

Technische Universität München

Fakultät für Informatik

Bildverstehen und Intelligente Autonome Systeme

Perspective-Adjusting Appearance
Model for Distributed Multi-View
Person Tracking

Martin Hermann Albert Eggers

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Florian Matthes
Prüfer der Dissertation: 1. Univ.-Prof. Dr. Bernd Radig (i.R.)
2. Univ.-Prof. Dr. Alois Knoll

Die Dissertation wurde am 11. 06. 2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 22. 10. 2014 angenommen.

Abstract

As a result of the growing miniaturization of electronic devices, scenarios where large numbers of interconnected cameras can be feasibly deployed in all manners of indoor environments are rapidly becoming more realistic. For sophisticated camera systems capable of observing and tracking humans, applications are developing in domains like surveillance of private and public spaces, ambient assisted living and human-robot interaction, which are currently advancing at an increasing pace.

This dissertation tackles the challenges of detecting, tracking and re-identifying pedestrians across multiple camera views in an indoor environment. From an algorithmic point of view, two characteristic challenges of multi-view tracking problems are addressed: the changes in perspective and the resulting variation in appearance of the target towards the sensor, as well as the view transition challenge, which lies in reliably maintaining the identity of moving targets across multiple cameras. From a technical point of view, further challenges are addressed in the correct placement of sensors, the calibration of these sensors, and the setup of the system to accommodate operation on the sensor data in real time.

To cope with the algorithmic challenges, a novel method to for appearance-based modeling of non-transparent objects, based on an adaptive mixture of color distributions, is introduced in this thesis and realized for pedestrian targets. This adaptive appearance model employs a two-stage, simplified, three-dimensional, geometric shape model, which is used to derive weights for the color distributions of object parts based on the observation perspective of the sensor towards the target, and allows for a refinement of the appearance model during the tracking process.

To verify the proposed system setup, a large-scale experiment was conducted on data recorded from a setup of 40 cameras observing a 10×10 m area from a top-down perspective, connected to a cluster of 40 computers for distributed image processing. Cameras were fully calibrated using a novel method, achieving an average reprojection error of 0.13 pixels for the complete system, which exceeds state-of-the art accuracy. Long-term testing has the system running with at least 99.994% availability for up to two weeks.

Experimental participants performed a total of 80 short walking sequences, during which they were tracked across the fields of view of a subset of the aforementioned camera system. The performance of the tracking was evaluated regarding the the accuracy of the predicted position and the success rate of the transfer of targets between different camera fields of view. Comparison of the evaluation results for the proposed adaptive appearance model and a state-of-the-art static color distribution model yielded an improvement of up to 12 percent in the error of the prediction precision, and 38 percent in the error of target view transfer.

Keywords: camera system, appearance modeling, multi-view, distributed image processing, pedestrian tracking

Zusammenfassung

Infolge der zunehmenden Miniaturisierung elektronischer Geräte werden Szenarien in denen miteinander verbundene Kamerasysteme effizient in allen Arten von Innenräumen eingesetzt werden können immer mehr zur Realität. Für komplexe Kamerasysteme, die dazu geeignet sind Menschen zu beobachten und ihre Position in Bildfolgen nachzuvollziehen, eröffnen sich Anwendungsmöglichkeiten in Bereichen wie der Überwachung privater und öffentlicher Räume, dem umgebungsunterstützten Leben (AAL) und der Interaktion von Mensch und Roboter (HRI), welche sich gegenwärtig mit großen Schritten voranbewegen.

Diese Dissertation nimmt sich der Herausforderungen des Erfassens, Verfolgens und Wiedererkennens von Fußgängern über multiple Kamerablickfelder in Innenräumen an. Von algorithmischer Seite wird auf zwei charakteristische Problemstellungen der Blickfeldübergreifenden Objektverfolgung Bezug genommen: Die Veränderungen in der Beobachtungsperspektive und die daraus resultierende Veränderung im äußeren Erscheinen der Zielperson gegenüber dem Sensor, sowie das Problem des Blickfeldwechsels, welches darin besteht die Identität sich bewegender Zielpersonen über mehrere Blickwinkel hinweg verlässlich zu bewahren. Aus technischer Sicht wird auf weitere Herausforderungen Bezug genommen, die aus der korrekten Platzierung der Sensoren, der Kalibrierung derselben, und dem Systemaufbau zur Ermöglichung einer Verarbeitung der Sensordaten unter Realzeitbedingungen bestehen.

Um die algorithmischen Herausforderungen zu meistern, wird in dieser Arbeit eine neuartige Methode zur Modellierung des äußeren Erscheinens nicht-transparenter Objekte vorgestellt, die auf einer adaptiven Mischung von Farbverteilungen beruht, und diese Methode zur Modellierung von Fußgängern als Zielobjekten umgesetzt. Dieses adaptive Erscheinungsmodell bedient sich eines zweistufigen vereinfachten dreidimensionalen geometrischen Umrissmodells, welches dazu verwendet wird die Gewichtungen der einzelnen Farbverteilungen basierend auf der Beobachtungsperspektive des Sensors gegenüber dem Zielobjekt herzuleiten, und erlaubt darüberhinaus die Verfeinerung des Erscheinungsmodells während des Vorgangs der Zielverfolgung.

Um den vorgeschlagenen Systemaufbau zu verifizieren wurde ein umfangreiches Experiment auf Daten durchgeführt, die mit einem System aus 40 Kameras aufgenommen wurden. Dieses System beobachtet eine 10×10 m große Fläche aus von der Decke abwärts gerichteter Kameraperspektive beobachtet, wobei die Sensoren an einen Verbund aus 40 Rechnern angeschlossen sind, welche zur verteilten Verarbeitung der Bildfolgen eingesetzt werden. Die Kameras wurden mittels eines neuartigen Verfahrens intrinsisch und extrinsisch kalibriert, wobei ein durchschnittlicher Rückprojektionsfehler von 0.13 Pixeln für das Gesamtsystem erreicht wurde, was den gegenwärtigen Stand der Technik bezüglich der Genauigkeit übertrifft. Langfristige Stabilitätstests erfassen die Systemverfügbarkeit mit 99.994 Prozent über einen Zeitraum von zwei Wochen.

Insgesamt 80 kurze Gehsequenzen wurden durch die Experimentsteilnehmer durchgeführt, während der diese durch die Blickfelder einer Teilmenge der oben erwähnten Sensoren verfolgt wurden. Das Ergebnis der Zielverfolgung wurde hinsichtlich der Genauigkeit der vorhergesagten Position und der Erfolgsrate des Blickfeldübergangs der Zielpersonen ausgewertet. Ein Vergleich der Auswertungsergebnisse des vorgeschlagenen adaptiven Modellierungsansatzes mit einem statischen farbbasierten Modellierungsansatz nach Stand der Technik erbrachte eine Verbesserung von bis zu 12 Prozent in der Schätzung der Position der Zielperson und von 38 Prozent in der Fehlerquote des Blickfeldübergangs.

Schlagwörter: Kamerasysteme, Erscheinungsmodellierung, Mehrere Blickwinkel, Verteilte Bildverarbeitung, Verfolgung von Fußgängern

Acknowledgments

First of all, I would like to thank my supervisor and academic mentor Prof. Dr. Bernd Radig for his staunch support and acute insight during the various stages of the work leading up to my dissertation.

My good friend and colleague Dr. Christoph Mayer, whose inspiring example influenced me greatly in my decision to pursue the research presented here, and who has been a paragon to me during my time at TUM.

My colleagues Sikandar Amin, Dr. Claus Lenz and Thorsten Röder, all of whom worked in partner projects and contributed to the work leading up to this thesis at one point or another. Dr. Giorgio Panin, the initiator and project leader of the Open Tracking Library (OPENTL), which was employed in the implementation of this project, as well as Prof. Dr. Carsten Steger and Veselin Dikov of MVTEC SOFTWARE GMBH, creators of the HALCON machine vision system, who collaborated with me on the publication of the calibration setup and results.

My students Thomas Kisler, Muhanad Zaki, Michael Neumann and Nathan Obermaier, each of whom contributed to different parts of the system during the course of their theses or research assistant contracts.

My predecessor at the Institute for Image Understanding and Knowledge-based Systems (TUM-IUKS), Dr. Matthias Wimmer, who initiated the project and wrote the proposals for the research grants.

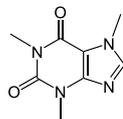
My colleagues at the CoTESys Central Robotics Laboratory (CCRL), where Martin Lawitzky, Alexander Mörtl, Thomas Nierhoff, Markus Huber and Sebastian Erhart are only a few to be mentioned, whom I collaborated with on applications of the vision system during various stages of my work at the laboratory.

Our network administrator at the institute, Quirin Lohr, whose expertise in running large Linux computer networks was indispensable during the setup of the distributed image processing system.

Everyone who assisted with the editorial work during the final days of composing this thesis, such as with proof-reading or layout decisions, namely Christoph Mayer, Michael Herrmann and Sikandar Amin.

Finally, I would like to thank my parents, Dr. Harald Eggers and Gabriele Eggers, my siblings Christina Eggers, Florian Eggers and Barbara Eggers, and my girlfriend Lena Blumentritt for their continued moral support, especially during the final months of composing this thesis.

Dedicated to my family, and my favourite molecule,



I could not have written this dissertation without you.

Contents

List of Acronyms	xxiii
1 Introduction	1
1.1 Scientific Contributions	8
1.2 Outline of the Thesis	9
2 Real-Time Multi-View Pedestrian Tracking	11
2.1 Problem Statement	11
2.2 Outline of this Chapter	12
2.3 Related Work	13
2.3.1 Definitions and Terminology	14
2.3.2 Methods of Pedestrian Tracking	15
2.3.3 Visual Tracking	17
2.3.4 Tracking and Data Association Algorithms	18
2.3.5 Tracking with Multiple Cameras and View Transition	19
2.3.6 Real-Time Multi-View Tracking Systems	20
2.3.7 Applications of Person Tracking	24
2.4 Solution Idea	26
2.5 Conditions at the Laboratory	29
2.6 Hardware Layer Installation	30
2.6.1 Cameras	32
2.6.1.1 Camera Types and Specifications	33
2.6.1.2 Lenses	33

2.6.1.3	Placement and Number of Cameras	34
2.6.1.4	Sampling Density and Redundancy	36
2.6.2	Network Architecture	38
2.6.3	Computers Used For Image Processing	41
2.7	Camera Calibration	42
2.7.1	Camera Calibration Accuracy	45
2.7.2	Calculation of the Floor Plane	46
2.8	Preprocessing Layer Implementation	47
2.8.1	Image Preprocessing	48
2.8.2	Synchronization	49
2.8.3	Monitoring and Stability	49
2.9	Application Layer Implementation	50
2.9.1	Distribution and Communication	50
2.9.2	Person Detection	51
2.9.2.1	Foreground-Background Segmentation	51
2.9.2.2	Blob Clustering and Association	53
2.9.2.3	Model Initialization for Tracking	53
2.9.2.4	Performance of the Detection Process	54
2.9.3	Person Tracking	55
2.9.3.1	Multi-target MCMC filter	56
2.9.3.2	View Transition	57
2.10	Experimental Evaluation	62
2.10.1	Recording the Data set	63
2.10.2	Single-View Tracking Accuracy	63
2.10.2.1	Error metric	67
2.10.2.2	Results	68
2.10.2.3	Adjusted Error Metric and Results	69
2.10.3	Multi-View Tracking Performance	70
2.10.3.1	Performance metric	71
2.10.3.2	Results	72

2.10.4	Target Identity Maintenance and Recovery	72
2.10.4.1	Performance metric	74
2.10.4.2	Results	74
2.10.5	System Uptime and Robustness	75
2.11	Summary and Discussion	77
3	Appearance Modeling	79
3.1	Problem Statement	79
3.2	Outline of this Chapter	81
3.3	Related Work	81
3.3.1	Statistical Color Descriptors	82
3.3.2	Appearance Modeling	83
3.4	Solution Idea	85
3.4.1	Abstract Solution Idea	86
3.4.2	Application to Pedestrian Tracking	88
3.4.3	Assumptions and Constraints	88
3.5	Appearance Model	90
3.5.1	Combined Color Distribution	91
3.5.2	Color Histograms	92
3.5.2.1	Properties of Color Histograms	93
3.5.3	Shape Model	94
3.5.4	Shape Model Proportions	95
3.5.5	Polygon Mesh Shape Model	99
3.6	Model Usage During Tracker Operation	102
3.6.1	Model Reprojection and Weight Computation	102
3.6.2	Determining the Appearance Models for Body Parts	104
3.6.3	Generating the Color Distribution for Tracking Hy- potheses	107
3.6.4	Transition of a Target Between Views	107
3.7	Experimental Evaluation	108
3.7.1	Single-View Tracking Accuracy	109

3.7.2	Multi-View Tracking Performance	113
3.7.2.1	Results	113
3.7.3	Target Identity Maintenance and Recovery	114
3.7.3.1	Results	115
3.8	Summary and Discussion	116
4	Applications	123
4.1	Outline of this Chapter	124
4.2	Handshake Recognition	125
4.2.1	Related Work	125
4.2.2	Method	126
4.2.3	Integration	127
4.3	Pointing Gesture Recognition	127
4.3.1	Related Work	127
4.3.2	Method	129
4.3.3	Integration	132
4.4	Discussion	132
5	Summary and Outlook	135
5.1	Discussion	135
5.1.1	Camera System Concept and Architecture	136
5.1.2	Automated Multi-Camera Calibration	137
5.1.3	Shape-Aware Adaptive Appearance Modeling	138
5.2	Future Work and Outlook	138
A	Publications	145
B	Additional Tables	147
C	Glossaries of Terms	155
	List of Operators	155
	List of Symbols	157

CONTENTS

xv

List of further Terms	165
Bibliography	169

List of Figures

2.1	Pipeline for visual object tracking	18
2.2	Funnel model for information condensation	19
2.3	Architecture of the 3D SURVEILLANCE vision system	22
2.4	Architecture of the KNIGHT vision system	22
2.5	Architecture for the vision system proposed by Zhao <i>et al.</i>	23
2.6	Layered modular architecture of the proposed camera system	27
2.7	Experimental area at the CCRL	30
2.8	Map of the experimental area	31
2.9	Two examples for human-robot interaction	32
2.10	2D scheme of camera FOV calculation	35
2.11	Plan of camera positions within the target area	37
2.12	Overview of the hardware setup	39
2.13	Multiple camera setup projection model	43
2.14	Calibration object with 49 circular marks	44
2.15	Fiduciary markers for determining the floor plane	47
2.16	Architecture of the preprocessing layer	48
2.17	NTP synchronization offsets for the processing nodes	49
2.18	Architecture of the application layer	50
2.19	Different steps of pedestrian detection	52
2.20	Sample images from pedestrian detection	55
2.21	Cylindrical pedestrian shape model for tracking	56
2.22	Hypotheses update for the MCMC tracker	58

2.23	Transfer tree for view transition	60
2.24	Transition areas for view transition	60
2.25	Full camera grid for human-readable output	62
2.26	Example images from the evaluation data set	65
2.27	Experimental area used to record the evaluation data set . . .	65
2.28	Camera positions within experimental area	66
2.29	Schematic trajectories for single-view tracking evaluation . . .	66
2.30	Error metrics for single-view tracking	68
2.31	Scheme for identity recovery experiment	73
3.1	Pedestrian from different perspectives within FOV.	80
3.2	Sample objects with vertical appearance pattern	86
3.3	Schematic of the adaptive appearance modeling approach . . .	89
3.4	Schematic of the generalized cylinder model	96
3.5	Bust of Marcus Vitruvius Pollio; Vitruvian Man	97
3.6	Triangle strip and fan techniques	100
3.7	Example polygon mesh shape model	101
3.8	Example composite histogram of a clothed pedestrian	105
3.9	Two-dimensional scheme of model during view transition . . .	108
3.10	Extrapolation of tracking success for varying transition counts	115
3.11	Comparison of performance for static and adaptive appearance	117
3.12	Shape model proposed by Isard and MacCormick	118
3.13	Configuration for surveillance of underground platform	120
3.14	Examples of potential tracking targets	121
4.1	Handshake as seen from supracranial perspective	126
4.2	Anatomical model for pointing gesture extraction	129
4.3	Feature extraction for model fitting	131
5.1	Concept for further development on the system architecture .	139
5.2	Ideas for further automation of the calibration routine	140

5.3 Annotations for full body pose estimation 142

List of Tables

2.1	Specifications of the cameras used in the CCRL setup	33
2.2	Specifications of the lenses used in the CCRL setup	34
2.3	Data rates for streaming on the camera network	41
2.4	Specifications of the processing nodes, first phase	41
2.5	Specifications of the processing nodes, second phase	42
2.6	Error comparison against other research reports	46
2.7	Results for the evaluation of the pedestrian detection	54
2.8	Statistics for sequences in the evaluation data set	64
2.9	Statistics for participants in the evaluation data set	64
2.10	Accuracy evaluation for single-view tracking	69
2.11	Accuracy evaluation for single-view tracking	70
2.12	Performance evaluation for multi-view tracking	72
2.13	Results for target identity recovery	75
2.14	Availability and downtimes for robustness test	76
3.1	Specifications for the generalized cylinder model	98
3.2	Accuracy evaluation for single-view tracking	110
3.3	Accuracy evaluation for single-view tracking	110
3.4	Performance evaluation for multi-view tracking	113
3.5	Extrapolation of tracking success for varying transition counts	114
3.6	Results for target identity recovery	116
4.1	Anatomical landmarks tracked for gesture recognition	130

B.1	Mathematical style and typesetting	147
B.2	Text style and typesetting	147
B.3	Poses of the camera CCD sensors, pt.I	148
B.4	Poses of the camera CCD sensors, pt.II	149
B.5	List of image sequences used during evaluation, pt.I	150
B.6	List of image sequences used during evaluation, pt.II	151
B.7	List of image sequences used during evaluation, pt.III	152
B.8	Terminology for camera perspectives w.r.t. pedestrians	153
B.9	Software packages and libraries used in the implementation	154

List of Acronyms

2D	two-dimensional
3D	three-dimensional
AAL	ambient assisted living
AFR	annualized failure rate
ANN	artificial neural network
BTF	brightness transfer function
CCD	charge-coupled device
CCRL	CoTESYS Central Robotics Laboratory
CCTV	closed-circuit television
CORBA	Common Object Request Broker Architecture
CoTESYS	Cognition for Technical Systems
CPU	central processing unit
DOF	degree of freedom
EM	expectation maximization
FOV	field of view
FPS	frames per second
GEV	GigE-Vision
GigE	Gigabit Ethernet

GMM	Gaussian mixture model
GPU	graphics processing unit
GVSP	GigE-Vision Streaming Protocol
HCI	human-computer interface
HDD	hard disk drive
HMM	hidden Markov model
HOG	histogram of oriented gradients
HRI	human-robot interaction
HSI	hue/saturation/intensity
ICE	Internet Communications Engine
IP	Internet Protocol
IPC	inter-process communication
IPPF	independent partition particle filter
JPDA	joint probabilistic data association
KoGMo-RTDB	Real-Time Database for Cognitive Automobiles
LM	Levenberg-Marquardt
LOS	line of sight
LRV	light reflectance value
MAP	maximum a-posteriori
MCMC	Markov chain Monte Carlo
MLP	multilayer perceptron
MTU	maximum transmission unit
NIC	network interface controller
NTP	network time protocol
OTS	off-the-shelf

PCA	principal components analysis
PTZ	pan-tilt-zoom
RFID	radio frequency identification
RGB	red/green/blue
RGB-D	red/green/blue-depth
RMS	root mean square
ROI	region of interest
RVM	relevance vector machine
SIFT	scale invariant feature transform
SIR	sequential importance resampling
SMC	sequential Monte Carlo
SSPF	sequential sampling particle filter
SVM	support vector machine
TOF	time of flight
UAV	unmanned aerial vehicle
UDP	User Datagram Protocol
YCbCr	luminance/chrominance
YUV	luminance/chrominance

Chapter 1

Introduction

At the dawn of the third millennium, the pervasion of our living and working environments with electronic devices is progressing at an ever-increasing pace. Among the observations relating to this development, the prediction dubbed Moore's Law [196, 240] is certainly one of the most prominent. It states that the number of components in integrated circuits doubles every year, and has become synonymous with the exponential development in microelectronics over the past five decades,

The increase in transistor density fuels two different but related trends, with an increase of processing power for equal-size devices on the one hand, and an increase in miniaturization of equal-power devices on the other. As a side-effect, the price of equal-power, equal-size devices is continually dropping, making those devices accessible to a higher number of individuals. Consequently, Moore's Law has spawned several scions regarding related developments with exponential growth, such as Kryder's Law [165, 289] (hard disk storage capacity), Butters' Law [233] (network transmission capacity) and Hendy's Law [250] (camera and screen resolution).

Among the devices benefiting from the trend in miniaturization are CCD cameras (*cf.* Tompsett *et al.* [275]). As those sensors are becoming ever smaller and more affordable, the opportunity arises to furnish environments like factory halls, offices, public spaces and even private homes with larger numbers of optical sensors, an endeavor that is becoming increasingly affordable for small companies and private citizens alike. Combined with the increasing computing power available to process all the generated data, a new generation of surveillance systems, dubbed Smart Surveillance (*cf.* Hampapur *et al.* [110]) has been on the rise for the last decade.

An argument can be made for the installation of cameras over different kinds

of sensors benefiting from the same developments, such as microphones or radio frequency identification (RFID) scanners, on grounds of their versatility. Since preexisting public and private environments were fashioned with human sensory capacities in mind, the use of visual sensors comes as a natural choice given the reliance of humans on their visual perception. Furthermore, visual surveillance systems, such as closed-circuit television (CCTV) systems, have already been in place in many public environments for years, to serve as a basis for real-time manual surveillance and recordings for *post facto* analysis. In many cases these systems can be retrofitted or upgraded to accommodate Smart Surveillance approaches.

Benefits to be gained from such surveillance systems are manifold, and largely depend on the application domain, three of which are illustrated exemplatively in the following.

For the surveillance of public spaces and critical installations, applications of particular interest in the security domain include the automated detection of persons entering or leaving the target area (*cf.* Freer *et al.* [88], Snidaro *et al.* [253]), the identification of known persons using previously acquired biometric information (such as face recognition or gait recognition, *cf.* Riaz *et al.* [230] and Lee *et al.* [173]) and the detection of the presence of potentially dangerous or absence of valuable objects (*cf.* Chuang *et al.* [47]).

Apart from the security aspect, research has also focused on the creation of so-called assistive environments or smart rooms (*cf.* Pentland [216]), where data gathered from sensors placed in the environment is used to decide when automated tasks are to be performed by connected actuators. Examples include automated control of lighting, heating or ventilation depending on the presence or activities of persons detected within a smart environment (*cf.* Focken *et al.* [87]).

From a roboticist's point of view, the ability of robots to perform joint tasks with humans has been a research focus for many years (*cf.* Kosuge *et al.* [162] for a survey on the topic or Lenz *et al.* [174] for a more recent example). For sophisticated cooperation between humans and mobile robots, the capacities of the robots to understand their environment from data acquired by optical sensors – analogous to what is termed *visual perception* (*cf.* Gibson [95, 96], Cornsweet [57]) in humans – are often a crucial element, as Steinfeld *et al.* [256] mention. Unlike humans, however, robots are not necessarily limited to data from sensors mounted on their chassis. In locations where they can be installed, access to data from external sensor arrays can greatly extend the area perception of the robot beyond the limitations of its own platform, and facilitate tasks like navigation towards targets not within the robots

original line of sight. Thus, argumentatively speaking, the loop to the two aforementioned application domains is closed.

A common denominator of those application domains is the existence of a certain area that is under visual surveillance, which is termed *target area* or *area of observation* in the following, where the former denotes the intended area to be observed, while the latter denotes the effectively observed area. Naturally, the exact topology of this area varies, and consequently so does the ideal sensor configuration (*cf.* Hörster and Lienhart [119], Bodor *et al.* [24]).

More prominently, the above-mentioned application domains furthermore share the challenge of detecting and tracking persons within said target area, which are also termed *pedestrians* in the following in reference to their natural method of locomotion. This challenge has remained a staple research subject for many years, and the terms “person tracking” (*cf.* [164]) and “pedestrian tracking” (*cf.* [65, 261]) have been coined to describe it.

Among others, application-centered research on multi-view visual pedestrian tracking has focused on employing techniques in surveillance tasks [19, 134] and safety applications [268]. Challenges specific to multi-view approaches can be divided into two categories: technical challenges, pertaining to the realization of the system and largely related to the scale of the system, and algorithmic challenges, which stem from the requirement to coordinate the use of multiple sensors simultaneously, establish consistency and avoid ambiguity, especially with regard to varying perspectives.

The first challenge to mention regarding the technical side is posed by the amount of data generated from such systems. A single state-of-the-art industrial camera usually generates between 25 and 30 images per second, a number that is geared towards human sensory capacities and preferences (*cf.* Apteker *et al.* [6]). From a data perspective, these digital images number several megabytes each for high-resolution cameras. For the complete coverage (*i.e.* without blind spots or static occlusion) of a small apartment, however, the number of required sensors easily exceeds single digits, not to mention large and complex target areas such as factory halls, train stations, or airports. These considerations affect the network topology and bandwidth requirements, as well as the amount of processors that have to be employed.

Another challenge, which affects both the technical and the algorithmic part of the task, is posed by the requirement to perform the tracking in real time, in order to allow for simultaneous use of the extracted tracks in other technical systems connected to the tracking system, such as robots (in human-robot interaction (HRI)) or alarm systems (in surveillance contexts), and in order to avoid latency-related coordination problems within the system

itself. This can be expected to cause difficulties for approaches using high-resolution models of the human body, approximating the position of single limbs. For example, Caillette *et al.* [38] report a reconstruction time of approximately 70 ms per frame for their visual full-body tracking approach running on off-the-shelf hardware, which clearly exceeds the desired response times for cameras operating at up to 30 fps. These considerations suggest algorithmic approaches treating pedestrians as monolithic entities, using only those features to describe their appearance which can be effectively extracted and processed within the time frame allowed for by the hardware.

Furthermore, camera calibration is another challenge which touches upon both the technical and the algorithmic. It has to be addressed in order for the camera system to provide information about objects in real-world geometry, which are most convenient for exchange among different technical systems (*e.g.* actuators or other sensors) as well as for interpretation by humans. Although, in general, this topic has been explored for many decades for single cameras, the exact calibration of multiple cameras against a common world coordinate system remains a challenging task, the complexity of which increases non-linearly with the number of cameras involved, which is compounded by the fact that in most common cases, only a small percentage of the cameras fields of view (FOVs) intersect with each other.

The second challenge related to the intersection of camera FOVs is the *view transition* problem. When a target leaves the FOV of one camera and enters that of another, the tracking has to continue within the new FOV, without interruptions, or worse, target loss. Furthermore, the identity of the target has to be verified, in order to avoid the confusion of targets in the moment of transition. For targets which appear in multiple FOVs at the same time, the camera with the optimal observation perspective, best suited to track the target, has to be determined. Furthermore, a decision has to be made, whether a target is tracked in the maximum number of possible FOVs and the results merged using a data fusion approach, or if it is more efficient to track a target only in a single FOV at a time.

Finally, when tracking pedestrians, the fact that their appearance varies with the perspective the camera has on the target has to be taken into account. To disambiguate, this perspective is termed the *observation perspective* in the following. A robust approach has to be able to compensate for shifts in observation perspective (and consequently, appearance) caused by target movement within the limits of a single FOV, as well as transition of a target between two neighboring or intersecting FOVs. This is especially important when tracking multiple targets, since this introduces the possibility of con-

fusing targets, consequently increasing the need for meaningful appearance descriptors to avoid those confusions.

This thesis presents approaches to the mentioned challenges, realized in terms of a vertically integrated pedestrian tracking system. An area of 10 by 10 m is selected for observation. From the technical side, consistent area coverage is achieved by mounting 40 cameras in a grid, facing top down at the observation area with interlocking fields of view. Flexibility regarding networking is achieved by selecting GigE-Vision (GEV) camera technology, which operates via Ethernet and allows for connections of up to 100 m. To provide sufficient data processing power, these 40 cameras are connected to the same number of computers, located in a nearby server room. Off-the-shelf hardware is employed to allow for flexible replacement and addition of components. Collectively, these decisions address the challenge posed by the high amount of generated image data, and provide a solid technical basis to perform the algorithmic steps required for pedestrian detection and tracking in real time.

From a software architecture perspective, the technical groundwork is extended by building a two-layered distributed application system on top of it. The *service layer* addresses the communication and synchronization between the connected hardware components. Furthermore, functionality such as storage and replaying of multi-camera video sequences and real-time pre-processing of images, *e.g.* removing lens distortion, is realized through this layer. For inter-process communication (IPC) within the service layer, and for cross-layer communication, the Real-Time Database for Cognitive Automobiles (KOGMO-RTDB) (*cf.* Goebel and Färber [98, 99]) is employed, which provides some data time stamping and synchronization functionality. The service layer is realized as independent processes on each of the processing clients. Communication between processes within the layer is only employed for two purposes: Firstly, for synchronization of the processing client system clocks, which is performed in regular intervals, and secondly, for the purpose of camera calibration, which is a maintenance task requiring operator initiative and conducted irregularly.

Situated on top of the service layer, the *application layer* provides a modular framework for applications, foremost of which to mention is pedestrian tracking. In general, the large amounts of raw image data suggest a distributed approach to image processing, in order to reduce the amount of network traffic generated by eliminating the need to transfer all raw image data to a single location. and instead transmitting extracted high-level results. Therefore, not unlike the service layer, the application layer employs a distributed image processing approach, with *client modules* for the differ-

ent applications processing data from a single camera each. For intra-layer IPC, the framework relies on the Internet Communications Engine (ICE) (*cf.* Henning and Spruiell [116]). Two central administrative modules coordinate the client modules: Firstly, the *registration module* provides setup information and coordinates the IPC. Secondly, the *server module* integrates the high-level results (*i.e.* world poses) obtained from different client modules, manages global target identities, and handles outgoing communication to connected technical systems, *i.e.* result broadcast.

For the pedestrian tracking module itself, a two-step tracking approach is employed. The pedestrian tracking step is preceded by a *pedestrian detection* step. For the purpose of detection, an adaptive background subtraction approach using Gaussian mixture models (GMMs) (*cf.* Power and Schoones [223]) is integrated, which provides an estimate of initial target positions, as well as color descriptors for an initial target-specific appearance model. For the subsequent *tracking step*, the Bayesian tracking approach suggested by Panin (*cf.* [211]) is integrated. Accordingly, the tracking is performed using a multi-target Markov chain Monte Carlo (MCMC) particle filter (*cf.* Panin *et al.* [212]) to generate the tracking hypotheses, in combination with the color-based appearance descriptors extracted in the detection step for hypothesis verification. To address the challenge of tracking targets across overlapping FOV boundaries, static *transition areas* are defined within the overlapping FOV parts, which trigger a view transition for a target as soon as the target enters these areas, switching the tracking of the target from one camera (and connected tracking module) to the next, by transferring the current pose and appearance descriptors to the appropriate tracking module, and initializing another MCMC tracking sequence with the transmitted data. The generation of the transition areas is performed through the use of a target *transfer tree*, which partitions the observation area into responsibility zones, and handles camera neighborhood relations.

Regarding the specifics of the appearance modeling, two different approaches are employed. Firstly, a state-of-the-art *static appearance* approach is integrated, where the initial color descriptors acquired at detection are used throughout the entire lifetime of a target across several FOVs. In addition, a novel *adaptive appearance* approach is presented, where color descriptors are varied according to the observation perspective, thereby improving the prediction of the color descriptors for tracking hypotheses. To that end, a two-step modeling technique is employed to generate a static anthropometric shape model consisting of regular polygons, which is exploited, in combination with the observation perspective, over the course of multiple observations, to refine the color predictions, and consequently improve the accuracy

of the tracking under varying observation perspectives. Special attention is given to the moment of FOV transition, where the observation perspective - and consequently, the appearance model - displays the largest gradient.

To put the role of this dissertation in a wider perspective, the document at hand provides two distinctive features which define its scope against the related work in this field. Firstly, the possibilities of distributed tracking and surveillance systems with large numbers of cameras are explored in a pilot experiment. As a proof of concept, an example system is assembled via integration of state-of-the-art hardware components, software components, and Computer Vision methods. As a consequence, critical elements of the system and challenges caused by the scale of the system, in contrast to systems with comparable functionality but less area coverage and fewer sensors, are identified and solutions to those challenges are presented.

Secondly, a novel adaptive appearance modeling approach is presented, which is designed with the challenges of the previously mentioned system in mind. This approach improves on state-of-the-art methods in color-based tracking for objects of known object classes (*i.e.* with common geometric features and, to a degree, color schemes) for camera systems with known camera parameters and environmental geometry, *i.e.* topography of the plane that the movement of those objects is restricted to. As a consequence, the area of application of the presented approach is mostly for tracking within man-made environments, such as within buildings. The improvement of the state-of-the-art is achieved by the incorporation of the observation perspective into the modeling of the appearance of the target using a non-deformable geometric shape model, which results in a perspective-independent model that can in turn be used to calculate appearance descriptors for a target under varying observation perspectives which provide more accurate results than the standard static appearance approach.

To provide a summary of these remarks, the following section reiterates the scientific contributions of this thesis in a concise manner. A similarly concise formulation of problem statements and solution ideas can be found in the respective chapters (*cf.* Section 2.1 on page 11 and Section 3.1 on page 79).

1.1 Scientific Contributions

In detail, the contributions of this thesis are as follows:

- (1) **A vertically integrated multi-camera system for pedestrian tracking in an indoor area is presented.** The presented system distinguishes itself through its unique scale for camera systems of its type regarding the coverage area and number of integrated components. It is organized in a three-layered architecture, consisting of a hardware layer with two software layers, one of which comprises service, system maintenance and preprocessing tasks, while the second one provides high-level image processing capabilities. Solutions are presented for multiple challenges occurring in various stages of the system integration, *e.g.* sensor placement, synchronization of image processing, and the transition of targets between multiple FOVs. The system is demonstrated to be capable of tracking the movements of multiple pedestrians across the target area in real time. Finally, the performance of the system regarding pedestrian detection, pedestrian tracking and long-term system stability is evaluated on the recorded data.

- (2) **A semi-automated camera calibration method suitable for the calibration of a multi-camera system with overlapping FOVs is presented.** The presented approach is divided into two steps, and employs a well-defined calibration object with circular marks. In the first step, the calibration object is exposed to cameras in overlapping and non-overlapping parts of the FOVs, while varying distance and rotation around all three axes. Ideally, these degrees of freedom are exploited to the maximum possible extent. Synchronized images from all cameras are taken in regular intervals, and the local poses of the calibration object for all successful detections are stored. In the second step, both internal and external camera parameters are estimated simultaneously from the stored poses. This is achieved by treating the calibration as a bundle-adjustment problem, which is solved using a sparse Levenberg-Marquardt (LM) optimization algorithm. The approach is experimentally evaluated on the framework described in (1). Using the reprojection error as a metric, the accuracy of the presented calibration method is compared against results obtained by other researchers for multi-camera system calibration, as taken from related work. The results of the comparison indicate superior performance of the described approach.

- (3) **A novel method to model the appearance of different objects belonging to an object class with known geometric properties for tracking is presented.** As a restriction, the camera parameters have to be known for the presented method to be applicable, and the movement of the objects has to be restricted to a plane, the topography of which has to be known. Consequently, the approach is best suited for man-made environments, such as indoor areas. The approach employs a non-deformable geometric model constructed from regular polygons to model the shape of the object. The model is then divided into parts which are expected to share certain appearance properties, such as color statistics. Using multiple observations from varying observation perspectives, the appearance properties for the model parts are computed from the appearance properties of the entire object, which allows for an extrapolation of the appearance of the object under arbitrary observation perspectives. This information can in turn be used in multiple ways, *e.g.* to refine the testing of hypotheses when employing a particle-filter based approach for tracking. As a proof of concept, this approach is exemplatively realized to model the appearance for clothed pedestrians using normalized color histograms as appearance descriptors, and tested against the state-of-the-art within the framework described in (1). The experimental results indicate an improvement for the categories of single-view tracking, multi-view tracking, and target identity recognition.

1.2 Outline of the Thesis

To reflect the groups of contributions mentioned in the previous chapter, the remainder of this thesis is organized in four chapters. Aside from the last chapter, these chapters are thematically grouped, where each chapter focuses on a specific part of the whole task, and follows a generic internal structure with the presentation of challenge, solution idea, approach, experimental evaluation, and results. In detail, the chapters are arranged as follows:

- Chapter 2 on page 11, titled “Real-Time Multi-View Pedestrian Tracking”, contains the complete top-to-bottom description of a distributed camera system designed for real-time indoor surveillance, focusing on scalability and modular design. The chapter describes hardware as well as software components. This chapter provides the most general overview over the whole challenge presented in the previous sections,

and the following chapters integrate into the framework provided by it. Its main contribution lies with the vertically integrated, three-layered architecture composed of state-of-the-art components and methods.

- Chapter 3 on page 79, titled “Appearance Modeling”, is focused around a lean pedestrian appearance model, designed specifically with a real-time multi-view tracking application in mind. This chapter integrates with the previous chapter by providing improvements for the methodology employed for color-based pedestrian detection and tracking. The contribution lies with the presentation and evaluation of a novel approach to adaptively model target appearance of multi-colored objects based on observation perspective for use in color-based tracking.
- Chapter 4 on page 123, titled “Applications”, provides application examples for two modular extensions of the camera system presented in this thesis. The contribution lies with the demonstration of the versatility and extensibility of the system described in Chapter 2 for additional surveillance and action interpretation tasks, beyond its main application focus of pedestrian tracking.
- Chapter 5 on page 135, titled “Summary and Outlook”, concludes the thesis by summarizing results and scientific contributions from the previous chapters, and providing an outlook into further application and research opportunities tying in with this work.

Chapter 2

Real-Time Multi-View Pedestrian Tracking

The surveillance of large structured environments (*e.g.* train stations, factory halls or street sections) is a challenge that surpasses the limits of single-camera smart surveillance approaches. Among other factors, occlusion and camera resolution impose limitations on the maximum area that can be covered by a single monocular camera. For indoor environments, this is compounded by the limitations of camera FOV induced by room size (*e.g.* wall-to-wall distance, ceiling height). Consequently, when aiming for coverage of a sufficiently large area, information gathered from multiple camera views has to be combined. This chapter specifically focuses on the problem of tracking pedestrians (*cf.* Section 2.3.1 on page 14) through multiple views within such an environment, and the challenges to be expected under these circumstances.

2.1 Problem Statement

From an algorithmic perspective, according to Cai *et al.* [37], the problem of multi-view person tracking can be divided into a series of single-view person tracking problems and view transition problems. The single-view tracking problem in turn can be broken down into the initial detection problem, where a pedestrian has to be detected within an image without any previous information, and a subsequent tracking problem, where information about the existence of a pedestrian and its previous position within the target area can already be considered.

However, apart from the algorithmic perspective, concerns regarding hardware and software architecture have to be addressed when considering a system which is able to provide the tracks of several pedestrians across multiple views in real-time. Efficient coverage of the target area requires a systematic approach to sensor selection and placement, with the requirements introduced by the desired applications already in mind. An efficient solution has to be found for the provision of sufficient computing power to process the large amount of image data generated by the cameras in real-time, and the capacity of the system to operate for prolonged stretches of time has to be ensured.

A final item to be kept in mind for the conceptual work is the facility with which a surveillance system can be scaled to cover a larger area. Ideally, the scaling process should not require the existing hardware installation to be modified, but simply allow the addition of new parts to the existing configuration to extend the system without interference with the existing setup.

To summarize, the problem covered in this chapter of the thesis can be stated as the design and implementation of a multi-camera vision system meeting the following criteria:

- (1) Optimal camera coverage of an indoor area with a planar floor level (*e.g.* laboratory, factory hall, office), with regard to the tracking of pedestrians, preferably referred to as *target area* in the following.
- (2) Detection of pedestrians and tracking of their position on the target area floor plane with two degrees of freedom (DOF), in real-time.
- (3) Seamless transition of targets between fields of view of adjacent cameras while preserving their identity (*i.e.* avoiding confusion of targets), also in real-time.
- (4) Capacity to operate for extended periods of time, *i.e.* several hours.
- (5) Extensibility of the system, with regard to (a) coverage and (b) functionality, requiring modular design and accessibility of data and results to multiple modules.

2.2 Outline of this Chapter

The remainder of this chapter is organized as follows:

Section 2.3 delivers a discussion of related work on visual pedestrian tracking and related topics, with a special emphasis on multi-view approaches and real-time capable tracking systems.

Section 2.4 on page 26 outlines the solution approach presented in this thesis, provides an overview of the architecture, and explains the rationale behind the design decisions taken.

Section 2.5 on page 29 describes the initial conditions under which the implementation of the system had to be realized.

Section 2.6 on page 30 provides details and specifications for the hardware components integrated in the CoTESYS Central Robotics Laboratory (CCRL) installation.

Section 2.7 on page 42 focuses on the combined multi-camera calibration method employed for the system, and provides a comparison of the results with other state-of-the-art multi-camera calibration methods.

Section 2.8 on page 47 describes architecture and tasks for the first of the two layers constituting the software part of the tracking system, focusing on image buffering and preprocessing.

Section 2.9 on page 50 describes architecture and tasks for the second software layer, focusing on pedestrian detection and tracking.

Section 2.10 on page 62 presents several experiments conducted to validate the concepts and algorithms employed for the system described in the previous sections.

Section 2.11 on page 77 provides a discussion of the work presented in the chapter and summarizes the most important results and observations.

2.3 Related Work

The following is a survey of academic literature pertaining to the topics of object detection and tracking, covering methodology, technical realization, and applications. Special attention is given to the aspects of pedestrian detection and tracking, multi-camera systems, and real-time compatible approaches. The author's observations and conclusions are found at the bottom of each section.

2.3.1 Definitions and Terminology

When delving into the wealth of academic literature pertaining to the topic, it is quickly noted that some variations exist in the terminology. Therefore, it seems prudent to clarify a few terms according to their use in this thesis.

For the purpose of this thesis, a *pedestrian* is considered to be a human with their posture limited to being upright, that is either standing or ambulating. This coincides with the definition given by Gray *et al.* [102]. At some points of this thesis, the terms *person* or *human* may be used synonymously to refer to a pedestrian, *e.g.* *person tracking* refers to the tracking of pedestrians. Note, that in contrast to this thesis, most research papers literally referring to pedestrians are written from an automotive background, and therefore almost exclusively regard pedestrians from a lateral perspective.

Tracking is the process of repeatedly locating an object over a period of time. Consequently, the set of object locations obtained this way is called the *track*, whereas a consecutive subset of the track is referred to as *tracklet*. For the purpose of this thesis, unless stated otherwise, the term tracking is sloppily used to refer to *visual tracking*, which refers to the process of tracking using a visual sensor, such as an eye or a camera. In the sense of tracking, the object that is being tracked is referred to by the term *target* in the following.

The term *appearance* refers to the properties of an object that can be visually observed (*cf.* Hunter and Harold [127]). The most important properties falling under these definitions are properties of the object surface (color) and shape. While in theory object surfaces might have transmissive as well as reflective properties, for the purpose of this thesis, only reflective properties are considered due to the nature of the objects being modeled. It should be noted, that due to the optical sense relying on light reflected by the object, appearance properties are subject to change upon variations in illumination conditions. Consequently, the term *appearance model* refers to any approach to modeling the appearance of an object, *e.g.* using color or brightness statistics of its digital image. In the sense of employing appearance models in tracking, a *static appearance model* refers to an appearance model that does not vary over time, the opposite of which is termed an *adaptive appearance model* here.

For further definitions and clarifications on the terminology employed in this thesis, the reader is kindly referred to Chapter C on page 167.

2.3.2 Methods of Pedestrian Tracking

The topic of visual pedestrian tracking merits an overview of the plethora of methods devised to cope with its inherent challenges.

Regarding the difference between tracking and detection, there are two possibilities to approach visual object tracking in general that also transfer to pedestrian tracking. On the one hand, each frame in a sequence can be treated entirely as an individual image, applying the same detector algorithm (*e.g.* Haar feature-based [286] or histogram of oriented gradients (HOG)-based detectors [77], *cf.* Dollár *et al.* [68] for a survey comparing different algorithms.) to every single frame, a method that is referred to as *tracking-by-detection* [5, 30].

In contrast, *Bayesian tracking* utilizes the eponymous theorem [14] to exploit the information about the prior state (*i.e.* estimated prior positions and confidence) of the tracked pedestrian in every subsequent frame after the initial detection. This method incurs the advantage of being less costly from a computational perspective, as features only have to be sampled for a smaller portion of subsequent images in contrast to a full-fledged detection approach. Furthermore, the approach also has benefits when tracking multiple targets with similar appearance, since confusions are less likely due to the fact that previous tracks are being considered, *cf.* Kettner and Zabih [153].

As a further distinction, approaches are divided into *marker-based* approaches, *i.e.* those using any type of *fiducial* or marker attached to the target, and *markerless* approaches, which do not require any such expedients to operate, relying instead on descriptive features of persons that can directly be extracted from the image.

Markers, or fiducials as they are also frequently referred to, are objects whose properties are well defined, and that are attached to the target *a priori*. They exist in different shapes and sizes, and depend on the exact type of sensor used. Examples include infrared markers (*cf.* Maeda *et al.* [180]), color markers (*cf.* Wang *et al.* [290]) or binary black/white patterns such as ARToolkit or ARTag markers (*cf.* [83, 149]). As a general rule, it can be stated that marker-based approaches are capable of supplying increased robustness (*i.e.* reliability under adverse conditions) and accuracy at the cost of a narrowed-down area of applicability. Therefore, these approaches are especially suited for high-precision requirements in controlled environments, such as human motion capture (*cf.* Moeslund *et al.* [194, 195] for a survey on the topic, and Kirk *et al.* [157] for an application example.). Since they are well defined and incur little risk of being confused with each other, markers

are commonly tracked using tracking-by-detection approaches (*cf.* Zhang *et al.* [301] for a comparative study).

Conversely, markerless approaches can be applied under a more varied set of circumstances, which makes them especially suitable whenever there is little to no control over targets. Examples include surveillance tasks, where targets are usually non-compliant (*cf.* Fuentes *et al.* [89] or Wei *et al.* [205]), or the analysis of images provided by third parties, such as television broadcasts of sports events (*cf.* Watanabe *et al.* [291] or von Hoyningen-Huene and Beetz [121] for televised soccer; Pingali *et al.* [218] for tennis matches).

Markerless approaches to pedestrian tracking can be further differentiated into those operating solely on the image data, resulting in a track of two-dimensional (2D) image positions (*cf.* Comaniciu *et al.* [49]), and approaches operating on the three-dimensional (3D) position of objects, which can be obtained using calibrated cameras (*cf.* Balan *et al.* [11]). While an approach operating on real world position data allows for the inclusion of human motion models (*cf.* Arechavaleta *et al.* [7] for unconstrained locomotion at floor level; Urtasun *et al.* [278] for diverse activities) to improve hypothesis generation in the prediction step, it should be noted that the frequent projections and re-projections of target positions between world space and image space introduce another possible source of numerical instability.

Regarding the features of pedestrians being tracked in markerless approaches, one set of methods can be classified as holistic, where appearance descriptors are used to describe the body as a single monolithic entity. Examples include the approaches of Comaniciu *et al.* [50], using histograms on scale-invariant ellipsoidal regions of kernel-transformed images, Gandhi and Rivedi [90], using panoramic color appearance maps, and Allen *et al.* [2], using ratio histograms in hue/saturation/intensity (HSI) color space for tracking with the CamShift [28] algorithm. Those can be distinguished from atomistic approaches, where different descriptors for different parts of the body are employed. To provide an example, Izadinia *et al.* [129] propose a tracking method where different body parts are tracked independently using tracking-by-detection [77] with HOG [62] descriptors, and merged by flow network optimization [219].

To conclude, the challenge of tracking pedestrians has been tackled under a multitude of constraints with a plethora of methods, each of which have distinct advantages and disadvantages. The system investigated for this thesis uses a holistic, markerless approach to visual tracking, employing Bayesian tracking in the image domain with calibrated cameras to infer world position of targets. This kind of approach is well suited for semi-controlled

environments, where there is free access to cameras, but not to targets. In other words, this thesis describes an approach operating under typical circumstances for an indoor surveillance application.

2.3.3 Visual Tracking

Panin [211, p.8] proposes a *tracking pipeline* (cf. Figure 2.1 on the next page) as general *modus operandi* for visual object tracking, consisting of the following steps for every iteration:

Step 1: *Data acquisition* from sensors, providing image data and time stamps

Step 2: *State prediction* by Bayesian tracker at the given time stamp, providing multiple hypotheses

Step 3: *Preprocessing* of new sensor data, independent of hypotheses (*e.g.* color space conversion, background subtraction etc.)

Step 4: *Feature sampling* from the target hypotheses.

Step 5: *Data association*, where the sampled features are matched against the image data to produce measurements for the target.

Step 6: *Data fusion*, where target-associated data from all cameras and modalities is combined to produce a global measurement vector.

Step 7: *State update*, where the maximum a-posteriori (MAP) likelihoods for each target are computed to provide an output state.

Step 8: *Feature update*, where the model state is exploited to sample online reference features for the subsequent frame.

Regarding adequate features applicable in Steps 4 and 8, van de Sande *et al.* [239] provide a comprehensive evaluation of different types of color descriptors to be used for object recognition. The work of Ozturk *et al.* [210] deserves special attention here, since they tackle the problem of tracking pedestrians in indoor environments from a similar top-view perspective as presented in this paper. They employ histogram models in RGB color space with sequential importance resampling (SIR) particle filtering. Additionally, they employ scale invariant feature transform (SIFT) [178] flow vector matching against manually annotated data to determine the orientation of tracked targets.

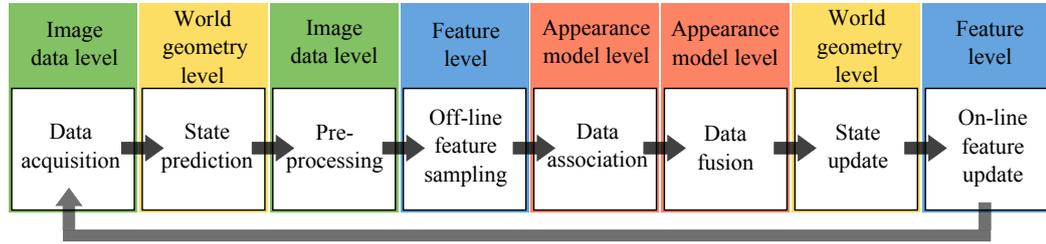


FIGURE 2.1: Pipeline for visual object tracking, as proposed by Panin (*cf.* [211]). In addition, the distinct steps have been assigned to the information levels, as proposed by the author (*cf.* Figure 2.2 on the facing page).

2.3.4 Tracking and Data Association Algorithms

Regarding the *state prediction* and *data association* steps for Panin’s pipeline model referenced in the previous section, the literature provides a plethora of methods to choose from.

As mentioned in Section 2.3.2 on page 15, the underlying problem is the estimation of the probability of a state regarding the given previous state. this requires a Bayesian concept of probability [14] and is therefore often referred to as Bayesian tracking or Bayesian application. For the tracking of multiple targets, one major discriminating quality between methods is whether the states of different targets are updated sequentially (*e.g.* SIR, sequential Monte Carlo (SMC) [39]/ MCMC approach) or simultaneously (*e.g.* joint probabilistic data association (JPDA) filter).

MCMC, which implements the Metropolis-Hastings Algorithm [112, 190] can be considered one of the standard solutions for this particular problem (*cf.* Geyer [94], Smith and Roberts [251] and Green [103] for tutorials on the subject). On the other hand, Karlsson and Gustafsson [147] propose a JPDA filter, while von Hoyningen-Huene and Beetz [122, 123] propose a Rao-Blackwellized SIR particle filter. Vermaak *et al.* [282] compare Markov chain JPDA filter, sequential sampling particle filter (SSPF), independent partition particle filter (IPPF) regarding the applicability of the approaches for multi-target tracking. They conclude, that the MC-JPDA filter outperforms the other proposed methods regarding convergence and ability to deal with multiple targets.

2.3.5 Tracking with Multiple Cameras and View Transition

Regarding the general problem tracking of targets across multiple cameras, the paper of Cai *et al.* [37] can be considered a seminal work, in that it proposes a comprehensive theoretical framework for multi-view transition tracking and establishes the method of dissecting the multi-view tracking problem into an alternating sequence of single view tracking problems and view transition problems to be tackled individually, as already briefly touched upon in Section 2.1 on page 11. Since techniques for single-view tracking have been discussed comprehensively in the previous sections, approaches to address the view transition challenge shift into focus at this point.

Observation I: Tracking across multiple cameras introduces the problem of view transition. In this thesis, multi-view tracking is treated as an alternating series of single-view tracking and view transition.

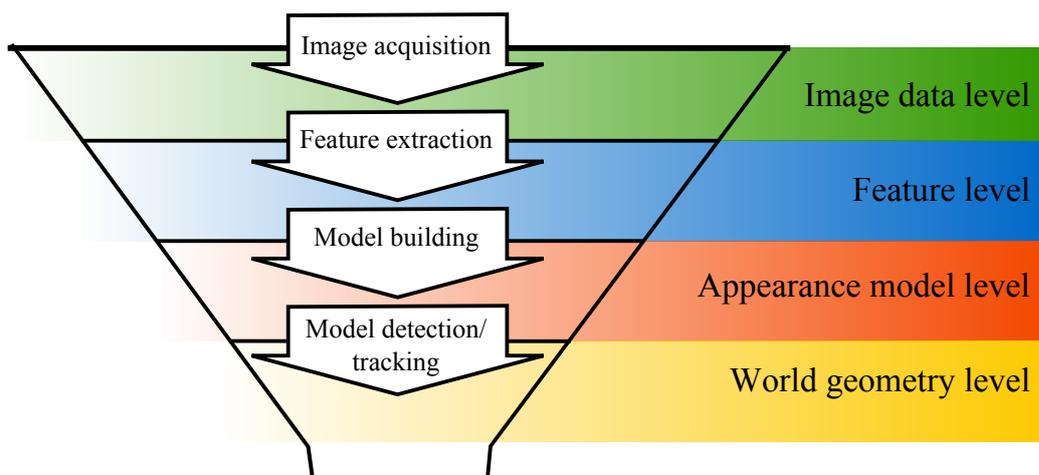


FIGURE 2.2: Funnel model for information condensation during successive steps of image processing, specifically object tracking. Information on the object of interest is extracted and condensed from top to bottom, discarding redundant parts while preserving relevant bits. This reduction in data significantly enhances processing speed for certain operations (*e.g.* data fusion) when performed on lower levels as opposed to higher ones.

The challenge of robust target transition between camera FOVs has been tackled before under various conditions, yielding many different approaches. For example, Khan *et al.* [154] consider the transition of targets between intersecting camera FOVs by establishing intersection lines for uncalibrated

cameras, while Javed *et al.* [132, 133] approach the problem of target transition between non-intersecting cameras by employing machine learning to discern human path conformity, inter-camera relationships and inter-camera brightness transfer functions. Kuo *et al.* [166] tackle the same challenge using appearance affinity models which are learned during operation.

Observation II: The class of view transition problems can be divided into two subclasses, transition of intersecting views and transition of non-intersecting views. This thesis is primarily concerned with the former.

One distinction to be pointed out with regard to compatibility to real-time result availability is whether identities of tracked targets are maintained during tracking (*cf.* Javed *et al.* [131]; Nummiaro *et al.* [206]; Fleck *et al.* [85], all of whom use SMC for that purpose), or whether they are assembled *post facto*. The work of Zamir *et al.* [299] on multi-target tracking, using generalized minimum clique (*cf.* Karp [148]) graphs to combine several tracklets (*i.e.* partial tracks) with uncertain identities to tracks with a single unique identity, serves as an example for the latter.

Observation III: Tracking targets in single views yields tracklets. These can be combined to tracks either when view transition occurs, or during a post-processing step. SMC is a preferred approach for the former. In this thesis, it is selected to allow for the generation of tracking results in real-time.

2.3.6 Real-Time Multi-View Tracking Systems

In theory, the most convenient setup to process images from multiple cameras would be centralized processing on a single computer. While this is certainly a feasible approach in situations where processing speed is not critical, it becomes increasingly difficult to realize in real-time as soon as the number of cameras exceeds a certain threshold, which is derived from one of two bottlenecks. Either the combined data rate of the cameras exceeds the networking capacity of the machine (*e.g.* four GEV cameras on a 1 Gbit network interface controller (NIC)), or amount of processing required exceeds the power of the central processing unit (CPU), or graphics processing unit (GPU), respectively.

As a result of this, real-time image processing with multiple cameras usually requires a dedicated, distributed architecture, where different network controllers and processing units supply sets of cameras. Different architectures impose restrictions on the level of interaction between the processing of images from different cameras, *i.e.* the level on which data from multi-

ple cameras may be combined (*e.g.* image data, feature level, object poses; *cf.* Figure 2.2 on page 19)

One approach to this challenge are smart cameras (*cf.* Belbachir [18]), where image processing is performed directly on a dedicated processor built into the camera. The resulting architecture leads to a very linear processing approach, where images from all cameras are processed independently, without any exchange of information beyond the result level.

The 3D SURVEILLANCE system proposed by Fleck *et al.* [85] implements an architecture consisting of a server node and multiple camera nodes, realized preferably as smart cameras, or alternatively as camera/personal computer combination. The cameras are mounted statically in the environment, which is exploited for target detection by application of foreground segmentation. They use a color-based particle filter with histograms in HSI color space (*cf.* again Fleck *et al.* [86]), and report a live frame rate of 15–17 Hz for single or dual targets, respectively.

Regarding approaches using spatially separated image acquisition and processing, Javed *et al.* [131] present an approach combining single camera tracking with a voting algorithm based on color and shape cues, with automated FOV-line detection for view transition between uncalibrated cameras. Their system follows a modular architecture (*cf.* Figure 2.4 on the following page), runs the module for each camera on a separate PC, and is capable of operating at 10 Hz.

To give a further example, Straw *et al.* [260] propose a system to track the movements of flies and birds for neurobiological studies using 11 GEV cameras supported by 9 Pentium 4/Core 2 Duo computers; using extended Kalman filter [143, 144] and nearest neighbor standard filter [13] for data association, reporting a cycle time of 40 ms, *i.e.* 25 Hz.

Nummiaro *et al.* [206] introduce a real-time multi-view tracker operating on calibrated cameras, using color-based particle filtering [207] as in the previously described approach. They constrain their descriptors to human heads, and use multiple a priori trained model histograms to account for perspective changes. Their system is reported to run at 5–8 Hz on Pentium III personal computers with 160×120 px video feeds, with a 1:1 mapping of cameras and computers.

Zhao *et al.* [304] published a real-time vision system using multiple camera nodes realized by stereo-vision sensors, with monocular person detection and result fusion for each camera node (*cf.* Figure 2.5 on page 23). They address single view tracking through the expectation maximization (EM) al-

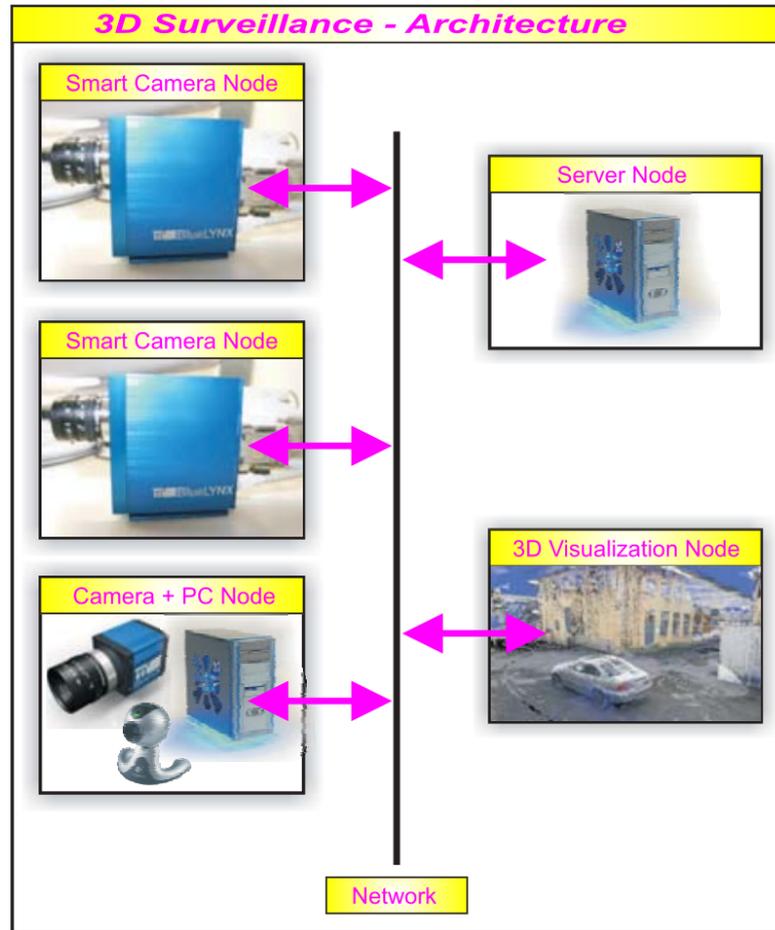


FIGURE 2.3: Schematic of the architecture for the 3D Surveillance real-time vision system, using smart cameras. Taken from [85].

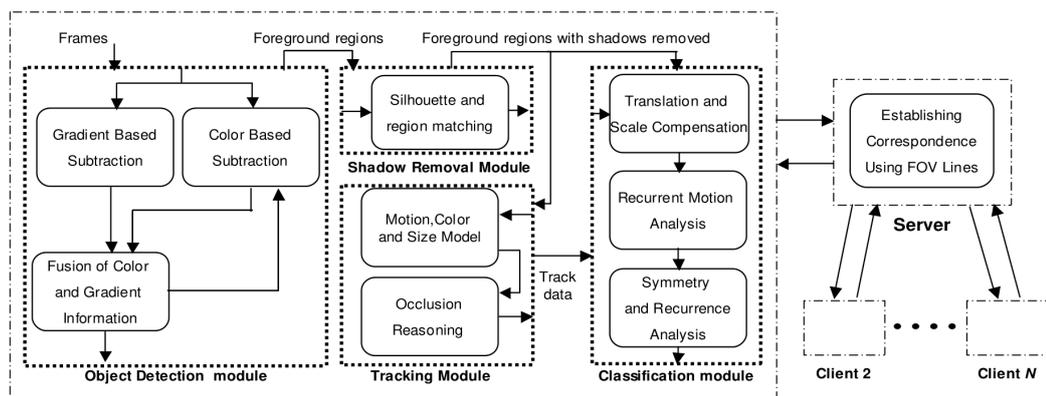


FIGURE 2.4: Schematic of the modular architecture for the KNIGHT real-time vision system. Taken from [131].

gorithm [64] using shape, appearance and depth descriptors, and the view transition problem by matching Kalman-filtered estimated object states (*i.e.* position and velocity) in a decision module. The reported operating frequency for the system is 15 Hz, with two camera nodes being connected to a dual Pentium IV computer each for processing of the 160×120 px video feeds.

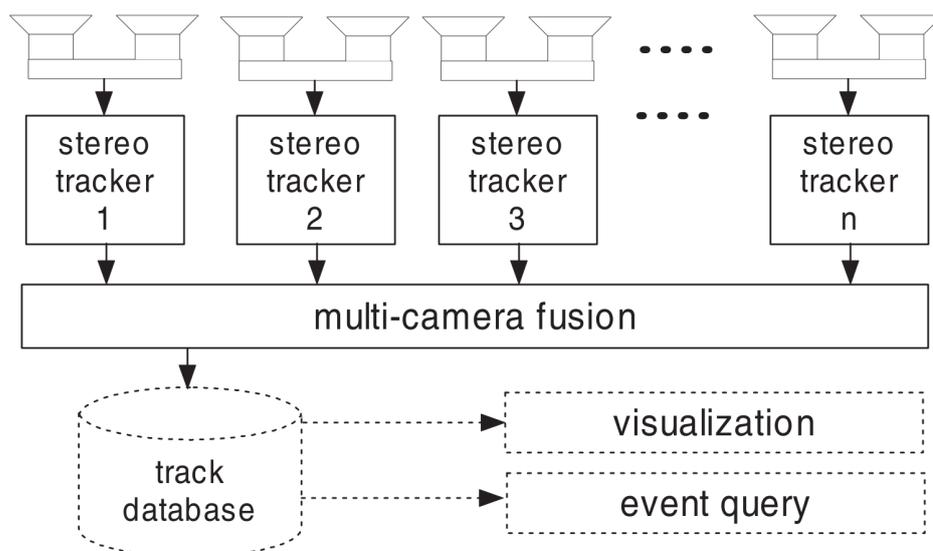


FIGURE 2.5: Schematic of the architecture for the real-time vision system proposed by Zhao *et al.*, taken from [304].

Observation IV: A commonality of the examined real-time approaches (with the exception of [132]) is their reliance on single camera nodes with subsequent fusion of results, although these camera nodes can be realized as stereo cameras as well [304]. This kind of architecture facilitates independent distributed processing, and is therefore conducive to processing speed. Regarding the technical realization, they either rely on smart camera approaches with on-board processing [85], or off-the-shelf (OTS) hardware [85, 206, 260].

The work presented in this thesis opts for the second approach regarding the technical realization, and adopts the independent single-view tracking approach with fusion on the object pose level. Regarding the scope of the system, the work in this thesis exceeds the scope of all examined systems in the categories of number of cameras, camera resolution, amount of data processed, and operating frequency.

2.3.7 Applications of Person Tracking

Intelligent camera surveillance is employed commonly both for security purposes as well as for smart rooms, which can autonomously act on perceived situations. Surveillance systems can operate both in real-time or focus on the post-processing of previously acquired video data. The state of the art for that kind of visual surveillance systems is described in several surveys, such as Valera *et al.* [280] (with an emphasis on distributed systems) or Šegvić *et al.* [247]. A multi-agent-based approach is presented by Patricio *et al.* [213].

Smart rooms also frequently employ visual tracking, such as Lanz *et al.* [171]. Teixeira *et al.* [267] present a camera sensor network for behavior recognition using address-event image sensors and sensory grammars in an assisted living environment. Other approaches using smart-cameras with on-board processing that directly deliver data instead of images are presented by Rinner and Wolf [232] or in Hengstler *et al.* [115], with a focus on application oriented design of the sensor network. A related approach, employing color information and Monte-Carlo filtering while using distributed cameras for processing, is described by Yamasaki *et al.* [297].

Regarding the field of HRI, most visual pedestrian tracking approaches described in the literature are designed to work in real-time, and with cameras installed on the robot's platform. For example, the approach by Nickel and Stiefelhagen [202] is based on using an *a priori* trained skin color model to identify clusters of human skin in the image, which allows for the tracking of head and hands by applying topographical reasoning (*i.e.* head on top). This necessitates a lateral perspective, as applicable when used with mobile robots. Their approach allows for the extraction of pointing gestures from the data, which are of interest in HRI because of their potential to communicate directions. Koenig [160] presents a hierarchical machine-learning-based approach, that combines point cloud data from a time of flight (TOF) camera (for close range, up to 5m) with a HOG person detector for longer ranges, while the tracking component is realized via Kalman filtering. This approach yields only the position of pedestrians, comparably to the work described in this thesis.

Furthermore, optical person tracking systems have been applied as human-computer interfaces (HCIs), particularly for immersion in gaming, both with single-camera and multi-camera systems. This area of application features similarities with the previously discussed HRI, in that it requires real-time processing of the tracking results. In contrast, however, in most cases the coverage area is less extensive, tracking only single individuals. If multiple cameras are employed in this domain, it is usually with largely overlapping

FOVs, with the intention of improving accuracy or extracting depth information (*i.e.* stereo vision). To provide an example, the PFINDER system described by Wren *et al.* [294], uses a single camera facing a person top-down in an approximate 45 degree angle. It is capable of estimating body pose using statistical properties of extracted blobs (*cf.* Pentland [215]), and processing at a frequency of 10 Hz. Among other applications, it has been successfully employed as a control for video games, such as SURVIVE (*cf.* Wren *et al.* [295]). Additional cameras are employed to add stereo vision information for hand tracking and head tracking.

Stødle *et al.* [257] propose a multi-camera tracking system with application in multi-user interaction. Their description is focused on the technical challenges of parallel processing of images from the 16 cameras employed, and reach positive conclusions regarding the scalability of their system. Although the paper is focused on object tracking rather than the algorithmic perspective on person tracking, it provides a good basis to understand the requirements for camera systems in the application domain of HRI. The authors demonstrate the feasibility of multi-camera systems as an input device for video games, which require high accuracy and very low latencies for input processing.

Observation V: There is a broad spectrum of application domains where person tracking approaches have been successfully employed, either with single cameras or multiple-cameras. Areas of application for person tracking differ with respect to the specific requirements regarding coverage area, tracking accuracy and processing speed requirements. Furthermore, as a consequence of the different coverage requirements, they differ in the way multiple cameras are commonly employed – either to extend the area of observation, or to provide higher accuracy through the use of stereo information.

Although this thesis deals with multi-camera tracking for the most part, similarities exist with single-camera approaches in that tracking is performed in single camera views with little overlap. This constitutes a technique found mostly in the surveillance application domain. On the other hand, the data is processed in real time, in which the work described in this thesis exhibits proximity to approaches found in the application domains of smart rooms, HRI and HCI.

2.4 Solution Idea

The initial abstract concept for the solution idea consists of an architecture of three distinct layers: hardware layer, preprocessing/service layer, and image processing/application layer. These layers aim to cope with the challenges presented in the previous section, which is detailed out in the following paragraphs. In addition to the three continuously operationable system layers, maintenance operations such as photometric camera calibration [220] and camera resectioning [277, 302], called camera calibration in the following in accordance with standard nomenclature in the field, have to be performed whenever changes are made to the camera setup.

As the goal consists of a system applicable on a setup of static cameras, the next step toward a solution is to perform an accurate complete calibration of the whole camera setup against a common coordinate system [74], determining intrinsic and extrinsic parameters, which can later be used in the different system layers (*cf.* Figure 2.6 on the facing page) for image processing tasks. Additionally, it is assumed that persons being tracked in the setup will move on a plane, which can also be determined during the calibration procedure.

Hardware The issue of coverage extensibility is closely linked to the hardware used in the setup. Installation of the cameras at maximum possible height near the ceiling, facing top-down towards the floor plane of the target area, ensures minimal occlusion with regard to pedestrian movement tracking, which has been emphasized in the problem statement as the key task to be solved. In combination with arraying the cameras in a grid, this setup allows for easy extension of the coverage area by adding more cameras at the edges of the current observation area, without the need to modify the existing setup.

To supply sufficient processing power for an increased number of cameras, images from different cameras have to be processed on different computers. The proposed system consists of multiple *camera nodes*, consisting of N cameras connected to a *processing node* that handles the image processing, and a single *server node*, similar to the systems showcased in Section 2.3.6 on page 20 [131, 260, 304].

The ideal N for the mapping of cameras to computers depends on two factors, real-time processing constraints and networking constraints, which both impose an upper limit for N . For complex and computing-intensive image processing tasks such as pedestrian tracking, the real-time processing constraints are expected to outweigh the networking constraints, therefore it is

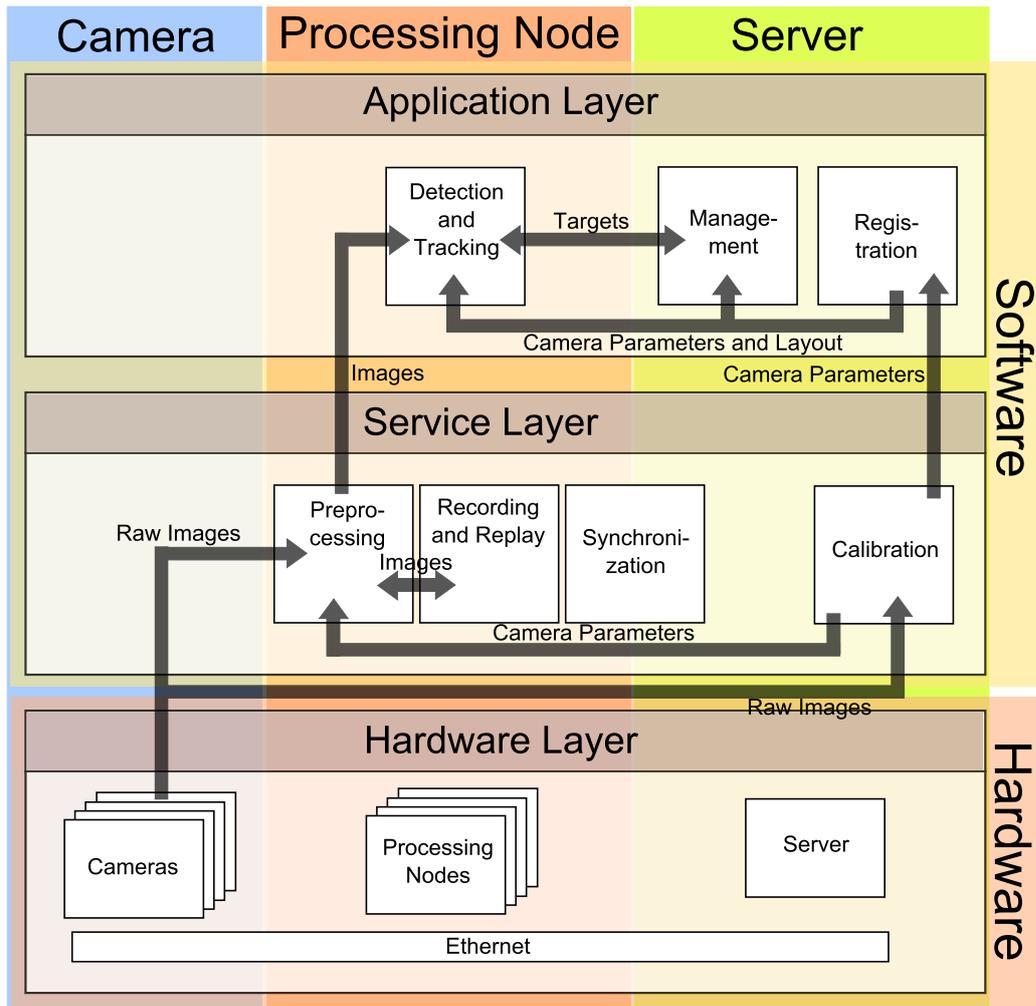


FIGURE 2.6: Schematic of the layered modular architecture of the proposed camera system.

expected that $N \approx 1$.

By using mainly off-the-shelf hardware for the processing and networking, and by physically separating the coverage area and the space where the image processing units are set up through the use of Ethernet-connected cameras, extensibility in these domains is again facilitated.

Buffering and Preprocessing To address the issue of functional system extensibility, a modular approach regarding the software architecture is taken, splitting the architecture into two separate layers. From bottom to top, the *maintenance layer* is responsible for the acquisition of the raw images from the cameras, as well as performing common preprocessing steps required by modules in the following layer, *e.g.* removal of lens distortion from the images [283]. Furthermore, this layer buffers the images, making them available to several modules from a higher layer simultaneously, and allows for the simultaneous recording and playback of images from all cameras. One instance of the maintenance layer is to be active for each camera. By means of the buffer, the layer also abstracts from *camera* to *view* \mathfrak{V} , since images can also be buffered from another source, such as previously recorded or artificial image data.

$$\mathfrak{V} = (\mathcal{I}, \mathfrak{C}, \mathcal{J}) \quad (2.1)$$

Image processing On top of that, the *image processing layer*, which is situated on the processing nodes, exhibits an essentially modular structure, where each module receives their images from the maintenance layer buffer for further processing. For the scope of this thesis, attention will be focused on the development of a module capable of pedestrian tracking.

As stated above, the challenge of multi-view pedestrian tracking can be decomposed into a series of single-view tracking challenges and view transition problems. Regarding a single view, the challenge of pedestrian tracking can be broken down into detection and data association. To approach the real-time observation challenge efficiently, rather than implementing a tracking-by-detection approach, a Bayesian approach, *e.g.* SMC [70], is used in combination with Brownian motion, modeled by the Wiener process (*cf.* Karatzas [145], Wiener [292], Brown [33]) and a color histogram model in HSI color space for each pedestrian, to tackle the data association problem and facilitate the detection problem past the initial detection step.

Taking the multi-view challenge into account, distributed processing has to

be considered because of the constraints explained above. To provide an abstraction from the hardware layer, each camera is assigned its own instance of the image processing module, even if multiple cameras are connected to a single processing node. Inter-process communication between concurrent modules is handled via middleware, *e.g.* Common Object Request Broker Architecture (CORBA) [197] or ICE [116], regardless of the physical machine the module is being run on. The view transition problem explicitly requires modules to communicate, in order to preserve target identities in real-time. The tracking module is initialized with the position and color model transmitted from the module handling the previous view. This is made possible because of the multi-camera calibration performed during system setup, and serves to omit the time-consuming step of initial pedestrian detection for tracking in subsequent views.

2.5 Conditions at the Laboratory

The CCRL was created in 2008 when the Cluster Cognition for Technical Systems (COTESYS) was approved as part of the German “Excellence Initiative” by the federal government. The work focus of the laboratory was to provide an environment for research into human-robot interaction, involving scientists from different fields of engineering as well as computer science, medicine and psychology. Figure 2.9 on page 32 depicts some example scenarios to illustrate the broader vision of interaction scenarios considered at the laboratory. The remainder of this section describes the initial conditions present in the CCRL prior to the installation of the camera system. These conditions are relevant for design decisions taken during the camera systems’ development and integration.

For this thesis, the relevant part of the CCRL is the area intended for experiments into human-robot cooperation. The area is 10×10 m wide, and it is part of an indoor laboratory 4 m high. It has been set up as a mock-up of a small apartment, divided halfway by a wall 2.5 m in height. Figures 2.7 and 2.8 on the next page and on page 31 depict the area prior to sensor installation. A metal scaffolding has been attached to the ceiling in 3.2 m height above the floor, to which cameras as well as other sensors, such as infrared tracking devices and omnidirectional microphones, can be attached. At a later date, a workshop mock-up with a used car on a hoisting platform was added to the experimental area to allow for further demonstration scenarios.

The camera system was set up with the intention to provide smart video surveillance of the experimental area over extended periods of time, allow-



FIGURE 2.7: Experimental area at the CCRL, prior to the installation of the camera system.

ing for real-time detection of the positions of humans, robots and relevant objects. The information extracted from the image data can then be made available to the mobile robots to enable them to adjust their behavior during the experiments themselves (*e.g.* driving towards the position of a human hidden behind a wall) as well as being collected for post-experimental analysis. As hinted on above, the major advantage of having a global sensor system in addition to sensors mounted on the robot platforms is the ability to “see everything” within the target area rather than being limited to the line of sight (LOS) of the robots.

2.6 Hardware Layer Installation

The following section focuses on the hardware used in the camera system setup. Over a period of 4 years, there were two phases to the hardware setup, referred to in this section as *initial setup* and *final setup*. A large part of the hardware initially installed was replaced by superior hardware in the second phase. The adjustments were made in order to draw consequences from lessons learned from the setup up to this point, and enable an improvement in the overall performance of the final system (*cf.* Section 2.6.1.1 on page 33).

The different hardware components of the system are described in detail in the following sections. Due to the changes made to the setup, the following descriptions discriminate between the initial setup and the final setup at

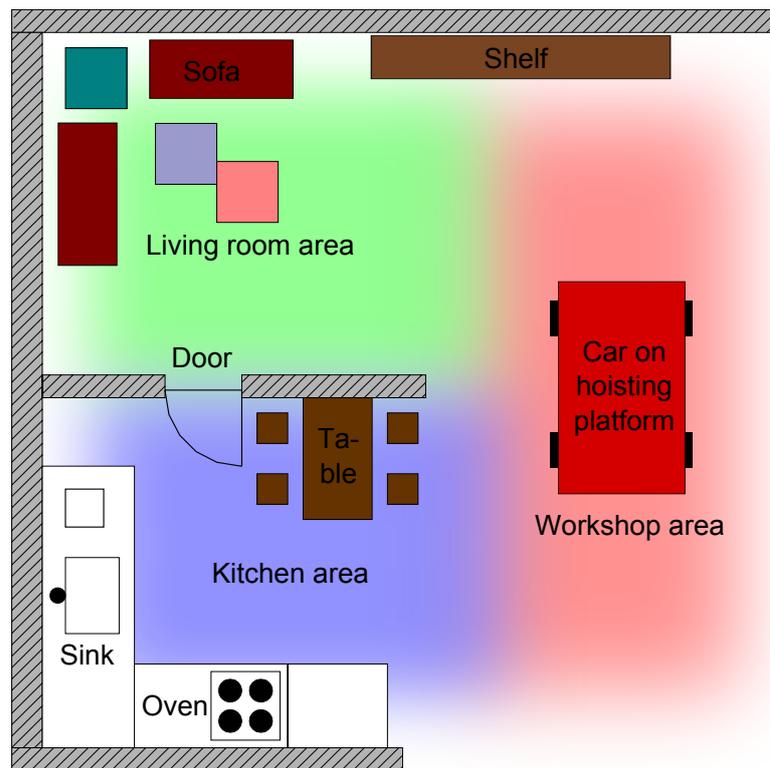


FIGURE 2.8: Schematic map of the experimental area at the CCRL, depicting extent and obstacles. Thematically, the area can be roughly divided into three areas serving as mock-ups for different HRI scenarios: Kitchen, living room and car workshop. Note, that the objects on the map are not entirely up to scale.

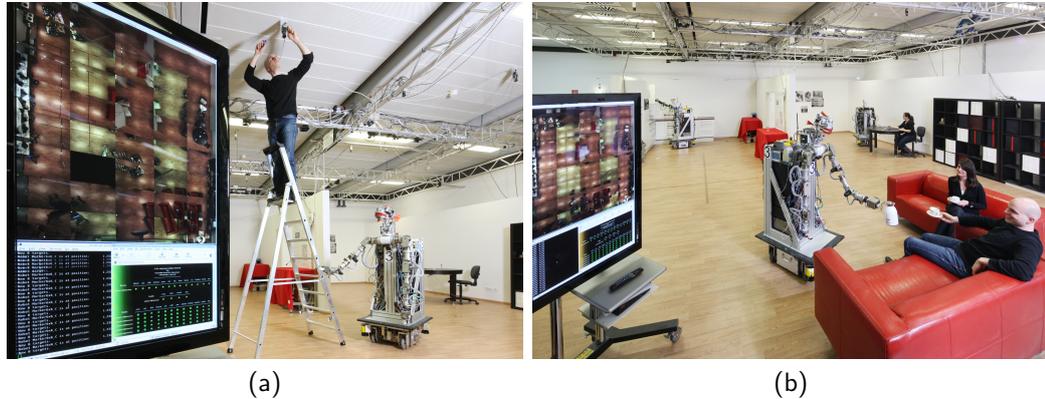


FIGURE 2.9: Two examples to illustrate interaction between humans and robots as envisioned at the CCRL: A robot assisting during a maintenance task on the camera system (a), and a robot serving refreshments to humans relaxing on a couch (b).

some points. During the remainder of the thesis, all mentions of the setup refer to the final setup, unless specifically stated otherwise. In particular, the entire experimental evaluation of the system (*cf.* Section 2.10 on page 62) was conducted on the final setup. To anticipate briefly, the improvements for the final setup consisted of an increase in computing power for image processing, the addition of storage space for images to allow for the recording of images and repeated experiments under similar conditions, and the standardization of the cameras and lenses used in the setup.

2.6.1 Cameras

Several decisions have to be taken regarding the number and type of cameras to be used, as well as the positioning of the cameras. Since one of the main objectives of the system is to monitor the entire experimental area without any gaps, the cameras are set up in a way that minimizes occlusion of pedestrians, by having them face top-down at the experimental area floor plane. This results in the cameras observing the pedestrians from what is termed the *supracranial perspective* for the purpose of the thesis (*cf.* Table B.8 on page 153 for further explanation).

2.6.1.1 Camera Types and Specifications

Although the exact number of cameras required is discussed in the subsequent section, it can be anticipated here that camera coverage of such a comparatively large area without gaps produces a high amount of image data. Consequently, the real-time processing of these images necessitates a large amount of computing power. In further consequence, it is desirable for image processing to be distributed among multiple computers rather than centralized on a single machine. Since a large number of computers requires sufficient space and a dedicated cooling system, it is advantageous in that regard to set up a dedicated server room to concentrate the processing nodes. Therefore, image generation and image processing are spatially separated, and the camera technology to be employed has to reflect this fact. Alper [3] provides an overview of the camera connectivity standards available, specifically CameraLink, GigE Vision and IEEE-1394b (*i.e.* FireWire). Comparing the 100 m range of GigE Vision versus 10 m for CameraLink and 4.5 m for FireWire, Gigabit Ethernet (GigE) [41] cameras are the technology of choice under the above conditions.

These considerations limit the selection of sensors to industrial grade charge-coupled device (CCD) cameras. In the initial setup, two different types of cameras with a similar frame rate are used. a decision which was later revoked in favor of greater uniformity in the system (Phase B). Table 2.1 provides camera types and relevant specifications for the cameras used in both phases of the setup.

Camera Type	N_A	N_B	FPS	Resolution
Baumer TXG08c	30	40	28	1024 × 768 px
Basler Scout scA1000-30gc	10	0	30	1024 × 768 px

TABLE 2.1: Specifications of the cameras used in the CCRL setup, and quantities used in the initial setup (N_A) and the upgraded setup (N_B).

2.6.1.2 Lenses

The photographic objectives for the cameras were selected according to the calculations regarding the number of cameras, *cf.* Section 2.6.1.3 on the next page. It was decided to employ wide-angle lenses with a fixed focal length, suitable for surveillance applications. The primary reason for using fixed-angle lenses is that they are less prone to accidental tampering. This results

in reduced maintenance requirements, *i.e.* manual corrections of the lens angles and re-calibration of the system. Specifications for the precise types of photographic objectives used are given in Table 2.2.

Lens Type	N_A	N_B	Angle of view (at $\frac{1}{3}''$)	Focal length
Pentax H416(KP)	38	40	64.27°	4.2 mm
Tamron M12VM412	2	0	68.8 × 51.0°	4.0–12.0 mm

TABLE 2.2: Specifications of the camera lenses used in the CCRL setup, and quantities used in the initial setup (N_A) and the upgraded setup (N_B).

2.6.1.3 Placement and Number of Cameras

Generally speaking, the number of required cameras depends on the FOV lenses being used, as well as the exact positioning of the cameras. With the decision to have the cameras facing top-down at the target area, possible variations are camera height h_c and distance between cameras. The camera height determines the number of cameras required to cover the area, greater height means greater coverage per camera. The limiting factor for camera height indoor environments is the height of the room, possibly further limited by any ceiling installations such as ventilation (*cf.* Figure 2.7 on page 30). Distance between cameras, on the other hand, has to be small enough to allow the FOVs to overlap to a desired degree at the relevant observation height h_o . The optimal camera setup derived from this constraint consists of a regular grid of cameras, arranged at maximum uniform height. However, it has to be expected that slight aberrations from the optimal setup have to be accepted because of the qualities of the target area itself, *e.g.* obstacles such as the wall in the CCRL experimental area.

As the aspect ratio of the area covered by the camera is usually not equal to 1:1, in the following the terms *primary direction* and *secondary direction* denote the orientation of the longer and shorter axes of the FOV, respectively. With the maximum height of the cameras h_c given as 3.2 m above the floor by initial constraints, the relationship between camera angle α and the covered floor distance in the primary direction d_x is as follows:

$$d_x = 2 \cdot \tan \frac{\alpha}{2} \cdot h_c \quad (2.2)$$

Figure 2.10 on the facing page illustrates this calculation, as well as the

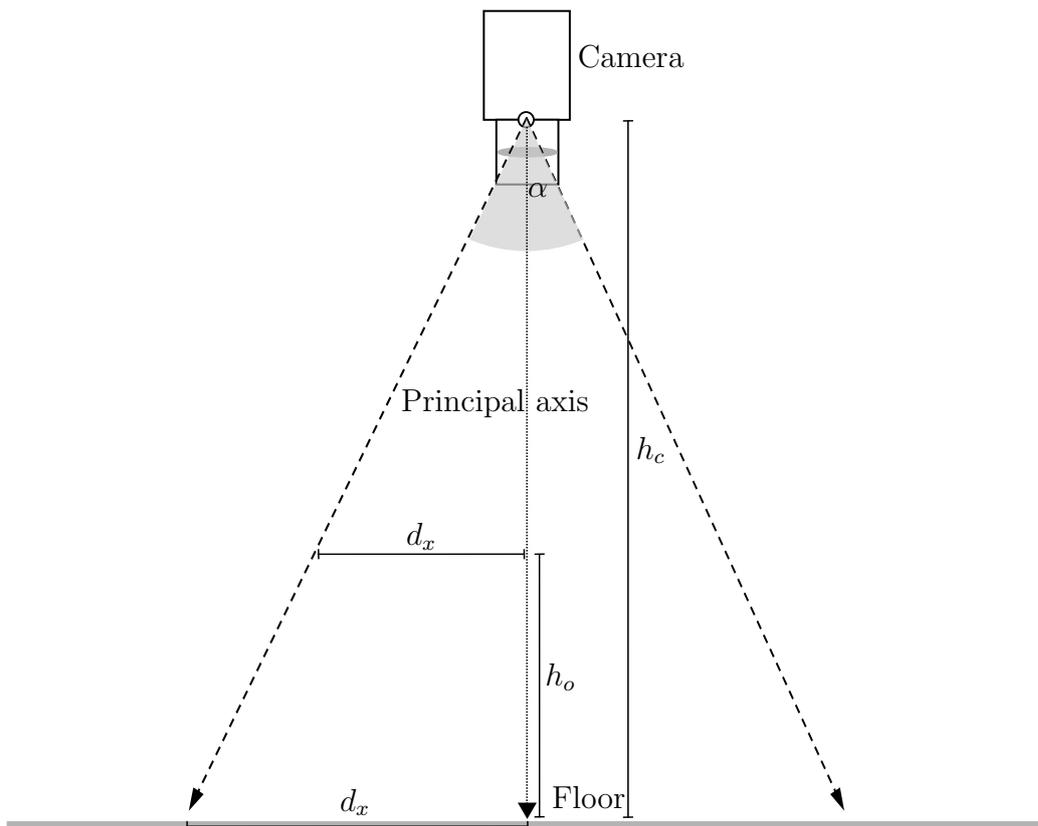


FIGURE 2.10: Two-dimensional scheme of the camera FOV calculation for the covered distance d_x in the primary direction depending on camera angle α , camera height h_c and observation height h_o .

following calculations for d_x . Assuming the use of a CCD chip with a common aspect ratio a of 4:3, the covered distance in the secondary direction d_y can be deduced by

$$d_y = \frac{1}{a} \cdot d_x = \frac{3}{4} \cdot d_x \quad (2.3)$$

To reliably monitor humans and robots, complete coverage of the scene, without gaps, at reference height h_o above the floor plane is required. For the following considerations, a reference height of $h_o = 1.7$ m is assumed, which equals the average height of an adult person (*cf.* Ogden *et al.* [208]). Consequently, the equation for the covered distance d_x at height h_o reads as follows:

$$d_x = 2 \cdot \tan \frac{\alpha}{2} \cdot (h_c - h_o) \quad (2.4)$$

Using a lens with an field of view of $\alpha = 64.27^\circ$ (*cf.* Section 2.6.1.2 on page 33), the equation yields a requirement of 5×7 cameras to cover the 10 m distance in the respective directions. For the CCRL setup, it was ultimately decided to use an array of 5×8 cameras to cover a slightly larger area in the secondary direction. Figure 2.11 on the facing page depicts the positions of the cameras and their FOVs at $h_o = 1.7$ m.

2.6.1.4 Sampling Density and Redundancy

To provide a metric of the sampling density of the camera system as a whole, the number of pixels per area on a reference plane with distance h_o from the floor plane is calculated. The sampling density ρ_i for a single camera \mathcal{C}_i is constant within the FOV, and is calculated as follows:

$$\rho_i = \frac{(n_x)^2 \cdot \frac{1}{a}}{(2 \cdot \tan \frac{\alpha}{2} \cdot (h_c - h_o))^2 \cdot \frac{1}{a}} \quad (2.5)$$

where n_x denotes the number of pixels in x -direction and a denotes the aspect ratio. The term is reduced by multiplying both numerator and denominator by a , consequently the aspect ratio is irrelevant to the pixel density.

As the height h_c of the cameras varies slightly, so does the sampling density at h_o . To provide an exemplary sampling density, the value obtained for camera \mathcal{C}_{22} (at $h_c=3.26$ m) using the above formula is $2.73 \cdot 10^5 \frac{\text{px}}{\text{m}^2}$. For comparison, the camera's sampling density at floor level is $6.25 \cdot 10^4 \frac{\text{px}}{\text{m}^2}$.

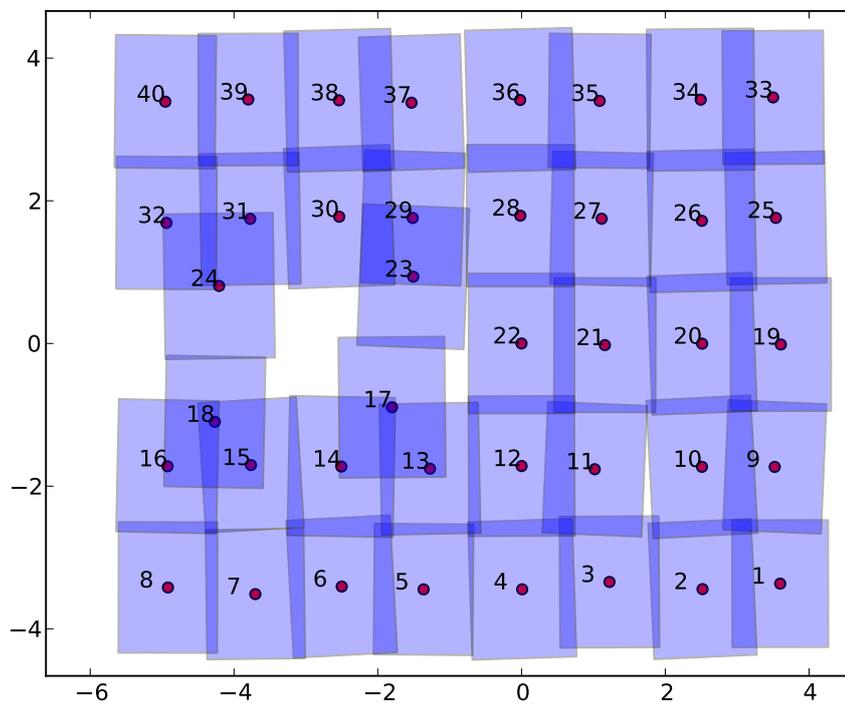


FIGURE 2.11: Plan of camera positions within the target area, and their fields of view at $h_o = 1.7$ m from the floor plane. The conspicuous irregularity in the setup grid to the left of the plan is the result of a wall extending into the target area at this point.

Note, however, that these calculations are slightly inaccurate, as they do not factor in the radial image distortion caused by the lens, which is measured during camera calibration. As a consequence of radial distortion, the sampling density is slightly skewed towards the intersection of the camera's principal axis with the reference plane, which is neglected in the above calculation.

When calculating the pixel density for the complete system, the number is affected by the intersecting areas between camera FOVs. Overlap causes the sampling density to increase, therefore the sampling density is no longer constant. The mean sampling density $\bar{\rho}$ for the whole camera system is calculated as follows:

$$\bar{\rho} = \frac{\sum_1^{n_c} (n_x)^2 \cdot \frac{1}{a}}{x_t \cdot y_t} \quad (2.6)$$

where $n_c = 40$ denotes the number of cameras while $x_t = 10$ m and $y_t = 9$ m denote the total dimensions of the area of observation at the reference plane. With these values, the above formula yields a mean sampling density for the system of $3.50 \cdot 10^5 \frac{\text{px}}{\text{m}^2}$.

From the mean sampling density and the mean camera height \bar{h}_c , the coverage redundancy, *i.e.* the ratio of the combined overlapping FOV area of all cameras to the total size of the area of observation, can be calculated as follows:

$$D = \frac{(n_x)^2 \cdot \bar{\rho} \cdot n_c}{(2 \cdot \tan \frac{\alpha}{2} \cdot (\bar{h}_c - h_o))^2} - \bar{\rho} \quad (2.7)$$

For a mean camera height of $\bar{h}_c=3.21$ m, this equation yields a coverage redundancy of 0.20. Note, that this number for redundancy does not signify the portion of the area that is covered by at least two cameras (which could also be understood as a measure of redundancy), since parts of the area where more than two FOVs overlap are incorporated more than twice in the calculation.

2.6.2 Network Architecture

While the network architecture for the camera system changed with the different phases of the setup, a unifying feature remains that the architecture is divided into two distinct networks, the *client network* and the *camera*

network.

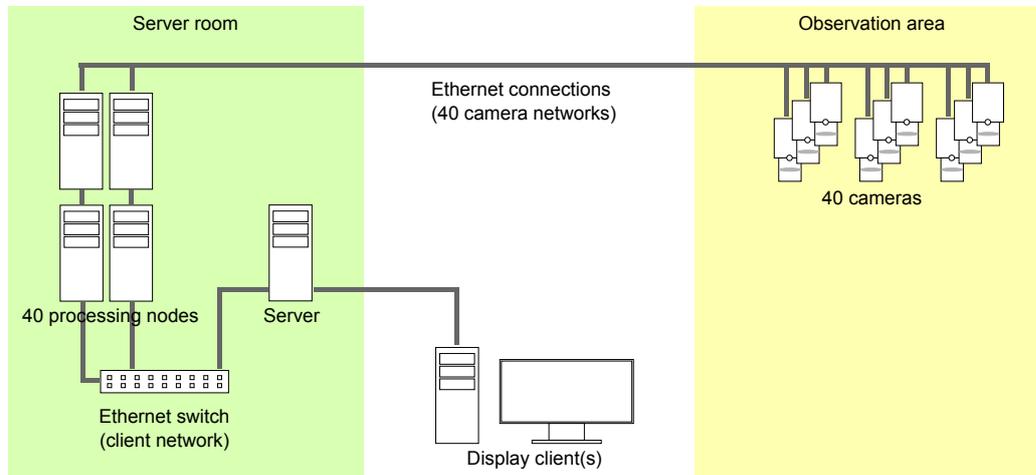


FIGURE 2.12: Overview of the hardware setup that constitutes the camera system, during the second phase of the setup.

The client network interconnects all of the processing clients (described in Section 2.6.3 on page 41) and the server computer, which handles all centralized processing tasks, via an 48-port GigE switch. Data rates required on this network are not critical, since images do not have to be streamed continuously at high frame rates.

The camera network connects the processing clients to the cameras themselves, and is essentially not a single network but a set of uniform networks. During Phase A, each distinct camera network consisted of a processing client and 2 or 3 cameras, connected via 8-port GigE switches. Cameras with adjacent FOVs were assigned to different camera networks, a choice that was made due to the observed fact that human beings in social scenarios such as the coffee-break demonstration scenario tend to flock together, rather than distribute evenly over the target area. Furthermore, with the worst case in mind, this setup reduces the likelihood of adjacent cameras becoming unavailable simultaneously in case of problems caused by single processing nodes, thereby improving system robustness and load-balancing between the image processing nodes.

However, with the increase in processing clients during Phase B, these considerations became obsolete. The camera networks were dissolved, and switches removed so that each processing node directly connects to a single camera.

An important factor for the camera network is the load the network can handle to support continuous streaming of images from the cameras to the

processing clients. Since GEV uses User Datagram Protocol (UDP) packets on top of the GigE-Vision Streaming Protocol (GVSP) over Ethernet, a data overhead of 54 bytes is created for each Ethernet packet, consisting of:

- Ethernet header (14 bytes)
- Internet Protocol (IP) header (20 bytes)
- UDP header (8 bytes)
- GVSP Header (8 bytes)
- Ethernet trailer (4 bytes)

With the header sizes at fixed values, the payload s_p and gross Ethernet packet size s_e depend on the maximum transmission unit (MTU) s_m , *i.e.* size of the IP packet:

$$s_p = s_m - 36 \text{ bytes} \quad (2.8)$$

$$s_e = s_m + 18 \text{ bytes} \quad (2.9)$$

where the 36 bytes result from IP header, UDP header and GVSP header and the 18 bytes stem from Ethernet header and trailer. This results in the following equation for the gross data rate R :

$$R = n_x^2 \cdot \frac{1}{a} \cdot n_p \cdot f_i \cdot \frac{s_p}{s_e} \quad (2.10)$$

where n_x denotes the number of pixels in the primary image direction, a denotes the aspect ratio, n_p denotes the number of bits used to encode each pixel, and f_i denotes the image frequency (*i.e.* number of images/frames per second).

In the described setup, $n_x = 1024 \text{ px}$, $a = \frac{4}{3}$, $n_p = 3 \frac{\text{bytes}}{\text{px}}$, $f_i = 30 \text{ Hz}$ and $s_m = 9000 \text{ bytes}$ (*i.e.* jumbo frames, *cf.* Murray *et al.* [198]). Table 2.3 on the facing page illustrates the load on the client networks using different numbers of cameras and different pixel formats.

Pixel Format / Cameras	1 (Phase B)	2 (Phase A)	3 (Phase A)
8 bit per pixel (bpp)	23.7 MB/s	47.5 MB/s	71.2 MB/s
16 bpp	47.5 MB/s	95.0 MB/s	142.5 MB/s
24 bpp	71.2 MB/s	142.5 MB/s	213.7 MB/s

TABLE 2.3: Gross data rates for continuous streaming on the camera network, using different quantities of 1024×768 pixel cameras with varying pixel formats. For comparison, the maximum data rate on a 1 Gbps connection is 125 MB/s.

2.6.3 Computers Used For Image Processing

For the image processing computers required, it was decided to employ off-the-shelf hardware, which allow for easy extensibility and a flexible exchange of single components in case of technical defects. Initially, 14 AMD Phenom computers were installed to supply one camera group each (consisting of 2–3 cameras) in the first phase of the system layout. Table 2.4 lists the important specifications for these computers.

Component	Type
CPU	AMD Phenom (4 cores)
GPU	NVIDIA GeForce 8600 GT
RAM	2×Patriot 2GB DDR2
HDD	none (diskless)
NIC	Intel PRO/1000 GT Desktop Adapter, 1000 Mbps

TABLE 2.4: Specifications of the processing clients used during Phase A of the hardware setup.

Similar to the aforementioned situation with the cameras, the setup of the processing nodes was changed in Phase B in order to improve system stability and performance. The old processing nodes were replaced by 40 AMD Phenom II computers, now supplying only a single camera each. Their specifications can be seen in Table 2.5 on the following page.

Component	Type
CPU	AMD Phenom II (6 cores)
GPU	NVIDIA GeForce 580 GTX
RAM	4×RipJaws 2GB DDR3
HDD	6×Hitachi DeskStar 7K2000, 2000 GB, as RAID 0
NIC	Intel Gigabit CT Desktop Adapter, 1000 Mbps

TABLE 2.5: Specifications for the processing clients used in the second phase of the hardware setup.

2.7 Camera Calibration

When aiming for accurate global position estimation of any kind of objects or subjects in the target area, camera calibration becomes a necessity. For a multi-camera system, the calibration process proves to be more challenging than for single-camera systems, since it becomes harder to spot errors in the calibration, and manually re-calibrating the system is more time-intensive. Therefore, this step deserves special attention, although it has to be performed only once during the system integration, or in regular maintenance intervals respectively.

According to Tsai [277], full calibration aims to solve a two-step optimization problem:

Step 1: Estimation of the *intrinsic* camera parameters \mathfrak{J} , consisting of the pinhole projection Π and the lens distortion \mathfrak{D} .

Step 2: Estimation of the pose of the camera within a global coordinate system, also called the *extrinsic* camera parameters \mathfrak{E} .

There are several different angles from which to approach this problem for a multi-camera system. It has to be decided which steps are to be solved best independently for each camera, and which parts of the problem might benefit from global optimization opportunities. A common approach for a small number of cameras would be to calibrate the cameras independently against an arbitrarily defined Cartesian world coordinate system, using external measuring to determine the position of the calibration object within the world coordinate system, and subsequently performing the calibration for each camera. For a system with a large number of cameras, however,

this process becomes increasingly cumbersome and error-prone because of the high degree of manual intervention and input it necessitates. Therefore, a combined multi-camera calibration method was employed, using the HALCON [255] image processing libraries.

In HALCON, the i -th camera from a setup of N cameras is specified by two sets of parameters $\mathfrak{E}_i = (\mathbf{R}_i, \mathbf{T}_i)$ and $\mathfrak{J}_i = (\Pi_i, \mathfrak{D}_i)$ modeling the projection of 3D points from the camera coordinate system into the camera image, where $i = 1 \dots N$. Π_i describe a standard linear pin-hole camera projection, whereas \mathfrak{D}_i define a non-linear radial and decentering distortion using a divisional distortion model [170].

A 3D point \mathbf{X}^w in world coordinates is transformed into the camera coordinate system $\mathbf{X}^c = \mathfrak{E}_i \cdot \mathbf{X}^w = \mathbf{R}_i \mathbf{X}^w + \mathbf{T}_i$ and then is projected in the camera image $\mathbf{p} = \Pi_i(\mathbf{X}^c, \mathfrak{J}_i)$ (cf. [170, 255]).

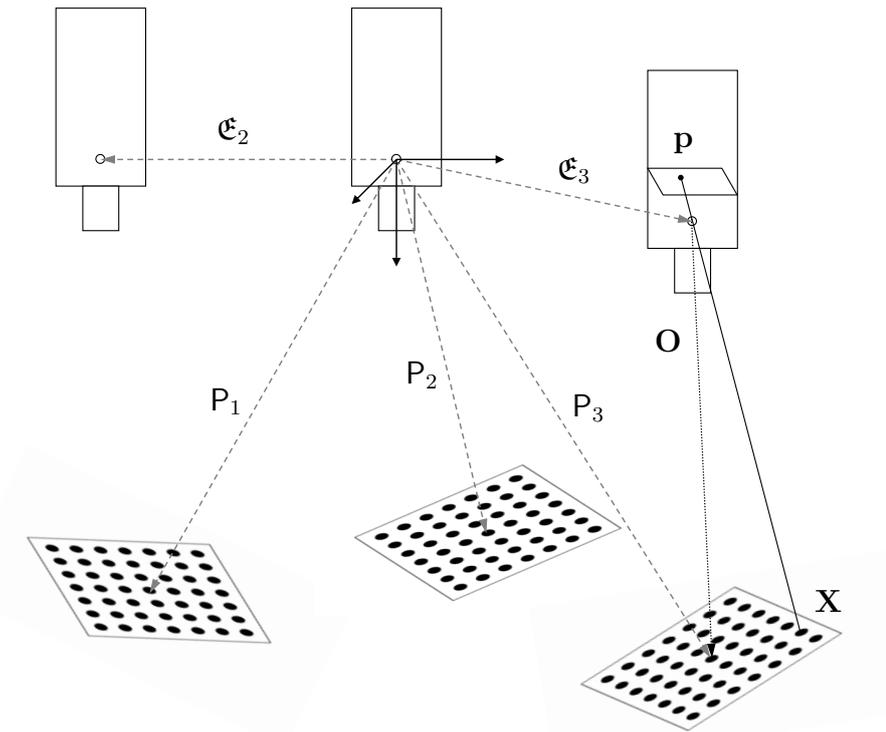


FIGURE 2.13: Multiple camera setup projection model for the calibration step.

A *reference camera* is selected, whose coordinate system is the world coordinate system (cf. Fig Figure 2.13). To calibrate the setup, a known calibration object (as depicted in Figure 2.14 on the following page) with M control

points (marks) is used. Each mark has known coordinates $\mathbf{x}_m, m = 1 \dots M$ in the local coordinate system of the calibration object. The object is exposed in K different poses $\mathbf{P}_k, k = 1 \dots K$, in front of the cameras. Thus the calibration marks define KM control points $\mathbf{X}^w_{(k,m)} = \mathbf{P}_k \cdot \mathbf{x}_m$ in the world coordinate system. For each \mathbf{P}_k , all cameras simultaneously take an image. Images in which the calibration object is not fully visible are ignored.

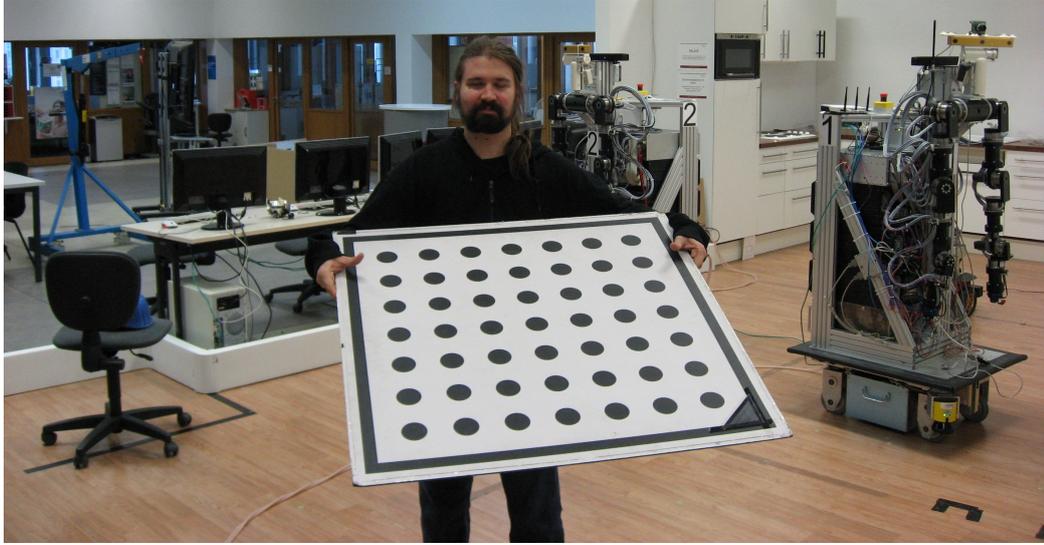


FIGURE 2.14: The calibration object used for the calibration of the camera system equates to a standard quadratic HALCON calibration plate with side length of 0.85 m and 49 circular marks. The orientation is disambiguated by the black triangle-shaped mark in a single corner.

In the presented setup, the centermost camera, Camera 22, was selected as reference camera (*cf.* Figure 2.11 on page 37). To obtain calibration data, the calibration object was slowly moved across the entire experimental area and exposed to the cameras, while varying height, pitch and roll. Concurrently, a total of 9071 synchronized images, representing $K = 2910$ poses, was recorded, with the calibration plate being observed by up to five cameras simultaneously.

The calibration of multiple cameras is formulated as a minimization problem:

$$e^d_{(i,k,m)} = \|\mathbf{p}_{(i,k,m)} - \Pi(\mathfrak{E}_i \cdot \mathbf{P}_k \cdot \mathbf{x}_m, \mathfrak{J}_i)\| \quad (2.11)$$

$$e^d = \sum_{i=1}^N \sum_{k=1}^K v_{(i,k)} \left(\sum_{m=1}^M e^d_{(i,k,m)} \right) \rightarrow \min \quad (2.12)$$

Equation 2.11 is the *reprojection error* for control point m in pose k into the i -th camera image and $v_{(i,k)}$ is 1 if pose k is visible from camera i , and 0 otherwise. This is a typical bundle-adjustment problem formulation, in which an estimation for both the camera parameters and the calibration object pose is found.

Typically, bundle adjustment problems are solved using numerical optimization (*cf.* Triggs *et al.* [276] for a classical survey, Jeong *et al.* [135] for newer approaches). For the iterative numerical approach employed here, initial values for \mathfrak{J}_i , \mathfrak{E}_i and \mathbf{P}_k are required. Each \mathfrak{J}_i is initialized from the product specifications of the respective camera. Then a pose, in which each camera is observing the calibration object in its own coordinate system, can be estimated. Finally, through a chain of shared observations from different cameras on overlapping calibration object poses, the poses of cameras are transformed into the reference coordinate system and used as initial values for \mathfrak{E}_i . All poses of the calibration object \mathbf{P}_k are similarly transformed. The optimization is implemented by a general sparse LM algorithm as described by Hartley and Zisserman [111], which scales linearly with the size of the camera setup.

The circular calibration features of the calibration plate projected onto camera images deform to ellipses, whose centers define the corresponding image points $\mathbf{p}_{(i,k,m)}$. Note that ellipse centers do not represent precisely the projection of the circular center due to perspective and radial distortions. The distortion of the extracted marks is corrected with the calibrated \mathfrak{D}_i , their centers $\mathbf{p}_{(i,k,m)}$ are re-estimated and perspective corrected (as proposed by Heikkilä *et al.* [114]) with the calibrated parameters Π_i . Subsequently, the calibration is performed again with the corrected $\mathbf{p}_{(i,k,m)}$ and the calibrated setup parameters as initial values. The calibration procedure reports the root mean square (RMS) of d as average error:

$$e^r = \sqrt{\frac{1}{M \sum_{i=1}^N \sum_{k=1}^K v_{(i,k)}}} e^d \quad (2.13)$$

2.7.1 Camera Calibration Accuracy

Camera calibration accuracy is usually specified by the average reprojection error, which denotes the distance between an original image point and the one projected using the estimated camera parameters. For the calibration

performed on the described camera setup, an average reprojection error of 0.13 pixels was achieved, which compares favorably with results achieved for multi-camera calibration by other researchers. See Table 2.6 for details.

Researcher	Error e^r	Number of cameras
Pollefeys <i>et al.</i> [222]	0.11–0.26 px	25–4
Svoboda <i>et al.</i> [264]	0.2 px	16
Devarajan <i>et al.</i> [66]	0.59 px	60 (simulated)
Kurillo <i>et al.</i> [167]	0.153 px	4
Waizenegger <i>et al.</i> [288]	0.1–0.26 px	16
This system	0.13 px	40

TABLE 2.6: Comparison of the reprojection error e^r for multi-camera systems reported by different researchers. Note, that the reprojection error of 0.11 px for [222] was achieved on synthetic data rather than a real-world camera system.

Several factors contribute to the high accuracy achieved by HALCON. Firstly, using centers of circular features as control points provides a robust and accurate method for extracting them in the camera images. Then the adopted non-linear distortion models, both division and polynomial, correct the projection errors efficiently. In particular, re-estimating \mathbf{p}_{ikm} with the calibrated parameters corrects both projective and distortion bias and further improves the information extracted from the projected marks. Finally, defining the calibration as a bundle-adjustment problem yields a geometrically optimal calibration for the entire camera setup, which scales well with respect to the setup size because of the sparse LM optimization algorithm.

2.7.2 Calculation of the Floor Plane

For the majority of indoor environments, the floor can be assumed as being planar with negligible error. The *floor plane* is defined as the plane in the Cartesian world coordinate system (generated as described in Section 2.7 on page 42) which corresponds to the floor of the target area. Its calculation can be considered the final step of the system calibration process. Knowing the floor plane is useful in pedestrian detection (*cf.* Section 2.9.2 on page 51), as it allows for the elimination of 1 DOF from the target position estimation.

To determine the equation of the floor plane, which has 4 DOF, a 10×10 cm fiducial marker as depicted in Figure 2.15 on the next page is successively

exposed to the cameras in $N \geq 4$ poses at floor level.

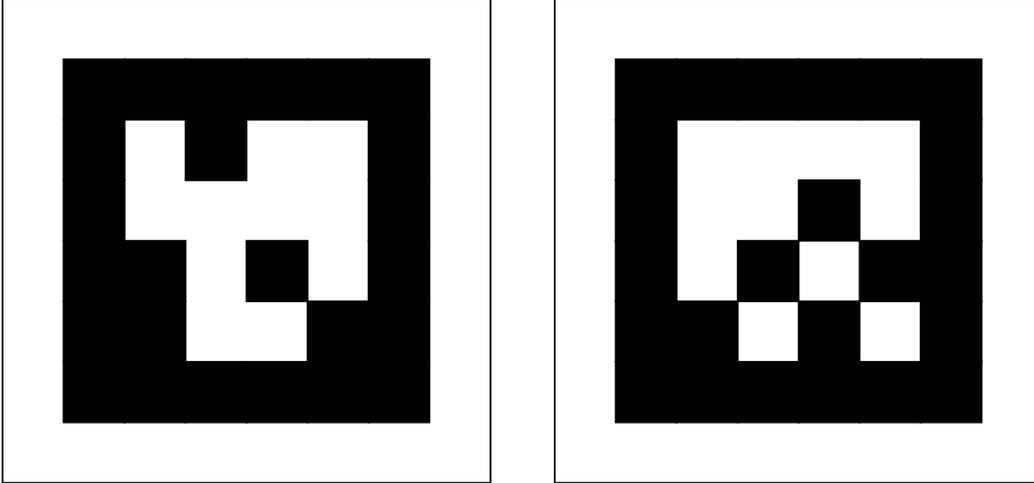


FIGURE 2.15: Examples for the fiduciary markers used to determine the equation of the floor plane. The markers display a 16 bit binary black-and-white pattern, where 4 bit are used to disambiguate the rotation and 12 bit encode the identity of the marker.

The marker is detected in the image using a state-of-the-art marker detection algorithm [82, 83], and its position in world coordinates estimated using the previously determined camera parameters $(\mathcal{J}, \mathcal{C})$. For $N > 4$, this yields an overdetermined equation system (in matrix form):

$$\mathbf{Ax} = \mathbf{b} \quad (2.14)$$

and consequently:

$$\arg \min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\| \quad (2.15)$$

which is numerically solved using the ordinary least squares approach [227] with QR decomposition [258, p. 415 ff.] to yield the planar equation for the floor level.

2.8 Preprocessing Layer Implementation

As mentioned in Section 2.4 on page 26, the preprocessing layer, also termed *service layer*, is the first of the two software layers situated on the processing

nodes. In the CCRL installation, the preprocessing layer is realized by a dedicated process for each camera, that communicates with the application layer via the KOGMO-RTDB (*cf.* Goebel [98, 99]), a real-time optimized shared memory buffer originally developed for use in cognitive automobiles. A schematic of the architecture for the preprocessing is depicted in Figure 2.16.

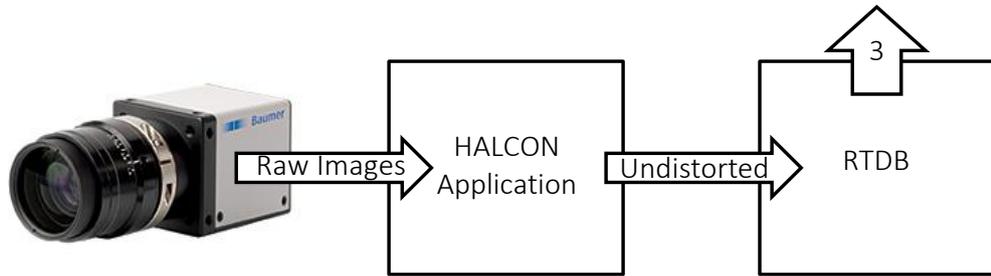


FIGURE 2.16: Schematic for the architecture of the preprocessing layer. The abbreviations refer to the KogMo-RTDB and the HALCON image processing library (*cf.* Eckstein/Steger [72]; Steger *et al.* [255]) used in the implementation.

The preprocessing layer addresses a series of three tasks, that are (a) image acquisition from the GigE cameras via the GenICam [67] interface, (b) preprocessing of the images and (c) storage of the preprocessed images in the KOGMO-RTDB. Additionally, in Phase B of the CCRL setup, it provides the functionality to store timestamped images on hard disk and replay them in real-time, which allows for simulated real-time experiments on the system.

2.8.1 Image Preprocessing

In the described implementation of the system, the single preprocessing step performed at this layer consists of image rectification, *i.e.* removal of lens distortion from the images. A divisional radial decentering distortion model (*cf.* Brown [32], or specifically Lanser [170]) is assumed.

Calculating the distortion-corrected image is a computationally expensive process. However, since the lens distortion is static, the resulting geometric image transformation

$$\text{dst}(x, y) = \text{src}(f_x(x, y), f_y(x, y)) \quad (2.16)$$

only has to be calculated once from the camera parameters, at system ini-

tialization, and can be repeatedly applied to each image. To improve the quality of the resulting image, bilinear interpolation [100, p.88] is applied to compute the pixel values for the distortion-corrected image. Finally, the camera parameters have to be adjusted, specifically $\mathfrak{D} = 0$.

2.8.2 Synchronization

To enable the synchronous replaying of pre-recorded image sequences, the system clocks for the processing nodes have to be synchronized within the order of magnitude of at least $t_o < 5$ ms, which constitutes a barely acceptable offset of $\frac{1}{6}$ frame. This task is addressed by the service layer using the network time protocol (NTP) [191] to synchronize the clocks in intervals t_i , monitoring the reported temporal offsets, and adjusting t_i until the offsets t_o are within the desired bounds. A test consisting of $N = 1579$ measurements returned a mean absolute offset of $\mu = 1.16$ ms with a standard deviation of $\sigma = 0.77$ ms, *cf.* Figure 2.17 for details.

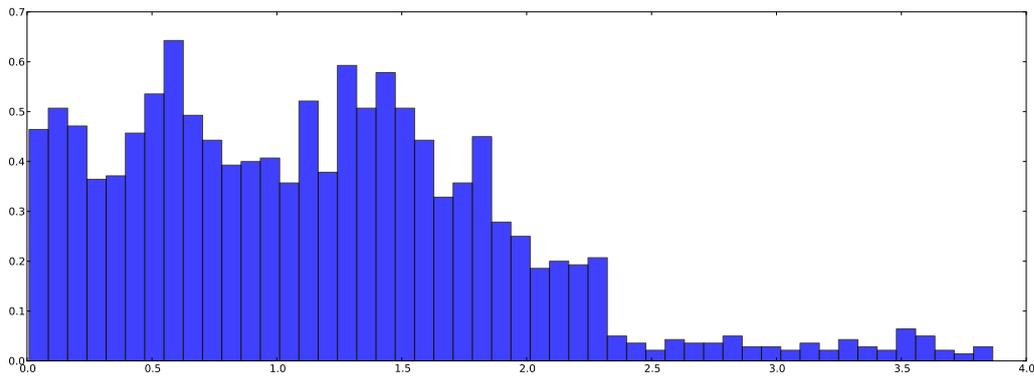


FIGURE 2.17: Histogram for the distribution of NTP offsets for the processing nodes. The x -axis denotes the offset in milliseconds.

To summarize, the service layer provides the application layer with Unix-timestamped, distortion-corrected images in red/green/blue (RGB) color format via a real-time optimized shared memory implementation, the KOGMO-RTDB.

2.8.3 Monitoring and Stability

To monitor and recover minor problems with the application layer, a watchdog timer (*cf.* Namjoo and McCluskey [200]) is employed to supervise the

processes in the application layer. If at any time one of the processes in the application layer malfunctions, the watchdog timer re-initiates the correct process and restores full functionality of the application layer.

2.9 Application Layer Implementation

The image processing layer, or *application layer*, consists of several modules distributed on processing nodes and server node, that communicate via the ICE [116]. A schematic for the communication between the modules is depicted in Figure 2.18, and the modules are described in the following section.

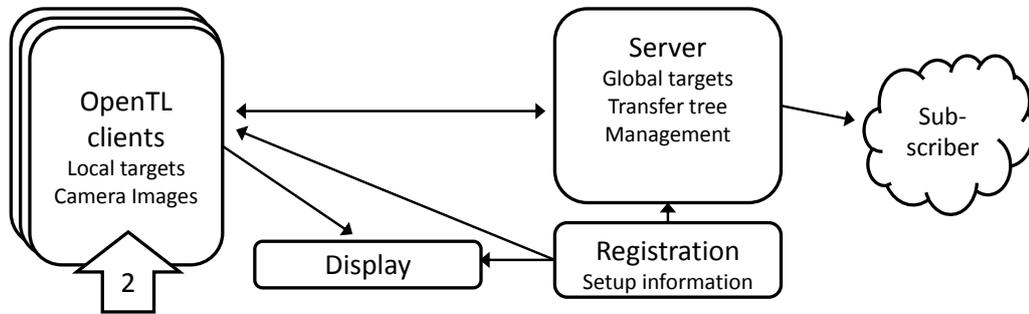


FIGURE 2.18: Schematic for the modular architecture of the application layer.

2.9.1 Distribution and Communication

The *registration module* is located on the server node. The module manages information on the system setup and camera parameters, which it provides for the tracking server, client and display modules.

The *tracking server module* is located on the server node. It receives tracking results from different client modules, manages target identities, performs tasks such as pose interpolation if necessary, assigns the correct client nodes during view transition, and transmits the finalized tracking results, composed of timestamped poses, to connected subscribers.

The *tracking client modules* are located on the processing node, operate on the camera images provided by the service layer, and perform the local pedestrian detection and tracking. The distinct steps of these tasks are explained in detail in the following sections. The system provides one dedicated tracking client module for each camera.

Finally, the *display module* provides human-legible output for the pedestrian tracker by displaying the FOV of all connected cameras and additional debugging output, such as the positions of currently tracked targets and estimated track history. An example for the display output is depicted in Figure 2.25 on page 62.

2.9.2 Person Detection

As stated in Section 2.3.3 on page 17, the tracking approach of Panin [211] necessitates a successful detection of the target's position for the initialization of the tracker. The goal of the detection step is to determine whether a new potential target has entered the sensor FOV, and to determine its approximate initial position in world coordinates. For the implementation described in this thesis, a linear three-step approach to pedestrian detection is applied: foreground-background segmentation, blob clustering, and target association. All three steps are considered as single-view problems, and will be examined in more detail in the following.

2.9.2.1 Foreground-Background Segmentation

For the segmentation of the foreground, two different background subtraction strategies were evaluated: static background [105], and dynamic background models using GMMs [223, 254]. In accordance with Piccardi [217], the latter was found to be more reliable with regard to variations in brightness effected by shadows and changes in illumination, and is consequently employed thenceforth.

To eliminate possible noise artifacts in the segmentation, and to separate neighboring blobs, morphological opening [249] is applied to yield the foreground map. Subsequently, as a safeguard against duplicate detection, existing targets are re-projected into the image plane, and their area subtracted from the foreground map using a simple elliptical shape model for the approximation.

The result of this step is a binary map of the image foreground relevant for new target detection. Figure 2.19 on the following page depicts a sample result for the applied foreground segmentation.

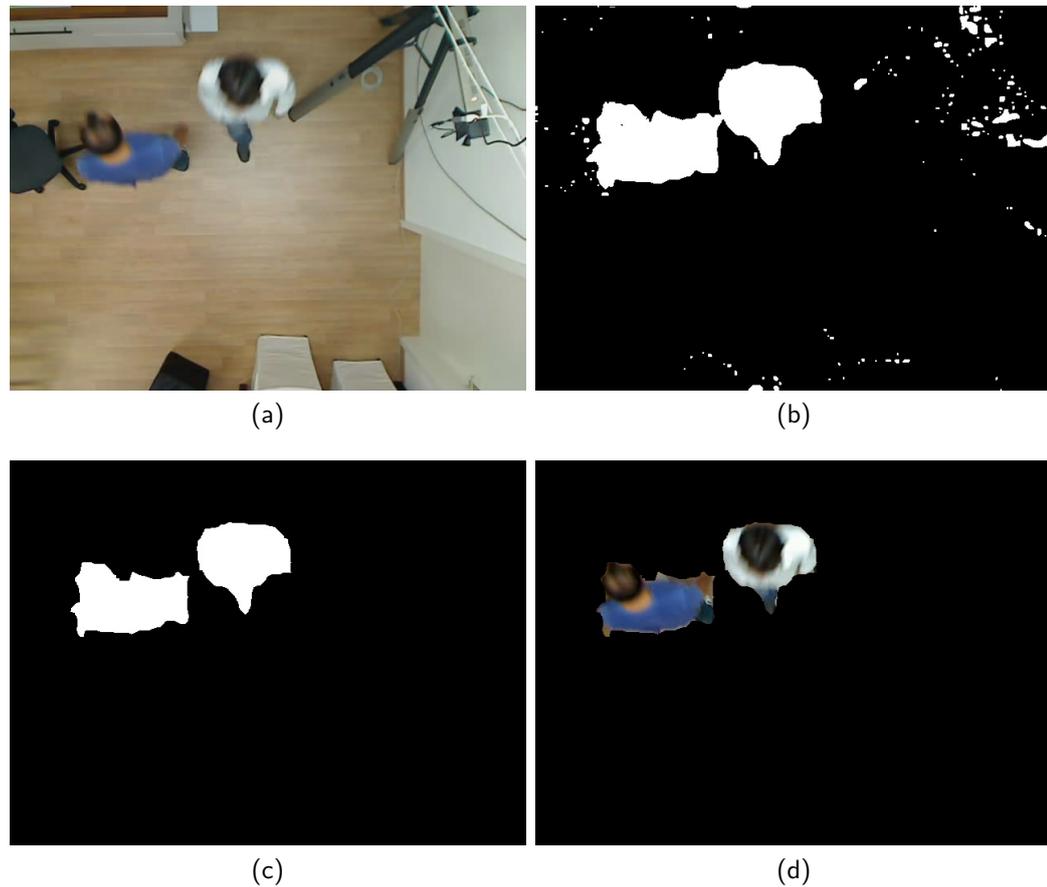


FIGURE 2.19: Illustrating the different steps of the pedestrian detection process. In (a), the input image is shown, depicting two pedestrians observed within a camera FOV from above. In (b), foreground segmentation (as described in Section 2.9.2.1 on page 51) has been performed, but the resulting binary image still shows some noise and undesirable coherence between the two blobs. In (c), morphological operations have been applied to remove the noise and separate the two blobs. In (d), the original image is depicted with the foreground image as a mask, which provides an impression of the regions used for color histogram collection in the model initialization step (*cf.* Section 2.9.2.3 on the next page)

2.9.2.2 Blob Clustering and Association

In the next step, the foreground map resulting from the segmentation is used to calculate blobs, as proposed by Rocha *et al.* [234] using Hu Moments [125]. The mass center of each blob is used as a rough initial 2D position in the image coordinate system. A rule-based binary classifier is applied to separate valid blobs representing targets from invalid blobs, with the parameters for this classifier being the blobs' area and aspect ratio of their outer dimensions. This process can result in detecting multiple targets simultaneously.

Subsequently, this initial 2D pose is used to approximate the 3D translational pose of each valid target in world coordinates. The 3D translation pose can be computed using the extrinsic camera parameters \mathcal{E} , casting a virtual ray given by the focal point and the computed mass center of a given blob, and intersecting it with the floor plane, which was computed during the calibration step (*cf.* Section 2.7.2 on page 46). Minor uncertainties arise from the fact that information about person height is unavailable at this point, therefore an average height of 1.7 m (*cf.* Section 3.5.4 on page 95) is used. This does not cause any problems for the purposes of detection though, since the goal of this step is to find a rough initial position which will later be refined during tracking, and since the variation in height for humans is comparatively small.

Since every blob that results from a suitable foreground region is classified as a target, the described approach is restricted to human moving targets only, in order to avoid erroneous detection caused by false-positive classifications. Extension to different types of targets would necessitate adjustment of the survival criteria applied to blobs, as well as sufficient difference in size and shape to human targets.

To summarize, the clustering and association step results in a list of valid targets and their approximate 3D position in world coordinates.

2.9.2.3 Model Initialization for Tracking

Although the previous step was already sufficient to address the problem of pedestrian detection, in order for the tracker to be properly initialized, an appearance model has to be generated for each newly detected target.

Pedestrian shape is modeled by a rigid 3D cylinder in the real-world coordinate system, whose height and width approximately correspond to average human measurements (*cf.* Section 3.5.4 on page 95). A statistical color model is obtained by collecting the image pixels for the re-projection of the shape

of the respective target into image coordinates. To improve the accuracy of the histogram collection, the background map computed in the segmentation step is used as a mask in this step.

To increase robustness versus variances in lighting and shadows cast by other pedestrians or objects in the target area, the image is transformed into HSI color space for the histogram collection process. Since intensity values reflect the illumination conditions, different bin sizes were used for each channel to increase the relative importance of the color attributes, hue and saturation. For the final implementation, bin sizes of 16 bins each for hue and saturation and 8 bins for intensity (*i.e.*, 2048 bins in total) were used.

Summarily, the model initialization step results in a target-specific appearance model for all valid targets, consisting of a histogram in HSI color space, on top of the 3D positions extracted in the previous step.

2.9.2.4 Performance of the Detection Process

To evaluate the performance of the detection process, an experiment was conducted by measuring the detection accuracy (*i.e.* percentage of successful detections). A set of $N = 360$ detection tests was performed on single images taken from the cameras (as described in Section 2.6.1 on page 32) using the hardware described in Section 2.6.3 on page 41 for processing, *i.e.* matching live conditions of the system at run-time. Table 2.7 lists the results obtained, while Figure 2.20 on the next page depicts some sample images used for the experiment.

N_p	Detection Rate	\bar{t}	N_i
0	N/A	21.9 ms	50
1	1	38.0 ms	200
2	0.95	50.4 ms	50
3	0.938	66.1 ms	60

TABLE 2.7: Results for the evaluation of the pedestrian detection process. Where applicable, mean and standard deviation for the respective values are indicated. N_p denotes the number of targets, *i.e.* pedestrians, while N_i denotes the number of images of the respective type and \bar{t} denotes the average processing time recorded for the entire detection step.

The results of the detection performance test indicate that the detection

process constitutes the most computationally demanding part of the entire tracking pipeline. Furthermore, detection time increases approximately linearly with the number of targets, while the accuracy deteriorates, mostly due to aggregation of targets causing problems with blob separation. As a consequence, a tracking-by-detection approach is not real-time feasible in this scenario, especially with regard to multiple targets. Therefore, “routine” detection (*i.e.* to detect new targets entering the FOV) is performed in reasonable intervals. A pedestrian moving at an average speed of $\approx 1.34 \frac{\text{m}}{\text{s}}$ (*cf.* Daamen and Hoogendorn [61]) traverses the full FOV in $h_o = 1.7 \text{ m}$ in $\approx 1.5 \text{ s}$, which equates to ≈ 42 frames. An interval of 0.4 s or ≈ 10 frames is selected for the detection, which means that in the worst case a target is still detected if it traverses only a quarter of the FOV.



FIGURE 2.20: Sample images for the detection performance experiment, depicting one, two, and three targets respectively. Note, that the field of view gets comparatively crowded for three targets, and it becomes difficult to avoid each other’s personal space of $r \approx 0.46 \text{ m}$ (*cf.* Hall [107, 108]). Therefore, simultaneous appearance of more than two targets within a single FOV is considered unlikely for the sensor configuration described here.

2.9.3 Person Tracking

With the target successfully detected, the next challenge is to track its movement across multiple camera FOVs. As stated before in Section 2.3.5 on page 19, the multi-view person tracking process is divided into alternating phases of single-view person tracking and view transition. For the single-view tracking step, a MCMC filter is used in combination with statistical color models as described in Section 2.9.2.3 on page 53, whereas for the view transition problem, two different approaches were investigated. Both steps are detailed out in the following.

2.9.3.1 Multi-target MCMC filter

The task of the single-view tracker is to continuously update the world position of each target until it leaves the camera FOV, while avoiding the confusion of multiple simultaneous targets. Initially, each target is considered separately.

Tracking operates on a pre-defined set of DOF. Since the targets are assumed to walk on the previously determined floor plane, and since a rigid 3D shape model is used, each target t possesses 2 DOF in the x and y translation on the floor plane. Therefore, the state vector of the i -th target is given by $s^i = (t_x^i, t_y^i)$.

For the tracking process, each target's appearance model has to be matched against histograms collected from the current camera image at the re-projected predicted target positions in the image. The procedure for histogram collection is consistent with the procedure used for appearance model generation, and is described in Section 2.9.2.3 on page 53. Since a calibrated camera model is being used, perspective effects are taken into account when computing the target silhouette in the camera image. Because of the relative distance between the camera and the person being comparable with the depth extension of the target (*i.e.* the person's height), these effects have a comparatively high impact for the described implementation. Therefore, they cannot be neglected, especially for people in the peripheral view field. The impact of perspective effects on histogram collection merits further inspection, and is covered at length in Chapter 3 on page 79.



FIGURE 2.21: Illustrating the monocular MCMC tracker. Here, the rigid cylindrical shape model is visible, which is used as a coarse but sufficiently approximated representation of the human body, as it is observed from a supracranial perspective.

In order to estimate the state for each of the n targets within the camera FOV, a Bayesian Monte-Carlo tracking approach is implemented, as proposed by Panin *et al.* [212]. This methodology employs a particle filter to maintain the global system state, $s = (s^1, \dots, s^n)$ by means of a set of hypotheses s_h , the eponymous *particles*. In the described implementation, particles are updated in successive frames using MCMC sampling.

In particular, the Markov chain generation proceeds by iterating two steps, that equate to the *Metropolis-Hastings* algorithm [45], for each particle $n = 1, \dots, N$:

1. *Prediction*: propose a new state s'_t from the previous one s_t^{n-1} by sampling from a proposal density $Q(s'_t | s_t^{n-1})$
2. *Correction*: Compute the *acceptance ratio*

$$a = \frac{P(s'_t | Z^t) Q(s'_t | s_t^{n-1})}{P(s_t^{n-1} | Z^t) Q(s_t^{n-1} | s'_t)} \quad (2.17)$$

3. If $a \geq 1$, accept the proposed state $s_t^n \leftarrow s'_t$. Otherwise, accept it with probability a ; in case of rejection, the old state is kept $s_t^n \leftarrow s_t^{n-1}$

The proposal distribution Q can be arbitrary to some extent, and for this purpose the dynamical model itself is chosen $P(s_t | s_{t-1})$. Since this model is symmetric, the second ratio in Equation 2.17 is canceled out. The efficiency of the MCMC formulation is due to the fact that only a single randomly chosen target i is updated at a time, and the resulting consequence that the proposal ratio $P(s_{i,t} | s_{i,t-1})$ has to be calculated for this target only. Under the assumption of independent measurements for each target, the two likelihoods $P(z_t | s'_t)$ and $P(z_t | s_t^{n-1})$ differ only for a single target as well. Figure 2.22 on the following page provides a graphical overview of the hypothesis update step.

The first b samples of this chain are the so-called *burn-in* samples, obtained before the Markov chain reaches its steady state, and will be discarded from the sample. This set is usually a small percentage of the overall sample set.

2.9.3.2 View Transition

The view transition problem can be broken down into three sub-problems. The decisions that have to be taken are as follows:

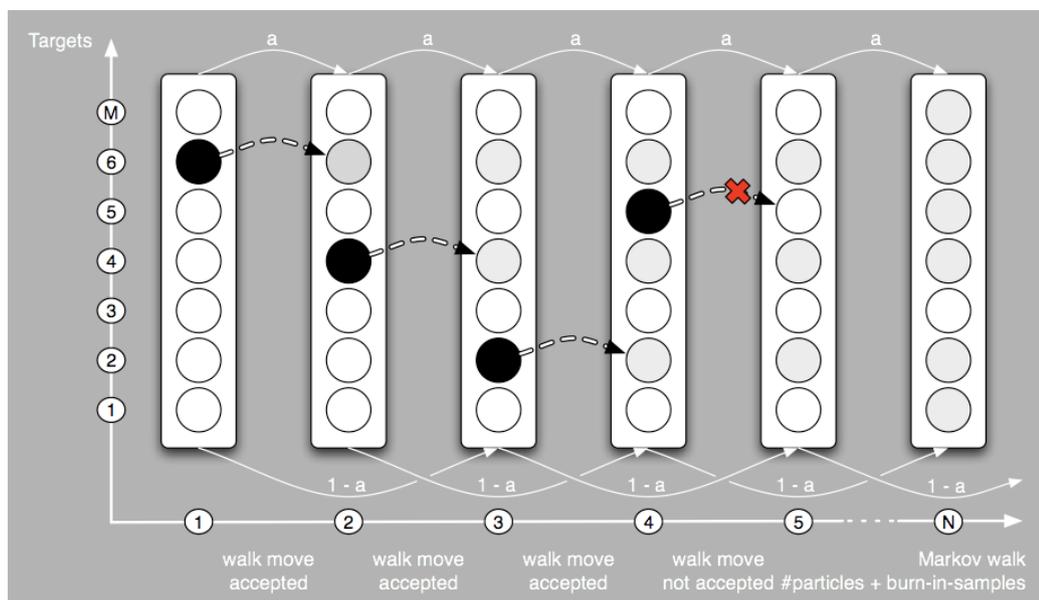


FIGURE 2.22: Hypotheses update for the multi-target MCMC tracker, according to [211]. This illustrates the serial nature of the MCMC update (walk) for multiple targets, where each hypothesis is tested during a single step pertains to only a single target. Each walk-move is accepted with probability a , as described by Equation 2.17.

- (a) when to transfer a target to another view, *i.e.* when a target leaves the current view
- (b) where to transfer the target, *i.e.* which view is adjacent and suitable to continue the tracking
- (c) what to transfer, *i.e.* which information is used to instantiate the target in the tracker for the other view

Initially during the system implementation, the view transition problem was addressed by a *target transfer tree*, which is generated from the camera parameters and handles the decision *where* to transfer a target leaving the view, in combination with the *transition areas*, which address the problem of *when* a target will leave the view, and therefore has to be transferred. Both of those concepts are detailed out below, and unlike the solution to the third sub-problem, were not varied throughout the development of the system.

Target Transfer Tree For a set of 40 camera tracking modules operating at 28 Hz, constant intercomparison of all camera positions to evaluate the transfer decision is a waste of processing power, given the fact that the camera positions are known to be static. To speed up the evaluation of the decision when and where to transfer a target, a transfer tree is generated from the evaluation of the camera parameters at system start.

The transfer tree is a quad tree, which is obtained by projection of the camera centers and FOV on the floor plane (*cf.* Section 2.7.2 on page 46), and successive division of the floor plane into quarters, until only a single camera remains per node. This transfer tree can be efficiently evaluated at each position update, and returns the decision if and where to transfer the target. Figure 2.23 on the following page depicts an exemplary target transfer tree for a section of the CCRL installation.

Transition Areas Transition areas are a direct result of the transfer tree explained in the previous section, and are derived by a space discrete test of points in the floor plane against the transfer tree at system initialization. They are not strictly necessary for successful operation of the tracking, but can be depicted in the display module to enhance legibility of the system operation. Figure 2.24 on the next page depicts these transition areas overlaid in the FOV of two adjacent cameras in the CCRL installation.

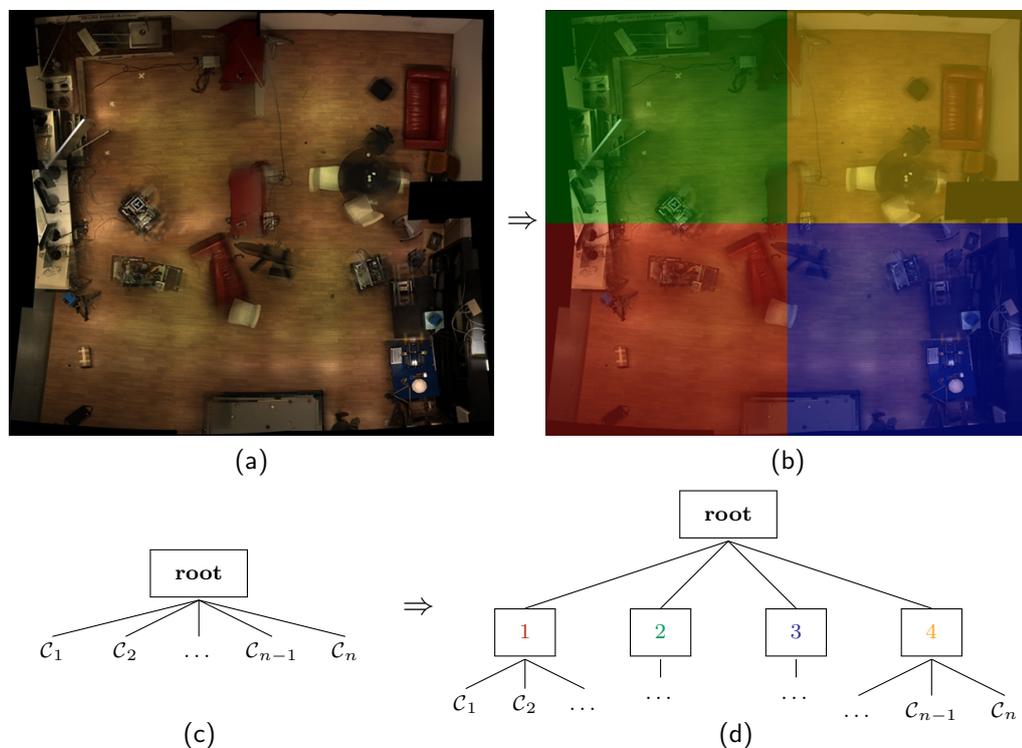


FIGURE 2.23: Scheme of the transfer tree generation scheme for the CCRL camera installation. The area is successively divided (a) \Rightarrow (b), and the depicted node insertion step (c) \Rightarrow (d) repeated until each leaf node contains only a single camera.

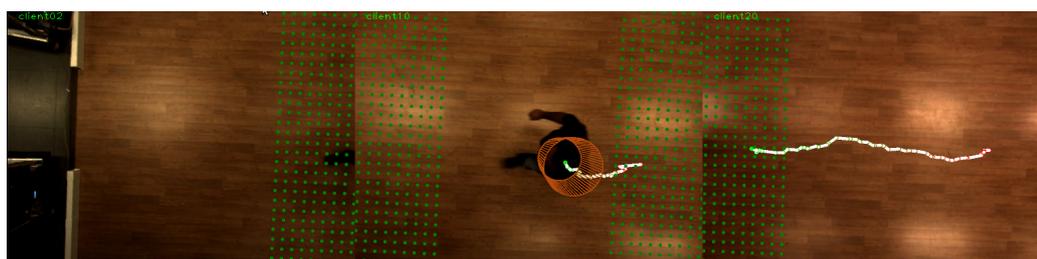


FIGURE 2.24: Transition areas overlaid in the FOVs of a camera in the CCRL installation. The dots represent valid transfer points between cameras, resulting from a grid test with 0.05 m intervals on the floor plane against the target transfer tree. Additionally, the track of a pedestrian target during view transition is depicted, illustrating the point of transition.

Transmitted Data Regarding the problem of *what* to transfer, it has to be kept in mind that in the described implementation, processing for different views occurs on different machines and in real-time. Therefore, the amount of data required to be transferred between target origin and destination to solve the view transition problem has to be kept in check. This becomes a factor, for example, when considering approaches which would require entire images to be transferred during this step.

For the solution to this part of the view transition challenge, three different approaches were implemented and qualitatively evaluated:

Approach I: Only the information that a new target has entered the view is transferred to the camera node supposed to take over the tracking, as determined by the transfer tree. This signal initiated a local detection process (*cf.* Section 2.9.2 on page 51), in which the target was detected in the corresponding view and a new model generated for the initialization of the local tracker (*cf.* Section 2.9.3 on page 55). However, due to the comparatively high computational cost of the detection process (*cf.* Table 2.7 on page 54), this procedure proves to be inconvenient for real-time processing, as it introduces undesirable delays in the system.

Approach II: In addition to the steps involved in Approach 1, a fast frame differencing [51] is performed in the view for the new camera node to narrow down the region of interest (ROI) for the detection process. This serves to speed up the entire transition process by addressing its most time-intensive step.

Approach III: The entire target information available, *i.e.* world position, position history and color model, is sent to the next responsible camera node. Complete multi-camera calibration with a common coordinate system (*cf.* Section 2.7 on page 42) greatly facilitates the view transition problem, since the pose information can be used to instantiate the tracking module of that camera node directly in combination with the color model, eliminating the need for a further detection process and in consequence significantly speeding up the transition process.

In concordance with the expectations, Approach III yielded the most desirable results regarding speed, since it does not require the time-consuming detection process to be repeated. As lack of spatio-temporal consistency has a significant negative impact on the accuracy of the Bayesian tracking approach, the effect of the speed-up in the transition process is advantageous enough to decide in favor of this design approach. For an impression of the

tracking of a target across multiple cameras, Figure 2.25 provides a glimpse of the human-readable output produced by the tracking system.

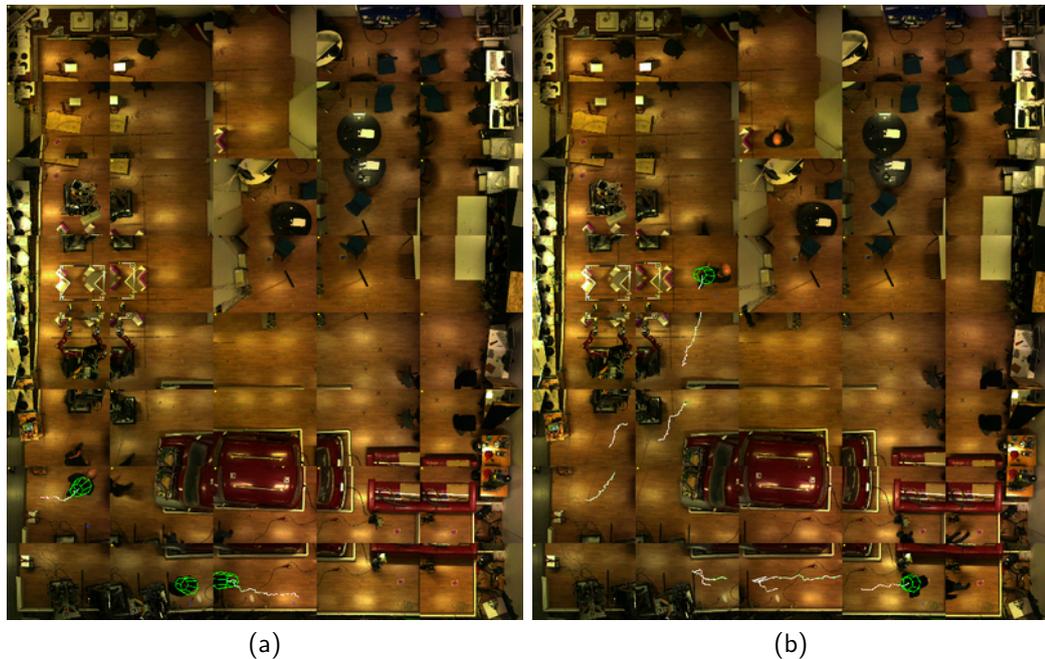


FIGURE 2.25: Illustrating the tracking at the CCRL installation. Here, the human-readable output for the display unit is depicted, with views for all 40 cameras shown in a grid. Identified targets and estimated tracks are overlaid. From left to right, a target transitioning between camera views (a), and tracks after several transitions can be seen in the images (b).

2.10 Experimental Evaluation

Several aspects of the real-time multi view person tracking system presented in this chapter were evaluated experimentally to verify the system's functionality.

While the experiments relating to the technical features (*e.g.* stability, update rate) of the system were conducted on the live system, due to economic reasons it was decided to use a prerecorded data set to test the algorithmic parts of the system rather than testing *in situ* in real-time. Since, under most circumstances, algorithmic properties can be more easily varied, this procedure ensures a higher degree of comparability between subsequent test runs

of the system by keeping the variances low. It is worth mentioning though, that repeated experiments were not entirely deterministic, due to the probabilistic nature of the SMC algorithm used for tracking. As an additional benefit of this method, the recorded data is preserved, and can be used in the future to test enhancements made to the current algorithms under the most similar conditions possible.

2.10.1 Recording the Data set

For the recording of the data set, a section of the area covered by the system was deemed sufficient to prove algorithmic concepts, which allowed for reduction of the image data required for storage. The selected section measures approximately 5×7 m and is covered by 16 cameras (*cf.* Figures Figure 2.27 on page 65 and Figure 2.28 on page 66).

The data set for the evaluation of the multi-view tracker consists of three types of short video sequences depicting clothed pedestrians. The first type shows a single pedestrian crossing the area, moving between the FOVs of several different cameras. The second type of sequence depicts two pedestrians crossing the area simultaneously in opposite directions, with their trajectories approaching each other near the center. Finally, the third type of sequence has two pedestrians crossing the area simultaneously in the same direction, which their trajectories being approximately parallel.

Additionally, image sequences for the evaluation of the pedestrian detection (*cf.* Section 2.9.2.4 on page 54) and the single-view part of the tracking (*cf.* Section 2.10.2) were recorded, which are described in the respective sections of this chapter. Table 2.8 on the next page provides further details about the recorded sequences.

Table 2.9 on the following page provides details about the subjects participating in the experiment.

2.10.2 Single-View Tracking Accuracy

For the evaluation of single-view tracking accuracy, two image sequences with circular trajectories were recorded, to allow for continuous evaluation through looping of the image data. Two different circular shapes were recorded for the trajectories: ellipsoid and lemniscatoid (*cf.* Figure 2.29 on page 66), both with the center of the trajectory situated approximately on the camera principal axis.

N_p	Action	Test case	N_V	N_i	N_s
1	walking	tracking	≥ 2	≈ 600 (≈ 21 s)	12
2	walking, opposite direction	tracking	≥ 2	≈ 600 (≈ 21 s)	12
2	walking, same direction	tracking	≥ 2	≈ 600 (≈ 21 s)	12
1	walking, ellipse	tracking	1	3000 (≈ 107 s)	3
1	walking, lemniscate	tracking	1	3000 (≈ 107 s)	2
1	standing/walking	detection	1	10 ($\approx \frac{1}{3}$ s)	20
2	standing/walking	detection	1	10 ($\approx \frac{1}{3}$ s)	5
3	standing/walking	detection	1	10 ($\approx \frac{1}{3}$ s)	6

TABLE 2.8: Statistics for the recorded image sequences in the evaluation data set. N_p denotes the number of pedestrians, N_v the number of views (multiple-view or single-view), N_i the number of images per sequence, and N_s the number of sequences of the respective type in the data set.

Participant	Height	Color, torso	Color, legs	Color, hair
\mathcal{P}_1	1.83 m	black	dark blue	bald(ish)
\mathcal{P}_2	1.74 m	red	dark blue	black
\mathcal{P}_3	1.72 m	light green	blue	brown

TABLE 2.9: Statistics for the participants recorded in the evaluation image sequences. Note, that the colors stated in the table correspond to the consensus of visual inspection by the investigator and the participants.

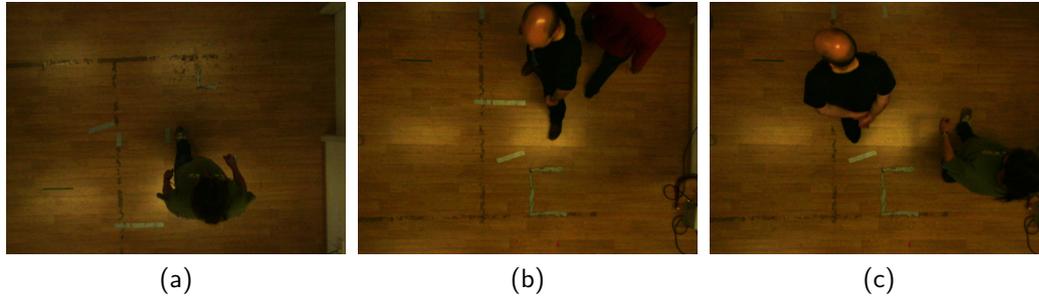


FIGURE 2.26: Example pictures from the image data collected for the evaluation of the multi-view tracking system. From left to right: Single pedestrian crossing (a), two pedestrians in opposite directions (b), and two pedestrians with concordant direction (c).



FIGURE 2.27: Impression of the experimental area, used for recording of the evaluation data set. Start and end positions for the paths taken by the participants were marked with small tape markings on the floor. The flooring consists of parquet, and has a comparatively high reflectance with a light reflectance value (LRV) of ≈ 30 . Some specular reflections are clearly visible.

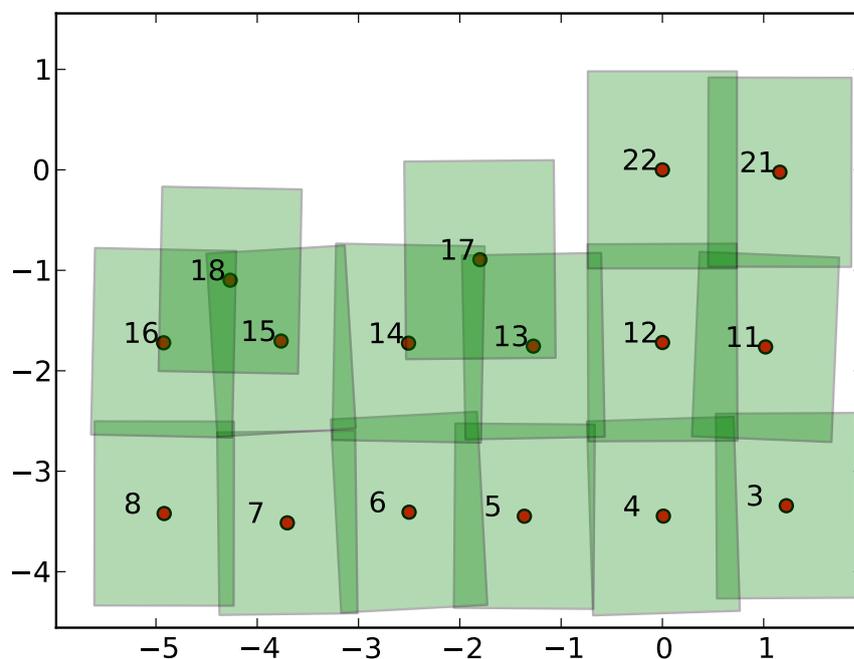


FIGURE 2.28: Schematic of the area used for the recordings, showing the camera positions and fields of view at $h_o = 1.7$ m.



FIGURE 2.29: Schematic trajectories for the two sequences recorded for monocular pedestrian tracking evaluation, overlaid on camera images. The first trajectory (a) resembles an ellipse, while the second trajectory (b) resembles a Bernoulian lemniscate [169] (*i.e.* eight-shape).

Ground truth data for these sequences was annotated manually in the form of target positions in image coordinate space. This type of annotation was chosen since the goal of the experiment was to measure the performance of the tracker, which operates with hypotheses on the image coordinate level. Furthermore, to provide a fair evaluation of the tracking process, the result of the detection process (*cf.* Section 2.9.2 on page 51) was used as ground truth position in the initial frame, and the correspondence point manually annotated in subsequent frames. This procedure helps to alleviate additional errors in measurement possibly introduced by disparities between manual ground truth annotation and detection process.

For evaluative purposes, the advantage of the lemniscatoid trajectory compared to the ellipsoid trajectory is that while the perspective of the camera towards the pedestrian remains approximately constant for the circular trajectory, it undergoes almost the full range of perspectives possible under the setup constraints for the lemniscatoid path. Although not particularly relevant for the evaluation of the tracking accuracy at this point, this fact is of particular interest with regard to the work described in the subsequent chapter, since the same image sequences are also used to evaluate the adjustments made to the appearance model to compensate for perspective effects.

2.10.2.1 Error metric

To measure the performance of the single-view tracking, a suitable metric for the tracking error has to be defined. To define a meaningful metric, the conditions under which the experiment is conducted have to be considered. For example, Maggio and Cavallaro [181] use the density of true positive pixels as a metric for the accuracy of tracking objects with highly variable distance from the camera, *e.g.* objects approaching the camera. While such a metric makes sense in their case, it would be unnecessarily complicated in the case of the experiment described here, since the distance between camera and object remains within the same order of magnitude. Similarly, a varying image size would require normalization of the error by image size to produce meaningful results, *et cetera*.

For the experiment discussed here, the Euclidean distance between the ground truth target position \mathbf{t}_g and the target candidate \mathbf{t}_c in the image plane provides the basis for a suitable error metric, and constitutes the *absolute tracking error*:

$$d_a = |\mathbf{t}_g - \mathbf{t}_c| \quad (2.18)$$

However, the disadvantage of this error metric is the fact that it takes into account neither the size of the image (n_x, n_y) nor the size of the target. Therefore, the error metric is adjusted by the average size of the target (diameter $2\bar{r}$ of the of the circumcircle of the blob) in pixels, which accounts for both the size of the image and the size of the target, to provide the *relative tracking error*:

$$d_{\bar{r}} = \frac{d_a}{2\bar{r}} \quad (2.19)$$



FIGURE 2.30: Illustrating the error metrics used to evaluate single-view tracking performance. (a) depicts the absolute tracking error d_a (yellow), measuring the distance between ground truth position (red) and candidate position (blue). (b) depicts the current circumcircle of the target blob (turquoise), used to calculate the relative tracking error d_r .

2.10.2.2 Results

Table 2.10 on the next page provides mean and standard deviation of $d_{\bar{r}}$ for both the ellipsoid and the lemniscatoid trajectory, measured over looped sequences with a normalized length of $N_i = 1000$ images. As the results indicate, the accuracy of the tracking is slightly better for the lemniscatoid path.

To provide an explanation for this observation, it has to be taken into account that for the camera configuration in the experiment, where the orientation of the camera coincides with the normal of the floor plane, the position of the center of a tracked target is less ambiguous when the target is located

near the principal axis of the camera. This is because the center of gravity of the silhouette of the target, the reprojection of the center of gravity of the target in world space, and the reprojection of the standpoint of the target (the average of the position where both feet touch the floor plane) all coincide at this point, whereas they diverge increasingly with greater distance from the center.

The average distance of the target from the principal axis, on the other hand, is larger for the ellipsoid path, where the target moves near the borders of the camera’s FOV, while for the lemniscatoid path, the target repeatedly crosses the center of the FOV.

Sequence	Trajectory	$\mu(d_{\bar{r}})$	$\sigma(d_{\bar{r}})$	N_i
\mathfrak{S}_{42}	ellipsoid	0.237	0.09	1000
\mathfrak{S}_{43}	ellipsoid	0.210	0.08	1000
\mathfrak{S}_{44}	ellipsoid	0.242	0.08	1000
$\sum_{i=44}^{42} \mathfrak{S}_i$	ellipsoid	0.229	0.08	3000
\mathfrak{S}_{45}	lemniscatoid	0.186	0.08	1000
\mathfrak{S}_{46}	lemniscatoid	0.165	0.06	1000
$\sum_{i=46}^{45} \mathfrak{S}_i$	lemniscatoid	0.176	0.07	2000
$\sum_{i=46}^{42} \mathfrak{S}_i$	lemn./ellipt.	0.208	0.08	5000

TABLE 2.10: Accuracy evaluation for the MCMC pedestrian tracker for single-view tracking, mean μ and standard deviation σ for the relative tracking error $d_{\bar{r}}$ (cf. Equation 2.19). N_i denotes the number of video frames for which the accuracy was measured.

2.10.2.3 Adjusted Error Metric and Results

To compensate for the effect of precision gradient in the FOV inherent to the setup as explained in the previous paragraph, a second measure for the relative tracking error is introduced, where the diameter $2r$ of the circumcircle of the target blob the current position is used to compute the error, instead of the mean diameter $2\bar{r}$ calculated from the entire track:

$$d_r = \frac{d_a}{2r} \quad (2.20)$$

Sequence	Trajectory	$\mu(d_r)$	$\sigma(d_r)$	N_i
\mathfrak{S}_{42}	ellipsoid	0.228	0.09	1000
\mathfrak{S}_{43}	ellipsoid	0.206	0.08	1000
\mathfrak{S}_{44}	ellipsoid	0.229	0.07	1000
$\sum_{i=44}^{42} \mathfrak{S}_i$	ellipsoid	0.221	0.08	3000
\mathfrak{S}_{45}	lemniscatoid	0.214	0.09	1000
\mathfrak{S}_{46}	lemniscatoid	0.201	0.08	1000
$\sum_{i=46}^{45} \mathfrak{S}_i$	lemniscatoid	0.208	0.09	2000
$\sum_{i=46}^{42} \mathfrak{S}_i$	lemn./ellipt.	0.216	0.08	5000

TABLE 2.11: Accuracy evaluation for the MCMC pedestrian tracker for single-view tracking, mean μ and standard deviation σ for the relative tracking error d_r (cf. Equation 2.19). N_i denotes the number of video frames for which the accuracy was measured.

Table 2.11 depicts the updated results, using the new error metric for the relative tracking error with the above equation.

In comparison to the preliminary results from Table 2.10 on page 69, the mean error $\mu(d_r)$ for the lemniscatoid paths increases significantly, from 0.176 to 0.208, while the effect on the error for the ellipsoid path is hardly affected by the change in metric.

To summarize, the updated metric for the relative tracking error leads to significantly closer values for $\mu(d_r)$ on the lemniscatoid and ellipsoid paths, and appears to have successfully leveled the effect of the precision gradient induced by the perspective variation in the FOV, as theorized above.

2.10.3 Multi-View Tracking Performance

As the multi-view tracking method employed differs from the single-view tracking method only in the added transition process, the accuracy of the track is not expected to differ significantly. Therefore, the evaluation of the multi-view tracking is not focused on the exact accuracy of the track, but rather application-centered, that is with the suitability of the result for application in *e.g.* human-robot interaction in mind. Consequently, the evaluation of the multi-view tracking performance looks at the overall performance of

the system, with the question in mind if the tracking manages to yield a target position within a tolerable error range.

2.10.3.1 Performance metric

To measure the performance of the multi-view tracking as a whole, the success of tracking a target in a sequence of images is defined as the target candidate \mathbf{t}_c being within d_m of the ground truth \mathbf{t}_g in the final frame \mathcal{I}_n of the sequence.

$$|\mathbf{t}_g - \mathbf{t}_c| \leq d_m \quad (2.21)$$

In accordance with the results from single view tracking, the maximum tolerable error distance d_m was set at $2 \cdot \mu(d_a)$ *i.e.* ≈ 136 px for the experiment.

In theory, this definition of a tracking success implies that the target can be lost and recovered during the tracking process, which is a less restrictive definition than requiring $d_a \leq d_m \quad \forall \mathcal{I}_i$. However, it is considerably easier to test, as annotation, which is especially time-intensive for multi-view sequences, is only required for the final frame of the sequence.

In addition, manual inspection of the experimental data indicates that:

$$P(R) \ll P(L) \ll P(\bar{L}) \quad (2.22)$$

where R indicates the event that a target was lost and recovered, L indicates the event that a target was lost, and consequently \bar{L} indicates the complementary event that a target was never lost during tracking. All events refer to the tracking of the target over the whole sequence. This observation serves as a further argument in favor of the proposed performance metric.

With success and failure of a tracking a single target in an image sequence defined, the success rate

$$R = \frac{S}{N} \quad (2.23)$$

is used in the following to measure the performance of tracking multiple targets over multiple sequences, where S denotes the number of successfully tracked targets, whereas N denotes the total number of targets.

Targets	Relative Direction	N	S	R
1	N/A	12	12	1.00
2	$\approx \updownarrow$ (antiparallel)	24	23	0.96
2	$\approx \upuparrows$ (parallel)	24	20	0.83
< 3	all of the above	60	55	0.92

TABLE 2.12: Performance evaluation for the MCMC pedestrian tracker for multi-view tracking. N denotes the total number of targets in all sequences, whereas S denotes the number of targets tracked successfully and R denotes the success rate.

2.10.3.2 Results

2.10.4 Target Identity Maintenance and Recovery

For the evaluation of the capacity of the appearance model to maintain the identity of a target when the track is lost (*e.g.* when a pedestrian leaves the target area entirely and later returns), a sequence of images where several pedestrians repeatedly cross the FOV of a single camera, while varying direction and speed, was synthesized from the image data recorded (*cf.* Table 2.8 on page 64).

The procedure for the identity recovery experiment is displayed in Figure 2.31 on the next page. Essentially, it can be considered a very basic machine learning experiment, or precisely, a multi-class classification experiment with $M = 4$ classes, where the four classes result from the number of three participants, with one additional class constituted by previously unseen targets. Accordingly, the process consists of two distinct steps, *training step* and *test step*, which are detailed out in the following paragraph.

As a prerequisite to test the identity recovery properties of the appearance model, the appearance models for all potential targets must first be acquired once. This is done in the *training step*, on separate image sequences for each target (more precisely, the image sequences already used in Section 2.10.2 on page 63), and the resulting models are referred to as *reference models* in the following. Subsequently, in the *test step*, these appearance models are compared against the *candidate models*, *i.e.* the appearance models obtained from targets detected in the test sequence. Depending on the result of the comparison, the candidates are classified either as one of the previously ac-

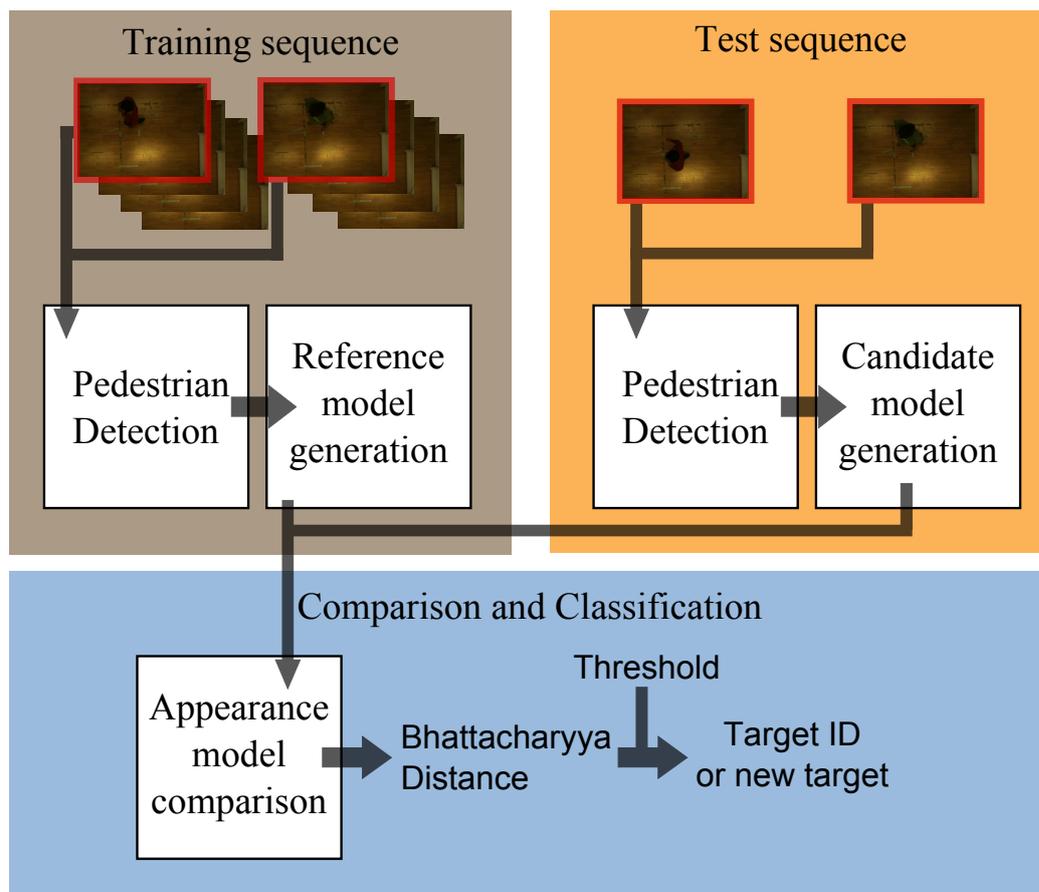


FIGURE 2.31: Scheme for the identity recovery experiment, depicting the sequence of processing steps for the experiment conducted on the respective image sequences.

quired targets, or as a new target, whose model does not match any of the reference models acquired during the training step.

To solve the classification task, a similarity metric for the appearance models is required. Here, the same method is employed as during the testing of hypotheses during the tracking step (*cf.* Section 2.9.3 on page 55). Since the appearance models constitute discrete color distributions, the Bhattacharyya distance D_B (*cf.* Kailath [142]; based on Bhattacharyya [21]) is employed, as a state-of-the-art measurement of the similarity between two distributions:

$$D_B(p, q) = -\ln \left(\sum_{x \in X} \sqrt{p(x)q(x)} \right) \quad (2.24)$$

where p and q denote the reference model and candidate model (*i.e.* color histograms), respectively, and X denotes the k, m, n -bin partition of the HSI color space.

2.10.4.1 Performance metric

The multi-class classification problem described above is considered as a series of binary classification problems for the purpose of the evaluation: Each target is classified either correctly or incorrectly. Consequently, the recall rate of successfully identified targets:

$$f_r(C) = \frac{f_a(C)}{N}, \quad (2.25)$$

where $f_a(C)$ denotes the number of true positives and N denotes the total number of targets, provides an adequate metric for the performance of the target recognition. Note, however, that this metric does not discriminate between confusion of two targets from the test set and the confusion of a target from the test set with a new target. Here, it is implicitly assumed that both types of errors are equally undesirable.

2.10.4.2 Results

Table 2.13 on the facing page provides the results of the evaluation. It is apparent that the results leave some room for improvement. Some of this can be attributed to the fact that the reference models for the recognition are generated from a single initial detection, which does lead to a low robustness

of the initial approach. To anticipate, this is one of the deficiencies of the approach that is addressed by the improvements suggested in the subsequent chapter.

Participant	N	$f_a(C)$	$f_r(C)$
P_1	8	7	0.88
P_2	8	6	0.75
P_3	8	8	1.00
$P_{1\dots 3}$	24	21	0.88

TABLE 2.13: Results for the evaluation of target identity management and recovery. N denotes the total number of targets, and $f_r(C)$ denotes the recall rate of successfully re-identified targets. $P_{1\dots 3}$ denote the participants, as listed in Table 2.9.

2.10.5 System Uptime and Robustness

Since from a design perspective, assembling a robust and highly available system was a critical concern, an experiment was conducted to measure the continuous uptime t^u of several system components, in order to gain a measure for system availability.

Data was collected using the Xymon system monitor [259] on the updated setup (Phase B) of the CCRL installation (*cf.* Section 2.5 on page 29) during a period of 14 days, by sampling the state of various system components in 30 s intervals.

Hardware components investigated for long-term availability were the server computer and the connected processing clients, as well as the Ethernet-connected cameras. Regarding the software layers, both the implementation of the preprocessing layer (*cf.* Section 2.8 on page 47) and the implementation of the person tracking module from the application layer (*cf.* Section 2.9 on page 50) were investigated.

Table 2.14 on the following page lists the inherent availability V of the important system components over the course of the experiment period with:

$$V_i = \frac{t_i^u}{t_i^d + t_i^u} \quad (2.26)$$

for each component, where t_i^u signifies the uptime of a system component while t_i^d signifies the downtime.

For the downtimes experienced during the experiment, no manual intervention or repair was required. All outages could be recovered by monitoring the software modules and automatically initiating a restart for any module that became unavailable.

Component	Server	Client	Camera	Preprocessing	Tracking
V	1	1	0.99996	1	0.99994
t^d/d , avg.	0 s	0 s	3.46 s	0 s	5.19 s

TABLE 2.14: Availability and average downtimes per day of selected system components, hard- and software, as recorded during a continuous two-week system test.

The results indicate, that while the two-week test period allows for results regarding the availability of camera and tracking modules, it proves too short to provide any insights into the availability of the used server and processing nodes. These results are to be expected considering the fact that server and processing nodes consist of OTS hardware components. According to several studies from Schroeder *et al.* [243–245], the two components most likely to cause outages in architectures comparable to that of the server and processing nodes are hard disks and memory, with probability for failure within the order of magnitude of 0.033 annualized failure rate (AFR) for hard disks [243] and 0.0022 AFR for memory modules [245]. For a period of two weeks, 164 memory modules and 246 hard disk drives (HDDs), the expected probability of at least one failure occurring during the experiment therefore was within the region of ≈ 0.0139 (memory) and ≈ 0.312 (hard disk). However, the comparatively high value for HDD failure is based on the assumption of almost continuous HDD access, as it occurs in web servers. For the proposed system, such a high frequency of HDD access would only occur if image data was recorded indiscriminately, which was not the case during the experiment.

Note, that the test period of two weeks exceeds the usual demands on the system for continued surveillance of human-robot experiments, by a factor of at least 20. Since

$$V \approx \prod_{i=1}^N V_i \quad (2.27)$$

for N components with independent availability, the total availability V of the system can be stated as at $V \approx 0.99990$, or, to put it in different terms, within the order of magnitude of “four nines” (regarding the terminology, *cf.* Bottomley [27]).

2.11 Summary and Discussion

In this chapter, a vertically integrated system using multiple static cameras to track multiple pedestrians across a target area in real-time was presented. Evaluation results demonstrate that the system is capable of addressing this task with an update rate and accuracy suitable for applications in HRI (*cf.* Section 2.10.2.2 on page 68) over extended periods of time (*cf.* Section 2.10.5 on page 75).

Among the different steps performed for the successful implementation of the system, the multi-camera calibration deserves some special note, as it exceeds the state of the art regarding accuracy, measured by the reprojection error, as evidenced by the comparison with results from other multi-camera calibration routines (*cf.* Section 2.7.1 on page 45).

The modeling of pedestrians for tracking using histograms in HSI color space, which is commonly used for pedestrian tracking from planar views, *i.e.* posterior, lateral and anterior perspectives, proves to be an effective method from a supracranial perspective as well. However, for conditions comparable to those presented for the experimental area described in this chapter, where the depth extension (*i.e.* height) of the targets is within the same order of magnitude as the distance between target and sensor, perspective effects on the color distribution exhibited by pedestrians merit a closer inspection. This issue is addressed in the subsequent chapter.

Chapter 3

Appearance Modeling

As mentioned in Section 2.3 on page 13, the multi-view tracking task can be broken down into a single detection task, and a subsequent alternating series of single-view tracking tasks and view transition tasks. Basic solutions for the challenges presented by those tasks have been fleshed out in the previous chapter (Sections 2.9.2 to 2.9.3.2).

A common feature linking single-view tracking and view transition tasks is their dependence on how the appearance of the tracking targets is being modeled. In the following chapter, a closer look is taken at appearance modeling for pedestrian tracking, and the static color distribution approach for appearance modeling presented in the aforementioned sections is improved upon.

3.1 Problem Statement

The color model initially used in tracking humans on the CCRL camera setup, as explained in Section 2.9.2.3 on page 53, can be categorized as a *static appearance model*. Pedestrian appearance is modeled as a static histogram in HSI color space, which is acquired at person detection. Static appearance constitutes a standard approach for modeling the appearance of pedestrians in tracking and identification tasks (*cf.* Bird *et al.* [23], Bazzani *et al.* [15]).

In most cases these tasks are performed within images depicting pedestrians from a predominantly lateral perspective, where the face of the pedestrian towards the camera varies mostly from turning around the vertical axis. Since most articles of clothing display similar statistical properties (*e.g.* regarding their color or texture) from all directions, variations in pedestrian appearance

originating from rotation around the vertical axis do not cause complications with the statistical color modeling approach.



FIGURE 3.1: The same person standing at different points within the camera field of view. Note the differences in the area ratio of visible colored parts, especially clothing.

For a multi-view setup, this observation holds true as well, and static appearance modeling produces favorable results as long as the perspective remains predominantly lateral. From supracranial perspectives, however, variation in perspective has a larger effect on the appearance of pedestrians, especially if the distance between pedestrian and camera is comparatively short. Accordingly, the perspective variation of view transitions has a greater effect on appearance in such setups.

Consider, for example, the person depicted in Figure 3.1. If the person is standing directly below one camera, the visible face, in a geometric sense of the term, consists mainly of the top of the head and shoulders of the person. Due to self-occlusion, certain clothing surfaces (*e.g.* trousers, shoes) are more likely to become occluded than others. Should the person move towards the edge of the camera's FOV, however, torso and legs gradually become more visible. In the most likely case, that the colors of head, torso and legs of the person are different from each other, this will also cause the color distribution of the observed person as a whole to change.

As a consequence, it would be desirable to describe the appearance of a clothed pedestrian not as a static model, but with an adaptive approach, to be able to more closely reflect the changes occurring during those perspective shifts and in turn improve the robustness and accuracy during tracking and re-identification tasks.

In short, the problem covered in this chapter can be summarized as the design of an appearance modeling approach for the tracking of pedestrians with the following qualities:

- (1) The appearance should be modeled adaptively, so that changes in camera perspective towards a pedestrian can be taken into account during single-view tracking.
- (2) The model should be applicable to multi-view tracking, and consider the resulting view transition task specifically.
- (3) The approach should be sufficiently generic to be applicable to objects other than pedestrians with the appropriate modifications, *e.g.* considering their geometric properties in the approach.

3.2 Outline of this Chapter

The remainder of the chapter is divided into the following sections:

Section 3.3 discusses related work on appearance modeling, statistical properties of color and histogram collection with regard to the work presented in this chapter.

Section 3.4 on page 85 explains the rough idea of the solution for the problem stated in the previous section.

Section 3.5 on page 90 includes the exact specifications of the proposed appearance model, and explains its integration into the tracking, detection and reacquisition tasks, as described in the previous chapter.

Section 3.7 on page 108 evaluates the proposed appearance model regarding its performance in tracking and detection tasks.

Section 3.8 on page 116 includes a comparison of the evaluation against the results achieved with the basic appearance model (*cf.* Section 2.10 on page 62), further discussion, and outlook.

3.3 Related Work

Since related work on most of the tracking pipeline has already been discussed in Section 2.3 on page 13, this section will focus solely on the topics of appearance modeling, statistical color descriptors, and their respective applications in tracking.

3.3.1 Statistical Color Descriptors

The use of statistics to describe the color of objects in digital images reaches back more than twenty years, where seminal work was conducted by Swain and Ballard [265, 266], who first report the use of color histograms for object indexing and identification.

A variety of methods has been proposed to model the color distribution of images or image ROIs. The straightforward method consists of determining the frequencies of pixel values for each channel within a certain color space (*e.g.* RGB, luminance/chrominance (YUV), HSI), dividing them into n bins, usually equally sized, and combining the statistics for each channel into a single 3D color histogram. Variations include the omission of channels, *e.g.* modeling hue and saturation while discarding intensity (*cf.* Sebastian *et al.* [246]).

Statistical color models have been applied to a wide range of challenges in computer vision, that can be roughly divided into three larger categories – detection tasks, classification tasks, and indexing – and a smaller number of miscellaneous applications.

To provide some examples regarding detection tasks, statistical color modeling has been applied to skin color detection (*cf.* Jones and Rehg [139], Lee and Yoo [172]), which is an important sub-task in many face detection or face image analysis approaches, fire detection (*cf.* Celik *et al.* [43], Cho *et al.* [46]), with application in surveillance systems for building security, and many different approaches to object detection (*cf.* Kim *et al.* [156], Cucchiara *et al.* [58], Mason and Duric [184], Utsumi *et al.* [279], Okuma *et al.* [209], and Juang *et al.* [140])

Classification and identification tasks – as a binary subtype of the former – are commonly solved using a machine learning technique, such as support vector machines (SVMs), in combination with statistical features of an image or image region to generate models of the objects or classes of interest. Among similar applications, statistical color modeling has been employed to provide features for the classification of plants (*cf.* Burks *et al.* [35]) and recognition of household objects (*cf.* Gevers and Smeulders [93]). Furthermore, similar techniques involving color modeling have been used to distinguish synthetic images from photographic images (*cf.* Chen *et al.* [44]), and to distinguish face images from images depicting nudity (*cf.* Duan *et al.* [71]).

A diverse set of further applications of miscellaneous type has been reported. For instance, Tian [271] reports an application for automatic focus window selection in digital cameras, by employing skin color segmentation with sta-

tistical color models to calculate focus ROI. Varma and Zisserman [281] apply statistical color models to texture classification, using frequency histograms of textons to assemble texton dictionaries used in the classification, whereas Fine *et al.* [84] describe a biomimetic approach, where statistical color models are employed in surface segmentation.

To summarize, statistical color descriptors have been used in computer vision in a great variety of forms and applications for over a decade. Statistical color modeling is a proven and well-described state-of-the-art technique for the description of surfaces. Therefore, it is employed in this thesis as the starting point of the search for viewpoint-aware adaptations for clothed pedestrians, which display a set of different surfaces that can easily be distinguished by human visual perception.

3.3.2 Appearance Modeling

In the terminology of this thesis, an appearance model is any underlying hypothesis about an objects qualities, such as brightness, color, or geometry, that results in assumptions about the object's representation in an image. Mathematical or statistical properties of this representation, called features, are used to link model and representation.

Appearance modeling is relevant to several related challenges in surveillance applications, namely detection, recognition, reacquisition and tracking. Since models are used to describe specific objects, models of similar structure describing different objects can be referred to as classes of models. Over time, such classes have been proposed in great variety.

The first group of appearance models do not account for any geometric features of the modeled object, relying instead solely on properties of the object's surface, like color or brightness. Common examples for such appearance models include models based on color histograms [265].

Appearance modeling with color histograms has been proposed in various nuances. Gray *et al.* [102] evaluate different approaches regarding their performance on specific data sets. Color histograms have been used to model appearance for a broad variety of objects, for example in skin detection by Jones *et al.* [139] or head tracking by Birchfield *et al.* [22]. For tracking persons, specifically, improvements can be made by adapting the model during operation, a technique used *e.g.* by Nummiaro *et al.* [207] in combination with a particle filter.

Regarding the advantageousness of different color spaces with regard to track-

ing, Sebastian *et al.* [246] investigate the effect of statistical modeling in several color spaces (grayscale, RGB, luminance/chrominance (YCbCr), and HSI) on tracking robustness. They conclude, that the highest performance is consistently achieved for HSI color space, and that performance can be increased further by disregarding the intensity value altogether, *i.e.* modeling properties of hue and saturation exclusively.

Appearance models based on the HOG approach were originally introduced by Dalal and Triggs [62] to detect humans in varying poses, but have also been generalized to accommodate different objects, *e.g.* by Felzenswalb *et al.* [78]. To circumvent the comparatively large computational complexity of the approach, Geismann and Schneider [92] combine HOG with Haar-like features (*cf.* Viola and Jones [284] for the original features, Lienhardt and Maydt [176] for an extended feature set) into a two-step detector, where Haar-like features are used to identify pedestrian candidates while the HOG features are used to verify those in the second step. However, since the HOG approach considers the orientation of gradients, the results vary with changes in observation perspective (especially from lateral/anterior perspectives to supracranial perspectives, *cf.* B.8). Therefore, HOG do not constitute a particularly advantageous set of features for the purpose of modeling pedestrians under the conditions presented here.

As a further sophistication, the tracked object's geometric and topological properties can also be taken into consideration, through the use of a so-called *shape model*. Shape models come in different varieties, of which the *rigid shape model* is the most basic. For example, Lanz *et al.* [171] transition from considering only surface properties to including shape properties in pedestrian tracking by using a simplified human shape model, with different color distributions for different body parts. In contrast to the approach proposed in this chapter, however, they track each body part separately and merge the tracking results, as opposed to merging the color distributions and tracking the person as a monolithic entity.

Another super-class of models are geometrically deformable shape models. These models have a high number of degrees of freedom, and during operation their deformation parameters have to be optimized to fit the model to the image. Active Shape Models [55, 56] and Active Appearance Models [54] constitute two well-known examples for these classes of models. Originally introduced to be used to model faces and facial expressions, they have also been extended to other applications, particularly in the medical field [17, 192].

While these models are specifically suited to model complex, deformable

objects, one of their drawbacks is the complexity of the model fitting and parameter adjustment process, which can render their application difficult when low response times are required, *e.g.* for real-time tracking. On the other hand, similar approaches are especially suited if not only the object itself but its full body pose or activity state has to be detected. For example, Bandouch *et al.* [12] use a deformable geometric model with 51 degrees of freedom to track and analyze human motions. Another example for such use of geometrically deformable models, applied to the problem of facial expression recognition, is provided by Mayer [185].

Summarily, it is striking that the topic of appearance modeling is mostly researched on in combination with research on its applications like object tracking, detection and recognition. There are relatively few publications that explicitly and exclusively consider appearance modeling, as a separate discipline. Consequently, there is a dearth of research infrastructure on the topic, as evidenced by the lack of image databases and accepted evaluation metrics specifically published with appearance modeling in mind. This is a factor which complicates efforts to conduct substantiated performance comparisons of different modeling techniques.

3.4 Solution Idea

An object consisting of surfaces that exhibit different appearance properties (*e.g.* color, texture) changes its appearance toward the viewer significantly in different perspectives. Examples are provided in Figure 3.2 on the following page. Therefore, an adaptive appearance modeling approach is used to model these changes and provide an increase in modeling accuracy compared to static appearance modeling. To that end, the appearance of the object of interest is considered as a combination of models of the appearance of its parts.

In the following, the solution idea is sketched out twice. The first part aims to be a general description of the concept behind the solution idea, and is put in abstract terms, without restriction to specific classes of appearance models or objects. The second part of the description, on the other hand, constitutes a concretization of the ideas sketched out in the first part, with specific application to the task of pedestrian tracking in different perspectives.

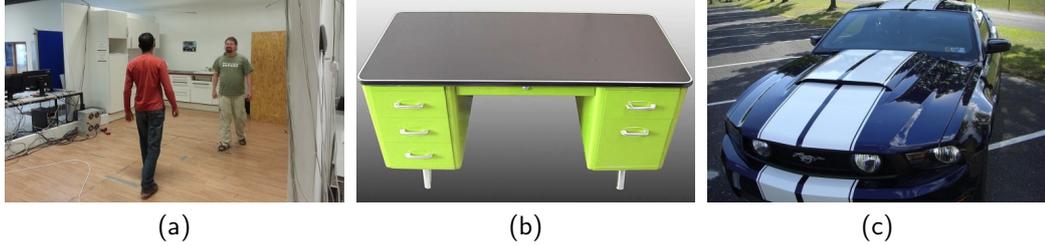


FIGURE 3.2: Illustrating the solution idea. Depicted are several objects with vertical appearance patterns, for which the described technique can be applied. From left to right, clothed pedestrians (a), a colored table with dark tabletop (b) and a car with “beauty stripes” on top.

3.4.1 Abstract Solution Idea

The realization of the solution idea requires two steps to be performed manually in preparation, along with knowledge about the geometry and surface areas with similar appearance of the object of interest. Although the exact appearance (*e.g.* color, texture) of these surface areas is unknown, it is known that the areas are relatively uniform with respect to appearance (*e.g.* torso clothing in a pedestrian, colored stripes on a car, or a tabletop).

Firstly, the visual appearance \mathcal{A} of a reflective object is considered as a weighted sum of the appearance of its parts, where the *weights* w are determined by the size of the respective parts in the camera projection. The descriptors to model the appearance are selected accordingly, so that addition of models and multiplication of models with scalars are defined (*e.g.* histograms, Gaussian distributions). Consequently, the *appearance model* \mathcal{A} of the object can be expressed as a linear combination of *partial appearance models* \mathcal{A}^p :

$$\mathcal{A} = \sum w_i \mathcal{A}^p \quad (3.1)$$

In a second step, the shape of the object of interest is considered. A coarse, rigid, three-dimensional model of its surface, the *shape model* is constructed of simple shapes (called *shape atoms* in the following) that can be described geometrically, *e.g.* circles or regular polygons. Subsequently, N different *shape model parts* are defined, that correspond to the partial appearance models. The model parts are defined in such a way that they are expected to be relatively uniform in appearance, and each shape atom is assigned to a model part.

Consider the situation where a target has just been detected and is to be tracked in subsequent images. Its current world position is known, as well as its appearance model \mathcal{A}^0 for the current perspective, which has been created at detection. At this point, the partial appearance models are unknown. The intermediate goal is to obtain the partial appearance models, which will subsequently allow for refinement of the composite appearance model with regard to perspective.

The initial period of tracking constitutes the *burn-in phase* for the appearance model. During that phase, the processing step for a single image is as follows: Firstly, the shape atoms are projected into the image plane using the camera projection Π . The self-occlusion of the shape model, *i.e.* the occlusion of shape atoms by others, is considered in this step. From the area of the projected shape atoms on the image plane, and the assignment of shape atoms to shape model parts conducted in the preparation step, the ratio of the areas P_i^p of each projected model part to the area P of the projected model is obtained. This ratio provides the weights of the partial appearance models for the current perspective:

$$w_i = \frac{P_i^p}{P} \quad \forall i \in \{1, \dots, N\} \quad (3.2)$$

In the following, this entire procedure is referred to as *weight computation*. Finally, weights and appearance model are stored, and processing moves on to the next image. During the burn-in phase, the initial appearance model \mathcal{A}^0 is used to test the tracking hypotheses (*cf.* 2.9.3)

The exact number of images required for the burn-in period depends on the number of model parts. After a sufficient number of K images with $K > N$ has been processed, the weights and appearance models constitute an overdetermined system of K linear equations of the form:

$$\mathcal{A}_k = \sum_{i=1}^N w_{(i,k)} \mathcal{A}_{(i,k)}^p \quad \forall i \in \{1, \dots, N\} \wedge k \in \{1, \dots, K\} \quad (3.3)$$

This marks the start of the *operational phase* for the model adaptation. The system of linear equation is now solved numerically, using an appropriate technique such as ordinary least squares [227], yielding the partial appearance models \mathcal{A}^p for the shape model parts. This procedure is termed the *model decomposition* step. From this point on, the appearance model used to test hypotheses is generated for each image depending on the perspective, by combining the partial appearance models with the weights that are

computed for the respective hypothesis using the method described in the previous paragraph, which is termed the *model composition* step. A sliding window of the K most recent weights and appearance models for the last K images continues to be stored, and the partial appearance models \mathcal{A}^p are re-computed in regular intervals.

3.4.2 Application to Pedestrian Tracking

For the concretization of the approach, a suitable modeling technique has to be selected, which allows for the computation of partial appearance models \mathcal{A}^p from weights w and appearance model \mathcal{A}^c . For pedestrians wearing different items of clothing which are distinguishable in color, statistical color modeling is selected as a good fit. The color distributions C are represented as color histograms H in HSI color space. Histograms allow for addition, subtraction and multiplication with scalar values, and consequently fit the necessary criteria for model composition and decomposition. The properties of color histograms and normalized color histograms are discussed in detail in Section 3.5.2 on page 92.

As the selected tracking approach does not provide orientation information (*cf.* 2.9.3) for the targets, the shape of the tracking target (*i.e.* the pedestrian) is modeled from a combination of frusta and cylinders, which share the quality of cylindrical symmetry, *i.e.* they are invariant to rotation around their central axis. The frusta and cylinders constitute the model parts: head, torso and legs. The exact sizes of the model parts are derived from average measurements for the corresponding body parts stated in literature. Subsequently, these frusta and cylinders are approximated with a mesh of triangles in the *model refinement* step, which is done to exploit the facility of the projection of triangles in the weight computation step. The model refinement is described in Section 3.5.5 on page 99.

A graphical overview of the approach is depicted in Figure 3.3 on the next page.

3.4.3 Assumptions and Constraints

Some assumptions are made about the targets to further specify the conditions under which the proposed method is expected to operate. The method described here is designed with the tracking of pedestrians in mind, that is to say humans with their pose restricted to being upright. To extend the method to targets of different shape, *e.g.* cars or certain animals, a different

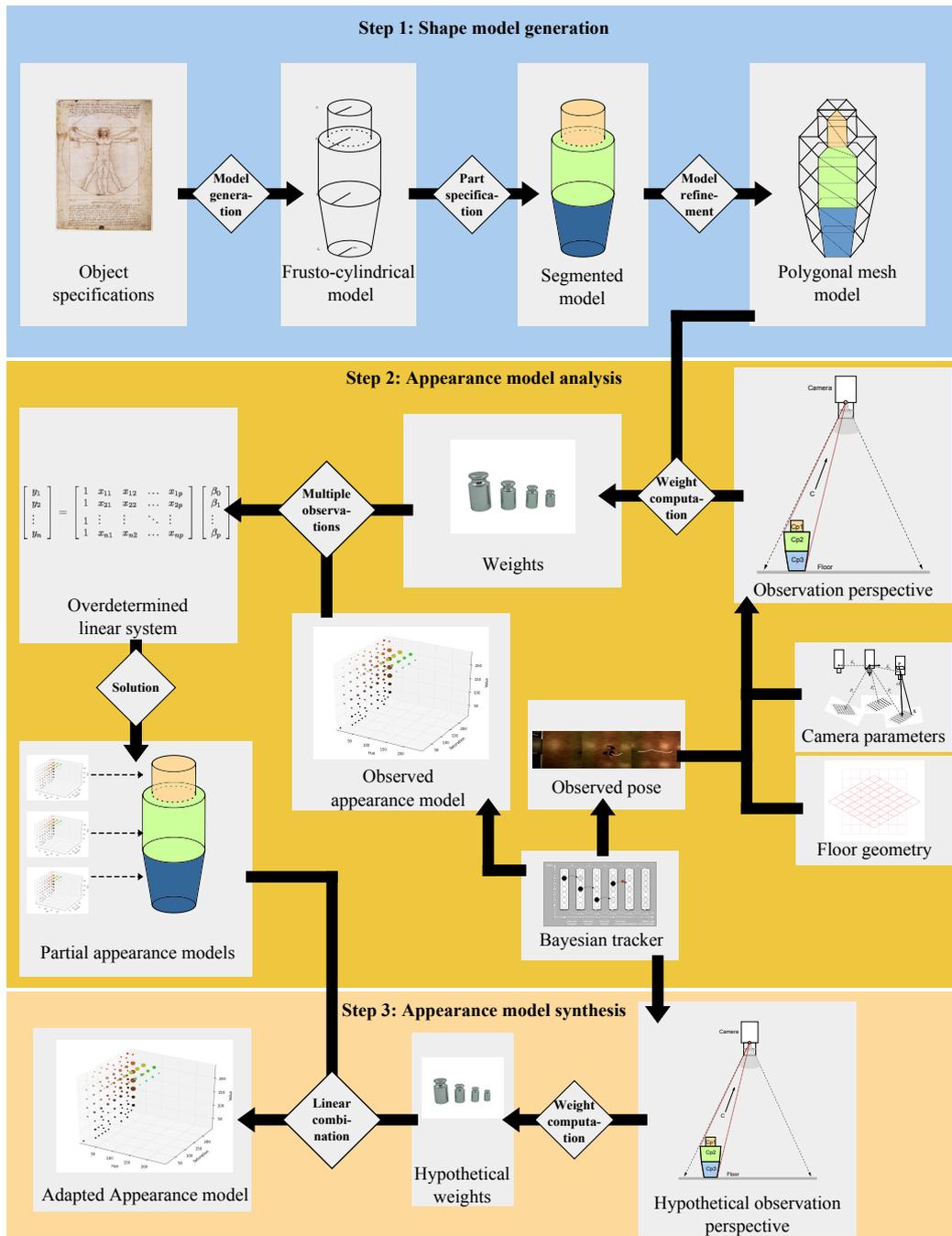


FIGURE 3.3: Illustrating the solution idea. Schematic presentation of the adaptive appearance modeling approach, exemplarily integrated into a Bayesian tracking framework for pedestrian tracking.

geometric model (*cf.* Section 3.5.3 on page 94) would have to be manually designed matching the proportions of the intended target.

As a further constraint, it is assumed that the targets' overall outward appearance does not change significantly over the course of the tracking. Foremost, this means that the targets do not change their clothing during the operation of the tracking. Furthermore, it is assumed that the clothing on the relevant body parts is relatively uniform in color and does not look completely different from varying angles, *e.g.* light blue in the front and dark red in the back.

For the expected body proportions of the tracked humans, proportions exhibited by adult persons without any health-related modifications, such as published by Nakanishi *et al.* [199], are assumed for the general case. In cases where body proportions vary greatly from these numbers, for example due to certain diseases like achondroplasia [120, 231] (the most common cause for dwarfism), the method described in the following would have to be applied with an a priori customized model, with adjusted proportions (*cf.* Section 3.5.4 on page 95).

3.5 Appearance Model

As far as the pedestrian tracking approach considered here is concerned, a target t is represented by its translation $\mathbf{T}_t^W \in \mathbb{R}^3$, which is defined as the translation of the model origin $\mathbf{O}^m \in \mathbb{R}^3$ from the world origin $\mathbf{O}^w \in \mathbb{R}^3$,

$$\mathbf{T}_t^W \equiv \mathbf{O}^m - \mathbf{O}^w \quad (3.4)$$

and the appearance model \mathcal{A} , which represents properties of the part of its surface that is visible from the camera. As an additional constraint, the translation \mathbf{T}_t^W is restricted to a two-dimensional translation of the target on the floor plane, since only upright (*i.e.* standing or walking) targets are considered.

$$t \equiv (\mathbf{T}_t^W, \mathcal{A}) \quad (3.5)$$

For the purpose of this thesis, only statistical color properties are considered for the appearance of t :

$$\mathcal{A} \equiv C \quad (3.6)$$

where C is a statistical model of the target's color which satisfies the criteria required of \mathcal{A} (addition and multiplication with scalars defined, *cf.* Section 3.4.1 on page 86). Consequently, the target is represented by the dyad of translation and color:

$$t = (\mathbf{T}_t^W, C) \quad (3.7)$$

3.5.1 Combined Color Distribution

The normalized color distribution C of the image of a surface can be expressed as a linear combination of N weighted color distributions C_i^p for a partition of the surface into N sub-surfaces:

$$C = \sum_{i=1}^N w_i C_i^p \quad (3.8)$$

where the C_i^p signify the normalized color distributions of the images of the surface parts.

The general assumption is that pedestrians wear differently colored articles of clothing, with the most significant color differences usually being between clothing worn on legs and upper body, since these constitute the largest visible surfaces. In most cases, the third-largest significant surface area can be assumed to be the head, although this attribution may vary depending on hair length and style. In accordance with these considerations, a tri-partition of the pedestrian surface into *head*, *torso* and *legs* is employed:

$$C = w_1 C^h + w_2 C^t + w_3 C^l \quad (3.9)$$

where C^h , C^t and C^l constitute the color distributions of the respective body parts, and w_i constitute the respective weights, signifying their relative area in the image of the pedestrian surface.

From a modeling point of view, this approach presents two challenges. Firstly, an appropriate method has to be selected to model the color distributions C in digital images. This can be done via the color histogram approach as proposed by Sural *et al.* [262]. The color distribution C is modeled as a three-dimensional histogram in HSI color space. The approach is detailed out in Section 3.5.2 on the next page.

Secondly, a method is needed to obtain the image area ratio for arbitrary

observation perspectives, *i.e.* combinations of target translation \mathbf{T}_t^W , target vertical orientation as represented by the normal of the floor plane \mathbf{F} and camera pose \mathcal{E} . This step requires modeling of the geometric properties of a pedestrian, also referred to as shape. It results in a two-fold benefit, as the ratios (also called *weights* in the following) allow for estimation of the color distributions C^p of the parts from the overall distribution C , as well as for an approximation of C from C^h , C^t and C^l for hypothesis generation (*cf.* Section 3.6.1 on page 102). To determine the size of the area of each model part in the image, and consequently the weights for the normalized color distributions, a rigid pedestrian shape model consisting of three body parts is designed, which is explained in detail in Section 3.5.3 on page 94.

3.5.2 Color Histograms

For the calculations in the subsequent sections, *color histograms*, as proposed by Sural *et al.* [262], are used to model the color distribution within a region \mathcal{R} of a digital image in a certain color space.

$$C \equiv \mathbf{H} \quad (3.10)$$

In the following, a three-channel color space (*e.g.* RGB, HSI) is assumed, where pixel values v are $\in [0, 1]$ for each channel. The color histogram operator $\mathbf{H}(\mathcal{R})$ is used to create a color histogram for \mathcal{R} . Each channel of the color space is independently partitioned into disjoint intervals A : k intervals for the first channel, m for the second, and n for the third channel. In the following, the intervals are obtained by partition of $[0,1]$ into equal-length subintervals.

$$\mathbf{H} \equiv \mathbf{H}(\mathcal{R}) \quad (3.11)$$

A color histogram constitutes a three-dimensional data cube, consisting of $k \times m \times n$ entries, called bins. The value of each bin $b_{(u,v,w)}$ corresponds to the frequency of pixels whose values v are within the u -th interval for the first channel, v -th interval for the second channel and w -th interval for the third channel.

$$b_{(u,v,w)} = \sum f_a(v_{(c,i)}) \quad \forall \begin{cases} v_c \in \left[\frac{u-1}{k}, \frac{u}{k}\right[, & \text{if } c = 1 \\ v_c \in \left[\frac{v-1}{m}, \frac{v}{m}\right[, & \text{if } c = 2 \\ v_c \in \left[\frac{w-1}{n}, \frac{w}{n}\right[, & \text{if } c = 3 \end{cases} \quad (3.12)$$

$$\forall u \in [1, k] \quad \forall v \in [1, m] \quad \forall w \in [1, n]$$

where f_a is the (absolute) frequency function, $v_{(c,i)}$ is the value of the c -th channel of the i -th pixel, and b is the corresponding bin.

For the color histograms in this thesis, the HSI color space (*cf.* Kender [151], Smith [252]) is used during histogram collection.

3.5.2.1 Properties of Color Histograms

The sum of two $k \times m \times n$ color histograms yields another $k \times m \times n$ histogram, which is obtained by addition of all bins with identical indices:

$$b_{(u,v,w)}^3 = b_{(u,v,w)}^1 + b_{(u,v,w)}^2 \quad \forall u \in [1, k]; v \in [1, m]; w \in [1, n] \quad (3.13)$$

The set \mathfrak{H} of $k \times m \times n$ bin color histograms forms a vector space over \mathbb{R} with vector addition $+$ and scalar multiplication. The addition of two color histograms is associative:

$$\mathbf{H}_1 + (\mathbf{H}_2 + \mathbf{H}_3) = (\mathbf{H}_1 + \mathbf{H}_2) + \mathbf{H}_3 \quad \forall \mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3 \in \mathfrak{H} \quad (3.14)$$

as well as commutative:

$$\mathbf{H}_1 + \mathbf{H}_2 = \mathbf{H}_2 + \mathbf{H}_1 \quad \forall \mathbf{H}_1, \mathbf{H}_2 \in \mathfrak{H}. \quad (3.15)$$

\mathfrak{H} possesses an identity element $\mathbf{0}$:

$$\mathbf{H}_1 + \mathbf{0} = \mathbf{H}_1. \quad (3.16)$$

and each \mathbf{H} has an additive inverse in \mathfrak{H} :

$$\mathbf{H}_1 + -\mathbf{H}_1 = \mathbf{0} \quad (3.17)$$

The histogram operator is distributive, meaning that the histogram of the union of two disjoint regions equals the sum of the histograms of both regions:

$$H(\mathcal{R}_1 \cup \mathcal{R}_2) = H(\mathcal{R}_1) + H(\mathcal{R}_2) \quad \forall \{\mathcal{R}_1, \mathcal{R}_2 | \mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset\} \quad (3.18)$$

The norm of a color histogram is defined as the sum of all its bins:

$$\|H\| = \sum_{u,v,w=0}^{k,m,n} b_{(u,v,w)} \quad (3.19)$$

For the purpose of this thesis, a *general normalized color histogram* \hat{H} of a region \mathcal{R} of a digital image \mathcal{I} is obtained by applying the histogram operator $H : \mathcal{R} \rightarrow H$, and subsequently normalizing the histogram, dividing the amount of pixels in each bin by the total number of pixels in all bins, *i.e.* the norm of the histogram:

$$\hat{H} = H \cdot \frac{1}{\|H\|} \quad (3.20)$$

The advantage of working with normalized color histograms, as opposed to the original color histograms, is their independence from the size of the region \mathcal{R} , which allows for easier comparison of histograms of regions of different sizes.

3.5.3 Shape Model

As stated before, from the perspective of the tracking algorithm, the position of a person is represented as a 2D point \mathbf{T}_t^F on the floor plane, which corresponds to the model origin \mathbf{O}^m in world coordinates.

Therefore, the observation perspective possesses twelve degrees of freedom with regard to the world coordinate system, with a possible thirteenth. Of these degrees of freedom,

- three originate from the camera translation vector \mathbf{T}_s ,
- three from the camera rotation matrix \mathbf{R}_s ,
- three from the target translation vector \mathbf{T}_t^W ,
- and three from the floor plane normal vector \mathbf{F}^W (in world coordinates), which represents the vertical orientation of the target.

Alternatively, the last two items can be substituted by four DOF from the floor plane F and two from the translation of the target on the floor plane \mathbf{T}_t^F .

The potential thirteenth degree of freedom originates the target heading η , which is the rotation of the target around its vertical axis. For reasons stated below, the heading is not considered for the current target shape model. The inclusion of the floor plane for the model is relevant since the persons vertical orientation can be obtained by assuming the person to be upright. Any major changes in body pose, such as kneeling or sitting down, are not modeled explicitly. Consequently, any variable geometric properties of the proposed shape model can only depend on these twelve degrees of freedom, unless the tracking process itself is altered. Since none of these DOF relate in any way to the shape of a pedestrian, the shape model is considered to be static with regard to the temporal progression of tracking, and the pedestrian is treated as a rigid, non-deformable shape.

As color differences between front and back are expected to be less pronounced than those between head, torso and legs, the target heading η can be disregarded for the purpose of modeling. Consequently, the pedestrian shape model has to be symmetric towards its vertical axis, which results in a generalized cylinder shape, approximately comparable to the pedestrian shape model employed by Isard *et al.* [128]. Figure 3.4 on the following page depicts a schematic configuration of the model.

3.5.4 Shape Model Proportions

For the proportions of the shape model, Da Vinci's proportions from his notes on his famous *Vitruvian Man* [60] drawing (*cf.* Figure 3.5 on page 97) are used, which in turn refer to Vitruvius [287, pages 3.1.2-3]. Vitruvius states the proportions of the different parts of the body in sevenths of the total height. Accordingly, the proportions for the model are set at $\frac{3}{7}$ of the total height h_t for the legs, $\frac{3}{7}$ for the torso and $\frac{1}{7}$ for the head. These proportions roughly concur with more recent anthropometric data, such as Nakanishi *et al.* [199], and are therefore considered to be a sufficient approximation for average values. For the total body height h_t , an average value of 1.7 m is assumed, as given by Ogden *et al.* [208] for healthy adults.

In addition to the heights, the radii of the shape model at the horizontal junctions of the model parts have to be specified. A shoulder radius r_s of 0.21 m is used for the shape model, again based on Da Vinci and Vitruvius [60, 287]. As neither author provides further proportions for hips, feet or head

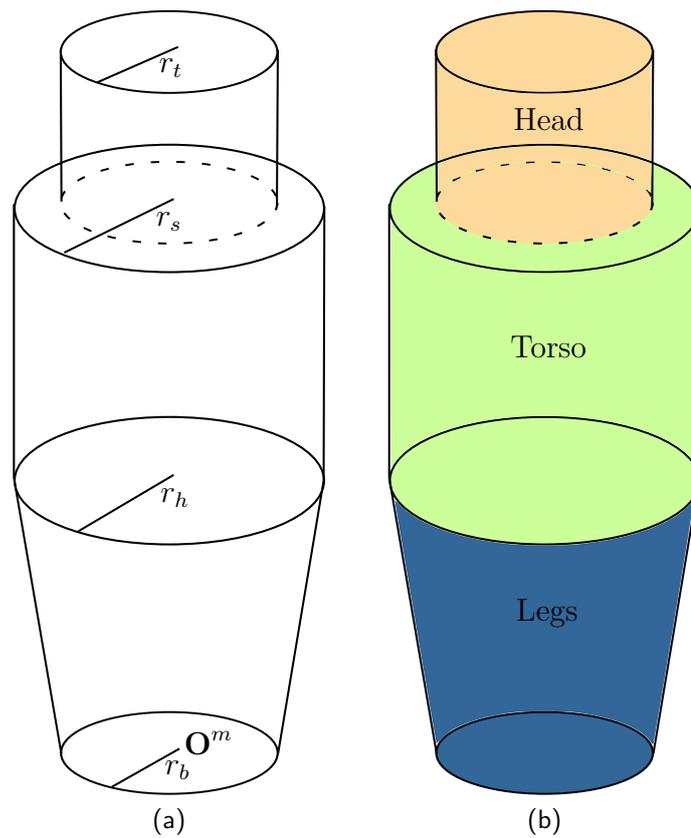


FIGURE 3.4: Schematic of the generalized cylinder model used to approximate human shape, (a) displaying the radii at different height segments and (b) colored to exemplify a typical distribution for clothing colors. Note, that the schematic is not up to scale.

circumference, different sources are required. The hip radius r_h is set at 0.17 m, based on the hip circumference stated in [199]. The foot radius r_b of 0.13 m is also based on [199] while the head radius r_t is based on [36].

Effectively, this means that the model consist of a cylinder for the head, and a frusta for torso and legs each. A comprehensive list of the resulting shape model measurements can be found in Table 3.1 on the following page.



FIGURE 3.5: Bust of Marcus Vitruvius Pollio, known as *Vitruvius*, at the main entrance of TUM (a). The measurements of the shape model are based on the observations in his work *De Architectura* [287], one of the most important surviving contemporary works on roman architecture. Drawing of the *Vitruvian Man* by Da Vinci [60] (b), based on the body proportions stated by Vitruvius.

The origin \mathbf{O}^m of the local coordinate system of the shape model is located at the center of the model base, and its z -axis is equal to the vertical model axis. As the model is circular, the x and y axes are chosen arbitrarily, as long as all axes remain orthogonal to each other. Within the local coordinate system, the surface of the generalized cylindrical pedestrian shape model is described by the following equations:

Radii	Symbol	Value
Top/head radius	r_t	0.09 m
Shoulder radius	r_s	0.21 m
Hip radius	r_h	0.17 m
Base/foot radius	r_b	0.13 m
Heights	Symbol	Value
Total height	h_t	1.70 m
Head height	h_h	0.24 m
Upper body height	h_u	0.73 m
Leg height	h_l	0.73 m

TABLE 3.1: Specifications for the generalized cylindrical approximated pedestrian shape model.

$$x^2 + y^2 = r^2 \quad (3.21)$$

for x and y in each circular model section (where the intersecting plane is orthogonal to \mathbf{F}^W) and

$$r(z) = \begin{cases} r_b + z \cdot \frac{r_h - r_b}{h_l}, & \text{if } 0 \leq z < h_l \\ r_h + (z - h_l) \cdot \frac{r_s - r_h}{h_u} & \text{if } h_l \leq z < h_l + h_u \\ r_t & \text{if } h_l + h_u \leq z < h_t \end{cases} \quad (3.22)$$

for the radius r of the circular model sections.

To summarize, the shape model generation step results in a rigid three-dimensional shape model, composed of three body parts: head, torso and legs. Each body part is composed of either a cylinder (head) or a conical frustum (torso and legs). The measurements of the model parts are based on average human body proportions, as reported in the literature. However, the model in its current form is inconvenient, since the self-occlusion of the model is difficult to compute. Consequently, the model is further adjusted, as described in the subsequent section.

3.5.5 Polygon Mesh Shape Model

As mentioned in Section 3.4 on page 85, the generalized cylinder shape model is approximated with shape atoms to produce a second-level shape model, convenient for further processing. In the following, shape atoms are realized as triangles.

Processing of polygon meshes, particularly triangle meshes, is a common technique in the domain of computer graphics (*cf.* Botsch *et al.* [26]). A mesh consists of polygonal *faces* and *vertices*, which span the faces. To convert the shape model into a triangle mesh, circles are transformed into regular polygons, which creates a structure composed of prisms (prism and frusta) for the body parts. A set of two techniques, known as *triangle strips* and *triangle fans* is employed to transform the faces of these prisms into triangles, a process also referred to as *tessellation*.

Triangle strips are created by evenly distributing n vertices $\mathbf{V}_{(i,s)}$, $i \in [0, \frac{n}{2}] \vee s \in [0, 1] \vee i, s \in \mathbb{N}_0$ along the longer edges of a rectangle, so that the first and last vertices align with the corners of the rectangle. Subsequently, triangles are created from vertices $\mathbf{V}_{(i,0)}$, $\mathbf{V}_{(i,1)}$, $\mathbf{V}_{(i+1,0)}$ and $\mathbf{V}_{(i+1,0)}$, $\mathbf{V}_{(i+1,1)}$, $\mathbf{V}_{(i,1)} \vee \{i | i \bmod 2 = 0\}$. Triangle fans, on the other hand, are created by inserting vertices at the corners and center of a polygon, and creating one triangle each from the center and two neighboring corners, so that all triangles meet in the center. Both techniques are illustrated in Figure 3.6 on the following page.

To determine the exact size of the triangles used for the tessellation, the sampling density of model vertices has to be specified. To that end, the approximate vertex distance d_v is introduced. From d_v , the number of edges of polygons approximating circles n_e is derived by inspecting the largest circle to be transformed, which in case of the shape model presented here is the circle at shoulder height level:

$$n_e = \left\lceil \frac{2r_s\pi}{d_v} \right\rceil \quad (3.23)$$

Consequently, the vertex density increases for smaller circles, such as at the base or top of the model. However, although this method produces a higher number of triangles, it is more convenient than operating with different numbers of polygon edges for all circles. The rectangular faces of the prisms (facing laterally) for the body parts are tessellated using the triangle strip technique.

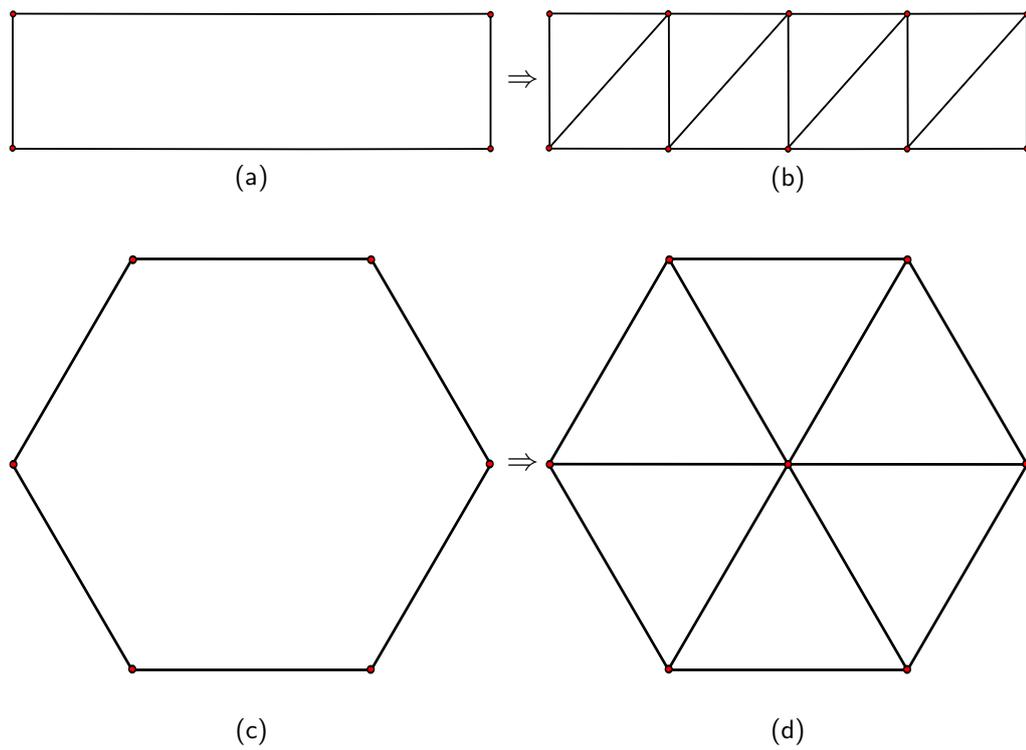


FIGURE 3.6: From top to bottom, triangle strip technique, used during the tessellation process to convert the majority of faces of the primatoids into triangles (a,b) and the triangle fan technique, used to convert the innermost polygons on the top and bottom of the primatoids (c,d).

With n_e determined, the number of vertices n_v and number of faces n_f can be calculated using the following equations:

$$n_f = 2n_e \left(\left\lfloor \frac{r_t}{d_v} \right\rfloor + \left\lfloor \frac{r_s - r_t}{d} \right\rfloor + \left\lfloor \frac{h_h}{d_v} \right\rfloor + \left\lfloor \frac{h_u}{d_v} \right\rfloor + \left\lfloor \frac{h_l}{d_v} \right\rfloor + 1 \right) \quad (3.24)$$

$$n_v = n_e \left(\left\lfloor \frac{r_t}{d_v} \right\rfloor + \left\lfloor \frac{r_s - r_t}{d} \right\rfloor + \left\lfloor \frac{h_h}{d_v} \right\rfloor + \left\lfloor \frac{h_u}{d_v} \right\rfloor + \left\lfloor \frac{h_l}{d_v} \right\rfloor \right) + 2 \quad (3.25)$$

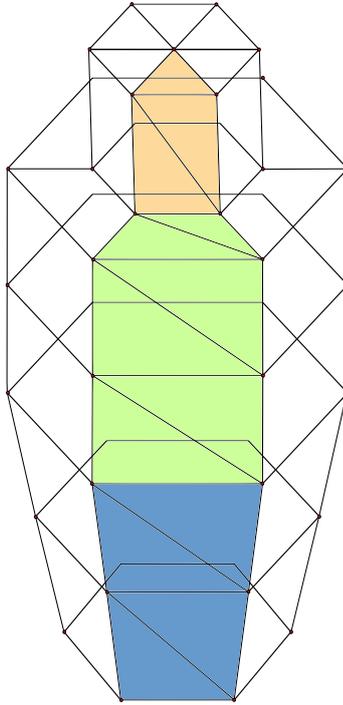


FIGURE 3.7: Illustrating the tessellation of the shape model faces, depicting the resulting polygon mesh with an exemplary vertex distance of $d_v = 20$ cm. The closest faces have been colored according to their assignment to the respective body parts.

To provide a concrete example, an approximate vertex distance of $d_v = 10$ cm results in a polygon mesh shape model consisting of $n_e = 13$, $n_e = 236$ and $n_f = 494$. Compared to state-of-the-art applications in the computer graphics domain, where several million polygons have to be projected in every frame (*e.g.* Mudbox, a 3D sculpting application, *cf.* Kermanikian [152]), this is a very low number which does not cause any problems regarding performance, but still guarantees sufficient accuracy.

To summarize, the mesh generation step results in a rigid three-dimensional geometric shape model of a pedestrian, which is an adjusted version of the model created in the previous step. The reason for the adjustment is increased convenience for the employment of the model during the operation of the tracker. The surface of the shape model is composed of a set of triangles, which in turn are defined by a set of vertices and edges. Each triangle is assigned to one of three body parts.

3.6 Model Usage During Tracker Operation

While the previous sections have focused on the architecture of the shape model, the following section is concerned with its usage in a tracking system in general, and the tracking system described in Chapter 2 on page 11 in particular. This includes the projection of the shape model, the generation of partial appearance models for the body parts, and the resulting synthesis of partial appearance models for hypothesis testing in the tracker.

The constraint that has to be imposed for the method to work is that the target walks perpendicular to the floor plane, *i.e.* is in an upright position at all times. This corresponds to the definition of a pedestrian (*cf.* Section 2.3.1 on page 14) and is an assumption that can be made regardless of the camera setup.

This constraint is essential to the proposed method, and deviation (*e.g.* the person crawling instead of walking) will cause the described method to fail, at least when using the model described in the previous section. However, as far as the ability to transfer the method to other targets is concerned, the important constraint here is that the pose of the target to be tracked is sufficiently static and known. As long as these two requirements are fulfilled, the model can be designed to reflect this knowledge, *e.g.* in case of a crawling person, an animal, or a car.

3.6.1 Model Reprojection and Weight Computation

To reiterate, the goal of the shape model is to provide an estimate for the ratio of the visible surface area of different body parts during tracking. To that end, the shape model has to be reprojected from the current estimated position of the target into the image plane. This task has to be performed once per frame of the pedestrian tracker for the partial model estimation (*cf.* Section 3.6.2 on page 104), and once for every particle of the tracker for

the generation of the hypothesis models (*cf.* Section 3.6.3 on page 107).

At this point, the advantage of the polygon mesh model becomes apparent. The projection of the triangular faces into the image plane is a straightforward task. As before, a pinhole camera model [111, p. 153] is assumed. Each face is projected into the image plane by the perspective projection $\Pi \in \mathbb{R}^{4 \times 3}$:

$$\mathbf{V}'_i = \mathbf{V} \cdot \Pi \quad \forall i \in [1, 3] \quad (3.26)$$

where $\mathbf{V}_{1\dots 3}$ are the vertices defining the triangular face $\Delta(\mathbf{V}_{1\dots 3})$. Consequently, the area of the projected triangle P^Δ can be stated as:

$$P^\Delta = \frac{1}{2} |(\mathbf{V}'_2 - \mathbf{V}'_1) \times (\mathbf{V}'_3 - \mathbf{V}'_1)| \quad (3.27)$$

In addition to the calculation of projected triangle surfaces, a technique used for managing self-occlusion of the model is required to determine the visibility of the surface parts represented by the triangles. For the purpose of the modeling, visibility is treated as a binary decision on a per-triangle basis, a process commonly referred to as polygon culling. For further reading, Sutherland *et al.* [263] provide a survey on various approaches to the hidden surface problem in general.

There are several prevalent approaches to the visibility problem that involve culling, *e.g.* view frustum culling (relevant for clipping at the edges of the camera fov; *cf.* Assarsson and Möller [9, 10]), backface culling (relevant for directed faces; *cf.* Zhang and Hoff [300], Johannsen and Carter [137]) and occlusion culling (*cf.* Coorg and Teller [53], Hudson *et al.* [126]).

For the purpose of the described shape model, the approach of culling polygons by hierarchical depth buffer scan conversion (*cf.* Greene *et al.* [104]) is selected. In short, this algorithm employs a comparison of the depth (distance) of faces from the camera to solve the visibility decision, and faces at least partly occluded are marked as invisible and discarded. Consequently, a full run of the culling algorithm results in the visible faces of the model, and the accuracy of the calculated visible surface depends on the resolution of the tessellation, as discussed in the previous section.

The area of the reprojection of a shape model part P_i^p is calculated as follows:

$$P_i^p = \sum v^\Delta \cdot P^\Delta \quad \forall \Delta \in P_i^p \quad \forall i \in [1, N] \quad (3.28)$$

where v^Δ is the binary visibility of the triangle. Consequently, the total area

of the target is derived from the sum of the area of all parts:

$$P = \sum_{i=1}^N P_i^p \quad (3.29)$$

In turn, the weights are calculated from areas of the model parts and the total area of the reprojection:

$$w_i = \frac{P_i^p}{P} \quad (3.30)$$

On a side note, by adding a view frustum culling approach, partially visible targets, *i.e.* targets that move very close to the borders of the field of view, can be accounted for. This is an additional benefit of the approach which has not yet been included in the current implementation.

To summarize, at the end of this step, the weights $w_{(i,j)}$ and the appearance model C_j have been obtained for the position of the target at the current frame. To that end, the faces of the polygon mesh shape model are reprojected into the camera, occlusion culling is applied to obtain the visible faces, the visible faces assigned to the corresponding body parts, and the visible area of the body parts in the reprojection is calculated. The weights are subsequently derived from the ratio of the visible areas of each body part.

3.6.2 Determining the Appearance Models for Body Parts

After the target has been detected and the initial appearance model has been acquired, for the first n frames, the initial appearance model continues to be employed for the tracking.

However, after each tracking step, the current appearance model C_j for the target is collected, using a ROI obtained from the segmentation of the image from the preprocessing of the frame. After a sufficient number of appearance models C_j at different positions have been acquired, and the corresponding weights have been computed from the reprojection of the shape model, the next step is the calculation of the partial appearance models from the acquired data.

The exact method of the calculation of the partial appearance models is as follows. Using Equation 3.9, an overdetermined system of J equations is obtained:

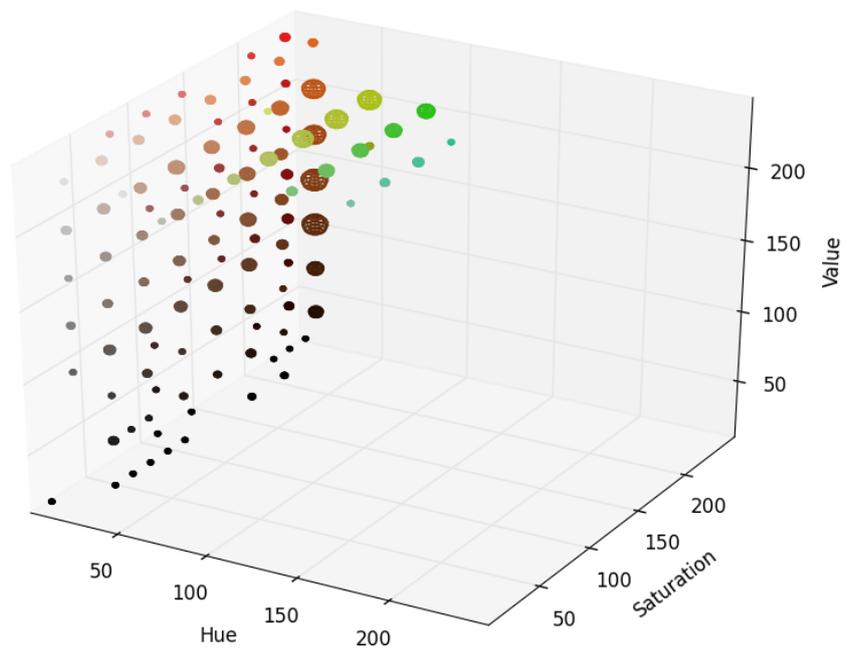


FIGURE 3.8: Example composite histogram of a clothed pedestrian. The size of the spheres represents the number of pixels within the corresponding bin in HSI color space, while the color represents the average color of the pixels in that bin.

$$C_j = \sum_{i=1}^I w_{(i,j)} \cdot C_i^p \quad (3.31)$$

where C_j are the complete appearance models, $w_{(i,j)}$ are the weights for each model part for each j , and C_i^p are the appearance models for the respective model parts, and $I > J$.

At this point, a least squares approach (as described in previous sections, *e.g.* Section 2.7.2 on page 46) is employed to solve the overdetermined system of equations, yielding estimates for the C_j :

$$\begin{aligned} C_1 &= w_{(1,1)}C_1^p + w_{(1,2)}C_2^p + w_{(1,3)}C_3^p \\ C_2 &= w_{(2,1)}C_1^p + w_{(2,2)}C_2^p + w_{(2,3)}C_3^p \\ &\vdots \\ C_J &= w_{(1,J)}C_1^p + w_{(2,J)}C_2^p + w_{(3,J)}C_3^p \end{aligned} \quad (3.32)$$

As the color distributions are modeled as histograms, and the values of the histograms' bins are assumed to be independent according to this model, this system of equations can be split into a set of n overdetermined systems where n is the number of bins in the histogram. Effectively, this means that the value of each bin is computed independently. For the color histogram models employed in this work, $n = 16 \times 16 \times 8 = 2048$.

During the further operation of the tracker, this procedure is repeated in regular intervals to refine the distributions. However, due to increasing computing time for the solution of the overdetermined system, there is an upper limit J_{max} to the number of samples J processed. Therefore, only a fraction of the weights $w_{(i,j)}$ and appearance models C_j are kept for processing.

To achieve good results for the approximation of the C_i^p , the selection of the correct samples from the available samples is important. At some point during the tracking, the number of observations made exceeds J_{max} , and a subset of samples is determined.

Ideally, should be selected where the weights are not too similar, in order to avoid the model adaptation from driving itself into local maxima. To that end, the following algorithm is applied when adding a new observation:

- in the initial state, there are J observations, and the Euclidean distance of their weights $w_{(i,j)}$ to each other is known

- calculate the Euclidean distance of the weights $w_{(i,J+1)}$ of the $J + 1$ th observation to each of the J previous observations
- for each of the $J + 1$ observations, calculate the sum of the two least distances of its weights from the ones of other observations
- discard the observation for which the above step yields the smallest value, consequently ending up with J observations again

Over time, the model is refined further, as observations from the entire possible range of perspectives are being processed, and the sampled observations are distributed evenly across the full range of observations.

To summarize, the step described in this section yields the appearance models C^p_i for each body part, by solving an overdetermined system which in turn was obtained from the weights provided by the shape model and the appearance models C_j collected.

3.6.3 Generating the Color Distribution for Tracking Hypotheses

Once estimates for C^p_i have been obtained, assembling an appearance model C used to test a tracking hypothesis (*cf.* Section 2.9.3.1 on page 56) at position \mathbf{T}_t^W is straightforward. The weights $w_{(i,j)}$ are calculated from \mathbf{T}_t^W via the shape model reprojection, as described in Section 3.6 on page 102. Subsequently, the values obtained are applied to Equation 3.9, which yields the predicted appearance model for the hypothesis. The remainder of the hypothesis test is unchanged from the method described in Section 2.9.3.1.

3.6.4 Transition of a Target Between Views

If a target transitions between camera FOVs, the estimates for C^p_i are retained. This is done assuming that neither disparities in illumination, nor in camera properties create the requirement to use any kind of intensity balancing or color balancing, such as a color brightness transfer function (BTF) [69], or that these methods have previously been applied in the preprocessing step, which would otherwise have to be applied to the appearance models C^p_i as well, in an appropriate form.

At the point of transition, the target translation \mathbf{T}_t^W and floor plane \mathbf{F} remain constant, while the translation of the sensor \mathbf{T}_s is varied. As a consequence,

the weights w_i change, and the combined distribution C has to be recalculated using Equation 3.3.

This procedure results in an appearance model C which more closely reflects the appearance of the person under the changed perspective, providing an advantage when compared to simply using the combined distribution from the last observation in the previous FOV.

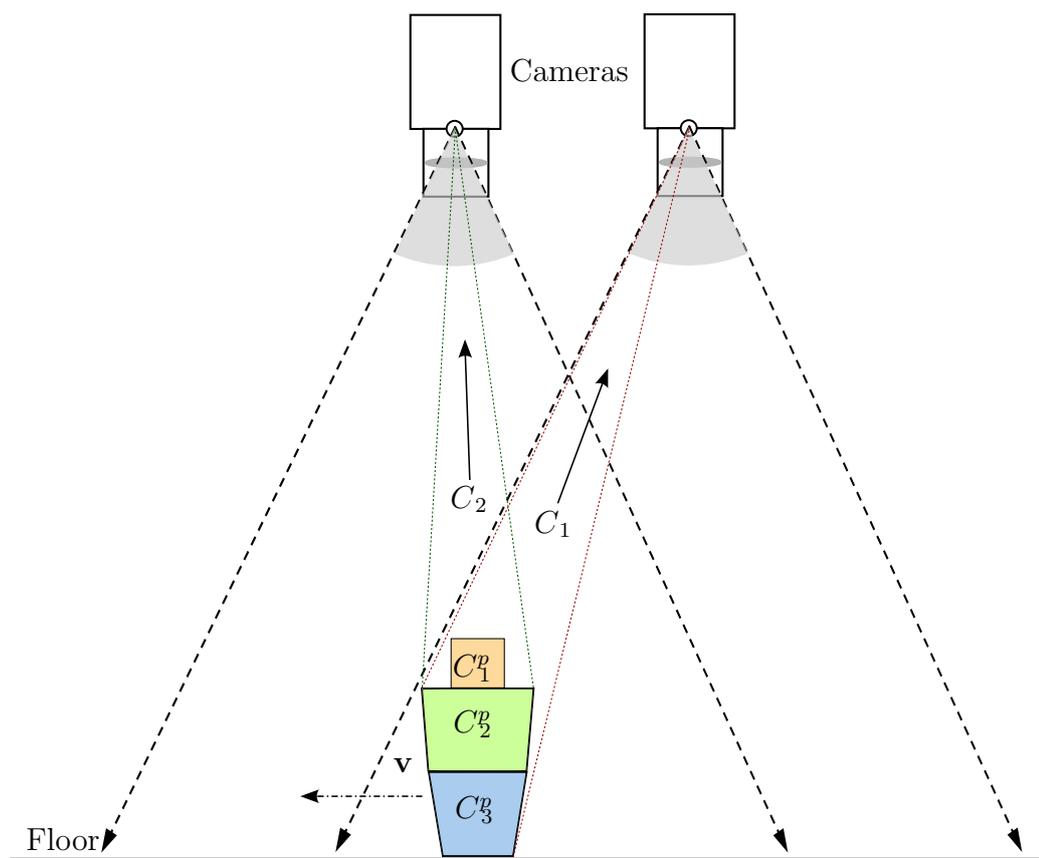


FIGURE 3.9: A two-dimensional scheme depicting the transition of a tracked pedestrian from one camera field of view to another for a top-down camera grid layout, illustrating the change in perspective and visible pedestrian surface parts.

3.7 Experimental Evaluation

The experiments described in this section were performed on the same image data as the evaluation described in the previous chapter (Section 2.10 on

page 62), therefore the experimental setup and method are not reiterated at length at this point.

The performance of the complete tracking system regarding accuracy in single view, multi-view transition and identity recovery is evaluated, on the exact same data as the evaluations in Section 2.10. In that, the relevant difference from the previous evaluations is constituted by the fact that the static appearance model for the tracking targets is replaced with the adaptive appearance model described at length in this chapter. Afterward, conclusions on the relative performance of the adaptive appearance modeling approach versus the static appearance modeling approach can be drawn by comparing the results of both evaluation runs.

3.7.1 Single-View Tracking Accuracy

The evaluation of the single view tracking accuracy of the adaptive appearance modeling approach follows the same process as its counterpart in Section 2.10.2 on page 63. To reiterate, the error metric used for the accuracy of the single view tracking is the relative tracking error $d_{\bar{r}}$, where:

$$d_{\bar{r}} = \frac{d_a}{2\bar{r}} \quad (3.33)$$

and

$$d_a = |\mathbf{t}_g - \mathbf{t}_c| \quad (3.34)$$

with \mathbf{t}_g and \mathbf{t}_c being the positions of ground truth and target candidate, respectively, in the image coordinate system, and $2\bar{r}$ being the average size of the circumcircle of the target silhouette in the image.

Table 3.2 on the following page and Table 3.3 on the next page list the results of the evaluation for $d_{\bar{r}}$ and d_r , respectively.

When comparing the results to the results for the static appearance model (*cf.* Section 2.10.2.2 on page 68), several observations stick out. The observations for $d_{\bar{r}}$ shall be considered first. Comparing the overall performance as measured by $d_{\bar{r}}$ over all sequences, there is only a slight increase of 4 percent, from 0.208 to 0.200. However, the improvement is more pronounced when considering only the lemniscatoid tracks. Here, an improvement from 0.176 to 0.156 is observed, amounting to a significant increase in accuracy of approximately 11 percent. In comparison, the performance of the tracking

Sequence	Trajectory	$\mu(d_{\bar{r}})$	$\sigma(d_{\bar{r}})$	N_i
\mathfrak{S}_{42}	ellipsoid	0.233	0.09	1000
\mathfrak{S}_{43}	ellipsoid	0.208	0.08	1000
\mathfrak{S}_{44}	ellipsoid	0.246	0.07	1000
$\sum_{i=44}^{42} \mathfrak{S}_i$	ellipsoid	0.229	0.08	3000
\mathfrak{S}_{45}	lemniscatoid	0.158	0.06	1000
\mathfrak{S}_{46}	lemniscatoid	0.154	0.05	1000
$\sum_{i=46}^{45} \mathfrak{S}_i$	lemniscatoid	0.156	0.06	2000
$\sum_{i=46}^{42} \mathfrak{S}_i$	lemn./ellipt.	0.200	0.08	5000

TABLE 3.2: Accuracy evaluation for the MCMC pedestrian tracker for single-view tracking, mean μ and standard deviation σ for the relative tracking error $d_{\bar{r}}$. N_i denotes the number of video frames for which the accuracy was measured.

Sequence	Trajectory	$\mu(d_r)$	$\sigma(d_r)$	N_i
\mathfrak{S}_{42}	ellipsoid	0.224	0.08	1000
\mathfrak{S}_{43}	ellipsoid	0.203	0.07	1000
\mathfrak{S}_{44}	ellipsoid	0.235	0.07	1000
$\sum_{i=44}^{42} \mathfrak{S}_i$	ellipsoid	0.221	0.08	3000
\mathfrak{S}_{45}	lemniscatoid	0.181	0.07	1000
\mathfrak{S}_{46}	lemniscatoid	0.188	0.06	1000
$\sum_{i=46}^{45} \mathfrak{S}_i$	lemniscatoid	0.184	0.07	2000
$\sum_{i=46}^{42} \mathfrak{S}_i$	lemn./ellipt.	0.206	0.08	5000

TABLE 3.3: Accuracy evaluation for the MCMC pedestrian tracker for single-view tracking, mean μ and standard deviation σ for the relative tracking error d_r . N_i denotes the number of video frames for which the accuracy was measured.

on the ellipsoid paths remains unchanged at a relative error $d_{\bar{r}}$ of 0.229.

Similarly, for d_r , the performance of the tracking on the ellipsoid paths remains unchanged at 0.221, while the performance on the lemniscatoid paths improves from 0.208 to 0.184, which constitutes an increase in accuracy of 12 percent. If the overall increase in performance is slightly larger in 0.216 to 0.206, it is only due to statistical reasons, since the weight of the lemniscatoid paths in the performance over all tracks is higher for d_r than for $d_{\bar{r}}$ because of the increased error values.

Consequently, the improvement in the overall performance of the tracking can be attributed unequivocally to the increase in accuracy on the lemniscatoid tracks, which is interpreted in the following. Since every other part of the tracking pipeline remains unchanged, the increase in accuracy consequently has to be linked to the changes in the appearance model. Therefore, the changes in the appearance model have to account for two separate observations:

- (A) Tracking using the adaptive appearance model proposed in this chapter exhibits a significant reduction in the tracking error, and consequently an increase in performance, on the image sequences where the target moves in a lemniscatoid paths.
- (B) Under the same circumstances as in (A), tracking does not exhibit a significant change in performance on the image sequences where the target moves in an ellipsoid path.

From the viewpoint of pedestrian tracking, the main difference in the ellipsoid and lemniscatoid paths under the current experimental setup is found in the perspective the camera has on the target in both tracks. As discussed previously, the silhouette of the target is larger when the target moves toward the edges of the FOV, caused by the perspective change. Similarly, the central axis of the target aligns with the camera principal axis near the FOV center, whereas the angle between these two axes increases monotonically towards the FOV borders.

The spatial relation between camera and target for the ellipsoid tracks, taking direction into consideration, varies only by the z -rotation of the target when considering an ideal elliptic track. In comparison, an ideal lemniscatic path exhibits considerable variation of the angle between camera principal axis and target, almost the full range of variation possible within the camera FOV without truncating the visual representation of the target at the FOV edges. Naturally, it has to be considered that some deviation from the ideal

paths occurs in the image data. Still, the lemniscatoid paths display a much more profound variation in the perspective than their ellipsoid counterparts.

There are three items resulting from the differences in perspective which pertain to the interpretation of the experimental results, and these are as follows:

- (1) The center of gravity of the target's silhouette, standpoint of the target (*i.e.* model origin \mathbf{O}^m) and center of gravity of the target in world space align near the center of the FOV and diverge towards the edges of the FOV. This observation has been used previously to explain the higher average accuracy (*i.e.* lower pixel error d_a and relative pixel error $d_{\bar{r}}$) for the lemniscatoid tracks compared to the ellipsoid tracks.
- (2) The size of the silhouette of a target increases monotonically from the center of the FOV towards the edges. Consequently, the size of the silhouette is approximately constant for the ellipsoid tracks, while it varies strongly for the lemniscatoid tracks. This fact has been exploited previously to design the error metric d_r in Section 2.10.2.3 on page 69 to compensate for (1).
- (3) The appearance of the target, as represented by its color distribution C is considerably less varied for the ellipsoid paths than it is for the lemniscatoid paths.

The effects of (1) and (2) on tracking accuracy have already been discussed in Section 2.10.2.2 on page 68, and none of them relate in any way to the changes made to the appearance model between the experiments described at that point and the experiments described here. Consequently, (3) is left to explain for (A) and (B).

To summarize, given the fact that the goal of the work described in this chapter was to improve the performance of the tracking under varying perspectives, there is a strong indication that

- (α) the increase in accuracy is explained by the improved capacity of the adaptive appearance model presented here to predict the appearance of the target under varying perspectives compared to the static appearance model acquired at target detection and that
- (β) this improved capacity directly results in an improved performance of tracking of the target when the perspective deviates from the perspective present at target acquisition.

3.7.2 Multi-View Tracking Performance

As stated before (*cf.* Section 2.10.3 on page 70), for the approach taken in this thesis, the relevant difference between single-view tracking and multi-view tracking is the point of view transition between multiple views, and therefore multi-view tracking is evaluated holistically, with the view transition process in mind. Experimental setup and matters of performance measurement have already been covered in Section 2.10.3.1 on page 71.

3.7.2.1 Results

Table 3.4 displays the results of the evaluation for multi-view tracking with the adaptive appearance model. In comparison to the results for the static appearance model (*cf.* Table 2.12 on page 72), the success rate $f_r(S)$ of targets being tracked successfully through an entire sequence improved from 0.92 to 0.95, or 3 percent. At first glance, this difference might seem negligibly small. However, if you compare the differences in failure rate

$$f_r(F) = 1 - f_r(S) \quad (3.35)$$

instead of the success rate, the improvement from 0.08 to 0.05 constitutes an improvement of ≈ 38 percent.

Targets	Relative direction	N_t	N_S	$f_r(S)$	$f_r(F)$
1	N/A	12	12	1.00	0.00
2	$\approx \uparrow\downarrow$ (antiparallel)	24	23	0.96	0.04
2	$\approx \uparrow\uparrow$ (parallel)	24	22	0.92	0.08
< 3	all of the above	60	57	0.95	0.05

TABLE 3.4: Performance evaluation for the MCMC pedestrian tracker for multi-view tracking. N_t denotes the total number of targets in all sequences, N_S denotes the number of targets tracked successfully, $f_r(S)$ denotes the success rate (*cf.* 2.23), and $f_r(F)$ denotes the failure rate.

Another way to illustrate the significance of the difference between both results is to consider the situation where targets have to be tracked over larger distances, with a significantly higher amount of view transitions. To that end, the approximate transition success probability

$$P(S_t) = \overline{N_v} \sqrt{f_r(S)} \quad (3.36)$$

is calculated, where $f_r(S)$ is the success rate of the experiment and $\overline{N_v}$ is the mean number of transitions occurring for one target during a single try of the experiment. For reasons of simplicity, the potential of identity recovery or track recovery is not considered here, which equates to a single failed transition leading to a failed experiment try. This allows for an extrapolation of $f_r(S)$ for an experiment with a significantly higher number of transitions N_v per track. Table 3.5 and Figure 3.10 on the facing page provide a comparison of the resulting success rates for different orders of magnitude of transitions per track.

Method / N_v	1	5	10	50	100	250
Static	0.97	0.88	0.77	0.28	0.08	0.00
Adaptive	0.98	0.92	0.85	0.45	0.21	0.02
Relative	0.99	0.95	0.91	0.61	0.37	0.09

TABLE 3.5: Extrapolation of tracking success rate $f_r(S)$ for increased transition counts N_v per track, based on the experimental results for the static appearance approach (*cf.* Table 2.12 on page 72) and for the adaptive appearance approach (*cf.* Table 3.4 on page 113). For the experiments conducted here, $\overline{N_v} = 3.27$.

3.7.3 Target Identity Maintenance and Recovery

The general procedure for the evaluation of the capacity of the appearance model to recover the identity of a lost target has previously been described in Section 2.10.4 on page 72. To summarily reiterate, a classification experiment with $M = 4$ classes is conducted, consisting of a training step to assemble the reference appearance models, and a test step, in which these models are compared against candidate appearance models obtained from the test data using Bhattacharyya distance D_B (*cf.* Equation 2.24) as a measure of similarity. To evaluate the success of the experiment, classification for each class of test targets is treated as a binary classification task, and the recall rate of the classification (*cf.* Section 2.10.4.1 on page 74) is employed as a measure of performance.

Although the experimental procedure remains constant, two effective differences are caused by the switch from static appearance model to adaptive

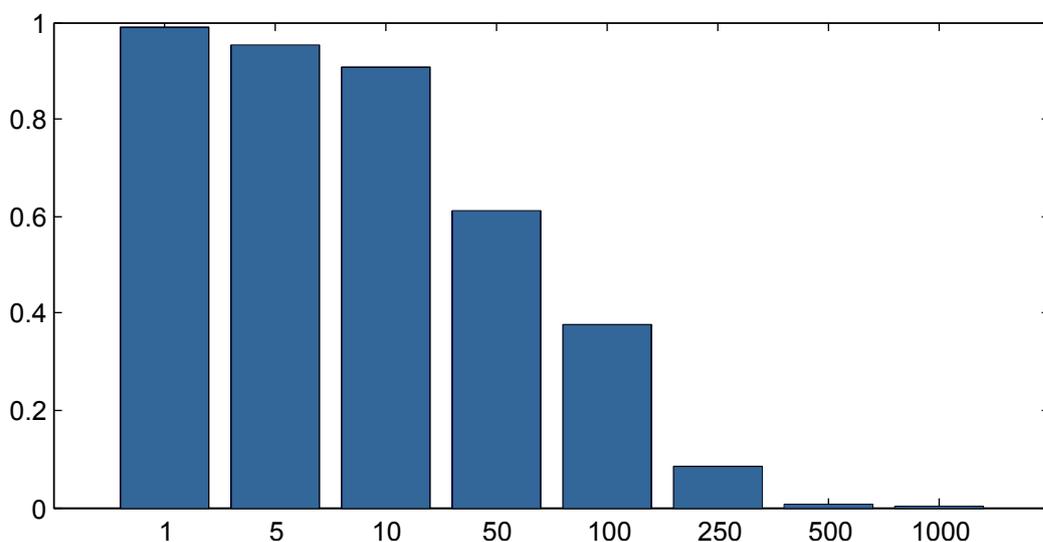


FIGURE 3.10: Illustrating the extrapolation of tracking success rate $f_r(S)$ for increased transition counts N_v per track. Here, the extrapolated success probability of the static appearance approach is depicted as a percentage of the corresponding success probability of the adaptive appearance approach, based on the numbers in Table 3.5 on page 114. The x -axis denotes the varying transition counts N_v per track, while the y -axis denotes the relation of success probabilities.

appearance model. During the training phase, the reference model is no longer assembled from a single detection, but from multiple histogram collections over the course of the entire training phase. Therefore, training phase actually holds meaning beyond the first frame, as opposed to the situation for the static appearance model.

During the test phase, on the other hand, the candidate model is still assembled from a single frame. The reference model it is tested against, however, is generated from the partial appearance models \mathcal{A}^p using Equation 3.9. This way, a more accurate representation of the appearance model of the reference target at the current position can be used in the comparison.

3.7.3.1 Results

Table 3.6 on the following page provides the results of the evaluation. In comparison to the results for the static appearance model (*cf.* Table 2.13), the overall performance has increased slightly, from 0.88 to 0.92. Certainly, it is debatable whether this constitutes a significant improvement or merely a statistical outlier, since due to the small N , the difference is caused by a

single instance. Again, however, the increase can be put into perspective by comparing the rates of erroneously classified instances

$$f_r(I) = 1 - f_r(C), \quad (3.37)$$

where a reduction from 0.12 to 0.08 caused by the switch from static to adaptive appearance modeling constitutes an improvement of ≈ 33 percent.

Participant	N	$f_a(C)$	$f_r(C)$
\mathcal{P}_1	8	7	0.88
\mathcal{P}_2	8	7	0.88
\mathcal{P}_3	8	8	1.00
$\mathcal{P}_{1\dots 3}$	24	22	0.92

TABLE 3.6: Results for the evaluation of target identity management and recovery. N denotes the total number of instances (*i.e.* targets), $f_a(C)$ denotes the number of correctly classified instances, and $f_r(C)$ denotes the recall rate (*cf.* Equation 2.25). $\mathcal{P}_{1\dots 3}$ denote the participants, as listed in Table 2.9.

3.8 Summary and Discussion

In this chapter, an adaptive method to refine color distribution models for tracking pedestrians by incorporating perspective information was demonstrated. Several related experiments were conducted to evaluate the described approach versus a static appearance modeling approach. To briefly summarize the result of the detailed analysis conducted in the previous section, the adaptive appearance modeling approach proposed here outperforms the static appearance approach in all three categories that were evaluated. Figure 3.11 on the next page provides an overview of the relative performances in these areas.

The initial shape modeling approach providing the groundwork for the remainder of the appearance modeling process is based on BRAMBLE (*cf.* Isard and MacCormick [128]). In contrast to their approach, however, a proper projection model is employed in lieu of the simplified projection model which Isard and MacCormick propose (*cf.* Figure 3.12 on page 118). This incurs the advantage of being able to vary the floor plane with regard to the observation perspective, so that the model can be employed for images from

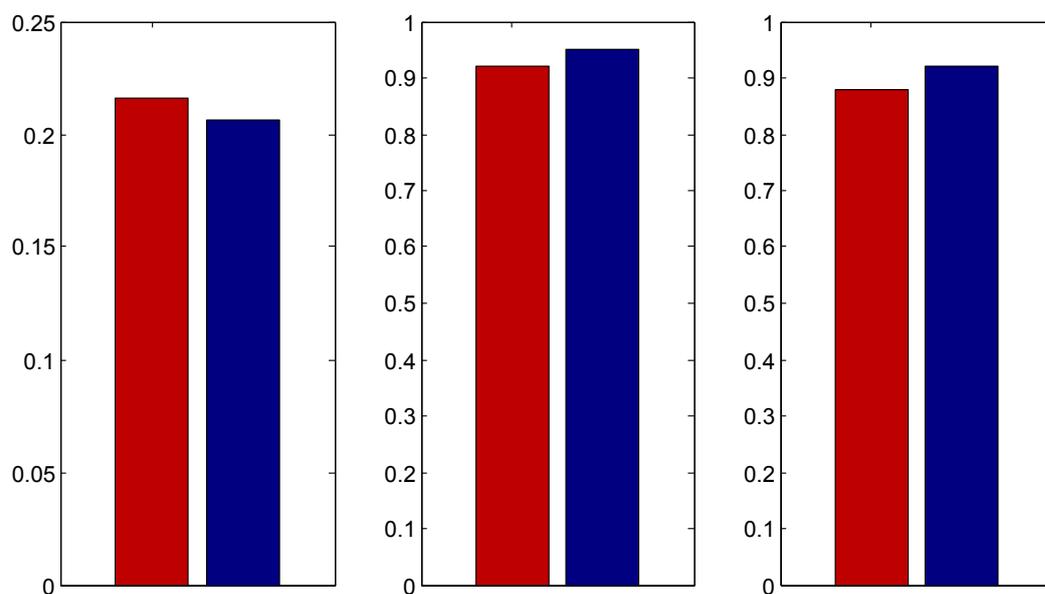


FIGURE 3.11: Comparison of the tracking performance for the static (red) and adaptive (blue) appearance approaches presented in this thesis, across the three evaluated performance categories of single-view tracking (left), view transition and multi-view tracking (center), and target identity management and recovery (right). Note, that the single view tracking performance was measured by the relative tracking error d_r , where a smaller value signifies superior performance. Contrarily, multi-view tracking and view transition performance was measured in success rate $f_r(S)$, where a higher value signifies superior performance.

cameras tilted against the floor plane, as well as being able to model perspective changes in non-lateral views. Both of these capacities have been demonstrated by the evaluation in the previous section, where the majority of images employed in the evaluation were taken from supracranial observation perspectives, and the relation of observation perspective and floor plane varies as well, as evidenced by the extrinsic camera parameters found in the appendices (*cf.* Table B.3 on page 148 and Table B.4 on page 149).

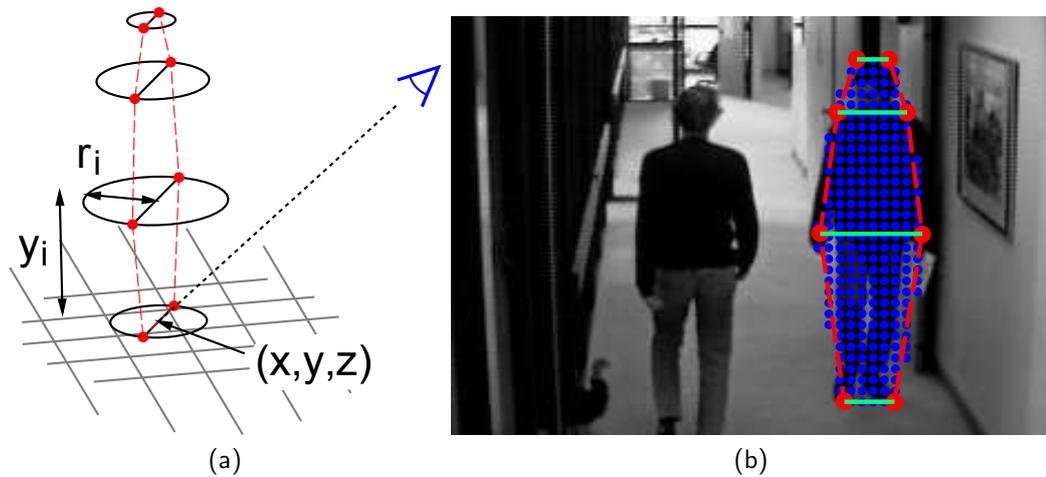


FIGURE 3.12: Generalized cylinder pedestrian shape model, as proposed by Isard and MacCormick [128]. The pedestrian shape model used in this dissertation was developed with Isard and MacCormick's model as a starting point. To the left, a schematic of the generalized cylinder model and its projection (a). To the right, the projection of the shape model overlaid on an image depicting a pedestrian (b). Images taken from [128].

A substantiated performance comparison of the appearance modeling approach presented in this thesis versus other approaches found in related work proves to be difficult for two reasons. Firstly, approaches on appearance modeling are strongly interconnected with their intended application environment, and often tailored to accommodate a specific observation perspective. Secondly, the experiments described in the previous section were conducted on the camera system described in Chapter 2 on page 11, which has not been available to other research groups.

Consequently, comparison of the results with those found in related work could only provide a very rough impression of the relative performance of the approaches. However, what can be stated with high confidence regarding the performance of the approach presented here, is that it outperformed the

static appearance approach during evaluation in all relevant categories, as illustrated by Figure 3.11 on page 117.

Several further areas of application where the described approach can be put to use come into mind. Take, for example, a typical camera configuration for the surveillance of underground train platforms, as depicted in Figure 3.13 on the next page. The peculiar overlap pattern in the camera FOVs causes repetitive gradual shift in observation perspective for a pedestrian walking the length of the platform within a single FOV, and sharp change for a target crossing from one FOV into the next one. The adaptive appearance approach presented in this chapter would provide a great benefit to color-based tracking under these conditions, especially during view transition.

Further areas of application present themselves when looking beyond pedestrians as potential targets. One characteristic trait which qualifies clothed humans as targets for this approach is the fact that regarding their color properties, they consist of different parts which feature relatively uniform color distributions. By modifying the shape of the geometric model, the approach can be customized for targets which share that quality, although in some cases, the orientation of the target would have to be considered as well. Some examples are provided in Figure 3.14 on page 121. The concretization of this, however, is beyond the scope of this thesis and would have to be investigated further at a later date.

If one considers the plethora of scientific image and video databases for pedestrian tracking, such as those published by Krinidis *et al.* [163] or those used for the regular Performance Evaluation of Tracking and Surveillance (PETS) challenges [76, 80, 81], or evaluation frameworks such as the CLEAR MOT metrics [20], it is obvious that the research infrastructure in that domain is well developed. A similar abundance does exist for detection and object recognition tasks. On the other hand, universally accepted performance metrics and frameworks for the evaluation of appearance modeling are currently lacking in the scientific community, for the reasons stated above. It stands to hope, that future work on this topic will see the emancipation of appearance modeling from tracking, detection and recognition as a separate discipline with its own infrastructure of comprehensive evaluation frameworks and databases.

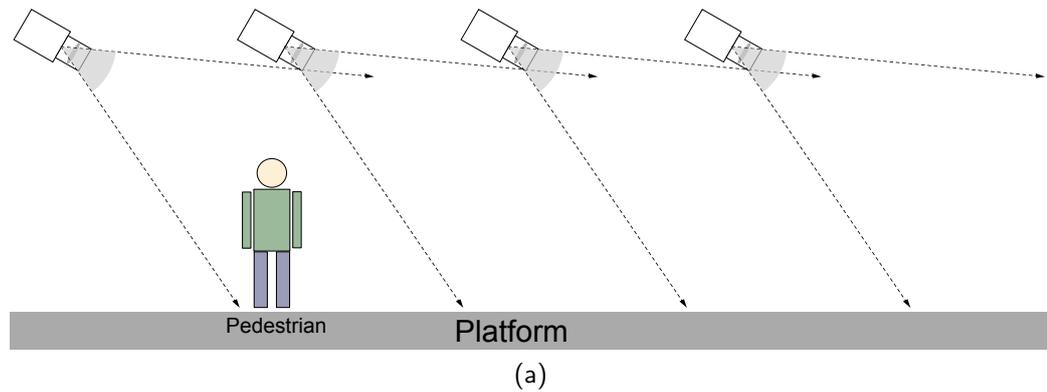


FIGURE 3.13: Illustrating a typical camera configuration for surveillance of train underground platforms. (a) depicts a two-dimensional scheme of an example camera configuration for the surveillance of an underground train platform. The cameras' FOV overlap similar to scales. (b) provides a real-world example of a similar camera configuration at an underground train station in Munich. Cameras are highlighted for improved visibility.

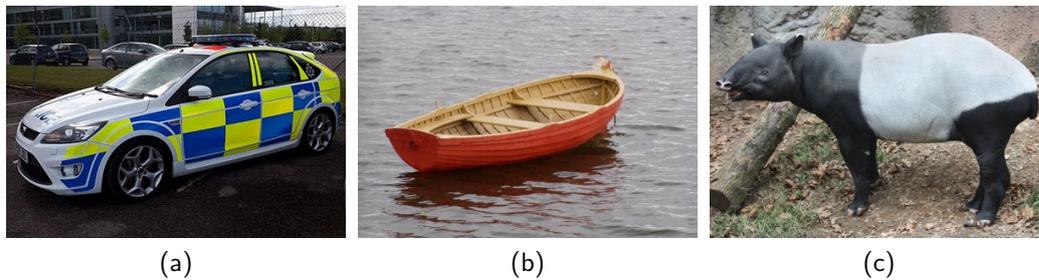


FIGURE 3.14: Examples for potential targets where color-based tracking can benefit from the adaptive appearance modeling approach described in this chapter, given an appropriate model. From left to right, a police car (a), a boat (b) and a tapir (c). All these potential targets feature well-defined vertical orientations and geometric proportions, and display different color distributions from varying perspectives.

Chapter 4

Applications

Although the system described in Chapter 2 on page 11 has been designed primarily with the end application of pedestrian tracking in mind, further opportunities present themselves regarding the analysis of the observed pedestrians. This is one of the reasons for the modular architecture of the application layer (*cf.* Section 2.9 on page 50), which facilitates the flexible, independent addition of different end applications, provided that the processing power of the hardware is not stressed beyond the point where the satisfaction of the real-time requirements is no longer possible. With respect to the sensor configuration employed (*cf.* Section 2.6.1.3 on page 34), the observation perspective also has to figure into the considerations of which end applications can reasonably be integrated into the existing framework, since some interpretation tasks prove to be incompatible with a top-down perspective, such as the interpretation of facial imagery (*cf.* Mayer [185] and Riaz [229]).

From a survey of the body of literature on the broader category of human action recognition and interpretation, several manifest possibilities for such applications come to mind. These can be grouped into larger sub-categories, the first of which to mention is the recognition of gestures. As one of the most active fields in Computer Vision, there are numerous reports on applications falling into that category, ranging from head gestures (*cf.* Kjeldsen [159]) via full body gestures (*cf.* Tollmar *et al.* [274]) to complex hand gestures, such as sign language interpretation (*cf.* Cooper *et al.* [52]). Goals of gesture recognition applications vary from the facilitation of HCI (*cf.* Jaimes and Sebe [130]) to the inference of targets' mental and emotional states (*cf.* Castellano *et al.* [42] ; El Kaliouby and Robinson [75]).

As opposed to analytic approaches in gesture recognition, work on full-body

pose recognition aims at simultaneous holistic inference of human joint configurations. However, most full-body pose estimation algorithms require a high amount of processing, which is why to a large degree, they are not compatible with real-time processing. The work of Amin *et al.* [4] constitutes a recent example. Prevalent goals of full body pose estimation are the inference of emotional states (*cf.* Schindler *et al.* [241]) or providing input for the generation of action and activity models, as described in the following paragraph.

Finally, possible applications also include the field of human activity recognition, which is achieved by combining identified atomic actions or states of the observed targets (such as “hand movements” or “standing still”) into higher order activities (such as “communicating”, “waiting” or “working”). Generally speaking, these approaches employ machine-learning techniques to pre-recorded data to generate activity models, which can subsequently be used to classify ongoing activities from real-time observations. Examples include the work of Beetz *et al.* [16], with a focus on action hierarchies, and Bodor *et al.* [25], with a focus on activity recognition from vision data.

This chapter presents two exemplative applications, which are both thematically located within the field of gesture recognition, a sub-field of automated human action interpretation. They were realized and integrated into the system described in Chapter 2 as proofs of concept, to demonstrate its versatility and extensibility.

4.1 Outline of this Chapter

The remainder of this chapter is organized as follows:

Section 4.2 on the facing page showcases the modular extensibility of the system described in Chapter 2 on page 11 for action and activity recognition, by adding a module for the recognition of handshakes occurring between pedestrians within the target area.

Section 4.3 on page 127 provides another extensibility showcase by describing a module for the recognition of pointing gestures performed by pedestrians within the target area.

4.2 Handshake Recognition

The shaking of hands can hold several meanings, the most significant of which in western culture is greeting another person. From a HRI perspective, the information that a person was greeted by another person can be an indicator that this person might have newly arrived or hold a special importance.

A handshake between two persons can be divided into two subsequent phases. In the first phase, the handshake request, both individuals extend their hands toward each other, with the extension motion from the requesting party leading. According to Jindai and Watanabe [136], who analyze the handshake request motion with the goal of transfer to a robot, this phase takes approximately 1.1 s. Assuming a resting position of the arm next to the body, the request motion has vertical and horizontal components, with the arm being raised and extended. In the second phase, the palms are clasped and the actual shaking occurs, a primarily vertical movement. The horizontal and vertical components of the motion are of note regarding the perspective of the camera towards the persons, since while a lateral perspective displays horizontal and vertical components clearly, a supracranial perspective means that horizontal motion is easier to identify than vertical motion.

4.2.1 Related Work

Gesture recognition in general has been one of the most active fields in computer vision research over the past years, as evidenced by multiple surveys on the subject, such as the ones by Daugman [63] (with a focus on faces), Gavrilu [91], who also includes work on tracking and detecting humans, Wu and Huang [296], or more recently Mitra and Acharya [193].

Regarding handshake gestures specifically, some attempts have been made at detection of these gestures from camera images, but the body of literature on the subject is comparatively sparse. Work on the subject appears more focused on identifying intent to shake hands in humans than the execution of the actual process. Sakagami *et al.* [238] report a handshake detection system developed for the ASIMO robot, which detects the extended hand of a single person in order to allow ASIMO to shake it. Unfortunately, they do not report any details with regard to their approach, with the exception of the fact that the feature extraction is performed in 2D.

Similarly, Kim *et al.* [155] regard handshake gestures from a humanoid robot's point of view, in order to distinguish them from several other gestures using a multilayer perceptron (MLP) based approach. They use the results of

Haar-cascade face detection [284, 285] to initialize a skin color model, which is subsequently used to identify the position of the hands. The information thus obtained constitutes the training data for a three-layer perceptron with squared instantaneous error backpropagation [168]. They report a success rate of 83% at recognizing handshake gestures on unknown data.



FIGURE 4.1: Two persons shaking hands, as seen from a supracranial perspective. In (a), the optical flow fields extending from the tracked pedestrian positions towards each other can be seen, before the handshake occurs. In (b), the handshake has occurred and was detected by the system, which is signified by the icon in the upper left corner in the display unit for human-readable output.

4.2.2 Method

As the distance between a dyad of targets A and B falls below a pre-determined value d_{min} , a cone-shaped field of sparse optical flow is generated extending from each person towards the other, sampling 984 points using the Lucas-Kanade method [179]. Subsequently, principal components analysis (PCA) [138, 214] is applied to reduce the dimensionality of the data, which allows for a reduction of the data by 60% without negative impact on the classification rate.

The PCA data from annotated image sequences, depicting handshake gestures and persons not exhibiting any particular gesture, is used to train a C4.5 binary decision-tree classifier (*cf.* Quinlan [224, 225]) with the WEKA machine learning suite [109]. To obtain test data for the classifier, n -fold stratified cross-validation [161] is employed. As a result of the evaluation,

83.45% correct classification rate was achieved on a per-image basis, similar to that reported by Kim *et al.* [155].

4.2.3 Integration

Regarding the system architecture as described in Section 2.4 on page 26, the handshake detection module is situated within the application layer and requires approximately 22 ms for the computations on a single core of the hardware described in Section 2.6.3 on page 41. Image data and timestamps are provided by the service layer via the KOGMO-RTDB, while the position of the persons shaking hands is provided via the pedestrian tracking module (*cf.* Sections Section 2.9.2 on page 51–Section 2.9.3 on page 55).

The work was conducted as a student project supervised by the author, and is described in greater detail in [201].

4.3 Pointing Gesture Recognition

Among their seven principles for efficient HRI, Goodrich and Olsen [101] postulate that robots should be capable of using non-verbal communication channels in order for humans and robots to be able to communicate intuitively, and thus efficiently. Pointing gestures are among the most important non-verbal communication channels, which human infants are capable of employing at the early age of 12 months, contemporaneously with the development of speech (*cf.* Thompson/Massaro [270]).

Consequently, pointing gesture recognition and extraction is among the first applications that come to mind regarding enhancing the system described in Chapter 2 on page 11 with functionality to support HRI. At first glance, the supracranial perspective of the cameras at the CCRL installation appears particularly well-suited for pointing gesture extraction since these gestures, if applied to objects distributed across a larger area, tend to have a strong horizontal component and only little vertical variation.

4.3.1 Related Work

As it happens, the literature is ripe with examples for a wide range of approaches on pointing gesture extraction. One of the earlier works is the PERSEUS system by Kahn and Swain [141], developed for use with cameras

mounted on robotic platforms and therefore perceiving humans from a lateral/anterior perspective. It combines intensity, edge, motion and disparity features to segment the person and assign body parts (hand and head) using reasoning based on anatomical properties. Pointing gestures are assumed to occur when the position of head and hand remains constant for several seconds, and the direction is extracted as the line of sight between head and hand.

In contrast to that, Carbini *et al.* [40] present a system operating on images from cameras from an anterior/supracranial perspective, which is more comparable with the perspective in the CCRL setup. As in the approach of Kahn and Swain, the positions of face and hands are extracted (using a neural-network based face detector [79] and skin color models for the hands), and the pointing direction is assumed as the line between head and hand. As an additional constraint, gestures are only assumed to occur if the distance between head and hand is sufficient, *i.e.* the hands are not held close to the body.

Kehl and van Gool [150] present an approach for use in immersive environments. They use an entire array of cameras focused on a person to extract pointing gestures, one of which is facing the person from a supracranial perspective. After applying foreground segmentation (*cf.* Mester *et al.* [189]), this perspective is used to extract the head position as the center of gravity in the overhead silhouette of a person, and the hand position, as being the point with maximum distance from this center, as an initial estimate for establishing point correspondences with the remaining cameras to extract the 3D position. Again, the direction of the gesture is estimated as the vector between head and hand.

From a household robotics perspective, Nickel and Stiefelhagen [203] use a stereo camera system to track hand and head position using skin color clusters (*cf.* Yang/Ahuja [298]) and disparity features, and head orientation using a three-layered artificial neural network (ANN) [106] with downsampled intensity and disparity histograms. The feature set obtained this way is employed to train a hidden Markov model (HMM) [226], using temporal data to identify whether a pointing gesture has occurred following a begin-hold-end pattern. Finally, the extracted features were evaluated regarding its descriptiveness for the pointing gesture direction using three approaches: (a) head-to-hand line (b) forearm line and (c) head orientation. The authors conclude, that the highest percentage of targets was correctly identified using the head-hand line approach.

Martin *et al.* [183] also approach pointing gesture extraction from an anterior

perspective in monocular images, with the application of directing a robot towards a certain point on the floor in mind. They apply cascade-boosted face detection as proposed by Viola and Jones [284, 285] to detect persons in the image and calculate a ROI for the extraction of Gabor features [188]. They evaluate different machine-learning strategies regarding their performance at the task of estimating the correct pointing angle φ and radius r , concluding that the MLP [237] outperformed the other tested algorithms for their purposes.

4.3.2 Method

At the body pose exhibited when performing a pointing gesture, the underlying skeletal structure of the relevant arm and shoulder regions can be clearly identified from the supracranial perspective by the human eye. Because of earlier positive experiences with modeling approaches considering anatomical properties, particularly with regard to facial muscle structure (*cf.* Mayer *et al.* [185–187]), it was decided to follow an approach that tries to model the rough skeletal structure underlying shoulders and extended arms from a top-down view.

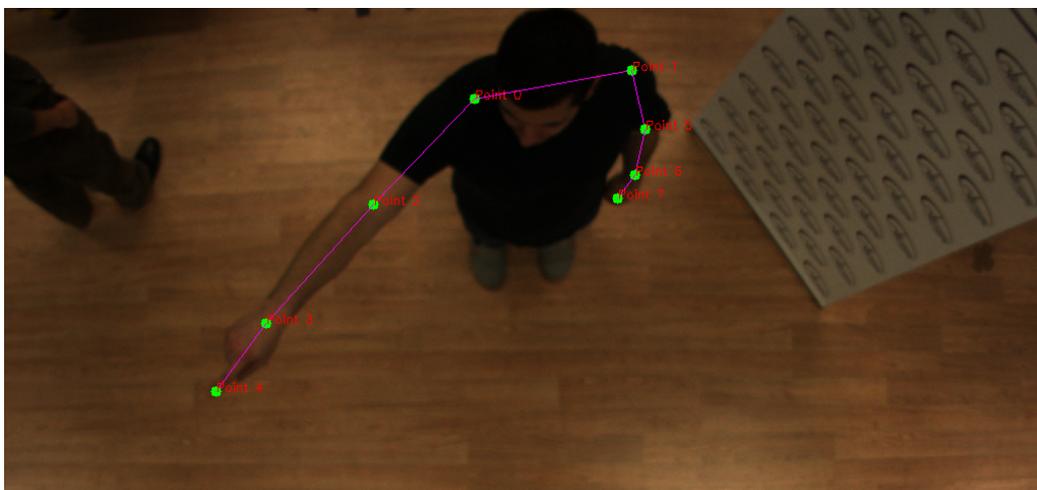


FIGURE 4.2: Illustrating the pointing gesture extraction. In this picture, the output of the shoulder/arm tracker with the 8 tracked anatomical landmarks is depicted.

To that end, a two-dimensional model with 17 degrees of freedom, classified into bone lengths and joint angles, was designed. Alternatively, the model parameters can be defined by the image coordinates of 8 points, as listed in

Table 4.1, and is depicted in Figure 4.2 on page 129. It should be noted that the model is specifically tailored to the supracranial view, and therefore not applicable to images of pointing gestures taken from *e.g.* a lateral perspective.

Landmark	Shoulder	Elbow	Wrist	Fingertip
Left	1	5	6	7
Right	0	2	3	4

TABLE 4.1: List of anatomical landmarks modeled and tracked for the pointing gesture extraction approach.

To determine the correct model parameters within each image, the model has to be fitted to the appearance of the person depicted, a task that was approached using a displacement expert (*cf.* Williams *et al.* [293]), which constitutes a relevance vector machine (RVM) [272, 273] that tracks a certain ROI within the image by estimating its displacement. Several constraints are imposed on the possible values for the different classes model parameters (angles and bone length) according to anatomical reasoning, *e.g.* the elbow angles cannot be greater than 180° .

For the training of the displacement expert, different types of feature transformations were evaluated to eventually settle on intensity and distance-to-edge features. Features were extracted from 10×10 px regions at the landmark points, and equidistant sampling of the lines connecting these landmarks with perpendicular lines. An example for the feature sampling is depicted in Figure 4.3 on the facing page.

Because of the use of images taken from image sequences rather than independently obtained images, employing n -fold stratified cross-validation to obtain test data for the training of the displacement expert would likely introduce overfitting (*cf.* Hawkins [113] for a detailed explanation), and thus be counterproductive. Therefore, different sets of sequences were recorded for testing and training purposes, and cross-database evaluation (Minus-1-DB Method, *cf.* Livhsin *et al.* [177]) was performed. The resulting displacement expert function is used to fit the model parameters to unknown data during the application of the gesture recognition module.

For the final interpretation of the model parameters regarding the pointing direction, the direction of the forearm proved to be more reliable than the direction of the hand or fingertip, respectively, since the fitting accuracy for this part of the model was found to be superior during operation on unseen data. For an assertively performed pointing gesture, as depicted in Figure 4.2

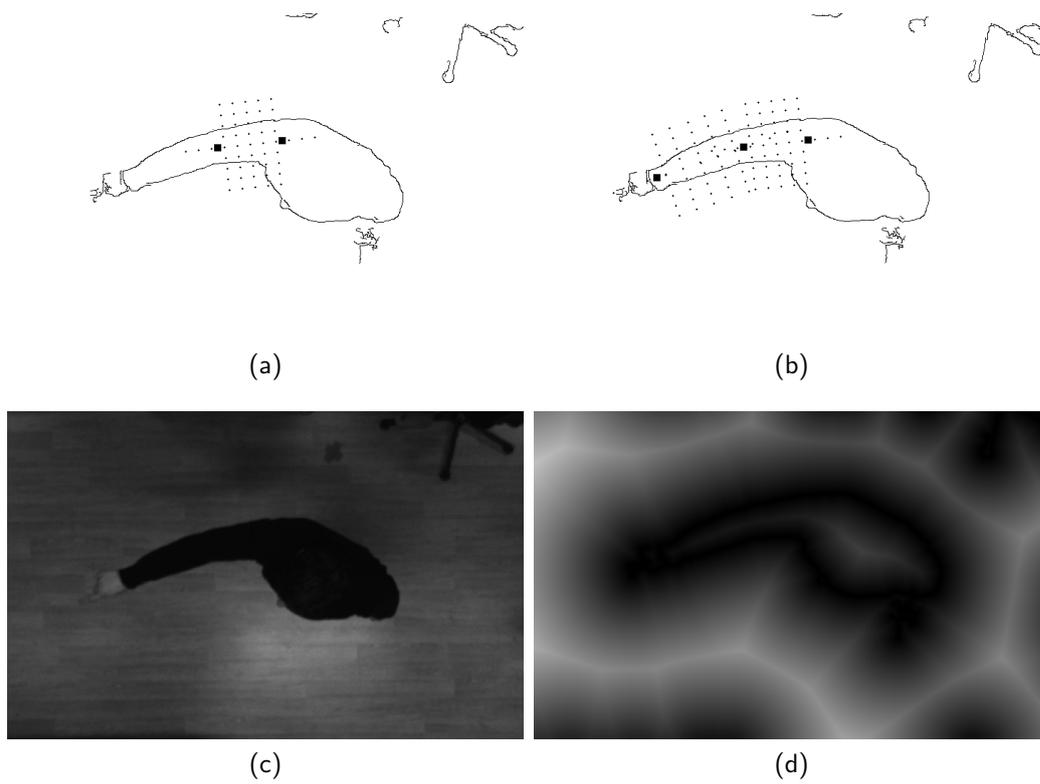


FIGURE 4.3: Feature extraction for 2D model fitting. In (a), the sampling points for the upper arm length parameter are depicted within the edge-transformed image. In (b), the sampling points for the elbow angle parameter are shown within the same image. Features sampled at these points are (c) intensity and (d) distance to edge, which are depicted in the bottom row.

on page 129, the difference between both directions is negligible regarding the intended target of the pointing gesture.

4.3.3 Integration

Similarly to the previously described handshake detection module, the pointing gesture recognition module is located within the application layer (*cf.* Section 2.4 on page 26), receives timestamped images from the service layer via the KOGMO-RTDB and pedestrian positions from the multi-view tracking module.

The work was conducted as a student project supervised by the author, and is described in greater detail in [158].

4.4 Discussion

In this chapter, two application examples for the camera system presented in Chapter 2 on page 11 were presented, where optional additional functionality was added to the system in the form of modules integrated in the application layer (*cf.* Section 2.9 on page 50). The capacity to employ the system to perform selected gesture recognition and extraction tasks was demonstrated, in the form of handshake gestures executed by two persons, and in the form of pointing gestures performed by a single person.

Combining applications such as those presented in this chapter with pedestrian tracking systems are a step towards the vision of completely integrated scene observation systems, which combine the extraction of multi-modal information from observed targets within the scene, in order to produce more efficient automated surveillance and assistance systems. The modular approach, with separate modules for each application task, communicating via IPC middleware, provides an extensible framework for such systems.

However, the modular architecture also leads to redundancy in several image transformation tasks (*e.g.* background subtraction), and consequently to unnecessary processing overhead. This issue is only partially alleviated by the transfer of shared image transformation tasks to the preprocessing layer (*e.g.* lens distortion removal, *cf.* Section 2.8.1 on page 48). A conceivable future solution, following current trends in Computer Vision, would be to generate the source code for the required modules automatically, similar to the approach of Herrmann *et al.* [117], and implement redundancy optimization within the code generation.

Although the architecture of a system with full scene observation capabilities is beyond the scope of this thesis, these small examples demonstrate the general role of the described system within the context of anticipated developments in smart surveillance and ambient-assisted living, and demonstrate the extensibility of the general approach.

Chapter 5

Summary and Outlook

As is inherent in the scientific process, gaining some answers always leads to at least as many new questions. This chapter is dedicated to rounding off the dissertation by providing a retrospective summary of the contributions and results, discussing them in the context of related work and recent trends in the connected scientific fields, and outlining some ideas and thoughts on future work and further developments.

5.1 Discussion

In this dissertation, three major scientific contributions were presented. Firstly, a vertically integrated pedestrian tracking system, consisting of three layers ranging from hardware configuration to high-level application modules, was engineered and implemented in a laboratory area, covering an indoor area of 100m^2 (*cf.* Section 2.4 to Section 2.11). Secondly, a semi-automated calibration approach for extensive multi-camera systems was implemented and evaluated, with the results comparing quite favorably against the state of the art reported by other researchers on multi-camera calibration (*cf.* Sections 2.7 and 2.7.1). Thirdly, a novel adaptive approach to color-based appearance modeling was conceptualised and implemented for the modeling of pedestrian appearance, and subsequently evaluated within the framework provided by the first system (*cf.* Section 3.4 to Section 3.8), with the performance exceeding that of state of the art color-based static modeling.

The structure of the following sections is selected in accordance with these contributions.

5.1.1 Camera System Concept and Architecture

As already hinted on in Section 3.8 on page 116, comparing the camera system described in this dissertation to other systems described in related work proves complicated, since the system is unique in its scale, with regard to the number of cameras employed (at 40) in combination with the area covered (at 100 m²).

Overall, the evaluation results for the system, as detailed in Section 2.10.2.2 on page 68 demonstrate the capability to operate for extended periods of time.

One of the trends manifesting themselves during recent years is the increasingly prevalent use of red/green/blue-depth (RGB-D) cameras, such as the cheaply available Microsoft Kinect (*cf.* Zhang [303]), in experimental surveillance setups. In retrospect, this has been one of the hardware developments which overtook the work on the surveillance system described in this thesis, and consequently merits some discussion at this point. To provide a few application examples, Collazos *et al.* employ RGB-D sensors to detect abandoned objects in controlled scenes, such as railroad cars [48]. Similarly, Hsieh *et al.* employ RGB-D sensors to count pedestrian flow at doorways, to monitor the number of persons in a building [124]. All of these examples constitute surveillance applications, in which the RGB-D sensors is uses similarly to how RGB cameras are used in the work described in this thesis. Finally, the work of Gill *et al.* [97] employs the RGB-D sensor for the recognition of individuals. Additionally, they refer to the advantages and disadvantages of the RGB-D sensor versus traditional RGB-cameras, such as those employed in the work described in this thesis. Apart from the low cost and the obvious advantage in the additional depth information, they emphasize the capacity to operate in otherwise unfavorable lighting conditions. Conversely, they identify the limited range and poor performance in natural and halogen light as disadvantages of the RGB-D sensor. In addition, the structured light pattern projected by the Kinect sensor specifically causes image quality to degrade when multiple sensors face the same surface (*cf.* Schröder *et al.* [242]). Consequently, the emergence of of RGB-D sensors does not herald the decline of traditional RGB cameras, rather both camera types are complementary regarding their application opportunities. In further consequence, research with RGB cameras is not invalidated by this development.

The fact that the system provides functionality for timed storage and replay of the image data renders it especially suited for use in research on real-time Computer Vision algorithms for multi-camera systems, where the

reproduction of identical conditions for test runs is paramount for objective comparison. This is evidenced by the evaluation of the work performed on the third contribution mentioned above, the work on color-based appearance modeling, which would not have been possible in that manner without the infrastructure provided by the multi-camera system. As such, the camera system not only constitutes a proof of concept, but, perhaps even more importantly, an invaluable research tool.

5.1.2 Automated Multi-Camera Calibration

Camera calibration remains a crucial topic for the real-world applicability of multi-camera systems, a fact that the author was painfully reminded of at several points during the practical work on this thesis. In case of a system like the one presented in Chapter 2 on page 11, where FOVs overlap only partially, severe miscalibration of the system not only invalidates the results produced for a single view, but also negates any chance of obtaining valid correspondences between views, and perform successful view transitions of moving objects. Regarding lessons learned from the work on the system, one of those is certainly that the importance of accurate and regular calibration of the system cannot be overstated.

For this reason, the efforts to equip the system with facile methods to perform the camera calibration, have proven to greatly enhance the flow of working with the system by reducing the time needed to perform the calibration and allowing for more frequent re-calibration of the system. This, in turn, was experienced to be beneficial since the manipulation-sensitive hardware components of the system – mainly, the cameras – were not sufficiently safe from accidental tampering by other activities within the observation area to exclude any manipulations over the course of several months. Although this might be avoided in theory for static multi-camera systems, it is a common problem with experimental systems such as those employed in research. What would render a higher degree of automation in calibration even more appealing is the use of non-static multi-camera systems, such as proposed by Senior *et al.* [248], where multiple pan-tilt-zoom (PTZ) cameras are employed with varying configurations.

Summarily, the multi-camera calibration procedure presented here has proven itself to be both highly accurate, as evidenced by the comparisons in Section 2.7.1 on page 45, as well as convenient in day-to-day use, as evidenced by the researcher’s experiences stated above.

5.1.3 Shape-Aware Adaptive Appearance Modeling

While a direct performance comparison versus radically different approaches proves difficult without extensive re-implementation, which would go beyond the scope of this thesis, the results from Section 3.7 on page 108 evidence that the novel approach for adaptive color-based appearance modeling presented here outperforms the state-of-the art in color-based appearance modeling. However, as already mentioned before, this state of affairs underlines the importance of an increase in public research infrastructure for appearance-based modeling, consisting of extensive image and video databases, evaluation metrics, and regular performance challenges, similar to those found in the object tracking community. For further discussion of those topics, the reader is kindly referred to Section 3.8 on page 116.

5.2 Future Work and Outlook

Unfortunately, the effort that can be put into a project such as the one detailed in this thesis in the time frame available for its completion has its limits. Consequently, some ideas for improvements on the work detailed in the previous chapters persist, either because realizing them would have been beyond the scope of the thesis, or because they result from the conclusions drawn therein. These directions of future development on the concepts and the more palpable aspects presented in this dissertation are briefly sketched out in the following.

Starting with the overall architecture of the system, a possible direction of investigation would be to change the mapping of image processing tasks (from the application layer) to processing nodes (from the hardware layer). To recapitulate, in the current system configuration, the applications are grouped by their image sources (*i.e.* cameras) and all applications processing images from one camera run on the same processing node. However, regarding the extensibility of the system, a contrary design paradigm is conceivable, where applications are grouped by functionality instead. This allows for the addition of functionality to the system without touching the system core (*i.e.* the existing hardware layer and service/preprocessing layer), beyond the limitations otherwise imposed by the 1:1 mapping of cameras to processing clients. The expected benefit, compared to the approach realized for this thesis, is constituted by a longer lifetime of the entire system due to higher versatility in the addition of new features.

Regarding the semi-automated calibration approach discussed in Section 2.7

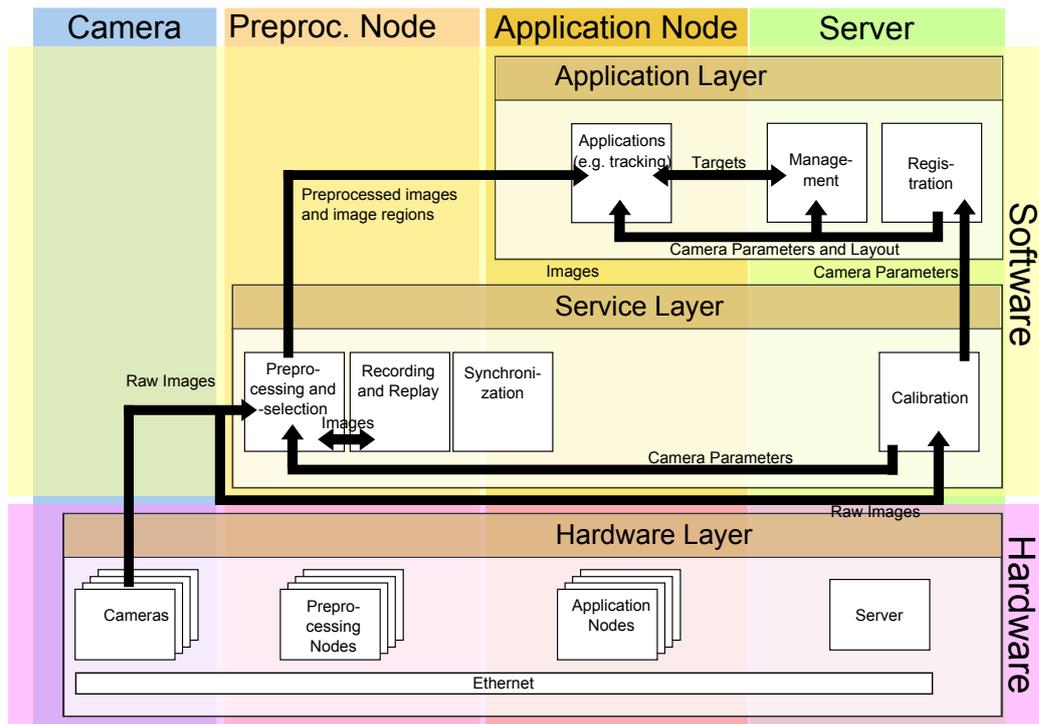


FIGURE 5.1: Illustrating an approach for future work on the architecture of the application layer (*cf.* Section 2.9 on page 50). Applications are grouped by functionality instead of by image source. This allows for more flexible extension of the system, especially where processing power constitutes a bottleneck. However, the drawback of the depicted architecture is the increased strain on communication resources, due to the requirement to transfer high-resolution images or image segments in real time between the image acquisition nodes and image processing nodes.

on page 42, the next important step would be to eliminate the need for manual contribution to the calibration process entirely. Although several self-calibration approaches have been published (*e.g.* by Armstrong *et al.* [8] or Pollefeys *et al.* [221]), which do not require calibration objects but instead use natural correspondences within the observed scene, these approaches generally lag behind with regard to the precision achieved. Furthermore, the existing approaches require moving cameras for the estimation of the internal parameters, which is also inconvenient for minimizing inter-camera re-projection error in a multi-camera system.

To promote automation while maintaining the precision allowed for by a well-defined calibration object, a possible future approach would be to attach said calibration object to an autonomous vehicle, such as a mobile robot or a unmanned aerial vehicle (UAV), which then follows a pre-defined path across all camera FOVs. Examples are depicted in Figure 5.2. This procedure would eliminate the need for a human to intervene in the calibration process by manually moving the calibration object, and would allow consequently allow for effortless re-calibration of the system in regular intervals.

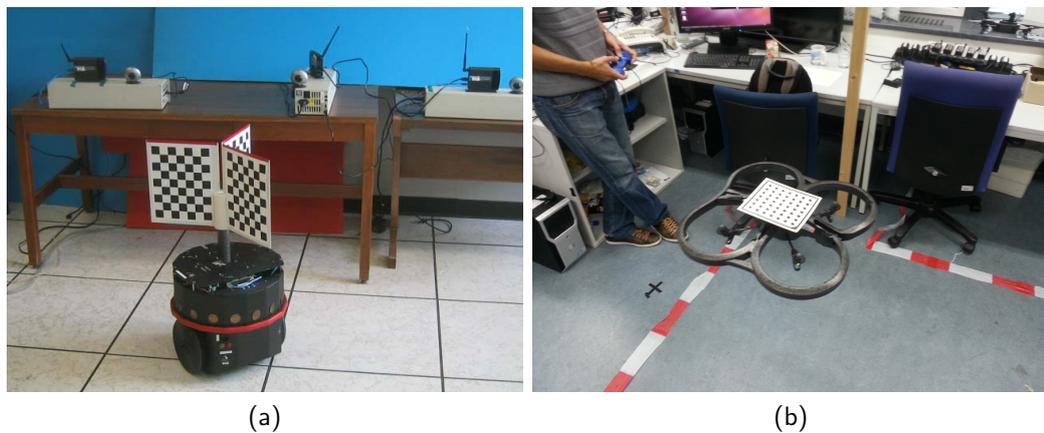


FIGURE 5.2: Illustrating approaches for future work on automation of the calibration routine. To the left, a ground-based mobile robot with a calibration object attached, that could be used to substitute manual intervention in the calibration process, taken from [228] (a). To the right, the AR.Parrot commercial UAV (*cf.* Bristeau *et al.* [31]) with a calibration object attached, which has the additional advantage of being able to vary the distance from the cameras, and has a higher variability regarding its pitch and yaw when navigating (b).

In the case of tracking on the CCRL camera setup, since the cameras are quasi-parallel, a pedestrian moves across multiple fields of view with a very

similar relation of camera and floor plane, finding himself in repetitive geometric positions in relation to the respective cameras. A practical direction for further enhancement of the proposed appearance modeling technique would be to exploit the regularity of the configuration to reduce the computational effort by performing the weight calculations based on the shape model (*cf.* Section 3.6.1 on page 102) in advance for a regular grid of potential target positions, and subsequently using radial interpolation between the acquired data points to obtain the weights during the operation of the tracker. The design of this augmented model takes the conditions at the CCRL into account, but could be applied at a wider range. It is assumed to be especially beneficial in all cases where a target with a non-uniform color distribution should be tracked across several quasi-parallel cameras using its color properties. The subway platform surveillance setup (*cf.* Figure 3.13 on page 120) provides a further example of a camera configuration meeting these requirements.

Another item that merits further investigation regarding its effect on the precision of the adaptive appearance modeling approach described in Chapter 3 on page 79 is the degrees of freedom of the pedestrian shape model (*cf.* Section 3.5.3 on page 94). Currently, the geometry of the model is static, generated using average body proportions, and not adjusted specifically to the size of the target. By combining the approach described here with an approach to estimate certain parameters (*e.g.* total height, radius) of the model based on the observations of the target, the accuracy of the weight calculations (*cf.* Section 3.6.1 on page 102) could potentially be improved. An example for a suitable approach to deformable model fitting can be found in the work of Mayer *et al.* [185, 186], where machine learning techniques are employed to estimate the deformation parameters of the CANDIDE-III 3D deformable face model [1], using Haar-like features (*cf.* Viola and Jones [284]) as image descriptors.

As of the state described in this thesis, the concrete realization of the approach presented in Chapter 3 on page 79 is restricted to pedestrians, that is human targets with an upright body pose. Within the presented application domain, one possible avenue of improvement is constituted by the concrete extension of the approach to humans displaying other body poses. Briefly put, this could be achieved by using different shape models for a pre-defined set of possible body poses (*e.g.* standing, sitting, crouching, prone), and then implementing a classifier for each frame, which determines the pose in that specific frame, and chose the corresponding shape model for all remaining computations for that frame. End applications made possible by such an extension include the detection of toppled elderly persons in ambient assisted

living (AAL) contexts (*cf.* Cucchiara *et al.* [59]).

Integration of a full body pose estimation approach, though more computationally demanding and therefore more suited to post-processing than real time application in the current state, could constitute an alternative to the procedure described above. The report by Amin *et al.* [4] provides a detailed overview of a full body pose estimation approach applied to images from a database of cooking activities (*cf.* Rohrbach *et al.* [235]), recorded from a supracranial observation perspective, and consequently similar in that regard to the images provided by the camera system described in Chapter 2 on page 11. Figure 5.3 depicts the full body model applied to images obtained from the camera system described in this thesis.



FIGURE 5.3: Example pictures for full body pose estimation on images obtained from the camera system described in Chapter 2, depicting the annotated full body poses used to train the estimator. The poses consist of 15 points with a combined 30 DOF: head, collarbone, left and right shoulders, elbows and wrists, lower torso, left and right hips, knees and ankles.

Further directions in the development of applications for the camera system have already been mentioned in Section 4.4 on page 132. To recapitulate, the ultimate goal for the system would be completely integrated smart scene observation, combining pedestrian tracking with body pose, gesture recognition and facial image analysis, provided the current observation perspective allows for these application, and using the acquired data to classify states and activities of the observed targets (*cf.* Tenorth [269] for high-level reasoning with actions and plans, Rohrbach *et al.* [236] for activity scripts).

Barring catastrophes on a global scale, it is beyond the author's doubt that the miniaturization-fueled trends mentioned in the introduction will continue in the coming decades, providing the basis for the development for ever more

sophisticated surveillance and monitoring systems. It stands to hope that these systems will be employed in an ethical manner, and that the benefits contributed to society will outweigh the dangers inherent in the capacities provided by such systems. One way or the other, the author remains confident that the coming years will bring exciting opportunities for scientific and public discussion regarding the topics touched upon in this thesis, and looks forward to the developments of the upcoming decades.

Appendix A

Publications

The following is a list of publications referencing the work conducted by the author leading up to this thesis, with a brief description of the relevant content each.

- Brščić *et al.* [34] contains a technical description of the first phase of the camera system setup at the CCRL, providing an insight into the research environment that the work described in this thesis was embedded into.
- Lenz *et al.* [175] contains a description of the system setup, software architecture and tracking algorithm.
- Eggers *et al.* [74] contains an updated technical description of the camera system, and details about the camera calibration, as well as an accuracy comparison to state-of-the art multi-camera calibration approaches (*cf.* Section 2.7 on page 42).
- Nierhoff *et al.* [204] contains a brief technical description of the camera system, and illustrates its application within a scenario where a mobile robot is used to change tires on a car. The camera system is used to track a human coordinating with the robot.
- Eggers *et al.* [73] contains an updated and extended version of the work presented in [74], comprising performance and stability evaluations.

Appendix B

Additional Tables

Style	Used for	Example
Italic	Scalars, events and various others	N
Bold	Vectors	\mathbf{x}
Sans serif	Matrices	\mathbf{C}
Fraktur	Tuples, sets, and sequences	\mathfrak{E}
Calligraphy	Complex objects (<i>e.g.</i> images, regions)	\mathcal{I}

TABLE B.1: Use of different typesetting in mathematical formulae.

Style	Used for	Example
Small capitals	Company and product names	MVTEC
Italic	Foreign words and abbreviations, emphasis	<i>et al.</i>

TABLE B.2: Use of different typesetting in text.

No.	x	y	z	φ	θ	ψ
1	-3.367 m	3.599 m	0.143 m	358.03°	0.25°	359.89°
2	-3.443 m	2.518 m	0.043 m	358.54°	359.37°	357.47°
3	-3.343 m	1.223 m	0.097 m	359.13°	359.08°	359.67°
4	-3.447 m	0.01 m	0.023 m	356.88°	1.22°	358.19°
5	-3.448 m	-1.363 m	0.101 m	359.46°	1.09°	0.48°
6	-3.408 m	-2.501 m	0.029 m	358.13°	0.3°	356.96°
7	-3.514 m	-3.703 m	0.115 m	0.17°	2.04°	359.28°
8	-3.421 m	-4.919 m	0.102 m	1.08°	0.57°	0.1°
9	-1.73 m	3.522 m	0.113 m	358.64°	0.41°	2.41°
10	-1.729 m	2.51 m	0.021 m	359.24°	0.74°	357.03°
11	-1.763 m	1.018 m	0.098 m	0.05°	1.21°	2.36°
12	-1.718 m	0.001 m	-0.0 m	357.59°	1.83°	359.75°
13	-1.755 m	-1.274 m	0.104 m	358.85°	2.94°	358.91°
14	-1.725 m	-2.506 m	0.005 m	0.15°	1.49°	1.08°
15	-1.704 m	-3.764 m	0.11 m	0.31°	3.02°	356.57°
16	-1.721 m	-4.924 m	0.083 m	0.41°	359.55°	1.23°
17	-0.894 m	-1.801 m	-0.005 m	359.12°	1.07°	359.47°
18	-1.099 m	-4.267 m	0.1 m	359.66°	1.49°	1.14°
19	-0.014 m	3.607 m	0.078 m	359.89°	358.95°	359.96°
20	-0.003 m	2.512 m	0.015 m	357.88°	0.07°	357.9°

TABLE B.3: Poses of the CCD sensors of the first set of 20 cameras used in the setup in a Cartesian world coordinate system, computed according to the global multi-camera calibration procedure described in Section 2.7 on page 42. See Table B.4 on the facing page for the poses of the remaining cameras, including the reference camera, Camera 22.

No.	x	y	z	α	β	γ
21	-0.023 m	1.16 m	0.063 m	0.28°	1.55°	0.11°
22	0.0 m	0.0 m	0.0 m	0.0°	0.0°	0.0°
23	0.936 m	-1.506 m	-0.013 m	1.68°	358.08°	2.34°
24	0.806 m	-4.207 m	-0.057 m	357.47°	0.25°	359.18°
25	1.76 m	3.538 m	0.091 m	0.76°	0.77°	358.96°
26	1.72 m	2.509 m	-0.012 m	359.32°	0.82°	358.77°
27	1.749 m	1.115 m	0.08 m	0.64°	1.98°	1.26°
28	1.792 m	-0.017 m	-0.02 m	0.13°	1.53°	0.02°
29	1.759 m	-1.514 m	0.082 m	0.45°	1.9°	1.89°
30	1.777 m	-2.536 m	-0.003 m	359.77°	359.74°	357.86°
31	1.746 m	-3.775 m	0.095 m	358.99°	1.73°	359.13°
32	1.69 m	-4.939 m	0.082 m	0.27°	0.22°	0.32°
33	3.449 m	3.5 m	0.072 m	1.48°	359.71°	359.64°
34	3.418 m	2.493 m	-0.018 m	359.62°	0.3°	359.47°
35	3.401 m	1.087 m	0.067 m	0.82°	0.1°	0.56°
36	3.413 m	-0.019 m	-0.021 m	358.04°	1.44°	358.55°
37	3.375 m	-1.531 m	0.065 m	358.93°	2.67°	358.08°
38	3.409 m	-2.539 m	-0.011 m	0.75°	0.49°	358.58°
39	3.42 m	-3.802 m	0.089 m	359.1°	359.18°	359.91°
40	3.389 m	-4.955 m	0.078 m	359.51°	1.02°	0.57°

TABLE B.4: Poses of the CCD sensors of the second set of 20 cameras used in the setup in a Cartesian world coordinate system, computed according to the global multi-camera calibration procedure described in Section 2.7 on page 42. Note, that the origin coordinate system is the sensor of Camera 22. See Table B.3 on page 148 for the poses of the remaining cameras.

Des.	N_i	Dur.	\mathfrak{T}	Cut	N_V	Desc.	Dir.
\mathfrak{S}_0	571	20.3 s	None		≥ 2	Empty sequence for background training	N/A
\mathfrak{S}_1	577	20.5 s	\mathcal{P}_1		≥ 2	Target walking, crossing multiple views in single direction	$\approx \uparrow$
\mathfrak{S}_2	569	20.3 s	\mathcal{P}_1		≥ 2	— " —	$\approx \uparrow$
\mathfrak{S}_3	583	20.8 s	\mathcal{P}_1		≥ 2	— " —	$\approx \uparrow$
\mathfrak{S}_4	587	20.9 s	\mathcal{P}_1		≥ 2	— " —	$\approx \uparrow$
\mathfrak{S}_5	578	20.6 s	\mathcal{P}_2		≥ 2	— " —	$\approx \uparrow$
\mathfrak{S}_6	568	20.2 s	\mathcal{P}_2		≥ 2	— " —	$\approx \uparrow$
\mathfrak{S}_7	581	20.7 s	\mathcal{P}_2		≥ 2	— " —	$\approx \uparrow$
\mathfrak{S}_8	574	20.4 s	\mathcal{P}_2		≥ 2	— " —	$\approx \uparrow$
\mathfrak{S}_9	588	20.9 s	\mathcal{P}_3		≥ 2	— " —	$\approx \uparrow$
\mathfrak{S}_{10}	565	20.1 s	\mathcal{P}_3		≥ 2	— " —	$\approx \uparrow$
\mathfrak{S}_{11}	579	20.6 s	\mathcal{P}_3		≥ 2	— " —	$\approx \uparrow$
\mathfrak{S}_{12}	584	20.8 s	\mathcal{P}_3		≥ 2	— " —	$\approx \uparrow$
\mathfrak{S}_{13}	566	20.2 s	$\mathcal{P}_1, \mathcal{P}_2$		≥ 2	Targets crossing the area simultaneously in opposite directions	$\approx \updownarrow$
\mathfrak{S}_{14}	591	21.0 s	$\mathcal{P}_1, \mathcal{P}_2$		≥ 2	— " —	$\approx \updownarrow$
\mathfrak{S}_{15}	572	20.4 s	$\mathcal{P}_1, \mathcal{P}_2$		≥ 2	— " —	$\approx \updownarrow$
\mathfrak{S}_{16}	578	20.6 s	$\mathcal{P}_1, \mathcal{P}_2$		≥ 2	— " —	$\approx \updownarrow$
\mathfrak{S}_{17}	577	20.5 s	$\mathcal{P}_1, \mathcal{P}_3$		≥ 2	— " —	$\approx \updownarrow$
\mathfrak{S}_{18}	561	20.0 s	$\mathcal{P}_1, \mathcal{P}_3$		≥ 2	— " —	$\approx \updownarrow$

TABLE B.5: Complete list of the image sequences used during the evaluation procedures for detection, tracking, and appearance modeling, part I. N_i denotes the number of images in the sequence, \mathfrak{T} denotes the targets in the sequence (*cf.* Table 2.9 on page 64 for details), and N_V denotes the number of views.

Des.	N_i	Dur.	\mathfrak{T}	Cut	N_V	Desc.	Dir.
\mathfrak{S}_{19}	579	20.6 s	$\mathcal{P}_1, \mathcal{P}_3$		≥ 2	Targets crossing the area simultaneously in opposite directions	$\approx \uparrow\downarrow$
\mathfrak{S}_{20}	569	20.3 s	$\mathcal{P}_1, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\downarrow$
\mathfrak{S}_{21}	565	20.1 s	$\mathcal{P}_2, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\downarrow$
\mathfrak{S}_{22}	582	20.7 s	$\mathcal{P}_2, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\downarrow$
\mathfrak{S}_{23}	560	19.9 s	$\mathcal{P}_2, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\downarrow$
\mathfrak{S}_{24}	594	21.2 s	$\mathcal{P}_2, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\downarrow$
\mathfrak{S}_{25}	599	21.3 s	$\mathcal{P}_1, \mathcal{P}_2$		≥ 2	Targets crossing the area simultaneously in the same direction (side-by-side)	$\approx \uparrow\uparrow$
\mathfrak{S}_{25}	600	21.4 s	$\mathcal{P}_1, \mathcal{P}_2$		≥ 2	— ” —	$\approx \uparrow\uparrow$
\mathfrak{S}_{26}	599	21.3 s	$\mathcal{P}_1, \mathcal{P}_2$		≥ 2	— ” —	$\approx \uparrow\uparrow$
\mathfrak{S}_{27}	600	21.4 s	$\mathcal{P}_1, \mathcal{P}_2$		≥ 2	— ” —	$\approx \uparrow\uparrow$
\mathfrak{S}_{28}	599	21.3 s	$\mathcal{P}_1, \mathcal{P}_2$		≥ 2	— ” —	$\approx \uparrow\uparrow$
\mathfrak{S}_{29}	589	21.0 s	$\mathcal{P}_1, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\uparrow$
\mathfrak{S}_{30}	583	20.8 s	$\mathcal{P}_1, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\uparrow$
\mathfrak{S}_{31}	579	20.6 s	$\mathcal{P}_1, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\uparrow$
\mathfrak{S}_{32}	591	21.0 s	$\mathcal{P}_1, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\uparrow$
\mathfrak{S}_{33}	581	20.7 s	$\mathcal{P}_2, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\uparrow$
\mathfrak{S}_{34}	588	20.9 s	$\mathcal{P}_2, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\uparrow$
\mathfrak{S}_{35}	573	20.4 s	$\mathcal{P}_2, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\uparrow$
\mathfrak{S}_{36}	592	21.1 s	$\mathcal{P}_2, \mathcal{P}_3$		≥ 2	— ” —	$\approx \uparrow\uparrow$

TABLE B.6: Complete list of the image sequences used during the evaluation procedures for detection, tracking, and appearance modeling, part II. N_i denotes the number of images in the sequence, \mathfrak{T} denotes the targets in the sequence (*cf.* Table 2.9 on page 64 for details), and N_V denotes the number of views.

Des.	N_i	Dur.	\mathfrak{T}	Cut	N_V	Desc.	Dir.
\mathfrak{S}_{37}	3000	106.8 s	\mathcal{P}_1		1	Target walking in ellipsoid path in a single view	$\approx \bigcirc$
\mathfrak{S}_{38}	3000	106.8 s	\mathcal{P}_2		1	— ” —	$\approx \bigcirc$
\mathfrak{S}_{39}	3000	106.8 s	\mathcal{P}_3		1	— ” —	$\approx \bigcirc$
\mathfrak{S}_{40}	3000	106.8 s	\mathcal{P}_1		1	Target walking in lemniscatoid path in single view	$\approx \infty$
\mathfrak{S}_{41}	3000	106.8 s	\mathcal{P}_2		1	— ” —	$\approx \infty$
\mathfrak{S}_{42}	1000	35.7 s	\mathcal{P}_1	✓	1	Looped normalized sequence; from \mathfrak{S}_{37}	$\approx \bigcirc$
\mathfrak{S}_{43}	1000	35.7 s	\mathcal{P}_2	✓	1	— ” —; from \mathfrak{S}_{38}	$\approx \bigcirc$
\mathfrak{S}_{44}	1000	35.7 s	\mathcal{P}_3	✓	1	— ” —; from \mathfrak{S}_{39}	$\approx \bigcirc$
\mathfrak{S}_{45}	1000	35.7 s	\mathcal{P}_1	✓	1	— ” —; from \mathfrak{S}_{40}	$\approx \infty$
\mathfrak{S}_{46}	1000	35.7 s	\mathcal{P}_2	✓	1	— ” —; from \mathfrak{S}_{41}	$\approx \infty$
\mathfrak{S}_{47}	3100	110.7 s	$\mathcal{P}_{1\dots3}$	✓	1	Training sequence; from $\{\mathfrak{S}_0, \mathfrak{S}_{42} \dots \mathfrak{S}_{46}\}$	$\approx \bigcirc$
\mathfrak{S}_{48}	2500	89.3 s	$\mathcal{P}_{1\dots3}$	✓	1	Test sequence; from $\{\mathfrak{S}_0, \mathfrak{S}_1 \dots \mathfrak{S}_{24}\}$	$\approx \updownarrow$
$\mathfrak{S}_{49} \dots \mathfrak{S}_{69}$	10	0.4 s	\mathcal{P}_2		1	Single target standing/walking in the camera FOV	N/A
$\mathfrak{S}_{70} \dots \mathfrak{S}_{74}$	10	0.4 s	$\mathcal{P}_{1\dots3}$		1	Two targets standing in the camera FOV	N/A
$\mathfrak{S}_{75} \dots \mathfrak{S}_{80}$	10	0.4 s	$\mathcal{P}_{1\dots3}$		1	Three targets standing in the camera FOV	N/A

TABLE B.7: Complete list of the image sequences used during the evaluation procedures for detection, tracking, and appearance modeling, part III. N_i denotes the number of images in the sequence, \mathfrak{T} denotes the targets in the sequence (*cf.* Table 2.9 on page 64 for details), and N_V denotes the number of views.

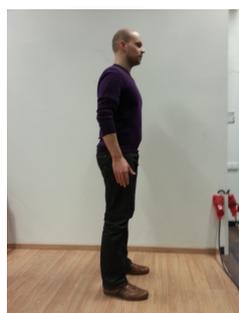
Perspective	Description	Example
Supracranial	Viewed from the top, above the head facing downwards.	
Anterior	Viewed from the front.	
Posterior	Viewed from behind.	
Lateral	Viewed from the side.	

TABLE B.8: Terminology for camera perspectives with regard to pedestrians, derived from terminology commonly used in human anatomy [182, pp. 12–19]. Note, that the camera system described in this thesis primarily delivers images from the supracranial perspective.

Name	Version	Author	Pub.	Used in
OpenCV	2.0–2.3	WillowGarage	[29]	Image processing
HALCON	9.0–11.0.1	MVTec	[72, 255]	Image processing
ICE	3.4	ZeroC	[116]	Middleware
WEKA	3.7.1	U. of Waikato	[109]	Machine learning
ntpd	3–4	D. Mills	[191]	Synchronization
Ubuntu	12.04 LTS	Canonical	[118]	Operating system
Boost	1.39–1.46.1	Boost	[146]	Various
KogMo-RTDB	N/A	M. Goebel	[98]	Real-time processing
OpenTL	0.8–0.9	TU Munich	[211]	Image processing

TABLE B.9: Information on the software packages and libraries used in in the implementations of the different parts of this dissertation.

Appendix C

Glossaries of Terms

List of Operators

$x \approx y$ x (left side) is approximately equal to y (right side). Operator is commutative.

$x \equiv y$ x (left side) is defined as y (right side); x is equal to y by definition. Operator is non-commutative.

$H(R)$ The histogram operator, which produces a color histogram from an image region (*cf.* Equation (3.12) on page 93).

\bar{x} The arithmetic mean of x .

$\|\mathbf{x}\|$ Vector norm of $\mathbf{x} = (x_1, \dots, x_n)$, $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^n x_i^2}$.

$\hat{\mathbf{x}}$ Normalization of vector \mathbf{x} , $\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$.

$x \cdot y$ The product of two scalar values x and y . Operator is commutative.

$\mathbf{x} \cdot \mathbf{y}$ The inner product (scalar product) of two vectors \mathbf{x} and \mathbf{y} . Operator is commutative.

$\mathbf{x} \times \mathbf{y}$ The cross product (vector product) of two vectors \mathbf{x} and \mathbf{y} . Operator is anticommutative.

$\lfloor x \rfloor$ x rounded to the nearest integer.

M^T The transposition of matrix M .

List of Symbols

- \mathcal{A} The visual appearance of a reflective object, termed *appearance* for short.
- \mathcal{A}^c The appearance of an object as composed of the appearances of its parts, *i.e.* the *composite appearance*.
- \mathcal{A}^0 The appearance of an object at detection, *i.e.* the *initial appearance*.
- \mathcal{A}^p The appearance of an object part, *i.e.* a *partial appearance*.
- C^h The color distribution of the head of a tracked pedestrian, modeled as a $k \times m \times n$ bin color histogram in HSI color space. See the entry for C for details.
- C^l The color distribution of the legs of a tracked pedestrian, modeled as a $k \times m \times n$ bin color histogram in HSI color space. See the entry for C for details.
- C^t The color distribution of the torso of a tracked pedestrian, modeled as a $k \times m \times n$ bin color histogram in HSI color space. See the entry for C for details.
- C^p The color distribution of a part of a target, modeled as a $n \times m \times k$ bin color histogram in HSI color space, so that $C = \sum_{i=1}^I w_i \times C^p$. See the entry for C for details on the color histogram.
- C The color distribution of a tracking target, modeled as a $k \times m \times n$ bin color histogram (*i.e.* discretization) in HSI color space. Each bin contains the number of pixels in the respective sub-range of HSI. From a computational perspective, C is represented as a $k \times m \times n$ matrix. For the dynamic appearance model proposed in Chapter 3 on page 79, $C = w_1 C^h + w_2 C^t + w_3 C^l$
- w A *weight*, which is a scalar that represents the share of a partial appearance \mathcal{A}^p in the total appearance \mathcal{A} . Regarding indices, w_i denotes the weight of the i -th partial appearance \mathcal{A}^p_i for a single frame, while $w_{(i,k)}$ denotes the weight of partial appearance i at frame k for objects in image sequences.
- \mathcal{C} A *camera*, whose properties can be mathematically described by the camera parameters \mathfrak{P} for single images \mathcal{I} . Regarding indices, \mathcal{C}_i denotes the i -th numbered *camera* in the CCRL setup.
- e^d The *reprojection error*, $e^d = \|\mathbf{p} - \Pi(\mathfrak{C} \cdot \mathbf{P} \cdot \mathbf{x}, \mathfrak{J})\|$

e^r The *RMS error* for the reprojection:

$$e^r = \sqrt{\frac{1}{M \sum_{i=1}^N \sum_{k=1}^K v_{ik}}} e^d$$

\mathfrak{D} The image distortion caused by the camera lens, modeled as non-linear radial and decentering distortion. It is described by the triple of radial distortion parameters $K_i \forall i \in \{1, 2, 3\}$, tangential distortion parameters $P_i \forall i \in \{1, 2\}$ and principal point (*i.e.* distortion center) \mathbf{C} .

\mathbf{R}_s The 3 DOF rotation of a sensor (*i.e.* camera) within the world coordinate system, represented by the rotation matrix.

\mathbf{T}_s The 3 DOF translation of a sensor (*i.e.* camera) within the world coordinate system.

\mathbf{C} The image center, *i.e.* the principal point where the principal axis intersects with the image plane.

f The *focal length* of a camera.

α The *angular field of view*, *i.e.* the angle between the edges of vision of the camera. It is determined by the photographic objective of the camera.

\mathfrak{P} The pair of *camera parameters*, consisting of extrinsic parameters and intrinsic parameters, $\mathfrak{P} = (\mathfrak{E}, \mathfrak{I})$.

\mathfrak{E} The *extrinsic camera parameters*, describing the world pose of the camera. They are defined as the pair of translation and rotation of the camera, $\mathfrak{E} = (\mathbf{T}_s, \mathbf{R}_s)$

\mathfrak{I} The *intrinsic camera parameters*, describing the optical properties of the camera. They are defined as the 4-tuple of focal length f , image format \mathfrak{F} , principal point \mathbf{C} and distortion \mathfrak{D} , $\mathfrak{I} = (f, \mathfrak{F}, \mathbf{C}, \mathfrak{D})$

Π A pinhole camera perspective projection, which is represented by the camera matrix:

$$\Pi = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- \mathfrak{V} A *view*, which is an abstraction from a camera for a single image, and is denoted as a triple consisting of image, extrinsic parameters, and intrinsic parameters, $\mathfrak{V} = (\mathcal{I}, \mathfrak{E}, \mathfrak{I})$
- \mathbf{H} A *color histogram* for the visible surface of an object. The histogram consists of $k \times m \times n$ bins in HSI color space. Each bin contains the number of pixels in the respective sub-range of HSI.
- b A *bin* of a color histogram \mathbf{H} .
- \mathfrak{H} The set of $k \times m \times n$ bin color histograms. With histogram addition, \mathfrak{H} forms a vector space over \mathbb{R} .
- $\mathbf{0}$ The *identity element* of \mathfrak{H} with respect to addition.
- A The disjoint intervals each color channel is partitioned into to create the color histogram.
- $\hat{\mathbf{H}}$ A *normalized color histogram* for the visible surface of an object, where each bin b is weighted by the total number of entries in all bins, so that $\|\hat{\mathbf{H}}\| = 1$.
- \mathbf{H} The *color histogram operator*. $\mathbf{H}(\mathcal{R}) = \mathbf{H}$
- D_B The *Bhattacharyya distance*, which provides a measure for the similarity of two probability distributions p and q . $D_B(p, q) = -\ln \left(\sum_{x \in X} \sqrt{p(x)q(x)} \right)$.
- M The total *number of classes* for a classification task.
- N The total *number of instances* for a class.
- N The total *number of instances* for a classification experiment.
- $f_a(C)$ The number of *true positives*, *i.e.* the absolute frequency of correctly classified instances of a class.
- $f_r(C)$ The *classification rate*, which is the relative frequency of correctly classified instances.
- $f_r(I)$ The *classification error rate*, which is the relative frequency of incorrectly classified instances.
- $f_r(C)$ The *recall rate*, which provides a measure for the performance of a classification task. $f_r(C) = \frac{f_a(C)}{N}$.
- N_F The *failure count*, which is the absolute count of failed tries (or instances) in an experiment.
- N_S The *success count*, which is the absolute count of successful tries (or instances) in an experiment.

N_t	The <i>total count</i> , which is the absolute count of tries (or instances) in an experiment.
N_v	The <i>transition count</i> , which is the amount of view transitions for a single of track in an experiment.
$\overline{N_v}$	The <i>mean transition count</i> , which is the mean amount of view transitions for a single track over multiple experiments.
d_a	The <i>absolute tracking error</i> , <i>i.e.</i> the distance between the target ground truth and the target candidate in the image plane, measured in pixels.
$d_{\bar{r}}$	The <i>relative tracking error</i> , <i>i.e.</i> the absolute tracking error d_a , scaled by the average diameter of the target.
d_r	The <i>relative tracking error</i> , <i>i.e.</i> the absolute tracking error d_a , scaled by the current diameter of the target.
d_m	The <i>maximum acceptable absolute tracking error</i> , <i>i.e.</i> the maximum distance between the target ground truth and the target candidate that is accepted without the track being considered lost.
C	The event that a certain instance was <i>classified correctly</i> during a classification experiment.
F	The event of <i>failure</i> , of a certain experiment or try.
I	The event that a certain instance was <i>classified incorrectly</i> during a classification experiment.
S	The event of <i>success</i> , of a certain experiment or try.
f	The <i>frequency</i> function.
f_a	The <i>absolute frequency</i> function.
f_r	The <i>relative frequency</i> function, also called <i>a-posteriori</i> probability.
P	The <i>probability</i> function.
$P(S_t)$	The <i>transition success probability</i> , which is the probability of the event that a single view transition is successful.
$f_r(F)$	The <i>failure rate</i> , is the relative frequency of erroneous tries (or instances) occurring in an experiment. $f_r(F) = \frac{N_F}{N_t} = 1 - f_r(S)$
$f_r(S)$	The <i>success rate</i> , is the relative frequency of successful tries (or instances) occurring in an experiment. $f_r(S) = \frac{N_S}{N_t} = 1 - f_r(F)$
\mathbf{t}_c	The target <i>candidate</i> , which is the highest probability hypothesis provided by the MCMC algorithm. It is represented by a point in the image plane: $\mathbf{t}_c = (x, y)^T$.

- \mathbf{t}_g The target *ground truth*, which is a point in the image plane: $\mathbf{t}_g = (x, y)^T$.
- \mathbf{O} The *origin* of a coordinate system.
- \mathbf{O}^m The origin of the model coordinate system.
- \mathbf{O}^w The origin of the world coordinate system.
- \mathbf{p} A two-dimensional point in an image, specified in image coordinates, $\mathbf{p} = (x, y)^T$. For $x \in \mathbb{N} \wedge y \in \mathbb{N}$, \mathbf{p} corresponds to a *pixel*. Otherwise, it specifies a point with subpixel accuracy.
- \mathbf{x} A 2D point in the local coordinate system of the calibration object, $\mathbf{x} = (x, y)^T$
- \mathbf{X} A *three-dimensional point*, $\mathbf{X} = (x, y, z)^T$
- \mathbf{X}^c A three-dimensional point, specified in a camera coordinate system.
- \mathbf{X}^w A three-dimensional point, specified in the world coordinate system.
- \mathbf{P} The pose of an object, which is represented by a 3×4 homogeneous transformation matrix:

$$\mathbf{P} = \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}^T & 1 \end{pmatrix}$$

- \mathbf{R} The *rotation* of an object within the world coordinate system, which has 3 DOF. It is represented by the 3×3 rotation matrix:

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where φ , θ and ψ are the Euler angles with extrinsic *z-x-z* rotation (*i.e.* around the original axes).

- η The 1 DOF *heading* of a target, represented by the rotation around its vertical axis, which is perpendicular to the floor plane. It is equivalent to the Euler angle φ .
- \mathbf{T} The 3 DOF translation of an object within the world coordinate system.
- \mathbf{T}_t^W The 3 DOF translation of a tracking target (*e.g.* a pedestrian) within the world coordinate system.

\mathbf{T}_t^F	The 2 DOF translation of a tracking target (<i>e.g.</i> a pedestrian) within the floor plane F .
a	The aspect ratio of an image, $\frac{n_x}{n_y}$
n_x	The horizontal resolution of a digital image, <i>i.e.</i> the number of pixels in the horizontal direction.
n_y	The vertical resolution of a digital image, <i>i.e.</i> the number of pixels in the vertical direction.
n_p	The amount of data used to encode each pixel of an image, measured in bits.
\mathfrak{F}	The image format of a digital image, consisting of the pair of horizontal and vertical resolution, $\mathfrak{F} = (n_x, n_y)$.
\mathcal{I}	A digital <i>image</i> .
v	The <i>value</i> of a pixel for a certain channel. $v \in [0, 1]$
\mathcal{R}	A <i>region</i> within a digital image \mathcal{I} , sometimes also called <i>region of interest</i> .
\mathfrak{S}	A <i>sequence</i> of N digital images, $\mathfrak{S} = (\mathcal{I}_1, \dots, \mathcal{I}_n)$, where $t(\mathcal{I}_{i+1}) > t(\mathcal{I}_i)$.
f_i	The <i>temporal frequency</i> of images delivered by an image source, such as a camera. Sometimes referred to as <i>frame rate</i> or frames per second (FPS).
R	The <i>data rate</i> transmitted on a network, $R = \frac{s}{t}$.
s	The size of an amount of data, measured in bits.
s_e	The size of an Ethernet packet.
s_m	The size of the MTU
s_p	The <i>payload</i> of a GVSP packet, <i>i.e.</i> the size of the packet less the size of the headers.
v	The boolean <i>visibility</i> of a point or object from a camera \mathcal{C} , $v = 1$ for all points that are visible from that camera, 0 otherwise. For objects, <i>e.g.</i> the calibration plate or pedestrians, $v = 1$ if no part of the object lies outside of the camera FOV.
x_t	The extension of the area of observation in x -direction.
y_t	The extension of the area of observation in y -direction.

n_c	The total <i>number of cameras</i> \mathcal{C}_i constituting the camera setup.
V	The <i>availability</i> of a hard- or software component, $V = \frac{t^u}{t^d+t^u}$.
t^d	The <i>downtime</i> of a hard- or software component, <i>i.e.</i> the time during which the component is not active or working.
t^u	The <i>uptime</i> of a hard- or software component, <i>i.e.</i> the time during which the component is active or working.
D	The <i>coverage redundancy</i> for the area of observation, <i>i.e.</i> the difference of the ratio of the combined FOVs of all cameras \mathcal{C}_i to the size of the area of observation on a reference plane.
ρ_i	The sampling density of a camera \mathcal{C}_i at reference height h_o , <i>i.e.</i> the ratio of pixels to FOV area.
$\bar{\rho}$	The mean sampling density of the camera system at h_o , <i>i.e.</i> the ratio of pixels to combined FOV area.
d_x	The distance covered by a camera in the primary direction, <i>i.e.</i> the x -extension of the camera's FOV.
d_y	The distance covered by a camera in the secondary direction, <i>i.e.</i> the y -extension of the camera's FOV.
\mathbf{F}	The 3 DOF normal vector of the <i>floor plane</i> .
F	The 4 DOF <i>floor plane</i> , represented by its normalized normal vector $\hat{\mathbf{F}}$ and distance d from the world origin, in the Hesse normal form:
	$\hat{\mathbf{F}} \cdot \mathbf{r} - d = 0$
	.
h_c	The distance between a camera \mathcal{C}_i and the floor plane, <i>i.e.</i> the <i>camera height</i> .
\bar{h}_c	The <i>mean distance</i> between camera and the floor plane, for all \mathcal{C}_i
h_o	The distance between the reference plane (<i>i.e.</i> observation level) and the floor plane, <i>i.e.</i> the observation height.
P_i^p	Visible <i>area of the reprojection</i> of the surface of a shape model (body) part into the image plane, <i>cf.</i> P .
h_t	The total height of the generalized cylinder model, from base to top.
h_h	The height of the head segment the generalized cylinder model.
h_l	The height of the leg segment of the generalized cylinder model.
h_u	The height of the upper body segment of the generalized cylinder model.

d_v	The <i>approximate vertex distance</i> , <i>i.e.</i> the approximate distance between vertices in the mesh. It is used to determine n_e , the exact geometry of the mesh, and consequently n_f and n_v .
n_e	The <i>number of edges</i> for the polygon used to approximate circles in the mesh shape model.
n_f	The <i>number of faces</i> for the polygon mesh shape model.
n_v	The <i>number of vertices</i> for the polygon mesh shape model.
\mathbf{V}	A shape model <i>vertex</i> , represented by its 3-DOF translation.
r_b	The radius of the generalized cylinder model at $h = 0$.
r_t	The radius of the generalized cylinder model at $h = h_t$.
r_h	The radius of the generalized cylinder model at $h = h_l$.
r_s	The radius of the generalized cylinder model at $h = h_t - h_h$.
P	Visible <i>area of the reprojection</i> of the entire shape model surface of a target into the image plane. The area is measured in pixels.
P^Δ	Visible <i>area of the reprojection</i> of a single face of the polygon mesh model into the image plane.
Q	The <i>proposal distribution</i> for the MCMC tracking algorithm.
s_h	A <i>hypothesis</i> , <i>i.e.</i> a proposed state of the MCMC tracking algorithm.
s	The <i>state vector</i> of a tracking target t .
t	A <i>target</i> being tracked by the MCMC tracking algorithm.

List of further Terms

appearance refers to the properties of an object that can be visually observed (*cf.* Hunter and Harold [127]). The most important properties falling under these definitions are properties of the object surface (color) and shape. While in theory object surfaces might have transmissive as well as reflective properties, for the purpose of this thesis, only reflective properties are considered due to the nature of the objects being modeled. It should be noted, that due to the optical sense relying on light reflected by the object, appearance properties are subject to change upon variations in illumination conditions.

appearance model refers to any approach to modeling the appearance of an object, *e.g.* using color or brightness statistics of its digital image. In the sense of employing appearance models in tracking, a *static appearance model* refers to an appearance model that does not vary over time, the opposite of which is termed an *adaptive appearance model* here.

area of observation denotes the unification of the FOVs of all cameras \mathcal{C}_i in a camera system, and consequently, the area that can be observed by the camera system.

floor plane denotes a plane which represents the floor of the area of observation. Since for most indoor environments, the floor of the area of observation can be expected to be planar, it is an important special case (or approximation) of the floor surface.

floor surface denotes a two-dimensional topological manifold that represents the floor of the area of observation. In that, it is the generalization of the floor plane.

frame denotes a single iteration of an image processing system, corresponding to the processing of one image \mathcal{I} .

image position of an object denotes the position of the projection of the object in image space.

image space denotes the set of all points that a camera can project to (*i.e.*, the *image points*), forming a surface. For a pinhole camera model, all of these points lie in the *image plane*. The distinction from the world space is relevant *e.g.* when evaluating the precision of position predictions of objects. The position of a point in image space is denoted using the *image coordinate system*, and distances are measured in *pixels*.

observation perspective denotes the perspective the camera has on an object. This perspective can be represented by the combination of target translation, vertical orientation, and camera pose. For targets with a defined vertical and horizontal orientation, the observation perspective is described in this thesis as *lateral*, *anterior*, *posterior* or *supracranial* (*cf.* Table B.8 on page 153).

anterior perspective denotes an observation perspective where the camera is located in front of the target, specifically for pedestrian targets.

lateral perspective denotes an observation perspective where the camera is located besides the target, specifically for pedestrian targets. In the strictest sense of the term, the camera is next to the target, and the camera principal axis and the object lateral axis are aligned. For relaxed use of the term, the angle between the camera principal axis and the lateral axis of the target is smaller than the angle between the camera principal axis and the vertical axis of the target and smaller than the angle between the camera principal axis and the horizontal axis of the target.

posterior perspective denotes an observation perspective where the camera is located behind of the target, specifically for pedestrian targets.

supracranial perspective denotes an observation perspective where the camera is located above the target, specifically for pedestrian targets. In the strictest sense of the term, the camera is located directly above the target (*i.e.*, above the head) and the camera principal axis and object vertical axis are aligned. For relaxed use of the term, the angle between the camera principal axis and the vertical axis of the target is smaller than the angle between the camera principal axis and the horizontal axis of the target and smaller than the angle between the camera principal axis and the lateral axis of the target.

observation plane denotes the planar special case (or approximation) of the observation surface.

observation surface denotes a two-dimensional topological manifold which is parallel to the floor surface, where the distance between observation surface and floor surface is the observation height h_o . The observation surface is used to describe the surface in which the objects that are being observed are positioned. Thus, contrary to the floor surface, it does not represent a real surface in the physical sense.

pedestrian denotes a person who is in an upright position, either standing or ambulating. This corresponds to the general definition given by Gray *et al.* [102]. In contrast to common usage of this term in other context, for this thesis there is no implication of participation of a pedestrian in road traffic.

shape model refers to any approach to modeling the shape of an object, *e.g.* using 3D geometric primitives or polygon meshes. While a *rigid shape model* displays only a single configuration for an entire class of objects (*e.g.* pedestrians, cars), a *deformable shape model* (*e.g.* the CANDIDE-3 face model, *cf.* Ahlberg [1]) may display several variations, depending on the current state (*e.g.* walking, standing) or specific properties (*e.g.* height, size) of the object.

target area is short for *target area of observation* and denotes the intended area of observation. It may differ from the real area of observation due to constraints in the available number or possible placement of cameras.

track denotes the sequence of positions of an object over multiple sequential measurements. Unless specifically stated otherwise, for this thesis this is to mean the sequence of positions of an object in a series of timestamped images.

tracker denotes a program or program part which performs the process of visual tracking by realizing the tracking pipeline, *cf.* Section 2.3.3 on page 17.

tracking denotes the process of repeatedly determining the position of an object over several subsequent measurements. Unless otherwise indicated, tracking is used short for *visual tracking*, which restricts the measurements to camera images.

tracklet denotes a shorter track that is part of a longer track, *e.g.* the track of an object in a single field of view.

world position of an object denotes the physical position of the object in world space.

world space denotes the set of all physical points in the real world, of which – unless specifically stated otherwise – only those that can be projected by a camera are relevant in the context of this thesis. The position of a point in world space is denoted using the *world coordinate system* and distances are measured in the SI (sub-)multiples of the *meter*.

Bibliography

- [1] J. Ahlberg. *Candide-3. An updated parameterised face*. English. Technical report. Linköping: Linköping University, 2001 (cited on pages 141, 167).
- [2] J. G. Allen, R. Y. D. Xu, and J. S. Jin. “Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces”. English. In: *VIP '05 Proceedings of the Pan-Sydney area workshop on Visual information processing*. Edited by M. Piccardi, T. Hintz, S. He, M. L. Huang, and D. D. Feng. Volume 36. Sydney: Australian Computer Society, 2006, pages 3–7 (cited on page 16).
- [3] G. Alper. *Vision connectivity interfaces*. English. Technical report. Eindhoven: Adimec Advanced Image Systems b.v., 2011 (cited on page 33).
- [4] S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele. “Multi-view Pictorial Structures for 3D Human Pose Estimation”. English. In: *British Machine Vision Conference 2013, Bristol, September 2013. Proceedings*. Bristol: BMVA Press, 2013 (cited on pages 124, 142).
- [5] M. Andriluka, S. Roth, and B. Schiele. “People-tracking-by-detection and people-detection-by-tracking”. English. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. Edited by K. Boyer, M. Shah, and T. Syeda-Mahmood. Anchorage, AK: IEEE Computer Society, 2008, pages 1–8. ISBN: 978-1-4244-2242-5. DOI: 10.1109/CVPR.2008.4587583 (cited on page 15).
- [6] R. T. Apteker, J. A. Fisher, V. S. Kisimov, and H. Neishlos. “Video acceptability and frame rate”. In: *MultiMedia, IEEE 2.3 (1995)*, pages 32–40. DOI: 10.1109/93.410510 (cited on page 3).
- [7] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz. “The Nonholonomic Nature of Human Locomotion: A Modeling Study”. English. In: *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*. Edited

- by P. Dario and D. Meldrum. Pisa: IEEE, 2006, pages 158–163. DOI: 10.1109/BIOROB.2006.1639077 (cited on page 16).
- [8] M. Armstrong, A. Zisserman, and R. Hartley. “Self-calibration from image triplets”. English. In: *Computer Vision — ECCV '96. 4th European Conference on Computer Vision Cambridge, UK, April 15–18, 1996 Proceedings, Volume I*. Edited by B. Buxton and R. Cipolla. Volume 1064. Lecture Notes in Computer Science. Cambridge: Springer Berlin Heidelberg, 1996, pages 1–16. ISBN: 978-3-540-61122-6. DOI: 10.1007/BFb0015519 (cited on page 140).
- [9] U. Assarsson and T. Möller. *Optimized View Frustum Culling Algorithms*. English. Technical report. Chalmers: Chalmers University of Technology, 1999 (cited on page 103).
- [10] U. Assarsson and T. Möller. “Optimized View Frustum Culling Algorithms for Bounding Boxes”. English. In: *Journal of Graphics Tools* 5.1 (Jan. 2000), pages 9–22. ISSN: 1086-7651. DOI: 10.1080/10867651.2000.10487517 (cited on page 103).
- [11] A. O. Balan, L. Sigal, and M. J. Black. “A Quantitative Evaluation of Video-based 3D Person Tracking”. English. In: *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*. Beijing: IEEE Computer Society, 2005, pages 349–356 (cited on page 16).
- [12] J. Bandouch, F. Engstler, and M. Beetz. “Accurate Human Motion Capture Using an Ergonomics-Based Anthropometric Human Model”. English. In: *Articulated Motion and Deformable Objects. 5th International Conference, AMDO 2008, Port d’Andratx, Mallorca, Spain, July 9-11, 2008. Proceedings*. Edited by F. J. Perales and B. Fisher. Port d’Andratx: Springer Berlin Heidelberg, 2008, pages 248–258. DOI: 10.1007/978-3-540-70517-8_24 (cited on page 85).
- [13] Y. Bar-Shalom and E. Tse. “Tracking in a cluttered environment with probabilistic data association”. English. In: *Automatica* 11.5 (1975), pages 451–460. ISSN: 0005-1098. DOI: 10.1016/0005-1098(75)90021-7 (cited on page 21).
- [14] T. Bayes. “An Essay towards Solving a Problem in the Doctrine of Chances”. English. In: *Philosophical Transactions of the Royal Society of London* 53 (Jan. 1763), pages 370–418. ISSN: 0261-0523. DOI: 10.1098/rstl.1763.0053 (cited on pages 15, 18).

- [15] L. Bazzani, M. Cristani, A. Perina, M. Farenzena, and V. Murino. “Multiple-Shot Person Re-identification by HPE Signature”. English. In: *Pattern Recognition (ICPR), 2010 20th International Conference on*. Istanbul, Turkey: IEEE, Aug. 2010, pages 1413–1416. ISBN: 978-1-4244-7542-1. DOI: 10.1109/ICPR.2010.349 (cited on page 79).
- [16] M. Beetz, M. Tenorth, D. Jain, and J. Bandouch. “Towards automated models of activities of daily life”. English. In: *Technology and Disability 22.1-2* (June 2010), pages 27–40. ISSN: 1055-4181. DOI: 10.3233/TAD-2010-0285 (cited on page 124).
- [17] R. Beichel, H. Bischof, F. Leberl, and M. Sonka. “Robust active appearance models and their application to medical image analysis.” English. In: *Medical Imaging, IEEE Transactions on*. 24.9 (Sept. 2005), pages 1151–69. ISSN: 0278-0062. DOI: 10.1109/TMI.2005.853237 (cited on page 84).
- [18] A. N. Belbachir. *Smart Cameras*. English. Edited by A. N. Belbachir. Boston, MA: Springer US, 2010. ISBN: 978-1-4419-0954-1. DOI: 10.1007/978-1-4419-0953-4 (cited on page 21).
- [19] N. Bellotto, E. Sommerlade, B. Benfold, C. Bibby, I. Reid, D. Roth, C. Fernandez, L. Van Gool, J. Gonzalez, and C. Fern. “A Distributed Camera System for Multi-Resolution Surveillance”. English. In: *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*. Edited by J. Park and S. S. Bhattacharyya. Como: IEEE, Aug. 2009, pages 1–8. ISBN: 9781424446209. DOI: 10.1109/ICDSC.2009.5289413 (cited on page 3).
- [20] K. Bernardin and R. Stiefelhagen. “Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics”. English. In: *EURASIP Journal on Image and Video Processing 2008* (2008), pages 1–10. ISSN: 1687-5176. DOI: 10.1155/2008/246309 (cited on page 119).
- [21] A. K. Bhattacharyya. “On a measure of divergence between two statistical populations defined by their probability distributions”. English. In: *Bulletin of the Calcutta Mathematical Society* 35.1 (1943), pages 99–109 (cited on page 74).
- [22] S. Birchfield. “Elliptical head tracking using intensity gradients and color histograms”. English. In: *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. Edited by D. Terzopoulos and Y.-F. Wang. Santa Barbara, CA: IEEE Computer Society, 1998, pages 232–237. ISBN: 0-8186-8497-6. DOI: 10.1109/CVPR.1998.698614 (cited on page 83).

- [23] N. D. Bird, O. Masoud, N. P. Papanikolopoulos, and A. Isaacs. “Detection of loitering individuals in public transportation areas”. English. In: *Intelligent Transportation Systems, IEEE Transactions on* 6.2 (2005), pages 167–177. ISSN: 1524-9050. DOI: 10.1109/TITS.2005.848370 (cited on page 79).
- [24] R. Bodor, A. Drenner, P. Schrater, and N. Papanikolopoulos. “Optimal Camera Placement for Automated Surveillance Tasks”. English. In: *Journal of Intelligent and Robotic Systems* 50.3 (Nov. 2007), pages 257–295. ISSN: 0921-0296. DOI: 10.1007/s10846-007-9164-7 (cited on page 3).
- [25] R. Bodor, B. Jackson, and N. Papanikolopoulos. “Vision-based human tracking and activity recognition”. English. In: *11th Mediterranean Conference on Control and Automation - MED’03. Proceedings of*. Edited by F. L. Lewis, K. P. Valavanis, and S. Bogdan. Rhodes: IEEE Control Systems Society, 2003, pages 18–20. DOI: 10.1.1.116/9664 (cited on page 124).
- [26] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy. *Polygon mesh processing*. English. AK Peters, 2010. ISBN: 978-1-56881-426-1 (cited on page 99).
- [27] J. E. J. Bottomley. “Implementing Clusters for High Availability”. English. In: *ATEC ’04. Proceedings of the Annual Conference on USENIX*. Boston, MA: USENIX Association, 2004, pages 44–51 (cited on page 77).
- [28] G. R. Bradski. “Computer Vision Face Tracking For Use in a Perceptual User Interface”. English. In: *Interface* 2.2 (1998), pages 12–21. DOI: 10.1.1.14.7673 (cited on page 16).
- [29] G. R. Bradski and A. Kaehler. *Learning OpenCV*. English. O’Reilly, 2008 (cited on page 154).
- [30] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. “Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33.9 (2011), pages 1820–1833. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2010.232 (cited on page 15).
- [31] P.-J. Bristeau, F. Callou, D. Vissière, and N. Petit. “The Navigation and Control technology inside the AR . Drone micro UAV”. English. In: *Proceedings of the 18th IFAC World Congress, 2011*. Edited by S. Bittanti and P. Colaneri. Volume 18. Milano: Elsevier, 2011,

- pages 1477–1484. DOI: 10.3182/20110828-6-IT-1002.02327 (cited on page 140).
- [32] D. C. Brown. “Decentering Distortion of Lenses”. English. In: *Photometric Engineering* 32.3 (1966), pages 444–462 (cited on page 48).
- [33] R. Brown. “A brief account of microscopical observations made in the months of June, July and August 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies”. English. In: *Philosophical Magazine Series 2* 4.21 (Sept. 1828), pages 161–173. ISSN: 1941-5850. DOI: 10.1080/14786442808674769 (cited on page 28).
- [34] D. Brščić, M. Eggers, F. Rohrmüller, O. Kourakos, S. Sosnowski, D. Althoff, M. Lawitzky, A. Mörtl, M. Rambow, V. Koropouli, J. Medina Hernández, X. Zang, W. Wang, D. Wollherr, K. Kühnlenz, C. Mayer, T. Kruse, A. Kirsch, J. Blume, A. Bannat, T. Rehl, F. Wallhoff, T. Lorenz, P. Basili, C. Lenz, T. Röder, G. Panin, W. Maier, S. Hirche, M. Buss, M. Beetz, B. Radig, A. Schubö, S. Glasauer, A. Knoll, and E. Steinbach. *Multi Joint Action in CoTeSys - Setup and Challenges*. English. Technical report. Munich: Technische Universität München, 2010 (cited on page 145).
- [35] T. F. Burks, S. A. Shearer, and F. A. Payne. “Classification of Weed Species Using Color Texture Features and Discriminant Analysis”. English. In: *Transactions of the ASAE* 43.2 (2000), pages 441–448. ISSN: 0001-2351 (cited on page 82).
- [36] K. M. D. Bushby, T. Cole, J. N. Matthews, and J. A. Goodship. “Centiles for adult head circumference.” English. In: *Archives of Disease in Childhood* 67.10 (1992), pages 1286–1287. DOI: 10.1136/adc.67.10.1286 (cited on page 97).
- [37] Q. Cai and J. K. Aggarwal. “Tracking human motion in structured environments using a distributed-camera system”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21.11 (1999), pages 1241–1247. DOI: 10.1109/34.809119 (cited on pages 11, 19).
- [38] F. Caillette, A. Galata, and T. Howard. “Real-time 3-D human body tracking using learnt models of behaviour”. English. In: *Computer Vision and Image Understanding* 109.2 (Feb. 2008), pages 112–125. ISSN: 10773142. DOI: 10.1016/j.cviu.2007.05.005 (cited on page 4).

- [39] O. Cappé, S. J. Godsill, and E. Moulines. “An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo”. English. In: *Proceedings of the IEEE* 95.5 (2007), pages 899–924. DOI: 10.1109/JPROC.2007.893250 (cited on page 18).
- [40] S. Carbini, J.-E. Viallet, and O. Bernier. “Pointing Gesture Visual Recognition by Body Feature Detection and Tracking”. English. In: *Computer Vision and Graphics. International Conference, ICCVG 2004, Warsaw, Poland, September 2004, Proceedings*. Edited by K. Wojciechowski, B. Smolka, B. Palus, R. S. Kozera, W. Skarbek, and L. Noakes. Warsaw: Springer Netherlands, 2004, pages 203–208. DOI: 10.1007/1-4020-4179-9_29 (cited on page 128).
- [41] E. Carey and V. Rowley. *GigE Vision: Video Streaming and Device Control Over Ethernet Standard*. English. Technical report. Ann Harbor, MI: Automated Imaging Association (AIA), 2011 (cited on page 33).
- [42] G. Castellano, L. Kessous, and G. Caridakis. “Emotion Recognition through Multiple Modalities: Face, Body Gesture, Speech”. English. In: *Affect and Emotion in Human-Computer Interaction SE - 8*. Edited by C. Peter and R. Beale. Volume 4868. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pages 92–103. ISBN: 978-3-540-85098-4. DOI: 10.1007/978-3-540-85099-1_8 (cited on page 123).
- [43] T. Celik, H. Demirel, H. Ozkaramanli, and M. Uyguroglu. “Fire detection using statistical color model in video sequences”. English. In: *Visual Communication and Image Representation, Journal of* 18.2 (Apr. 2007), pages 176–185. ISSN: 10473203. DOI: 10.1016/j.jvcir.2006.12.003 (cited on page 82).
- [44] W. Chen, Y. Q. Shi, and G. Xuan. “Identifying Computer Graphics using HSV Color Model and Statistical Moments of Characteristic Functions”. English. In: *Multimedia and Expo, 2007 IEEE International Conference on*. Beijing: IEEE, July 2007, pages 1123–1126. ISBN: 1-4244-1016-9. DOI: 10.1109/ICME.2007.4284852 (cited on page 82).
- [45] S. Chib and E. Greenberg. “Understanding the Metropolis-Hastings Algorithm”. English. In: *The American Statistician* 49.4 (1995), pages 327–335. DOI: 10.1080/00031305.1995.10476177 (cited on page 57).

- [46] B.-H. Cho, J.-W. Bae, and S.-H. Jung. “Image Processing-Based Fire Detection System Using Statistic Color Model”. English. In: *Advanced Language Processing and Web Information Technology, 2008. AL-PIT '08. International Conference on*. Dalian Liaoning: IEEE, 2008, pages 245–250. ISBN: 978-0-7695-3273-8. DOI: 10.1109/ALPIT.2008.49 (cited on page 82).
- [47] C.-H. Chuang, J.-W. Hsieh, L.-W. Tsai, S.-Y. Chen, and K.-C. Fan. “Carried Object Detection Using Ratio Histogram and its Application to Suspicious Event Analysis”. English. In: *Circuits and Systems for Video Technology, IEEE Transactions on*. 19.6 (2009), pages 911–916. DOI: 10.1109/TCSVT.2009.2017415 (cited on page 2).
- [48] A. Collazos, D. Fernandez-Lopez, A. S. Montemayor, J. J. Pantrigo, and M. L. Elgado. “Abandoned Object Detection on Controlled Scenes Using Kinect”. English. In: *Natural and Artificial Computation in Engineering and Medical Applications. 5th International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2013, Mallorca, Spain, June 10-14, 2013. Proceedings, Part II*. Edited by J. Ferrández Vicente, J. Álvarez Sánchez, F. Paz López, and F. Toledo Moreo. Lecture Notes in Computer Science. Palma de Mallorca: Springer Berlin Heidelberg, 2013, pages 169–178. ISBN: 978-3-642-38621-3. DOI: 10.1007/978-3-642-38622-0_18 (cited on page 136).
- [49] D. Comaniciu, V. Ramesh, and P. Meer. “Real-time Tracking of Non-Rigid Objects using Mean Shift”. English. In: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. Edited by D. Forsyth and D. Kriegman. Volume 2. 5. Hilton Head Island, SC: IEEE Computer Society, May 2000, pages 142–149. ISBN: 0-7695-0662-3. DOI: 10.1109/CVPR.2000.854761 (cited on page 16).
- [50] D. Comaniciu, V. Ramesh, and P. Meer. “Kernel-based Object Tracking”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25.5 (2003), pages 564–577. DOI: 10.1109/TPAMI.2003.1195991 (cited on page 16).
- [51] D. J. Connor and J. O. Limb. “Properties of frame-difference signals generated by moving images”. English. In: *Communications, IEEE Transactions on*. 22.10 (1974), pages 1564–1575. DOI: 10.1109/TCOM.1974.1092083 (cited on page 61).
- [52] H. Cooper, B. Holt, and R. Bowden. “Sign Language Recognition”. English. In: *Visual Analysis of Humans - Looking at People*. Edited by T. B. Moeslund, A. Hilton, V. Krüger, and L. Sigal. London: Springer

- London, 2011, pages 539–562. ISBN: 978-0-85729-996-3. DOI: 10.1007/978-0-85729-997-0_27 (cited on page 123).
- [53] S. Coorg and S. Teller. “Real-time occlusion culling for models with large occluders”. English. In: *Proceedings of the 1997 symposium on Interactive 3D graphics - SI3D '97*. New York City, NY: ACM Press, 1997, pages 83–90. ISBN: 0897918843. DOI: 10.1145/253284.253312 (cited on page 103).
- [54] T. F. Cootes, G. J. Edwards, and C. J. Taylor. “Active appearance models”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.6 (June 2001), pages 681–685. ISSN: 01628828. DOI: 10.1109/34.927467 (cited on page 84).
- [55] T. F. Cootes and C. J. Taylor. *Statistical Models of Appearance for Computer Vision*. English. Technical report. Manchester: University of Manchester, 2004 (cited on page 84).
- [56] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. “Active Shape Models-Their Training and Application”. English. In: *Computer Vision and Image Understanding* 61.1 (Jan. 1995), pages 38–59. ISSN: 10773142. DOI: 10.1006/cviu.1995.1004 (cited on page 84).
- [57] T. N. Cornsweet. *Visual perception*. English. New York, New York: Academic Press, 1970. ISBN: 0-12-189750-8 (cited on page 2).
- [58] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti. “Improving Shadow Suppression in Moving Object Detection with HSV Color Information”. English. In: *Intelligent Transportation Systems, 2001. Proceedings*. Oakland, CA: IEEE, 2001, pages 334–339. ISBN: 0-7803-7194-1. DOI: 10.1109/ITSC.2001.948679 (cited on page 82).
- [59] R. Cucchiara, A. Prati, and R. Vezzani. “A multi-camera vision system for fall detection and alarm generation”. English. In: *Expert Systems* 24.5 (Nov. 2007), pages 334–345. ISSN: 0266-4720. DOI: 10.1111/j.1468-0394.2007.00438.x (cited on page 142).
- [60] L. Da Vinci. *The Vitruvian Man*. Venice, 1487 (cited on pages 95, 97).
- [61] W. Daamen and S. P. Hoogendoorn. “Experimental Research of Pedestrian Walking Behavior”. English. In: *Transportation Research Record* 1828.1 (Jan. 2003), pages 20–30. ISSN: 0361-1981. DOI: 10.3141/1828-03 (cited on page 55).

- [62] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. English. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Edited by C. Schmid, S. Soatto, and C. Tomasi. Volume 1. San Diego, CA: IEEE Computer Society, 2005, pages 886–893. DOI: 10.1109/CVPR.2005.177 (cited on pages 16, 84).
- [63] J. Daugman. “Face and Gesture Recognition: Overview”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19.7 (1997), pages 675–676. ISSN: 0162-8828. DOI: 10.1109/34.598225 (cited on page 125).
- [64] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum Likelihood from Incomplete Data via the EM Algorithm”. English. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1 (1977), pages 1–38 (cited on page 23).
- [65] J. Denzler and H. Niemann. “Real-time pedestrian tracking in natural scenes”. English. In: *Computer Analysis of Images and Patterns. 7th International Conference, CAIP '97 Kiel, Germany, September 10–12, 1997 Proceedings*. Edited by G. Sommer, K. Daniilidis, and J. Pauli. Volume 1296. Lecture Notes in Computer Science. Kiel: Springer Berlin Heidelberg, 1997, pages 42–49. ISBN: 978-3-540-63460-7. DOI: 10.1007/3-540-63460-6_98 (cited on page 3).
- [66] D. Devarajan, Z. Cheng, and R. J. Radke. “Calibrating Distributed Camera Networks”. English. In: *Proceedings of the IEEE* 96.10 (2008), pages 1625–1639. ISSN: 00189219. DOI: 10.1109/JPROC.2008.928759 (cited on page 46).
- [67] F. Dierks, H. Nebelung, M. Albrecht, A. Happe, E. Carey, P. Zhou, F. Mathieu, K. I. Christensen, L. Zeineh, M. Krag, S. Thommen, J. Becvar, S. Maurice, C. Zierl, M. Rüder, J. Scholtz, E. Gross, A. Rivard, F. Gobeil, V. Rowley, R. Stelz, and S. Dorenbeck. *GenICam Standard*. English. Technical report. European Machine Vision Association (EMVA), 2009 (cited on page 48).
- [68] P. Dollár, C. Wojek, B. Schiele, and P. Perona. “Pedestrian Detection: A Benchmark”. English. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Edited by D. Huttenlocher, G. Medioni, and J. Rehg. Miami, FL: IEEE Computer Society, June 2009, pages 304–311. ISBN: 978-1-4244-3992-8. DOI: 10.1109/CVPR.2009.5206631 (cited on page 15).

- [69] T. D’Orazio, P. Mazzeo, and P. Spagnolo. “Color Brightness Transfer Function evaluation for non overlapping multi camera tracking”. English. In: *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*. Edited by J. Park and S. S. Bhattacharyya. Como: IEEE, Aug. 2009, pages 1–6. ISBN: 9781424446209. DOI: 10.1109/ICDSC.2009.5289365 (cited on page 107).
- [70] A. Doucet, S. Godsill, and C. Andrieu. “On sequential Monte Carlo sampling methods for Bayesian filtering”. English. In: *Statistics and Computing* 10.3 (2000), pages 197–208. ISSN: 0960-3174. DOI: 10.1023/A:1008935410038 (cited on page 28).
- [71] L. Duan, G. Cui, W. Gao, and H. Zhang. “Adult Image Detection Method Base-on Skin Color Model and Support Vector Machine”. English. In: *Computer Vision, 5th Asian Conference on. ACCV 2002. Proceedings*. January. Melbourne, 2002, pages 1–4 (cited on page 82).
- [72] W. Eckstein and C. Steger. “The Halcon Vision System: An Example for Flexible Software Architecture”. English. In: *Practical Applications of Real-Time Image Processing, 3rd Japanese Conference on. Proceedings*. Edited by ‘Technical Committee of Image Processing Applications’. Japanese Society for Precision Engineering, 1999, pages 18–23 (cited on pages 48, 154).
- [73] M. Eggers, V. Dikov, C. Mayer, C. Steger, and B. Radig. “Setup and Calibration of a Distributed Camera System for Surveillance of Laboratory Space”. English. In: *Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications* 23.4 (2013), pages 481–487. ISSN: 1054-6618. DOI: 10.1134/S1054661813040032 (cited on page 145).
- [74] M. Eggers, V. Dikov, C. Steger, and B. Radig. “Setup and Calibration of a Distributed Camera System for Surveillance of Laboratory Space”. English. In: *Pattern Recognition and Image Understanding, 8th Open German-Russian Workshop on. OGRW-8-2011. Proceedings*. Edited by H. Niemann, Y. Zhuralev, I. Gurevich, B. Radig, and Y. Vasin. Nizhny Novgorod: N. I. Lobachevsky Nizhny Novgorod State University, 2011, pages 44–47 (cited on pages 26, 145).
- [75] R. El Kaliouby and P. Robinson. “Real-Time Inference of Complex Mental States from Facial Expressions and Head Gestures”. English. In: *Real-time Vision for Human-Computer Interaction*. Springer, 2005, pages 181–200. ISBN: 0387276971 (cited on page 123).

- [76] A.-L. Ellis, A. S. Shahrokni, and J. M. Ferryman. “PETS2009 and Winter-PETS 2009 results: A combined evaluation”. English. In: *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*. Boston, MA: IEEE, 2009, pages 1–8. ISBN: 978-1-4244-8310-5. DOI: 10.1109/PETS-WINTER.2009.5399728 (cited on page 119).
- [77] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. “Object Detection with Discriminatively Trained Part-Based Models”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.9 (2010), pages 1627–1645. DOI: 10.1109/TPAMI.2009.167 (cited on pages 15, 16).
- [78] P. F. Felzenszwalb, D. McAllester, and D. Ramanan. “A Discriminatively Trained, Multiscale, Deformable Part Model”. English. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. Edited by K. Boyer, M. Shah, and T. Syeda-Mahmood. Anchorage, AK: IEEE Computer Society, June 2008, pages 1–8. ISBN: 978-1-4244-2242-5. DOI: 10.1109/CVPR.2008.4587597 (cited on page 84).
- [79] R. Féraund, O. J. Bernier, J.-E. Viallet, and M. Collobert. “A fast and accurate face detector based on neural networks”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.1 (2001), pages 42–53. ISSN: 01628828. DOI: 10.1109/34.899945 (cited on page 128).
- [80] J. M. Ferryman and A.-L. Ellis. “PETS2010: Dataset and Challenge”. English. In: *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*. Boston, MA: IEEE, 2010, pages 143–150. ISBN: 978-0-7695-4264-5. DOI: 10.1109/AVSS.2010.90 (cited on page 119).
- [81] J. M. Ferryman and A. S. Shahrokni. “An overview of the PETS 2009 challenge”. English. In: *Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*. Miami, FL: IEEE, 2009 (cited on page 119).
- [82] M. Fiala. *ARTag, an improved marker system based on ARToolkit*. English. Technical report July. Ottawa, Canada: NRC Institute for Information Technology; National Research Council Canada, 2004 (cited on page 47).

- [83] M. Fiala. “ARTag, a fiducial marker system using digital techniques”. English. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Edited by C. Schmid, S. Soatto, and C. Tomasi. Volume 2. San Diego, CA: IEEE Computer Society, 2005, pages 590–596. DOI: 10.1109/CVPR.2005.74 (cited on pages 15, 47).
- [84] I. Fine, D. I. A. MacLeod, and G. M. Boynton. “Surface segmentation based on the luminance and color statistics of natural scenes”. English. In: *Journal of the Optical Society of America A: Optics, Image Science, and Vision* 20.7 (July 2003), pages 1283–1291. ISSN: 1084-7529. DOI: 10.1364/JOSAA.20.001283 (cited on page 83).
- [85] S. Fleck, F. Busch, P. Biber, and W. Straßer. “3D Surveillance – A Distributed Network of Smart Cameras for Real-Time Tracking and its Visualization in 3D”. English. In: *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*. New York City, NY: IEEE Computer Society, 2006, pages 118–126. ISBN: 0769526462. DOI: 10.1109/CVPRW.2006.6 (cited on pages 20–23).
- [86] S. Fleck, S. Lanwer, and W. Straßer. “A Smart Camera Approach to Real-Time Tracking”. English. In: *13th European Signal Processing Conference (EUSIPCO 2005)*. Edited by A. E. Çetin and A. M. Tekalp. 3. Antalya: Curran Associates, 2005, pages 1894–1897 (cited on page 21).
- [87] D. Focken and R. Stiefelhagen. “Towards vision-based 3-D people tracking in a smart room”. English. In: *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*. Pittsburgh, PA: IEEE Computer Society, 2002, pages 400–405. ISBN: 0-7695-1834-6. DOI: 10.1109/ICMI.2002.1167028 (cited on page 2).
- [88] J. A. Freer, B. J. Beggs, H. L. Fernandez-Canque, F. Chevrier, and A. Goryashko. “Automatic Intruder Detection Incorporating Intelligent Scene Monitoring with Video Surveillance”. English. In: *Security and Detection, 1997. ECOS 97., European Conference on*. London, 1997, pages 109–113 (cited on page 2).
- [89] L. M. Fuentes and S. A. Velastin. “People tracking in surveillance applications”. English. In: *Image and Vision Computing* 24.11 (Nov. 2006), pages 1165–1171. ISSN: 02628856. DOI: 10.1016/j.imavis.2005.06.006 (cited on page 16).

- [90] T. Gandhi and M. M. Trivedi. “Person tracking and reidentification: Introducing Panoramic Appearance Map (PAM) for feature representation”. English. In: *Machine Vision and Applications* 18.3-4 (Feb. 2007), pages 207–220. ISSN: 0932-8092. DOI: 10.1007/s00138-006-0063-x (cited on page 16).
- [91] D. M. Gavrila. “The Visual Analysis of Human Movement: A Survey”. English. In: *Computer Vision and Image Understanding* 73.1 (Jan. 1999), pages 82–98. ISSN: 10773142. DOI: 10.1006/cviu.1998.0716 (cited on page 125).
- [92] P. Geismann and G. Schneider. “A two-staged approach to vision-based pedestrian recognition using Haar and HOG features”. English. In: *2008 IEEE Intelligent Vehicles Symposium*. Eindhoven: IEEE, June 2008, pages 554–559. ISBN: 978-1-4244-2568-6. DOI: 10.1109/IVS.2008.4621148 (cited on page 84).
- [93] T. Gevers and A. W. M. Smeulders. “Color-based object recognition”. English. In: *Pattern Recognition* 32.3 (Mar. 1999), pages 453–464. ISSN: 00313203. DOI: 10.1016/S0031-3203(98)00036-3 (cited on page 82).
- [94] C. J. Geyer. “Practical Markov Chain Monte Carlo”. English. In: *Statistical Science* 7.4 (1992), pages 473–483 (cited on page 18).
- [95] J. J. Gibson. *The perception of the visual world*. English. Oxford: Houghton Mifflin, 1950 (cited on page 2).
- [96] J. J. Gibson. *The ecological approach to visual perception*. English. Hillsdale, NJ: Lawrence Erlbaum Associates, 1986. ISBN: 0-89859-959-8 (cited on page 2).
- [97] T. Gill, J. M. Keller, D. T. Anderson, and R. H. Luke. “A system for change detection and human recognition in voxel space using the Microsoft Kinect sensor”. English. In: *2011 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. Washington, DC: IEEE, Oct. 2011, pages 1–8. ISBN: 978-1-4673-0216-6. DOI: 10.1109/AIPR.2011.6176347 (cited on page 136).
- [98] M. Goebel. “Eine realzeitfähige Architektur zur Integration kognitiver Funktionen”. German. Dissertation. Munich: Technische Universität München, 2009. ISBN: 3868531661 (cited on pages 5, 48, 154).

- [99] M. Goebel and G. Färber. “A Real-Time-capable Hard-and Software Architecture for Joint Image and Knowledge Processing in Cognitive Automobiles”. English. In: *Intelligent Vehicles Symposium, 2007 IEEE*. Edited by P. Ioannau, C. Stiller, and T. Hasegawa. Istanbul: IEEE, 2007, pages 734–740. ISBN: 1424410673. DOI: 10.1109/IVS.2007.4290204 (cited on pages 5, 48).
- [100] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing using Matlab*. English. 2nd edition. Gatesmark Publishing, 2009. ISBN: 978-0982085400 (cited on page 49).
- [101] M. A. Goodrich and D. R. Olsen. “Seven Principles of Efficient Human Robot Interaction”. English. In: *Systems, Man and Cybernetics, 2003. IEEE International Conference on*. Volume 4. Washington, DC: IEEE, 2003, pages 3942–3948. DOI: 10.1109/ICSMC.2003.1244504 (cited on page 127).
- [102] D. Gray, S. Brennan, and H. Tao. “Evaluating appearance models for recognition, reacquisition, and tracking”. English. In: *Performance Evaluation for Tracking and Surveillance (PETS), 10th International Workshop on*. Edited by J. Ferryman, J. L. Crowley, and D. Tweed. Volume 3. Rio de Janeiro: IEEE, 2007, pages 41–47 (cited on pages 14, 83, 167).
- [103] P. J. Green. “Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination”. English. In: *Biometrika* 82.4 (1995), pages 711–732. DOI: 10.1093/biomet/82.4.711 (cited on page 18).
- [104] N. Greene, M. Kass, and G. Miller. “Hierarchical Z-buffer visibility”. English. In: *SIGGRAPH '93 Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. New York, New York: ACM Press, 1993, pages 231–238. ISBN: 0897916018. DOI: 10.1145/166117.166147 (cited on page 103).
- [105] A. Griesser. *Real-Time, GPU-based Foreground-Background Segmentation*. English. Technical report. Zürich: Eidgenössische Technische Hochschule Zürich, 2005 (cited on page 51).
- [106] M. T. Hagan, H. B. Demuth, and M. Beale. *Neural network design*. English. Boston, MA: PWS Publishing Co., 1996. ISBN: 0-534-94332-2 (cited on page 128).
- [107] E. T. Hall. “A System for the Notation of Proxemic Behavior”. English. In: *American Anthropologist* 65.5 (1963), pages 1003–1026. DOI: 10.1525/aa.1963.65.5.02a00020 (cited on page 55).

- [108] E. T. Hall. *The Hidden Dimension*. English. Anchor Books, 1966. ISBN: 978-0385084765 (cited on page 55).
- [109] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. “The WEKA Data Mining Software: An Update”. English. In: *ACM SIGKDD Explorations Newsletter* 11.1 (2009), pages 10–18. DOI: 10.1145/1656274.1656278 (cited on pages 126, 154).
- [110] A. Hampapur, L. Brown, J. Connell, S. Pankanti, A. Senior, and Y. Tian. “Smart surveillance: applications, technologies and implications”. English. In: *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*. Volume 2. Singapore: IEEE, 2003, pages 1133–1138. ISBN: 0-7803-8185-8. DOI: 10.1109/ICICS.2003.1292637 (cited on page 1).
- [111] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. English. 2nd edition. Cambridge University Press, 2004. Chapter 189. ISBN: 0521540518. DOI: 10.1016/S0143-8166(01)00145-2 (cited on pages 45, 103).
- [112] W. K. Hastings. “Monte Carlo Sampling Methods Using Markov Chains and Their Applications”. English. In: *Biometrika* 57.1 (1970), pages 97–109. DOI: 10.1093/biomet/57.1.97 (cited on page 18).
- [113] D. M. Hawkins. “The problem of overfitting.” English. In: *Journal of Chemical Information and Computer Sciences* 44.1 (2004), pages 1–12. ISSN: 0095-2338. DOI: 10.1021/ci0342472 (cited on page 130).
- [114] J. Heikkilä. “Geometric camera calibration using circular control points”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.10 (2000), pages 1066–1077. ISSN: 01628828. DOI: 10.1109/34.879788 (cited on page 45).
- [115] S. Hengstler and H. Aghajan. “Application-Oriented Design of Smart Camera Networks”. English. In: *Distributed Smart Cameras, 2007. ICDSC '07. First ACM/IEEE International Conference on*. Edited by H. Adhajan and R. Kleihorst. Vienna: IEEE, Sept. 2007, pages 12–19. ISBN: 978-1-4244-1353-9. DOI: 10.1109/ICDSC.2007.4357500 (cited on page 24).
- [116] M. Henning and M. Spruiell. *Distributed Programming with Ice*. English. Technical report. ZeroC Inc., 2007 (cited on pages 6, 29, 50, 154).

- [117] M. Herrmann, C. Mayer, and B. Radig. “Automatic Generation of Image Analysis Programs”. English. In: *Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications*. Samara: SP MAIK Nauka/Interperiodica, 2013 (cited on page 132).
- [118] B. M. Hill, M. Helmke, and C. Burger. *The Official Ubuntu Book*. English. 5th edition. Prentice Hall, 2010. ISBN: 978-0137081301 (cited on page 154).
- [119] E. Hörster and R. Lienhart. “On the optimal placement of multiple visual sensors”. In: *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks. VSSN '06*. New York, NY, USA: ACM, 2006, pages 111–120. ISBN: 1-59593-496-0. DOI: 10.1145/1178782.1178800 (cited on page 3).
- [120] W. A. Horton, J. G. Hall, and J. T. Hecht. “Achondroplasia”. English. In: *Lancet* 370.9582 (July 2007), pages 162–72. ISSN: 1474-547X. DOI: 10.1016/S0140-6736(07)61090-3 (cited on page 90).
- [121] N. von Hoyningen-Huene and M. Beetz. “Importance Sampling as One Solution to the Data Association Problem in Multi-target Tracking”. English. In: *Computer Vision, Imaging and Computer Graphics. Theory and Applications. International Joint Conference, VISIGRAPP 2009, Lisboa, Portugal, February 5-8, 2009. Revised Selected Papers*. Edited by A. Ranchordas, J. M. Pereira, H. J. Araújo, and J. M. R. Tavares. Lisboa: Springer Berlin Heidelberg, 2009, pages 309–325. ISBN: 978-3-642-11839-5. DOI: 10.1007/978-3-642-11840-1_23 (cited on page 16).
- [122] N. von Hoyningen-Huene and M. Beetz. “Rao-Blackwellized Resampling Particle Filter for Real-time Player Tracking in Sports”. In: *Computer Vision Theory and Applications. International Conference on, VISAPP 2009, Lisboa, Portugal*. Lisboa, 2009, pages 464–471 (cited on page 18).
- [123] N. von Hoyningen-Huene and M. Beetz. “Robust Real-Time Multiple Target Tracking”. English. In: *Computer Vision – ACCV 2009. 9th Asian Conference on Computer Vision, Xi’an, September 23-27, 2009, Revised Selected Papers, Part II*. Edited by H. Zha, R.-I. Taniguchi, and S. Maybank. Xi’an: Springer Berlin Heidelberg, 2009, pages 247–256. DOI: 10.1007/978-3-642-12304-7_24 (cited on page 18).
- [124] C.-T. Hsieh, H.-C. Wang, Y.-K. Wu, L.-C. Chang, and T.-K. Kuo. “A Kinect-based people-flow counting system”. English. In: *Intelligent Signal Processing and Communications Systems (ISPACS), 2012 International Symposium on*. Edited by J.-M. Guo, H. Kiya, T. Shih,

- and Y.-H. Li. Ispacs. New Taipei: IEEE, Nov. 2012, pages 146–150. ISBN: 978-1-4673-5082-2. DOI: 10.1109/ISPACS.2012.6473470 (cited on page 136).
- [125] M.-K. Hu. “Visual pattern recognition by moment invariants”. English. In: *Information Theory, IRE Transactions on*. 8.2 (1962), pages 179–187. DOI: 10.1109/TIT.1962.1057692 (cited on page 53).
- [126] T. Hudson, D. Manocha, J. Cohen, M. Lin, and H. Zhang. “Accelerated Occlusion Culling using Shadow Frusta”. English. In: *SCG '97 Proceedings of the thirteenth annual symposium on Computational geometry*. New York City, NY: ACM New York, 1997, pages 1–10. DOI: 10.1145/262839.262847 (cited on page 103).
- [127] R. S. Hunter and R. W. Harold. *The Measurement of Appearance*. English. 2nd edition. Wiley-Interscience, 1987. ISBN: 978-0-471-83006-1 (cited on pages 14, 165).
- [128] M. Isard and J. MacCormick. “BraMBLe: A Bayesian multiple-blob tracker”. English. In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Edited by R. Horaud, T. Matsuyama, and R. Szeliski. Vancouver, BC: IEEE, 2001, pages 34–41. ISBN: 0-7695-1143-0. DOI: 10.1109/ICCV.2001.937594 (cited on pages 95, 116, 118).
- [129] H. Izadinia, I. Saleemi, W. Li, and M. Shah. “(MP)2T: Multiple People Multiple Parts Tracker”. English. In: *Computer Vision – ECCV 2012. 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI*. Edited by A. Fitzgibbon, S. Labeznik, P. Perona, Y. Sato, and C. Schmid. Florence: Springer Berlin Heidelberg, 2012, pages 100–114. DOI: 10.1007/978-3-642-33783-3_8 (cited on page 16).
- [130] A. Jaimes and N. Sebe. “Multimodal human-computer interaction: A survey”. English. In: *Computer Vision and Image Understanding. Special Issue on Vision for Human-Computer Interaction* 108.1-2 (Oct. 2007), pages 116–134. ISSN: 10773142. DOI: 10.1016/j.cviu.2006.10.019 (cited on page 123).
- [131] O. Javed, Z. Rasheed, O. Alatas, and M. Shah. “KNIGHTTM: A Real Time Surveillance System for Multiple Overlapping and Non-Overlapping Cameras”. English. In: *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*. Edited by K. J. R. Liu, S. F. Chang, R. Civanlar, J. Ostermann, and J. Sorensen. Baltimore, MD: IEEE, 2003, 649–652 Vol.1. DOI: 10.1109/ICME.2003.1221001 (cited on pages 20–22, 26).

- [132] O. Javed, K. Shafique, Z. Rasheed, and M. Shah. “Modeling inter-camera space–time and appearance relationships for tracking across non-overlapping views”. English. In: *Computer Vision and Image Understanding* 109.2 (Feb. 2008), pages 146–162. ISSN: 10773142. DOI: 10.1016/j.cviu.2007.01.003 (cited on pages 20, 23).
- [133] O. Javed, K. Shafique, and M. Shah. “Appearance modeling for tracking in multiple non-overlapping cameras”. English. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Edited by C. Schmid, S. Soatto, and C. Tomasi. Volume 2. San Diego, CA: IEEE Computer Society, 2005, pages 26–33 (cited on page 20).
- [134] O. Javed and M. Shah. “Tracking and Object Classification for Automated Surveillance”. English. In: *Computer Vision – ECCV 2002. 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part IV*. Edited by A. Heyden, G. Sparr, M. Nielsen, and P. Johansen. Copenhagen: Springer Berlin Heidelberg, 2002, pages 343–357. DOI: 10.1007/3-540-47979-1_23 (cited on page 3).
- [135] Y. Jeong, D. Nistér, D. Steedly, R. Szeliski, and I.-S. Kweon. “Pushing the envelope of modern methods for bundle adjustment”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.8 (Aug. 2012), pages 1605–17. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2011.256 (cited on page 45).
- [136] M. Jindai and T. Watanabe. “Development of a handshake request motion model based on analysis of handshake motion between humans”. English. In: *Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on*. Faculty of Computer Science and System Engineering, Okayama Prefectural University, 111 Kuboki, Soja, 719-1197, JAPAN. Budapest: IEEE, 2011, pages 560–565. ISBN: 9781457708374. DOI: 10.1109/AIM.2011.6026975 (cited on page 125).
- [137] A. Johannsen and M. B. Carter. “Clustered Backface Culling”. English. In: *Journal of Graphics Tools* 3.1 (1998), pages 1–14. DOI: 10.1080/10867651.1998.10487484 (cited on page 103).
- [138] I. T. Jolliffe. *Principal Component Analysis*. English. 2nd edition. New York City, NY: Springer New York, 2002. ISBN: 978-0-387-95442-4. DOI: 10.1007/b98835 (cited on page 126).

- [139] M. J. Jones and J. M. Rehg. “Statistical color models with application to skin detection”. English. In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*. Edited by R. Beveridge and K. Price. Volume 1. 1. Fort Collins, CO: IEEE Computer Society, 1999, pages 81–96. DOI: 10.1109/CVPR.1999.786951 (cited on pages 82, 83).
- [140] C.-F. Juang, W.-K. Sun, and G.-C. Chen. “Object detection by color histogram-based fuzzy classifier with support vector learning”. English. In: *Neurocomputing* 72.10–12 (June 2009), pages 2464–2476. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2008.11.016 (cited on page 82).
- [141] R. E. Kahn and M. J. Swain. “Understanding people pointing: the Perseus system”. English. In: *Computer Vision, 1995. Proceedings., International Symposium on*. Edited by A. Blake, S. Shafer, and K. Sugihara. Coral Gables, FL: IEEE Computer Society, 1995, pages 569–574. ISBN: 0-8186-7190-4. DOI: 10.1109/ISCV.1995.477062 (cited on page 127).
- [142] T. Kailath. “The divergence and Bhattacharyya distance measures in signal selection”. English. In: *Communication Technology, IEEE Transactions on* 15.1 (Feb. 1967), pages 52–60. ISSN: 0018-9332. DOI: 10.1109/TCOM.1967.1089532 (cited on page 74).
- [143] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems”. English. In: *Journal of Fluids Engineering* 82.1 (1960), pages 35–45. DOI: 10.1115/1.3662552 (cited on page 21).
- [144] R. E. Kalman and R. S. Bucy. “New Results in Linear Filtering and Prediction Theory”. English. In: *Journal of Fluids Engineering* 83.1 (1961), pages 95–108. ISSN: 00219223. DOI: 10.1115/1.3658902 (cited on page 21).
- [145] I. Karatzas and S. E. Shreve. “Brownian Motion”. English. In: *Brownian Motion and Stochastic Calculus*. Volume 113. Graduate Texts in Mathematics. New York City, NY: Springer US, 1988. Chapter 2, pages 47–127. ISBN: 978-1-4684-0304-6. DOI: 10.1007/978-1-4684-0302-2_2 (cited on page 28).
- [146] B. Karlsson. *Beyond the C++ Standard Library: An Introduction to Boost*. English. 1st edition. Addison-Wesley Professional, 2005. ISBN: 978-0321133540 (cited on page 154).

- [147] R. Karlsson and F. Gustafsson. “Monte Carlo data association for multiple target tracking”. English. In: *Target Tracking: Algorithms and Applications (Ref. No. 2001/174)*, IEE. Volume 1. Enschede: IEEE, 2001, 13:1–13:5. DOI: 10.1049/ic:20010239 (cited on page 18).
- [148] R. M. Karp. “Reducibility Among Combinatorial Problems.” English. In: *Proceedings of a Symposium on the Complexity of Computer Computations*. Volume 40. 4. Plenum Press, Dec. 1972, pages 85–103. DOI: 10.2307/2271828 (cited on page 20).
- [149] H. Kato and M. Billinghurst. “Marker tracking and HMD calibration for a video-based augmented reality conferencing system”. English. In: *Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on*. Edited by G. Klinker. San Francisco, CA: IEEE Computer Society, 1999, pages 85–94. ISBN: 0-7695-0359-4. DOI: 10.1109/IWAR.1999.803809 (cited on page 15).
- [150] R. Kehl and L. Van Gool. “Real-time pointing gesture recognition for an immersive environment”. English. In: *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*. Seoul: IEEE Computer Society, 2004, pages 577–582. ISBN: 0-7695-2122-3. DOI: 10.1109/AFGR.2004.1301595 (cited on page 128).
- [151] J. R. Kender. *Saturation, hue, and normalized color: Calculation, digitization effects, and use*. English. Technical report. Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science, 1976 (cited on page 93).
- [152] A. Kermanikian. *Introducing Mudbox*. English. Sybex, 2010. ISBN: 978-0470537251 (cited on page 101).
- [153] V. Kettner and R. Zabih. “Bayesian multi-camera surveillance”. English. In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*. Edited by R. Beveridge and K. Price. Volume 2. c. Fort Collins, CO: IEEE Computer Society, 1999. DOI: 10.1109/CVPR.1999.784638 (cited on page 15).
- [154] S. Khan and M. Shah. “Consistent Labeling of Tracked Objects in Multiple Cameras with Overlapping Fields of View”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25.10 (2003), pages 1355–1360. DOI: 10.1109/TPAMI.2003.1233912 (cited on page 19).

- [155] K. K. Kim, K. C. Kwak, and S. Y. Chi. “Gesture Analysis for Human-Robot Interaction”. English. In: *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference on*. Phoenix Park: IEEE, 2006, 1823–1827 Vol. 3. ISBN: 89-5519-129-4. DOI: 10.1109/ICACT.2006.206345 (cited on pages 125, 127).
- [156] S.-H. Kim, N.-K. Kim, S. C. Ahn, and H.-G. Kim. “Object oriented face detection using range and color information”. English. In: *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. Nara: IEEE Computer Society, 1998, pages 76–81. ISBN: 0-8186-8344-9. DOI: 10.1109/AFGR.1998.670928 (cited on page 82).
- [157] A. G. Kirk, J. F. O’Brien, and D. A. Forsyth. “Skeletal Parameter Estimation from Optical Motion Capture Data”. English. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Edited by C. Schmid, S. Soatto, and C. Tomasi. Volume 2. San Diego, CA: IEEE Computer Society, 2005, 782–788 Vol. 2. ISBN: 0-7695-2372-2. DOI: 10.1109/CVPR.2005.326 (cited on page 15).
- [158] T. Kisler. “Evaluation of Pointing Gestures For Natural Human-Robot Interaction Using a Top-Down Multi-Camera System”. English. Master Thesis. Augsburg: Fachhochschule Augsburg - University of Applied Sciences, 2010 (cited on page 132).
- [159] R. Kjeldsen. “Head Gestures for Computer Control”. English. In: *Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*. Vancouver, BC: IEEE Computer Society, 2001, pages 61–67. ISBN: 0-7695-1074-4. DOI: 10.1109/RATFG.2001.938911 (cited on page 123).
- [160] N. Koenig. “Toward real-time human detection and tracking in diverse environments”. English. In: *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*. Edited by D. Mareschal, B. Scassellati, and J. Weng. London: IEEE Computational Intelligence Society, July 2007, pages 94–98. ISBN: 978-1-4244-1115-3. DOI: 10.1109/DEVLRN.2007.4354077 (cited on page 24).
- [161] R. Kohavi. “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. English. In: *International Joint Conference on Artificial Intelligence (IJCAI-95)*. Volume 5. Montreal: Morgan Kaufmann Publishers, 1995, pages 1137–1143 (cited on page 126).

- [162] K. Kosuge and Y. Hirata. “Human-Robot Interaction”. English. In: *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*. Shenyang: IEEE, 2004, pages 8–11. ISBN: 0-7803-8614-8. DOI: 10.1109/ROBIO.2004.1521743 (cited on page 2).
- [163] M. Krinidis, G. Stamou, H. Teutsch, N. Nikolaidis, R. Rabenstein, and I. Pitas. “An audio-visual database for evaluating person tracking algorithms”. English. In: *Acoustics; Speech; and Signal Processing; 2005. Proceedings. (ICASSP '05). IEEE International Conference on*. Philadelphia, PA: IEEE, 2005, pages 237–240. ISBN: 0-7803-8874-7. DOI: 10.1109/ICASSP.2005.1415385 (cited on page 119).
- [164] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. “Multi-camera multi-person tracking for EasyLiving”. English. In: *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on*. Dublin: IEEE Computer Society, 2000, pages 3–10. ISBN: 0-7695-0698-4. DOI: 10.1109/VS.2000.856852 (cited on page 3).
- [165] M. H. Kryder and C. S. Kim. “After Hard Drives—What Comes Next?” English. In: *Magnetics, IEEE Transactions on*. 45.10 (Oct. 2009), pages 3406–3413. ISSN: 0018-9464. DOI: 10.1109/TMAG.2009.2024163 (cited on page 1).
- [166] C.-h. Kuo, C. Huang, and R. Nevatia. “Inter-camera association of multi-target tracks by on-line learned appearance affinity models”. English. In: *Computer Vision – ECCV 2010. 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I*. Edited by K. Daniilidis, P. Maragos, and N. Paragios. Heraklion: Springer Berlin Heidelberg, 2010, pages 383–396. DOI: 10.1007/978-3-642-15549-9_28 (cited on page 20).
- [167] G. Kurillo, Z. Li, and R. Bajcsy. “Framework for hierarchical calibration of multi-camera systems for teleimmersion”. English. In: *IMMERSCOM '09 Proceedings of the 2nd International Conference on Immersive Telecommunications*. Edited by D. Mukherjee, A. Smolic, G. Alregib, and K. Nahrstedt. Berkeley, CA: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pages 1–6 (cited on page 46).
- [168] R. H. Kwong and E. W. Johnston. “A variable step size LMS algorithm”. English. In: *Signal Processing, IEEE Transactions on* 40.7 (July 1992), pages 1633–1642. ISSN: 1053587X. DOI: 10.1109/78.143435 (cited on page 126).

- [169] J. C. Langer and D. A. Singer. “Reflections on the Lemniscate of Bernoulli: The Forty-Eight Faces of a Mathematical Gem”. English. In: *Milan Journal of Mathematics* 78.2 (2010), pages 643–682. DOI: 10.1007/s00032-010-0124-5 (cited on page 66).
- [170] S. Lanser, C. Zierl, and R. Beutlhauser. “Multibildkalibrierung einer CCD-Kamera”. German. In: *Mustererkennung 1995. Verstehen akustischer und visueller Informationen. 17. DAGM-Symposium. Bielefeld, 13. - 15. September 1995*. Edited by G. Sagerer, S. Posch, and F. Kummert. Informatik aktuell. Bielefeld: Springer Berlin Heidelberg, 1995, pages 481–491. DOI: 10.1007/978-3-642-79980-8_57 (cited on pages 43, 48).
- [171] O. Lanz, P. Chippendale, and R. Brunelli. “An Appearance-based Particle Filter for Visual Tracking in Smart Rooms”. English. In: *Multi-modal Technologies for Perception of Humans. International Evaluation Workshops CLEAR 2007 and RT 2007, Baltimore, MD, USA, May 8-11, 2007, Revised Selected Papers*. Edited by R. Stiefelhagen, R. Bowers, and J. Fiscus. Volume 4625. Baltimore, MD: Springer Berlin Heidelberg, 2008, pages 57–69. DOI: 10.1007/978-3-540-68585-2_4 (cited on pages 24, 84).
- [172] J. Y. Lee and S. I. Yoo. “An Elliptical Boundary Model for Skin Color Detection”. English. In: *Imaging Science, Systems, and Technology, 2002 International Conference on. CISST’02. Proceedings*. Edited by H. Arabnia and Y. Mun. Las Vegas, NV: CSREA Press, 2002. ISBN: 9781892512932 (cited on page 82).
- [173] L. Lee and W. E. L. Grimson. “Gait Analysis for Recognition and Classification”. English. In: *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. Washington, DC: IEEE Computer Society, 2002, pages 148–155. ISBN: 0769516025. DOI: 10.1109/AFGR.2002.1004148 (cited on page 2).
- [174] C. Lenz, S. Nair, M. Rickert, A. Knoll, W. Rösel, J. Gast, A. Bannat, and F. Wallhoff. “Joint-Action for Humans and Industrial Robots for Assembly Tasks”. English. In: *RO-MAN 2008 - The 17th IEEE International Symposium on Robot and Human Interactive Communication*. Munich: IEEE, Aug. 2008, pages 130–135. ISBN: 978-1-4244-2212-8. DOI: 10.1109/ROMAN.2008.4600655 (cited on page 2).
- [175] C. Lenz, T. Röder, M. Eggers, S. Amin, T. Kisler, B. Radig, G. Panin, and A. Knoll. “A Distributed Many-Camera System for Multi-person Tracking”. English. In: *Ambient Intelligence. First International Joint Conference, AmI 2010, Malaga, Spain, November 10-12,*

2010. *Proceedings*. Edited by B. DeRuyter, R. Wichert, D. V. Keyson, P. Markopoulos, N. Streitz, M. Divitini, N. Georgantas, and A. Mana Gomez. Malaga: Springer Berlin Heidelberg, 2010, pages 217–226. DOI: 10.1007/978-3-642-16917-5_22 (cited on page 145).
- [176] R. Lienhart and J. Maydt. “An extended set of Haar-like features for rapid object detection”. English. In: *Image Processing. 2002. Proceedings. 2002 International Conference on*. Volume 1. Rochester, NY: IEEE Signal Processing Society, 2002, pages 900–903. DOI: 10.1109/ICIP.2002.1038171 (cited on page 84).
- [177] A. Livshin and X. Rodet. “The Importance of Cross Database Evaluation in Sound Classification”. English. In: *ISMIR 2003, 4th International Conference on Music Information Retrieval, Baltimore, Maryland, USA, October 27-30, 2003, Proceedings*. Edited by H. H. Hoos and D. Bainbridge. Baltimore, MD: Johns Hopkins University, 2003 (cited on page 130).
- [178] D. G. Lowe. “Object Recognition from Local Scale-Invariant Features”. English. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. Edited by A. Blake, Y. Ohta, and S. Zucker. Volume 2. Kerkyra: IEEE, 1999, pages 1150–1157. ISBN: 0-7695-0164-8. DOI: 10.1109/ICCV.1999.790410 (cited on page 17).
- [179] B. D. Lucas and T. Kanade. “An Iterative Image Registration Technique with an Application to Stereo Vision”. English. In: *Artificial Intelligence (IJCAI '81), 7th International Joint Conference on. Proceedings*. Edited by P. J. Hayes. Vancouver, BC: William Kaufman, 1981, pages 674–679 (cited on page 126).
- [180] M. Maeda, T. Ogawa, K. Kiyokawa, and H. Takemura. “Tracking of User Position and Orientation by Stereo Measurement of Infrared Markers and Orientation Sensing”. English. In: *Wearable Computers, 2004. ISWC 2004. Eighth International Symposium on*. Edited by M. Smith and B. H. Thomas. Volume 1. Arlington, VA: IEEE, 2004, pages 77–84. ISBN: 0-7695-2186-X. DOI: 10.1109/ISWC.2004.46 (cited on page 15).
- [181] E. Maggio and a. Cavallaro. “Multi-part target representation for color tracking”. In: *IEEE International Conference on Image Processing 2005 (2005)*, pages I–729. DOI: 10.1109/ICIP.2005.1529854 (cited on page 67).
- [182] E. N. Marieb and K. Hoehn. *Human Anatomy & Physiology*. English. 9th edition. Benjamin Cummings (Pearson), 2012. ISBN: 978-0321743268 (cited on page 153).

- [183] C. Martin, F.-F. Steege, and H.-M. Gross. “Estimation of pointing poses for visually instructing mobile robots under real world conditions”. English. In: *Robotics and Autonomous Systems* 58.2 (Feb. 2010), pages 174–185. ISSN: 09218890. DOI: 10.1016/j.robot.2009.09.013 (cited on page 128).
- [184] M. Mason and Z. Duric. “Using histograms to detect and track objects in color video”. English. In: *Applied Image Pattern Recognition Workshop (AIPR 2001), Analysis and Understanding of Time Varying Imagery, 10-12 October 2001, Washington, DC, USA, Proceedings. 30th*. Edited by C. J. Cohen. Washington, DC: IEEE Computer Society, 2001, pages 154–159. ISBN: 0-7695-1245-3. DOI: 10.1109/AIPR.2001.991219 (cited on page 82).
- [185] C. Mayer. “Facial Expression Recognition With A Three-Dimensional Face Model”. English. Dissertation. Munich: Technische Universität München, 2012 (cited on pages 85, 123, 129, 141).
- [186] C. Mayer, M. Wimmer, M. Eggers, and B. Radig. “Facial Expression Recognition with 3D Deformable Models”. English. In: *Advances in Computer-Human Interactions, 2009. ACHI '09. Second International Conferences on*. Edited by S. Dascalu and I. Poupyrev. Cancun: IEEE, 2009, pages 26–31. ISBN: 9781424433513. DOI: 10.1109/ACHI.2009.33 (cited on pages 129, 141).
- [187] C. Mayer, M. Wimmer, F. Stulp, Z. Riaz, A. Roth, M. Eggers, and B. Radig. “A Real Time System for Model-based Interpretation of the Dynamics of Facial Expressions”. English. In: *Automatic Face & Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*. Edited by K. Bowyer, M. S. Bartlett, D. Metaxas, I. Patras, D. Hogg, M. Nixon, M. Petrou, I. Pitas, V. Blanz, C. Pelachaud, Q. Ji, and A. Amir. Amsterdam: IEEE Computer Society, 2008, pages 1–2. ISBN: 9781424421534. DOI: 10.1109/AFGR.2008.4813440 (cited on page 129).
- [188] R. Mehrotra, K. R. Namuduri, and N. Ranganathan. “Gabor filter-based edge detection”. English. In: *Pattern Recognition* 25.12 (1992), pages 1479–1494. ISSN: 0031-3203. DOI: 10.1016/0031-3203(92)90121-X (cited on page 129).
- [189] R. Mester, T. Aach, and D. Lutz. “Illumination-invariant change detection using a statistical colinearity criterion”. English. In: *Pattern Recognition. 23rd DAGM Symposium Munich, Germany, September 12-14, 2001 Proceedings*. Edited by B. Radig and S. Florczyk. Munich:

- Springer Berlin Heidelberg, 2001, pages 170–177. DOI: 10.1007/3-540-45404-7_23 (cited on page 128).
- [190] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. “Equation of State Calculations by Fast Computing Machines”. English. In: *The Journal of Chemical Physics* 21.6 (1953), pages 1087–1092. ISSN: 00219606. DOI: 10.1063/1.1699114 (cited on page 18).
- [191] D. L. Mills. “Internet time synchronization: the network time protocol”. English. In: *Communications, IEEE Transactions on*. 39.10 (1991), pages 1482–1493. ISSN: 00906778. DOI: 10.1109/26.103043 (cited on pages 49, 154).
- [192] S. C. Mitchell, J. G. Bosch, B. P. F. Lelieveldt, R. J. van der Geest, J. H. C. Reiber, and M. Sonka. “3-D active appearance models: segmentation of cardiac MR and ultrasound images.” English. In: *Medical Imaging, IEEE Transactions on*. 21.9 (Sept. 2002), pages 1167–1178. ISSN: 0278-0062. DOI: 10.1109/TMI.2002.804425 (cited on page 84).
- [193] S. Mitra and T. Acharya. “Gesture Recognition: A Survey”. English. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 37.3 (2007), pages 311–324. DOI: 10.1109/TSMCC.2007.893280 (cited on page 125).
- [194] T. B. Moeslund and E. Granum. “A Survey of Computer Vision-Based Human Motion Capture”. English. In: *Computer Vision and Image Understanding* 81.3 (Mar. 2001), pages 231–268. ISSN: 10773142. DOI: 10.1006/cviu.2000.0897 (cited on page 15).
- [195] T. B. Moeslund, A. Hilton, and V. Krüger. “A Survey of Advances in Computer Vision-Based Human Motion Capture”. English. In: *Computer Vision and Image Understanding* 104.2-3 (Nov. 2006), pages 90–126. ISSN: 10773142. DOI: 10.1016/j.cviu.2006.08.002 (cited on page 15).
- [196] G. E. Moore. “Cramming more components onto integrated circuits”. English. In: *Electronics* 38.8 (1965), pages 114–117 (cited on page 1).
- [197] T. J. Mowbray and R. Zahavi. *The essential CORBA: Systems Integration Using Distributed Objects*. English. 1st edition. New York, NY: John Wiley & Sons, Inc., 1995. ISBN: 978-0471106111 (cited on page 29).

- [198] D. Murray, T. Koziniec, K. Lee, and M. Dixon. “Large MTUs and internet performance”. English. In: *High Performance Switching and Routing (HPSR), 2012 IEEE 13th International Conference on*. Edited by J. Chao, E. Oki, and C. Minkenberg. Belgrade, Serbia: IEEE, June 2012, pages 82–87. ISBN: 978-1-4577-0833-6. DOI: 10.1109/HPSR.2012.6260832 (cited on page 40).
- [199] Y. Nakanishi and V. Nethery. “Anthropometric comparison between Japanese and Caucasian American male University students”. English. In: *Applied Human Science* 18.1 (Jan. 1999), pages 9–11. ISSN: 1341-3473 (cited on pages 90, 95, 97).
- [200] M. Namjoo and E. J. McCluskey. “Watchdog Processors and Capability Checking”. English. In: *Twenty-Fifth International Symposium on Fault-Tolerant Computing, 1995, ' Highlights from Twenty-Five Years'*. Volume III. IEEE, 1996, page 94. ISBN: 0-8186-7150-5. DOI: 10.1109/FTCSH.1995.532618 (cited on page 49).
- [201] M. Neumann. “Gestenerkennung mit Optischem Flußin Kamerabildern aus Top-Down Perspektive”. German. Diplomarbeit in Informatik. Munich: Technische Universität München, 2010 (cited on page 127).
- [202] K. Nickel and R. Stiefelhagen. “Real-time person tracking and pointing gesture recognition for human-robot interaction”. English. In: *Vision in Human-Computer Interaction. ECCV 2004 Workshop on HCI, Prague, Czech Republic, May 16, 2004. Proceedings*. Edited by N. Sebe, M. Lew, and T. S. Huang. Prague: Springer Berlin Heidelberg, 2004, pages 28–38. DOI: 10.1007/978-3-540-24837-8_4 (cited on page 24).
- [203] K. Nickel and R. Stiefelhagen. “Visual recognition of pointing gestures for human-robot interaction”. English. In: *Image and Vision Computing* 25.12 (Dec. 2007), pages 1875–1884. ISSN: 02628856. DOI: 10.1016/j.imavis.2005.12.020 (cited on page 128).
- [204] T. Nierhoff, L. Lou, V. Koropouli, M. Eggers, T. Fritzsich, O. Kourakos, K. Kühnlenz, D. Lee, B. Radig, S. Hirche, and M. Buss. “Tire Mounting on a Car Using the Real-Time Control Architecture ARCADE”. English. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. Edited by E. Guglielmelli, D. Accoto, and C. Laugier. Vilamoura: IEEE, 2012, pages 4804–4805. DOI: 10.1109/IROS.2012.6386094 (cited on page 145).

- [205] W. Niu, L. Jiao, D. Han, and Y.-F. Wang. “Real-time multiperson tracking in video surveillance”. English. In: *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*. Edited by S. S. Abeysekera, Q. Tian, A. Alphones, and T. H. Chiang. Volume 2. December. Singapore: IEEE, 2003, pages 1144–1148. ISBN: 0780381858. DOI: 10.1109/ICICS.2003.1292639 (cited on page 16).
- [206] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L. Van Gool. “Color-Based Object Tracking in Multi-Camera Environments”. English. In: *Pattern Recognition. 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003. Proceedings*. Edited by B. Michaelis and G. Krell. Magdeburg: Springer Berlin Heidelberg, 2003, pages 591–599. DOI: 10.1007/978-3-540-45243-0_75 (cited on pages 20, 21, 23).
- [207] K. Nummiaro, E. Koller-Meier, and L. Van Gool. “An Adaptive Color-Based Particle Filter”. English. In: *Image and Vision Computing* 21.1 (2003), pages 99–110. DOI: 10.1016/S0262-8856(02)00129-4 (cited on pages 21, 83).
- [208] C. L. Ogden, C. D. Fryar, M. D. Carroll, and K. M. Flegal. “Mean body weight, height, and body mass index, United States 1960-2002.” English. In: *Advance Data* 347 (Oct. 2004), pages 1–17. ISSN: 0147-3956 (cited on pages 36, 95).
- [209] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe. “A Boosted Particle Filter: Multitarget Detection and Tracking”. English. In: *Computer Vision - ECCV 2004. 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I*. Edited by T. Pajdla and J. Matas. Lecture Notes in Computer Science. Prague: Springer Berlin Heidelberg, 2004, pages 28–39. ISBN: 978-3-540-21984-2. DOI: 10.1007/978-3-540-24670-1_3 (cited on page 82).
- [210] O. Ozturk, T. Yamasaki, and K. Aizawa. “Tracking of humans and estimation of body/head orientation from top-view single camera for visual focus of attention analysis”. In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. Kyoto: IEEE, Sept. 2009, pages 1020–1027. ISBN: 978-1-4244-4442-7. DOI: 10.1109/ICCVW.2009.5457590 (cited on page 17).

- [211] G. Panin. *Model-Based Visual Tracking*. English. Hoboken, NJ: John Wiley & Sons, Inc., Apr. 2011. ISBN: 9780470943922. DOI: 10.1002/9780470943922 (cited on pages 6, 17, 18, 51, 58, 154).
- [212] G. Panin, T. Röder, and A. Knoll. “Integrating robust likelihoods with Monte-Carlo filters for multi-target tracking”. English. In: *Vision, Modeling, and Visualization 2008: Proceedings, October 8-10, 2008. 13th International Fall Workshop on*. Edited by O. Deussen, D. Keim, and D. Saupe. Konstanz: Akademische Verlagsgesellschaft AKA, 2008, pages 293–301 (cited on pages 6, 57).
- [213] M. A. Patricio, J. Carbó, O. Pérez, J. García, and J. M. Molina. “Multi-Agent Framework in Visual Sensor Networks”. English. In: *EURASIP Journal on Advances in Signal Processing* 2007.1 (2007), pages 1–21. ISSN: 16876172. DOI: 10.1155/2007/98639 (cited on page 24).
- [214] K. Pearson. “On Lines and Planes of Closest Fit to Systems of Points in Space”. English. In: *The London, Edinburgh, and Dublin Philosophical Magazine, Series 6*. 2.11 (1901), pages 559–572. DOI: 10.1080/14786440109462720 (cited on page 126).
- [215] A. P. Pentland. “Classification by clustering”. English. In: *Symposium on Machine Processing of Remotely Sensed Data*. Edited by P. H. Swain, D. B. Morrison, and D. E. Parks. West Lafayette, IN: IEEE, 1976, page 137 (cited on page 25).
- [216] A. P. Pentland. “Smart Rooms”. English. In: *Scientific American* 274.4 (1996), pages 54–62 (cited on page 2).
- [217] M. Piccardi. “Background subtraction techniques: a review”. In: *Systems, Man and Cybernetics, 2004 IEEE International Conference on*. Volume 4. Seoul: IEEE, 2004, pages 3099–3104. ISBN: 0-7803-8567-5. DOI: 10.1109/ICSMC.2004.1400815 (cited on page 51).
- [218] G. S. Pingali, Y. Jean, and I. Carlbom. “Real Time Tracking for Enhanced Tennis Broadcasts”. English. In: *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. Edited by D. Terzopoulos and Y.-F. Wang. Santa Barbara, CA: IEEE Computer Society, 1998, pages 260–265. DOI: 10.1109/CVPR.1998.698618 (cited on page 16).
- [219] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. “Globally-optimal greedy algorithms for tracking a variable number of objects”. English. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. Providence, RI: IEEE Computer Society, 2011,

- pages 1201–1208. ISBN: 978-1-4577-0394-2. DOI: 10.1109/CVPR.2011.5995604 (cited on page 16).
- [220] F. Pitié, A. C. Kokaram, and R. Dahyot. “N-Dimensional Probability Density Function Transfer and its Application to Colour Transfer”. English. In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. Edited by B. Freeman, L. van Gool, and S. Chaudhuri. Volume 2. Beijing: IEEE, 2005, pages 1434–1439. DOI: 10.1109/ICCV.2005.166 (cited on page 26).
- [221] M. Pollefeys, R. Koch, and L. Van Gool. “Self-Calibration and Metric Reconstruction In spite of Varying and Unknown Intrinsic Camera Parameters”. English. In: *International Journal of Computer Vision* 32.1 (Aug. 1999), pages 7–25. ISSN: 0920-5691. DOI: 10.1023/A:1008109111715 (cited on page 140).
- [222] M. Pollefeys, S. N. Sinha, L. Guan, and J.-S. Franco. “Multi-View Calibration, Synchronization, and Dynamic Scene Reconstruction”. English. In: *Multi-Camera Networks: Principles and Applications*. Edited by H. Aghajan and A. Cavallaro. Elsevier, 2008. Chapter Chapter 2, pages 29–75. DOI: 10.1016/B978-0-12-374633-7.00004-5 (cited on page 46).
- [223] P. W. Power and J. A. Schoonees. “Understanding Background Mixture Models for Foreground Segmentation”. English. In: *Image and Vision Computing New Zealand 2002 (IVCNZ 2002). Proceedings*. November. Auckland, 2002 (cited on pages 6, 51).
- [224] J. R. Quinlan. *C4.5: Programs for Machine Learning*. English. 1st edition. San Francisco, CA: Morgan Kaufmann Publishers Inc., 1993. ISBN: 1558602402 (cited on page 126).
- [225] J. R. Quinlan. “Improved Use of Continuous Attributes in C4.5”. English. In: *Journal of Artificial Intelligence Research* 4 (1996), pages 77–90. arXiv: 9603103 [arXiv:cs] (cited on page 126).
- [226] L. R. Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition”. English. In: *Proceedings of the IEEE* 77.2 (1989), pages 257–286. ISSN: 0018-9219. DOI: 10.1109/5.18626 (cited on page 128).
- [227] C. R. Rao. *Linear Statistical Inference and Its Applications*. English. 2nd edition. Volume 22. Wiley-Interscience, 2001. ISBN: 978-0471218753 (cited on pages 47, 87).

- [228] I. Rekleitis and G. Dudek. “Automated calibration of a camera sensor network”. English. In: *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*. Edited by H. Zhang, M. Buss, K. Goldeberg, Z. Li, and T. Shibata. Edmonton, Alberta: IEEE, 2005, pages 3384–3389. ISBN: 0-7803-8912-3. DOI: 10.1109/IROS.2005.1545014 (cited on page 140).
- [229] Z. Riaz. “Visual Interpretation of Human Body Language for Interactive Scenarios”. English. Dissertation. Munich: Technische Universität München, 2011 (cited on page 123).
- [230] Z. Riaz, C. Mayer, M. Wimmer, M. Beetz, and B. Radig. “A Model Based Approach for Expressions Invariant Face Recognition”. English. In: *Advances in Biometrics. Third International Conference, ICB 2009, Alghero, Italy, June 2-5, 2009. Proceedings*. Edited by M. Tistarelli and M. S. Nixon. Alghero: Springer Berlin Heidelberg, 2009, pages 289–298. DOI: 10.1007/978-3-642-01793-3_30 (cited on page 2).
- [231] P. Richette, T. Bardin, and C. Stheneur. “Achondroplasia: from genotype to phenotype.” English. In: *Joint Bone Spine* 75.2 (Mar. 2008), pages 125–30. ISSN: 1778-7254. DOI: 10.1016/j.jbspin.2007.06.007 (cited on page 90).
- [232] B. Rinner and W. Wolf. “An Introduction to Distributed Smart Cameras”. English. In: *Proceedings of the IEEE* 96.10 (2008), pages 1565–1575. ISSN: 00189219. DOI: 10.1109/JPROC.2008.928742 (cited on page 24).
- [233] G. Robinson. *Speeding Net Traffic With Tiny Mirrors*. Sept. 2000 (cited on page 1).
- [234] L. Rocha, L. Velho, and P. C. P. Carvalho. “Motion reconstruction using moments analysis”. English. In: *Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on*. Curitiba: IEEE, 2004, pages 354–361. DOI: 10.1109/SIBGRA.2004.1352981 (cited on page 53).
- [235] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. “A database for fine grained activity detection of cooking activities”. English. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. Providence, RI: IEEE Computer Society, 2012, pages 1194–1201. ISBN: 978-1-4673-1228-8. DOI: 10.1109/CVPR.2012.6247801 (cited on page 142).

- [236] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal, and B. Schiele. “Script Data for Attribute-Based Recognition of Composite Activities”. English. In: *Computer Vision – ECCV 2012. 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I*. Edited by A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid. Lecture Notes in Computer Science. Florence: Springer Berlin Heidelberg, 2012, pages 144–157. ISBN: 978-3-642-33717-8. DOI: 10.1007/978-3-642-33718-5_11 (cited on page 142).
- [237] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning representations by back-propagating errors”. English. In: *Nature* 323.6088 (1986), pages 533–536. DOI: 10.1038/323533a0 (cited on page 129).
- [238] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. “The intelligent ASIMO: System overview and integration”. English. In: *Intelligent Robots and Systems (IROS), 2002. IEEE/RSJ International Conference on*. Edited by C. Laugier, H. Christensen, H. Hashimoto, G. Lee, and A. Zelinsky. Volume 3. Lausanne: IEEE, 2002, pages 2478–2483. DOI: 10.1109/IRDS.2002.1041641 (cited on page 125).
- [239] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. “Evaluation of color descriptors for object and scene recognition”. English. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. Edited by K. Boyer, M. Shah, and T. Syeda-Mahmood. Anchorage, AK: IEEE Computer Society, June 2008, pages 1–8. ISBN: 978-1-4244-2242-5. DOI: 10.1109/CVPR.2008.4587658 (cited on page 17).
- [240] R. R. Schaller. “Moore’s law: past, present and future”. English. In: *Spectrum, IEEE* 34.6 (June 1997), pages 52–59. ISSN: 00189235. DOI: 10.1109/6.591665 (cited on page 1).
- [241] K. Schindler, L. Van Gool, and B. de Gelder. “Recognizing emotions expressed by body pose: A biologically inspired neural model”. English. In: *Neural Networks* 21.9 (2008), pages 1238–1246. ISSN: 0893-6080 (cited on page 124).
- [242] Y. Schröder, A. Scholz, K. Berger, K. Ruhl, S. Guthe, and M. Magnor. “Multiple Kinect Studies”. English. In: *Technical Report 2011-09-15* (2011) (cited on page 136).

- [243] B. Schroeder and G. A. Gibson. “Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you”. English. In: *FAST '07. 5th USENIX Conference on File and Storage Technologies. Proceedings*. Edited by A. C. Arpaci-Dusseau and R. H. Arpaci-Dusseau. San José, CA: USENIX Association, 2007, pages 1–16 (cited on page 76).
- [244] B. Schroeder and G. A. Gibson. “Understanding disk failure rates”. English. In: *ACM Transactions on Storage* 3.3 (Oct. 2007), 8:1–8:31. ISSN: 15533077. DOI: 10.1145/1288783.1288785 (cited on page 76).
- [245] B. Schroeder, E. Pinheiro, and W.-D. Weber. “DRAM Errors in the Wild: A Large-Scale Field Study”. English. In: *SIGMETRICS '09 Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*. Edited by T. Bonald and J. Nieh. Seattle, WA: ACM New York, 2009, pages 193–204. ISBN: 9781605585116. DOI: 10.1145/1555349.1555372 (cited on page 76).
- [246] P. Sebastian, Y. V. Voon, and R. Comley. “The effect of colour space on tracking robustness”. English. In: *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on*. Singapore: IEEE, June 2008, pages 2512–2516. ISBN: 978-1-4244-1717-9. DOI: 10.1109/ICIEA.2008.4582971 (cited on pages 82, 84).
- [247] S. Šegvic and S. Ribaric. “A Software Architecture for Distributed Visual Tracking in a Global Vision Localization System”. English. In: *Computer Vision Systems* (2003) (cited on page 24).
- [248] A. W. Senior, A. Hampapur, and M. Lu. “Acquiring multi-scale images by pan-tilt-zoom control and automatic multi-camera calibration”. English. In: *Applications of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*. Edited by T. E. Boult, I. Cohen, and S. Soatto. Volume 1. Breckenridge, CO: IEEE, 2005, 433–438 Vol.1. ISBN: 0769522718 (cited on page 137).
- [249] J. Serra. *Image analysis and mathematical morphology*. English. 1st edition. London: Academic Press, 1982. ISBN: 0-12-637240-3 (cited on page 51).
- [250] E. G. Sirer and R. Farrow. “Some Lesser-Known Laws of Computer Science”. English. In: *;Login:* 32.4 (Aug. 2007), pages 25–28 (cited on page 1).

- [251] A. F. M. Smith and G. O. Roberts. “Bayesian Computation Via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods”. English. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 55.1 (1993), pages 3–23 (cited on page 18).
- [252] A. R. Smith. “Color Gamut Transform Pairs”. English. In: *SIGGRAPH Computer Graphics* 12.3 (Aug. 1978), pages 12–19. ISSN: 0097-8930. DOI: 10.1145/965139.807361 (cited on page 93).
- [253] L. Snidaro, C. Micheloni, and C. Chiavedale. “Video Security for Ambient Intelligence”. English. In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 35.1 (2005), pages 133–144. DOI: 10.1109/TSMCA.2004.838478 (cited on page 2).
- [254] C. Stauffer and W. E. L. Grimson. “Adaptive background mixture models for real-time tracking”. English. In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*. Edited by R. Beveridge and K. Price. Volume 2. Fort Collins, CO: IEEE Computer Society, 1999, pages 246–252. ISBN: 0-7695-0149-4. DOI: 10.1109/CVPR.1999.784637 (cited on page 51).
- [255] C. Steger, M. Ulrich, and C. Wiedemann. *Machine Vision Algorithms and Applications*. English. 1st edition. Weinheim: Wiley-VCH, 2008. ISBN: 3527407340 (cited on pages 43, 48, 154).
- [256] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich. “Common Metrics for Human-Robot Interaction”. English. In: *HRI '06 Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. Edited by A. C. Schultz and D. J. Bruemmer. Salt Lake City, UT: ACM New York, 2006, pages 33–40. ISBN: 1595932941. DOI: 10.1145/1121241.1121249 (cited on page 2).
- [257] D. Stødle, P. H. Ha, J. M. Bjørndalen, and O. J. Anshus. “Lessons Learned Using a Camera Cluster to Detect and Locate Objects”. English. In: *Parallel Computing: Architectures, Algorithms and Applications. Proceedings of the International Conference ParCo 2007*. Edited by C. Bischof, M. Bücker, P. Gibbon, G. R. Joubert, T. Lippert, B. Mohr, and F. Peters. Volume 15. Advances in Parallel Computing. Aachen: IOS Press, 2008, pages 71–78. ISBN: 978-1-58603-796-3. (Cited on page 25).
- [258] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. English. 3rd edition. Volume 2. Springer New York, 2002. ISBN: 978-0-387-95452-3. DOI: 10.1016/S0079-8169(04)80088-6 (cited on page 47).

- [259] H. Storner and B. Milne. *System Monitoring with Xymon*. English. September. San Francisco, CA: Wikimedia Foundation, 2013 (cited on page 75).
- [260] A. D. Straw, K. Branson, T. R. Neumann, and M. H. Dickinson. “Multi-camera real-time three-dimensional tracking of multiple flying animals.” English. In: *Journal of the Royal Society, Interface / the Royal Society* 8.56 (Mar. 2011), pages 395–409. ISSN: 1742-5662. DOI: 10.1098/rsif.2010.0230 (cited on pages 21, 23, 26).
- [261] M. J. Sullivan, C. A. Richards, C. E. Smith, O. Masoud, and N. P. Papanikolopoulos. “Pedestrian tracking from a stationary camera using active deformable models”. In: *Intelligent Vehicles '95 Symposium., Proceedings of the*. Detroit, MI: IEEE, 1995, pages 90–95. ISBN: 0-7803-2983-X. DOI: 10.1109/IVS.1995.528263 (cited on page 3).
- [262] S. Sural, G. Qian, and S. Pramanik. “Segmentation and histogram generation using the HSV color space for image retrieval”. English. In: *Image Processing, 2002. Proceedings. 2002 International Conference on*. Volume 2. Rochester, NY: IEEE, 2002, pages 589–592. ISBN: 0780376226. DOI: 10.1109/ICIP.2002.1040019 (cited on pages 91, 92).
- [263] E. E. Sutherland, R. F. Sproull, and R. A. Schumacker. “A Characterization of Ten Hidden-Surface Algorithms”. In: *ACM Computing Surveys* 6.1 (Jan. 1974), pages 1–55. ISSN: 03600300. DOI: 10.1145/356625.356626 (cited on page 103).
- [264] T. Svoboda, D. Martinec, and T. Pajdla. “A Convenient Multicamera Self-Calibration for Virtual Environments”. English. In: *Presence Teleoperators Virtual Environments* 14.4 (2005), pages 407–422. ISSN: 10547460. DOI: 10.1162/105474605774785325 (cited on page 46).
- [265] M. J. Swain and D. H. Ballard. “Indexing via color histograms”. English. In: *Computer Vision, 1990. Proceedings, Third International Conference on*. Edited by J.-O. Eklundh, A. Kak, and S. Tsuji. Osaka: IEEE, 1990, pages 390–393. DOI: 10.1109/ICCV.1990.139558 (cited on pages 82, 83).
- [266] M. J. Swain and D. H. Ballard. “Color indexing”. English. In: *International Journal of Computer Vision* 7.1 (Nov. 1991), pages 11–32. ISSN: 0920-5691. DOI: 10.1007/BF00130487 (cited on page 82).

- [267] T. Teixeira, D. Lymberopoulos, E. Culurciello, Y. Aloimonos, and A. Savvides. “A Lightweight Camera Sensor Network Operating on Symbolic Information”. English. In: *Distributed Smart Cameras DSC 2006, First Workshop on. Proceedings*. Edited by B. Rinner and W. Wolf. Boulder, CO, 2006 (cited on page 24).
- [268] J. Teizer and P. A. Vela. “Personnel tracking on construction sites using video cameras”. English. In: *Advanced Engineering Informatics* 23.4 (Oct. 2009), pages 452–462. ISSN: 14740346. DOI: 10.1016/j.aei.2009.06.011 (cited on page 3).
- [269] M. M. Tenorth. “Knowledge Processing for Autonomous Robots”. English. PhD thesis. Munich: Technische Universität München, 2011. DOI: 10.1007/SpringerReference_65611 (cited on page 142).
- [270] L. A. Thompson and D. W. Massaro. “Children’s integration of speech and pointing gestures in comprehension.” English. In: *Journal of Experimental Child Psychology* 57.3 (June 1994), pages 327–354. ISSN: 0022-0965. DOI: 10.1006/jecp.1994.1016 (cited on page 127).
- [271] Y. Tian. “Dynamic focus window selection using a statistical color model”. English. In: *Proceedings of SPIE 6069, Digital Photography II*. Edited by N. Sampat, J. M. DiCarlo, and R. A. Martin. Volume 6069. San José, CA: SPIE, Feb. 2006. DOI: 10.1117/12.641884 (cited on page 82).
- [272] M. E. Tipping. “The Relevance Vector Machine”. English. In: *Advances in Neural Information Processing Systems (NIPS’ 2000)*. 1. Denver, CO, 2000, pages 652–658 (cited on page 130).
- [273] M. E. Tipping. “Sparse Bayesian Learning and the Relevance Vector Machine”. English. In: *The Journal of Machine Learning Research* 1.1 (2001), pages 211–245. DOI: 10.1162/15324430152748236 (cited on page 130).
- [274] K. Tollmar, D. Demirdjian, and T. Darrell. “Gesture + Play: Full-Body Interaction for Virtual Environments”. English. In: *Extended Abstracts on Human Factors in Computing Systems - CHI ’03*. New York, NY: ACM Press, 2003, page 620. ISBN: 1581136374. DOI: 10.1145/765891.765894 (cited on page 123).
- [275] M. F. Tompsett, G. F. Amelio, W. J. J. Bertram, R. R. Buckley, W. J. McNamara, J. C. Mikkelsen, and D. A. Sealer. “Charge-coupled imaging devices: Experimental results”. English. In: *Electron Devices, IEEE Transactions on*. 18.11 (Nov. 1971), pages 992–996. ISSN: 0018-9383. DOI: 10.1109/T-ED.1971.17321 (cited on page 1).

- [276] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. “Bundle adjustment—a modern synthesis”. English. In: *Vision Algorithms: Theory and Practice. International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*. Edited by B. Triggs, A. Zisserman, and R. Szeliski. Corfu: Springer Berlin Heidelberg, 1999, pages 298–372. ISBN: 3540444807. DOI: 10.1007/3-540-44480-7_21 (cited on page 45).
- [277] R. Y. Tsai. “A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses”. English. In: *Robotics and Automation, IEEE Journal of 3.4 (1987)*. Edited by L. Wolff, S. Shafer, and G. Healey, pages 323–344. ISSN: 08824967. DOI: 10.1109/JRA.1987.1087109 (cited on pages 26, 42).
- [278] R. Urtasun, D. J. Fleet, and N. D. Lawrence. “Modeling human locomotion with topologically constrained latent variable models”. English. In: *Human Motion – Understanding, Modeling, Capture and Animation. Second Workshop, Human Motion 2007, Rio de Janeiro, Brazil, October 20, 2007. Proceedings*. Edited by A. Elgammal, B. Rosenhahn, and R. Klette. Rio de Janeiro: Springer Berlin Heidelberg, 2007, pages 104–118. DOI: 10.1007/978-3-540-75703-0_8 (cited on page 16).
- [279] O. Utsumi, K. Miura, I. Ide, S. Sakai, and H. Tanaka. “An object detection method for describing soccer games from video”. English. In: *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*. Volume 1. Lausanne: IEEE, 2002, pages 45–48. ISBN: 0-7803-7304-9. DOI: 10.1109/ICME.2002.1035714 (cited on page 82).
- [280] M. Valera and S. A. Velastin. “Intelligent distributed surveillance systems: a review”. English. In: *Vision, Image and Signal Processing, IEE Proceedings*. 152.2 (2005), pages 192–204. DOI: 10.1049/ip-vis:20041147 (cited on page 24).
- [281] M. Varma and A. Zisserman. “A Statistical Approach to Texture Classification from Single Images”. English. In: *International Journal of Computer Vision* 62.1-2 (2005), pages 61–81. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000046589.39864.ee (cited on page 83).
- [282] J. Vermaak, S. J. Godsill, and P. Pérez. “Monte carlo filtering for multi target tracking and data association”. English. In: *Aerospace and Electronic Systems, IEEE Transactions on*. 41.1 (2005), pages 309–332. DOI: 10.1109/TAES.2005.1413764 (cited on page 18).

- [283] J. P. de Villiers, F. W. Leuschner, and R. Geldenhuys. “Centi-pixel accurate real-time inverse distortion correction”. English. In: *Proceedings of SPIE 7266, Optomechatronic Technologies 2008*. Edited by Y. Otani, Y. Bellouard, J. T. Wen, D. Hodko, Y. Katagiri, S. K. Kassegne, J. Kofman, S. Kaneko, C. A. Perez, D. Coquin, O. Kaynak, Y. Cho, T. Fukuda, J. Yi, and F. Janabi-Sharifi. Volume 7266. 1. San Diego, CA: SPIE, Nov. 2008, pages 11/1–11/8. DOI: 10.1117/12.804771 (cited on page 28).
- [284] P. Viola and M. J. Jones. “Rapid Object Detection Using a Boosted Cascade of Simple Features”. English. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Edited by E. Grimson and D. Huttenlocher. Volume 1. Kauai, HI: IEEE Computer Society, 2001, pages 511–518. DOI: 10.1109/CVPR.2001.990517 (cited on pages 84, 126, 129, 141).
- [285] P. Viola and M. J. Jones. “Robust Real-Time Face Detection”. English. In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Edited by R. Horaud, T. Matsuyama, and R. Szeliski. Volume 2. Vancouver, BC: IEEE Computer Society, 2001, page 747. ISBN: 0-7695-1143-0. DOI: 10.1109/ICCV.2001.937709 (cited on pages 126, 129).
- [286] P. Viola, M. J. Jones, and D. Snow. “Detecting Pedestrians Using Patterns of Motion and Appearance”. English. In: *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. Edited by K. Ikeuchi, O. Faugeras, and J. Malik. Volume 2. Nice: IEEE, 2003, pages 734–741. ISBN: 0-7695-1950-4. DOI: 10.1109/ICCV.2003.1238422 (cited on page 15).
- [287] M. Vitruvius Pollio. *De architectura libri decem*. Latin and Greek. Rome (cited on pages 95, 97).
- [288] W. Waizenegger and I. Feldmann. “Calibration of a synchronized multi-camera setup for 3D videoconferencing”. English. In: *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2010*. Edited by A. Gotchev and L. Onural. Tampere: IEEE, June 2010, pages 1–4. ISBN: 978-1-4244-6377-0. DOI: 10.1109/3DTV.2010.5506303 (cited on page 46).
- [289] C. Walter. “Kryder’s Law”. In: *Scientific American* 293.2 (July 2005), pages 32–33. DOI: 10.1038/scientificamerican0805-32 (cited on page 1).

- [290] R. Y. Wang and J. Popović. “Real-Time Hand-Tracking with a Color Glove”. English. In: *Graphics, ACM Transactions on. (TOG) - Proceedings of ACM SIGGRAPH 2009* 28.3 (July 2009), 63:1–63:8. ISSN: 07300301. DOI: 10.1145/1531326.1531369 (cited on page 15).
- [291] T. Watanabe, M. Haseyama, and H. Kitajima. “A soccer field tracking method with wire frame model from TV images”. English. In: *Image Processing, 2004. ICIP '04. 2004 International Conference on*. Volume 3. Singapore: IEEE, 2004, pages 1633–1636. ISBN: 0-7803-8554-3. DOI: 10.1109/ICIP.2004.1421382 (cited on page 16).
- [292] N. Wiener. “Differential Space”. English. In: *Journal of Mathematical Physics* 2.58 (1923), pages 131–174 (cited on page 28).
- [293] O. Williams, A. Blake, and R. Cipolla. “Sparse Bayesian learning for efficient visual tracking”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27.8 (Aug. 2005), pages 1292–1304. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2005.167 (cited on page 130).
- [294] C. R. Wren, A. Azarbayejani, T. J. Darrell, and A. P. Pentland. “Pfinder: Real-Time Tracking of the Human Body”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19.7 (1997), pages 780–785. DOI: 10.1109/BIOROB.2006.1639077 (cited on page 25).
- [295] C. R. Wren, F. Sparacino, A. J. Azarbayejani, T. J. Darrell, T. E. Starner, A. Kotani, C. M. Chao, M. Hlavac, K. B. Russell, and A. P. Pentland. “Perceptive spaces for performance and entertainment untethered interaction using computer vision and audition”. English. In: *Applied Artificial Intelligence* 11.4 (June 1997), pages 267–284. ISSN: 0883-9514. DOI: 10.1080/088395197118154 (cited on page 25).
- [296] Y. Wu and T. S. Huang. “Vision-Based Gesture Recognition: A Review”. English. In: *Gesture-Based Communication in Human-Computer Interaction. International Gesture Workshop, GW'99 Gif-sur-Yvette, France, March 17-19, 1999 Proceedings*. Edited by A. Braffort, R. Gherbi, S. Gibet, D. Teil, and J. Richardson. Gif-sur-Yvette: Springer Berlin Heidelberg, 1999, pages 103–115. DOI: 10.1007/3-540-46616-9_10 (cited on page 125).
- [297] T. Yamasaki, Y. Nishioka, and K. Aizawa. “Interactive Retrieval for Multi-Camera Surveillance Systems Featuring Spatio-Temporal Summarization”. English. In: *MM '08 Proceedings of the 16th ACM International Conference on Multimedia*. Edited by A. El Saddik, S. Vuong,

- C. Griwodz, A. Del Bimbo, K. S. Candan, and A. Jaimes. Vancouver: ACM New York, 2008, pages 797–800. ISBN: 9781605583037. DOI: 10.1145/1459359.1459490 (cited on page 24).
- [298] M.-H. Yang and N. Ahuja. “Gaussian mixture model for human skin color and its applications in image and video databases”. English. In: *Proceedings of SPIE 3656, Storage and Retrieval for Image and Video Databases VII* (Dec. 1998). Edited by M. M. Yeung, B.-L. Yeo, and C. A. Bouman, pages 458–466. DOI: 10.1117/12.333865 (cited on page 128).
- [299] A. R. Zamir, A. Dehghan, and M. Shah. “GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs”. English. In: *Computer Vision – ECCV 2012. 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II*. Edited by A. Fitzgibbon, S. Labeznik, P. Perona, Y. Sato, and C. Schmid. Florence: Springer Berlin Heidelberg, 2012, pages 343–356. DOI: 10.1007/978-3-642-33709-3_25 (cited on page 20).
- [300] H. Zhang and K. E. Hoff. “Fast backface culling using normal masks”. English. In: *Proceedings of the 1997 symposium on Interactive 3D graphics - SI3D '97*. New York, New York, USA: ACM Press, 1997, page 106. ISBN: 0897918843. DOI: 10.1145/253284.253314 (cited on page 103).
- [301] X. Zhang, S. Fronz, and N. Navab. “Visual Marker Detection and Decoding in AR Systems: A Comparative Study”. English. In: *Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on*. Darmstadt: IEEE Computer Society, 2002, pages 97–106. ISBN: 0-7695-1781-1. DOI: 10.1109/ISMAR.2002.1115078 (cited on page 16).
- [302] Z. Zhang. “A Flexible New Technique for Camera Calibration”. English. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.11 (2000), pages 1330–1334. ISSN: 01628828. DOI: 10.1109/34.888718 (cited on page 26).
- [303] Z. Zhang. “Microsoft Kinect Sensor and Its Effect”. English. In: *MultiMedia, IEEE* 19.2 (Feb. 2012), pages 4–10. ISSN: 1070-986X. DOI: 10.1109/MMUL.2012.24 (cited on page 136).
- [304] T. Zhao, M. Aggarwal, R. Kumar, and H. Sawhney. “Real-time wide area multi-camera stereo tracking”. English. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Edited by C. Schmid, S. Soatto, and C. Tomasi. Volume 1.

San Diego, CA: IEEE Computer Society, 2005, pages 976–983. ISBN: 0769523722. DOI: 10.1109/CVPR.2005.296 (cited on pages 21, 23, 26).