	5604 ms	2885 ms

processing speed could benefit from parallelization. For offline processing, a mobile computer (Dell M4800) is used, having a maximum power consumption of less than 80 Watt. Therefore, the complete hardware could be designed with less than 100 W, including the mobile computer, the thermal camera (<2.5 Watt) and the Microsoft Kinect (12 Watt) [161].

Table 5.9 shows the time for a single estimation for four different platforms, a desktop computer, a mobile computer and two embedded computers: The processing time for the desktop computer and the mobile computer is similar with around a quarter of a second till the result is presented on the monitor. The time on the embedded computer is longer: With a reduced visualization, and without the saving of the sensor’s data to the database, this configuration provided the estimation of body weight in around 5 seconds for the raspberry pi 3. The system is then limited in real-time visualization, as well as process time, and the estimation of the body weight is available with a higher delay. However, the embedded computer can have the benefit of lower power consumption and a smaller footprint, which provides easier integration in the clinical environment. The Asus Tinkerboard, which is equipped with a more powerful processor in contrast to the raspberry pi 3, takes about half of the time for the computation.

The estimation in less than five seconds is always faster than the measurement on a scale for the patient in the trauma room being treated - even for the platform with the lowest computational power. Experiments together with the University Hospital Erlangen, Germany, showed that a time of around five seconds is acceptable for the responsible physicians.

5.7 Summary

Various hand-crafted features are presented in this section. The features are selected based on hypotheses, which feature can improve the outcome of body weight estimation. The generated feature vector is then forwarded to a machine learning approach, to approximate the body weight. Because the size of the patient’s point cloud is several times smaller than the original scene’s

point cloud, the process time and the complexity of the algorithms for feature extraction are mostly low, compared to the segmentation of the patient, processing initially on the full point cloud. Additionally, this section illustrated the correlation between the extracted features to the ground truth body weight. Furthermore, the change in the feature values is presented for different postures and different kind of subjects. Table 5.10 summarizes the extracted features and provides.

Table 5.10: List of features for body weight estimation. The table further lists the invariances of each feature by scale (s), rotation (r), translation (t), perspective (pe) and posture of the person (po) with + (invariant), 0 (invariant with limitations) and - (not invariant).

	Feature	Invariance					Equation
		s	r	t	pe	po	
f_1	Volume in liters	+	+	+	0	-	v
f_2	Surface area in m^2	+	+	+	0	-	s
f_3	Number of Patient's Points	-	+	+	0	-	$ \mathcal{P}^P $
f_4	Density	-	+	+	0	-	$ \mathcal{P}^P / \mathcal{P} $
f_5	1. Eigenvalue	+	+	+	0	-	λ_1
f_6	2. Eigenvalue	+	+	+	0	-	λ_2
f_7	3. Eigenvalue	+	+	+	0	-	λ_3
f_8	Sphericity	+	+	+	0	-	$\lambda_3/\sum_j \lambda_i$
f_9	Flatness	+	+	+	0	-	$2 \cdot (\lambda_2 - \lambda_3) / \sum_i \lambda_i$
f_{10}	Linearity	+	+	+	0	-	$(\lambda_1 - \lambda_2) / \sum_i \lambda_i$
f_{11}	Compactness	+	+	+	0	-	$\sqrt{1/n-1 \sum_i (\mathbf{p}_j - \bar{\mathbf{p}})^2}$
f_{12}	Kurtosis	+	+	+	0	-	$1/n \sum_j \ \mathbf{p}_j - \bar{\mathbf{p}}\ $
f_{13}	Alt. compactness	+	+	+	0	-	$\sum_j (\mathbf{p}_j - \bar{\mathbf{p}})^4 / f_{11}$
f_{14}	Distance to person	+	+	+	+	0	d
f_{15}	Contour length in pixels	-	+	+	-	-	l_c
f_{16}	Contour area in pixels	-	+	+	-	-	a_c
f_{17}	Convex hull length in pixels	-	+	+	-	-	l_h
f_{18}	Convex hull area in pixels	-	+	+	-	-	a_h
f_{19}	Solidity	+	+	+	-	-	a_c/a_h
f_{20}	Equivalent Diameter	-	+	+	-	-	$\sqrt{4 \cdot a_c / \pi}$
f_{21}	Gender	+	+	+	+	+	
f_{22}	Age	+	+	+	+	+	
$f_{23} - f_{26}$	Temperature	+	+	+	+	+	

Chapter 6

Supervised Learning

With supervised learning, an algorithm optimizes its output by previously available data sets, including ground truth. An ANN is such a supervised learning approach. Neural networks were first mentioned in 1943 by Mcculloch and Pitts [134]. They wanted to find a mathematical representation of information processing of a biological system [221] and mimic the function of a neuron of a living being. Such networks can be used for classification or also for regression of non-linear functions. In this early stage, the build networks were trivial because of the limited computational processing power.

Due the last years and the gain in computational processing power, neural networks had a revival for image processing. Especially on GPU, a high amount of data can be used for learning, which is done with lower time consumption. Today big data sets are available for the training of machine learning approaches, like ANN. With the higher processing power, new techniques like deep learning emerge [117]. For a deep learning approach, a high amount of training data based on fully labeled images is forwarded to a neural network. Often, for each pixel in an image, the network provides an input neuron in the first layer, while a deep learning network consists of several layers with different tasks. In contrast to that, the classic machine learning approach first extracts features from an image and forwards them to a neural network. The amount of data forwarded to the network is therefore reduced by a multiple. However, the classic machine learning approach demands a selection of adequate features, suitable for training, while the deep learning selects suitable features from the image while training. A deep learning approach for the estimation of body weight is conceivable, but not possible with the amount of data available in the *Libra3D* project.

Computer vision and adaptive self-learning algorithms are one reason for the improvement of CAD in the last decades of medical imaging. Also in the treatment and detection of stroke, CAD is applied to speed-up and improve the diagnosis of acute ischemic stroke [188, 211, 148]. Furthermore, the treatment of stroke is mentioned in related work [206].

This section shows, how the previously extracted features are forwarded to an ANN, its learning behavior via backpropagation and the estimation of the body weight based on forward propagation. The here described machine learning approach was first implemented in Matlab [131]. However, for a better integration, the ANN was applied via the Fast Artificial Neural Network Library [147]. The core library provides all basic functionalities for classification and function

regression based on neural networks. The library is written in C, while wrapper classes for C++ exist.

6.1 Model of a Single Neuron

The basic idea of a neuron comes from biology: Nerve cells are connected via synapses, transporting electrical or chemical signals to other cells or actors, like muscles. The model for an artificial neuron is simplified and focuses on the main working principle of a real neuron. Figure 6.1 illustrates an artificial neuron, with its components. A single neuron has an input vector $\mathbf{x} = (x_1, \dots, x_n)$, containing n elements. Each element in the input vector owns a scalar weight w , which can be changed independently of each other. Additionally to the input vector, a bias b can be added, providing a constant offset. The output of an artificial neuron is a single scalar value u . The input vector \mathbf{x} directly determines the value of the output, the scalar weights $\mathbf{w} = (w_1, w_2, \dots, w_n)$ for each input value, the bias value b , and an activation function $g(\cdot)$. Therefore, the output of a single neuron is defined by

$$u(\mathbf{x}, \mathbf{w}) = g\left(\sum_{i=1}^n w_i x_i + b\right),$$

where n is the number of inputs of a neuron. Three different activation functions are common. Figure 6.2 shows the different characteristics of activation functions.

- **Step function:** With a step function, the output of a neuron can have only two values, 0 and 1. The function is defined by

$$g(x) = \begin{cases} 0 & \text{where } x \leq k \\ 1 & \text{where } x > k \end{cases} . \quad (6.1)$$

The threshold k is used to adapt the step function and the transition between 0 and 1. Figure 6.2a shows the step function with $k = 0$. Especially for classification, the step function is applied, to determine if a set of input values belongs to a class $u = \text{true}$ or

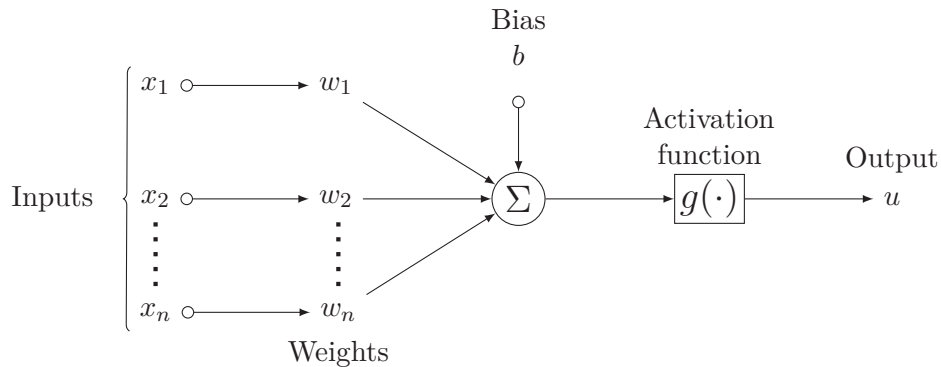
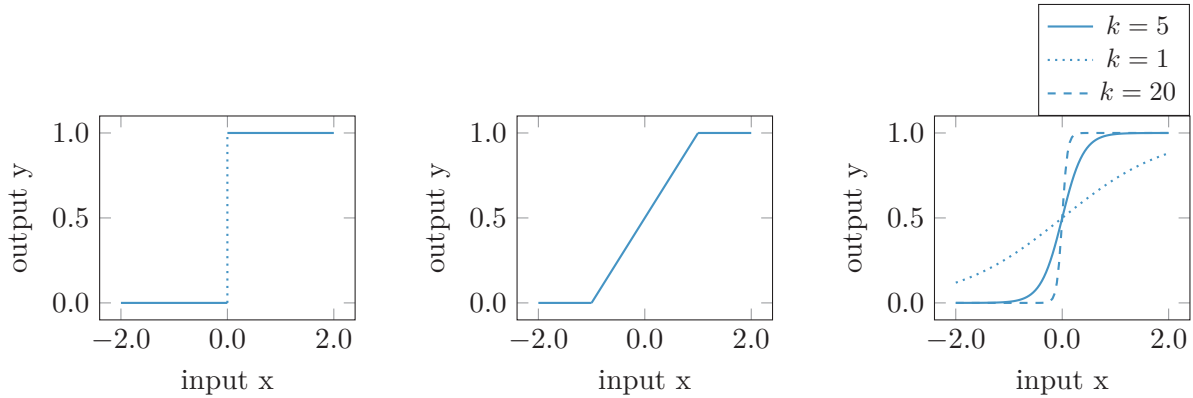


Figure 6.1: The structure of a single neuron: Each input is weighted by w_i . The output of a neuron is also influenced by a bias value b , and an activation function $g(\cdot)$.



(a) Step function (eq. (6.1)) where $k = 0$ (b) Linear function where $k_{\min} = -1$ and $k_{\max} = 1$ (eq. (6.2)) (c) Sigmoid function defined by eq. (6.3)

Figure 6.2: Common activation functions for a single artificial neuron [80]: The step function has a jump discontinuity, and is therefore not continuously differentiable (a). The same applies to the linear function, where the ascending points cause an issue for the differentiability (b). In contrast to that, the sigmoid function is continuously differentiable.

not $u = \mathbf{false}$. Nevertheless, the threshold function is not differentiable due to the jump discontinuity. Noise on the input close to the jump discontinuity will forward this signal directly to the output, causing an unstable system.

- **Linear function:** Here, a linear combination describes the output of the neuron. Often, the slope of the linear function can be adapted depending on the point of ascent k_{\min} and k_{\max} . The slope of the linear function between k_{\min} and k_{\max} is therefore defined by $k = 1/(k_{\max} - k_{\min})$ and the output of the activation function is described by

$$g(x) = \begin{cases} 0 & \text{where } x \leq k_{\min} \\ k \cdot x & \\ 1 & \text{where } x > k_{\max} \end{cases} . \quad (6.2)$$

Figure 6.2b illustrates a linear activation function, with the point of ascent at $k_{\min} = -1$ and $k_{\max} = 1$. The slope of the linear function in the example is therefore $k = 0.25$.

- **Sigmoid function:** The sigmoid function can be applied to all kind of non-linear problems. It is defined by

$$g(x) = \text{sig}(x) = \frac{1}{1 + e^{-k \cdot x}} \quad 0 \leq g(x) \leq 1 \quad , \quad (6.3)$$

while the parameter k adjusts the slope of the sigmoid function. Around zero, the sigmoid function is close to being linear. For high values of $k \rightarrow \infty$, the sigmoid function approximates towards the step function. In contrast to that, for $k \rightarrow 0$, the sigmoid function converges towards the linear function. Figure 6.2c shows the sigmoid function. The function

is strictly increasing and therefore differentiable. Furthermore, the sigmoid function is easy to differentiate by

$$\frac{\partial}{\partial x} \text{sig}(x) = \text{sig}(x)(1 - \text{sig}(x)) \quad .$$

The differentiability is a significant advantage for learning. Therefore, the sigmoid function is often applied to neural networks.

The choice of the activation function is determined by the nature of the problem given, the available data, and its distribution. A linear problem can be solved with a linear function as activation function. For a non-linear problem, a non-linear activation function should be selected. For the experiments with body weight estimation in the upcoming section, all networks have a sigmoid function as activation function for all layers.

6.2 Net Architecture

The here described approach for body weight estimation is classic machine learning with regression of an unknown function $f(\mathbf{x})$. Therefore, the function should provide a value, close to the real body weight, depending on the forwarded feature vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$. Neural networks exist in various shapes and structures. However, the here presented ANN is based on the multilayer perceptron, which is common for a non-linear regression task [199]. In a multilayer perceptron,

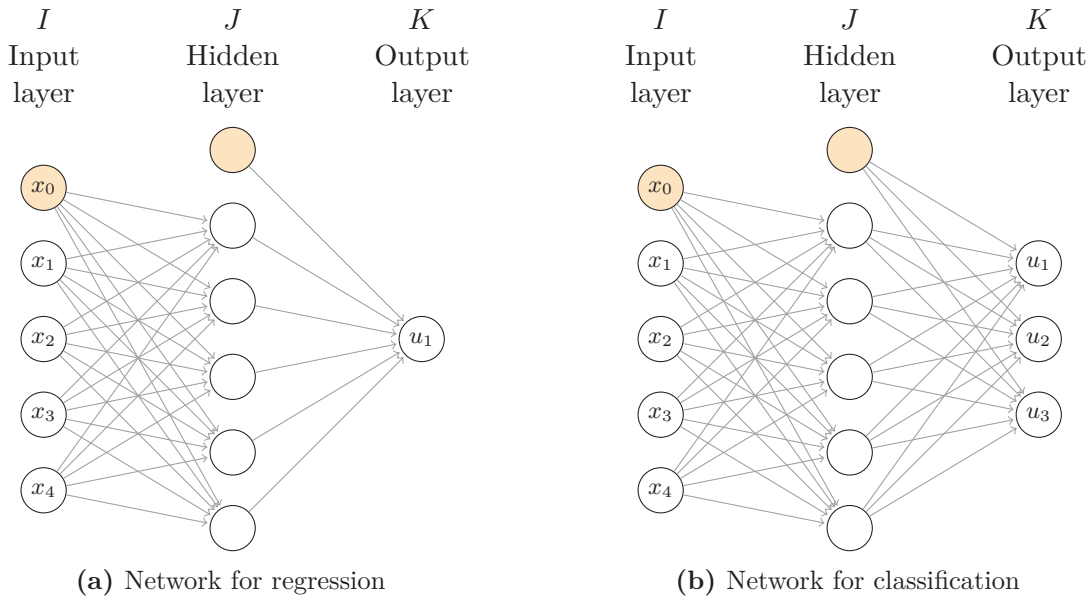


Figure 6.3: Network architectures for regression and classification: Both networks consist of an input layer. The size of this layer is defined by the number of features forwarded to the network. A hidden layer allows the network to fit non-linear functions. A bias neuron – here marked in orange – in the input and the hidden layer helps to improve generalization. The output layer shows the difference between regression and classification: For regression, a single neuron in the output layer is necessary (a), while a network for regression has more than one output neuron (b); one for each class – here three classes.

each neuron of a single layer is connected to all neurons of the following layer. Between the neurons of a single layer, no connections are established.

Figure 6.3 shows the structure of the network, which is used for experiments:

- The first layer is used to feed the data into the network. The number of neurons in this layer is given by the size of the feature vector forwarded to the ANN. This layer is also called input layer. In this layer, no calculation based on a weighting is performed. For equations in this section, the index i is used to refer to the input layer.
- The next layer is the hidden layer: An arbitrary neuron in this layer is connected to all neurons in the input layer. An ANN can have an arbitrary number of hidden layers. For a non-linear regression problem, the network should contain at least one hidden layer. Furthermore, the number of neurons for each hidden layer can be adapted. The hidden layer is essential to solve non-linear problems [23]. Also, the hidden layers can contain an additional bias value, which is equal for all neurons in a single layer. This layer is referred with the index j .
- Finally, the ANN has an output layer. Just as shown for the previous layers, all neurons in the output layer are connected to all neurons in the previous layer. Also in the output layer, a bias value can be applied. For a regression of the here presented body weight estimation process, the output layers consist of a single neuron, with its output value is representing the body weight. For equations in this section, the index k is used to refer to the output layer.

The difference between classification and regression can also be found in the network's architecture: While the regression of a two-dimensional function demands a single neuron in the output layer, a network for classification contains one neuron for each class to estimate. Each neuron provides a probability that the output is the searched class. With a value of one, the network is entirely sure that the forwarded input belongs to one class. The ANN-based classification is used in the experiment section to classify the BMI of subjects in a medical scenario, see section 7.7 on page 145.

For each neuron-to-neuron connection between the layers, a weight w is applied. For the upcoming equations, each neuron is marked by an index, depending on the layer. The i -th neuron layer is the input layer $i \in I$, the j -th layer is the hidden layer $j \in J$, and k represents the output layer $k \in K$. Two indices describe each weight in the network: For typographical convention, the first index illustrates the starting of the link, while the second index indicates the ending of an arbitrary link, e.g., w_{ij} describing the weight between a neuron in the input and the hidden layer.

6.3 Forward Propagation

The explanations in this section are taken from the work by Bishop [23]. The number of neurons in the input layer is described by the variable I , the number in the hidden layer with J , and the number in the output layer with K .

For the forward propagation, a set of values $\mathbf{x} = (x_1, \dots, x_I)$ is passed to the network's input. Forwarding values from the input of the network and getting values at the output is called

forward propagation. For the forward propagation, the weights do not change in this step and keep its values. The change of the weights is done via backpropagation and is described in the upcoming section.

The weights between the input and hidden layer are defined by $w_{ij}^{(1)}$, with the superscript (1) illustrating that the corresponding parameters are based on the first layer of the ANN. The input of an arbitrary neuron a_j in the hidden layer is therefore described by

$$a_j = \sum_{i=1}^I w_{ij}^{(1)} x_i + w_{j0}^{(1)} \quad ,$$

where $j = 1, \dots, J$, and J being the number of neurons in the hidden layer [23]. The bias value for this layer is represented by $w_{j0}^{(1)}$. To get the output of the j -th neuron, the activation function $g_j(\cdot)$ is applied by,

$$u_j = g_j(a_j) \quad .$$

For the output layer, the equation for an arbitrary neuron is formed similar by

$$a_k = \sum_{j=1}^J w_{jk}^{(2)} x_j + w_{k0}^{(2)} \quad ,$$

where $k = 1, \dots, K$, and K is the number of neurons in the output layer[23]. Also here, $w_{k0}^{(2)}$ defines the bias value for the output layer. Also in the output layer, an activation function is applied $g_k(\cdot)$. The output of the k -th neuron u_k is therefore defined by

$$u_k = g_k(a_k) \quad .$$

Finally, the output of the last neuron can be described by a single equation, dependent from the forwarded input vector \mathbf{x} , as well as the weights between the neurons in the network \mathbf{w} . The combination of the three layers is now defined by

$$u_k(\mathbf{x}, \mathbf{w}) = g_k \left(\sum_{j=1}^J w_{jk}^{(2)} g_i \left(\sum_{i=1}^I w_{ij}^{(1)} x_i + w_{j0}^{(1)} \right) + w_{k0}^{(2)} \right) \quad ,$$

where all weights and bias parameters are grouped into a vector \mathbf{w} [23]. Additionally, the equation can be simplified, when the bias values are absorbed into an additional input parameter x_0 , whose value is set constant to $x_0 = 1$. The output of the network $u_k(\mathbf{x}, \mathbf{w})$ is therefore calculated by

$$u_k(\mathbf{x}, \mathbf{w}) = g_k \left(\sum_{j=0}^J w_{jk}^{(2)} g_i \left(\sum_{i=0}^I w_{ij}^{(1)} x_i \right) \right) \quad .$$

Note, that the indices of the sum sign now starts with zero [23]. While this equation represents a network with a single hidden layer, it can be expanded towards an arbitrary number of hidden layers.

The forwarding of the values and the calculation of the network's output can be done efficiently. Therefore the calculation is suitable for real-time applications. In contrast to the previously presented segmentation, the calculation of the output's value is done in a fraction of the time.

6.4 Learning

This section explains how the connections between the neurons and its weights are calculated to achieve the best performance with an ANN. For the forward propagation, it is assumed that the weights in the ANN already have values, so the network can provide a good estimation close to the ground truth value. In this section, the finding of these weights is described via backpropagation of errors, also called training. Receptively values are forwarded to the network, the output of the network is compared to the ground truth value, and an error is calculated.

In advance of the training, the network is initialized randomly with values for the weights close to zero. Together with the randomized set and order of the training data set, the outcome of the finally trained network can differ with every new training.

The goal of a gradient-descent network is to find the global minimum. Changing the weights of the network might reach a local minimum, and the optimization terminates. First, to prevent this behavior, such a local minimum has to be detected. Second, a recovery behavior must exist to overcome such a local minimum. To avoid the trap of local minima, an adaptive learning rate can be added to the process of learning [23]. The maximum learning rate is defined at the startup with $\eta(0)$. Having a high learning rate in the beginning can help to descent the backpropagation error quickly, jumping over local minima. However, having a high learning rate close to the global minimum can lead to overshooting that minimum. Therefore the learning rate η is adapted and decreased over the number of training epochs n by

$$\eta(n) = \eta(0)e^{-\alpha n} \quad ,$$

where α is a constant factor to adapt the slope of the reduction of the learning rate [109]. An epoch describes how many times the algorithm sees the entire data set for training. Figure 6.4 demonstrates the behavior of the same network with different parameters for the learning rate.

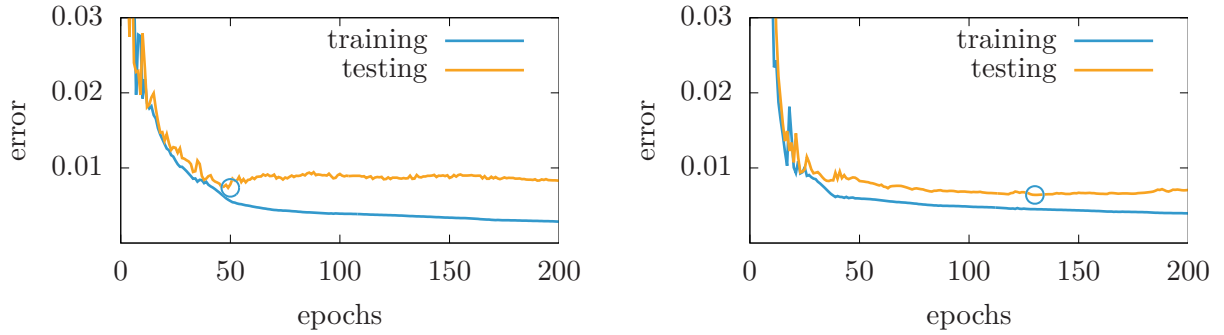
Additionally, to improve learning even more, a momentum can be added: If the weights are receptively changed in one direction – positive or negative – the momentum is added to increase the change for backpropagation [23, 119].

Essential for the backpropagation is the error rate: Suppose a set of features is forwarded to the network's inputs. In case of regression, the output of the network for every set in the training data $\mathbf{u} = (u_1, \dots, u_n)$ is compared with the target vector $\mathbf{t} = (t_1, \dots, t_n)$ [109]. A sum-of-squares error function E for the output of the neural network u_k is applied by

$$E = \frac{1}{2} \sum_{i=1}^n (u_i(\mathbf{x}, \mathbf{w}) - t_i)^2 \quad ,$$

where n is the size of a data set for training [23]. Calculating the squared error leads to an equal result for positive or relative errors. Moreover, errors are weighted non-linearly, with high errors being weighted more than small errors. The change of the error with respect to the given connective weight has to be calculated, so it can be minimized. The derivation of the equation can be found in Bishop [23]. The derivative of the error in the output layer is calculated by

$$\frac{\partial E}{\partial w_{jk}} = u_j \cdot u_k(1 - u_k)(u_k - t_k) \quad ,$$



(a) Mean square error in the ANN with a learning rate of 1×10^{-3} : The minimum for the testing error of 7.3×10^{-3} is reached in the 50th epoch.

(b) Mean square error in the ANN with a learning rate of 1×10^{-5} . The minimum for the testing error of 6.4×10^{-3} is reached in the 130th epoch.

Figure 6.4: Comparison of two networks having different learning rates: The network with the higher learning rate reaches a minimum earlier after the 50th epoch (a). While the second network has a lower learning rate, 130 epochs are necessary to reach a minimum (b). However, this minimum is lower than the minimum reached with a higher learning rate. Additionally, the network with the lower learning rate has less noise on the error function of the testing data.

where w_{jk} is an arbitrary weight for a connection between a neuron of the hidden and the output layer. The term can be rewritten providing a variable δ_k for the equation by

$$\frac{\partial E}{\partial w_{jk}} = u_j \cdot \delta_k \quad \text{where } \delta_k = u_k(1 - u_k)(u_k - t_k) \quad ,$$

while the term for δ_k is often referred to the responsibility of a neuron – here in the output layer.

The same equations are applied for the hidden layer – which is referred by the index j . For this layer, the error is not visible, because the target value t only exists for the output layer. The derived error is therefore calculated based on the output by

$$\frac{\partial E}{\partial w_{ij}} = u_i \cdot u_j(1 - u_j) \sum_{k \in K} (u_k - t_k) u_k(1 - u_k) w_{jk} \quad .$$

Also for the hidden layer, a responsibility is defined by

$$\frac{\partial E}{\partial w_{ij}} = u_i \cdot \delta_j \quad \text{where } \delta_j = u_j(1 - u_j) \sum_{k \in K} \delta_k w_{jk} \quad .$$

Now the current weights for the connections between the layers can be changed, depending on the learning rate η . The weights in the output layer are changed by

$$w_{jk} = w_{jk} + \eta \delta_k u_k \quad ,$$

and finally, the weights in between the hidden and the output layer w_{jk} are changed by

$$w_{ij} = w_{ij} + \eta \delta_j u_j \quad .$$

The algorithm for backpropagation terminates with a maximum number of epochs, or a given bound of the error E for the testing data is reached [119].

6.5 Issues in Training of Neural Networks

The design of a neural network and the backpropagation are straightforward. However, applying neural networks for regression demands the knowledge about possible issues.

6.5.1 Scaling of Feature Values

The values in the input vector can have every arbitrary scalar value, positive or negative. Looking back to the previously extracted features, all values are positive. When all elements of the input vector are positive, the weights of the neurons will decrease in the same epoch. Especially, if the step function is applied as an activation function, this would lead to a zigzagging behavior, which is not efficient and would slow down learning. To prevent this win of big values in an input vector \mathbf{x} , the values in the feature set have to be standardized [92]. The feature vector is standardized by

$$x_{\text{std}}(\mathbf{x}) = \frac{x - \bar{\mathbf{x}}}{\sigma(\mathbf{x})} \quad ,$$

while the variance of the set of feature $\sigma(\mathbf{x})$ is described by

$$\sigma(\mathbf{x}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{\mathbf{x}})^2} \quad \text{where } x_i \in \mathbf{x} \quad ,$$

where n is the size of the feature vector and $\bar{\mathbf{x}}$ its mean value. The standardization transforms a set of data to an expected value where $\bar{\mathbf{x}} = 0$ and a standard deviation $\sigma(\mathbf{x}) = 1$. In case, the standardized value should be transformed back, for example, the output value of the last value is applied for a function regression, the de-standardization is achieved by

$$x(\mathbf{x}) = \sigma(\mathbf{x}) \cdot x_{\text{std}} + \bar{\mathbf{x}} \quad .$$

Figure 6.5 shows the results from standardization of the feature values for the features f_1 to f_{12} . The standardized feature sets all have an expected value of zero and an equal standard deviation of one.

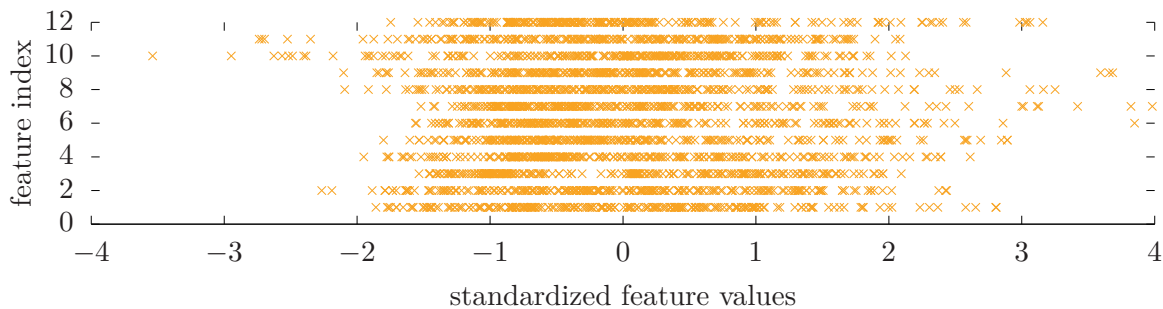


Figure 6.5: Standardization of the feature vector $\mathbf{f} = (f_1, \dots, f_{12})$.

6.5.2 Overfitting

The only way to affect the output of an ANN is the change of the weights between the neurons. With machine learning these weights are changed to achieve a small error for a single data set between the target and the output vector. Furthermore, having a good solution will lead to a small error for the complete training data set. One problem in machine learning is overfitting: In this case, the change of the weights in the network adapts ideally to the training data set. Forwarding new and unknown data to the network will lead to a significant error in the output.

One common way to prevent overfitting is the separation of the available data into two groups:

- A training data set is used to train the neural network. It should consist of 50 to 70 percent of the complete data set. With a high amount of overall data, the size of the training set can be increased.
- A validation data set is used to observe the progress in training. This set is never used for the backpropagation of the network. This set ensures that the network is not overfitting and stays generalized to possible future data. It is important that both sets contain a good mix of all possible feature values, containing minima and maxima of a set.

Depending on the overall size of the available data for testing and training, the ratio between testing and training can be adapted. Applying a low ratio of training data ensures a good generalization, and the network will likely provide good results for unknown new forwarded features. To make sure that the learning does not rely on a certain sequence of the forwarded training data, the complete data set should be shuffled in advance [23].

Figure 6.6 illustrates different results from training: In 6.6a a model with underfitting is presented. The approximated function has a low degree and therefore most of the points are not close to the function. However, this model provides a low variance. In contrast to that, 6.6c illustrates an overfitted model. Here, the points align good with the estimated function, leading to a low bias. Nevertheless, the function has a high variance, which is an indicator for overfitting. The plot in the middle – Figure 6.6b provides a model with low variance and low bias. The relation between the bias and the variance is illustrated in Figure 6.7.

6.5.3 Number of Neurons in the Hidden Layer

Neural networks are suitable to find the optimal solution if one exists [115]. However, the solution depends on the number of layers, and especially the number of neurons in a hidden layer. Having a network with a low number of hidden units leads to great generalization, but might deal poor for an overall result. In contrast to that, a network with a high amount of neurons in the hidden layer will adapt fast and accurate to the provided training data set. However, such a network would perform poorly to never-seen-before data. Applying a fixed number of neurons is therefore based on several trials and evaluation of the results. However, a rule of thumb exists to start training a multilayer perceptron with some hidden units in the range between the number of output neurons to the number of input neurons [123]. Nevertheless, it is hard to develop a good number of neurons for the hidden layer from the beginning. It is best to increase the number of neurons in the hidden layer while training [23]. Figure 6.8 shows the error in training and testing for an increasing size of units in the hidden layer.

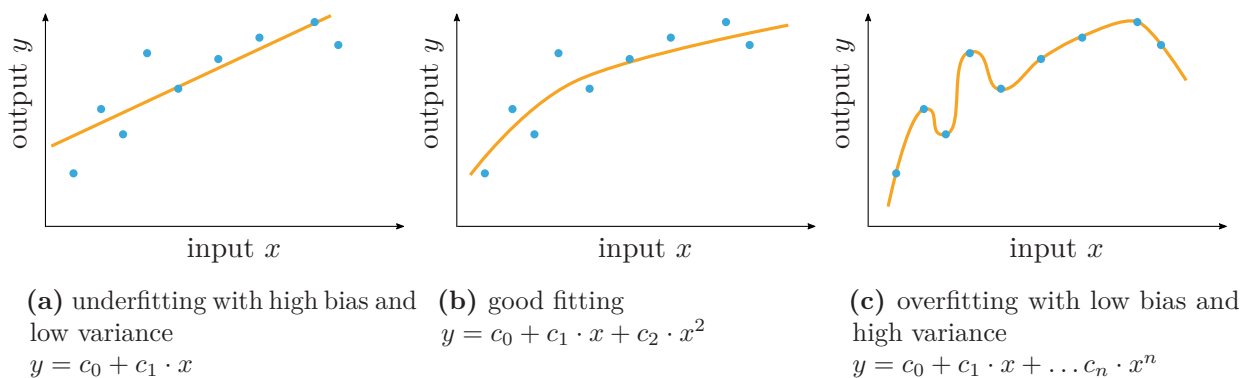


Figure 6.6: Function regression with three different fittings: Underfitting can appear if the structure of the applied network is simple, e.g., does not have a sufficiently high number of neurons or layers to adapt to the underlying function. With a proper fitting, the estimated function is close to the points (b). With a overfitting, the network loses its capability for generalization: The regression for data used for training fits perfectly, while the network provides poor output for new and never seen data (c). Source: Bishop [23]

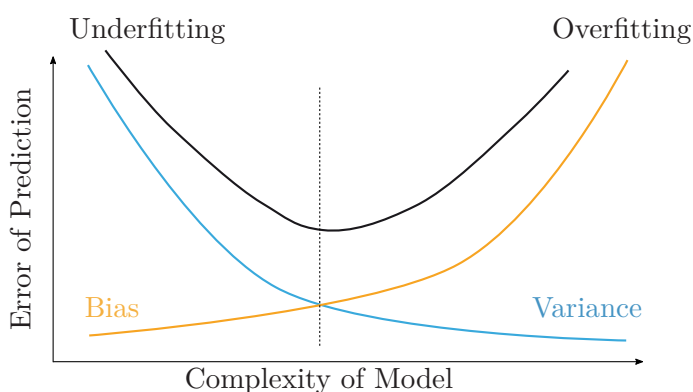


Figure 6.7: Relation between bias and variance: A neural network having a low complexity cannot provide an ideal regression, which leads to an underfitting. When the complexity is increased by adding hidden layers or neurons, the total error tends to lower, while the variance in output of the network increases. Increasing the complexity further, the network will lose the ability to adapt to generalized functions and the ANN tends to overfit. Source: [53, 227]

With wrong parameters, neural networks will likely lead to overfitting the data. If the network contains too many neurons, the network will adapt fast and accurate to the training data set but will fail at other data, which the network has never seen before.

The data sets for training are often chosen randomly. However, such a random set might not be ideal for training, because data sets are too similar: A network trained with persons between 50 to 70 kg can hardly estimate someone weighing 90 kg. Not only the ground truth value of the output of the neural network should have a maximum range; this counts for all features forwarded to the input of the ANN.

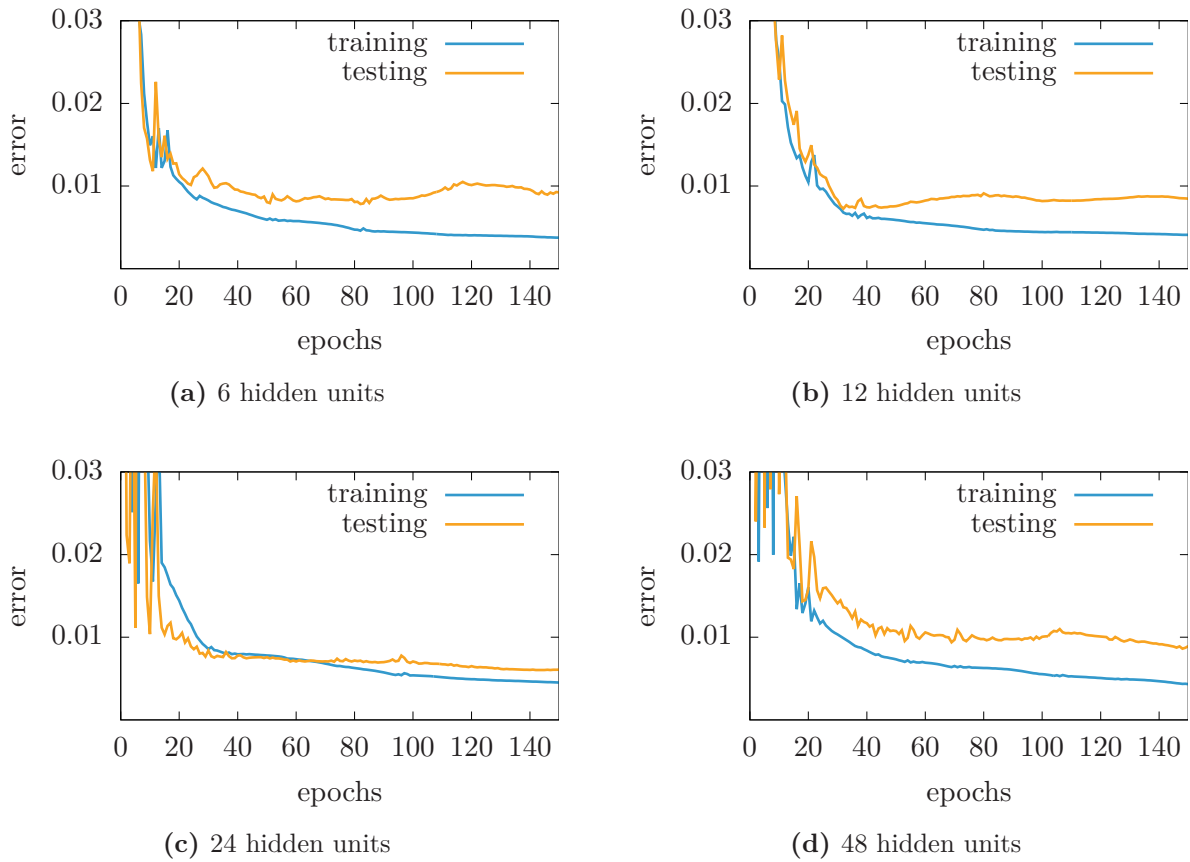


Figure 6.8: Training of networks with increasing amount of units in the hidden layer: The overall error for the testing data set can be decreased with a higher amount of hidden units. However, this can lead to overfitting. In (a), a low number of neurons in the hidden layer leads to an underfitted network, which can be seen due to the gap between training and testing error. With additional neurons (b), this difference can be reduced. Having a high number of neurons in the hidden layer can lead to a zigzagging behavior at the beginning of the training (c-d).

6.6 Summary

Supervised learning is applied for the estimation of the body weight. The previously extracted features are forwarded to a 3-layer neural network. The network's structure with a single hidden layer and a low amount of neurons in this layer provides enough variance and a low bias for a generalized regression. Furthermore, over- and underfitting is prevented by separating the data sets for training and testing. In the upcoming section, the here presented ANN is used to achieve the following results. The network is designed with three layers, one input, one hidden and one output layer. All neurons have a sigmoid activation function and are initialized with random weights close to zero. The hidden layer contains 12 neurons, for a sufficient generalization.

Chapter 7

Experiments and Results

This chapter presents different settings of experiments, as well as their results. The results for contact-less body weight estimation are discussed and compared to other state-of-the-art methods based on anthropometric estimations, as shown in section 2. Furthermore, the section contains an approach for the estimation and classification of BMI, which could be used to minimize the radiation dose for CT examinations.

7.1 Data for Testing and Validation

During this thesis, several subjects were estimated by this approach. All measurements were collected and saved in a MySQL database to be used for training and optimization of the algorithm afterward. The statistics about the different data sets are summarized in Table 7.1. The data sets are the following:

- **Hospital Data Set (H-DS):** From May to September 2014, this data set was recorded in a trauma room of the Universitätsklinikum Erlangen, Germany, for preliminary testing. In this data set, only RGB-D data from a Microsoft Kinect is available, without thermal data. The recorded scenes were used to evaluate the different approaches for segmentation. The thermal camera was added after these experiments. The data set does not only contain people with stroke; independent of the visible symptoms or the diagnoses, the subjects are added to the data set if they are treated in the trauma room with the sensor system in the ceiling.
- **Hospital with Thermal - Data Set (HT-DS):** This data set contains feature values from trauma room patients from the Universitätsklinikum Erlangen, Germany. The data set contains 133 measurements from people lying on a medical stretcher, recorded with a Microsoft Kinect. The subjects were recorded between February and July 2015. For this data set, a good distribution is achieved by having people of different ages, body weights and shapes, see Table 7.1. Additionally, this data set contains the patient's self-estimation of his or her own body weight, age, gender, as well as anthropometric features like the body height, abdominal girth, and waist circumference. The distance between the sensors and the subjects was around 2.3 m. In both scenes from the hospital, the medical stretcher

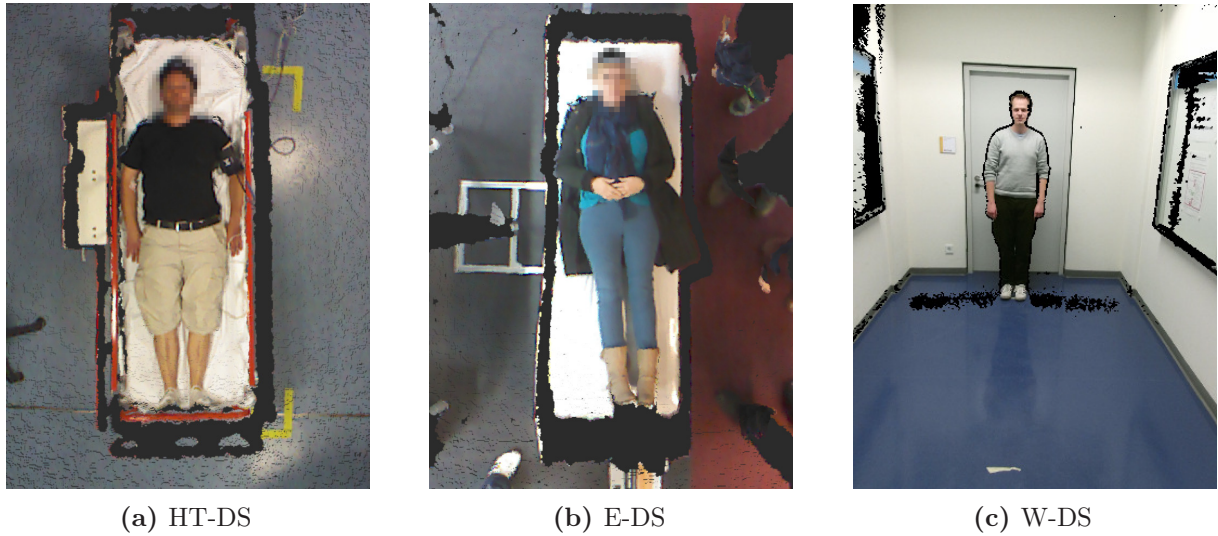


Figure 7.1: Scenes from three different data sets: The first data sets (a-b) contain subjects on a stretcher seen from above. The hospital data set provides people from the trauma room. The data from the event (E-DS) contains people wearing jackets, which add an additional challenge for the estimation. The third data set contains people walking towards the camera along a hallway (c).

was free to move in a certain range, illustrated with markers on the floor. Furthermore, in some scenes, several physicians are in the FOV as well as medical devices close to the patient, necessary for diagnosing and treatment.

- **Event - Data Set (E-DS):** The features from this data set were recorded at a public event, called Long Night of Science in October 2015 in Nuremberg, Germany. People in this data set were visitors of the public event. This data set contains 106 people. For this public event, it was not convenient to take anthropometric measurements. Ground truth was validated with a standard digital scale, after the estimation was performed. The data set consists of sensor values from a Kinect and a thermal camera. Additionally, this data set includes point clouds from Microsoft Kinect One. The stretcher has a fixed position in relation to the sensor system.
- **Walking - Data Set (W-DS):** Based on the results of the previous three data sets, experiments with people standing and walking in front of the camera are complemented. The data set consists of 15 people, mostly from the Nuremberg Campus of Technology. The people were placed in front of the camera, performing a set of movements, e.g., walking towards the camera, walking away from it or turning on the spot. The experiments with the recording were performed in January and February 2017. The recording of the sensor streams was achieved via the middleware Robot Operating System (ROS) [173] and afterwards converted to a set of Point Cloud Data (PCD) files.

Figure 7.1 shows characteristic scenes from the available data sets. The first three data sets are available via the Open Science Foundation www.osf.io [67, 159]. The data sets only contain the depth and thermal information, to preserve the anonymity of the subjects in the FOV. Each

Table 7.1: Comparison of the applied data sets: The table contains the statistical values of the ground truth body weight, the body size, as well as the distribution in sex. The minimum weight for the E-DS comes from some children who tried the *Libra3D* system at a public event. The W-DS contains only a small amount of subjects. For comparison, the last row contains the average data of body size and weight from the German population in 2009, published by the German Federal Office for statistics [55]. The average values for the first three data sets are close to those average values.

Data Set	Real Weight in kg				Body Size in m				Gender	
	min	max	mean	σ^2	min	max	mean	σ^2	female	male
H-DS	48.8	165	78.3	17.2	1.45	1.97	1.71	9.8	154	149
HT-DS	48.6	129	77.8	17.1	1.45	1.94	1.70	9.8	77	56
E-DS	26.8	114	74.7	14.8	–	–	–	–	28	68
W-DS	68	134	84.2	16.4	–	–	–	–		15
German Pop. [55]			73.9				1.72		41.7 M	40.1 M

measurement is already segmented by the previously presented algorithms, so the cloud data from a subject can directly be used for weight estimation approaches. The file name of a single PCD file contains the gender, the ground truth body weight, as well as a unique ID.

The upcoming experiments are performed with three out of the four data sets. The H-DS data set was used to evaluate the different approaches for segmentation, in an early stage of the project *Libra3D*. Because the patient is close to the stretcher, the patient’s clothes do not always provide a sufficient contrast for color segmentation, the thermal camera is necessary for a reliable segmentation of the patient and the stretcher.

Figure 7.2 illustrates the distribution in ground truth body weight and other anthropometric values. To evaluate different estimation methods, it is crucial to record the ground truth body weight for comparison. Furthermore, the gender and age is recorded because of the correlation between those two properties with body weight, as shown by Heymsfield et al. [84]. Additionally, body height in combination with the ground truth body weight gives a clue if someone is underweight, normal weight, overweight, or obese by the Body Mass Index (BMI). The BMI is defined by the body mass m in kilograms divided by the body height h in meters with

$$\text{BMI} = \frac{m}{h^2} \quad \text{with} \quad \begin{cases} \text{BMI} < 18.5 \rightarrow \text{underweight} \\ 18.5 < \text{BMI} \leq 25 \rightarrow \text{normal healthy weight} \\ 25.0 < \text{BMI} \rightarrow \text{overweight} \end{cases} .$$

The boundary values differ for different regions and ethnic groups around the world. Here, the data from the World Health Organization (WHO) is presented [15]. Often, the significance of the BMI in the context of obesity is discussed [128]. The data set presents a broad distribution for different constitutional types: The lowest BMI was recorded at 17.8 kg/m^2 . On the other side, a maximum value was set with a value of 46.6 kg/m^2 . No data set contains a subject with an amputation. Therefore the scenario of the estimation of a subject with amputations is not evaluated in this thesis.

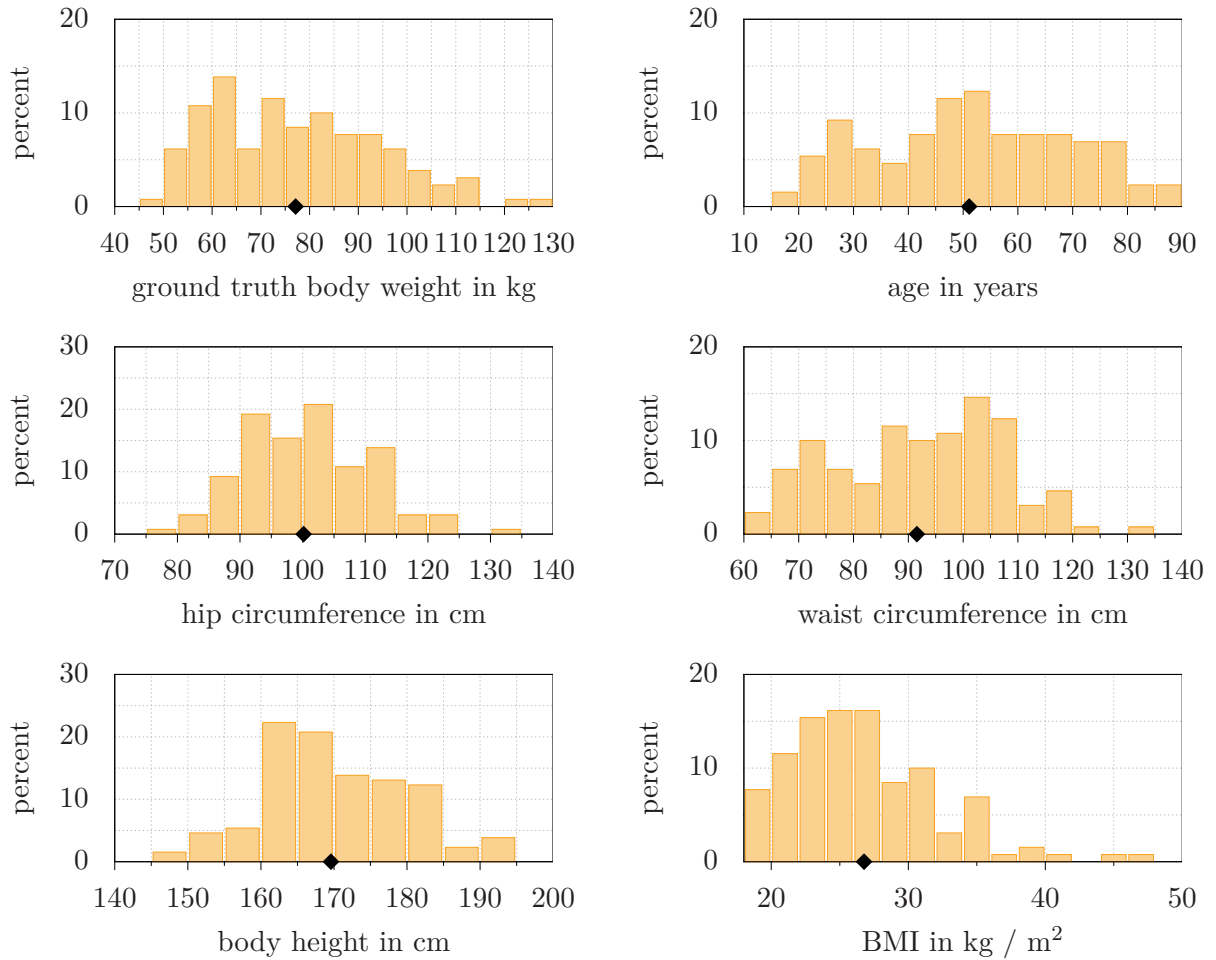


Figure 7.2: Distribution of different features from the HT-DS data set recorded in the trauma room: The black diamond on the x-axis marks the mean value for each histogram. In contrast to the other recorded data sets, here additional anthropometric features like the waist circumference or the age were recorded.

The waist and hip circumference are recorded for comparison with other anthropometric based estimation methods, like presented by Lorenz et al. [124].

7.2 Setting for Validation

The outcome of each performed estimation is important for the patient himself or herself, irrespective if the estimation is performed by a physician or by the *Libra3D* system. The goal of the system is, that it outperforms the estimation of the physicians over a set of patients. However, it can be, that individual subjects receive a better estimation from a physician. For the upcoming experiments, the defined criterias for evaluation and measurement of performance are discussed.

First, the tests for the evaluation of a single measurement are presented: Directly visible is the absolute error e for an arbitrary body weight estimation, having the ground truth value \hat{x} as well as the estimated value \tilde{x} , which is defined by a subtraction with

$$e = \tilde{x} - \hat{x} \quad .$$

The absolute error would be good to compare the estimation of a group of people having the same body weight and only differing in their visual appearance. However, the here presented group of people applied for testing has a high variety in body weight, and also visual appearance. Therefore the absolute error is not sufficient for comparison. Additionally, when the focus is set on the dosing of drugs, the relative error ϵ is applied to evaluate the correct dose. Furthermore, the relative error provides a better chance to compare the estimation over a bigger group of people with different body weights. In general, the calculation of the relative error is based on the absolute error e and the division of the ground truth value,

$$\epsilon = \frac{\tilde{x} - \hat{x}}{\hat{x}} = \frac{e}{\hat{x}} \quad .$$

In case the relative error has a positive sign, the estimated value is higher than the ground truth value. An example: A physician performs two estimations, one for a subject weighing 100 kg and one weighing 50 kg. In both cases, the physician overestimates 10 kg. For the heavier person, a relative error has a value of 10 percent, which would be all right for the treatment of stroke patients. In contrast to that, the lighter subject is estimated with a resulting relative error of 20 percent; not reliable for the dosing of stroke patients with tPA.

For the evaluation of a group of subjects, the relative error can be a good indicator: A good body weight estimation method provides a narrow area for the relative error. Therefore, the minimum relative error ϵ_{\min} and maximum relative error ϵ_{\max} are recorded for experiments. Furthermore, the mean average error $\bar{\epsilon}$ is investigated to proof a low bias for all data sets. Ideally, the applied model for estimation should have a low bias, and a low mean average error.

Another way to proof and benchmark the body weight estimation approach is the Mean Absolute Error (MAE). The absolute error of each data set e_i is summed up and divided with the total number of data sets n for benchmarking. It is defined by

$$e_{\text{mea}} = \frac{1}{n} \sum_{i=1}^n |e_i| \quad \text{with} \quad e_i = \hat{m}_i - \tilde{m}_i \quad ,$$

where \hat{m}_i is the ground truth body weight and \tilde{m}_i is the estimated weight of a subject. Moreover, the mean square error can be used for validation. Here the absolute error is squared before summation. It is defined by

$$e_{\text{mse}} = \frac{1}{n} \sum_{i=1}^n e_i^2 \quad \text{with} \quad e_i = \hat{m}_i - \tilde{m}_i \quad .$$

Compared to the mean absolute error, outliers were weighted stronger, due to the square factor. Especially for the medical use of the here presented algorithm, it should be evaluated how many subjects are estimated within a certain range of relative error. For the dosing with tPA for

ischemic strokes, the FDA prescribes a maximum deviation of ± 10 percent, as illustrated in the introduction. Additionally to that range, the proportion of subjects within 5, 10 and 20 percent are compared. Most experiments provide a cumulative plot about the relative error. The proportion of a certain range can be read off easily.

7.3 Evaluation from different Feature Assemblies

Depending on the scenario – a patient lying on a stretcher, or a subject walking in front of a camera – not all features can be calculated, as previously mentioned. Obviously, if all features can be calculated, the ANN-based model should provide the best estimation for the set of subjects. However, the calculation of all features might not be necessary, if only a rougher estimation is needed. This section presents the estimation based on different groups of features forwarded to the ANN. Table 7.3 shows the different assemblies of the features for the different experiments. The groups are the same for training and validation. In all experiments, 30 percent of the data is used for training and 70 percent are used for validation. The result of the here performed experiments with the different feature groups are presented in scatter plots in Figure 7.3 on page 133. Furthermore, Figure 7.4 presents the estimation in a cumulative plot, and Table 7.3 illustrates the raw error values from this section. For these experiments, the data set from the hospital with thermal data and the event data set are used together.

Experiment 1: Estimation based on Volume

For the first experiment in this section, only the volume is taken to estimate someone’s body weight. This approach was previously presented by Pfitzner et al. [164] due to the fact that there is a strong correlation between body weight and the volume, see Figure 5.13. Moreover, the density of a human body varies only in a small range. Figure 7.3a illustrates the results in a cumulative plot as well as a scatter plot. The range of outliers for the relative error spans an area of 38 percent between -22.9 percent and 26.8 percent. These outliers are best visible in the scatter

Table 7.2: Survey about the following experiments with different configurations for the forwarded features.

Experiment	1	2	3	4	5	6
Volume	✓	✓	✓	✓		✓
Surface		✓	✓	✓	✓	✓
Number of Points			✓	✓	✓	✓
Density			✓	✓	✓	✓
Eigenvalues			✓	✓	✓	✓
Features from Eigenvalues				✓	✓	✓
Features from Statistics				✓	✓	✓
Features from Contour					✓	✓
Gender					✓	✓
Features from Temperature					✓	✓

plot. The absolute relative error has a value of 0.21 percent. Besides, the distribution of the relative errors has a standard deviation of 7.6 percent. Having a look for the in-range-percentage, 50.6 percent of all data sets are estimated within a relative error of 5 percent. Looking for the 10 percent threshold in relative error, 83.7 percent are within this range. For a relative error below 20 percent, 99.1 percent of all measurements are included. In comparison to the estimation provided by the physicians – which estimate around two-third correctly – this estimation is already better [31, 163].

Experiment 2: Estimation based on Volume and Surface

Additionally, besides the volume, the surface is added as an input feature for this experiment. As shown in Figure 5.13, the surface also has a high positive correlation to the body weight. Figure 7.3b illustrates the results as cumulative error plot and scatter plot. Compared to the previous experiment with just the volume, the range of outliers is reduced, spanning now from -20.3 percent to 25.3 percent. The standard deviation is also reduced to 7.3 percent. However, the average relative error is slightly higher with 0.28 percent. The cumulative error is similar to the previous result: For the 5 percent margin, 51.1 percent are valid. Moreover, 83.3 percent of all measured data are included within the 10 percent margin. The cumulative error for 20 percent stays the same with 99.1 percent. Comparing this experiment with just the volume estimation, it is similar in results.

Experiment 3: Adding Features from Eigenvalues

Here, the features from eigenvalues are added to be forwarded to the neural network. Figure 7.3c demonstrates the results as cumulative error plot and scatter plot. The range of outliers is reduced from -19.0 to 19.8 percent, spanning an area of 29.7 percent. This experiment has the lowest bias over all measures: The average relative error is the best in all experiments, having a value of 0.07 percent. This is the first experiment to reach all estimations with an error below 20 percent. Compared to both previous experiments, all rating values are better. Therefore, adding values from eigenvalues can lead to an improved body weight estimation. Figure 7.3c illustrates the results.

Experiment 4: Adding Features from Statistics

For this experiment values from statistics are added. The minimum relative error was recorded with a value of -14.0 percent while the maximum was set at 14.3 percent, spanning a range of 28.3 percent. The average relative error is listed with 0.28 percent. Compared to all previous experiments, the standard deviation is slightly lower, having a value of 5.5 percent. For the 5 percent range, 63.9 percent were below this threshold. Looking for the 10 percent range, 91.4 percent reached this boundary level. All estimations were below the relative error of 20 percent. Therefore, this experiment demonstrates that adding features from statistics, the estimation can be improved. Figure 7.3d illustrates the results as cumulative error plot and scatter plot.

Experiment 5: Leaving out the Volume

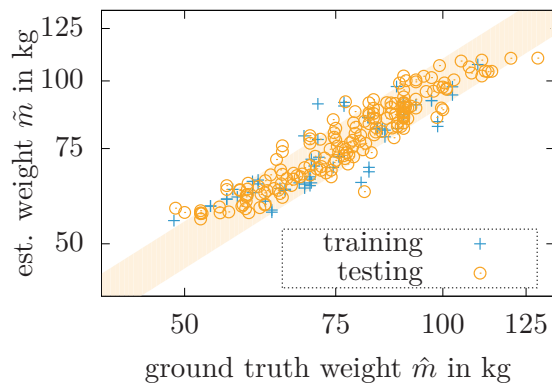
This experiment should demonstrate that the body weight estimation with a volume reconstruction is sufficiently possible. Although volume has the highest correlation for body weight, it is extensive in calculation. Having a reference plane eases the calculation of the volume. Certainly, this causes an error because it is only an approximation for the volume. To obtain a more precise model for volume calculation, a person has to be seen from different perspectives: On one hand this could be achieved by the use of several sensors from different perspectives. This has the advantage that all sensor data can be taken from a single triggered time event. Additional work has to be done due to calibration and time synchronizing. Another possibility is the data acquisition with a single sensor from different perspectives. These sensor frames have to be aligned, e.g., with an ICP algorithm and a person keeping still. Minor movements of the person could cause errors in registration. The result could be improved having a non-rigid registration algorithm [78], with the disadvantage of having an extra cost of algorithm.

The minimum in relative error has a value of -16.8 percent while the maximum relative error is set with a value of 22.7 percent. Compared to previous experiment configurations, this range is even better as shown in the first two experiments 1 and 2. The same is shown by the standard deviation for the relative error which has a value of 6.6 percent. 58.4 percent of the data set are estimated with a relative error of less than 5 percent. Also, this is better than shown in the first two experiments, 50.6 for experiment 1, and 51.1 for experiment 2. For the range of ± 10 percent, about 87.1 percent were included in this range. For the area of 20 percent 98.7 percent were within this range. Compared to all other experiments in this section, this is the worst for this benchmark category. Figure 7.3e shows the results of this experiment as a cumulative plot and a scatter plot. Although this experiment demonstrated to be not the best estimation method, it shows that volume can be skipped if the need for precision is sufficient for the application.

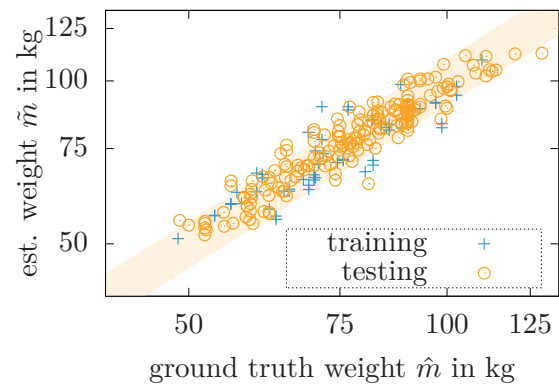
Experiment 6: All available Features, including Gender, Age and Thermal Features

For this experiment, the complete feature set is added: Looking for the thermal features extracted from the patient's data, all of these features only have a minimal correlation. Obviously, the ambient temperature does not correlate with the ground truth body weight. With a range of 21.3 percent the results show the lowest range in outliers for relative error with a minimum value of -12.9 percent and a maximum value of 17.9 percent. Moreover, the standard deviation has the lowest value in the set of experiments with 5.3 percent for relative error. Looking for the estimation with a relative error below 5 percent, 67.8 percent have a lower error. Furthermore, 94.8 percent of the data set are within an error range of 10 percent, which is the best result in this verification category. All measurements are within a relative error range of 20 percent. Using the sensor data from the Kinect camera, this is the best setting, having mostly the best benchmark values in all categories.

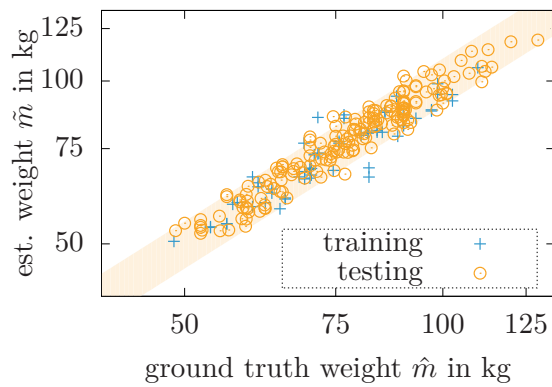
For all here presented experiments, the subset of testing has a size of 163 subjects, while the size of the training elements is 70. Repetitive training can change the result, therefore a single trial was performed. As presented in the section about ANN, overfitting can be an issue. Table 7.4 shows the statistics split for the subset of training and the subset for validation. The



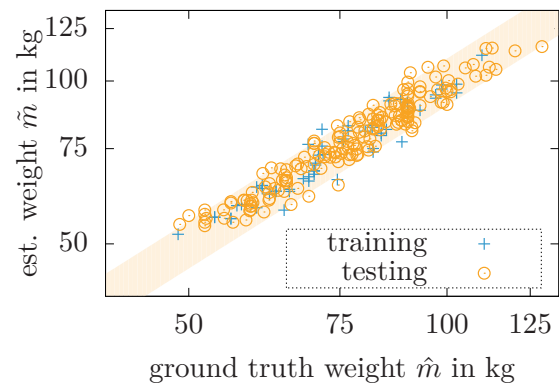
(a) Results from experiment 1



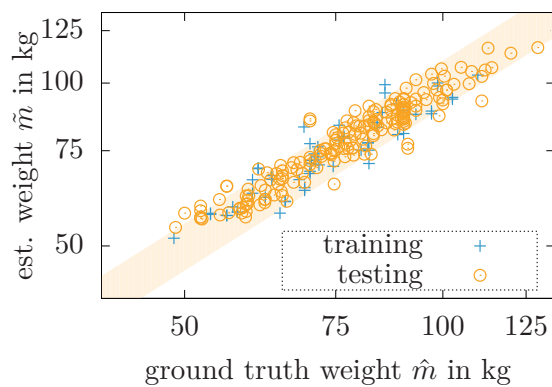
(b) Results from experiment 2



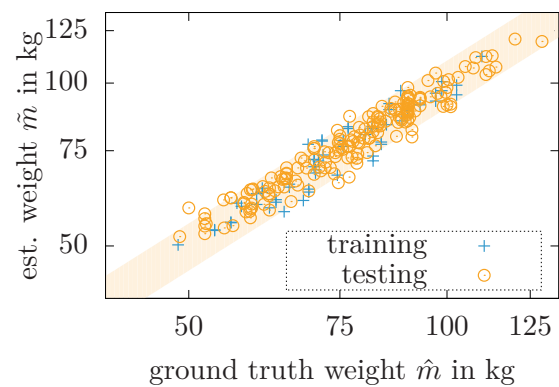
(c) Results from experiment 3



(d) Results from experiment 4



(e) Results from experiment 5



(f) Results from experiment 6

Figure 7.3: Results from experiments with different feature groups: The best result is achieved by the fifth experiment, using all available features for the estimation. In contrast to that, the worst result is presented by the first experiment. Here, the estimation only uses the volume of a subject.

Table 7.3: List of features used for experiments and statistical results. The best value over all experiments is highlighted in bold. The table presents the relative and absolute errors, as well as the in range estimations and the MAE and MSE.

Experiment		1	2	3	4	5	6
Rel. Error in %	min	-22.9	-20.3	-19.0	-14.0	-16.8	-12.9
	max	26.8	25.3	19.8	14.3	22.7	17.6
	range	38.0	34.8	29.7	26.2	34.2	21.3
	mean	0.21	0.28	-0.07	0.28	0.43	0.29
	σ^2	7.6	7.3	5.8	5.5	6.6	5.3
Abs. Error in kg	min	-18.8	-16.7	-15.6	-13.2	-18.4	-11.5
	max	19.2	18.1	14.2	13.0	15.9	9.8
	range	38.0	34.8	29.7	26.2	34.2	21.3
	mean	-0.3	-0.2	-0.3	-0.1	0.2	0.0
	σ^2	6.0	5.6	4.6	4.4	5.2	4.2
In Range	in 5%	50.6	51.1	61.8	63.9	58.4	67.8
	in 10%	83.7	83.3	91.0	91.4	87.1	94.8
	in 20%	99.1	99.1	100.0	100.0	98.7	100.0
MAE in kg		4.68	4.46	3.65	3.42	4.00	3.21
MSE in kg ²		35.3	32.8	22.0	19.2	26.9	17.0

Table 7.4: Evaluation of the difference between the subset of training and testing: In case of overfitting, the errors of the training set would be significantly lower compared to the results from the testing data set.

	Data Set	Size	Relative Error in %				In Range in %			Error in kg / kg ²	
			min	max	mean	σ^2	5	10	20	MAE	MSE
train	E-DS + H	163	-12.4	11.8	-0.6	5.65	65.5	91.4	100	4.47	31.7
test	E-DS + H	70	-12.9	17.6	0.6	5.20	68.6	96.0	100	4.07	27.2

values for the minimum, maximum, mean and standard deviation of the relative error are in the same order, although minor differences appear. For example, the maximum error for the testing data set is higher with 17.6 percent compared to the maximum relative error for training with 11.8 percent. The number of estimations within a specific range is similar for the 5 percent margin. Also all estimations – irrespective of training or testing – is within a range of 20 percent relative error. The amount of estimations within a range of ± 10 percent is better for the testing subset.

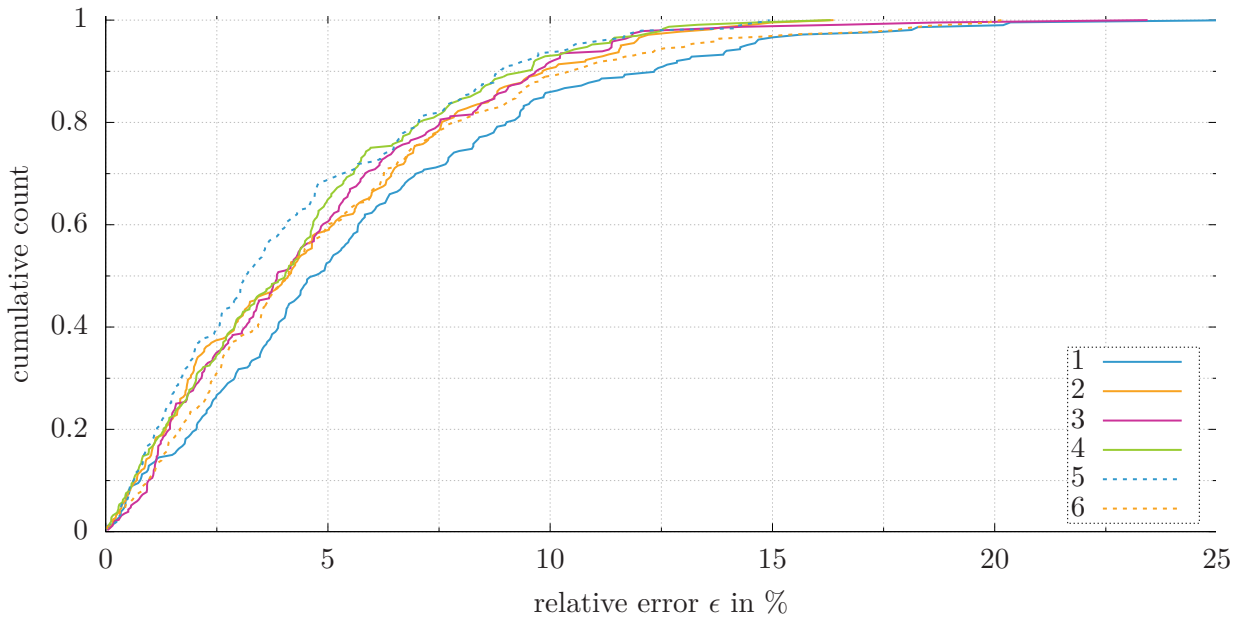


Figure 7.4: Comparison of the experiments with a cumulative plot: When all available features are used – see experiment 5 – the performance is superior to most other experiments. The worst performance is achieved by the first experiment, where only the volume is used.

7.4 Sensor Modalities

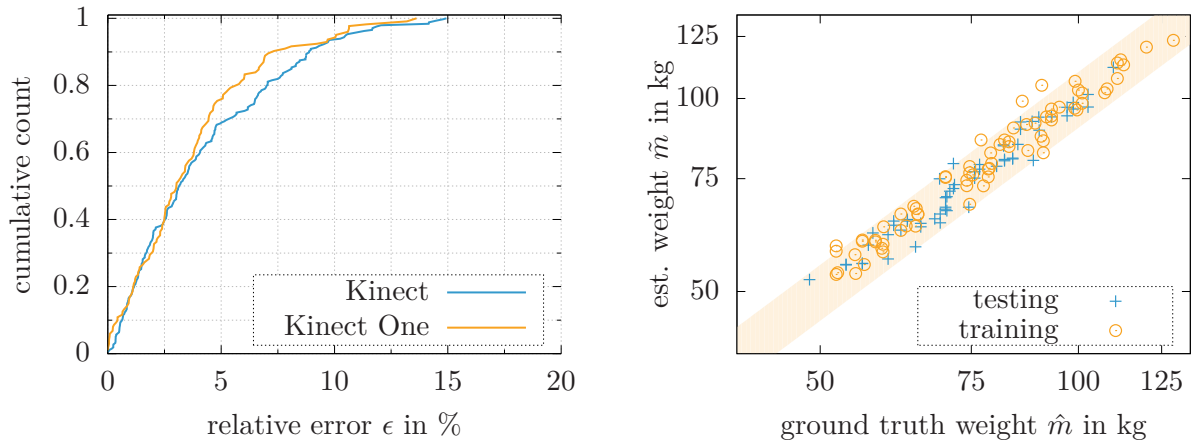
The here presented approach for body weight estimation is based on a fused point cloud. Until now, the experiments in this section presented results with data from the Kinect camera, a structured light sensor. This section should prove that the approach can provide a sufficient result for weight estimation, irrespective of the used 3D sensor. Therefore, one experiment with the Kinect One is presented in comparison to the estimation based on the Kinect camera. Furthermore, the estimation based on a sub-sampled data is evaluated.

Different Working Principles

The Kinect and the Kinect One both provide a point cloud, containing the Cartesian points, as well as color for each point. Figure 7.5a demonstrates the results of this experiment as a cumulative plot for both sensors: In a range of the relative error of zero to 4.5 percent, both sensors perform similarly. Up until the threshold of 10 percent is reached, the Kinect One outperforms the Kinect sensor, estimating more people with a higher accuracy. With a relative error of more than 10 percent, both sensors perform similar again. The difference in the estimation of both sensors can be found in the quality of the data, provided by the Kinect One, with less noise in depth, and as well a higher density of pixels. Figure 7.5b presents the raw data from all available estimations based on the data from the Kinect One. Additionally to the two graphs, Table 7.5 provides the statistical values for this experiment. The minimum and the maximum of the absolute relative error are smaller for the Kinect One; -8.7 and 14.3 percent for the Kinect One in contrast to -12.9 and 17.6 percent for the Kinect camera. Furthermore, the amount of

Table 7.5: Comparison of weight estimation between Kinect and Kinect One sensor: Due to the different working principles and the different signal and noise ratio, the Kinect One can provide a refined estimation of body weight.

	Data Set	Size	Relative Error in %				In Range in %			Error in kg / kg ²	
			min	max	mean	σ^2	5	10	20	MAE	MAE
Kinect One	E-DS	106	-8.7	14.3	0.90	4.80	75.6	95.3	100	2.86	13.8
Kinect	E-DS	233	-12.9	17.6	0.29	5.3	67.8	94.8	100	3.21	17.0



(a) Comparing the results from the weight estimation of the Kinect and the Kinect One camera based on all available features.

(b) Training and testing data for body weight estimation performed with the Kinect One.

Figure 7.5: Weight estimation performed with the Kinect One.

estimations within a certain range is better for the Kinect One. For the dosing of stroke patient – where the 10 percent range is important – the experiment with the Kinect One outperforms with 95.3 percent of all estimations with a relative error of less than 10 percent; 0.5 percent more compared to the Kinect sensor. Also, the MAE and the MSE are smaller for the Kinect One.

According to these results, the body weight estimation works better with the Kinect One sensor. This can be explained due to the higher data quality.

Resolution in Depth

To proof the concept of robust features, the sensor’s data is stepwise reduced. This shows, that the estimation does not rely on the amount of data perceived from the environment. However, it is expected, that with decreasing data, the body weight estimation gets also less reliable, providing an estimation with lower accuracy.

Each point cloud is resized by scaling the depth, color, and thermal image. Additionally, the data is improved by interpolation. The resizing based on a voxel grid would also be suitable,

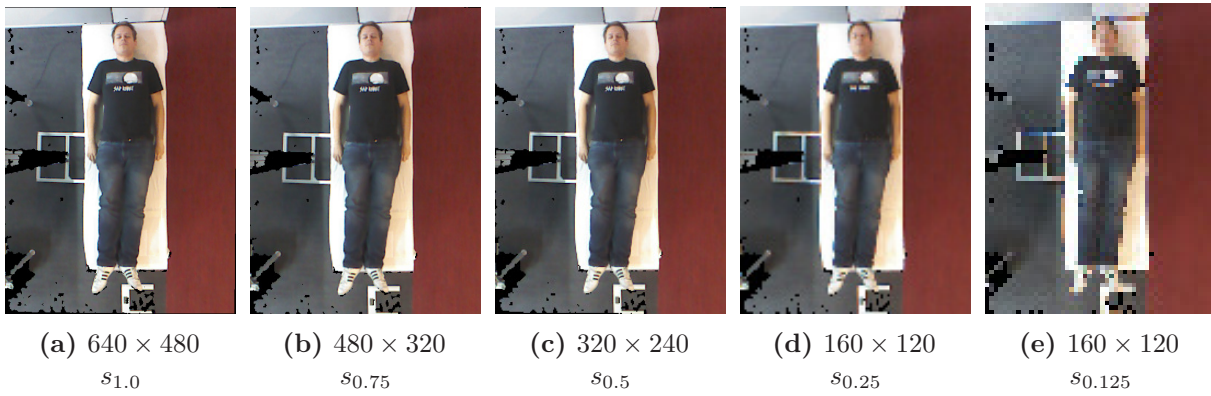


Figure 7.6: Resized point cloud: With every step, the point cloud is reduced by half of its total size.

however the algorithm for segmentation – with its morphological operations – relies on an ordered point cloud. The rescaling based on a voxel filter cannot provide an ordered point cloud [155].

Figure 7.6 illustrates the resizing of the point clouds: The left image shows the color stream in full resolution of 640×480 . In every subsampling step, the width and the height are reduced by 25 percent of the original image; which implies that the size of the data is reduced by half. After four steps of subsampling, the derived point cloud only consists of 4,800 points. For this experiment, the neural network was trained and validated for each single experiment. The networks were trained with 70 percent of the available data. The settings for the training are fixed for all experiments.

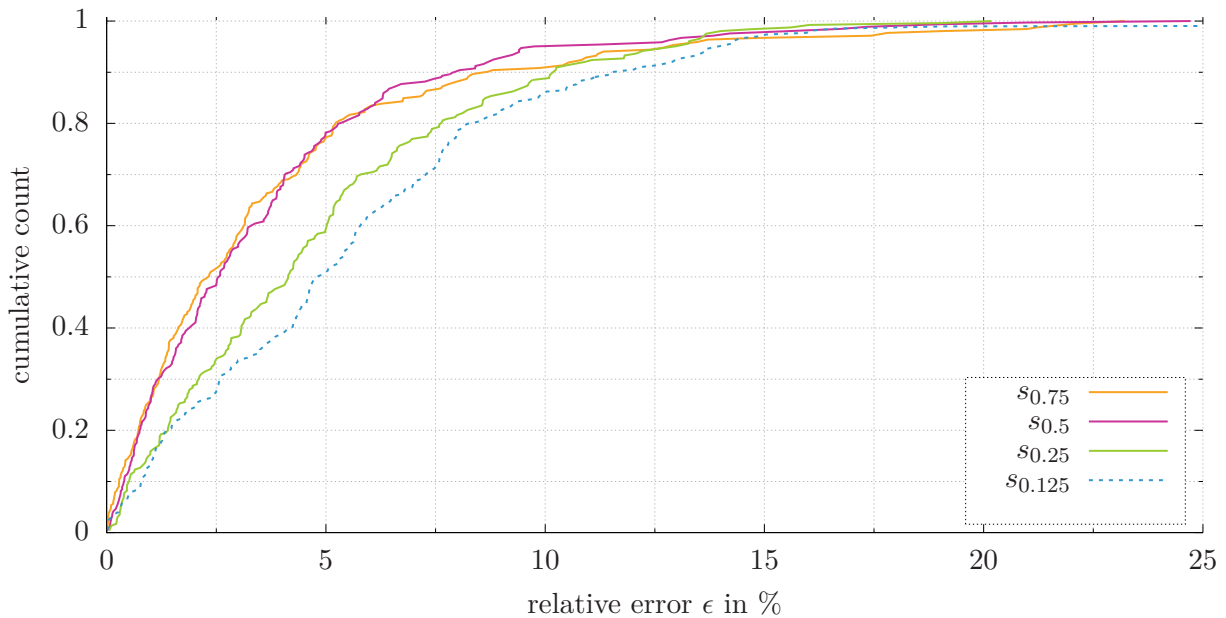


Figure 7.7: Results from subsampling the size of the point clouds: Due to the reduced size, the algorithm for the estimation can process the result faster.

Table 7.6: Statistics from subsampled data.

Experiment	Nr. of Points	Relative Error in %				In Range in %		
		min	max	mean	σ^2	5	10	20
$s_{1.0}$	302,700	-13.7	15.0	0.45	5.12	67.5	93.9	100
$s_{0.75}$	153,600	-18.8	26.9	0.23	3.94	75.8	91.8	98.3
$s_{0.5}$	76,800	-17.4	32.8	0.15	5.34	75.8	93.9	99.5
$s_{0.25}$	19,200	-16.0	25.3	0.27	6.18	61.9	89.2	99.5
$s_{0.125}$	4,800	-42.0	19.1	0.40	7.90	51.1	85.3	99.1

Figure 7.7 shows the results of this experiment as cumulative plot. Additionally, Table 7.6 provides the statistics with the relative error and the percentage of estimations within a given range. The cumulative density changes, depending on the subsampling: The original image first reaches a density of 100 percent. Especially for estimations below a relative error of 10 percent, the subsampling for 75 percent and 50 percent of the data outperform the original point cloud. The more points are removed from the points cloud, the result gets worse, resulting in an absolute relative error of 10 percent. The subsampling worsens the result in body weight estimation. Though, there is not a clear connection between the amount of subsampling and an increase in one of the statistic values. In most cases, the range between the minimum and the maximum relative error increases. However, for the 25 percent, the range decreases, providing a smaller range. These deviations can be explained by minor variances during training. The experiment for subsampling showed, that the estimation gets worse with a reduction of the amount of applied data. Depending on the application and the used sensor, a rough body weight estimation might be sufficient. Additionally, the computation on fewer points needs less time, so a real-time body weight estimation is possible with less powerful computers.

7.5 Comparison against Related Work

The here presented approach competes against methods from related work. This includes anthropometric methods from the clinical scenarios, as well as vision-based body weight estimation methods. The upcoming experiments are based on the data set from the clinical environment. To prevent an influence of the physician's estimation, a strict sequence for the estimations was set: First, the physician makes a visual guess, not being influenced by colleagues or the patient. After that, the patient is asked to provide his own body weight, in case the patient is knowledgeable, does not obviously suffer from dementia, or the symptoms of stroke. Thereafter, the visual body weight estimation is executed, providing an estimated value to the treating physician. The anthropometric features are taken afterward in the same step the patient is weighted on a scale or onto the stretcher. This ensures a fast treating with minimal effort for data acquisition. Table 7.7 illustrates the results from this section for comparison against other methods from related work.

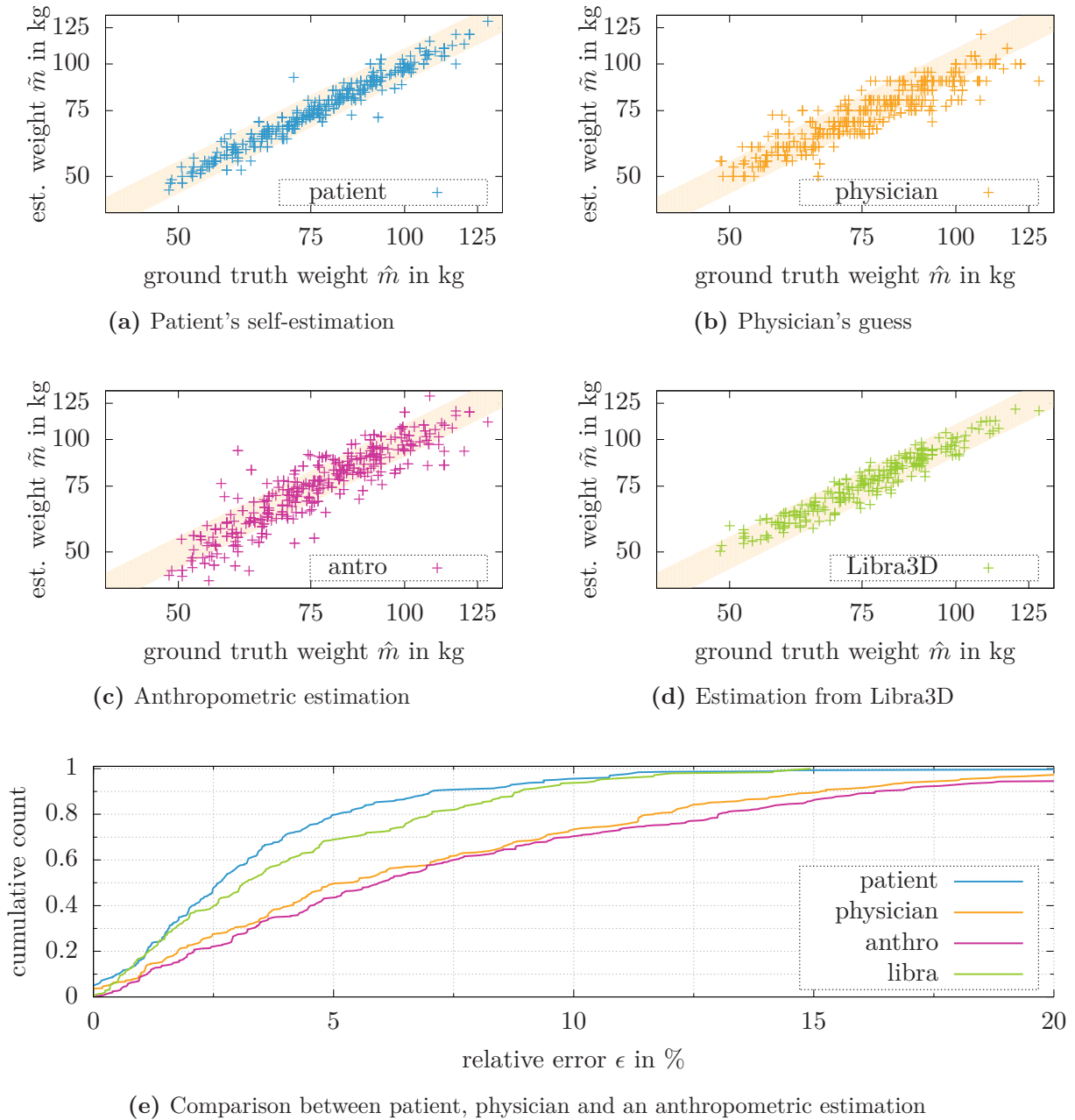


Figure 7.8: Comparison of *Libra3D* with related work: The cumulative plot shows the four tested methods in one graph (d). The other three plots illustrate the results for each experiment from body weight estimation provided by the patient (a), the treating physician (b), and the anthropometric method proposed by Lorenz et al. [124] (c). To ease comparison, the results from *Libra3D* from experiment 6 are illustrated (d).

Physician’s Guess

The guess from a physician is still the most common solution when it comes to an emergency weight estimation. The visual guess was taken from a single physician from the Universitätsklinikum Erlangen, although several physicians were in the trauma room for treatment. The single physician gave a guess for the patient’s weight, without a recommendation of his or her colleagues. For benchmarking, the data set from the hospital is taken.

The minimum relative error acquired has a value of -28.1 percent while the maximum was reached with a value of 30.2 percent. The range of those outliers is therefore more than 50 percent. The standard deviation for this configuration was listed with 8.3 percent, having a mean value of 3.3 percent. Compared to the patient’s self-estimation, all of these benchmark values are worse. Having a look for the 10 percent threshold set for medical dosing, only 73.9 percent of the patients in the data set were sufficiently estimated within this range. These results coincide with related work where only two-thirds of visual guesses by physicians are sufficient for drug dosing [31].

Patient’s self-estimation

If the patient is knowledgeable, he or she could be asked by a treating physician for his or her body weight. Only patients being able to indicate their body weight, were taken for experiments.

The minimum in relative error was recorded with a value of -20.8 percent; the maximum with 20.2 percent. The standard deviation for the relative error is set to 3.7 percent. Furthermore, the mean value in relative error was listed with 1.27 percent, demonstrating that people’s self-estimation tend to be lower than ground truth. There can be multiple reasons for the outliers: Due to anonymization, it is not reproducible which patients might suffer from dementia. Therefore those people might not remember their own body weight. Moreover, the date of weighing is

Table 7.7: Results from comparison with related work: The results from the system *Libra3D*, the physicians guess, the patient’s self-estimation, as well as the weight estimation based on the method presented by Lorenz et al. [124] are compared. The self-estimation is superior to all other methods, but in the scenario of stroke, patients are often not able to provide their body weight. The weight estimation based on *Libra3D* can provide sufficient estimations for medication if the patient is not able to speak. The best value in each category is marked in bold.

		Libra3D	Physician	Patient	Lorenz et al. [124]
Rel. Error in %	min	-12.9	-28.1	-20.8	-22.7
	max	17.6	30.2	20.2	27.3
	range	21.3	58.3	41.0	50.0
	mean	0.29	3.3	1.3	1.7
	σ^2	5.3	8.3	3.7	8.0
in Range	in 5%	67.8	51.0	79.5	41.9
	in 10%	94.8	73.9	95.7	69.3
	in 20%	100.0	96.5	99.3	94.7

long time ago and the people's body weight changes from that time. In addition, as shown in previous work, having only stroke patients, only 50 percent of them would be knowledgeable [31]. Nevertheless, this experiment showed good results for the patient's self-estimation: The number of outliers is small and if someone is knowledgeable and does not suffer from dementia, the chances of providing a sufficient body weight are quite high, especially for drug dosing in an emergency scenario. This result corresponds to the related work presented by Breuer et al. [31].

Figure 7.8a illustrates the scatter plot for this experiment, while Figure 7.8e shows the cumulative count for the relative error towards ground truth body weight. In the plot of the cumulative density, the subject's self-estimation is always the best solution.

Anthropometric Estimation

As shown in chapter 2, several approaches exist to estimate someone's body weight by measuring anthropometric landmarks. Here, the method by Lorenz et al. [124] will be compared which has the focus on stroke patients. The measurements at a patient's body were performed by a single physician with a measuring tape. The equation for the estimation is based on the body size, the hip and waist circumference, see equation (2.1) on page 18.

The extrema in relative error were recorded with a minimum of -22.7 percent and a maximum of 27.3 percent. The range of these estimations is smaller, compared to the visual guess from the physician. The standard deviation is listed with 8.19 percent having a bias value of 1.7 percent. The anthropometric approach developed by Lorenz et al. [124] proposed to estimating the weight of more than 93 percent of all subjects better than ± 10 percent. In contrast to that, the here received results differ and the number of correct dosages is lower. This can have various reasons: Applying the measurement tape around someone's body can produce errors in measurement. This also depends on the physician who is taking the measurements. Nevertheless, this method seems to be more suitable than the physician's visual guess. However, the expenditure of time is remarkable higher than a visual guess and the patient has to be moved to apply the measuring tape correctly.

Figure 7.8c illustrates the results in a scatter plot, while Figure 7.8e demonstrates the cumulative count for the relative error. In the cumulative plot, the curve for the anthropometric estimation is the worst over the complete applied data set. Figure 7.8b shows the result in a scatter plot.

7.6 Extension for Standing People

To extend the approach for clinical usage and the body weight adapted drug dosing application for stroke treatment, the upcoming experiments investigate if the same approach could be used to estimate body weight of standing people.

7.6.1 Weight Estimation from Standing People

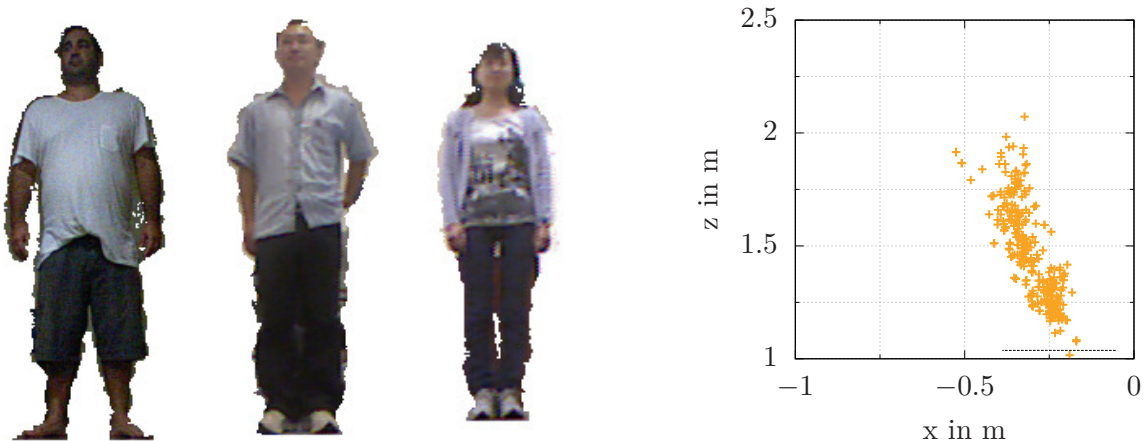
In contrast to the experiments for lying people, the most correlating feature – the volume – cannot be used because no reference surface for the back of a person exists. Therefore, the body weight estimation has to rely upon the remaining features. A previous experiment with the two

data sets from hospital and E-DS illustrated that the body weight estimation gets worse if the volume is missing.

For the experiment, the data set W8-300 generated by Nguyen et al. [145], is applied. This data set contains 299 people standing in front of a Microsoft Kinect camera. Color and depth frames are saved separately with a resolution for each channel of 8 bit. To perform reconstruction towards a colored point cloud, the projection from equation (3.2) is applied. The segmentation has been done in advance based on ground detection with the RANSAC model: The images only contain the person's data as a depth and color image; the background is not visible. The file name of each data set contains first the gender, second the ground truth body weight, and ends with the surname of the person. The ground truth body weight varies within a range starting from 40 kg up to 104 kg. In the experiments, 202 males and 97 females participated. Figure 7.9a shows four random subjects from the data set. Figure 7.10 illustrates the result from the applied data set: First the ground truth ordered data sets are shuffled. For the training of the ANN, 70 percent of the data set was used; the other 30 percent were applied for testing.

All people were not told to hold a fixed posture but most of them were standing normally with their arms aside. The pose of each person standing in front of the camera is not fixed and varies, which is shown in Figure 7.9b.

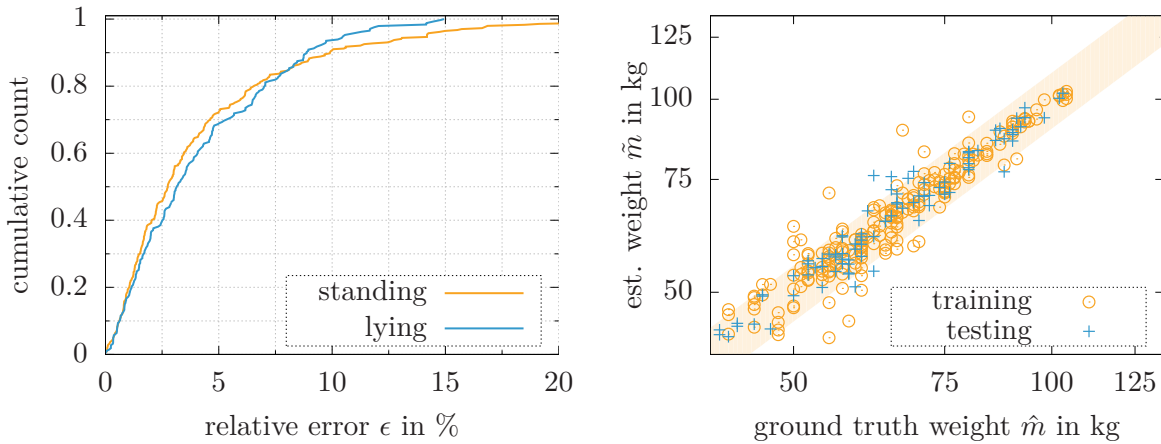
Nguyen et al. [145] compared the MAE in their publication: They reached a MAE of 4.62 kg for female and 5.59 kg for male persons. Without the discrimination in gender, the algorithm performs with a MAE of 5.20 kg. This experiment also includes the ground truth of the gender for the applied model. Compared to their results, the here performed experiment reaches a MAE of 4.6 kg including estimations from male and female subjects. Figure 7.10a compares the results for standing subjects based on the W8-300 data set and the estimations for the lying subjects from the medical application: Until a relative error of 7.5 percent, the estimation for standing subjects outperforms the algorithm for lying patients. However, after this value, the approach for



(a) Randomly selected subjects from the data set.

(b) Poses of standing people

Figure 7.9: Data from the W8-300 data set published by Nguyen et al. [145].



(a) Cumulative plot to compare the estimation for standing and lying subjects.

(b) Results for the body weight estimation of standing subjects, based on the data set provided by Nguyen et al. [145].

Figure 7.10: Results from the experiment with people standing in front of the camera.

lying people performs better. Table 7.8 presents the statistic values for this experiment: Both, the minimum and the maximum error are high, compared to the previously presented results; the minimum error has a value of 33.5 percent while the maximum error has a value of 22.8 percent. For the estimations within a fixed range, 70 percent of all subjects are estimated with a relative error below 5 percent, 88.6 percent of the subjects are in a range of 10 percent and 98.0 percent are within a range of ± 20 percent.

The result from this experiment indicates that the estimation for standing people is sufficiently possible. One possible application where the estimation for standing people can have a benefit is the previously mentioned airline which weighs passengers to save fuel [62]. In contrast to a common scale working with force, the optical weight estimation can have the benefit that clothing and cabin luggage can be filtered, based on a sufficiently trained model.

7.6.2 Weight Estimation from Walking Subjects

In addition to the previous experiments, also people walking are estimated for their body weight. Therefore, a data set was recorded with students and employees of the Technische Hochschule Nuremberg Georg Simon Ohm, walking in front of a Microsoft Kinect One. The person is walking towards the sensor, starting at a fixed distance. A marker on the floor shows the limitation of the recorded scene, due to the FOV of the sensor. The sensor is mounted on a tripod in a height of around 1.5 meters. A data set of a person \mathcal{D} consists out of several frames from the sensor $D_0, D_1, \dots, D_{|\mathcal{D}|}$. Every frame can be transformed into a point cloud \mathcal{P} .

Figure 7.11 illustrates the results of this experiment as a scatter plot: First, the person is segmented from the background. Second, for every frame of the data set the body weight estimation is applied. In a scatter plot together with the ground truth body weight, a line

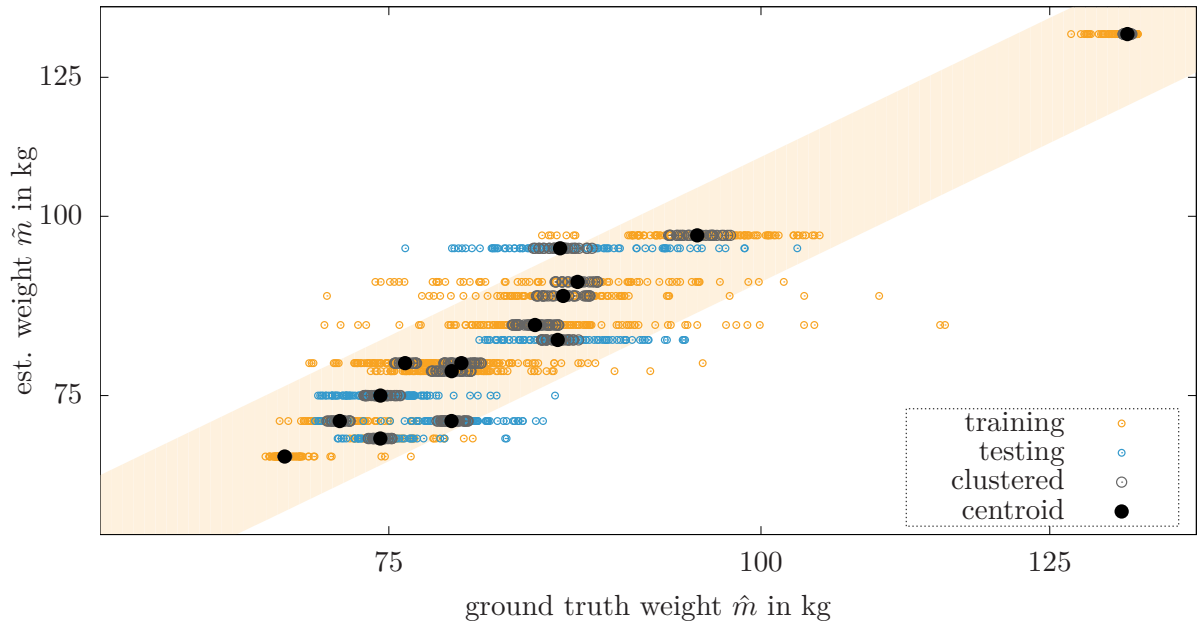


Figure 7.11: Results from the experiment with people walking towards the camera: The data set is divided in training and testing. Furthermore, the plot illustrates the clustering approach which is presented in Algorithm 5. The grey points show the 30 percent of all points for each subject used for the calculation of the centroid and therefore the final estimation.

becomes visible for every single person. Some of the estimations are close to the ground truth body weight. Even outliers of more than 30 percent occur. Therefore, taking an arbitrary frame from a person’s data set will likely lead to an insufficient result. Third, a clustering method is applied, so not only an arbitrary frame from a person’s data set provides an estimation for the body weight. A Euclidean clustering method is applied to improve the outcome. The clustering is applied as described in Listing 5.

Figure 7.12 illustrates the results in a cumulative plot in comparison to the results achieved by the estimator for the lying people. Here, the results for the walking people, recorded with the Kinect One are compared with the lying subjects, also recorded with the Kinect One. The plot illustrates that the estimation for walking people outperforms the approach of lying people. However, this can be explained by the limited amount of data used for evaluation of the walking people. It is expected that increasing the data for training and validation would decrease the performance of this application, due to a higher variety in subjects.

Table 7.8 compares the statistical values of walking and standing people for body weight estimation, in comparison to the previously presented values for lying subjects. It is to be noted that the number of subjects varies between the different approaches and experiments. The comparison between lying, standing and walking subjects illustrates, that the approach for body weight estimation is not only limited to subjects lying on a medical stretcher. With a higher amount of subjects, providing more data for training, the model can be refined.

Algorithm 5: Euclidean clustering and filtering to improve the outcome of weight estimation based on a set of images.

1. For every frame in the data set $D_i \in \mathcal{D} = \{D_1, \dots, D_n\}$, estimate the body weight based on the extracted features $\tilde{m}_i(D \rightarrow \mathbf{f})$ where $i=1, \dots, n$.
2. Calculate the mean distance \bar{d} for n single estimations of a data set towards all other estimations by

$$\bar{d}_i = \frac{1}{n} \sum_{j=1}^n |\tilde{m}_i - \tilde{m}_j| \quad i = 1, \dots, n \text{ where } i \neq j \quad .$$

and store the calculated average distances in a vector $\bar{\mathbf{d}} = (\bar{d}_1, \dots, \bar{d}_n)$. Now, every estimated body weight has a corresponding distance (m_i, \bar{d}_i) .

3. Sort the calculated distances in an ascending order $\bar{d}_1 \leq \bar{d}_2 \leq \dots \leq \bar{d}_n$.
4. Remove outliers in the distance vector by cropping the distance vector to a reduced size k , so high distance values are removed. For a smaller size, more estimations are removed and therefore potential outliers.
5. Calculate the centroid based on the remaining estimations by

$$\tilde{m} = \frac{1}{k} \sum_{i=1}^k m_i \quad .$$

Table 7.8: Results from experiments for standing and walking people. Additionally, the results from Pfitzner et al. [162] are added for comparison. The best results in each category are highlighted in bold.

	Data Set	Size	Relative error in %				In range in %			Error in kg / kg ²	
			min	max	mean	σ^2	5	10	20	MAE	MSE
Lying [162]	E-DS	127	-8.7	14.3	0.90	4.80	75.6	95.3	100	2.86	13.8
Stand	W8-300	299	-33.5	22.8	-0.6	6.73	70.0	88.6	98.0	4.60	45.6
Walk	W-DS	14	-6.7	9.38	0.32	3.88	78.5	100	100	3.30	20.5

7.7 Extension for BMI Estimation for CT Dose Reduction

During a CT scan, a patient can receive a high dose of X-ray. The effective radiation dose depends on the examined body parts. CT scans increase the risk of cancer, e.g., leukemia [156, 30].

As described in the introduction, besides the direct body weight, the BMI is a useful measure variable when it comes to dose reduction for CT examinations. Most manufacturers of CT scanners provide the BMI as an input parameter, which is entered by a medical technical assistant. Patients having a high BMI should be scanned with a higher dose, to enhance the

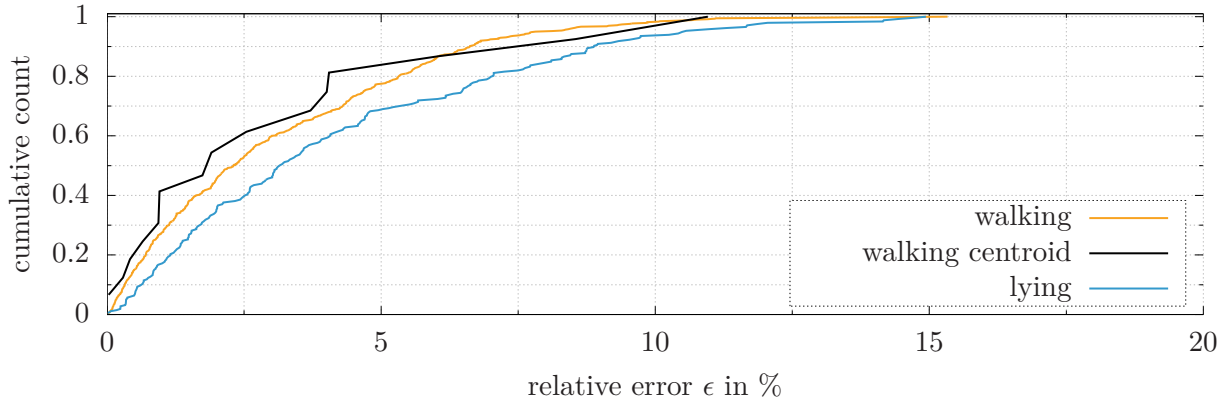


Figure 7.12: Comparison of body weight estimation with walking and lying subjects.

Table 7.9: Results from the estimation of BMI.

Data Set	Size	Relative error in %				In range in %			Error in kg/m^2 / kg^2/m^4	
		min	max	mean	σ^2	5	10	20	MAE	MSE
HWT-DS	127	-26.74	23.38	0.21	6.29	78.7	89.0	96.9	1.07	3.09

output of the CT scan; otherwise necessary details for the diagnosis can be pictured poorly. For people with a low BMI, the radiation dose should be minimized. This ensures that the tissue of the patient is exposed to less radiation. A more precise knowledge about the tissue of the patient can help to minimize the effective radiation dose to a minimum.

For the upcoming experiment the data set from the hospital is taken, as this one contains also the body height of the subjects, so the ground truth BMI can be calculated. The data set from the hospital contains subjects with a BMI of up to $46 \text{ kg} / \text{m}^2$. Figure 7.13 illustrates the result of the regression in a scatter plot, which is achieved by an ANN-based machine learning: Most of the subject's BMI can be estimated with a low relative error. Table 7.9 shows the results of this experiment in more detail: The estimation contains outliers between -26.74 and 23.38 percent. However, nearly 90 percent of all subjects are estimated with a relative error of less than 10 percent. The approach presented by Kocabey et al. [106] was able to estimate the BMI, so that 56.2 percent of all subjects can be estimated with a relative error of less than $5.5 \text{ kg} / \text{m}^2$. For the *Libra3D* approach, the subject has to be completely in the FOV of the sensor, while the approach by Kocabey et al. [106] only uses images from the face.

Beside the function regression, the BMI provides specific classes, as illustrated at the beginning of this section. In another experiment, the BMI is classified with an ANN. Now, the output of the ANN has one neuron for a possible class, in contrast to a single neuron for the function regression. The confusion matrix illustrates the successful rate for classification. This matrix provides the necessary information to evaluate if a classification is successful. The value on the diagonal represent the correct classifications. Values not being on the principal diagonal are

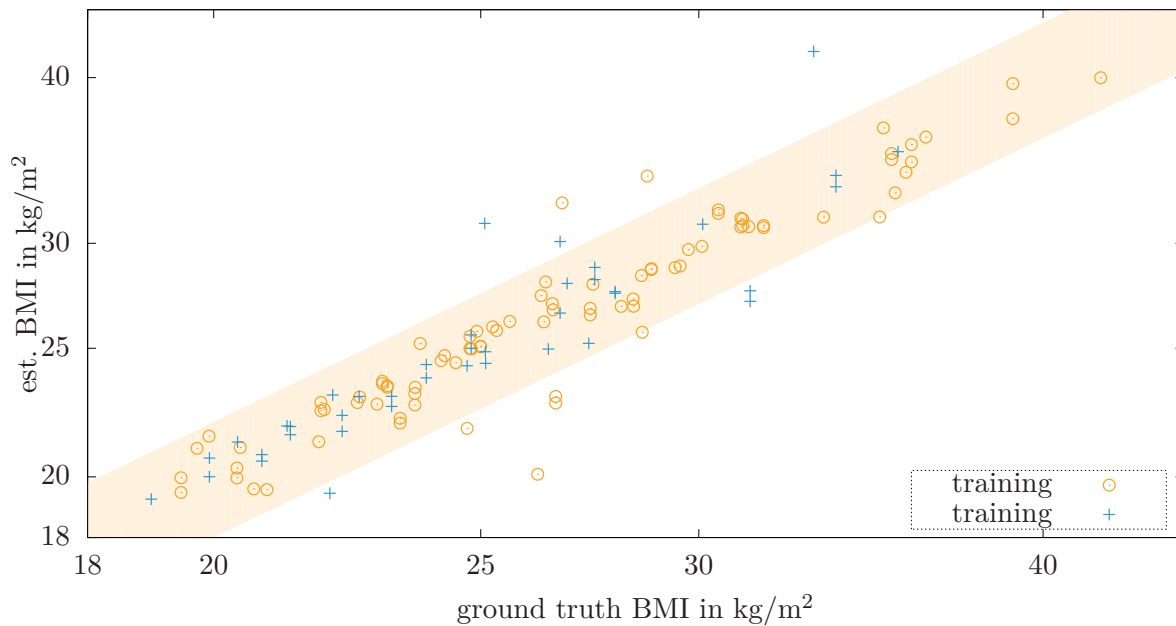


Figure 7.13: Regression of BMI: The yellow area marks the range for a relative error of less than ± 10 percent. Here in this experiment 89 percent of all subjects are estimated with a relative error of less than ± 10 percent.

missclassifications, where a class is mistaken for another class. The percentage for the correct classifications of each class can be found in the right column, which is identical with the last row in the confusion matrix. The total amount of correct classifications is found in the last cell of the principal diagonal.

The experiment for the classification of the BMI is done with two different sets of classes: First, the BMI of the subjects is structured in three classes – underweight, regular weight and overweight – as presented by Table 7.10. No underweight subjects are available. Most of the classifications are correct. Only 5 percent of the subjects with normal BMI are classified as overweight, and vice-versa. In total, 89 percent of all subjects are classified with the correct BMI.

Second, the BMI is classified based on six classes, leaving out the class for underweight due to no available subjects. The extension of the classes towards six classes provides a finer classification, but also bears the risk of missclassifications, due to the smaller range. The classes are normal weight and four classes for overweight. The results for this experiment are presented by Table 7.11. In contrast to the experiment with three classes, more false classifications appear. In total, 78 percent are classified correctly. Most miss-classifications appear for the subjects with the normal weight. The performance of a human visual BMI estimation is presented by [54]: In a study with 292 patients from an intensive care unit, nursing staff estimated the BMI of the patients. In this study, 176 patients were classified with the correct BMI. The study used underweight, normal obese and strongly obese for the classification. Comparing the here

Table 7.10: Confusion matrix for BMI classification with three classes: The applied data set does not contain underweight subjects with a BMI of less than 18. Therefore, the confusion matrix contains the value NaN for the estimation of underweight subjects. In total 89 percent of the subjects are classified correctly.

		Target			
		U	N	O	
Output	U	0.0	0.0	0.0	NaN
	N	0.0	0.39	0.05	0.88
	O	0.0	0.05	0.50	0.90
		NaN	0.88	0.90	0.89

Class		Interval
U	Underweight	$BMI \leq 18$
N	Normal range	$18 < BMI \leq 25$
O	Obese	$25 < BMI$

Table 7.11: Confusion matrix for BMI classification with five classes: In this experiment 78 percent of the subjects are classified correctly.

		Target					
		N	O1	O2	O3	O4	
Output	N	0.41	0.05	0.0	0.0	0.0	0.90
	O1	0.03	0.24	0.05	0.0	0.0	0.74
	O2	0.04	0.01	0.09	0.0	0.0	0.65
	O3	0.04	0.0	0.0	0.02	0.0	0.29
	O4	0.0	0.0	0.0	0.0	0.02	1.0
		0.79	0.82	0.61	1.0	1.0	0.78

Class		Interval
U	Underweight	$BMI \leq 18$
N	Normal range	$18 < BMI \leq 25$
O1	Obese Level 1	$25 < BMI \leq 30$
O2	Obese Level 2	$30 < BMI \leq 35$
O3	Obese Level 3	$35 < BMI \leq 40$
O4	Obese Level 4	$40 < BMI$

presented machine learning approach with a correct prediction for 78 percent, the human BMI estimation in the study was 60 percent.

7.8 Summary

The experiments validate the previously presented approach. The evaluation of the different sets of features is recorded by selected criteria, so comparison of different methods and feature sets is possible. For the best body weight estimation, all available features should be used. However, experiments with different feature groups illustrate that the estimation by physicians is outperformed even though only the volume is used as a single feature.

When comparing the *Libra3D* system with related work, for example, the physician's guess, the patient's self-estimation, or an anthropometric method, *Libra3D* provides the most reliable estimation, if the patient is not able to provide his or her body weight. The extension with the subjects standing and walking in front of the camera showed that in general the selected features can be used to train a model for body weight estimation. The results for the lying subjects are better, because the people on the stretcher have similar posture, in contrast to the people walking in front of a 3D sensor. Also, this section showed that the estimation and the classification of the BMI are possible and can outperform the visual estimation made by humans.

Chapter 8

Conclusion and Future Work

This study investigated the estimation of body weight in the context of medical applications based on the data from 3D depth cameras.

Conclusion

The introduction mentioned five contributions to research. In the conclusion, the evidence for the contributions is explained.

Can body weight be estimated reliably using a camera system?

As shown in the section on the experiments, even a rough measurement of the volume and a linear regression can outperform the method of visual guessing by physicians. These results were already achieved with low computational cost within the first year of the project. The volume-based estimation is published in Pfitzner et al. [163]. Although the system provides a better result for body weight estimation for more subjects, the treating physician still has the responsibility for the correct dosing and for the outcome of treatment. Therefore, the physician must evaluate whether the estimation seems reliable, or choose another state-of-the-art method for validation. The legal aspects of a commercially available visual body weight estimation system for stroke patients will form part of future work.

Does a low-cost consumer camera achieve enough accuracy for body weight estimation?

The Microsoft Kinect and the Kinect One are both sufficient for body weight estimation. However, the results with the Kinect One are slightly better, compared to the approach using the Kinect camera, due to less sensor noise and a higher resolution. The price of the complete system without the thermal camera is around 700 Euros for one 3D sensor and a computer in the trauma room. The computer does not need to be expensive or powerful. The current approach is based on a standard Linux system and does not rely on GPU processing. With a cheaper computer, the time taken for the result to be presented to the physician is longer. Currently, the most expensive part integrated into the system is the thermal camera, with a price of around

3,000 Euros. However, the expensive camera is not necessary, and a cheaper model could be used to achieve the segmentation of the subject and the environment. The thermal information in the image provides a reliable source of data for an easy segmentation approach.

Is a visual body weight estimation more reliable than state-of-the-art methods?

Asking the patient is still the most reliable method for the dosing of stroke patients. This fact was highlighted by the work of Breuer et al. [31], as well as by experiments in the hospital. However, in around 50 percent of all cases, the patient is not able to provide his or her body weight. The physician's guess is only adequate in around two-thirds of all cases. The results achieved by the formula from Lorenz et al. [124] are in a similar range. The proposed approach in this thesis outperforms the physician's guess and anthropometric estimation. In 95.5 percent of all subjects, the relative error was less than 10 percent, providing a good solution for the dosing of most stroke patients.

How reliable is the estimation of body weight for standing or walking people?

In an extension, this study showed, based on two data sets, that the estimation of body weight for standing or even walking people is possible. Based on the W8-300 data set provided by Nguyen et al. [145], body weight estimation for standing people is proved. The results from *Libra3D* are in most cases more precise than the results proposed by Nguyen et al. [145]. For the segmentation in this scenario, no thermal camera was used; the subject was segmented from the environment by state-of-the-art algorithms such as background subtraction and the removal of the floor plane by applying the RANSAC algorithm. In another experiment, the approach was extended for the weight estimation of walking subjects. Here a clustering method over a sequence of frames is applied, which can provide better results compared to the estimation based on a single frame.

Can the approach estimate the BMI of a patient on a stretcher?

As shown in the experimental section, the *Libra3D* system can be used to estimate the BMI of a patient lying on a stretcher. The approach was tested with patients from the trauma room. If the system used is attached to a CT scanner, the accuracy of the estimation should improve further. While a medical stretcher can be adapted in height and position, the stretcher of a CT scanner is in a fixed position, without a soft mattress on top. In addition, it is expected that the postures of subjects vary less than in the trauma room.

Future Work and Applications

As mentioned in various parts of this thesis, future work can improve the system further. The data set collected at the public event also contains data from children, see Figure 8.1. The basis for this event was the data collected from patients from the hospital over the age of 18 years. Due to the previously trained model, the ANN provided for estimations of children's body weights



Figure 8.1: Child from data set: The estimation of children is problematic due to a low amount of subjects in the data set for training. The child here is estimated with a body weight of 50 kg, while the ground truth weight is 36 kg.

has a significant relative error. The network cannot adapt to something it has not seen before. For the emergency treatment of children, methods like the Broselow table are still more reliable compared to a visual estimation system.

To enhance the estimation for stroke patients, more data sets are necessary. A medical study with 2,000 emergency patients at the Universitätsklinikum Erlangen, Germany, is still pending and will increase the number of subjects in the data set for training. It will prove the system for clinical applications and provide data to optimize contactless weight estimation for the future. As well as the data set recorded in Erlangen, a cross-validation study is planned, together with the University Hospital Essen, Germany. These two simultaneous studies ensure that new data are available twice as fast. Furthermore, different types of subjects are expected to be used.

Currently, the approach is limited to a flat plane stretcher, although the backrest of the stretchers used can be inclined. For some issues, such as a heart attack, it is more comfortable for the patient to sit in an upright position. Therefore, the extraction of the plane must be extended to a more generalized stretcher model. With this change, additional training sets are necessary, because the geometric features will also change for a subject sitting in an upright position.

As shown in this thesis, the volume correlates the most with the body weight for all extracted features. With the single 3D camera and a frontal view of the subject, the extraction of the volume is only possible due to the back surface of the stretcher. Due to sinking into the mattress, as well as breathing, the value for the volume is inaccurate. With a tracking of the breathing, the estimated volume of a subject can be refined, while the breath tracking with a 3D camera is already presented by Procházka et al. [171]. Approaches like the tensor body framework proposed by Barmpoutis [16] provide a reconstruction of the human body based on data from an RGB-D sensor suitable for real-time applications. Using a Cartesian tensor model, the back of a subject can be modeled more precisely, compared to using a single plane. In addition, thick or wide clothes can create a bias in the volume. This error can be reduced by applying a reconstruction of the human body and virtually removing the clothes, as presented by Balan and Black [13].

Not all extracted geometric features are invariant with respect to the posture of the subject. Most patients have a similar pose when lying on a stretcher, while the posture of a walking subject changes from frame to frame. Posture recognition can increase the performance of the approach for different postures of the subject. Such approaches are presented for 3D sensors [189], as well as for monocular cameras [39, 40]. They provide the localization and classification of

joints and body parts, which can be forwarded to the ANN. However, with a greater variety of postures, the learning approach needs more data for training.

As mentioned in chapter 6, a deep learning approach is possible, but in this case, many more training data are necessary. To obtain a data set with more than 100,000 images, including the ground-truth body weight, even the planned studies would take several years. Inspired by the state-of-the-art work on real-time human pose estimation of people in front of a 3D sensor proposed by Shotton et al. [189], a simulation can be implemented, providing a synthetic human model with different weights, constitutional types and postures. A framework for modeling different body types is Blender [25], where plug-ins exist to model different human body models, which are also movable and can be configured in different postures. Such a plug-in is MakeHuman [17]. Additional plug-ins like Blendsor [75] provide a framework to generate realistic depth data from a virtual scene, with the characteristic sensor noise. The biggest challenge here is the production of the human model, providing a weight which would correlate with a real human. An approach based on deep learning to model the BMI of the human body is presented by Nahavandi et al. [143]. The data for training, with a size of 100,000 models, are generated in Blender. With the help of a deep residual network, the authors propose to estimate the BMI score with an accuracy of 95 percent. This approach should be extended in the future, so that body weight estimation is also possible.

The algorithm proposed in this thesis should not be limited to the treatment of stroke patients in the trauma room. The work presented in this thesis will be integrated into an upcoming research project for the health monitoring of premature infants, called NeoWatch. The project will start at THN in 2018. As well as providing a contactless visual heart and breathing monitor, *Libra3D* will also be used to monitor the weight of a baby lying in an incubator. While incubators can be equipped with weighing devices, the weight measurement is often imprecise due objects close to the baby, e.g., a diaper or sheets over the body. Therefore, babies are often weighed outside the incubator, which carries a risk of infections and complications. While patients in the trauma room mostly have similar postures, babies in an incubator can have various different postures, so that more data are required for training. It would also be possible to combine integrated scales with the visual weight estimation algorithm. The data can be fused in the ANN combining the features from visual body weight estimation and the output of the scales, helping the system to be invariant with respect to posture and clothing.

Bibliography

- [1] Abdel-Rahman, S. *Pediatric weight estimate device and method*. US2011,055,840. 2012.
- [2] Adams, H. P., Brott, T. G., Furlan, A. J., Gomez, C. R., Grotta, J., Helgason, C. M., Kwiatkowski, T., Lyden, P. D., Marler, J. R., Torner, J., Feinberg, W., Mayberg, M., and Thies, W. “Guidelines for Thrombolytic Therapy for Acute Stroke: A Supplement to the Guidelines for the Management of Patients With Acute Ischemic Stroke”. In: *Circulation* 94.5 (Sept. 1996), pp. 1167–1174. DOI: 10.1161/01.CIR.94.5.1167.
- [3] Ailisto, H., Vildjiounaite, E., Lindholm, M., Mäkelä, S.-M., and Peltola, J. “Soft biometrics—combining body weight and fat measurements with fingerprint biometrics”. In: *Pattern Recognition Letters*. Vol. 27. 2006, pp. 325–334. DOI: 10.1016/j.patrec.2005.08.018.
- [4] Aitchison, J. *The seeds of speech: Language origin and evolution*. Cambridge University Press, 1996. DOI: 10.1086/420164.
- [5] Al-Busaidi, A. A., Jeyaseelan, L., and Al-Barwani, H. M. “The Accuracy of the Broselow™ Pediatric Emergency Tape for Weight Estimation in an Omani Paediatric Population”. In: *Sultan Qaboos University Medical Journal* 17.2 (May 2017), e191–e195. DOI: 10.18295/squmj.2016.17.02.009.
- [6] Andersen, K. K., Olsen, T. S., Dehlendorff, C., and Kammergaard, L. P. “Hemorrhagic and ischemic strokes compared: Stroke severity, mortality, and risk factors”. In: *Stroke* 40.6 (2009), pp. 2068–2072. DOI: 10.1161/STROKEAHA.108.540112.
- [7] Ansari, S., McConnell, D. J., Velat, G. J., Waters, M. F., Levy, E. I., Hoh, B. L., and Mocco, J. “Intracranial Stents for Treatment of Acute Ischemic Stroke: Evolution and Current Status”. In: *World Neurosurgery* 76.6 (Sept. 2017), S24–S34. DOI: 10.1016/j.wneu.2011.02.031.
- [8] Arens, T., Hettlich, F., Karpfinger, C., Kockelkorn, U., Lichtenegger, K., and Stachel, H. *Mathematik*. 1. Aufl. 2008. Spektrum Akademischer Verlag, Feb. 2008. DOI: 10.1007/978-3-642-44919-2.
- [9] Argall, J. A. W., Wright, N, Mackway-Jones, K, and Jackson, R. “A comparison of two commonly used methods of weight estimation”. In: *Archives of Disease in Childhood* 88.9 (2003), pp. 789–790. DOI: 10.1136/adc.88.9.789.

- [10] Arigbabu, O. A., Ahmad, S. M. S., Adnan, W. A. W., Yussof, S., Iranmanesh, V., and Malallah, F. L. “Estimating body related soft biometric traits in video frames”. In: *Scientific World Journal* 2014 (2014). DOI: 10.1155/2014/460973.
- [11] Artac, M., Jogan, M., and Leonardis, A. “Incremental PCA for on-line visual learning and recognition”. In: *Object recognition supported by user interaction for service robots*. Vol. 3. 2002, 781–784 vol.3. DOI: 10.1109/ICPR.2002.1048133.
- [12] Azuma, Y., Barat, P., Bartl, G., Bettin, H., Borys, M., Busch, I., Cibik, L., D’Agostino, G., Fujii, K., Fujimoto, H., Hioki, A., Krumrey, M., Kuetgens, U., Kuramoto, N., Mana, G., Massa, E., Meeß, R., Mizushima, S., Narukawa, T., Nicolaus, A., Pramann, A., Rabb, S. A., Rienitz, O., Sasso, C., Stock, M., Vocke, R. D., Waseda, A., Wundrack, S., and Zakel, S. “Improved measurement results for the Avogadro constant using a Si-enriched crystal”. In: *Metrologia* 52.2 (2015), pp. 360–375. DOI: 10.1088/0026-1394/52/2/360.
- [13] Balan, A. O. and Black, M. J. “The naked truth: Estimating body shape under clothing”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5303 LNCS (2008), pp. 15–29. DOI: 10.1007/978-3-540-88688-4_2.
- [14] Ballard, D. H. “Generalizing the Hough transform to detect arbitrary shapes”. In: *Pattern recognition* 13.2 (1981), pp. 111–122. DOI: 10.1016/0031-3203(81)90009-1.
- [15] Barba, C., Cavalli-Sforza, T., Cutter, J., Darnton-Hill, I., et al. “Appropriate body-mass index for Asian populations and its implications for policy and intervention strategies”. In: *The lancet* 363.9403 (2004), p. 157. DOI: 10.1016/S0140-6736(03)15268-3.
- [16] Barmpoutis, A. “Tensor Body: Real-time Reconstruction of the Human Body and Avatar Synthesis from RGB-D”. In: *IEEE Transactions on Cybernetics, Special issue on Computer Vision for RGB-D Sensors: Kinect and Its Applications* 43.5 (Oct. 2013), pp. 1347–1356. DOI: 10.1109/TCYB.2013.2276430.
- [17] Bastioni, M., Re, S., and Misra, S. “Ideas and Methods for Modeling 3D Human Figures: The Principal Algorithms Used by MakeHuman and Their Implementation in a New Approach to Parametric Modeling”. In: *Proceedings of the 1st Bangalore Annual Compute Conference*. COMPUTE ’08. New York, NY, USA: ACM, 2008, 10:1–10:6. DOI: 10.1145/1341771.1341782.
- [18] Benjamin, E. J., Blaha, M. J., Chiuve, S. E., Cushman, M., Das, S. R., Deo, R., Ferranti, S. D. de, Floyd, J., Fornage, M., Gillespie, C., Isasi, C. R., Jiménez, M. C., Jordan, L. C., Judd, S. E., Lackland, D., Lichtman, J. H., Lisabeth, L., Liu, S., Longenecker, C. T., Mackey, R. H., Matsushita, K., Mozaffarian, D., Mussolino, M. E., Nasir, K., Neumar, R. W., Palaniappan, L., Pandey, D. K., Thiagarajan, R. R., Reeves, M. J., Ritchey, M., Rodriguez, C. J., Roth, G. A., Rosamond, W. D., Sasson, C., Towfighi, A., Tsao, C. W., Turner, M. B., Virani, S. S., Voeks, J. H., Willey, J. Z., Wilkins, J. T., Wu, J. H., Alger, H. M., Wong, S. S., and Muntner, P. “Heart Disease and Stroke Statistics—2017 Update: A Report From the American Heart Association”. In: *Circulation* (2017). DOI: 10.1161/CIR.0000000000000485.

- [19] Bentley, J. L. “Multidimensional Binary Search Trees Used for Associative Searching”. In: *Commun. ACM* 18.9 (Sept. 1975), pp. 509–517. DOI: 10.1145/361002.361007.
- [20] Berg, M. de, Kreveld, M. van, Overmars, M., and Schwarzkopf, O. C. “Delaunay Triangulations”. In: *Computational Geometry: Algorithms and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 183–210. DOI: 10.1007/978-3-662-03427-9.
- [21] Bernstein, M. and Repinski, G. *Medical image scanner with automatic patient weight determination*. US Patent 6,026,318. Feb. 2000.
- [22] Besl, P. J. and McKay, N. D. “A Method for Registration of 3-D Shapes”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 14.2 (Feb. 1992), pp. 239–256. DOI: 10.1109/34.121791.
- [23] Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [24] Blanchette, J. and Summerfield, M. *C++ GUI Programming with Qt 4*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2006.
- [25] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Institute, Amsterdam: Blender Foundation. URL: <http://www.blender.org>.
- [26] Board, C. A. S. *Aviation Occurrence Report, Arrow Air Inc. Douglas DC-8-63 N950JW, Gander International Airport, Newfoundland, 12 December 1985*. 1988.
- [27] Bois, D. “Clinical calorimetry: Tenth paper a formula to estimate the approximate surface area if height and weight be known”. In: *Archives of Internal Medicine* XVII (1916), pp. 863–871. DOI: 10.1001/archinte.1916.00080130010002.
- [28] Bradski, A. *Learning OpenCV*. 1. ed. Gary Bradski and Adrian Kaehler. O’Reilly Media, 2008.
- [29] Bray, B. D., Campbell, J., Cloud, G. C., Hoffman, A., Tyrrell, P. J., Wolfe, C. D. A., and Rudd, A. G. “Bigger, faster?: Associations between hospital thrombolysis volume and speed of thrombolysis administration in acute ischemic stroke”. In: *Stroke* 44.11 (2013), pp. 3129–3135. DOI: 10.1161/STROKEAHA.113.001981.
- [30] Brenner, D. J. and Hall, E. J. *Cancer risks from CT scans: now we have data, what next?* 2012. DOI: 10.1148/radiol.12121248.
- [31] Breuer, L., Nowe, T., Huttner, H., Blinzler, C., Kollmar, R., Schellinger, P., Schwab, S., and Körmann, M. “Weight approximation in stroke before thrombolysis: The WAIST-Study: a prospective observational quot dose-finding study”. In: *Stroke* 41.12 (2010). DOI: 10.1161/STROKEAHA.110.578062.
- [32] Broselow, J. B. *Color-coded medical dosing container*. CA Patent 2,494,314. 2009.
- [33] Broselow, J. B. *Measuring tape for directly determining physical treatment and physiological values*. US Patent 4,713,888. 1987.

- [34] Brožek, J., Grande, F., Anderson, J. T., and Keys, A. “Densitometric analysis of body composition: revision of some quantitative assumptions”. In: *Annals of the New York Academy of Sciences* 110.1 (1963), pp. 113–140. DOI: 10.1111/j.1749-6632.1963.tb17079.x.
- [35] Butler, D. A., Izadi, S., Hilliges, O., Molyneaux, D., Hodges, S., and Kim, D. “Shake’N’Sense: Reducing Interference for Overlapping Structured Light Depth Cameras”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’12. New York, NY, USA: ACM, 2012, pp. 1933–1936. DOI: 10.1145/2207676.2208335.
- [36] Bykat, A. “Convex hull of a finite set of points in two dimensions”. In: *Information Processing Letters* 7.6 (1978), pp. 296–298. DOI: 10.1016/0020-0190(78)90021-2.
- [37] Cannon, C. P. “Thrombolysis medication errors: benefits of bolus thrombolytic agents”. In: *The American journal of cardiology* 85.8 (2000), pp. 17–22. DOI: 10.1016/S0002-9149(00)00874-2.
- [38] Canny, J. “A Computational Approach to Edge Detection”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 8.6 (1986), pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.
- [39] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. “Realtime multi-person 2d pose estimation using part affinity fields”. In: *arXiv preprint arXiv:1611.08050* (2016). DOI: 10.1109/CVPR.2017.143.
- [40] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. A. *Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*. Tech. rep. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University, Apr. 2017.
- [41] Chen-Scarabelli, C., Scarabelli, T. M., Ellenbogen, K. A., and Halperin, J. L. “Device-Detected Atrial Fibrillation: What to Do With Asymptomatic Patients?” In: *Journal of the American College of Cardiology* 65.3 (2015), pp. 281–294. DOI: 10.1016/j.jacc.2014.10.045.
- [42] Coe, T., Halkes, M, Houghton, K, and Jefferson, D. “The accuracy of visual estimation of weight and height in pre-operative supine patients”. In: *Anaesthesia* 54.6 (1999), pp. 582–586. DOI: 10.1046/j.1365-2044.1999.00838.x.
- [43] Community, D. O. *Doxygen – Generate documentation from source code*. [Online; accessed 02-October-2017]. 2017. URL: <http://www.stack.nl/~dimitri/doxygen/index.html>.
- [44] Cook, T. S., Couch, G., Couch, T. J., Kim, W., and Boonn, W. W. “Using the Microsoft Kinect for Patient Size Estimation and Radiation Dose Normalization: Proof of Concept and Initial Validation”. In: *J. Digital Imaging* 26.4 (2013), pp. 657–662. DOI: 10.1007/s10278-012-9567-2.
- [45] Corke, P. I. *Robotics, Vision and Control: Fundamental Algorithms in {MATLAB}@*. Springer tracts in advanced robotics. Berlin: Springer, 2011. DOI: 10.1007/978-3-319-54413-7.
- [46] Cubison, T. and Gilbert, P. “So much for percentage, but what about the weight?” In: *Emergency medicine journal* 22.9 (2005), pp. 643–645. DOI: 10.1136/emj.2003.011304.

- [47] Dagum, L. and Menon, R. “OpenMP: an industry standard API for shared-memory programming”. In: *Computational Science & Engineering, IEEE* 5.1 (1998), pp. 46–55. DOI: 10.1109/99.660313.
- [48] Dankert, J and Dankert, H. *Technische Mechanik: Statik, Festigkeitslehre, Kinematik / Kinetik*. Studium : Mechanik. Vieweg + Teubner, 2009. DOI: 10.1007/978-3-8348-2235-2.
- [49] Dantcheva, A., Velardo, C., D’angelo, A., and Dugelay, J.-L. “Bag of soft biometrics for person identification : New trends and challenges”. In: *Multimedia Tools and Applications, Springer* (2010). DOI: 10.1016/j.patrec.2005.08.018.
- [50] Davies, E. R. *Computer and machine vision theory, algorithms, practicalities*. Waltham, Mass.: Elsevier, 2012.
- [51] Davis, R. “The SI unit of mass”. In: *Metrologia* 40.6 (2003), pp. 299–305. DOI: 10.1088/0026-1394/40/6/001.
- [52] Dempster, P. and Aitkens, S. “A new air displacement method for the determination of human body composition”. In: *Medicine and science in sports and exercise* 27.12 (1995), pp. 1692–1697. DOI: 10.1249/00005768-199512000-00017.
- [53] Deng, B.-C., Yun, Y.-H., Liang, Y.-Z., Cao, D.-S., Xu, Q.-S., Yi, L.-Z., and Huang, X. “A new strategy to prevent over-fitting in partial least squares models based on model population analysis”. In: *Analytica Chimica Acta* 880.Supplement C (2015), pp. 32–41. DOI: 10.1016/j.aca.2015.04.045.
- [54] Determann, R. M., Wolthuis, E. K., Spronk, P. E., Kuiper, M. A., Korevaar, J. C., Vroom, M. B., and Schultz, M. J. “Reliability of height and weight estimates in patients acutely admitted to intensive care units”. In: *Critical care nurse* 27.5 (2007), pp. 48–55.
- [55] Deutschland, S. B. *Mikrozensus - Fragen zur Gesundheit 2009*. URL: <https://www.destatis.de/DE/Publikationen/Thematisch/Gesundheit/Gesundheitszustand/Koerpermasse.html>.
- [56] Diedler, J., Ahmed, N., Glahn, J., Grond, M., Lorenzano, S., Brozman, M., Sykora, M., and Ringleb, P. “Is the maximum dose of 90 mg alteplase sufficient for patients with ischemic stroke weighing greater 100 kg?” In: *Stroke* 42.6 (2011), pp. 1615–1620. DOI: 10.1161/STROKEAHA.110.603514.
- [57] Django. *Django (Version 1.5)*. 2013. URL: <https://djangoproject.com>.
- [58] Durnin, J. and Womersley, J. “Body fat assessed from total body density and its estimation from skinfold thickness: measurements on 481 men and women aged from 16 to 72 years”. In: *British Journal of Nutrition* 32.01 (1974), pp. 77–97. DOI: 10.1079/BJN19740060.
- [59] Eberly, D. “Least squares fitting of data”. In: *Chapel Hill, NC: Magic Software* (2000).
- [60] Ebner, N. C. and Fischer, H. “Emotion and aging: evidence from brain and behavior”. In: *Frontiers in Psychology* 5 (Sept. 2014), p. 996. DOI: 10.3389/fpsyg.2014.00996.

- [61] El-laithy, R. A., Huang, J., and Yeh, M. “Study on the use of Microsoft Kinect for robotics applications”. In: *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*. May 2012, pp. 1280–1288. DOI: 10.1109/PLANS.2012.6236985.
- [62] Elliott, A. F. *Why a Finnish airline is weighing passengers before they board*. Nov. 2017. URL: <http://www.telegraph.co.uk/travel/news/why-a-finnish-airline-is-weighing-every-passenger-before-they-board/>.
- [63] Fernandes, C., Clark, S., Price, A., and Innes, G. “How accurately do we estimate patients’ weight in emergency departments?” In: *Canadian Family Physician* 45 (1999), p. 2373. DOI: 10.1111/j.1742-6723.2005.00701.x.
- [64] Fischler, M. A. and Bolles, R. C. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395. DOI: 10.1145/358669.358692.
- [65] Fleishman, S., Cohen-Or, D., and Silva, C. T. “Robust Moving Least-squares Fitting with Sharp Features”. In: *ACM SIGGRAPH 2005 Papers*. SIGGRAPH ’05. New York, NY, USA: ACM, 2005, pp. 544–552. DOI: 10.1145/1186822.1073227. URL: <http://doi.acm.org/10.1145/1186822.1073227>.
- [66] Fonarow, G. C., Smith, E. E., Saver, J. L., Reeves, M. J., Hernandez, A. F., Peterson, E. D., Sacco, R. L., and Schwamm, L. H. “Improving door-to-needle times in acute ischemic stroke: The design and rationale for the American Heart Association/American Stroke Association’s target: Stroke initiative”. In: *Stroke* 42.10 (2011), pp. 2983–2989. DOI: 10.1161/STROKEAHA.111.621342.
- [67] Foster, E. D. and Deardorff, A. “Open Science Framework (OSF)”. In: *Journal of the Medical Library Association : JMLA* 105.2 (Apr. 2017), pp. 203–206. DOI: 10.5195/jmla.2017.88.
- [68] Freedman, J. E., Becker, R. C., Adams, J. E., Borzak, S., Jesse, R. L., Newby, L. K., O’Gara, P., Pezzullo, J. C., Kerber, R., Coleman, B., et al. “Medication Errors in Acute Cardiac Care An American Heart Association Scientific Statement From the Council on Clinical Cardiology Subcommittee on Acute Cardiac Care, Council on Cardiopulmonary and Critical Care, Council on Cardiovascular Nursing, and Council on Stroke”. In: *Circulation* 106.20 (2002), pp. 2623–2629. DOI: 10.1161/01.CIR.0000037748.19282.7D.
- [69] Fuchs, S. and Hirzinger, G. “Extrinsic and depth calibration of ToF-cameras”. In: *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR* (2008), pp. 1–6. DOI: 10.1109/CVPR.2008.4587828.
- [70] Gascho, D., Ganzoni, L., Kolly, P., Zoelch, N., Hatch, G. M., Thali, M. J., and Ruder, T. D. “A new method for estimating patient body weight using CT dose modulation data”. In: *European Radiology Experimental* 1.1 (Dec. 2017), p. 23. DOI: 10.1186/s41747-017-0028-z.
- [71] Gehan, E. A. and George, S. L. “Estimation of human body surface area from height and weight.” In: *Cancer chemotherapy reports. Part 1* 54.4 (1970), pp. 225–235.

- [72] Gonzalez-Jorge, H, Rodríguez-Gonzálvez, P, Martínez-Sánchez, J, González-Aguilera, D, Arias, P, Gesto, M, and Díaz-Vilariño, L. “Metrological comparison between Kinect I and Kinect II sensors”. In: *Measurement* 70.Complete (2015), pp. 21–26. DOI: 10.1016/j.measurement.2015.03.042.
- [73] Graf, A. and Wichmann, F. “Gender classification of human faces”. In: *Biologically Motivated Computer Vision*. Springer. 2002, pp. 1–18. DOI: 10.13140/RG.2.2.26891.90407.
- [74] Graham, R. L. “An efficient algorithm for determining the convex hull of a finite planar set”. In: *Information processing letters* 1.4 (1972), pp. 132–133. DOI: 10.1016/0020-0190(72)90045-2.
- [75] Gschwandtner, M., Kwitt, R., Uhl, A., and Pree, W. “BlenSor: Blender Sensor Simulation Toolbox”. In: *Advances in Visual Computing: 7th International Symposium, ISVC 2011, Las Vegas, NV, USA, September 26-28, 2011. Proceedings, Part II*. Ed. by Bebis, G., Boyle, R., Parvin, B., Koracin, D., Wang, S., Kyungnam, K., Benes, B., Moreland, K., Borst, C., DiVerdi, S., Yi-Jen, C., and Ming, J. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 199–208. DOI: 10.1007/978-3-642-24031-7_20.
- [76] Hacke, W., Kaste, M., Bluhmki, E., Brozman, M., Dávalos, A., Guidetti, D., Larrue, V., Lees, K. R., Medeghri, Z., Machnig, T., et al. “Thrombolysis with alteplase 3 to 4.5 hours after acute ischemic stroke”. In: *New England Journal of Medicine* 359.13 (2008), pp. 1317–1329. DOI: 10.1056/NEJMoa0804656.
- [77] Hackel, T., Wegner, J. D., and Schindler, K. “Contour Detection in Unstructured 3D Point Clouds”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 1610–1618. DOI: 10.1109/CVPR.2016.178.
- [78] Hahnel, D., Thrun, S., and Burgard, W. “An Extension of the ICP Algorithm for Modeling Nonrigid Objects with Mobile Robots”. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence*. IJCAI’03. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 915–920.
- [79] Harris, C. and Stephens, M. “A combined corner and edge detector.” In: *Alvey vision conference*. Vol. 15. 50. Manchester, UK. 1988, pp. 10–5244. DOI: 10.5244/c.2.23.
- [80] Hastie, T. J., Tibshirani, R. J., and Friedman, J. H. *The elements of statistical learning : data mining, inference, and prediction*. Springer series in statistics. New York: Springer, 2009.
- [81] Haycock, G. B., Schwartz, G. J., and Wisotsky, D. H. “Geometric method for measuring body surface area: a height-weight formula validated in infants, children, and adults”. In: *The Journal of pediatrics* 93.1 (1978), pp. 62–66. DOI: 10.1016/S0022-3476(78)80601-5.
- [82] He, K., Zhang, X., Ren, S., and Sun, J. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

- [83] Hertzberg, J., Lingemann, K., and Nüchter, A. *Mobile Roboter: Eine Einführung aus Sicht der Informatik*. eXamen.press. Berlin: Springer Vieweg, 2012. DOI: 10.1007/978-3-642-01726-1.
- [84] Heymsfield, S. B., Wang, Z., Baumgartner, R. N., Dilmanian, F. A., Ma, R., and Yasumura, S. “Body composition and aging: a study by in vivo neutron activation analysis.” In: *The Journal of nutrition* 123.2 Suppl (1993), pp. 432–437. DOI: 10.1093/jn/123.suppl_2.432.
- [85] Holz, D. and Behnke, S. “Fast Range Image Segmentation and Smoothing Using Approximate Surface Reconstruction and Region Growing”. In: *Intelligent Autonomous Systems 12: Volume 2 Proceedings of the 12th International Conference IAS-12, held June 26-29, 2012, Jeju Island, Korea*. Ed. by Lee, S., Cho, H., Yoon, K.-J., and Lee, J. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 61–73. DOI: 10.1007/978-3-642-33932-5_7.
- [86] Holz, D., Holzer, S., Rusu, R. B., and Behnke, S. “Real-Time Plane Segmentation Using RGB-D Cameras”. In: *RoboCup 2011: Robot Soccer World Cup XV*. Ed. by Röfer, T., Mayer, N. M., Savage, J., and Saranlı, U. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 306–317. DOI: 10.1007/978-3-642-32060-6_26.
- [87] Holzer, S., Rusu, R. B., Dixon, M., Gedikli, S., and Navab, N. “Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Oct. 2012, pp. 2684–2689. DOI: 10.1109/IR0S.2012.6385999.
- [88] Hu, M., Zhai, G., Li, D., Fan, Y., Duan, H., Zhu, W., and Yang, X. “Combination of near-infrared and thermal imaging techniques for the remote and simultaneous measurements of breathing and heart rates under sleep situation”. In: *PloS one* 13.1 (2018). DOI: <https://doi.org/10.1371/journal.pone.0190466>.
- [89] Huang, T., Yang, G., and Tang, G. “A fast two-dimensional median filtering algorithm”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27.1 (Feb. 1979), pp. 13–18. DOI: 10.1109/TASSP.1979.1163188.
- [90] Hussmann, S., Edeler, T., and Hermanski, A. “Real-Time Processing of 3D-TOF Data in Machine Vision Applications”. In: *Machine Vision-Applications and Systems*. InTech, 2012.
- [91] *Infrarotkamera optris PI 400 / PI 450*. URL: <https://www.optris.de/infrarotkamera-pi400>.
- [92] Ioffe, S. and Szegedy, C. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” In: *ICML*. Ed. by Bach, F. R. and Blei, D. M. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 448–456.

- [93] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. “KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera”. In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST '11. New York, NY, USA: ACM, 2011, pp. 559–568. DOI: 10.1145/2047196.2047270.
- [94] Jang, H. Y., Shin, S. D., and Kwak, Y. H. “Can the Broselow Tape Be Used to Estimate Weight and Endotracheal Tube Size in Korean Children?” In: *Academic Emergency Medicine* 14.5 (2007), pp. 489–491. DOI: 10.1197/j.aem.2006.12.014.
- [95] Jauch, E. C., Saver, J. L., Adams, H. P., Bruno, A., Connors, J. J. B., Demaerschalk, B. M., Khatri, P., McMullan, P. W., Qureshi, A. I., Rosenfield, K., Scott, P. A., Summers, D. R., Wang, D. Z., Wintermark, M., and Yonas, H. “Guidelines for the early management of patients with acute ischemic stroke: A guideline for healthcare professionals from the American Heart Association/American Stroke Association”. In: *Stroke* 44.3 (2013), pp. 870–947. DOI: 10.1161/STR.0b013e318284056a.
- [96] Jin, S., Lewis, R. R., and West, D. “A comparison of algorithms for vertex normal computation”. In: *The Visual Computer* 21.1 (Feb. 2005), pp. 71–82. DOI: 10.1007/s00371-004-0271-1.
- [97] John, C. *System and method for measuring animals*. 2011.
- [98] Jolliffe, I. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002.
- [99] Jovin, T. G., Gupta, R., Uchino, K., Jungreis, C. A., Wechsler, L. R., Hammer, M. D., Tayal, A., and Horowitz, M. B. “Emergent Stenting of Extracranial Internal Carotid Artery Occlusion in Acute Stroke Has a High Revascularization Rate”. In: *Stroke* 36.11 (2005), pp. 2426–2430. DOI: 10.1161/01.STR.0000185924.22918.51.
- [100] Kang, D. W., Chalela, J. A., Dunn, W., and Warach, S. “MRI screening before standard tissue plasminogen activator therapy is feasible and safe”. In: *Stroke* 36.9 (2005), pp. 1939–1943. DOI: 10.1161/01.STR.0000177539.72071.f0.
- [101] Kessler, W. *Multivariate Datenanalyse: für die Pharma, Bio- und Prozessanalytik*. Wiley, 2006. DOI: 10.1002/9783527610037.
- [102] Khoshelham, K. and Elberink, S. O. “Accuracy and resolution of kinect depth data for indoor mapping applications”. In: *Sensors* 12.2 (2012), pp. 1437–1454. DOI: 10.3390/s120201437.
- [103] Kim, S. K., Lee, S. Y., Bae, H. J., Lee, Y. S., Kim, S. Y., Kang, M. J., and Cha, J. K. “Pre-hospital notification reduced the door-to-needle time for iv t-PA in acute ischaemic stroke”. In: *European Journal of Neurology* 16.12 (2009), pp. 1331–1335. DOI: 10.1111/j.1468-1331.2009.02762.x.
- [104] Klasing, K., Althoff, D., Wollherr, D., and Buss, M. “Comparison of surface normal estimation methods for range sensing applications.” In: *ICRA*. IEEE, 2009, pp. 3206–3211. DOI: 10.1109/ROBOT.2009.5152493.

- [105] Klir, G. J. and Yuan, B. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1995.
- [106] Kocabey, E., Camurcu, M., Offi, F., Aytar, Y., Marin, J., Torralba, A., and Weber, I. “Face-to-bmi: Using computer vision to infer body mass index on social media”. In: *arXiv preprint arXiv:1703.03156* (2017).
- [107] Koecher, M. and Krieg, A. *Ebene Geometrie*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2007.
- [108] Kongsro, J. “Estimation of Pig Weight Using a Microsoft Kinect Prototype Imaging System”. In: *Comput. Electron. Agric.* 109.C (Nov. 2014), pp. 32–35. DOI: 10.1016/j.compag.2014.08.008.
- [109] Kubat, M. *An Introduction to Machine Learning*. 1st. Springer Publishing Company, Incorporated, 2015. DOI: 10.1007/978-3-319-20010-1.
- [110] Kuchling, H. *Taschenbuch der Physik*. Fachbuchverl. Leipzig im Carl-Hanser-Verlag, 2007.
- [111] Kwah, L. K. and Diong, J. “National Institutes of Health Stroke Scale (NIHSS)”. In: *Journal of Physiotherapy* 60.1 (2014), p. 61. DOI: 10.1016/j.jphys.2013.12.012.
- [112] Labati, R., Genovese, A., Piuri, V., and Scotti, F. “Weight Estimation from Frame Sequences Using Computational Intelligence Techniques”. In: *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSAs)* (2012), pp. 29–34. DOI: 10.1109/CIMSAs.2012.6269603.
- [113] Lachat, E., Macher, H., Mittet, M. A., Landes, T., and Grussenmeyer, P. “First experiences with kinect V2 sensor for close range 3D modelling”. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* 40.5W4 (2015), pp. 93–100. DOI: 10.5194/isprsarchives-XL-5-W4-93-2015.
- [114] Langhorne, P., Pollock, A., and Others. “What are the components of effective stroke unit care?” In: *Age and ageing* 31.5 (2002), pp. 365–371. DOI: 10.1093/ageing/31.5.365.
- [115] Lawrence, S., Giles, C. L., and Tsoi, A. C. *What size neural network gives optimal generalization? Convergence properties of backpropagation*. Tech. rep. Institute for Advanced Computer Studies University of Maryland College Park, 1998.
- [116] Learmonth, S. R., Ireland, A., Mckiernan, C. J., and Burton, P. “Does initiation of an ambulance pre-alert call reduce the door to needle time in acute myocardial infarct?” In: *Emergency Medicine Journal* (2006), pp. 79–81. DOI: 10.1136/emj.2004.022376.
- [117] LeCun, Y., Bengio, Y., and Hinton, G. “Deep learning”. In: *Nature* 521.7553 (May 2015), pp. 436–444. DOI: DOI:10.1038/nature14539.
- [118] Lees, K. R., Bluhmki, E., Von Kummer, R., Brott, T. G., Toni, D., Grotta, J. C., Albers, G. W., Kaste, M., Marler, J. R., Hamilton, S. A., et al. “Time to treatment with intravenous alteplase and outcome in stroke: an updated pooled analysis of ECASS, ATLANTIS, NINDS, and EPITHET trials”. In: *The Lancet* 375.9727 (2010), pp. 1695–1703. DOI: 10.1016/S0140-6736(10)60491-6.

- [119] Leonard, J. and Kramer, M. “Improvement of the backpropagation algorithm for training neural networks”. In: *Computers & Chemical Engineering* 14.3 (1990), pp. 337–341.
- [120] Levi, G and Hassner, T. “Age and gender classification using convolutional neural networks”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2015, pp. 34–42. DOI: 10.1109/CVPRW.2015.7301352.
- [121] Linder, T., Wehner, S., and Arras, K. O. “Real-Time Full-Body Human Gender Recognition in (RGB)-D Data”. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 3039–3045. DOI: 10.1109/ICRA.2015.7139616.
- [122] Liptak, B. G. *Instrument Engineers’ Handbook, Third Edition, Volume Three: Process Software and Digital Networks*. CRC Press, 2002.
- [123] Lopez, R. *Open NN User’s Guide*. 2015.
- [124] Lorenz, M. W., Graf, M., Henke, C., Hermans, M., Ziemann, U., Sitzer, M., and Foerch, C. “Anthropometric approximation of body weight in unresponsive stroke patients”. In: *Journal of Neurology, Neurosurgery & Psychiatry* 78.12 (2007), pp. 1331–1336. DOI: 10.1136/jnnp.2007.117150.
- [125] Luhmann, T., Piechel, J., and Roelfs, T. “Geometric Calibration of Thermographic Cameras”. In: *Thermal Infrared Remote Sensing: Sensors, Methods, Applications*. Ed. by Kuenzer, C. and Dech, S. Dordrecht: Springer Netherlands, 2013, pp. 27–42. DOI: 10.1007/978-94-007-6639-6_2.
- [126] Lussier, J. T. and Thrun, S. “Automatic calibration of RGBD and thermal cameras”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE. 2014, pp. 451–458. DOI: 10.1109/IROS.2014.6942598.
- [127] *MAGNETOM Sola*. URL: <https://www.healthcare.siemens.de/magnetic-resonance-imaging/0-35-to-1-5t-mri-scanner/magnetom-sola>.
- [128] Mahadevan, S. and Ali, I. “Is body mass index a good indicator of obesity?” In: *International Journal of Diabetes in Developing Countries* 36.2 (June 2016), pp. 140–142. DOI: 10.1007/s13410-016-0506-5.
- [129] Marr, D. and Hildreth, E. “Theory of edge detection”. In: *Proceedings of the Royal Society of London B: Biological Sciences* 207.1167 (1980), pp. 187–217. DOI: 10.1098/rspb.1980.0020.
- [130] Martin, D. R., Soria, D. M., Brown, C. G., Pepe, P. E., Gonzalez, E., Jastremski, M., Stueven, H., and Cummins, R. O. “Agreement between paramedic-estimated weights and subsequent hospital measurements in adults with out-of-hospital cardiac arrest”. In: *Pre-hospital and disaster medicine* 9.01 (1994), pp. 54–56. DOI: 10.1017/S1049023X0004996.
- [131] *MATLAB version 8.5.0.197613 (R2015a)*. The Mathworks, Inc. Natick, Massachusetts, 2015.
- [132] May, S. *3D Time-of-Flight Ranging for Robotic Perception in Dynamic Environments*. VDI-Verlag, 2009.

- [133] May, S., Werner, B., Surmann, H., and Pervözl, K. “3D time-of-flight cameras for mobile robotics.” In: *IROS*. IEEE, 2006, pp. 790–795. DOI: 10.1109/IROS.2006.281670.
- [134] Mcculloch, W. and Pitts, W. “A Logical Calculus of Ideas Immanent in Nervous Activity”. In: *Bulletin of Mathematical Biophysics* 5 (1943), pp. 127–147. DOI: 10.1007/BF02478259.
- [135] Menon, S. and Kelly, A.-M. “How accurate is weight estimation in the emergency department?” In: *Emergency Medicine Australasia* 17.2 (2005), pp. 113–116. DOI: 10.1111/j.1742-6723.2005.00701.x.
- [136] Mensah, G. A., Organization, W. H., (U.S.), C. f. D. C., and Prevention, eds. *The atlas of heart disease and stroke*. Geneva: World Health Organization, 2004.
- [137] Mohammad, Y. M. “Mode of Arrival to the Emergency Department of Stroke Patients in the United States”. In: *Journal of Vascular and Interventional Neurology* 1.3 (July 2008), pp. 83–86.
- [138] Moré, J. J. “The Levenberg-Marquardt algorithm: implementation and theory”. In: *Numerical analysis*. Springer, 1978, pp. 105–116. DOI: 10.1007/BFb0067700.
- [139] Morissette, S. *Livestock weight estimation device*. US Patent 6,314,654. 2001.
- [140] Mosley, I., Nicol, M., Donnan, G., Patrick, I., and Dewey, H. “The Impact of Ambulance Practice on Acute Stroke Care”. In: *Stroke* 38.2 (2007), pp. 361–366. DOI: 10.1161/01.STR.0000254528.17405.cc.
- [141] Mosteller, R. “Simplified calculation of body-surface area.” In: *The New England journal of medicine* 317.17 (1987), p. 1098. DOI: 10.1056/NEJM198710223171717.
- [142] Mozaffarian, D., Benjamin, E. J., Go, A. S., Arnett, D. K., Blaha, M. J., Cushman, M., Das, S. R., Ferranti, S. de, Després, J.-P., Fullerton, H. J., Howard, V. J., Huffman, M. D., Isasi, C. R., Jiménez, M. C., Judd, S. E., Kissela, B. M., Lichtman, J. H., Lisabeth, L. D., Liu, S., Mackey, R. H., Magid, D. J., McGuire, D. K., Mohler, E. R., Moy, C. S., Muntner, P., Mussolino, M. E., Nasir, K., Neumar, R. W., Nichol, G., Palaniappan, L., Pandey, D. K., Reeves, M. J., Rodriguez, C. J., Rosamond, W., Sorlie, P. D., Stein, J., Towfighi, A., Turan, T. N., Virani, S. S., Woo, D., Yeh, R. W., and Turner, M. B. “Executive Summary: Heart Disease and Stroke Statistics—2016 Update”. In: *Circulation* 133.4 (Jan. 2016), pp. 447–454. DOI: 10.1161/CIR.0000000000000366.
- [143] Nahavandi, D, Abobakr, A, Haggag, H, Hossny, M, Nahavandi, S, and Filippidis, D. “A skeleton-free kinect system for body mass index assessment using deep neural networks”. In: *Systems Engineering Symposium (ISSE), 2017 IEEE International*. IEEE. 2017, pp. 1–6. DOI: 10.1109/SysEng.2017.8088252.
- [144] Nguyen, C. V., Izadi, S., and Lovell, D. “Modeling kinect sensor noise for improved 3D reconstruction and tracking”. In: *Proceedings - 2nd Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT 2012* (2012), pp. 524–530. DOI: 10.1109/3DIMPVT.2012.84.

- [145] Nguyen, T. V., Feng, J., and Yan, S. “Seeing Human Weight from a Single RGB-D Image”. In: *Journal of Computer Science and Technology* 29.5 (2014), pp. 777–784. DOI: 10.1007/s11390-014-1467-0.
- [146] Nickolls, J., Buck, I., Garland, M., and Skadron, K. “Scalable parallel programming with CUDA”. In: *Queue* 6.2 (2008), pp. 40–53.
- [147] Nissen, S. *Implementation of a Fast Artificial Neural Network Library (fann)*. Tech. rep. <http://fann.sf.net>. Department of Computer Science University of Copenhagen (DIKU), 2003.
- [148] Nowinski, W. L., Qian, G., and Hanley, D. F. “A CAD System for Hemorrhagic Stroke”. In: *The Neuroradiology Journal* 27.4 (Sept. 2014), pp. 409–416. DOI: 10.15274/NRJ-2014-10080.
- [149] Nüchter, A. “Semantische dreidimensionale Karten für autonome mobile Roboter.” PhD thesis. Rheinische Friedrich-Wilhelms-Universität Bonn, 2006, pp. 1–164.
- [150] Obdrzalek, S., Kurillo, G., Ofli, F., Bajcsy, R., Seto, E., Jimison, H., and Pavel, M. “Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population”. In: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Aug. 2012, pp. 1188–1193. DOI: 10.1109/EMBC.2012.6346149.
- [151] OpenNI Online Community. *OpenNI Documentation*. <https://structure.io/openni>. (Accessed on 06/07/2014).
- [152] Paddon, C. and Wetzell, D. *Medical patient weighing scale*. US Patent 4,363,368. Dec. 1982.
- [153] Palmero, C., Clapés, A., Bahnsen, C., Møgelmoose, A., Moeslund, T. B., and Escalera, S. “Multi-modal RGB–depth–thermal human body segmentation”. In: *International Journal of Computer Vision* 118.2 (2016), pp. 217–239. DOI: <https://doi.org/10.1007/s11263-016-0901-x>.
- [154] Pawar, A. Y., Sonawane, D. D., Erande, K. B., and Derle, D. V. “Terahertz technology and its applications”. In: *Drug Invention Today* 5.2 (2013), pp. 157–163. DOI: <https://doi.org/10.1016/j.dit.2013.03.009>.
- [155] *PCL - point cloud library*. (Accessed on 09//2017). URL: <http://www.pointclouds.org/>.
- [156] Pearce, M. S., Salotti, J. A., Little, M. P., McHugh, K., Lee, C., Kim, K. P., Howe, N. L., Ronckers, C. M., Rajaraman, P., Craft, A. W., et al. “Radiation exposure from CT scans in childhood and subsequent risk of leukaemia and brain tumours: a retrospective cohort study”. In: *The Lancet* 380.9840 (2012), pp. 499–505. DOI: 10.1016/S0140-6736(12)60815-0.

- [157] Penne, J., Höller, K., Stürmer, M., Schrauder, T., Schneider, A., Engelbrecht, R., Feußner, H., Schmauss, B., and Hornegger, J. “Time-of-flight 3-D endoscopy”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009* (2009), pp. 467–474. DOI: 10.1007/978-3-642-04268-3_58.
- [158] Pfitzner, C., May, S., and Merkl, C. *Vorrichtung und Verfahren zur optischen Erfassung eines Gewichtes einer Person*. DE Patent App. DE201,610,103,543. Aug. 2017.
- [159] Pfitzner, C. *RGB-D(-T) Datasets for Body Weight Estimation of Stroke Patients from the Libra3D Project*. June 2017. DOI: 10.17605/OSF.IO/RHQ3M. URL: osf.io/rhq3m.
- [160] Pfitzner, C. “Robotic Vision in Medical Applications: Visual Weight Estimation for Emergency Patients”. In: *PhD Workshop of the IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*. 2015.
- [161] Pfitzner, C., May, S., and Nüchter, A. “Body Weight Estimation for Dose-Finding and Health Monitoring of Lying, Standing and Walking Patients Based on RGB-D Data”. In: *Sensors (Basel, Switzerland)* 18.5 (Apr. 2018). DOI: 10.3390/s18051311. URL: <https://doi.org/10.3390/s18051311>.
- [162] Pfitzner, C., May, S., and Nüchter, A. “Evaluation of Features from RGB-D Data for Human Body Weight Estimation”. In: *Proceedings of the 20th World Congress of the International Federation of Automatic Control* (2017). DOI: 10.1016/j.ifacol.2017.08.1761.
- [163] Pfitzner, C., May, S., Merkl, C., Breuer, L., Kohrmann, M., Braun, J., Dirauf, F., and Nüchter, A. “Libra3D: Body weight estimation for emergency patients in clinical environments with a 3D structured light sensor”. In: *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*. 2015, pp. 2888–2893. DOI: 10.1109/ICRA.2015.7139593.
- [164] Pfitzner, C., May, S., and Nüchter, A. “Neural network-based visual body weight estimation for drug dosage finding”. In: *Proceedings of the SPIE Medical Imaging 2016* (2016).
- [165] Pirker, K., Rüter, M., Bischof, H., and Skrabal, F. *Human Body Volume Estimation in a Clinical Environment*. 2010.
- [166] Pöhlmann, S. T. L., Harkness, E. F., Taylor, C. J., and Astley, S. M. “Evaluation of Kinect 3D Sensor for Healthcare Imaging”. In: *Journal of Medical and Biological Engineering* 36.6 (Dec. 2016), pp. 857–870. DOI: 10.1007/s40846-016-0184-2.
- [167] Pollarolo, A., Qu, J., Rogalla, H., Dresselhaus, P. D., and Benz, S. P. “The BIPM Watt Balance: Improvement and Developments”. In: *2010 Conference on Precision Electromagnetic Measurements*. January. Daejeon, Korea, 2010, pp. 490–491. DOI: 10.1088/0026-1394/21/4/007.
- [168] Popa, M., Sirbu, D., Curseu, D., and Ionutas, A. “The Measurement of Body Composition by bioelectrical Impedance”. In: *Automation, Quality and Testing, Robotics, 2006 IEEE International Conference on*. Vol. 2. 2006, pp. 437–440.

- [169] Poppinga, J., Vaskevicius, N., Birk, A., and Pathak, K. “Fast plane detection and polygonalization in noisy 3D range images”. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sept. 2008, pp. 3378–3383. DOI: 10.1109/IRoS.2008.4650729.
- [170] Powers, W. J., Rabinstein, A. A., Ackerson, T., Adeoye, O. M., Bambakidis, N. C., Becker, K., Biller, J., Brown, M., Demaerschalk, B. M., Hoh, B., et al. “2018 guidelines for the early management of patients with acute ischemic stroke: a guideline for healthcare professionals from the American Heart Association/American Stroke Association”. In: *Stroke* 49.3 (2018), e46–e110. DOI: 10.1161/STR.0000000000000158.
- [171] Procházka, A., Schätz, M., Vyšata, O., and Vališ, M. “Microsoft kinect visual and depth sensors for breathing and heart rate analysis”. In: *Sensors* 16.7 (2016), p. 996.
- [172] Publication, C. A. A. *Civil Aviation Safety Authority – Standard Passenger and Baggage Weights*. 1990.
- [173] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. “ROS: an open-source Robot Operating System”. In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe. 2009, p. 5.
- [174] Radiopaedia Online Community. *Radiopaedia*. URL: <http://radiopaedia.org/>.
- [175] Raichle, M. E. and Gusnard, D. A. “Appraising the brain’s energy budget”. In: *Proceedings of the National Academy of Sciences* 99.16 (2002), pp. 10237–10239. DOI: 10.1073/pnas.172399499.
- [176] Ramarajan, N., Krishnamoorthi, R., Strehlow, M., Quinn, J., and Mahadevan, S. V. “Internationalizing the Broselew tape: How reliable is weight estimation in Indian children”. In: *Academic Emergency Medicine* 15.5 (2008), pp. 431–436. DOI: 10.1111/j.1553-2712.2008.00081.x.
- [177] Rapp, H. “Experimental and Theoretical Investigation of Correlating TOF-Camera Systems”. MA thesis. IWR, Fakultät für Physik und Astronomie, University Heidelberg, 2007.
- [178] Robinson, M and Parkinson, M. B. “Estimating Anthropometry with Microsoft Kinect”. In: *Proceedings of the 2nd International Digital Human Modeling Symposium*. 2013.
- [179] Rusu, R. B. “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments”. In: *KI - Künstliche Intelligenz* (2010), pp. 1–4. DOI: /10.1007/s13218-010-0059-6.
- [180] Rusu, R. B. and Cousins, S. “3D is here: Point Cloud Library (PCL)”. In: *International Conference on Robotics and Automation*. Shanghai, China, 2011. DOI: 10.1109/ICRA.2011.5980567.
- [181] Sacco, J. J., Botten, J., Macbeth, F., Bagust, A., and Clark, P. “The average body surface area of adult cancer patients in the UK: a multicentre retrospective study”. In: *PloS one* 5.1 (2010), e8933. DOI: 10.1371/journal.pone.0008933.

- [182] Saver, J. L. “Time is brain - Quantified”. In: *Stroke* 37.1 (2006), pp. 263–266. DOI: 10.1161/01.STR.0000196957.55928.ab.
- [183] Saver, J. L., Smith, E. E., Fonarow, G. C., Reeves, M. J., Zhao, X., Olson, D. M., and Schwamm, L. H. “The ”golden hour” and acute brain ischemia: Presenting features and lytic therapy in >30 000 patients arriving within 60 minutes of stroke onset”. In: *Stroke* 41.7 (2010), pp. 1431–1439. DOI: 10.1161/STROKEAHA.110.583815.
- [184] Schwartz, B., Zaitsev, P., Tkachenko, V., Zawodny, J. D., Lentz, A., and Balling, D. J. *High Performance MySQL: Optimization, Backups, Replication, and Load-Balancing*. 2.A. O’Reilly Media, 2008.
- [185] Seguin, G., Alahari, K., Sivic, J., and Laptev, I. “Pose estimation and segmentation of multiple people in stereoscopic movies”. In: *IEEE transactions on pattern analysis and machine intelligence* 37.8 (2015), pp. 1643–1655. DOI: 10.1109/TPAMI.2014.2369050.
- [186] Sendroy, J and Collison, H. A. “Determination of human body volume from height and weight”. In: *J Appl Physiol* 21.1 (1966), pp. 167–72. DOI: 10.1152/jappl.1966.21.1.167.
- [187] Sharma, N. and Aggarwal, L. M. “Automated medical image segmentation techniques”. In: *Journal of Medical Physics / Association of Medical Physicists of India* 35.1 (Apr. 2010), pp. 3–14. DOI: 10.4103/0971-6203.58777.
- [188] Shieh, Y., Chang, C.-H., Shieh, M., Lee, T.-H., Chang, Y. J., Wong, H.-F., Chin, S. C., and Goodwin, S. “Computer-Aided Diagnosis of Hyperacute Stroke with Thrombolysis Decision Support Using a Contralateral Comparative Method of CT Image Analysis”. In: *Journal of Digital Imaging* 27.3 (June 2014), pp. 392–406. DOI: 10.1007/s10278-013-9672-x.
- [189] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. “Real-time Human Pose Recognition in Parts from Single Depth Images”. In: *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR ’11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1297–1304. DOI: 10.1109/CVPR.2011.5995316.
- [190] Shrivakshan, G., Chandrasekar, C, et al. “A comparison of various edge detection techniques used in image processing”. In: *IJCSI International Journal of Computer Science Issues* 9.5 (2012), pp. 272–276.
- [191] Siciliano, B. and Khatib, O. *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer-Verlag, 2007.
- [192] Simonyan, K. and Zisserman, A. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).
- [193] Sinanović, O., Mrkonjić, Z., Zukić, S., Vidović, M., and Imamović, K. “Post-stroke language disorders”. In: *Acta Clinica Croatica* 50.1 (2011), pp. 79–93.

- [194] Smith, E. E. and Von Kummer, R. “Door-to-needle times in acute ischemic stroke: How low can we go?” In: *Neurology* 79.4 (2012), pp. 296–297. DOI: 10.1212/WNL.0b013e31825d602e.
- [195] Smith, E. E., Shobha, N., Dai, D., Olson, D. M., Reeves, M. J., Saver, J. L., Hernandez, A. F., Peterson, E. D., Fonarow, G. C., and Schwamm, L. H. “Risk score for in-hospital ischemic stroke mortality derived and validated within the get with the guidelines-stroke program”. In: *Circulation* 122.15 (2010), pp. 1496–1504. DOI: 10.1161/CIRCULATIONAHA.109.932822.
- [196] *Soehnle Industrial Solutions*. URL: <https://www.soehnle-professional.com/en/products>.
- [197] Soehnle Industrial Solutions GmbH, ed. *BED SCALE 7711, Operating Manual*. Soehnle Industrial Solutions GmbH. 2016.
- [198] *SOMATOM Force*. URL: <https://www.healthcare.siemens.de/computed-tomography/dual-source-ct/somatom-force>.
- [199] Specht, D. F. “A general regression neural network”. In: *IEEE Transactions on Neural Networks* 2.6 (Nov. 1991), pp. 568–576. DOI: 10.1109/72.97934.
- [200] SS, R., Baraniuk, S., Parker, S., Wu, T., Bowry, R., and JC, G. “Implementing a mobile stroke unit program in the united states: Why, how, and how much?” In: *JAMA Neurology* 72.2 (Feb. 2015), pp. 229–234. DOI: 10.1001/jamaneurol.2014.3618.
- [201] Steiner, T., Al-Shahi Salman, R., Beer, R., Christensen, H., Cordonnier, C., Csiba, L., Forsting, M., Harnof, S., Klijn, C. J., Krieger, D., et al. “European Stroke Organisation (ESO) guidelines for the management of spontaneous intracerebral hemorrhage”. In: *International journal of stroke* 9.7 (2014), pp. 840–855. DOI: 10.1111/ijs.12309.
- [202] Stock, M. “The watt balance: determination of the Planck constant and redefinition of the kilogram”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 369.1953 (2011), pp. 3936–3953. DOI: 10.1098/rsta.2011.0184.
- [203] Stone, J. E., Gohara, D., and Shi, G. “OpenCL: A parallel programming standard for heterogeneous computing systems”. In: *Computing in science & engineering* 12.3 (2010), pp. 66–73. DOI: 10.1109/MCSE.2010.69.
- [204] Swersey, B. L. *Hospital bed with an integrated scale*. US Patent 4,793,428. 1988.
- [205] Szeliski, R. *Computer Vision: Algorithms and Applications*. 1st. New York, NY, USA: Springer-Verlag New York, Inc., 2010. DOI: 10.1007/978-1-84882-935-0.
- [206] Tang, S. S., Diamond, C., and Arouh, S. *Neural network drug dosage estimation*. US Patent 6,658,396. 2003.
- [207] Tatlisumak, T. “Is CT or MRI the Method of Choice for Imaging Patients With Acute Stroke? Why Should Men Divide if Fate Has United?” In: *Stroke* 33.9 (2002), pp. 2144–2145. DOI: 10.1161/01.STR.0000026862.42440.AA.
- [208] Toga, A. W. and Mazziotta, J. C. *Brain mapping: the methods*. Academic press, 2002.

- [209] Tomasi, C. and Manduchi, R. “Bilateral Filtering for Gray and Color Images”. In: *Proceedings of the Sixth International Conference on Computer Vision. ICCV '98*. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–846. DOI: 10.1109/ICCV.1998.710815.
- [210] Turk, M. and Pentland, A. “Eigenfaces for Recognition”. In: *J. Cognitive Neuroscience* 3.1 (Jan. 1991), pp. 71–86. DOI: 10.1162/jocn.1991.3.1.71.
- [211] Tyan, Y.-S., Wu, M.-C., Chin, C.-L., Kuo, Y.-L., Lee, M.-S., and Chang, H.-Y. “Ischemic stroke detection system with a computer-aided diagnostic ability using an unsupervised feature perception enhancement method”. In: *Journal of Biomedical Imaging* 2014 (2014), p. 19. DOI: 10.1155/2014/947539.
- [212] Velardo, C. and Dugelay, J. “What can computer vision tell you about your weight?” In: *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. Aug. 2012, pp. 1980–1984. DOI: 10.1109/ESPA.2012.6152447.
- [213] Velardo, C., Dugelay, J.-L., Paleari, M., and Ariano, P. “Building the space scale or how to weigh a person with no gravity”. In: *ESPA 2012, IEEE 1st International Conference on Emerging Signal Processing Applications, January 12-14, 2012, Las Vegas, USA*. Las Vegas, ÉTATS-UNIS, Jan. 2012.
- [214] Vidas, S., Moghadam, P., and Bosse, M. “3D thermal mapping of building interiors using an RGB-D and thermal camera”. In: *2013 IEEE International Conference on Robotics and Automation* (May 2013), pp. 2311–2318. DOI: 10.1109/ICRA.2013.6630890.
- [215] Vidović, M., Sinanović, O., Šabaškić, L., Hatičić, A., and Brkić, E. “Incidence and types of speech disorders in stroke patients”. In: *Acta Clinica Croatica* 50.4 (2011), pp. 491–493.
- [216] Vollmer, M. and Möllmann, K. *Infrared Thermal Imaging: Fundamentals, Research and Applications*. Wiley, 2017. DOI: 10.1002/9783527693306.
- [217] Wang, J., Thornton, J. C., Russell, M., Burastero, S., Heymsfield, S., and Pierson, R. “Asians have lower body mass index (BMI) but higher percent body fat than do whites: comparisons of anthropometric measurements.” In: *The American journal of clinical nutrition* 60.1 (1994), pp. 23–28. DOI: 10.1093/ajcn/60.1.23.
- [218] Wasenmüller, O. and Stricker, D. “Comparison of Kinect v1 and v2 Depth Images in Terms of Accuracy and Precision”. In: *Asian Conference on Computer Vision*. Springer, 2016, pp. 34–45. DOI: 10.1007/978-3-319-54427-4_3.
- [219] Waterlow, J. C. “Nutrition and the Developing Brain”. In: *The Lancet* 301.7800 (1973), pp. 425–426. DOI: 10.1016/S0140-6736(73)90281-X.
- [220] Weinmann, M., Jutzi, B., Mallet, C., and Weinmann, M. “Geometric features and their relevance for 3D point cloud classification”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences (to appear)* (2017).
- [221] Widrow, B. and Hoff, M. E. “Adaptive Switching Circuits”. In: *1960 {IRE} {WESCON} Convention Record, Part 4*. New York: IRE, 1960, pp. 96–104.

- [222] Wikipedia. *Balança* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 2017-09-27]. 2017. URL: <https://pt.wikipedia.org/wiki/Balan%C3%A7a>.
- [223] Wikipedia. *Broselow Pediatric Emergency Tape* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Broselow%20Pediatric%20Emergency%20Tape&oldid=748497708>. [Online; accessed 27-June-2017]. 2017.
- [224] Witten, I. H., Frank, E., and Hall, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.
- [225] Wuerz, R. C., Travers, D., Gilboy, N., Eitel, D. R., Rosenau, A., and Yazhari, R. “Implementation and Refinement of the Emergency Severity Index”. In: *Academic Emergency Medicine* 8.2 (2001), pp. 170–176. DOI: 10.1111/j.1553-2712.2001.tb01283.x.
- [226] Yang, L., Zhang, L., Dong, H., Alelaiwi, A., and Saddik, A. E. “Evaluating and improving the depth accuracy of Kinect for Windows v2”. In: *IEEE Sensors Journal* 15.8 (2015). DOI: 10.1109/JSEN.2015.2416651.
- [227] Yu, L., Wang, S., and Lai, K. K. “An Integrated Data Preparation Scheme for Neural Network Data Analysis”. In: *IEEE Trans. on Knowl. and Data Eng.* 18.2 (Feb. 2006), pp. 217–230. DOI: 10.1109/TKDE.2006.22.
- [228] Zadeh, L. A. “Fuzzy sets”. In: *Information and Control* 8.3 (1965), pp. 338–353. DOI: [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X).
- [229] Zhang, H., Fritts, J. E., and Goldman, S. A. “Image segmentation evaluation: A survey of unsupervised methods”. In: *computer vision and image understanding* 110.2 (2008), pp. 260–280. DOI: 10.1016/j.cviu.2007.08.003.
- [230] Zhang, Z. “A Flexible New Technique for Camera Calibration”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22.11 (Nov. 2000), pp. 1330–1334. DOI: 10.1109/34.888718.
- [231] Zhao, T. and Nevatia, R. “Tracking multiple humans in complex situations”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.9 (Sept. 2004), pp. 1208–1221. DOI: 10.1109/TPAMI.2004.73.
- [232] Zivin, J. A. “Acute stroke therapy with tissue plasminogen activator (tPA) since it was approved by the U.S. Food and Drug Administration (FDA).” eng. In: *Annals of neurology* 66.1 (July 2009), pp. 6–10. DOI: 10.1002/ana.21750.

A

Appendix

A.1 Software structure

The developed software for this thesis is mostly written in the programming language C++. The framework for body weight estimation is divided in several sublibraries:

- `licore` holds the basic data types and interfaces for data types from other software dependencies. The core data type applied in the software is the point cloud: For processing, the `pcl::PointCloud<T>` is used. For the C++ template the `pcl::PointXYZRGBL` is applied, having three values for Cartesian coordinates (x, y, z) , three values for color (r, g, b) and one value for the raw temperature values (t) . Therefore, most algorithms from PCL can be applied with minor adaptations [155].
- `lialgorithm` contains the algorithms for segmentation and feature extraction. Furthermore, the library contains wrapper classes to apply the neural network framework `libFANN` [147].
- `lidatabase` is the interface for the `mySQL` database which is responsible for the data management. All data from the patient and the measurements of the sensors are saved via `lidatabase` and its interfaces.
- `liqt` contains all classes for visualization and human-machine interaction. Moreover, this library contains classes for process control, based on a state machine, as well as multithreading improvements to speed up of offline computation.
- `lisensors` is responsible for the data acquisition from the sensors. Therefore, this library contains wrappers to drivers of the Kinect, the Kinect One and the Optris thermal camera. Additionally, this library contains the basic classes to apply intrinsic and extrinsic calibration.

The proposed framework depends on some other libraries which are presented as follows:

- `PCL`: The point cloud library is the most common state-of-the-art library for all kind of processing with point clouds [180]. The framework is programmed in C++ and provides core functionalities for filtering, segmentation, modeling and registration.
- `Open Computer Vision (OpenCV)`: This library for computer vision is common within the open source community. It provides most state-of-the-art image processing algorithms for basic operations in image processing, e.g., filtering and segmentation.
- `Qt`: The developed GUI is generated by the framework `Qt`. This framework is common for GUIs running on different platforms, e.g., computer, embedded computers or mobile phones.
- `libFann`: `libFANN` is a generic framework for common ANNs. The framework itself is lightweight and delivers core functionalities to apply different learning methods and initializations for an ANN [147].
- `QDjango`: `QDjango` is an easy-to-use interface to access data via `mySQL` queries. The framework is based on `Qt`.

- **mySQL:** MySQL is responsible to store the acquired data. The framework is common for different kinds of databases. The database is stored locally on the computer for processing, but can also be reached via the local network or the Internet.
- **Matlab** is applied for the initial version of the artificial neural network [131]. It is capable to export the Matlab script into C++ code, which can be used in other applications. Furthermore, Matlab is used to evaluate statistics of the accomplished experiments.
- **Doxygen** is an approach for the documentation of code. The Doxygen comments are included in the header of the developed software and provide information about the author, as well as the interfaces and usage of classes and functions. Via a script, Doxygen generates out of these comments a documentation, e.g., in hypertext markup language or \LaTeX [43].

A.2 Finding the Nearest Neighbors

To support the extraction of inliers, the normals of each point are used for RANSAC estimation. The normal vector \mathbf{n} of an arbitrary point in an ordered point cloud is calculated by its neighbored points. For a point cloud from a projective 3D sensor, every point has eight neighbors; except points from the edge of the depth image.

If the point cloud relies on a projective depth sensor, the data is called ordered. Therefore, the nearest neighbors are calculated by their index in the depth image u and v , see Figure A.1a. To prevent affiliation over edges in depth, the Euclidean distance is calculated additionally; points with a high distance are removed as a neighbor. The computation of nearest neighbors in an ordered point cloud can be done efficiently.

In contrast to that, the calculation in an unordered point cloud has to deal with a higher computational load: Here the neighbors cannot be found in linear complexity. Two possibilities exist: Either, a fixed number of nearest neighbors for a given point is searched. This is often referred to the k -nearest neighbor search. Therefore, the Euclidean distances to all points in the cloud are calculated and saved in an ascending order $D = (d_1, d_2, \dots, d_{n-1})$ where $d_1 < d_2 < \dots < d_{n-1}$ for a point cloud with n points. To receive a list of the k -nearest neighbors \mathcal{N}_k , the set of saved distances are cropped to the first k elements and the corresponding

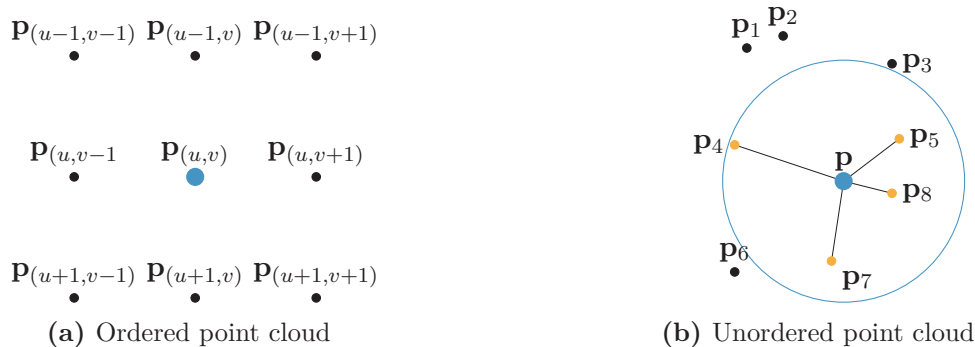


Figure A.1: Nearest neighbors in an ordered point cloud (a) and in an unordered point cloud (b).

points are added to the set of neighbors. The other possibility is to search for neighbors based on a fixed radius. Thus, also the Euclidean distances for all other points have to be calculated. If the distance d_i is below the set radius $d_i < r$, then the point is chosen to be one of the neighbors of a point $\mathbf{p}_i \in \mathcal{N}$. Figure A.1b illustrates the approach to find a set of neighbors with a fixed radius.

A.3 Computation of Normals

Figure A.2 illustrates three different settings to compute the normal \mathbf{n} for an arbitrary point \mathbf{p} with its k -nearest neighbors as a set of points $\mathcal{N}_k = (\mathbf{p}_1 \dots \mathbf{p}_k) \in \mathbb{R}^3$. Three approaches are presented by Klasing et al. [104]: The first normal estimation is based on fitting a plane to the set points including the point \mathbf{p} and its k -nearest neighbors. The arbitrary point \mathbf{p} is located in the plane, while the parameters for the plane are chosen by minimizing the distances of all nearest neighbors for the k -nearest neighbors. Different approaches exist to weight the distance of the nearest points to calculate the plane. One is weight-based on the Euclidean distance between the neighbor and the point. Figure A.2a shows the calculation based on a fitting plane, e.g., via the least square optimization as shown by equation (4.5) on page 69.

Second, the computation of a normal based on nearest neighbored points can be achieved by maximizing angles between the normal and the tangential vectors

$$\mathbf{n}_i = \frac{1}{k} \sum_{i=1}^k w \frac{(\mathbf{p}_i - \mathbf{p}) \times (\mathbf{p}_i - \mathbf{p})}{\|(\mathbf{p}_i - \mathbf{p}) \times (\mathbf{p}_i - \mathbf{p})\|} \quad \text{where } \mathbf{p}_i \in \mathcal{N}_k \quad ,$$

where k is the size of the neighbors and w is a weighting parameter, which can be modified, e.g., depending on the size of the surface triangle, as presented by Jin et al. [96]. For simplicity of the notation, it is assumed that the last-plus-one index maps onto the first neighbor again, e.g., $k + 1 = 1$. Figure A.2b illustrates the approximation of the surface normal based on this method.

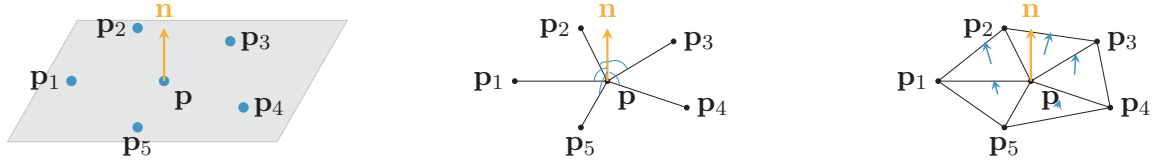
Last, the normal estimation is based on a Delaunay tessellation with non-overlapping triangles [20]. Based on the k -nearest neighbors, a triangle mesh is generated. Each triangle is seen as a surface element, having an own normal vector \mathbf{n}_i . To get the normal for the point \mathbf{p} , the mean value of all adjoining normal vectors is applied with

$$\mathbf{n} = \frac{1}{k} \sum_{i=1}^k \mathbf{n}_i \quad .$$

Figure A.2c shows the estimation of an arbitrary normal based on the triangle mesh.

A.4 Principal Component Analysis

The PCA is a method to look for dependencies between variables, or also to reduce the dimension of a given multi-dimensional space [98], e.g., the face recognition via the eigenfaces proposed by Turk and Pentland [210]. The application for the PCA in the scenario of body weight estimation is the transformation of a set of input variables, so independent and statistic significant differences are highlighted [83, 101].



(a) Calculation based on a fitting plane

(b) Calculation based on angle maximization

(c) Calculation by triangle surface normals

Figure A.2: Different methods to estimate normals for a set of points: Here the k -nearest neighbors towards the point \mathbf{p} are $\mathbf{p}_1 \dots \mathbf{p}_5$, with k having a size of 5. Source: Klasing et al. [104]

The here given explanation sets the focus on PCA for a given set of points $\mathcal{P} \in \mathbb{R}^3$. PCA aims to get the eigenvalues λ for a set of points. Based on these eigenvalues, a prediction for the orientation of the dataset and its dimension can be given. The result of the PCA will result in a covariance matrix Σ , which is equal to an orthogonal coordinate system, while the axes differ in their length.

The calculation of the PCA is structured in four steps. The explanation of the following equations are taken from Artac et al. [11], which is also the basis for the implementation in the PCL.

1. To compute the principal components of a given point cloud $\mathcal{P} \in \mathbb{R}^3$, the centroid has to be known by,

$$\bar{\mathbf{p}} = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_i,$$

for a point cloud containing n points [83, 8].

2. In the next step, the covariance matrix $\Sigma_{3 \times 3} \in \mathbb{R}^3$ of the point cloud is calculated with the help of the centroid $\bar{\mathbf{p}}$ by

$$\Sigma = \frac{1}{n} \sum_{i=1}^n (\mathbf{p}_i - \bar{\mathbf{p}})(\mathbf{p}_i - \bar{\mathbf{p}})^T .$$

The covariance matrix is symmetric as well as diagonalizable. Moreover, the covariance matrix with its three dimensions provides the variance between the Cartesian coordinates along the principal diagonal by

$$\Sigma = \begin{pmatrix} \text{var}(x, x) & \text{cov}(x, y) & \text{cov}(x, z) \\ \text{cov}(y, x) & \text{var}(y, y) & \text{cov}(y, z) \\ \text{cov}(z, x) & \text{cov}(z, y) & \text{var}(z, z) \end{pmatrix} .$$

Because of the symmetry of the matrix, $\text{cov}(y, x) = \text{cov}(x, y)$. The covariance matrix can be fractionized into the three eigenvectors \mathbf{v}_1 , \mathbf{v}_2 , and \mathbf{v}_3 – one for each eigenvalue. The corresponding eigenvectors are calculated by

$$\Sigma \cdot \mathbf{v}_i = \lambda_i \cdot \mathbf{v}_i \quad \text{where } i = 1, 2, 3 .$$

The eigenvalues are arranged by the size of their value with $\lambda_1 \geq \lambda_2 \geq \lambda_3 \in \mathbb{R}^3$.

3. The eigenvalues λ of the covariance matrix Σ can be calculated based on the characteristic polynomial by

$$0 = \det(\Sigma - \lambda_i \mathbf{1}) = \begin{vmatrix} \text{var}(x, x) - \lambda_i & \text{cov}(x, y) & \text{cov}(x, z) \\ \text{cov}(y, x) & \text{var}(y, y) - \lambda_i & \text{cov}(y, z) \\ \text{cov}(z, x) & \text{cov}(z, y) & \text{var}(z, z) - \lambda_i \end{vmatrix},$$

showing the covariances of each coordinate, as well as the variances of x , y , and z on the principal diagonal.

4. Finally, the corresponding eigenvectors can be calculated. The i -th eigenvector \mathbf{v}_i processed by solving

$$\det(\Sigma - \lambda_i \mathbf{1}) \cdot \mathbf{v}_i = 0 \quad .$$

For a covariance matrix $\Sigma \in \mathbb{R}^3$, three eigenvalues and three corresponding eigenvectors exist. The eigenvector, which belongs to the highest eigenvalue, is the principal component of the point cloud. It reflects the strongest correlation between the different dimensions of the point cloud.