3D Real Time Object Recognition

DISSERTATION

zur Erlangung des akademischen Grades

Dr. rer. nat.
im Fach Informatik

eingereicht an der
Mathematisch-Naturwissenschaftliche Fakultät
der Humboldt-Universität zu Berlin

von
M.Sc. Konstantinos Amplianitis

Präsidentin der Humboldt-Universität zu Berlin:
Prof. Dr.-Ing. Dr. Sabine Kunst

Dekan der Mathematisch-Naturwissenschaftliche Fakultät:
Prof. Dr. Elmar Kulke

Gutachter:

1. Prof. Dr. rer. nat. Ralf Reulke
2. Prof. Dr.-Ing. Peter Eisert
3. Prof. Dr. rer. nat. Andreas Tewes

eingereicht am: 14.06.2016

Tag der mündlichen Prüfung: 24.02.2017

Abstract

Object recognition is a natural process of the human brain performed in the visual cortex and relies on a binocular depth perception system that renders a three-dimensional representation of the objects in a scene. Hitherto, computer and software systems are been used to simulate the perception of three-dimensional environments with the aid of sensors to capture real-time images. In the process, such images are used as input data for further analysis and development of algorithms, an essential ingredient for simulating the complexity of human vision, so as to achieve scene interpretation for object recognition, similar to the way the human brain perceives it.

The rapid pace of technological advancements in hardware and software, are continuously bringing the machine-based process for object recognition nearer to the inhuman vision prototype. The key in this field, is the development of algorithms in order to achieve robust scene interpretation. A lot of recognisable and significant effort has been successfully carried out over the years in 2D object recognition, as opposed to 3D.

It is therefore, within this context and scope of this dissertation, to contribute towards the enhancement of 3D object recognition; a better interpretation and understanding of reality and the relationship between objects in a scene. Through the use and application of low-cost commodity sensors, such as Microsoft Kinect, RGB and depth data of a scene have been retrieved and manipulated in order to generate human-like visual perception data. The goal herein is to show how RGB and depth information can be utilised in order to develop a new class of 3D object recognition algorithms, analogous to the perception processed by the human brain.

This dissertation presents my original work for the simulation of human vision in 3D objection recognition, focusing in the following three areas:

3D Human Recognition: The first area addresses the problem of localisation and spatial extent determination of a human in three-dimensional space. To this end, a Conditional Random Field (CRF) pairwise energy function is defined for the segmentation task using features from both RGB and depth space. Furthermore, the maximum a-posteriori (MAP) labelling is determined in polynomial time by minimising the energy function with the use of graph cuts. The novelty of this proposed approach is that no user interaction is required for determining the segmentation, as opposed to related work in the field. Moreover, the

segmentation is computed within the detection box, for which the latter is determined using HOG-based features. In conclusion, my results and findings are then compared against state–of–the–art approaches.
3D Human Motion Understanding: In the second part, a new approach is introduced for capturing and tracking the shape variations of a human instance in RGBD space. The proposed methodology consists of two components: (1) a workflow that enhances the accuracy of an existing octree-based foreground estimation algorithm in order to determine the shape of a human body and (2) the use of the Minimum Volume Enclosing Ellipsoid (MVEE) algorithm for capturing the spatio-temporal changes of the moving object in a 3D scene.
Preparatory work for a Potential multi-Kinect object recognition system: Finally, in the last part of this dissertation an evaluation workflow is presented for assessing the reliability of merging point clouds generated from different Kinect sensors. The proposed three-step evaluation pipeline, could be very useful for future object recognition applications; based on multiple Kinect sensors, when accurate combination of 3D datasets is required from different sensors.

Zusammenfassung

Die Objekterkennung ist ein natürlicher Prozess im Menschlichen Gehirn. Sie findet im visuellen Kortex statt und nutzt die binokulare Eigenschaft der Augen, die eine dreidimensionale Interpretation von Objekten in einer Szene erlaubt. Kameras ahmen das menschliche Auge nach. Bilder von zwei Kameras, in einem Stereokamerasystem, werden von Algorithmen für eine automatische, dreidimensionale Interpretation von Objekten in einer Szene benutzt.
Die Entwicklung von Hard- und Software verbessern den maschinellen Prozess der Objekterkennung und erreicht qualitativ immer mehr die Fähigkeiten des menschlichen Gehirns. Das Hauptziel dieses Forschungsfeldes ist die Entwicklung von robusten Algorithmen für die Szeneninterpretation. Sehr viel Aufwand wurde in den letzten Jahren in der zweidimensionale Objekterkennung betrieben, im Gegensatz zur Forschung zur dreidimensionalen Erkennung.
Im Rahmen dieser Arbeit soll demnach die dreidimensionale Objekterkennung weiterentwickelt werden: hin zu einer besseren Interpretation und einem besseren Verstehen von sichtbarer Realität wie auch der Beziehung zwischen Objekten in einer Szene. In den letzten Jahren aufkommende low-cost Verbrauchersensoren, wie die Microsoft Kinect, generieren Farb- und Tiefendaten einer Szene, um menschenähnliche visuelle Daten zu generieren. Das Ziel hier ist zu zeigen, wie diese Daten benutzt werden können, um eine neue Klasse von dreidimensionalen Objekterkennungsalgorithmen zu entwickeln - analog zur Verarbeitung im menschlichen Gehirn.
Diese Dissertation präsentiert meine Arbeit zur Simulation von menschlicher Wahrnehmung in dreidimensionaler Erkennung, fokussiert auf die drei folgenden Gebiete:
Dreidimensionale Erkennung von Menschen: Das erste Teilgebiet behandelt die Problematik des Lokalisierens und der Erkennung räumlicher Ausdehnung von Menschen im dreidimensionalen Raum. Dafür wrd eine Conditional Random Field (CRF) Energiefunktion definiert, die der Segmentierung dient und Eigenschaften von Farb- und Tiefenraum benutzt. Zusätzlich wird die Maximium–A–Posteriori Klassifikatorzuordnung in polynomieller Zeit generiert, mithilfe von Graph Cuts. Die Neuheit bei diesem Verfahren besteht darin, dass keinerlei Benutzerinteraktion benötigt wird, im Gegensatz zu anderen Verfahren. Weiterhin wird die Segmentierung innerhalb einer Detektionsbox vollzogen,

welche mittels HOG-basierter Eigenschaften bestimmt wird. Abschließend werden meine Ergebnisse mit den Stand der Technik verglichen.

3D Human Motion Understanding: Im zweiten Teil wird ein neuer Ansatz vorgestellt, der die Erscheinungsvarianten von Menschen in einem Farb- und Tiefenraum erfassen und verfolgen kann. Die vorgeschlagene Methode besteht aus zwei Komponenten: erstens, aus einem Prozess der die Genauigkeit eines existierenden Octree-basiertem dreidimensionalen Hintergrundschätzer verbessert, um die Form eines Menschen zu erkennen. Zweitens, um aus der Verwendung des Algorithmus Minimum Volume Enclosing Ellipsoid (MVEE) räumlich-zeitliche Veränderungen eines sich bewegenden Objektes im dreidimensionalen Raum zu erfassen.

Vorbereitende Arbeiten für ein potentielles Multi-Kinect-Objekt-Erkennungssystem: Schließlich wird ein Arbeitsablauf für die Evaluierung der Zuverlässigkeit von zusammengeführten Punktwolken, verschiedener Kinect-Sensoren, präsentiert. Die Evaluierungspipeline kann für zukünftige Objekterkennungsapplikationen sehr nützlich sein. Sie basiert auf einer akkuraten Kombination von 3D-Daten mehrerer Kinect-Sensoren.

To Nikolaos and Florentia, my parents

Acknowledgements

First and foremost, I would like to express my special appreciation and thanks to my supervisor Prof. Dr. rer. nat. Ralf Reulke for his guidance and support throughout my doctoral studies. His continued efforts of engagement in multi-disciplined industrial projects helped me immensely in obtaining a better understanding of expectations in the field and enhancement of my skills. Above all, a person of understanding and encouragement, helped in bringing this dissertation to its end.

I am also indebted to many of my colleagues during my lengthy stay at the Humboldt University of Berlin. Specifically, I would like to thank Michele Adduci, for his experience in software architecture, an essential contribution for designing, implementing and testing parts of my proposed work, Martin Misgaiski-Haß for setting up the test field and organising the mounting of the sensors, Jan Hieronymus for reviewing the calibration and relative transformation part of my dissertation and Dominik Rueß for his translation services of my abstract, from English to German.

Furthermore, I would also like to express my gratitude to Dr.-Ing. Ronny Hänsch from the Computer Vision and Remote Sensing Group of the Technische Universität Berlin, for the excellent collaboration we had during my last year of research. His scientific advices, insightful discussions and suggestions played an integral role for finalising my research work. Moreover, I'd like to also thank him very much for his time to evaluate my data with his Projection-Based Random Forest (ProB-RF) framework and its integration in the latest part of my work.

A special gratitude also goes to my friend Dr. Evangelos Golias, who patiently revised my dissertation and resulted with some very helpful comments.

Most importantly, none of this would have been possible without the support of my parents. My parents, to whom this dissertation is dedicated to, have been a constant source of love, concern and strength throughout all these years.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

Mathematical symbols

$f(x)$	Function f of input x
$f(x; w)$	Function f of input x, parametrised by w
$\|\|\mathbf{x}\|\|$	L^2 norm or Euclidean norm of a vector $\mathbf{x}$
$\|\boldsymbol{x}\|$	Cardinality (number of elements) of a set (or clique) $\boldsymbol{x}$
abs(x)	Absolute value of x
$\det(\mathbf{C})$	Determinant of a matrix $\mathbf{C}$
$\sum_{i=1}^{n} \mathrm{x}_i$	Summation of all elements $\mathrm{x}_1 + \mathrm{x}_2 + \cdots + \mathrm{x}_n$ of a vector $\mathbf{x}$
$\prod_{i=1}^{n} \mathrm{x}_i$	Multiplication of all succeeding terms $\mathrm{x}_1 \cdot \mathrm{x}_2 \cdot\ \ldots\ \cdot \mathrm{x}_n$ of a vector $\mathbf{x}$
$\bigcup\limits_{i \in I} \mathbf{x}_i$	Union of a collection of sets $\{\mathbf{x}_i : i \in I\}$
$x \perp\!\!\!\perp y$	Conditional independence between random variables x and y
$\langle \mathbf{x}, \mathbf{y} \rangle$	Inner product between vectors $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{x} \times \mathbf{y}$	Cross product between vectors $\mathbf{x}$ and $\mathbf{y}$
$\triangle(\boldsymbol{x}, \boldsymbol{y})$	Loss (difference) between random fields $\boldsymbol{x}$ and $\boldsymbol{y}$
$p(\boldsymbol{x})$	Join probability distribution of a random field $\boldsymbol{x}$
$p(\boldsymbol{x} \mid \boldsymbol{y})$	Probability of a configuration $\boldsymbol{x}$ to occur given configuration $\boldsymbol{y}$
$p(\boldsymbol{x} \mid \boldsymbol{y}; w)$	Probability of a configuration $\boldsymbol{x}$ to occur given configuration $\boldsymbol{y}$, parametrised by w
$\mathbf{1}[i \neq j]$	Indicator function. Returns the value 1 if $i \neq j$ or 0 otherwise
$\mathbf{X} \cap \mathbf{Y}$	The probability that events $\mathbf{X}$ and $\mathbf{Y}$ both occur.
$\mathbf{X} \cup \mathbf{Y}$	The probability that events $\mathbf{X}$ or $\mathbf{Y}$ occur.

$\mathbf{X} \subseteq \mathbf{Y}$	Set $\mathbf{X}$ is a subset of set $\mathbf{Y}$
$\mathbf{X} \oplus \mathbf{Y}$	Exclusive disjunction (XOR) is a logical operation that outputs true only when $\mathbf{X}$ and $\mathbf{Y}$ inputs differ

Parameters

σ	Bandwidth parameter
σ_0	Standard deviation of the unit weight
x_0, y_0	Coordinates of the principal point
c_x, c_y	Focal length of the camera in terms of pixel dimensions
k_1, k_2, k_3	Radial distortion parameters
p_1, p_2	Decentering (tangential) distortion parameters
X_0, Y_0, Z_0	Translation parameters of a sensor
$r_{11}, \ldots, r_{33}$	Rotation elements of a 3×3 rotation matrix $\mathbf{R}$

Variables

$\mathcal{V}$	Number of vertices in a graph
$\mathcal{E}$	Number of edges in a graph
$\mathcal{G}$	Graph representation consisting of nodes and edges $\{\mathcal{V}, \mathcal{E}\}$
$\mathcal{X}$	Random field (complete state space)
$\boldsymbol{X}$	Set of random variables
X	Random variable
$\boldsymbol{x}$	Realisation of $\boldsymbol{X}$
x	Realisation (value) of $\boldsymbol{x}$
$\bar{x}$	Complementary variable $\bar{x} = 1 - x$
$\mathscr{S}$	Set $\mathscr{S}$ contains many subsets of sets

Vectors and Matrices

x	2D point

X	3D point and matrix representation
$\hat{\mathrm{x}}$	Optimised 2D vector
$\hat{\mathrm{X}}$	Optimised 3D vector or matrix

Acronyms

2D	2-Dimensional
3D	3-Dimensional
BA	Bundle Adjustment
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DLT	Direct Linear Transformation
DOF	Degrees of Freedom
DPM	Deformable Part Model
EO	Exterior Orientation
FOV	Field of View
FPS	Frames per Second
GCP	Ground Control Point
HOG	Histogram of Orientated Gradients
HSV	Hue-Saturation-Value colour model
ICP	Iterative Closest Point
IO	Interior Orientation
IOU	Intersection Over Union
IP	Infrared Projection
iif	if and only if
iid	independent and identically distributed
LM	Levenberg-Marquardt
LS	Least-Squares

MAP	Maximum a Posteriori
PCA	Principal Component Analysis
P*n*P	Perspective-n-Point
ProB-RF	Projection-Based Random Forest
MRF	Markov Random Field
MVEE	Minimum Volume Enclosing Ellipsoid
RGB	Red–Green–Blue
RGBD	Red–Green–Blue–Depth
RMS	Residual Mean Squared Error
SVM	Support Vector Machine
SSVM	Structured Support Vector Machine

Chapter 1

Introduction

"Object recognition systems constitute a deeply entrenched and omnipresent component of modern intelligent systems."

– A. Andreopoulos and John K. Tsotsos [1]

1.1 Motivation and Objectives

Object recognition is the ability to perceive an object's physical properties, such as shape, colour and texture, based on some prior experience and knowledge of the object. The range of applications span from optical character recognition, to medical imaging, biometrics, defence and surveillance systems. The field of object recognition in information technology and specifically in computer vision originated in the early 60s', whereby scientists began to investigate approaches and develop algorithms in order to distinguish and recognise simple shapes in images. Some pioneers in the field, such as Roberts [2], Lowe [3] and Biederman [4], marked the beginning of image-based object recognition for intelligent systems. Likewise, a considerable amount of work has been undertaken in order to improve and refine already esta blished methodologies in the field. For example, a method originally devised in 1962 by Hough [5] for the field of particle physics as a means of recognising basic geometric objects, such as lines and arcs, known as the Hough transform, was later on extended by Duda and Hart [6] to the so-called "Generalised Hough transform" for recognising more generalised objects.

However, an object recognition system is more than just recognising static objects in images. An ideal system should be able to recognise non-rigid objects undergoing temporal-dependent shape deformations. Yet, despite the progress made in algorithms and hardware technology, existing methods in the field are not able to capture the multiplicity of the available representations of some deformable objects such as humans; hence, leading contemporary recognition systems to erroneous and non-robust predictions. An accurate machine-based vision system should therefore be able to cope with the following process problems: viewpoint perceptiveness, illumination changes, object occlusion, object scaling, deformation, background cluttering and intra-class variations. As a consequence, dealing simultaneously with the above-mentioned issues lead us to the following conclusion: *vision is hard.*

The human brain is a prototypical system that can handle the aforementioned processes for object recognition, effectively and promptly. Understanding the intricate functionalities of the human brain is a complicated subject that requires interdisciplinary knowledge from many research fields and is beyond the scope of this dissertation. Nevertheless, it should be noted that approaches presented in this dissertation share common ground with our brain perception mechanism. That is, the recognition of objects as a whole based on already known patterns, a process that is naturally employed by our brain (Desolneux et al. [7]).

In the field of machine vision systems, a lot of focused effort has been conducted in two-dimensional (2D) perception and analysis of an object. However, depth perception is crucial if one wants to approach the effectiveness of the human visual system. To this end, a depth camera device is used in order to capture a scene and its objects in a three dimensional space. Low cost commodity depth sensors such as the Microsoft Kinect and Asus Xtion Pro Live are ideal for such purposes and have been widely used in the fields of computer vision, machine learning and robotics. One great advantage of these sensors is that they provide real-time RGBD information, even for untextured environments. The depth (or disparity) map is created by receiving and analysing a speckle pattern (near–infrared light) emitted by the infrared projector in the infrared image. This is the basic principle behind *structured light* sensors.

The aim of this dissertation is three-fold: firstly, to develop an approach that can determine the size and position of a human in 3D space, secondly to capture the motion of a human in 3D space and finally assess the quality of merging point cloud data acquired from multiple Kinect-like 3D data. Therefore, my goal is to employ realistic[1] data, analogous to the processing made by the human perception system.

Applications. The work presented in this dissertation can be useful for a variety of applications. Specifically, the RGBD human recognition system introduced in Chap. 4 can be used primarily for collecting a large set of three dimensional training data, an essential requirement for the purposes of learning a supervised object recognition system.

The research proposed within Chap. 4, provides reliable results up to ≈ 4 m and even though it has been only evaluated with a Kinect sensor, it can potentially be adapted to other sensors with a higher detection range, such as the SwissRanger 4000 or CamCube 2.0 TOF sensor. Furthermore, the work presented in Chap. 5 could lead to the following applications: Firstly, the proposed 3D background estimation algorithm could be used for an indoor surveillance system providing concurrent metric information of the moving object. Secondly, monitoring and tracking the spatio-temporal changes of a human figure; using a minimum bounding ellipsoid that could also be useful for classifying the behaviour of a normal or abnormal person. This algorithm has been developed as a proof of concept, consequently further refinements and enhancements are potentially possible. Finally, the work in Chap. 6 is targeted for potential RGBD-based multi-Kinect human recognition systems that require fusing 3D data from multiple Kinect sensors. These findings could be further useful in understanding the advantages and disadvantages of a multi-Kinect RGBD system.

[1] Realistic data contain information about colour, shape, relations between objects but also their time-dependent variations.

1.2 Contributions

The principal contributions of this dissertation may be summarised as follows:

- The human visual system is able to perceive an object's physical properties, such as its location and size through a life learning interaction/experience with the object. For a machine perception system, reality can be represented through different sources of knowledge, such as an image or depth perception. The Kinect sensor is able to provide both RGB and depth data in real-time. Utilising this information, I propose an approach for detecting and segmenting a human instance in RGBD space and this work could potentially help bridge the gap between human perception and machine vision.

  More specifically, localisation was found by evaluating the performance of two object detectors: the well known HOG detector, introduced by Dalal and Triggs [8] and an improved version of the deformable part model (Felzenszwalb et al. [9]) introduced by Dubout et al. [10]. The main difference between these two detectors is that the former uses a single HOG model or filter to learn a human shape, whereas the latter uses a star-like configuration of body-part models. Depending on the performance of each detector, the part of the object to be processed within the detection box may vary. The main contribution comes in the second part, whereby the segmentation decision is given, based on a rich set of RGB and depth features defined in a Conditional Random Field probabilistic framework. The maximum a posteriori (MAP) labelling is found by minimising a pairwise energy function using graph cuts. One can then implement the one-slack Structured Support Vector Machine algorithm for choosing the weights of the energy function which give the lowest testing error.

- Object recognition is a process for detecting instances of semantic objects from camera data. However, non-rigid objects such as humans do not remain static in time, but undergo significant spatio-temporal deformations. This means that the task of object recognition could be extended to the task of understanding object motion. Working entirely with humans in RGBD space, I propose using a minimum bounding ellipsoid as a mathematical figure for approximating the movement of the person in the scene. Compared to a sphere, an ellipsoid has more degrees of freedom, which allows the capturing of larger shape deformations. All information regarding the human ellipsoid is contained in a 3×3 variance-covariance matrix. The deduced information from the matrix is smoothed using a Kalman filter [11] for performing the tracking of the shape variations.

  Although the segmentation results from the previous contribution are promising, they are not able to provide the complete shape of the human due to the restrictions of the detection box. Thus, I considered using a 3D foreground estimation approach which is able to capture the complete human shape but also remain invariant to environmental conditions. The approach of Kammerl et al. [12] compares the octree representations of the background and the current cloud for detecting spatial changes in the current frame and assigning these changes to the foreground. However, depending on the size of the leaf node, the amount of noise in the foreground may differ. Therefore, I propose a pipeline for capturing and filtering noisy blobs in the point cloud, resulting in a clean foreground mask. The ellipsoid in this case is able to capture the complete deformation of the human shape.

- Object recognition in RGBD space is a very challenging but an essential task for many applications, such as surveillance systems, scene understanding and automatic driving assistant systems (ADAS). While RGB and depth based systems have drawn considerable attention from the computer vision and machine learning communities, little work has been done in combining RGBD information from multiple sensors. Existing approaches are restricted in using either a pair of Kinect sensors with significant overlapping between the point clouds or combining the RGB information from several Kinect sensors. In view of the above, I have therefore come to the conclusion that there is no evidence of finished work that points out the benefits and restrictions/drawbacks of a potential multi-Kinect RGBD-based object recognition system. Thus, I propose an evaluation pipeline that would provide reliable information for the accuracy of merging point clouds generated from a network of Kinect-like sensors.

  The evaluation procedure consists of three stages: (a) transforming all sensors into a global coordinate system using Perspective–*n*–Point algorithms, (b) optimising the exterior orientation of the sensors through a bundle block adjustment and (c) minimising the geometric error between different views by sequentially aligning all point clouds using the ICP algorithm. Every step of the process is extensively evaluated, highlighting its main advantages and disadvantages. The outcome of this work could be useful for future development in the multi-Kinect object recognition field, when a resurgent need may arise to combine 3D data from several sensors.

1.3 Outline of the Dissertation

The structure of the dissertation is organised as follows:

Chapter 2: Related Work. This chapter presents a brief overview on recent developments in the research areas involved in the current dissertation. Each of the contributions is treated independently and is accompanied with its own related work in the field. Section 2.1 makes a literature survey on detecting and segmenting human instances in RGB and RGBD space. Subsequently, Sect. 2.1 contains related work for monitoring and tracking human instances in RGBD and finally, Sect. 2.3 presents methods that have been developed that require fusing information from multiple Kinect-like RGBD system.

Chapter 3: Conditional Random Fields, Inference and Learning. Assuming that the reader has no prior knowledge on probabilistic graphical models, the purpose of this chapter is to introduce some basic concepts in graphical models, with the prospect of understanding the principles of Conditional Random Fields (CRF). This chapter lays the foundation of knowledge that is required for understanding the work presented in the following chapter.

Chapter 4: Human Recognition in RGBD. This chapter presents an approach for detecting and segmenting human instances in RGBD. The detection performance is evaluated using a single Histogram of Oriented Gradients (HOG) feature detector introduced by Dalal and Triggs [8] and a star-like part-based HOG feature representation with individual part scaling, introduced from Dubout et al. [10] and its based on the Deformable Part Model approach of Felzenszwalb et al. [9]. In order to determine the spatial extent of the person within the detection box, my approach makes use of a rich set of RGBD features modelled within a pairwise Conditional Random Field energy function. The most probable labelling is found by minimising the energy function using graph cuts. The chapter concludes by providing some qualitative and quantitative results of the proposed method but also comparison results against state–of–the–art approaches.

Chapter 5: Human Motion Estimation and Tracking in RGBD. Here, a framework is introduced for monitoring and tracking human instances in RGBD space. My results are divided into two parts: In the first part, I propose an improvement over the approach of Kammerl et al. [12] that takes care in removing noisy blobs from the foreground produced by the existing algorithm. In the second part, the result from the previous step is given to the minimum volume encapsulated ellipsoid (Moshtagh [13]) for capturing the spatio-temporal changes of the human motion. The noise in the observation data extracted from the variance-covariance matrix of the ellipsoid has been removed using a Kalman filter [11].

Chapter 6: Towards a Multi Camera 3D Object Recognition System. In this chapter I propose a workflow for assessing the reliability of merging point clouds generated from a network of Kinect-like RGBD data. The working pipeline consists of three concrete stages: (a) The orientation of all sensors in a global coordinate system using Perspective–*n*–Point algorithms, (b) the optimisation of the exterior parameters of the sensors by solving a bundle adjustment system and finally (c) the minimisation of the geometric error between pairs of point clouds using the ICP algorithm.

Chapter 7: Conclusions and Future Work. This chapter concludes the dissertation by discussing the overall contribution to in the field, pointing out the limitations of the methods used and proposing directions for future research work.

1.4 Research Publications

According to Paragraph § 7 of the doctorate regulations of the Faculty of Mathematics and Natural Sciences of the Humboldt University of Berlin, the vast majority of the results in the current dissertation have been published in double blind peer reviewed conferences and workshops, proving the originality of the presented work.
Results presented in the current dissertation have been published in the following conferences and workshops:

- International Conference on Computer Vision Theory and Applications (VISAPP)
- The International Society for Photogrammetry and Remote Sensing (ISPRS)
- 3D-NordOst, Application-oriented Workshop on Measuring, Modelling, Processing and Analysis of 3D-Data

Chapter 2

Related Work

2.1 Object Recognition in 2D and 3D Space

This section presents a brief overview on recent 2D and 3D object recognition algorithms that determine the position and spatial extent of the objects of interest in a scene.

Ladický et al. [14] approached this problem by combining object detectors with Conditional Random Fields (Lafferty et al. [15]), jointly estimating the class category, location and spatial extent of objects/regions in a scene. Their proposed submodular energy function is based on unary, pairwise and higher order potential terms combined with the hypothesis results of the deformable part model object detector, introduced by Felzenszwalb et al. [9]. The maximum a posteriori (MAP) labelling was found by minimising the proposed higher order energy function using swap-making algorithms (Boykov et al. [16]). This work was later on extended by the same authors for approaching the problem of human instance segmentation in a video stream [17]. Specifically, they proposed a CRF energy function for integrating instant level information such as shape prior and exemplar histograms, biasing the segmentation towards human shape. Incorporating higher level image representations, Shu et al. [18] introduced a method which improves generic detectors and iteratively refines the object region from the background using a superpixel-based Bag-of-Words model (Csurja et al. [19]). Furthermore, Hariharan et al. [20] was the first to present a Convolutional Neural Network approach for simultaneously detecting and segmenting objects in an image. Their algorithm is based on classifying region proposals using features extracted from both the bounding box of the region and the region foreground, integrated in a jointly trained CNN.

In the RGBD domain, Lai et al. [21] proposed a view-based approach for segmenting objects in a point cloud generated by a depth sensor. A sliding window detector trained from different object views was used for assigning class probabilities to every image pixel. Then, they performed an MRF inference over the projected probabilities in voxel space, combining cues from different views for labelling the scene. Moreover, Teichman et al. [22] proposed a semi-automatic approach for segmenting deformable objects in RGBD space, providing an initial seed as a prior hard constraint for inferring the segmentation. His approach makes use of a rich set of features defined in RGBD space. Most recent work in the field is the one of Gupta et al. [23] who studied the problem of object detection and segmentation in RGBD by combining an RGB feature-based CNN with a depth feature-based CNN, fed in an SVM classifier.

2.2 Human Motion Analysis and Tracking in 3D Space

This section presents an overview on recent approaches on human motion analysis and tracking based on depth and pure 3D information. Although very limited work has been done in using mathematical shapes to express a human motion, related work could also involve approaches that try to capture the human motion using other sources of information such as features.

According to Chen et al. [24] and Aggarwal et al. [25], depth based approaches can be further divided into *space time* approaches and *sequential* approaches. The difference between these two categories is that space time approaches make use of features (local or global) without modelling any temporal dynamics of the object, whereas space time approaches learn the dynamics of an object based on local features computed within a sequence. However, using local and global features in any of the two categories can lead to erroneous results. For example, depth sequences containing large occlusions between the objects may not be reliable for learning global features for a human motion recognition system. Furthermore, depending on the quality of the depth sensor but also of the scene, the amount of unknown areas in the depth map may vary. Thus, applying RGB based features on a depth image will not deliver satisfactory results. These problems stimulated researchers to develop depth-based features which are highly discriminative and robust against occlusions.

Li et al. [26] presented a human recognition system that uses an action graph for modelling the dynamics of the actions and a bag of 3D points representing a set of salient postures. Orefej et al. [27] introduced a depth feature known as the 4D Histogram of Oriented Normals descriptor (HON4D), capable of capturing complex joint shape-motion cues at a pixel-level. A depth sequence is described based on the distribution of the surface normal orientation in the 4D space of time, depth and spatial coordinates. The HON4D feature is then built by creating 4D projectors which quantise the 4D space, representing possible directions for the 4D normal. Another related work was the one of Wilson et al. [28] who presented the so-called Space-Time Occupancy Patterns (STOP) feature descriptor for performing human action recognition from sequences of depth images. The method works by dividing the space and time axes into equally sized segments, defining a 4D grid for every depth map sequence. The great advantage of STOP feature is that it is able to preserve spatio-temporal contextual information between space-time cells, allowing to also accommodate intra-action variations. Wang et al. [29] introduced a fast-to-train semi-local feature, called Random Occupancy Pattern (ROP). It is based on a sampling scheme that effectively explores an extremely large sampling space. A sparse coding approach is then used for encoding these features in the sample space. Furthermore, Shotton et al. [30] introduced two approaches that predict the 3D position of all body joins from a depth image without using any temporal information. The first method introduces a per-pixel classification of different body parts whereas in the second approach they directly regress the position of the body parts. Both methods can run in real-time using simple depth features and parallelised decision forests. This work was also commercialised for human-machine interaction games performed with the Microsoft Kinect console. More recent work in the field, Rafi et al. [31] proposed a human pose estimation approach based on a semantic occlusion model learned by a regression forest classifier.

Similar approaches have been introduced in 3D but are quite limited compare to the depth-based approaches. Specifically, Buys et al. [32] presented an easy-to-train human pose recognition system which combines both RGB and depth features. Also, Hegger et al. [33] proposed a 3D feature descriptor based on Local Surface Normals (LSN) which is capable of detecting human poses under severe occlusions. Features are learned in a supervised manner and partial occlusions are detected based on a top-down/bottom-up segmentation approach. Furthermore, Sigalas et al. [34] introduced a data-driven model-based method for 3D torso estimation from RGBD data. Starting with the detection of the face, the position of the shoulders is defined based on illumination, scale and pose invariant features on the RGB silhouette. Finally, the pose of the torso is found using 3D geometric primitives, put in a global optimisation scheme.

Interesting work was also introduced in the 3D object tracking literature: the Unscented Kalman Filter (Ziegler et al. [35]) and the Random Hypersurface Models (Baum et al. [36]) are some of the most recent developments in the area of 3D object tracking.

2.3 Multi-Sensor Human Recognition in RGBD

This section presents recent approaches for fusing 3D data from multiple Kinect sensors. However, it should be stressed that some approaches are not presented as an independent work but constitute a part of the proposed 3D object recognition system.

Schröder et al. [37] investigated the advantages and disadvantages of using multiple Kinect sensors in an indoor environment. The interference of the infrared lasers in space was solved using fast rotation disks, creating a time division multiple access (TDMA) scenario. They also developed an algorithm for evaluating the quality of the depth images, generated with and without their multiplexing approach. Tong et al. [38] proposed a scanning system for capturing 3D full human body models using multiple Kinect sensors. To eliminate the interference between their near-infrared emitter, two pair of sensors were used for scanning the upper and lower part of the body. A third sensor was placed on the opposite side of the body capturing its middle part. Different pair of point clouds were initially registered using a template-based registration approach. For eliminating the loop closure problem, a brute-force global solution was used for bringing all scans into the Iterated Closest Point (ICP) iteration loop. Florian et al. [39] introduced the so-called Random Hypersurface Models (RHMs), an extended object tracking modelling technique capable of tracking objects in 3D space. For a person walking in the scene, tracking is performed by observing a cylinder encapsulating the human figure from a network of four Kinect sensors. The observation data were placed in a measurement equation, smoothed using the Unscented Kalman Filter (Julier et al. [40]). Furthermore, Almazan et al. [41] developed a surveillance system for detecting and tracking people within an indoor environment using multiple Kinect sensors. Data extracted by each device was transformed into a world coordinate system using a plane-based technique. All moving 3D pixels (also known as voxels) were transformed in a "plan view" which monitors the activity of the people in the scene.

Finally, Michel et al. [42] presented a top-down solution for tracking the complete articulated movement of a human body from markerless visual observations acquired by two Kinect sensors. The complete tracking process was solved using stochastic optimisation techniques.

Chapter 3

Conditional Random Fields, Inference and Learning

3.1 Introduction

Many computer vision applications such as natural language parsing, Context-Free Grammars and image segmentation involve predicting a set of unobserved (latent) variables given a set of observed variables. Specifically, for the task of image segmentation, the goal is to partition the image into object classes (also known as segments), making an intuition that all pixels clustered under the same label also share similar properties. More precisely, a label from a predefined label set is assigned to every pixel in the image, taking into account (in its simplest case) the intensity of the pixel but also the intensity information of pixels lying in its close vicinity. A well suited approach for these tasks are Conditional Random Fields (CRF), which is a statistical modelling class used for structured prediction outputs.

This chapter is intended to provide the reader with a firm conceptual understanding of basic definitions and concepts in graph theory and probabilistic graphical models. For the latter, the primary focus is on understanding the basic principles on Conditional Random Fields, their inference and learning methods that are primarily used in Chap. 4. If the reader has no prior knowledge on the topic, it is highly recommended to read this chapter before moving on to Chap. 4

3.2 Preliminaries

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ represent a graphical model, expressed by a set of nodes $\mathcal{V}$ and edges $\mathcal{E}$. For every node $i \in \mathcal{V}$, let X_i correspond to a random variable assigned a value x_i from its state space $\mathcal{X}_i(x_i \in \mathcal{X}_i)$. If $\boldsymbol{X} = (X_i)_{i \in \mathcal{V}}$ denotes the joint distribution variable of the random field, then $\boldsymbol{x} = (x_i)_{i \in \mathcal{V}}$ is the realisation of $\boldsymbol{X}$. Every configuration $\boldsymbol{x} = (x_i)_{i \in \mathcal{V}}$ takes values from its state space $\boldsymbol{X}$ which is defined as the Cartesian product of the individual state spaces assigned to every random variable X_i, so that $\mathcal{X} = \prod_{i \in \mathcal{V}} \mathcal{X}_i$. For a subset of random variables, let $\boldsymbol{x_c} = (x_i)_{i \in C}$ represent a tuple of random variables defining a clique $c \in \mathcal{V}$. A clique is defined as a set of variables grouped together.

The probability distribution of a random variable X is denoted as $p(x)$ and the joint probability distribution over a set of random variables $\boldsymbol{X}$ is represented by $p(\boldsymbol{x})$. For consistency with the related literature in the current chapter, most definitions will use the realisation of the variables rather than the random variables themselves. Also, the words node and random variable are interchangeably used within the context.

3.3 Markov Random Fields

A *Markov Random Field*, also known as *Markov network*, is an undirected graphical model defined over a set of random variables satisfying a Markov property (see below). A Markov random field is similar to a Bayesian network in its representation of dependencies but differs in that Bayesian networks are directed and acyclic, whereas Markov fields are undirected and cyclic. For an undirected graph $\mathcal{G}$, a set of random variables $\mathbf{X}$ is said to form an MRF iif the following Markov properties are satisfied:

- **Pairwise Markov property:** Two non-adjacent random variables are said to be conditionally independent given all other variables:

$$\forall i \in \mathcal{V}, \forall j \in \mathcal{V},\ X_i \perp\!\!\!\perp X_j \mid X_{\mathcal{V} \setminus \{i,j\}}, \quad \text{if } \{i,j\} \notin \mathcal{E} \tag{3.1}$$

- **Local Markov property:** Every random variable is conditionally independent of all other variables given its neighbours[1]:

$$\forall i \in \mathcal{V},\ X_i \perp\!\!\!\perp X_{\mathcal{V}-\{i\}} \mid X_{\mathcal{N}_i} \tag{3.2}$$

  where $\mathcal{N}_i = \{j \mid \{i,j\} \in \mathcal{E}\}$ denotes all random variables X_j that are part of the Markov Blanket of random variable X_i and are connected by an edge $\mathcal{E}_{ij}$.

- **Global Markov property:** Any two cliques $\boldsymbol{X_{c_A}}$ and $\boldsymbol{X_{c_B}}$ are conditionally independent of all other cliques, iif a separating clique $\boldsymbol{X_{c_S}}$ exists:

$$\forall A \subseteq \mathcal{V}, \forall B \subseteq \mathcal{V}, \forall S \subseteq \mathcal{V},\ \boldsymbol{X_A} \perp\!\!\!\perp \boldsymbol{X_B} \mid \boldsymbol{X_S} \tag{3.3}$$

Finding the probability distribution over the complete random field is considered to be an intractable task. To resolve this, a class of Markov random fields exists that factorises the graph depending on its cliques. Specifically, a first order clique is represented by just one random variable, a second order (or pairwise) clique by a set of two random variables and a higher order clique by a set of more random variables. If a clique is not overlapping any other clique, it is known to be a *maximal* clique. According to the Hammersley-Clifford theorem [43], a family of such distributions can be represented as a *Gibbs distribution* in the following factorised form:

$$p(\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{c \in C} \psi_c(\boldsymbol{x_c}) \tag{3.4}$$

[1] For a grid like graph (*e.g.* representing an image) neighbourhood relationship can be either a 4- or 8-neighbourhood.

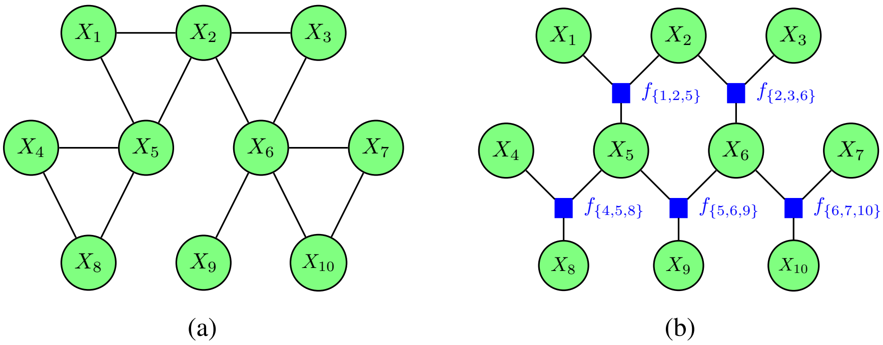


Figure 3.1: Example of a Markov Random Field and Factor Graph. The Markov Random Field in (a) is represented by the factor graph (b) and has factors of order 3. For the current graph configuration, the factor graph is defined on maximum cliques. (**Source:** The above figures have been adopted from the survey work of Wang et al. [44])

where $\psi_c(\boldsymbol{x_c})$ corresponds to a real value *potential function* of clique c and $Z(\boldsymbol{x})$ is known as the *normalised* or *partition function*, defined as:

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{x}\in\mathcal{X}} \prod_{c\in C} \psi_c(\boldsymbol{x_c}) \tag{3.5}$$

An MRF could also be represented by a *factor graph*, which uses additional nodes known as *factor nodes* to model the joint distribution in the graph. If $\mathcal{F}$ represents a set of all factors nodes in the graph, then the joint probability distribution over the complete random field could be expressed by the following form:

$$p(\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{f\in\mathcal{F}} \psi_f(\boldsymbol{x_f}) \tag{3.6}$$

where $\psi_f(\boldsymbol{x_f})$ is a potential function modelling a subset of random variables. An example of a Markov random field and the corresponding factor graph is gives in Fig. 3.1.
Inference of MRF involves finding a configuration $\boldsymbol{x} \in \mathcal{X}$ for which the probability distribution of $p(\boldsymbol{x})$ is maximum (denoted as $\hat{\boldsymbol{x}}$). This can be performed through a *Maximum a Posteriori* (MAP) estimation, expressed by:

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \; p(\boldsymbol{x}) \tag{3.7}$$

For a potential function $\psi_c(\boldsymbol{x_c}) : \mathbb{R} \to \mathbb{R}, \forall c \in C$, the corresponding *clique energy function* $\phi_c : \mathbb{R} \to \mathbb{R}$ is given by:

$$\phi(\boldsymbol{x_c}) = -\log \psi_c(\boldsymbol{x_c}) \tag{3.8}$$

Inserting 3.8 in 3.4, the probability distribution of $p(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}$ becomes:

$$p(\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp(-E(\boldsymbol{x})) \tag{3.9}$$

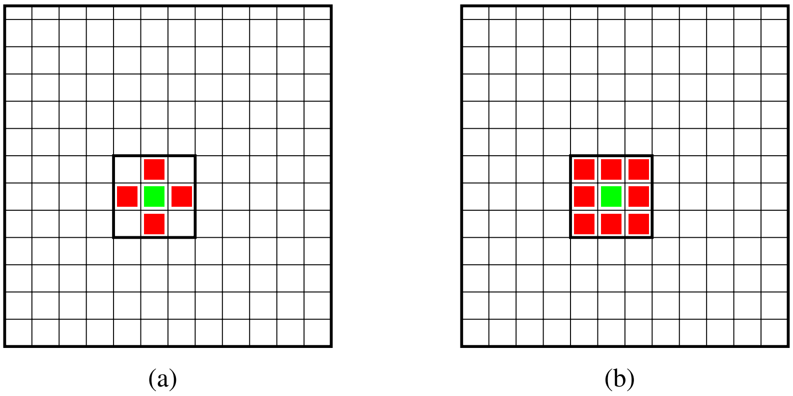


Figure 3.2: Pixel connectivity. Neighbourhood relationship of a random variable in a grid-like topological structure such as an image): (a) 4-neighborhood and (b) 8-neighborhood. The main random variable is represented by a green colour and all neighbourhood random variables by a red colour. (**Best viewed in colour**)

where $E(\boldsymbol{x})$ denotes the *energy function* of the MRF defined by the summation of all maximal cliques in the field, expressed by:

$$E(\boldsymbol{x}) = \sum_{c \in C} \phi_c(\boldsymbol{x_c}) \tag{3.10}$$

Due to the existence of the negative log function in the right part of Eq. 3.9, the MAP inference of $p(\boldsymbol{x})$ is equivalent to minimising $E(\boldsymbol{x})$, expressed as:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \; E(\boldsymbol{x}) \tag{3.11}$$

Therefore, the following equality should hold:

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) = \arg\min_{\boldsymbol{x} \in \mathcal{X}} E(\boldsymbol{x}) \tag{3.12}$$

Several methods has been proposed for minimising the energy function $E(\boldsymbol{x})$, but more attention is been given to approaches using graph cuts due to their computational performance and efficiency (refer to Sect. 3.6 for more information).

3.4 Pairwise MRF Energy Functions

Pairwise MRF energy functions strictly model cliques of order less than three. Specifically, for grid-like graphs (*e.g.* an image), the pairwise energy function consists of unary (also known as *singleton*) potentials functions $\phi_i(x_i)_{i \in \mathcal{V}}$ and pairwise potential functions $\phi_{ij}(x_i, x_j)_{\{i,j\} \in \mathcal{E}}$, defined over two neighbourhood random variables in a 4 or 8-neighbourhood system. Adapting the energy function 3.10 for pairwise relations, it becomes:

$$E(\boldsymbol{x}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{(i,j) \in \mathcal{N}_i} \phi_{ij}(x_i, x_j) \tag{3.13}$$

This form of an energy function is well established in the computer vision literature and has been used in a large variety of applications, such as segmentation (Kato et al. [45], Shi et al. [46], Björkman et al. [47]), object detection (Yin et al. [48], Colin et al. [49], Ghosh et al. [50]), 3D reconstruction (Choi et al. [51], Li et al. [52], Pan et al. [53]) and stereo processing (Sun et al. [54], Zhang et al. [55], Yamaguchi et al. [56]). Energy functions modelling up to pairwise relations have proven to work well for several computer vision problems. This form of relationship is considered as the most primitive type of interaction but also the most computationally efficient approach.

Every random variable is assigned to a physical quantity that is problem dependent. For an image segmentation task, a random variable representing a pixel can take a value from a predefined label set $\mathcal{L}$ containing a list of objects, or for an image denoising task, an intensity value within the range of 0-255. The *data likelihood* term encoded by the sum of the unary potentials, is also problem dependent. Given an RGB image representing an object and its background, one should assign a label class from the label set $\mathcal{L} = \{$"background","object"$\}$ to every pixel in the image. As shown initially by Boykov et al. [57] and later on by Rother et al. [58], a user can mark some regions (hard constraints/seeds) as a prior knowledge for the background and foreground classes, and build a Gaussian Mixture Model representing each of them. Then, every pixel in the image can be assigned a cost computed by the negative log-likelihood of the prior distributions of each class.

Furthermore, pairwise potentials also model *contextual constraints* between adjacent random variables defined within the local neighbourhood. One of the simplest contextual constraint is the *smoothness* constraint, enforcing that the states of all nodes should vary smoothly in the spatial domain. In the field of computer vision, one of the most fundamental decreasing costs used to define a pairwise potential is that of the *Potts model* [59], expressed by:

$$\phi_{ij}(x_i, x_j) = w(1 - \delta(x_i - x_j)) \tag{3.14}$$

where w is a weight coefficient that specifies the amount of penalisation between the pixels and $\delta(\cdot)$ is the *Kronecker delta* function which can only take the value 0 or 1. Several other variations of the Potts model exist, such as the truncated versions, in which the maximum cost assigned between two variables should not exceed a predefined value. A special, harder penalisation case of the Potts model is the *Ising model* [60], which can only take the value 0 or 1.

3.5 Conditional Random Fields

A Conditional Random Field (CRF), introduced by Lafferty et al. [15], is a discriminative undirected probabilistic graphical model used to predict the values of the latent (unobserved) variables given a set of observed variables. Modifying the previous notation, a set of random variables $\boldsymbol{x}$ representing the complete realisation of the random field, can be cloned into a second layer parallel to the first layer, following a one-to-one correspondence. Let the bottom layer describe the unobserved random field denoted by $\boldsymbol{y}$ and the top layer the observation field represented by $\boldsymbol{x}$ as depicted in Fig. 3.3. Every random variable $(y_i)_{i\in\mathcal{V}}$ in the unobserved layer should be assigned a class label from a predefined label

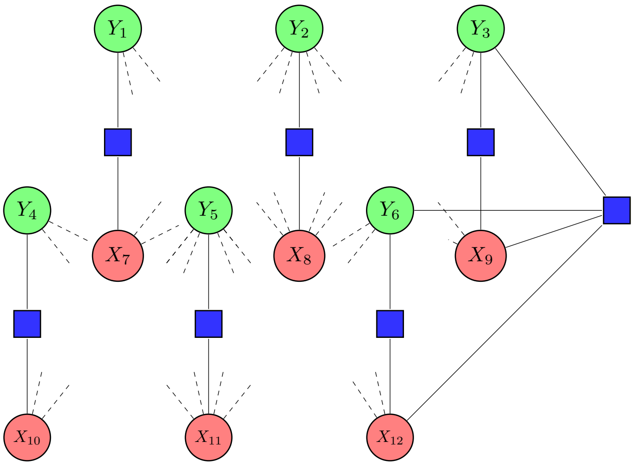


Figure 3.3: A grid like structure Conditional Random Field. The red nodes correspond to the observation variables (observation layer), the green nodes to the latent variables (unknown layer) and the blue rectangles to the potential functions modelling a unary or pairwise relationship. (**Best viewed in colour**)

space $\mathcal{Y}$ (previously denoted by $\mathcal{L}$). This label space, depending on the application, can take a discrete or continues number of classes. The most likely configuration of the unobserved random variables $\boldsymbol{y} \in \mathcal{Y}$ is revealed based on the observations $\boldsymbol{x}$ in the observation layer. In its mathematical representation, this form of relationship can be expressed by the conditional probability $p(\boldsymbol{y} \mid \boldsymbol{x})$, which can be read as "what it the probability of having a labelling $\boldsymbol{y} \in \mathcal{Y}$ given $\boldsymbol{x}$". The configuration $\boldsymbol{x}$ is also part of a state space $\mathcal{X}$, but due to its nature, an infinite number of solutions exist.
Similarly to an MRF, a CRF can be represented in a form of a Gibbs distribution as follows:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp(-E(\boldsymbol{x}, \boldsymbol{y})) \tag{3.15}$$

where $E(\boldsymbol{x}, \boldsymbol{y})$ corresponds to the energy function expressed by:

$$E(\boldsymbol{x}, \boldsymbol{y}) = \sum_{c \in C} \phi_c(\boldsymbol{x_c}, \boldsymbol{y_c}) \tag{3.16}$$

It is clear from 3.16, that no modelling over the probability distribution of the observed variables $\boldsymbol{x}$ exists, assuming a relaxation on the dependencies between them. Thus, a CRF can much easier model the joint probability distribution over the latent variables $\boldsymbol{y}$, given the observed variables $\boldsymbol{x}$. This is considered the main advantage of a CRF compare to an MRF. Also, all clique potentials $c \in C$ are data dependent, which provides a better interaction between the random variables in the clique.
In the context of computer vision, a CRF can be thought of as a two grid-like layer representation, where the bottom layer corresponds to a set of observed random variables $\boldsymbol{x}$ and the top layer to a set of latent variables $\boldsymbol{y}$ (see Fig. 3.3). For an image segmentation task, every pixel in the image will assign a value from the label set $\mathcal{Y}$ to the corresponding latent variable depending on the local interaction of the neighbourhood variables in the observation layer. The simplest form of pairwise modelling was introduced by Boykov et al. [57] for binary segmentation tasks, using *intensity contrast* and *spatial distance*

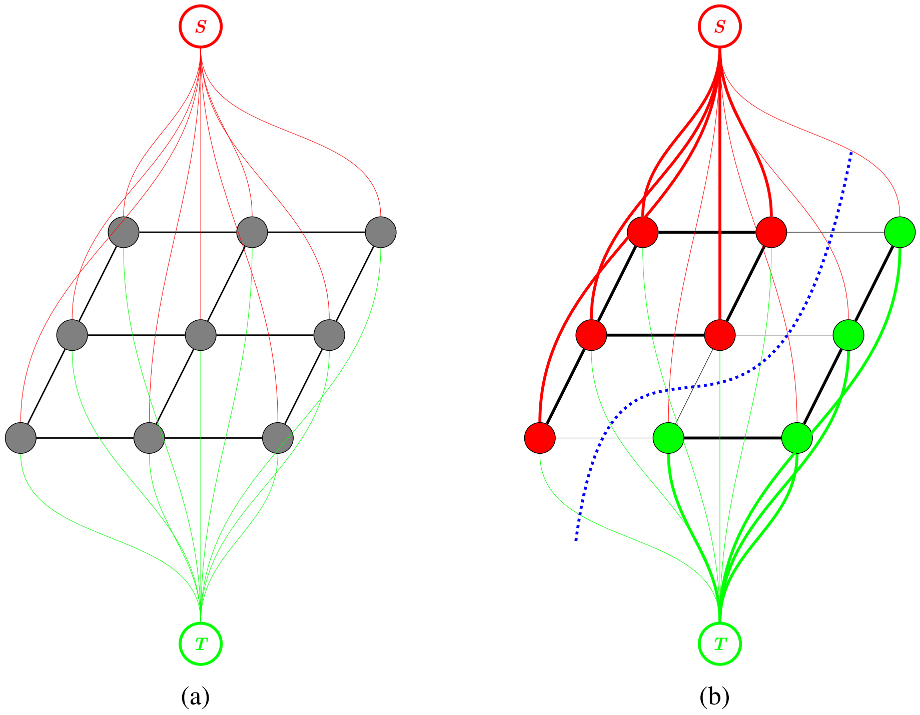


Figure 3.4: Graph representation with two terminal nodes and corresponding st-cut. (a) A graph $\mathcal{G}$ connected with two additional terminals called source and sink; (b) st-cut on graph $\mathcal{G}$. Red and green nodes correspond to a set of pixels grouped together due to the partition of the graph. The direction of the blue line defines the direction of the cut, considering all edges with minimum cost. (**Best viewed in colour**)

between neighbourhood pixels in an N-D image. This form of modelling is considered to be more reliable than the pairwise smoothness constraint of the Potts model [59]. Modifying the general form of the CRF energy function 3.16 to a pairwise case, results to the following form:

$$E(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \in \mathcal{V}} \phi_i(x_i, y_i) + \sum_{(i,j) \in \mathcal{N}_i} \phi_{ij}(x_i, x_j, y_i, y_j) \tag{3.17}$$

For the pairwise case, the CRF energy function 3.17 can take the following form:

$$E(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \in \mathcal{V}} \phi_i(\boldsymbol{x}, \boldsymbol{y}) + \sum_{(i,j) \in \mathcal{N}_i} \phi_{ij}(\boldsymbol{x}, \boldsymbol{y}) \tag{3.18}$$

Such a pairwise energy function is adopted in the work proposed in Chap. 4.

3.6 Graph Cuts

Graph cuts have been extensively used in the areas of computer vision and machine learning, solving a variety of problems in applications such as segmentation (Kohli et al. [61], Ladický et al. [14], Vineet et al. [17]), image restoration (Boykov et al. [16], Yan et al. [62]), stereo vision and 3D reconstruction (Kolmogorov et al. [63], Wang et al.

[64], Altantawy et al. [65]). They were first introduced in computer vision by Greig et al. [66], who proved that if an edge potential of two random variables defined in a discrete pairwise MRF energy function takes the form of an Ising model [60], then a exact solution is feasible in polynomial time[2] using graph cuts. This process is also known as the *s-t cut* problem.

As stated in the work of Kolmogorov and Zabih [67], the graph cut algorithm can find the global minimum of any arbitrary graph independently of its size and structure with a very low polynomial run time complexity. Even in cases where a global minimum is not achievable, it will return the best local minimum solution for the current energy function.

3.6.1 The Min-Cut/Max-Flow Algorithm

A graph $\mathcal{G}$, besides its nodes $\mathcal{V}$ and edges $\mathcal{E}$, can also contain some additional nodes called *terminals*. Each terminal node is assigned a label from a predefined label set and is initially linked to all nodes in the graph. In the simplest two label binary case, the constructed graph involves two terminals known as *source* and *sink*. For a binary image classification task, the source node can take the value 1 representing the "object" class, and the sink node the value 0 for representing the "background" class. A partition of the graph using the *min-cut/max-flow algorithm* (Boykov and Kolmogorov [68]) will create two strictly separable, non-overlapping clusters, where one cluster will contain a set of nodes representing the background class and the other cluster the foreground class.

Furthermore, the graph $\mathcal{G}$ consists of two additional sets of edges known as *n-links* and *t-links*. The n-links connect neighbourhood nodes connected by an edge and the t-links connect each node to both terminals. Thus, every node corresponding to a random variable $(X_i)_{i\in\mathcal{V}}$ has two t-links $\{X_i, S\}$ and $\{X_i, T\}$ and one n-link for every random variable $X_j \in \mathcal{N}_i$ that lies in the vicinity of $(X_i)_{i\in\mathcal{V}}$. Based on the previous information, the updated version of the graph incorporating the terminals and link edges can be expressed in the following way (Boykov et al. [57]):

$$\mathcal{V} = \mathcal{V} \,\cup\, \{S, T\} \tag{3.19}$$

$$\mathcal{E} = \mathcal{N} \bigcup_{i\in\mathcal{V}} \{\{X_i, S\}, \{X_i, T\}\} \tag{3.20}$$

Every edge $e \in \mathcal{E}$ in the graph is assigned a non-negative weight (or cost) w_e, computed by the pairwise potential function ϕ_{ij}.

The min-cut/max-flow algorithm tries to find a subset of edges $C \subset \mathcal{E}$ whose total cost is minimum. The induced graph $\mathcal{G}(C) = \{\mathcal{V}, \mathcal{E}\backslash C\}$ separates the terminals by excluding the C edges. As a result, all nodes will be assigned the label of the terminal they belong to.

According to Boykov and Kolmogorov [68], if the number of terminals added in the graph is two, then the min-cut/max-flow can be computed in low-order polynomial time. If more terminals are used (multi-label problem), the problem is NP-hard and can be converted into polynomial complexity using *move making* algorithms such as the $\alpha - expansion$ or

[2] An algorithm can be solved in polynomial time if the number of steps required to complete the algorithm is $\mathcal{O}(n)^k$, where k is a non-negative constant integer value, and n corresponds to the complexity of the input.

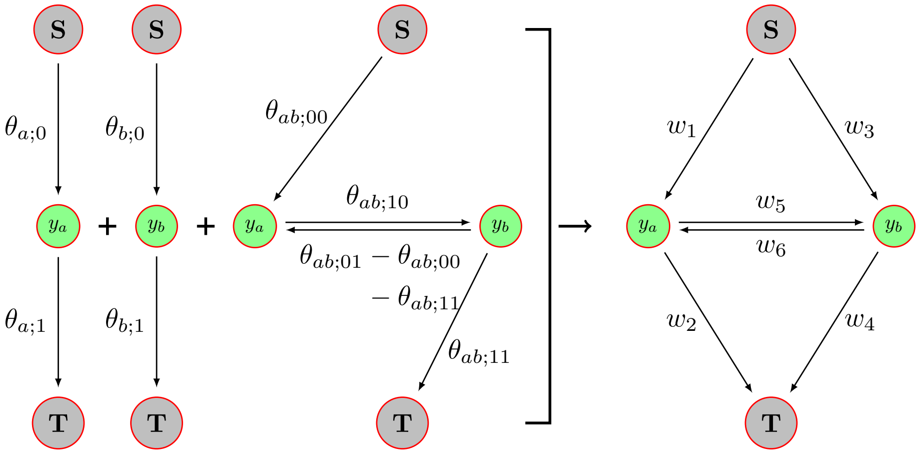


Figure 3.5: Graph representation of a Pseudo-Boolean submodular energy function. The figure shows the construction of a submodular energy function defined in the binary domain, adding its individual unary and pairwise terms. Every edge in the final graph is assigned a cost w, which is a summation of individual edge costs θ defined between the same nodes. Every configuration $\boldsymbol{y} \in \mathcal{Y}$ of an energy function $E(\boldsymbol{y})$ returns a different cost by the st-cut. Thus, the goal is to find the configuration $\boldsymbol{y} \in \mathcal{Y}$ for which the energy function $E(\boldsymbol{y})$ is minimum. (**Source:** The above figure has been reproduced with small modifications from the doctoral thesis of Kohli [71])

$\alpha\beta - swap$ making algorithms (Boykov et al. [16]). These approaches have been widely used in computer vision for solving multi-label image classification problems (Russell et al. [69], Kohli et al. [61], Huang et al. [70]) and are beyond the scope of this chapter.

3.6.2 Minimising Energy Functions using Graph Cuts

Any energy function satisfying the submodularity constraints (see Appx. A), can efficiently be solved in polynomial time using graph cuts. Minimising this form of energy functions involves minimising a sum of unary and pairwise functions expressed in the binary domain $\mathcal{Y} = \{0, 1\}$. For an MRF energy function $E(\boldsymbol{y})$, every configuration $\boldsymbol{y} \in \mathcal{Y}$ results a different cost by the st-cut. Thus, the objective is to find the configuration $\boldsymbol{y} \in \mathcal{Y}$ for which the energy function $E(\boldsymbol{y})$ is minimum. According to Kolmogorov [72], a pairwise binary submodular energy function can be represented in the following way:

$$\begin{aligned} E(\boldsymbol{y}; \theta) = \theta_{\text{const}} &+ \sum_{u \in \mathcal{V}, i \in \mathcal{Y}} \theta_{u;i} \mathbf{1}[y_u = i] \\ &+ \sum_{(u,v) \in \mathcal{E},\ (j,k) \in \mathcal{Y}} \theta_{uv;jk} \mathbf{1}[y_u = j] \mathbf{1}[y_v = k] \end{aligned} \tag{3.21}$$

where $\theta_{u;i}$ represents the penalty for assigning the value $i \in \mathcal{Y}$ to random variable y_u, $\theta_{uv;jk}$ is the penalty for assigning values $(j, k) \in \mathcal{Y}$ to random variables y_u and y_v respectively and $\mathbf{1}[y_u = y_v]$ is an indicator function[3] that takes the value one if the condition $y_u = y_v$ is satisfied and zero otherwise. The constant term θ_{const} is independent from the distribution

[3] This is the Iverson bracket representation of the indicator function, which is also used as an alternative to the $\delta_i(y_v)$ form.

of $\boldsymbol{y}$ and thus it is not involved in the minimisation process. This term ensures that the cost induced by the st-cut will never be zero.
Binary energy functions of the form 3.21 can also be represented as:

$$\begin{aligned} E(\boldsymbol{y};\theta) = \theta_{\text{const}} + \sum_{u\in\mathcal{V}}(\theta_{u;1}y_u + \theta_{v;0}\bar{y}_v) \\ + \sum_{(u,v)\in\mathcal{E}}(\theta_{st;11}y_uy_v + \theta_{st;01}\bar{y}_uy_v + \theta_{st;10}y_u\bar{y}_v + \theta_{st;00}\bar{y}_u\bar{y}_v) \end{aligned} \tag{3.22}$$

where $\bar{y} = 1 - y$ is the complementary variable of y.
This class of *pseudo-Boolean* energy functions were well investigated by Kolmogorov and Zabih [67], who stated that minimisation can be performed using graph cuts iif $E(\boldsymbol{y})$ is submodular and if all edge weights are strictly positive. Figure 3.5 represents the construction of a binary graph as an accumulation of individual node and edge weights.

3.7 Learning Structured Output Spaces

The main objective of *supervised* learning is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ that maps any form of input $\boldsymbol{x} \in \mathcal{X}$ to a discrete output $\boldsymbol{y} \in \mathcal{Y}$, based on a training set of input-output pairs $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\} \in \mathcal{X}\times\mathcal{Y}$. Although the size of the training dataset is said to be fixed, its probability distribution is considered to be unknown. For image segmentation applications, the function f should take as an input a set of image features (observations $\boldsymbol{x}$) and return the most probable label image $\hat{\boldsymbol{y}} \in \mathcal{Y}$. The main objective of this work is to learn a *discriminant* function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ such that for any given input $\boldsymbol{x}$, the function F will derive a prediction that maximises F over the space of the response variable. Concretely, the hypothesis function f is expressed by:

$$f(\boldsymbol{x};\mathrm{w}) = \arg\max_{y\in\mathcal{Y}} F(\boldsymbol{x},\boldsymbol{y};\mathrm{w}) \tag{3.23}$$

where w corresponds to a parameter vector. Presuming a linear relationship between input-output spaces, combined in a problem dependent function $\Psi(\boldsymbol{x},\boldsymbol{y})$, the function F is also considered to be linear and can take the form:

$$F(\boldsymbol{x},\boldsymbol{y};\mathrm{w}) = \langle \mathrm{w},\, \Psi(\boldsymbol{x},\boldsymbol{y})\rangle \tag{3.24}$$

In machine learning, the quality of a classifier is measured by a *loss* function. There is a variety of loss functions, each of them penalising in a different way the cost that has to be paid for an inaccurate prediction. The simplest loss function is the standard 0-1 loss, introduced by Weston et al. [73] and has been shown to work well for simple prediction problems. For more complicated prediction outputs, more sophisticated loss functions are needed. For example, using a 0-1 loss function for evaluating a natural language parsing (NLP) classifier is senseless, since it lacks of quality measure. Knowing the correct parse tree, the quality of the parsing can be evaluated based on the overlapping of the nodes between the correct and predicted tree (Johnson [74]). If $p(\boldsymbol{x},\boldsymbol{y})$ represents the probability distribution of the data, and $\triangle(\boldsymbol{y}, \hat{\boldsymbol{y}})$ any form of a loss function, the goal of the learning problem is to minimise the risk:

$$R_P^{\triangle}(f) = \int_{\mathcal{X}\times\mathcal{Y}} \triangle(\boldsymbol{y}, f(\boldsymbol{x}))\, dp(\boldsymbol{x}, \boldsymbol{y}) \tag{3.25}$$

Although the probability distribution $p(\boldsymbol{x}, \boldsymbol{y})$ is considered being unknown, given an n set of i.i.d training samples $\mathcal{S} = \{(\boldsymbol{x_1}, \boldsymbol{y_1}), (\boldsymbol{x_2}, \boldsymbol{y_2}), \ldots, (\boldsymbol{x_n}, \boldsymbol{y_n})\} \in \mathcal{X} \times \mathcal{Y}$, the problem boils down in minimising the *empirical* risk:

$$R_{\mathcal{S}}^{\triangle}(f) = \frac{1}{n}\sum_{i=1}^{n} \triangle(\boldsymbol{y_i}, f(\boldsymbol{x_i})) \tag{3.26}$$

For a function $f(\boldsymbol{x})$, the empirical risk is set to zero, iif $f(\boldsymbol{x})$ is parametrised by a weighted vector w, such that $\triangle(\boldsymbol{y}, \boldsymbol{y'}) > 0$ for $\boldsymbol{y} \neq \boldsymbol{y'}$ and $\triangle(\boldsymbol{y}, \boldsymbol{y}) = 0$. Concretely, the condition of zero training zero can be expressed by a set of non-linear constrains as follows:

$$\max_{\boldsymbol{y}\in\mathcal{Y}\setminus\boldsymbol{y_i}} \{\langle \mathrm{w},\ \Psi(\boldsymbol{x_i}, \boldsymbol{y})\rangle\} < \langle \mathrm{w},\ \Psi(\boldsymbol{x_i}, \boldsymbol{y_i})\rangle, \quad \forall i \in \mathcal{Y} \tag{3.27}$$

Converting inequality 3.27 into linear, the following formulation should hold:

$$\forall i,\, \forall \boldsymbol{y} \in \mathcal{Y} \setminus \boldsymbol{y_i} : \quad \langle \mathrm{w},\ \delta\Psi_i(\boldsymbol{y})\rangle > 0 \tag{3.28}$$

where $\delta\Psi_i(\boldsymbol{y}) \equiv \Psi(\boldsymbol{x_i}, \boldsymbol{y_i}) - \Psi(\boldsymbol{x_i}, \boldsymbol{y})$.
Solving inequality 3.28 provides more than one solution for w. A unique solution can be achieved by finding a weighted vector w, subject to $\|\mathrm{w}\| \leq 1$, which can estimate a current prediction score $\hat{\boldsymbol{y}}_{\boldsymbol{i}}(\mathrm{w}) = \arg\max_{\boldsymbol{y}\neq\boldsymbol{y_i}}\langle \mathrm{w},\ \Psi(\boldsymbol{x_i}, \boldsymbol{y})\rangle$ uniformly different from the true score $\boldsymbol{y_i}$. This is considered as the *generalised* version of the max-margin SVM principle introduced by Vapnik [75].
Several versions of the max-margin optimisation problem exist, with the simplest problem of the *hard-margin* formulated as follows:

$$\begin{aligned} \mathrm{SVM}_0 : \qquad &\min_{\mathrm{w}} \quad \frac{1}{2}\|\mathrm{w}\|^2 \\ \forall i,\, \forall \boldsymbol{y} \in \mathcal{Y} \setminus \boldsymbol{y_i} : &\quad \langle \mathrm{w},\ \delta\Psi_i(\boldsymbol{y})\rangle \geq 1 \end{aligned} \tag{3.29}$$

Hard-margin SVM is sensitive to noise in the training data. If a single outlier in the training set exists, this will effect the boundary (hyperplane) of the classifier. Thus, in order to compensate for some error in the training set, a *soft-margin* SVM was introduced, incorporating a slack variable for every non-linear constraint, resulting in a tighter upper bound on the empirical risk 3.26:

$$\begin{aligned} \mathrm{SVM}_1 : \qquad &\min_{\mathrm{w},\,\xi} \quad \frac{1}{2}\|\mathrm{w}\|^2 + \frac{\mathcal{C}}{n}\sum_{i=1}^{n}\xi_i, \quad \text{s.t. } \forall i, \xi_i \geq 0 \\ \forall i,\, \forall \boldsymbol{y} \in \mathcal{Y} \setminus \boldsymbol{y_i} : &\quad \langle \mathrm{w},\ \delta\Psi_i(\boldsymbol{y})\rangle \geq 1 - \xi_i \end{aligned} \tag{3.30}$$

where $\mathcal{C} > 0$ is a constant parameter that controls the tradeoff between training minimisation error and margin maximisation error.

The SVM_1 corresponds to a 0-1 loss, making it inappropriate for very large structured output spaces. Tsochantaridis et al. ([76], [77]), proposed two approaches which generalise SVM_1 to a more generic representation, incorporating arbitrary loss functions in the minimisation process. In the first approach, the slack variables are re-scaled (also known as *slack re-scaling* SVM) according to the loss evaluated in every linear constraint. For a solution $\boldsymbol{y_i}$ violating the margin constraint, penalisation is proportional to the loss $\triangle(\boldsymbol{y_i}, \boldsymbol{y})$. If the loss is severe, penalisation is high and if the loss is low, penalisation is also low. Mathematically, this can be expressed by scaling every slack variable ξ_i with the corresponding loss $\triangle(\boldsymbol{y_i}, \boldsymbol{y})$ of the solution $\boldsymbol{y_i} \in \mathcal{Y}$, formulating the problem as:

$$\begin{aligned} \mathrm{SVM}_1^{\triangle s}: \quad & \min_{\mathbf{w},\xi} \; \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\mathcal{C}}{n}\sum_{i=1}^{n}\xi_i, \quad \text{s.t. } \forall i, \xi_i \geq 0 \\ \forall i, \forall \boldsymbol{y} \in \mathcal{Y} \setminus \boldsymbol{y_i}: \quad & \langle \mathbf{w}, \, \delta\Psi_i(\boldsymbol{y})\rangle \geq 1 - \frac{\xi_i}{\triangle(\boldsymbol{y_i}, \boldsymbol{y})} \end{aligned} \tag{3.31}$$

The second approach involves a *margin re-scaling* (also known as *margin re-scaling* SVM), if the loss function is expressed by a Hamming loss, as proposed by Taskar et al. [78]. In this case, the learning problem is mathematically represented by:

$$\begin{aligned} \mathrm{SVM}_1^{\triangle m}: \quad & \min_{\mathbf{w},\xi} \; \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\mathcal{C}}{n}\sum_{i=1}^{n}\xi_i, \quad \text{s.t. } \forall i, \xi_i \geq 0 \\ \forall i, \forall \boldsymbol{y} \in \mathcal{Y} \setminus \boldsymbol{y_i}: \quad & \langle \mathbf{w}, \, \delta\Psi_i(\boldsymbol{y})\rangle \geq \triangle(\boldsymbol{y_i}, \boldsymbol{y}) - \xi_i \end{aligned} \tag{3.32}$$

All information provided in Sect. 3.7 was derived by the original work of Vapnik et al. [75] and Tsochantaridis et al. ([76], [77]).

3.8 Conclusions

This chapter was intended to make the reader familiar with the basic principles of Conditional Random Fields, their inference and learning process. This is important for understanding the proposed segmentation approach introduced in Chap. 4. Extensive analysis on the aforementioned topics are beyond the scope of this chapter, and the reader is encouraged to refer to other literature sources, such as the surveys of Wang et al. [44] and Sutton et al. [79] or the textbooks of Bishop [80], MacKay [81] and Koller et al. [82].

Chapter 4

Human Recognition in RGBD

4.1 Introduction

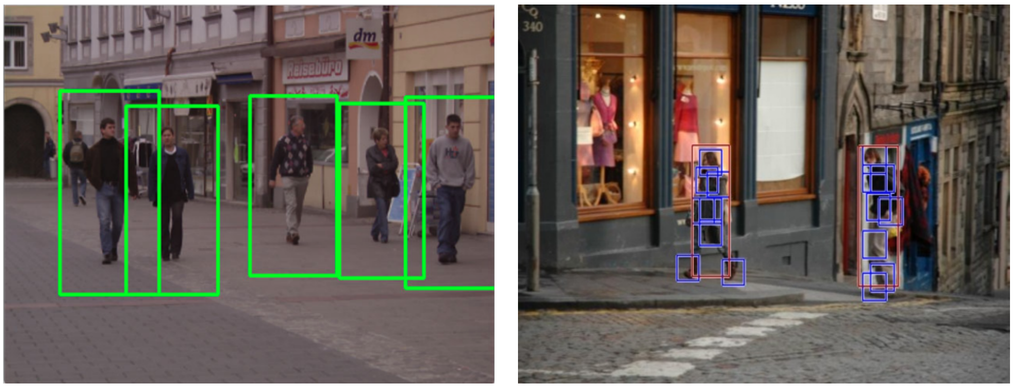
This chapter addresses the problem of localisation and spatial extent determination of a human instance in three-dimensional space. Both fields have been well studied in the two-dimensional domain, showing impressive results not only in accuracy but also in computational performance. With the rapid use of depth sensors, such as the Microsoft Kinect, a new field of research emerged, stimulating researchers in the computer vision and robotics fields to develop algorithms that can perceive the physical properties of a human, bridging the gap between human perception and machine vision.

To this end, a Conditional Random Field (CRF) pairwise submodular energy function is proposed for inferring the segmentation using features from both the RGB and depth domain. The maximum a posteriori (MAP) inference is found in polynomial time using graph cuts. Moreover, the segmentation is performed within the detection box, with the latter evaluated using a single Histogram of Oriented Gradients (HOG) filter and a star-like part-based HOG representation.

The novelty of the proposed method is that no user interaction is required for inferring the segmentation, as opposed to related work in the field.

4.2 People Detection

People detection is an essential component for a wide range of applications such as surveillance systems, people counting and behavioural understanding. However, due to the viewing variations and crowd density in the scene, the reliability of the detector may vary. This effect manifests itself by the coarse localisation of the bounding box, leading to an imprecise detection result. To assess the quality of the detection and its effect on the accuracy of the segmentation, the following approaches were used and evaluated: Dalal and Triggs [8] and Dubout et al. [10]. Both methods are sliding-window approaches and use the Histogram of Oriented Gradients (HOG) feature to learn a human representation. Specifically, Dalal and Triggs use a single feature to represent an object category while Dubout introduces an improved version of the deformable part model detector initially proposed by Felzenszwalb et al. [9]. The main advantage of the deformable part models is that is uses a star-like-structured part-based model defined by a main root filter and a set



(a) (b)

Figure 4.1: Human detection examples using a global HOG representation and a local star-like part-based HOG representation. (a) Example detection results using the approach of Dalal and Triggs [8] and (b) Felzenszwalb et al. [9]. (**Source:** INRIA Person dataset)

of associated parts filters. Dubout et al. [10] extended this work by efficiently deforming the parts across different scales allowing them to compensate for even a wider class of deformations and achieving a more accurate detection output. However, the quality of the detection is unrelated to the number of false positives proposed by the classifier. Therefore, it is important to eliminate those candidates and preserve the ones with higher detection scores.

Let $\mathbf{D} = \{d_1, \ldots, d_n\}$ represent the n amount of detections found in an image and $\mathbf{S} = \{s_1, \ldots, s_n\}$ their corresponding detection scores. In order to ensure that the detector will find all true positives, a low detection threshold t_d is given. This will produce a large amount of false positives but will guarantee all true positive solutions. For eliminating all false positives and preserving only the correct detection outputs, the detection scores were converted into conditional probabilities using the Platt scaling approach (Platt [83]). This method was originally invented in the context of support vector machines (Vapnik [75]) but was later on applied to other classification models as well.

Platt scaling (also known as Platt calibration) is used to relate the detection scores with the conditional probabilities according to the following regression formulation:

$$p(c \mid s_i) = \frac{1}{1 + \exp(A\, s_i + B)} \quad \forall s_i \in \mathbf{S}, c \in \{c_B, c_F\} \tag{4.1}$$

where (A, B) are the parameters of the sigmoid function and can be found by minimising the negative log-likelihood of the training or validation set and $\{c_B, c_F\}$ represents the foreground/background classes. In order to obtain a background probability for every detection rectangle, the following formulation should hold:

$$p(c_B \mid s) = 1 - p(c_F \mid s), \quad \forall s \in \mathbf{S} \tag{4.2}$$

If the background probability of a detection box is smaller than a predefined probability threshold, it should be removed. However, during run time, it may happen that two or more detection boxes have similar probabilities and correspond to the same person. In this case, only the detection box with the highest probability is preserved.

4.3 Energy Function

Following the notation introduced in Chap. 3, let $\mathcal{Y} = \{0, 1\}$ represent a binary label set where 0 corresponds to the background label and 1 to the foreground/object label. For an image of n number of pixels defined over a lattice $\mathcal{V} = \{1, \ldots, n\}$, let $\boldsymbol{y}$ be a labelling which takes values from the $\mathcal{Y}^n$ label space. As stated in Sec. 3.3, the goal of MAP inference is to find the most probable labelling $\boldsymbol{y} \in \mathcal{Y}^n$ conditioned on a set of observations $\boldsymbol{x}$ by minimising an energy function $E(\boldsymbol{y}, \boldsymbol{x})$. The proposed energy function is defined on a set of RGB and depth features and its formulated as follows:

$$E(\boldsymbol{y}, \boldsymbol{x}) = w_{\mathcal{N}} \sum_{i \in \mathcal{V}} \psi_i(\boldsymbol{y}, \boldsymbol{x}) + w_{\mathcal{E}} \sum_{(i,j) \in \mathcal{N}_i} \psi_{ij}(\boldsymbol{y}, \boldsymbol{x}) \tag{4.3}$$

where $\psi_i(\boldsymbol{y}, \boldsymbol{x})$ is a node potential function defined by the product of two conditionally independent events introduced in Sect. 4.3.1, $\psi_{ij}(\boldsymbol{y}, \boldsymbol{x})$ is an edge potential function capturing one of the different pairwise relations discussed in Sect. 4.3.2 and $\mathbf{w} = [w_{\mathcal{N}}, w_{\mathcal{E}}]$ are the corresponding node and edge weights. The proposed energy function adopts a 4-neighbourhood relationship for modelling the edge potentials.

4.3.1 Unary Potentials

Every pixel in the image should be classified as foreground or background label based on a cost defined in the unary term of energy function 4.3. In this framework, the cost is expressed by the product of two conditionally independent probability events, formulated as follows:

$$\psi_i(\boldsymbol{y}, \boldsymbol{x}) = \begin{cases} p_1(x_i)\, p_2(x_i), & \text{if } y_i = 1 \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

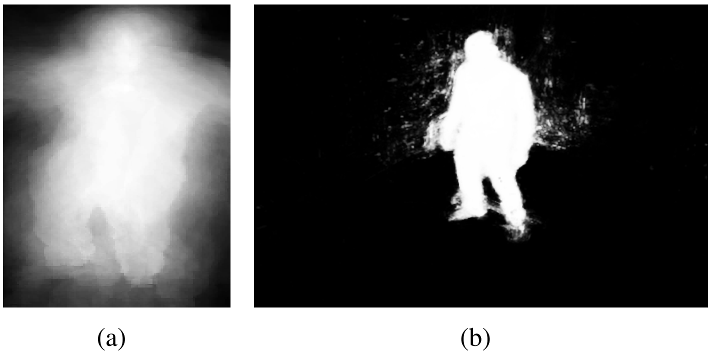
where $p_1(x_i)$ is the probability of pixel x_i to be assign the foreground label according to a learned prior shape probability map (see Algo. 1) and $p_2(x_i)$ refers to the probability of pixel x_i to belong to the foreground, based on the probability outcome of a decision tree classifier (Hänsch [84]), trained on RGB features.

4.3.1.1 Shape Prior

The probability $p_1(x_i)$ of pixel x_i to be assigned to the foreground class is based on a learned prior shape probability map. Every detection rectangle contains regions of pixels that do not correspond to the object of interest such as the corner areas of the rectangle. Using a shape prior, these regions will be assigned a low probability value. An example of a prior probability map is given in Fig. 4.2(a) with the generation process provided by Algo. 1.

4.3.1.2 Decision trees ensemble

As prior probability, $p_1(x_i)$ is completely independent of the measured RGBD data of a specific image. A data-dependent initial estimate is represented by $p_2(x_i)$, which corresponds to the probabilistic output of a pixel-wise classification algorithm known as



(a) (b)

Figure 4.2: Unary potentials: (a) Shape prior map, (b) ProB-RF classifier.

Algorithm 1 Generate human shape prior map

```
Require: A sequence of label images $\boldsymbol{y_m}$ and corresponding RGB images $\mathbf{I_m}$ of a person
    in the scene:
        $\mathscr{S} = \{(\boldsymbol{y_1}, \mathbf{I_1}), (\boldsymbol{y_2}, \mathbf{I_2}), \ldots, (\boldsymbol{y_n}, \mathbf{I_n})\}$
$\mathscr{R} = \emptyset$
for $(\boldsymbol{y_m}, \mathbf{I_m}) \in \mathscr{S}$ do
    Extract detection rectangle from image $\mathbf{I_m}$ using the deformable part model approach from Dubout et al. [10]
    Extract the same rectangle from the corresponding label image $\boldsymbol{y_m}$
    Resize rectangle to a $128 \times 64$ sized image $\boldsymbol{r_m}$
        $\mathscr{R} := \mathscr{R} \cup \{\boldsymbol{r_m}\}$
end for
return The probability map of $\mathscr{R}$
```

the Projection-Based Random Forest (ProB-RF), proposed by Hänsch [84]. The ProB-RF classifier is an ensemble supervised learning technique, which means that is not based only a single classifier but on multiple sub-optimal classifiers. The combined output from all these classifiers is expected to be more accurate compare to the output of a single classifier. The final prediction should assign each pixel a posteriori probability belonging to either the foreground or background based on many simple features extracted implicitly by the decision trees themselves. Specifically, these futures can be categorised into low-level features and high-level features. Low-level features are colour, grey and binary features whereas high-level features model radiometric, shape and semantic information.

The ProB-RF is a two-stage process: the first stage is purely based on low-level features, which provide an a-priori knowledge about the objects in the scene. This information is then used in the second stage for calculating the high-level features and predicting the final categorisation result. Figure 4.2(b) shows the estimated classification map of an exemplary scene. However, this first pixel-wise probability estimate serves as an additional cue to the shape prior and is now used in the global optimisation framework of CRFs. Figure 4.2(b) shows the estimated classification map of an exemplary scene. However, this first pixel-wise probability estimate serves as an additional cue to the shape prior and is now used in

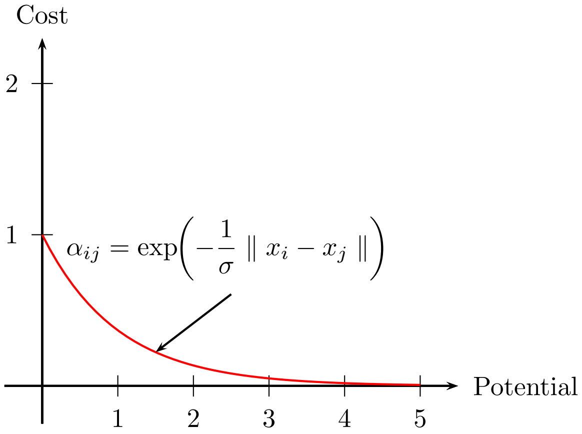


Figure 4.3: Cost assigned to neighbourhood pixels.

the global optimisation framework of CRFs.

4.3.2 Pairwise Potentials

Edge potentials capture the similarity between variables lying within a local neighbourhood. Two random variables sharing the same label should also be assigned a cost greater than zero. Specifically,

$$\psi_{ij}(\boldsymbol{y}, \boldsymbol{x}) = \begin{cases} \alpha_{ij} & \text{if } y_i = y_j \\ 0 & \text{otherwise,} \end{cases} \tag{4.5}$$

Taking advantage of the richness of RGBD data provided by the Kinect sensor, two variables taking the same label should not be separated by an edge, should have similar colours, similar depth and similar normal orientation. All these relationships are modelled by the following edge potentials:

4.3.2.1 Canny Edges

Canny edge extractor is a very known operator for extracting strong edges in an image. Within this framework, Canny edges were used for finding the boundaries between areas and objects, assigning a value of 1 for neighbourhood pixels that do not lie on a Canny edge and 0 otherwise[1].

4.3.2.2 Colour Distance

Two neighbourhood pixels having similar RGB colour should also be assigned the same label. However, in terms of colour quality, RGB space does not separate the *luma* (image intensity) from *chroma* (colour information). For computer vision applications, one may want to separate colour components from intensity for robustness against fast lighting

[1] This is the only edge potential within this work that does not follow the penalisation term proposed in Fig. 4.3

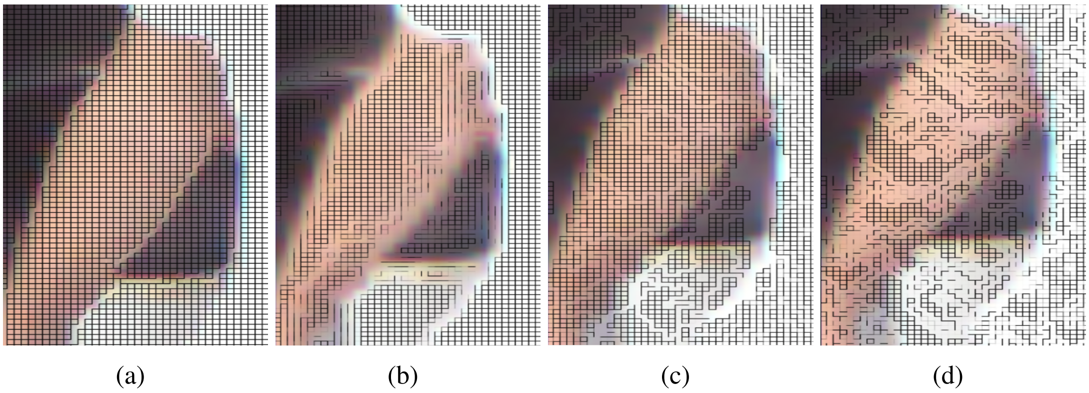


Figure 4.4: Edge potentials. (a) Canny edges, (b) HSV colour distance, (c) 3D Euclidean distance, (d) surface normals. (**Best viewed in colour**)

changes or shadows. After performing this conversion, the Euclidean HSV distance between two neighbourhood pixels is defined as follows:

$$\alpha_{ij} = \exp\left(- \frac{\|\mathbf{c}_i - \mathbf{c}_j\|}{\sigma_c} \right) \tag{4.6}$$

where $\mathbf{c}_i$, $\mathbf{c}_j$ correspond to the HSV values of pixels i and j respectively and σ_c is a bandwidth parameter whose value is set through cross validation.

4.3.2.3 3D Euclidean Distance

Two neighbourhood 3D points that are very close to each other are more likely to share the same label. This relationship is expressed as follows:

$$\alpha_{ij} = \exp\left(- \frac{\mathrm{abs}(\mathbf{p}_i - \mathbf{p}_j)^T \mathbf{n}_j}{\sigma_n} - \frac{\|\mathbf{p}_i - \mathbf{p}_j\|}{\sigma_d} \right) \tag{4.7}$$

where $\mathbf{p}_i$, $\mathbf{p}_j$ correspond to the 3D position of the points i and j respectively, $\mathbf{n}_k$ is the surface normal at point $\mathbf{p}_j$ and σ_n, σ_d are bandwidth parameters whose values are defined by cross validation.

4.3.2.4 Angles Normal

Two 3D points lying on the same part of the object should have similar normal orientation. This relationship is expressed as follows:

$$\alpha_{ij} = \exp\left(- \frac{\theta_{ij}}{\sigma_\theta} \right) \tag{4.8}$$

where θ_{ij} is the angle between two neighbourhood normals $\mathbf{n}_i$ and $\mathbf{n}_j$ defined as:

$$\theta_{ij} = \arccos\left(\frac{< \mathbf{n}_i, \mathbf{n}_j >}{\|\mathbf{n}_i\| \|\mathbf{n}_j\|} \right) \tag{4.9}$$

and σ_θ is a bandwidth parameter whose value is specified by cross validation.

Algorithm 2 Generate RGBD features

```
Require: $\mathscr{S} = \{(\boldsymbol{y_1}, \mathrm{d_1}), (\boldsymbol{y_2}, \mathrm{d_2}), \ldots, (\boldsymbol{y_n}, \mathrm{d_n})\}$
$\mathscr{D} = \emptyset$
for $(\boldsymbol{y_m}, \mathrm{d_m}) \in \mathscr{S}$ do
    Compute all features $\boldsymbol{x_m}$
        $$\mathscr{D} := \mathscr{D} \cup \{(\boldsymbol{y_m}, \boldsymbol{x_m})\}$$
end for
return $\mathscr{D}$
```

4.4 Learning

As discussed in Sect. 3.7, the learning process involves finding a set of weights $\mathbf{w}$ that can maximise the margin between a current segmentation result $\boldsymbol{y'}$ and its corresponding label image $\boldsymbol{y}$, assuring that $\triangle(\boldsymbol{y}, \boldsymbol{y'}) > 0$ for $\boldsymbol{y} \neq \boldsymbol{y'}$. If $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_m}\}$ represents a set of RGB and depth features and $\{\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_m}\}$ are the corresponding ground truth images, the goal is to learn a set of parameters $\mathbf{w}$ that maximise the likelihood:

$$\max_{\mathbf{w}} \prod_{m} p(\boldsymbol{y_m} \mid \boldsymbol{x_m}) \tag{4.10}$$

The training set was generated according to Algo. 2.

Influenced by the work of Szummer et al. [85], graph cuts were used to learn the parameters of the proposed energy function 4.3. Satisfying the submodularity constrains (see Appx. A), graph cuts could ensure an *efficient* maximum margin learning of the parameters with an *exact* solution, preserving generalisation for new images via a large margin regulariser. The one-slack margin rescaling SSVM (see Sect. 3.7) was employed for efficiently solving the minimisation problem and finding the set of weights $\mathbf{w}$ that best represent the given training set. The learning process is presented by Algo. 3. Here, $\mathcal{C}$ and ε are constant values, $\mathbf{w} = [w_{\mathcal{N}}, w_{\mathcal{E}}]$ are the weights that have to be optimised for a given training set $\mathscr{D}$, ξ is a slack variable and Δ corresponds to the Hamming loss. Depending on the expected accuracy, different loss functions could be used. Furthermore, it must be stressed that the constant parameter $\mathcal{C}$, also known as the slack penalisation parameter, shouldn't be set to a very high or low value because this can significantly effect the size of the final margin of the classifier.

Within the learning process, the goal is to enforce that the ground truth energy will have the lowest energy value from all other labelings. If this constraint is not satisfied, or if the margin is not achieved, this label solution will be added in the constraint set $\mathscr{W}$. This process continues until the values of the weights have converged. According to Joachims et al. [86], the objective function is quadratic to $\mathbf{w}$ and linear to the constraints, also known as the *quadratic programming problem* (Cottle et al. [87]). The advantage of this objective function is that a global minimum can be reached in polynomial time. The minimisation procedure was achieved by implementing the Nesterov non-linear quadratic algorithm, which is part of a family of algorithms known as *interior point solvers* (Boyd and Vandenberghe [88]) and are commonly used for minimising objective functions of the form 4.11.

Algorithm 3 The one-slack margin rescaling Structured Support Vector Machine

```
Require: A set of training examples, constant values $C$, $\varepsilon$
$\mathscr{W} \leftarrow \emptyset$
repeat
    Update the parameters $\mathbf{w} = [w_{\mathcal{N}}, w_{\mathcal{E}}]$ to maximise the margin
```

$$\begin{aligned} \min_{\mathbf{w},\,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \mathcal{C}\,\xi \\ \text{s.t} \quad & \mathbf{w} \geq 0, \;\; \xi \geq 0 \\ & \frac{1}{M}\sum_{m=1}^{M} E(\hat{\boldsymbol{y}}_m, \boldsymbol{x}_m) \; - \; E(\boldsymbol{y}_m, \boldsymbol{x}_m) \geq \frac{1}{M}\sum_{m=1}^{M} \Delta(\boldsymbol{y}_m, \hat{\boldsymbol{y}}_m) \; - \; \xi \\ & \forall \;\; (\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_M) \in \mathscr{W} \end{aligned} \tag{4.11}$$

```
    for $(\boldsymbol{y_m}, \boldsymbol{x_m}) \in \mathscr{D}$ do
        $\hat{\boldsymbol{y}}_{\boldsymbol{m}} \leftarrow \arg\min_{\boldsymbol{y}} E(\boldsymbol{y_m}, \boldsymbol{x_m})$
    end for
    $\mathscr{W} \leftarrow \mathscr{W} \cup \{\hat{\boldsymbol{y}}_{\boldsymbol{1}}, \ldots, \hat{\boldsymbol{y}}_{\boldsymbol{M}}\}$
until $\frac{1}{M}\sum_{m=1}^{M} \Delta(\boldsymbol{y_m}, \hat{\boldsymbol{y}}_{\boldsymbol{m}}) \; - \; E(\hat{\boldsymbol{y}}_{\boldsymbol{m}}, \boldsymbol{x_m}) \; + \; E(\boldsymbol{y_m}, \boldsymbol{x_m}) \; \leq \; \xi \; + \; \varepsilon$
```

4.5 Quantitative analysis

The proposed algorithm was tested and evaluated on people with different poses and costume changes observed in a simulated indoor environment of a train wagon (see Appx. B). Working exclusively with humans, the proposed work could be potentially generalised for recognising a variety of active objects in the scene.

Edge potentials defined on depth measurements require high precision between points lying in the local neighbourhood. Although the objective was to develop a 3D object recognition system using raw RGBD data, the Kinect sensor was calibrated for enhancing the quality of the observed data (see Appx. C).

A total of 25 sequences were generated, every sequence containing 200 frames. From all 5000 images, 3200 images over 16 sequences were used for training and the rest for testing. The training set was used for learning the weights of the Platt calibration, structured SVM and shape prior. For the ProB-RF, a 4-fold cross validation approach was used, taking into account all 25 sequences. This means that the algorithm was trained on 3 randomly selected sets of sequences and tested on the remaining sets of sequences. The training and testing times for each fold are provided in Tab. 4.1. Some classification results are shown in Fig. 4.5.

To the best of my knowledge, no publicly available RGBD dataset exists that could provide label images with ground truth detection boxes. Generating ground truth label images is a very time consuming process as it requires a lot of manual work by the user. For eliminating the effort, reference images were generated using the approach of Shotton et al. [30], a well known human pose estimation algorithm that was also commercialised for Kinect games.

The training set of the publicly available INRIA Person dataset was used for learning the weights for the classifiers for both Dalal and Triggs [8] and Dubout [10] approaches,

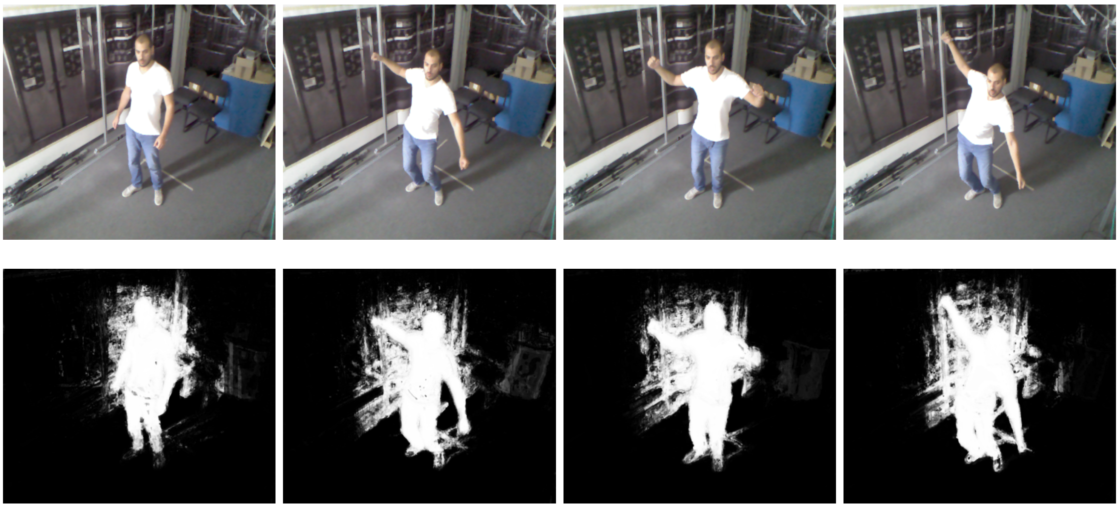


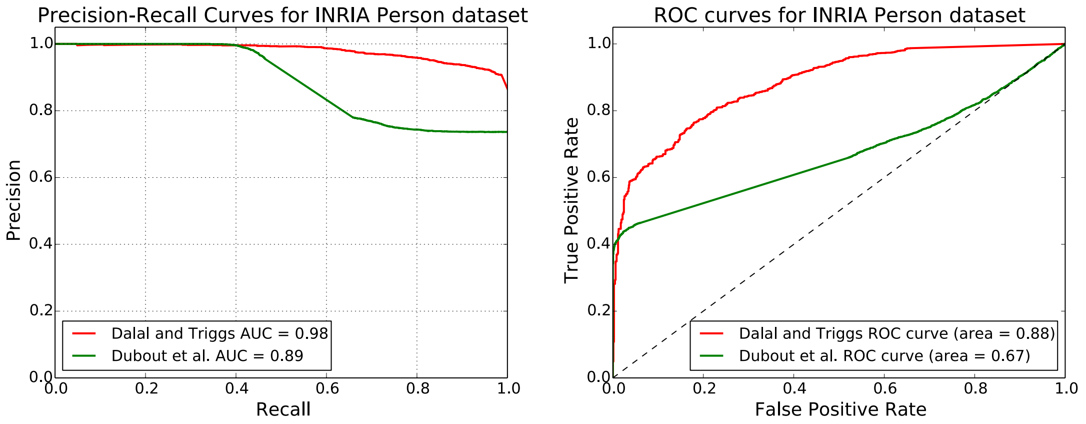
Figure 4.5: Probability distribution derived from the decision tree classifier. *Top row:* raw images from a sequence. *Bottom row:* a per-pixel a posteriori probability distribution. High intensity of a pixel indicates high probability of that pixel to belong to a person. (**Best viewed in colour**)

	Folds	**Average time per fold**	**Standard deviation per fold**	**Average time per sequence**
Training time	4-15	1462.20	35.13	365.55
	0-3,8-15	1482.31	28.09	370.58
	0-7,12-15	1487.15	30.10	371.79
	0-11	1496.49	31.30	374.12
Testing time	0-3	1092.39	53.17	273.10
	4-9	1101.13	43.73	275.28
	8-11	1102.03	46.43	275.51
	12-15	1080.47	45.66	270.12

Table 4.1: Training and testing times per fold for the decision tree classifier (in milliseconds).

following a bootstrapping process. Subsequently, the resulting classifiers were tested on the validation set for learning the parameters of the Platt calibration curves using a Newton non-linear optimiser. For a detection rectangle to be assigned to the background class, a probability threshold of 0.6 was given.

The average computational times recorded for a complete scene are presented in Fig. 4.7. It is apparent from the pie charts that the node and edge potentials require minimum computational effort while the object detectors are computationally more expensive. For a VGA image resolution the proposed implementation runs on ≈ 1.5 FPS using the global HOG human representation and ≈ 2.4 FPS for the improved DPM approach. Graph cuts require the least effort (0.7 ms) as they can be solved in polynomial time. All experiments were conducted on a DELL M4800 Workstation with an i7-4800MQ CPU at 2.70GHz processor and 16GB RAM. The complete pipeline was designed in a multithreaded fashion, parallelising all computations (Boost library [89]).



(a) (b)

Figure 4.6: Precision-Recall and ROC curves for the INRIA Person dataset. The precision-recall and ROC curves are a result of Dalal and Triggs algorithm [8] and Dubout et al. [10]. (**Best viewed in colour**)

4.6 Qualitative analysis

The qualitative analysis was divided into two main parts: evaluating the performance of the detectors and subsequently the quality of the segmentation as a consequence of the quality of the detection boxes. The precision-recall and ROC curves for both detectors were calculated using the training set provided by the INRIA people dataset. Results in Fig. 4.6 showed that the performance of Dalal and Triggs is much higher than Dubout et al. Likewise, the ROC curves showed that the accuracy of Dalal and Triggs object detection algorithm is much higher compare to Dubout's algorithm. Specifically, for a high detection threshold, the Dalal and Triggs algorithm provides a precision and recall $\approx 100\%$ and when the detection threshold begins to drop (relaxing the parameter), false negatives begin to appear, which makes precision go down. However, the point where the precision drops for Dubout's approach is in a much earlier point in time compare to Dalal. This means that the former is more sensitive and can provide false positive results even for a higher detection threshold. Similar interpretation could be given for the ROC curves respectively.

Furthermore, it should be stressed that the performance of the detectors is independent from the quality of the detection boxes. Thus, an additional metric was required for checking the overlapping accuracy against the corresponding ground truth detection box. Everingham et al. [90] evaluated the accuracy of the object detectors by measuring the area of the overlapping bounding box derived by the predicted bounding box (B_p) and the corresponding ground truth box (B_{GT}) using the following relation:

$$\text{overlapping} = \frac{\text{area}(B_p \cap B_{GT})}{\text{area}(B_p \cup B_{GT})} \tag{4.12}$$

To this end, for a predicted detection box to be considered as true positive, the area a of the overlapping region threshold was set to be greater than 50%. The accuracy of both detectors was checked on 1800 test images capturing a person undergoing different poses in the scene. Ground truth detection boxes were extracted from the label images of the

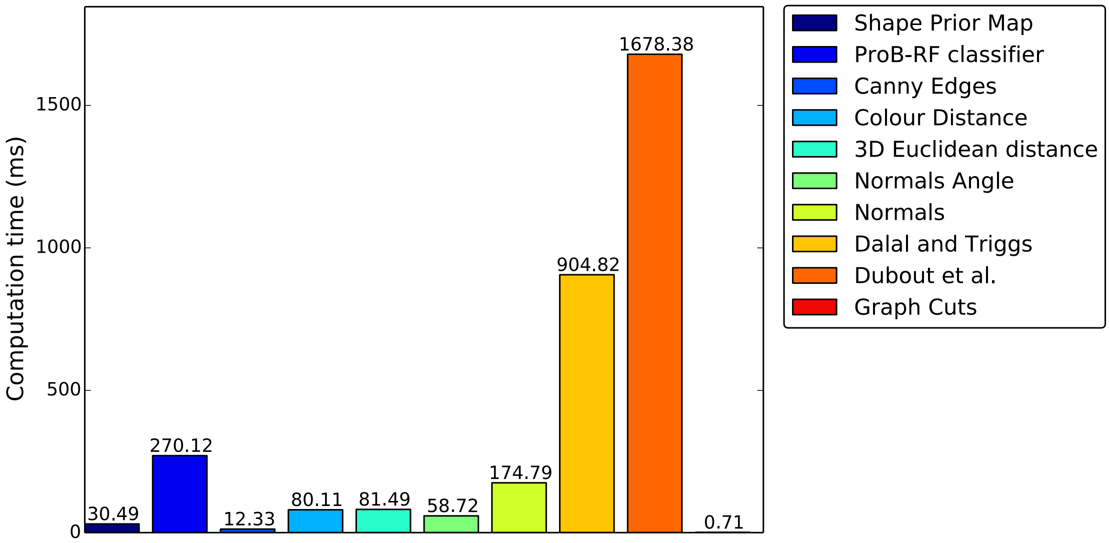


Figure 4.7: Average recognition time per frame. The decision tree classifier is computationally more expensive compare to all other potentials. The improved version of the deformable model introduced by Dubout et al. [10] requires more time compare to the fixed-template-style HOG-based detector of Dalal and Triggs [8]. For a VGA image resolution, the graph cut algorithm does not preclude a real-time operation for the segmentation. (**Best viewed in colour**)

corresponding test images by finding the minimum area bounding rectangle of the largest connected component (Suzuki et al. [91]). The overlapping accuracy achieved was 64.2% (Dalal) and 72.3% (Dubout) respectively. As expected, the DPM approach provides a better localisation compare to Dalal approach as it uses part-based star-like configuration of HOG features to infer the position of the bounding box. Bounding boxes with low foreground probability were removed during testing.

The segmentation accuracy was checked within the area of the ground truth detection boxes and the ones computed from the detectors. However, in cases of extreme poses, the detection box would partially capture the person in the scene. For instance, when a standing person stretches his hands horizontally, the detection box would fail to include the complete part of the arms. On the other hand, the ground truth detection box is generated using the complete body and therefore comparison in this case is not reliable. This problem was solved by extracting part of the label image that corresponded to the area of the given detection box.

The segmentation approach was assessed using three different metrics:

- **Hamming Loss:** This metric counts the number of mis-labelled pixels in the predicted image $\mathbf{y}'$ with respect to the corresponding ground truth image $\mathbf{y}$, expressed by:

$$\triangle_{HL}(\boldsymbol{y},\boldsymbol{y}') = \sum_{i=1}^{|\boldsymbol{y}|} y_i^{'} \oplus y_i \tag{4.13}$$

- **Normalised Hamming Loss:** It was first introduced by Teichman et al. [22] and it's considered a hard penalisation metric compare to other loss functions, as it gives a zero loss if the number of incorrectly labelled pixels is equal or exceeds the number of pixels

Metric	**Edge Potentials**			
	Canny Edges	**Colour Distance**	**3D Euclidean Distance**	**Surface Normals**
		Dalal and Triggs		
$\triangle_{HL}$	8032.102 ± 2477.052	8052.325 ± 2433.194	7988.132 ± 2401.440	8121.021 ± 2022.032
$\triangle_{N-HL}$	0.583 ± 0.132	0.511 ± 0.128	0.592 ± 0.105	0.554 ± 0.124
IOU	0.665 ± 0.103	0.523 ± 0.044	0.702 ± 0.022	0.698 ± 0.058
		Dubout et al.		
$\triangle_{HL}$	8012.790 ± 2578.160	7907.000 ± 2308.790	7911.910 ± 2205.740	7936.520 ± 2178.710
$\triangle_{N-HL}$	0.646 ± 0.113	0.651 ± 0.104	0.651 ± 0.102	0.650 ± 0.098
IOU	0.705 ± 0.093	0.712 ± 0.076	0.710 ± 0.075	0.710 ± 0.074
		Ground Truth Detection Box		
$\triangle_{HL}$	5508.970 ± 1626.130	5566.37 ± 1667.510	5464.600 ± 1590.560	5504.360 ± 1896.560
$\triangle_{N-HL}$	0.758 ± 0.063	0.755 ± 0.069	0.760 ± 0.064	0.758 ± 0.079
IOU	0.799 ± 0.050	0.797 ± 0.056	0.801 ± 0.052	0.798 ± 0.071

Table 4.2: Metric analysis on different edge potentials. Every row represents a different metric evaluator; Every column corresponds to a different edge potential; Top table presents segmentation results produced by the ground truth detection box; Bottom table presents segmentation results from [10].

corresponding to foreground in the label image. Specifically, this metric is formulated as follows:

$$\triangle_{N-HL}(\boldsymbol{y},\boldsymbol{y'}) = 1 - \min\left(1, \sum_{i=1}^{|\boldsymbol{y}|} \frac{y_i^{'} \oplus y_i}{\sum_{j=1}^{|\boldsymbol{y}|} \mathbf{1}[y_j = 1]}\right) \tag{4.14}$$

- **Intersection over union:** This is an segmentation metric introduced by Everingham et al. [90] and it is formulated by:

$$\text{seg. accuracy} = \frac{\text{true pos.}}{\text{true pos. + false pos. + false neg.}} \tag{4.15}$$

where true positives represent the number of correctly classified pixels, false positives the number of wrongly classified pixels and false negatives the number of pixels that were wrongly not classified as true positive.

Qualitative results are provided in Tab. 4.2 for all metrics, evaluated on randomly selected images from the generated test sequences. It is evident that all metrics computed by the ground truth bounding box show an overall improvement in the segmentation accuracy, outperforming the results produced by the detection boxes of Dalal and Dubout. Furthermore, comparing the metrics computed by the different edge potentials, it is easy to perceive the insignificance between the values. This can be explained as follows: the proposed method does not require any prior information from the user for enforcing the min-cut towards a human shape (Teichman et al. [22]) but performs the min-cut using

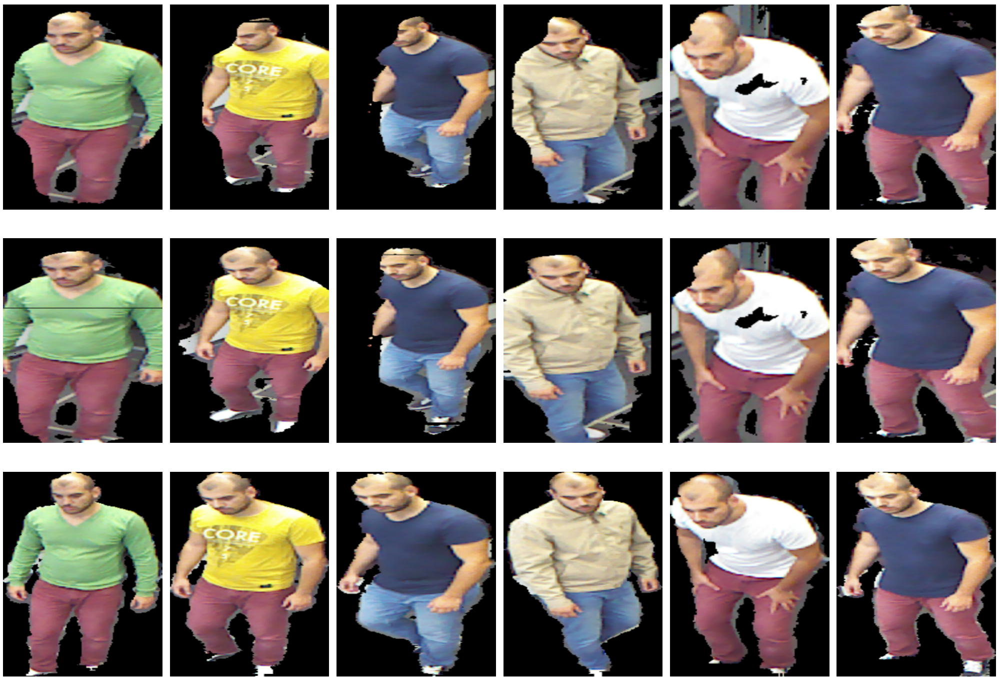


Figure 4.8: Qualitative image segmentation results of different human poses. These figures outline the improvement in quality of the proposed segmentation approach using the ground truth box rather than the detection box provided by Dubout et al. [10]. From top to bottom: segmentation results using Dalal and Triggs [8]; segmentation results using Dubout et al. [10]; segmentation results using the bounding box extracted by the labelled images; labelled images. (**Best viewed in colour**)

edge potential values that correspond to node potentials larger than a predefined probability threshold (85 %). Thus, only edge potential values that lie at the borders of the object should effect the cut.

Furthermore, the proposed segmentation approach was compared against the approach of Zheng et al. [92]. Their method is based on a new form of convolutional neural network that combines Convolutional Neural Networks and Conditional Random Fields probabilistic models. Specifically, they proposed a Conditional Random Field energy function that is based on a Gaussian pairwise potential function and a mean-field approximate inference as Recurrent Neural Network (RNN). This network, named as CRF-RNN, is then given to a CNN to obtain a deep network that has properties of both CNNs and CRFs.

To ensure a fair comparison, both approaches were tested on the test image set acquired in the test field by omitting the detection boxes and replacing them with the minimum bounding boxes generated from the label images, ensuring a complete body encapsulation. The parameters w of the proposed energy function were re-trained using as edge potentials the product of all edge potentials presented in Sect. 4.3.2. The proposed segmentation approach had an overall segmentation improvement of 58.2 % over the complete test set. Visual results are given in Fig. 4.10. One can see that the proposed segmentation approach provides better segmentation results for extreme poses compare to the CRF-CNN (rows one, three and four respectively). However, it should be stressed that the CRF-CNN approach performed better for some upright poses (see rows two, five and six respectively).

One of the main reasons may be that the authors used a pool of ≈ 75.000 training images, which means that a much larger variation in clothing, poses and number of people is provided. It is believed that using a similar RGBD-based training set could provide an even larger improvement over the CRF-CNN approach. The bandwidth parameter values related to the edge potentials were set to: $\sigma_c = 0.3$, $\sigma_n = 0.5$, $\sigma_d = 0.5$, $\sigma_\theta = 0.2$.

4.7 Conclusions

In this chapter an approach was introduced for detecting and segmenting human instances in RGBD space based on Kinect-like RGBD data. The detection performance was evaluated on a single HOG-based feature representation and a part-based HOG-feature representation. Results showed that part-based representations provide more accurate, but also more computationally expensive detection results compare to the former, allowing higher degree of spatial variance between body parts and thus, resulting into a more robust detection box.

In order to determine the spatial extent of the person within the detection box, the proposed method makes use of a rich set of RGB and depth features modelled within a Conditional Random Field pairwise energy function. For unary potentials with a probability larger than a predefined threshold, using any of the edge potentials produced good segmentation results. This means that the unary potentials play an integral role for the segmentation task. Comparing the proposed method to the CRF-CNN approach (Zheng et al. [92]) showed improved results mostly for extreme human poses. However, is should be stressed that in some normal human poses the CRF-CNN produced better results, which is assumed to be because of the larger number of available training data.

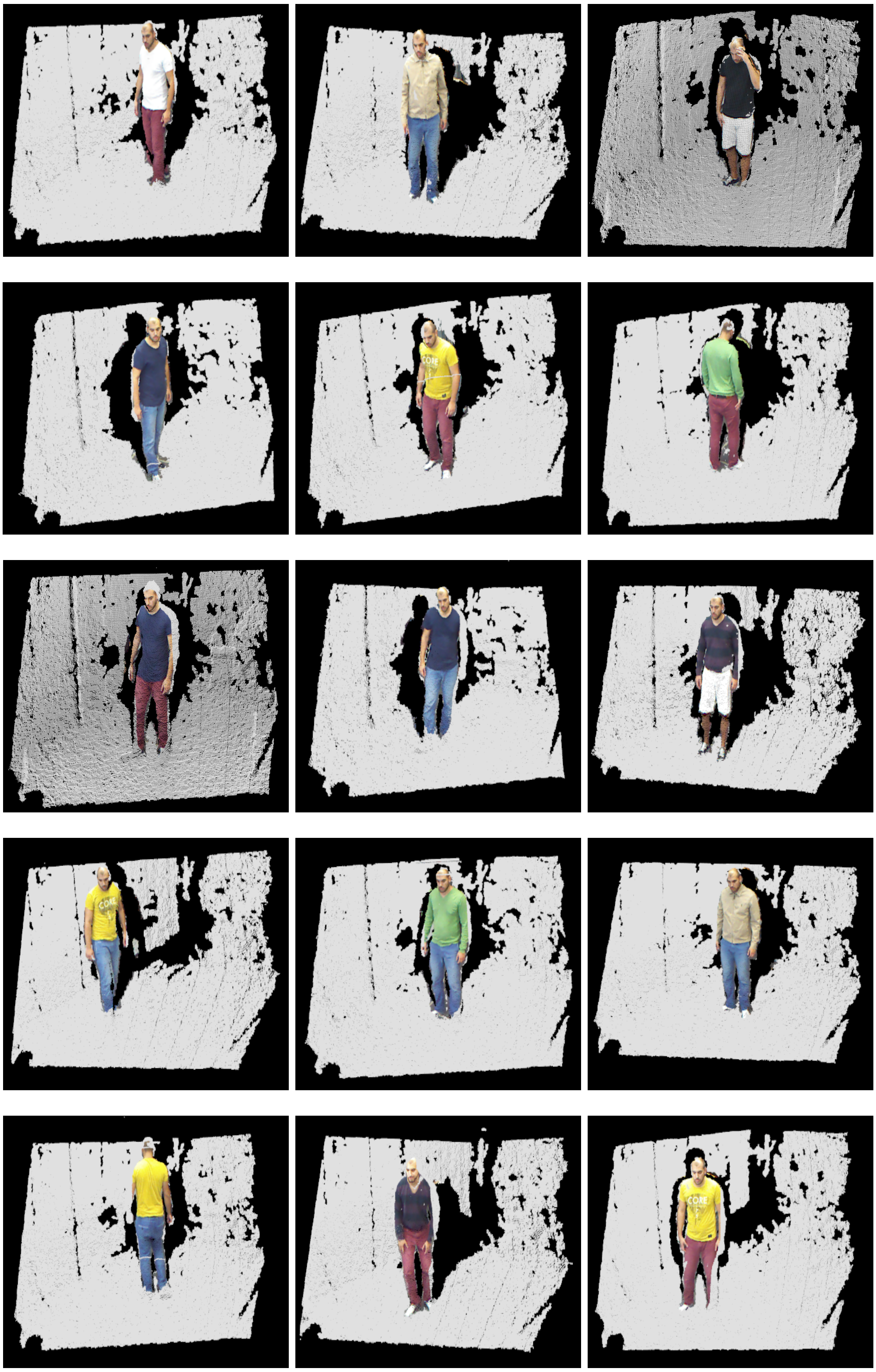


Figure 4.9: Results of human instance segmentations in RGBD space. (**Best viewed in colour**)

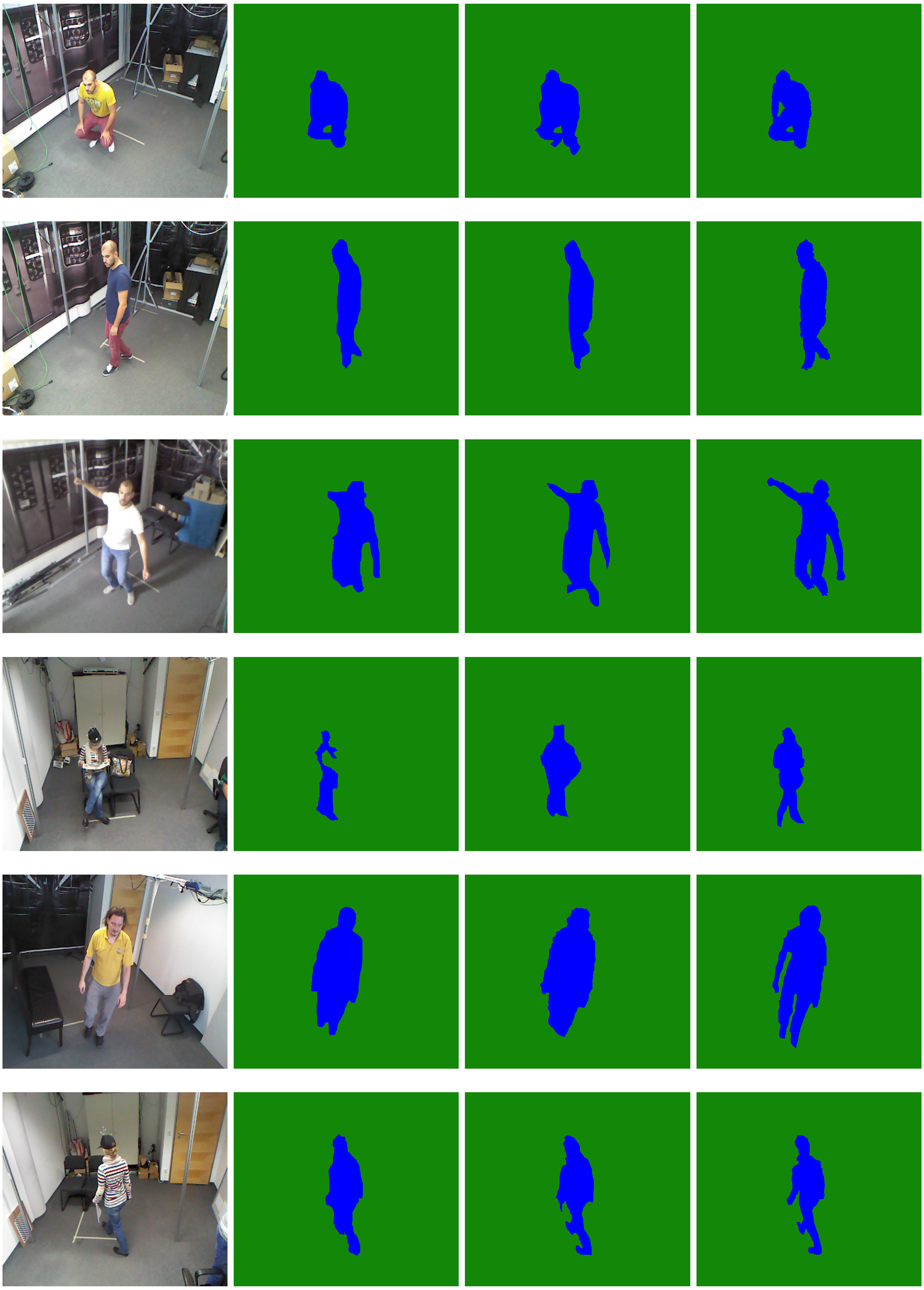


Figure 4.10: Qualitative comparison results. *First column:* raw RGB images. *Second column:* CRF-RNN results. *Third column:* Proposed approach. *Forth column:* label images. (**Best viewed in colour**)

Chapter 5

Human Motion Estimation and Tracking in RGBD

5.1 Introduction

Object recognition is known as the process for detecting instances of semantic objects from camera data. However, non-rigid objects such as humans do not have a fixed representation but undergo significant shape deformations in time. This means that the task of object recognition could also be extended to the task of understanding object motion. With the use of low-cost commodity sensors, such as the Microsoft Kinect, the human motion could be modelled with more human-like visual perception data using the real time RGB and depth information provided by the sensor.

This chapter introduces a method for capturing and tracking people's shape deformations in time in a dynamic indoor environment from Kinect-like RGBD data. The proposed methodology consists of two main components: (1) a workflow that enhances the accuracy of an octree-based foreground estimation algorithm proposed by Kammerl et al. [12] and (2) the use of the Minimum Volume Enclosing Ellipsoid algorithm for capturing the spatio-temporal changes of the person in a 3D scene.

The motivation behind the work was to understand normal and abnormal behaviours of people in an indoor public environment – such as a train wagon – from a network of multiple Kinect sensors.

5.2 Extract the geometry of human motion

The current section introduces an approach for monitoring and tracking human poses in 3D space from Kinect-like RGBD data. An outline of the proposed workflow is provided by Algo. 4 and consists of two main components: (1) a filtering process for improving an existing octree-based 3D foreground estimation approach and (2) a method for capturing and retrieving the geometry of a foreground instance. Specifically, steps 1–8 describe a process for improving the algorithm proposed by Kammerl et al. [12] and extracting accurate foreground masks and steps 9–12 use the Minimum Volume Enclosing Ellipsoid (MVEE) algorithm proposed by Moshtagh [13] for capturing and deriving meaningful information about its shape and the way it deforms in time. Each step of the algorithm is

Algorithm 4 Extract the geometry of a human shape

```
Require: A pair of point clouds representing the background static scene and the current scene
Trim both point clouds in the depth direction applying a pass-through filter
Extract foreground objects from the scene using the approach of Kammerl et al. [12]
if foreground exists then
    for all foreground do
        Map the foreground in an image plane using a perspective projection
        Check if a valid contour exists using the contour extraction algorithm of Suzuki et al. [91]
        if contour is valid then
            if contour size is larger than a predefined threshold then
                Compute the convex hull for this dataset
                Find the ellipsoid enclosing the human figure using the MVEE algorithm of Moshtagh [13]
                Decompose its variance-covariance matrix using PCA
                Smooth the data using Kalman Filter [11]
            end if
        end if
    end for
end if
return Ellipsoid information for every human instance in the scene
```

extensively analysed in the following sections.

5.2.1 Point Cloud Depth-Based Trimming

One major drawback of the Kinect sensor is the discretisation error in the depth measurements, which increases quadratically with respect to the distance from the sensor. According to the Kinect manufacturer, the ideal working distance should be in the range of 0.5 m – 4.5 m. Therefore, voxels lying outside this range should be removed. This was achieved by trimming the point cloud in the $\mathbf{Z}$ direction using a pass-through filter.

5.2.2 Detecting Spatial Changes using an Octree

The foreground extraction algorithm proposed in Algo. 4 is based on the approach introduced by Kammerl et al. [12] for the problem of real time point cloud compression and streaming. The method works by recursively encoding the structural differences between the octree representations of two point clouds based on a logical bitwise XOR (exclusive OR[1]) operator. These structural differences correspond to the spatial changes between the clouds (see Fig. 5.1).

An octree (Meagher [93]) is a tree based data structure in which every internal/leaf node has exactly eight children. Each node in the octree subdivides the space it represents into eight octans. For the case of object extraction, the spatial changes encode the movement

[1] Exclusive or is a logical operation that outputs true only when both inputs differ.

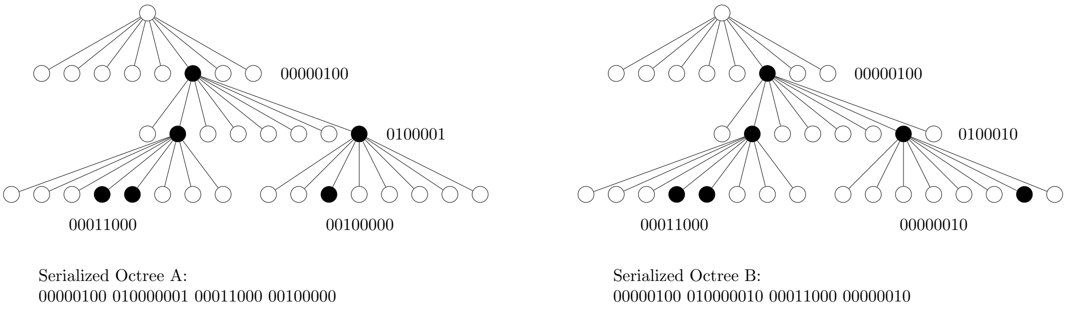


Figure 5.1: Comparing the octree data structures of two point clouds. (**Source:** The above figure has been reproduced from the work of Kammerl et al. [12])

of the object in the 3D scene. Spatial changes in the leaf nodes, such as sparsity of points and number of neighbours, can give an indication of these spatial changes. Depending on the predefined size of the leaf node, detection sensitivity rate and processing time may vary. Large leaf nodes are faster to process but provide less information of the moving foreground and very small leaf sizes are able to capture detailed spatial changes but with a substantial cost in the computation performance.

5.2.3 Perspective Plane Projection

The *pinhole camera model* (Hartley and Zisserman [94]) describes the mathematical relationship between a 3D point and its projection onto a 2D plane (usually image plane). This 2D↔3D mapping is also known as a *perspective projection* and can be written as:

$$\begin{aligned} \mathrm{x} &= f\,\frac{\mathrm{X}}{\mathrm{Z}} + x_0 \\ \mathrm{y} &= f\,\frac{\mathrm{Y}}{\mathrm{Z}} + y_0 \end{aligned} \tag{5.1}$$

where (x, y) are the projected image coordinates of the 3D point $(\mathrm{X}, \mathrm{Y}, \mathrm{Z})$, f represents the focal length of the camera expressed in pixel units and (x_0, y_0) corresponds to the principal point of the sensor. From the above formulations, it is clear that the calibration accuracy (see Appx. C) plays an important role for the quality of the mapping.

5.2.4 Convex Hull

In the field of computational geometry, convex hull of a shape is the smallest convex polygon containing all points of that object. In its 2D representation, a convex hull is defined by a number of "facets" composed of lines and edges where in 3D its always defined by a set of planar triangles.

An important property of the convex hull is that it does not introduce new values but uses existing ones from the original dataset. This means that it is expressed by a subset of points referenced in the original set. For a 3D human figure, the corresponding convex hull is defined by a set of 3D points located on its body silhouette. These points are then

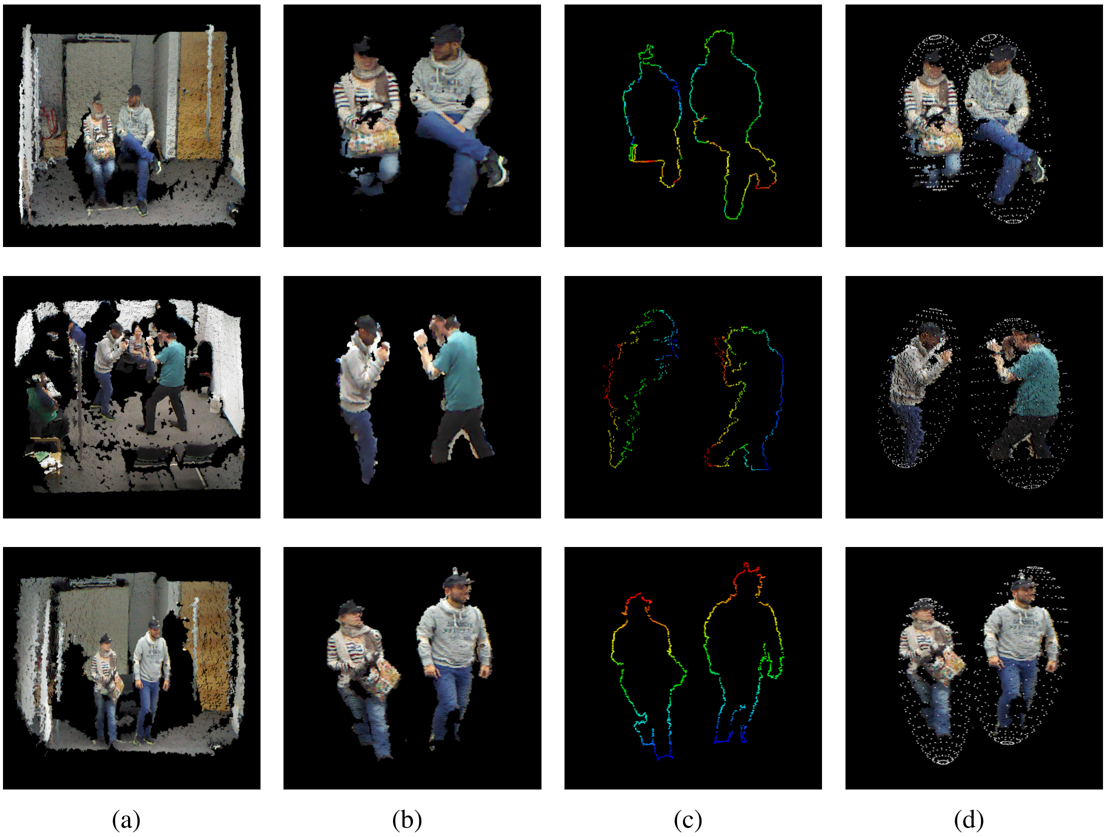


Figure 5.2: Foreground masks for different human poses in an indoor environment. From left to right: (a) raw point clouds capturing different human poses in a scene; (b) extracted foreground masks using the proposed filtering approach in Algo. 4; (c) convex hull of the human figures; (d) result of the Minimum Volume Enclosing Ellipsoid introduced by Moshtagh [13]. (**Best viewed in colour**)

given as an input to the Minimum Volume Enclosing Ellipsoid algorithm (Moshtagh [13]) for deriving the best fitted encapsulated ellipsoid. Using the complete set of foreground points would lead to a dramatic increase of the execution time of the proposed workflow.

5.2.5 Ellipsoid for human motion analysis

The shape of an ellipsoid is a well suited mathematical representation of the human figure. Compare to a sphere, an ellipsoid has higher degrees of freedom, which helps to capture a larger set of human poses. The Minimum Volume Enclosing Ellipsoid algorithm (Moshtagh [13]) was used to fit an ellipsoid to a human pose. The solver is based on the Khachiyan algorithm [95], which is able to solve non-linear convex functions in polynomial time.

Let $\mathbf{X} = \{\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_n}\} \in \mathbb{R}^3$ correspond to a set of 3D points representing the human figure. According to Sect. 5.2.4, the minimum bounding ellipsoid could be computed using only the convex points proposed from the given set. Therefore, let $\mathbf{X_{convhull}} = \{\mathbf{X_1}, \ldots, \mathbf{X_m}\} \in \mathbb{R}^3$ be a subset of this dataset ($\mathbf{X_{convhull}} \subset \mathbf{X}$). According to Moshtagh [13], the problem of determining the ellipsoid of least volume given the

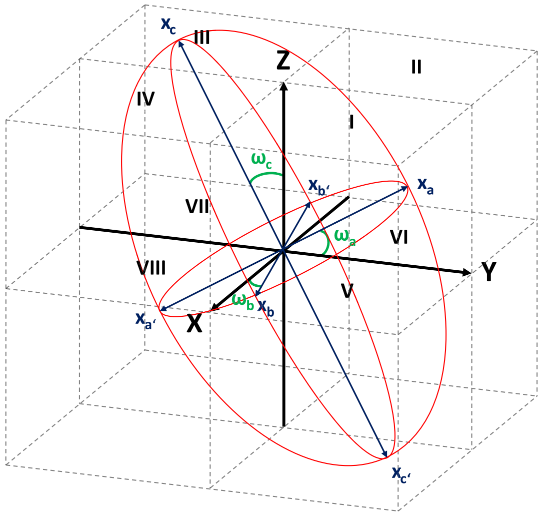


Figure 5.3: Mathematical representation of an ellipsoid.

convex set $\mathbf{X_{convhull}}$, is equivalent in finding a vector $\mathbf{X_c} \in \mathbb{R}^3$ representing the center of the ellipsoid and a 3×3 positive definite symmetric matrix $\mathbf{C}$ (that is a variance-covariance matrix) that satisfies the following condition:

$$\begin{aligned} \underset{\mathbf{C},\,\mathbf{X_c}}{\text{minimize}} \quad & \det(\mathbf{C^{-1}}) \\ \text{subject to} \quad & (\mathbf{X_i} - \mathbf{X_c})^T \mathbf{C} (\mathbf{X_i} - \mathbf{X_c}) \leq 1, \quad i = 1, \dots, n \\ & \mathbf{C} > 0. \end{aligned} \tag{5.2}$$

It should be pointed out, however, that $\mathbf{X_c}$ is not placed precisely at the center of the human figure but rather deviates depending on its shape variation. For instance, in situations where a standing person raises his hands, the center of the ellipsoid is not longer positioned at the center of his stomach but at a higher point. Hence, depending on the pose variation, the center of the ellipsoid will deviate from the true center of the person.

After convergence is reached, satisfied a pre-defined tolerance value, all information regarding the geometry of the ellipsoid is encoded within a 3×3 variance-covariance matrix. Principal Component Analysis (PCA) is then applied for finding the eigenvalues and eigenvectors of the matrix. For a covariance matrix representing a set of random data, the largest eigenvalue corresponds to the dimension and direction with the strongest correlation in the dataset, also known as the *principal component*. However, for the covariance matrix produced by the MVEE algorithm, the length of every eigenvector not only represents the amount of correlation in the corresponding direction, but captures the complete variation of the data in that direction.

Let $(\lambda_1, \lambda_2, \lambda_3)$ represent the eigenvalues of a 3×3 covariance matrix $\mathbf{C}$, satisfying the condition 5.2. For a human figure, the first eigen-value corresponding to the principal component should be much larger than the other two eigen-values ($\lambda_1 \gg \lambda_2 > \lambda_3$) due to the shape of the human figure.

If the total variation of the dataset (expressed in %) is equal to $\lambda_T = \sum\limits_{i=1}^{3} \lambda_i$, then each

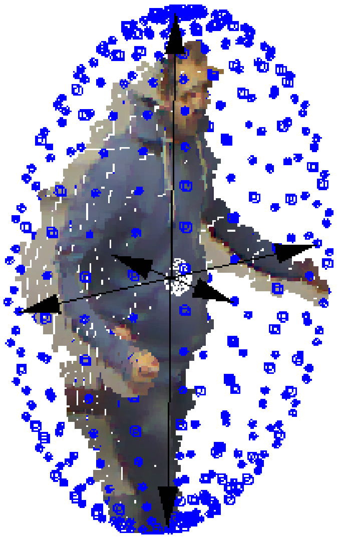


Figure 5.4: Ellipsoid representation of a human figure. (**Best viewed in colour**)

semi-major axis a, b and c will have a total amount of variation corresponding to:

$$\text{var}_a = \frac{\lambda_1}{\lambda_T} \times 100, \quad \text{var}_b = \frac{\lambda_2}{\lambda_T} \times 100, \quad \text{var}_c = \frac{\lambda_3}{\lambda_T} \times 100 \tag{5.3}$$

The length of each semi-major axis will also be equal to:

$$a = \sqrt{\frac{1}{\lambda_1}}, \quad b = \sqrt{\frac{1}{\lambda_2}}, \quad c = \sqrt{\frac{1}{\lambda_3}} \tag{5.4}$$

Every major or minor axis intersects the surface of the ellipsoid at two points known as *vertex points* or *fuci* points. Amongst the variety of representations for parameterising the position of a vertex, the Cartesian representation is the most commonly used and it's formulated as follows:

$$\begin{aligned} \mathrm{X} &= \mathrm{X_c} + a \cos u \cos v \\ \mathrm{Y} &= \mathrm{Y_c} + b \cos u \sin v \\ \mathrm{Z} &= \mathrm{Z_c} + c \sin u \end{aligned} \tag{5.5}$$

where $(\mathrm{X}, \mathrm{Y}, \mathrm{Z})$ represents a vertex position, and (u, v) are the rotation angles of the axis intersecting the ellipsoid at that vertex. The rotation angles for every vertex point are provided in Tab. 5.1.

A local fictitious coordinate system was defined and positioned at the center of the ellipsoid, remaining invariant to the ellipsoid shape variations. Specifically, this local coordinate system acts as a reference for the rotation changes of the ellipsoid, providing some reliable information about the movement of the object in the scene. All three axes were assigned to a pre-defined reference axis for monitoring the changes performed from each ellipsoid representation. These *constraint rotation angles* were computed according to Algo. 5.

The term *human approximated zero angle movement* refers to the upright position of a standing person with his arms placed in vertical position. This implies that very small

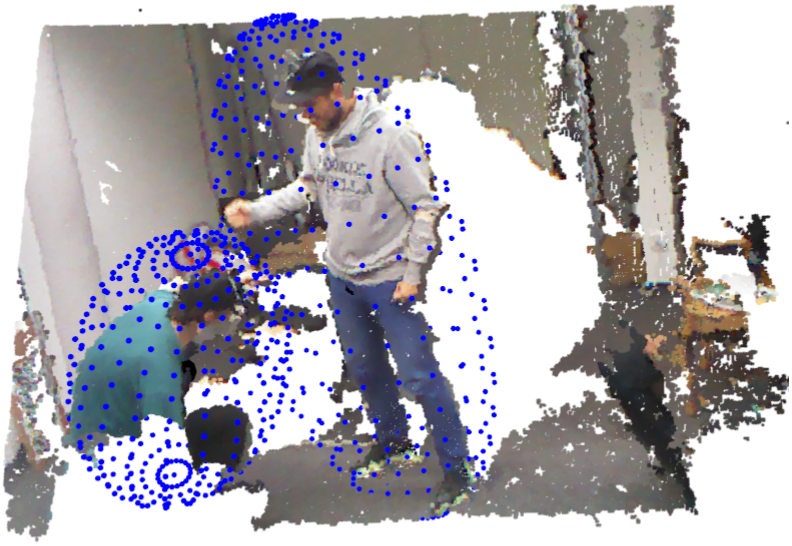


Figure 5.5: Ellipsoid representations of two people fighting. (**Best viewed in colour**)

	$\mathbf{x_a}$	$\mathbf{x'_a}$	$\mathbf{x_b}$	$\mathbf{x'_b}$	$\mathbf{x_c}$	$\mathbf{x'_c}$
u	0	π	0	0	$\frac{\pi}{2}$	$-\frac{\pi}{2}$
v	0	0	$\frac{\pi}{2}$	$-\frac{\pi}{2}$	π	$-\pi$

Table 5.1: Rotation angles for every semi-major and semi-minor axes of the ellipsoid. Every axis intersects the surface of the ellipsoid at a specific position based on a set of two angles (u, v).

variations are present, depicted by the very small values of the constrained rotation angles. As stated at an earlier point in this section, every axis of the ellipsoid is assigned to one of the axis of the reference system. Specifically, the reference axis $\mathbf{X}$ is assigned to the ellipsoid axis $\mathbf{b}$, characterising the width of the person, the $\mathbf{Y}$ axis to the $\mathbf{a}$ axis corresponding to the depth of the person and the $\mathbf{Z}$ axis to the axis $\mathbf{c}$ representing the height of the person.

Finally, the size of the human figure was approximated by the volume of the ellipsoid computed by:

$$V = u_0 \det \left(\mathcal{C}^{-1}\right)^{-1/2} \tag{5.12}$$

where u_0 represents the volume of the unit hypersphere in n dimensions and its equal to $4\pi/3$ for a 3D space. The geometric representation of the proposed approach can be visualised in Fig. 5.3.

5.3 Experimental Results

Different types of human poses were captured in an indoor simulated environment (see setup configuration in Appx. B) for the purpose of evaluating the accuracy of Algo. 4. The

Algorithm 5 Angle of each ellipsoid axis with respect to a 3D local reference system

```
Require: $\mathbf{x_a}, \mathbf{x'_a}, \mathbf{x_b}, \mathbf{x'_b}, \mathbf{x_c}, \mathbf{x_c}'$
Compute the direction vector of each ellipsoid axis:
    $\mathbf{x''_a} = \mathbf{x_a} - \mathbf{x'_a}, \quad \mathbf{x''_b} = \mathbf{x_b} - \mathbf{x'_b}, \quad \mathbf{x''_c} = \mathbf{x_c} - \mathbf{x'_c} \tag{5.6}$
Compute angles $\omega_a$, $\omega_b$ and $\omega_c$ corresponding to a predefined reference axis:
    $\omega_a = \arccos\left(\frac{\mathbf{x''_a} \cdot \mathbf{Y}}{\|\mathbf{x''_a}\| \cdot \|\mathbf{Y}\|}\right)$
    $\omega_b = \arccos\left(\frac{\mathbf{x''_b} \cdot \mathbf{X}}{\|\mathbf{x''_b}\| \cdot \|\mathbf{X}\|}\right) \tag{5.7}$
    $\omega_c = \arccos\left(\frac{\mathbf{x''_c} \cdot \mathbf{Z}}{\|\mathbf{x''_c}\| \cdot \|\mathbf{Z}\|}\right)$
Check in which octant area each direction vector corresponds to:
    $\mathrm{pos}_{\mathrm{x''_a}} \leftarrow \mathrm{CheckOctantArea}(\mathbf{x''_a})$
    $\mathrm{pos}_{\mathrm{x''_b}} \leftarrow \mathrm{CheckOctantArea}(\mathbf{x''_b}) \tag{5.8}$
    $\mathrm{pos}_{\mathrm{x''_c}} \leftarrow \mathrm{CheckOctantArea}(\mathbf{x''_c})$
if $\mathrm{pos}_{\mathbf{x''_a}}$ is within octants V, VI, VII or VIII then
    $\omega_a = -\omega_a \tag{5.9}$
end if
if $pos_{\mathbf{x''_b}}$ is within octants V, VI, VII or VIII then
    $\omega_b = -\omega_b \tag{5.10}$
end if
if $\mathrm{pos}_{\mathbf{x''_c}}$ is within octants III, IV, VII or VIII then
    $\omega_c = -\omega_c \tag{5.11}$
end if
return $\omega_a, \omega_b, \omega_c$
```

robustness of the proposed filtering method was compared against the original approach of Kammerl et al. [12] and a simple 3D background subtraction approach that classifies every voxel in the scene as foreground if the difference to its corresponding point in the reference static cloud is larger than a predefined value. Ground truth foreground masks were generated using the approach of Shotton et at. [30], a depth-based human pose algorithm that was later on commercialised for Kinect games. Figure 5.8 shows results from different behaviours in a scene acquired from a multi-Kinect sensor system. It is

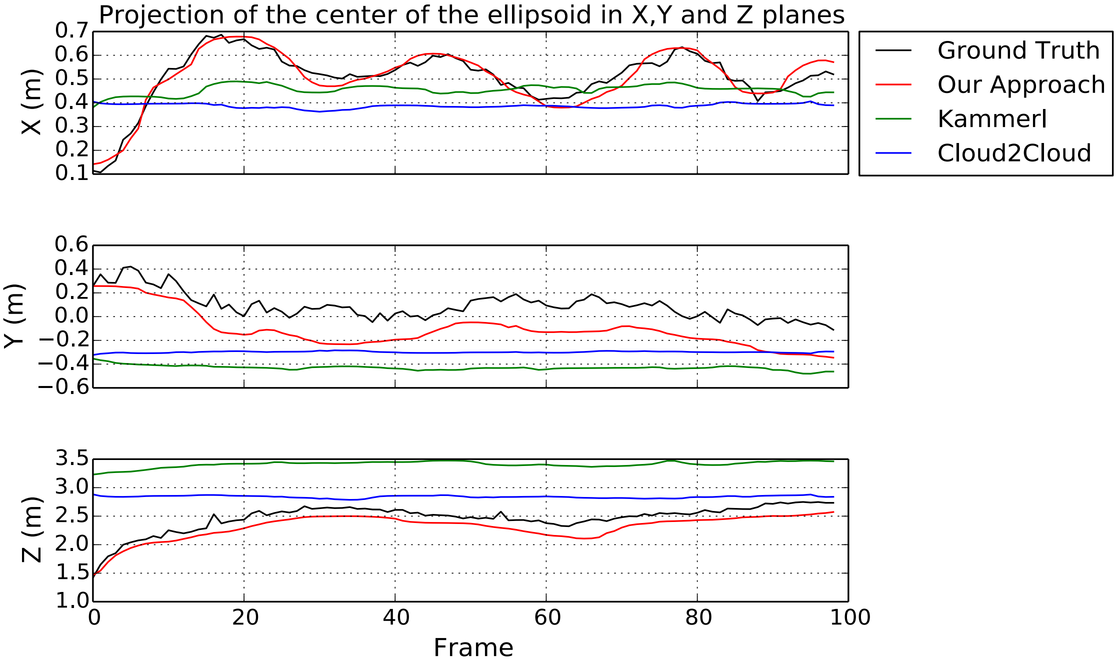


Figure 5.6: Trajectories of the center of the ellipsoid projected in **X**, **Y** and **Z** planes. (**Best viewed in colour**)

clear that the proposed method outperforms the other two approaches, as it is able to filter most of the noisy foreground blobs, resulting to a more accurate foreground mask.

According to Appx. B, evaluation and testing was performed in an indoor environment that emulates the internal part of a train wagon. However, existing state-of-the-art depth-based human pose estimators (see Sect. 2.2) are not able to cope with the different environmental changes in the scene such as fast illumination changes and partial occlusions, leading to erroneous predictions. Some of the reasons to be discussed is either because the viewing angle of the Kinect sensors is inappropriate for these kind of algorithms or because feature based approaches are highly sensitive to noisy depth maps or because features based approaches are not suitable enough to capture the complexity of the human shape.

The parameters involved in the proposed workflow were empirically defined after an extensive evaluation and testing: the leaf size of the octree, which controls the accuracy of the foreground mask was set to 0.1 m. The trimming of the point cloud was performed using a pass-through filter, preserving all voxels within the range of 4 m. Every contour in the binary mask with a size less than 1000 or larger than 7000 pixels was removed. Finally, the global distance threshold for assigning a voxel to the foreground using the Euclidean distance between a static background cloud and the current cloud was set to 5 cm.

A Kalman filter [11] was applied to smooth all extracted information of the ellipsoid and remove noisy representations in the data sequence.

Figure 5.6 shows the trajectory of a person for the first 100 frames of a random sequence along with the corresponding ground truth track. For a better visualisation of the 3D trajectory, each dimension was mapped in its own coordinate plane. One can visually observe that the proposed method follows the motion of the person much better compare to the other two approaches. This could be explained by the fact that the foreground

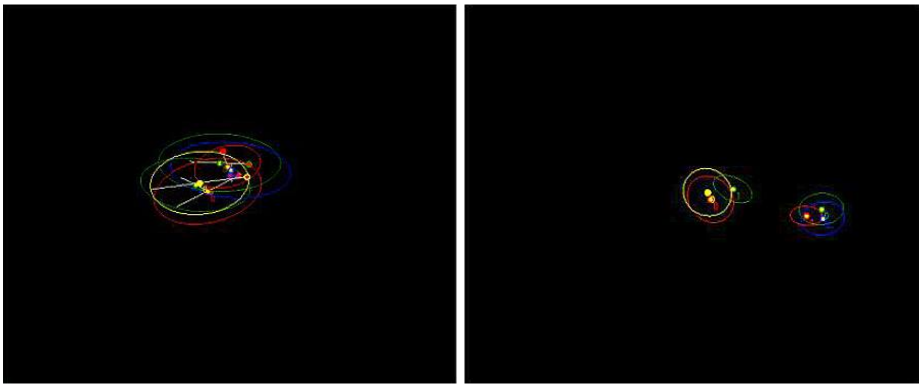


Figure 5.7: Tracking software. (a) Shape variations of a single person and (b) two people in the scene from a network of four Kinect sensors. (**Best viewed in colour**)

noise is randomly distributed in the complete scene, which forces the ellipsoid to also incorporate voxels that appear in extreme regions.

The produced trajectories were evaluated against the ground truth trajectory using the following likelihood formulation:

$$L = \sum_{t=1}^{n} \frac{\parallel \mathbf{X}_t - \mathbf{X}_t^{GP} \parallel}{\parallel \mathbf{X}_t^{GP} \parallel} \tag{5.13}$$

where $\mathbf{X}_t$ represents the position of the person in the scene at time t and $\mathbf{X}_{GP}^t$ is the corresponding ground truth position of the person at time t. The evaluation was carried out on a single person in the scene, achieving a likelihood of $82.4\,\%$ for the proposed method, $55.4\,\%$ for the original method of Kammerl et al. [12] and $38.6\,\%$ for the cloud to cloud background subtraction approach. One of the main drawbacks of the proposed method is the sudden increase of the size of the ellipsoid caused by the interaction between two or more human instances in the scene. Although this problem is controlled given a minimum and maximum size of an accepted blob size, it still remains an unsolved issue and should be investigated in a future work. All parameters of the ellipsoid were saved in an XML file (see Appx. D) and imported in a tracking visualiser[2] for monitoring and tracking the shape deformations of a person in the scene (see Fig. 5.7). The visual outcome of the software was provided to psychologists in the anthropology and disaster management field for classifying the behaviour of the people based on their shape deformations.

5.4 Conclusions

In this chapter a method was proposed for capturing and tracking people's temporal shape deformations in a dynamic indoor environment using Kinect-like RGBD data. The proposed methodology consisted of two components: (1) a workflow that enhances the accuracy of Kammerl's [12] foreground estimation algorithm and (2) the use of the Minimum Volume Enclosing Ellipsoid algorithm introduced by Moshtagh [13] for

[2] Visualisation was provided by an industrial partner

capturing the spatio-temporal changes of the moving objects in a 3D scene. For the first part, the filtering pipeline proposed in Algo. 4 showed a significant improvement over the original approach, providing accurate foreground masks even under different environmental conditions. The complete filtering workflow was performed in real-time independently from the amount of people present in the scene.
In the second part, the foreground mask from the previous step was used by the Minimum Volume Enclosing Ellipsoid algorithm for finding the best fitted ellipsoid representation for the given human pose. Results showed that the computational time of the algorithm does not affect the overall real-time performance of Algo. 4 due to the few number of silhouette points proposed by the convex hull algorithm. However, the accuracy of the parameters extracted from the variance-covariance matrix of the ellipsoid, depend highly on the quality of the convex hull, which in turn depends on the accuracy of the foreground mask. Kalman filter was used for smoothing the parameters of the ellipsoid, removing noisy measurements and providing a better representation of the motion.

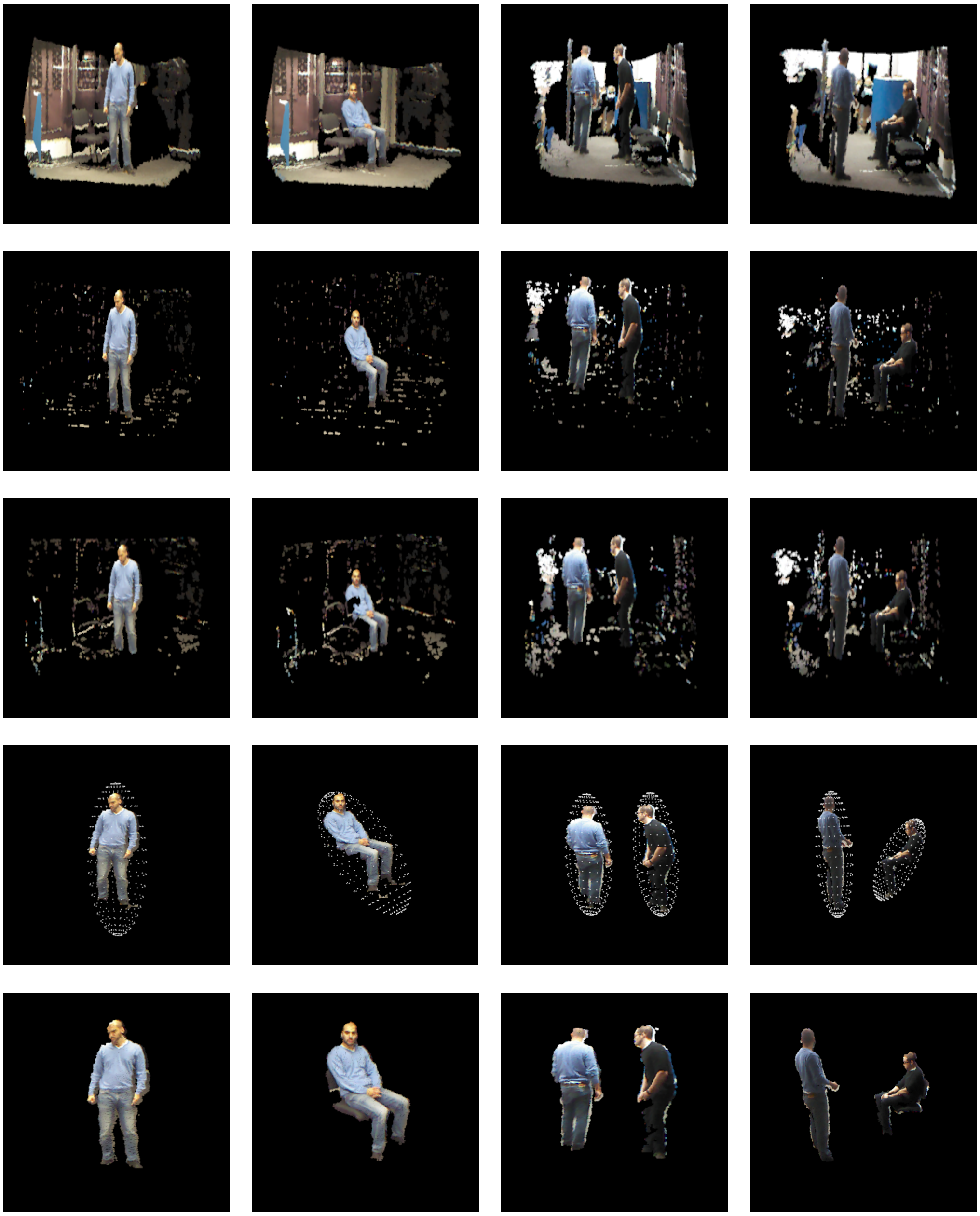


Figure 5.8: Some qualitative results of the proposed method. From top to bottom row: raw point clouds; foreground masks extracted from the original method of Kammerl et al. [12]; foreground masks extracted from the cloud to cloud approach; ellipsoids encapsulating the foreground mask from the proposed method; ground truth masks generated from the depth-based human pose estimation algorithm introduced by Shotton et al. [30], implemented in the OpenNI [96] framework. **(Best viewed in colour)**

Chapter 6

Towards a Multi Camera 3D Object Recognition System

6.1 Introduction

Human recognition has been an actively field of research from the early beginning of computer vision, maintaining its popularity due to its wide range of applications. While traditional approaches rely mostly on image-based information, with the appearance of low-cost commodity sensors, such as the Microsoft Kinect, a new field of research emerged, adding some advantages in the field. One of the most challenging problems in human recognition is the identification of humans under partial occlusion either in the form of intra-occlusion with other human instances or with respect to other objects in the scene. While much research has been done in this direction, distinguishing between instances depends strongly on the amount of their overlapping but also from the complexity of the environment. In order to overcome this problem, additional information of the human instances is required from different viewing angles of the scene.

Making use of the real-time RGB and depth information of the Kinect sensor, the aforementioned problems could be performed directly in 3D space. Working purely with 3D data has definitely some advantages that should be utilised to overcome the aforementioned problems. Development of such a multi-sensor 3D object recognition system requires the integration of information from all sensors present in the scene.

Therefore, the purpose of this chapter is to perform a preparatory work for a potential multi-Kinect object recognition system, introducing a workflow for assessing the reliability of merging point clouds from different sensors. The proposed work could be very useful for future object recognition applications, where accurate combination of 3D data maybe required.

6.2 Single Camera Orientation

6.2.1 Pinhole Camera Model

Pinhole camera model is known to be a special case of a *general projective camera model*. The origin of the model, which is also the origin of a Euclidean coordinate system, is defined at the centre of projection, where all points in 3D space are mapped from. The

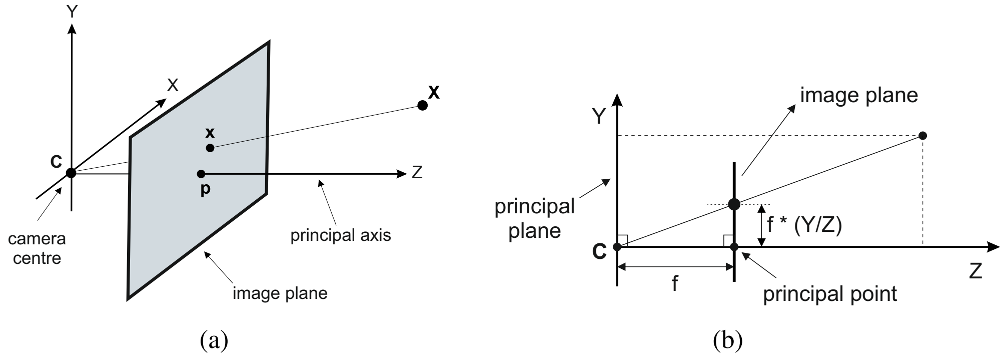


Figure 6.1: Pinhole camera geometry. (a) Pinhole camera geometry, where $\mathbf{C}$ represents the projection center, $\mathbf{p}$ the principal point and $\mathbf{X}$ a point in 3D space; (b) shows a projection of the pinhole camera model in the $\mathbf{Y}$ plane, demonstrating how a 3D point is mapped on the image plane positioned at a distance $\mathrm{Z} = f$ from the principal plane. (**Source:** The above figures have been reproduced with small modifications from the textbook of Hartley and Zisserman [94])

plane defined by the $\mathbf{X}$ and $\mathbf{Y}$ axis of the coordinate system is called the *principal plane* and the plane where $\mathrm{Z} = f$, is known as the *image plane* or *focal plane*. Every point $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z})$ in 3D space is mapped to a point on the image plane, defined by the intersection of the line connecting point $\mathbf{X}$ with the projection centre and the image plane. Also, the $\mathbf{Z}$ axis meets the image plane at the *principal point*, which is the mapping of the projection center (also known as *camera center*) on the image plane. Working with homogeneous coordinates in projective space, a 3D point $\mathbf{X}$ can be mapped to the point $\mathbf{x} = [f\mathrm{X}, f\mathrm{Y}, \mathrm{Z}]$ plane defined at depth Z through the following transformation matrix:

$$\begin{bmatrix} f\mathrm{X} \\ f\mathrm{Y} \\ \mathrm{Z} \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{bmatrix} \tag{6.1}$$

where the 3×4 matrix represents the *ideal camera projection matrix* with 7 DOF. Equation 6.1 can be expressed in a more compact representation as follows:

$$\mathbf{x} = diag(f, f, 1)[\mathbf{I} \mid \mathbf{0}]\mathbf{X} \tag{6.2}$$

The aforementioned transformation is a simple linear relation, which relates the projected 3D point to a 2D point, positioned on an plane at depth Z. For getting the corresponding 2D point defined on the image plane at $\mathrm{Z} = 1$, all elements of the point $\mathbf{x}$ must be divided by Z.

When the principal point does not go through the center of the image plane, but rather deviates from the ideal point in both x and y directions, then the ideal camera projection matrix is upgraded to the *Euclidean camera projection matrix* case with 9 DOF taking the following form:

$$\begin{bmatrix} f\mathrm{X} + \mathrm{Z}p_x \\ f\mathrm{Y} + \mathrm{Z}p_y \\ \mathrm{Z} \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{bmatrix} \tag{6.3}$$

where $\mathbf{p} = [p_x, p_y]$ are the parameters (in pixel units) that define the offset from the ideal point.

However, in the case of CCD cameras, the shape of a pixel may not be strictly square, introducing an additional scaling factors m_x and m_y for the $\mathbf{X}$ and $\mathbf{Y}$ axis respectively. Furthermore, adding a skewness factor s representing the non orthogonality of the image axes and its for most normal cameras set to zero. This is considered as the *finite* representation of the camera projection matrix, has 11 DOF and its expressed in the following form:

$$\mathbf{P} = \begin{bmatrix} c_x & s & x_0 & 0 \\ 0 & c_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{6.4}$$

where $c_x = m_x f, c_y = m_y f$ represent the focal length of the camera in terms of pixels dimensions, s corresponds to the skewing parameter defined by the ratio c_y/c_x and $x_0 = m_x p_x, y_0 = m_y p_y$ express the principal point in pixel dimensions. Substituting 6.4 in 6.1, the 2D$\leftrightarrow$3D mapping can be compactly represented by:

$$\mathbf{x} = \mathbf{K}[\mathbf{I} \mid \mathbf{0}]\mathbf{X} \tag{6.5}$$

where,

$$\mathbf{K} = \begin{bmatrix} c_x & s & x_0 \\ 0 & c_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.6}$$

is a 3×3 upper triangular matrix, known as the *calibration matrix*. The importance of this matrix is that it encapsulates all internal (intrinsic) information of the camera.

Equation 6.5 considers that the position of the camera is at the origin of the Euclidean coordinate system. Generally, all 3D points are defined within a coordinate system known as the *world coordinate system*. The relationship of this system with respect to a camera system is expressed by a non-ideal 3D rigid transformation matrix. Thus, for a 3D point to be mapped in the camera coordinate system, the following transformation should hold:

$$\mathbf{X} = \mathbf{R}(\mathbf{X_w} - \mathbf{C}) \tag{6.7}$$

where $\mathbf{C} = [\mathrm{X_c}, \mathrm{Y_c}, \mathrm{Z_c}]$ represents the coordinates of the camera centre in the world coordinate system, $\mathbf{X_w}$ is a 3D point defined in the world coordinate frame and $\mathbf{R}$ is a 3×3 rotation matrix representing the orientation of the camera coordinate with respect to the world coordinate frame. Extending relation 6.5, Eq. 6.7 takes the form:

$$\mathbf{x} = \mathbf{KR}[\mathbf{I} \mid -\mathrm{C}]\mathbf{X} \tag{6.8}$$

where the projection camera matrix $\mathbf{P} = \mathbf{KR}[\mathbf{I} \mid -\mathbf{C}]$ performs a *general mapping* of a pinhole camera model and has 11 DOF.

In the area of photogrammetry, the 3×4 projection camera matrix $\mathbf{P}$ is a reformulation of the well known *collinearity equation*, expressed by:

$$\begin{aligned}\mathrm{x} &= x_0 + f\,\frac{r_{11}(\mathrm{X}-\mathrm{X}_0)+r_{12}(\mathrm{Y}-\mathrm{Y}_0)+r_{13}(\mathrm{Z}-\mathrm{Z}_0)}{r_{31}(\mathrm{X}-\mathrm{X}_0)+r_{32}(\mathrm{Y}-\mathrm{Y}_0)+r_{33}(\mathrm{Z}-\mathrm{Z}_0)}\\ \mathrm{y} &= y_0 + f\,\frac{r_{21}(\mathrm{X}-\mathrm{X}_0)+r_{22}(\mathrm{Y}-\mathrm{Y}_0)+r_{23}(\mathrm{Z}-\mathrm{Z}_0)}{r_{31}(\mathrm{X}-\mathrm{X}_0)+r_{32}(\mathrm{Y}-\mathrm{Y}_0)+r_{33}(\mathrm{Z}-\mathrm{Z}_0)}\end{aligned} \tag{6.9}$$

where (x_0, y_0, f) are the interior camera parameters, $(\mathrm{X}_0, \mathrm{Y}_0, \mathrm{Z}_0)$ represents the position of the camera in the world coordinate system and $(r_{11}, \ldots, r_{33})$ are the rotation parameters of a right handed 3×3 rotation matrix $\mathbf{R}$. By definition, all parameters in the collinearity equation should be expressed in metric units such as meters, centimetres or millimetres. For strictly square pixels, the focal length f in the collinearity equation will be equal to $f = c_x = c_y$.

6.2.2 Lense distortions

In photography, two types of distortions exist: *optical* and *perspective*. While optical distortion is caused by the optical design of the lens (also known as "lens distortion"), perspective distortion is caused by the position of the camera relative to the object of interest or by the position of the object within the image frame. Within this context, image points will be corrected for errors caused by optical distortions. This process of correction is known as *camera resectioning* or *camera calibration*.

Camera calibration involves finding the parameters that will eliminate the offset in the observed image points caused by the lens. Most important type of optical distortion is the *radial (symmetric) distortion*, which causes an inward or outward position of the image point from its ideal position. This error constitutes the major imaging error for most camera systems and its mathematically defined by:

$$\begin{aligned}\Delta \mathrm{x}_{\text{rad}} &= \mathrm{x}\,[k_1 r^2 + k_2 r^4 + k_3 r^6 + \ldots]\\ \Delta \mathrm{y}_{\text{rad}} &= \mathrm{y}\,[k_1 r^2 + k_2 r^4 + k_3 r^6 + \ldots]\end{aligned} \tag{6.10}$$

where (x, y) are the distorted coordinates of an image point, r defines the *image radius* of the image point from the image's principal point, and $(k_1, \ldots, k_n)$ are the radial distortion coefficients that model the radial distortion curve. According to Eq. 6.10, the distortion curve is modelled with a polynomial series (Seidel series) and has a quadratic representation, which increases depending on the radius distance of an image point to the principal point. For most standard types of lenses, corrections larger than the third order parameter $(\leq k_3)$ could be neglected without any significant loss in the accuracy of the points.

Furthermore, the sign of the distortion coefficients defines the form of the distortion. According to Fig. 6.2, radial distortions can be represented either by a *barrel distortion* or a *pincushion distortion*. It is generally said, that negative values of the distortion coefficients correspond to a barrel distortion and positive values to a pincushion distortion. Nevertheless, cases exist where a combination of these two distortions may occur, leading to a distortion known as *mustache distortion*.

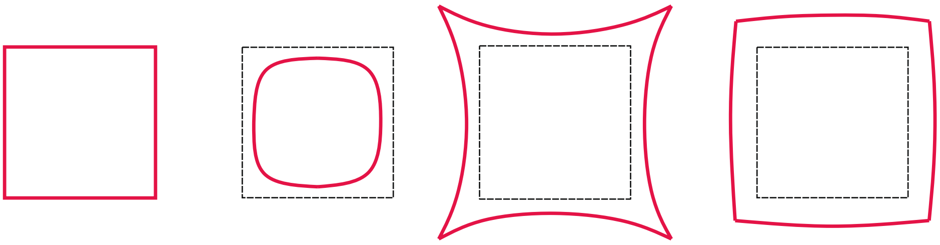


Figure 6.2: Types of radial distortion curves. From left to right: No distortion, Barrel distortion, Pincushion distortion and Mustache distortion. (**Best viewed in colour**)

Second form of distortion is the *decentering* (or tangential) distortion, which is caused by physical elements in a lens not being perfectly aligned to the image plane. The source of this error is mostly due to manufacturing defects, and can be compensated by the following equation:

$$\begin{aligned}\Delta \mathrm{x}_{\mathrm{dec}} &= p_1(r^2 + 2\mathrm{x}^2) + 2p_2\mathrm{xy} \\ \Delta \mathrm{y}_{\mathrm{dec}} &= p_2(r^2 + 2\mathrm{y}^2) + 2p_1\mathrm{xy}\end{aligned} \tag{6.11}$$

where p_1, p_2 correspond to the decentering parameters. This lens correction part can give large values for low cost lenses (such as the ones embedded in surveillance cameras) and smaller quantity distortion values for high quality lenses (*e.g. space cameras*).
Finally, the *affinity* and *shearing* parameters are used to describe deviations of the image coordinate system with respect to the non-orthogonality and uniform scaling of the coordinate axes. This is mathematically expressed by:

$$\begin{aligned}\Delta \mathrm{x}_{\mathrm{aff}} &= b_1\mathrm{x} + b_2\mathrm{y} \\ \Delta \mathrm{y}_{\mathrm{aff}} &= 0\end{aligned} \tag{6.12}$$

where b_1, b_2 correspond to the affinity and shearing parameters respectively. It is noteworthy that for the majority of cameras used in close range applications, b_1 and b_2 can be neglected as they are set to zero.
Finally, all individual error terms could be summarised a follows:

$$\begin{aligned}\Delta \mathrm{x} &= \Delta \mathrm{x}_{\mathrm{rad}} + \Delta \mathrm{x}_{\mathrm{dec}} + \Delta \mathrm{x}_{\mathrm{aff}} \\ \Delta \mathrm{y} &= \Delta \mathrm{y}_{\mathrm{rad}} + \Delta \mathrm{y}_{\mathrm{dec}} + \Delta \mathrm{y}_{\mathrm{aff}}\end{aligned} \tag{6.13}$$

Considering a radial distortion correction up to k_3, all optical lens distortion parameters $k_1, k_2, k_3, p_1, p_2, b_1, b_2$ together with the focal length f and principal point x_0, y_0, define the Brown 10-parametric calibration model [97].
Incorporating the optical correction terms from 6.13 into the collinearity Eq. 6.9, a more complete mathematical representation of the collinearity equations is given by:

$$\begin{aligned}\mathrm{x} &= x_0 + f\,\frac{r_{11}(\mathrm{X}-\mathrm{X}_0) + r_{12}(\mathrm{Y}-\mathrm{Y}_0) + r_{13}(\mathrm{Z}-\mathrm{Z}_0)}{r_{31}(\mathrm{X}-\mathrm{X}_0) + r_{32}(\mathrm{Y}-\mathrm{Y}_0) + r_{33}(\mathrm{Z}-\mathrm{Z}_0)} + \Delta\mathrm{x} \\ \mathrm{y} &= y_0 + f\,\frac{r_{21}(\mathrm{X}-\mathrm{X}_0) + r_{22}(\mathrm{Y}-\mathrm{Y}_0) + r_{23}(\mathrm{Z}-\mathrm{Z}_0)}{r_{31}(\mathrm{X}-\mathrm{X}_0) + r_{32}(\mathrm{Y}-\mathrm{Y}_0) + r_{33}(\mathrm{Z}-\mathrm{Z}_0)} + \Delta\mathrm{y}\end{aligned} \tag{6.14}$$

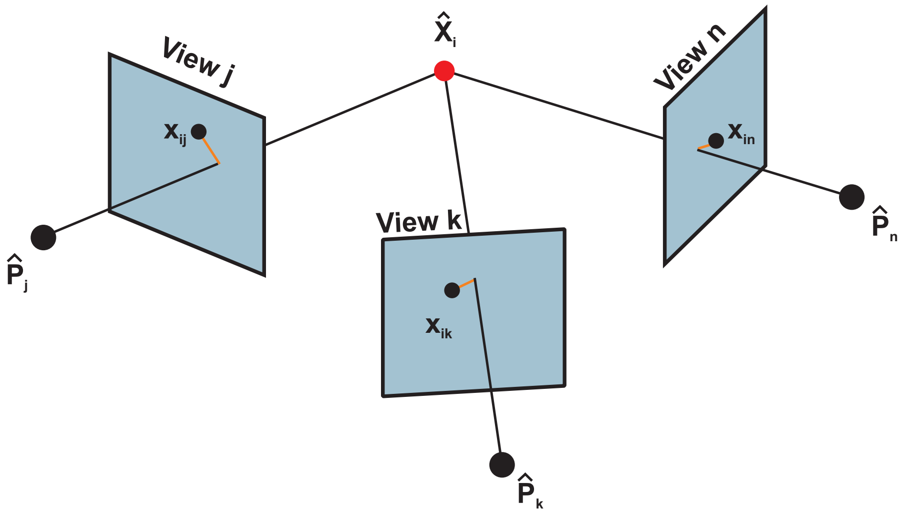


Figure 6.3: Bundle adjustment problem. Bundle adjustment from n images. The optimum 3D point $\mathbf{X}$ is reconstructed from all n images, finding the optimal camera poses and corresponding 2D points that will minimise the objective cost function 6.15. (**Best viewed in colour**)

6.3 Bundle Adjustment

Bundle adjustment is defined as the problem of *jointly* estimating optimal 3D structure and camera pose parameters. It was originally conceived in the field of photogrammetry during the 1950s' (Triggs et al. [98]) and has been extensively used by the computer vision community during recent years. It is a process involving three forms of observations: camera pose parameters, image points and corresponding 3D points. After performing a bundle adjustment, image points, 3D points and cameras pose parameters should minimise some form of a cost function. This boils down to minimising the *reprojection error* between the observed image points and the predicted image points. As this is a non-linear problem, the number of iterations required depends from several factors, such as accuracy of feature points, initial camera pose parameters, viewing angle, images overlapping, etc. Formally, the objective cost function is given by:

$$\min_{\hat{\mathbf{P}}_\mathrm{i},\hat{\mathbf{X}}_\mathrm{j}} \sum_{i=1}^{m}\sum_{j=1}^{n} \left\| \mathbf{x}_\mathrm{ij}, \hat{\mathbf{x}}_\mathrm{ij} \right\|^2, \quad \hat{\mathbf{x}}_\mathrm{ij} = \hat{\mathbf{P}}_\mathrm{i}\hat{\mathbf{X}}_\mathrm{j} \tag{6.15}$$

where $\mathbf{x}_\mathrm{ij}$ corresponds to the image point j of image i, $\hat{\mathbf{x}}_\mathrm{ij}$ is the corrected image point computed by the updated projection camera matrix $\hat{\mathbf{P}}_\mathrm{i}$, and $\hat{\mathbf{X}}_\mathrm{j}$ is the corrected 3D point. Depending on the input data, some parameters such as the internal calibration parameters, might have already been optimised from a previous calibration of the cameras. Thus, these parameters should be declared as constant during the minimisation process.

As was previously stated, every observed image point $\mathbf{x}_\mathrm{ij}$ can be represented by a set of two collinearity equations. Let's define a 1-dimensional *observation vector* $\mathbf{l}$, containing all image points observed across images m. These image points should be valid *tie* points lying in the overlapping region of two or more image. Furthermore, if k represents the number of unknown parameters to be optimised, let $\mathbf{dx}$ be a $k \times 1$ *correction vector* of the unknowns. If the number of observations is larger than the number of unknowns (over-determined non-linear system), an adjustment method is required for estimating the unknown parameters. The *Gauss-Markov linear model* is a least-squares (LS) adjustment method that could be used for this purpose. This solver is based on the assumption that the observations and unknowns have a *functional relation* to each other.

Since the collinearity equations are non-linear functions, they should be linearised using a Taylor series expansion of the function with respect to each of the unknown parameters. This will result in a linear function of the original collinearity form 6.14, evaluated on some estimated values of the unknowns. After linearisation, the resulting system of observation equations can be formulated with the following *functional model*:

$$\underset{\{2mn\times1\}}{\mathbf{l}} + \underset{\{2mn\times1\}}{\boldsymbol{\nu}} = \underset{\{2mn\times k\}}{\mathbf{J}}\underset{\{k\times1\}}{\mathbf{d\hat{x}}} \tag{6.16}$$

where $\mathbf{J}$ is the *design* or *Jacobean* matrix, containing the partial first-order derivatives of the observations with respect to the unknowns, evaluated on the approximated values of the unknowns, and $\boldsymbol{\nu}$ is the vector of residuals. Each component in $\boldsymbol{\nu}$ is equal to the difference between the observation l and the corresponding predicted value from the initial guess. Concretely, the residuals are as follows:

$$\underset{\{2mn\times1\}}{\boldsymbol{\nu}} = \underset{\{2mn\times k\}}{\mathbf{J}}\underset{\{k\times1\}}{\mathbf{d\hat{x}}} - \underset{\{2mn\times1\}}{\mathbf{l}} \tag{6.17}$$

The normal system of equations is given by:

$$\underset{\{k\times k\}}{\mathbf{N}}\underset{\{k\times1\}}{\mathbf{d\hat{x}}} + \underset{\{k\times1\}}{\mathbf{n}} = \underset{\{k\times1\}}{\mathbf{0}} \tag{6.18}$$

where,

$$\begin{aligned} \underset{\{k\times k\}}{\mathbf{N}} &= \underset{\{k\times2mn\}}{\mathbf{J^T}}\underset{\{2mn\times2mn\}}{\mathbf{W}}\underset{\{2mn\times k\}}{\mathbf{J}} \\ \underset{\{k\times1\}}{\mathbf{n}} &= \underset{\{k\times2mn\}}{\mathbf{J^T}}\underset{\{2mn\times2mn\}}{\mathbf{W}}\underset{\{2mn\times1\}}{\mathbf{l}} \end{aligned} \tag{6.19}$$

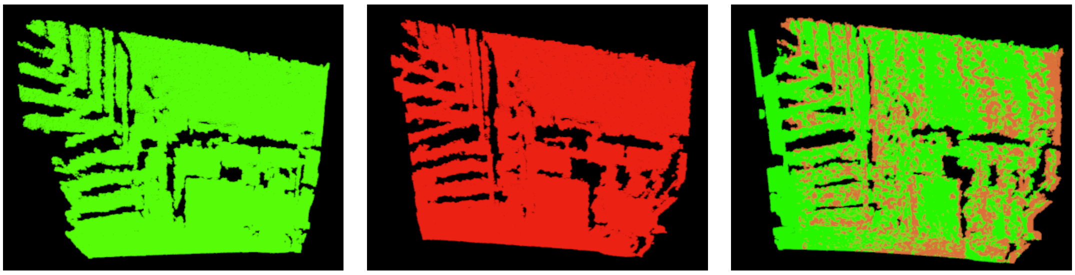
The matrix $\mathbf{N}$ is known as the *normal* matrix and $\mathbf{W}$ is a diagonal matrix containing the *weights* of the observations l. If no weights exist, this weighted matrix is set to unity.

Due to the non-linearity of the problem, an iteration process is required. This iteration process continues until a termination criterion is met. The most common conditions to be met is when no more changes in the correction vector appear or when a predefined number of iterations is reached. Solving Eq. 6.18 for $\mathbf{d\hat{x}}$, the solution vector of the unknowns is given by:

$$\underset{\{k\times1\}}{\mathbf{d\hat{x}}} = (\underset{\{k\times2mn\}}{\mathbf{J^T}}\underset{\{2mn\times2mn\}}{\mathbf{W}}\underset{\{2mn\times k\}}{\mathbf{J}})^{-1}\underset{\{k\times2mn\}}{\mathbf{J^T}}\underset{\{2mn\times2mn\}}{\mathbf{W}}\underset{\{2mn\times1\}}{\mathbf{l}} \tag{6.20}$$

After convergence, the standard deviation error of the unit weight of the unknowns σ_0^2 can be computed by:

$$\sigma_0^2 = \frac{\boldsymbol{\nu}^{\boldsymbol{T}}\mathbf{W}\boldsymbol{\nu}}{2m-k} \tag{6.21}$$



(a) (b) (c)

Figure 6.4: ICP registration example between two point clouds. Performing ICP registration between two point clouds capturing part of an interior staircase from different viewing angles; From left to right: (a) source point cloud, (b) target point cloud, (c) registered cloud. (**Best viewed in colour**)

The variance-covariance matrix of the unknowns $\mathbf{Q}$ is given by:

$$\underset{\{k\times k\}}{\mathbf{Q}} = \sigma_0^2 \, (\underset{\{k\times 2mn\}}{\mathbf{J}^{\mathrm{T}}} \underset{\{2mn\times 2mn\}}{\mathbf{W}} \underset{\{2mn\times k\}}{\mathbf{J}})^{-1} \tag{6.22}$$

which should result into a diagonal matrix where each element Q_{ii} represents the variance of every unknown.

The Gauss-Markov model is a well known mathematically model extensively used in the field of photogrammetry for solving bundle adjustment problems. However, applications lying in the computer vision domain, such as *Structure from Motion (SfM)* or *Simultaneous Localisation and Mapping (SLAM)* are expected to provide a real time 3D reconstruction or mapping of the scene. Thus, more robust objective cost functions or minimisation techniques are required that can provide a real time solution. Although several approaches have been proposed (Ni et al. [99], Agarwal et al. [100], Maier et al. [101]), *Levenberg-Marquardt* has proven to be the most successful, due to the existence of a damping factor that forces the objective function towards a fast convergence even for initial solution that are far away from convergence. Within this work, the classical Gauss-Markov mathematical model is used, due to the simplicity of the data and small number of sensors.

6.4 Registration of Point Clouds

The process of aligning a set of two point clouds representing the same object from a different view point is known as *registration*. The *Iterative Closest Point (ICP)* algorithm, introduced by Besl and McKay [102] is one of the most known and used registration method for performing these form of tasks. The basic principle of the ICP algorithm works is that one point cloud acts as the reference cloud (also known as *target*), while the other one, known as *source*, is transformed to best match the reference. This involves an iteration process where the source cloud undergoes a rigid transformation (that implies translation and rotation) for minimising the distance to the reference point cloud. Figure 6.4 shows a registration example between two point clouds capturing part of an interior staircase from different viewing angles. The complete staircase cloud is given in Fig. 6.4(c). A step by step explanation of the ICP process is provided by Algo. 6.

Algorithm 6 Iterative Closest Point (ICP) algorithm

```
Require: $\mathbf{P}$ (source) dataset; $\mathbf{M}$ (target) dataset; convergence threshold $T$
Ensure: Transformation $(\mathbf{R}', \mathbf{t}')$, error $\varepsilon$
$\mathbf{R}' \leftarrow \mathbf{I}, \mathbf{t}' \leftarrow \mathbf{0}, \varepsilon \leftarrow \infty$
while $(\varepsilon > T)$ do
    $\mathbf{Y} \leftarrow \{\mathbf{m} \in \mathbf{M} \mid \mathbf{p} \in \mathbf{P} : \mathbf{m} = \text{ClosestPoint}(\mathbf{p})\}$
    $(\mathbf{R}, \mathbf{t}, \varepsilon) \leftarrow \min_{\mathbf{R},\mathbf{t}} \sum_{k=1}^{n_p} |\mathbf{y_k} - (\mathbf{R}\mathbf{p_k} + \mathbf{t})|^2$
    $\mathbf{P} \leftarrow \mathbf{R} \cdot \mathbf{P} + \mathbf{t}$
    $\mathbf{R}' \leftarrow \mathbf{R} \cdot \mathbf{R}'$
    $\mathbf{t}' \leftarrow \mathbf{R} \cdot \mathbf{t}' + \mathbf{t}$
end while
return $\mathbf{R}', \mathbf{t}'$
```

6.5 Sensors Pose estimation and Bundle Adjustment

Finding the rotation and translation parameters of the sensors with respect to a world coordinate system requires a set of 2D↔3D point correspondences of a reference object in the scene. This was achieved by placing a regular chessboard in the center of the scene and setting its upper left corner as the origin of the reference system (see Fig. 6.5). Thus, every sensor could be oriented with respect to that corner.

Using a regular chessboard as a reference object for initialising a world coordinate system has the following advantages: *portability* and *explicitness*. For the latter, GCPs can be explicitly defined without using any measuring devices (error free GCPs) but only the dimensions and pattern size of the chessboard. This form of GCPs are known to be *degenerated* due to the lack of depth information, which is important for most camera pose estimators. For example, the Direct Linear Transformation (DLT) algorithm (Hartley and Zisserman [94]) is a well-known algorithm for finding the camera pose parameters. However, the main drawback of DLT is that it is not able to compute the pose parameters of a sensor from coplanar[1] reference object points. Thus, these cases require the use of more sophisticated algorithms as proposed by Fischler and Bolles [103]. Among the family of algorithms developed for this purpose, more attention is been given to a class of algorithms known as the "Perspective–*n*–Point", originally introduced by Fischler and Bolles [103] with many applications in Computer Vision and Robotics. This Perspective–*n*–Point problem" has received a great deal of attention in both the Photogrammetry (McGlove et al. [104]) and Computer Vision (Hartley and Zisserman [94]) communities. The aim of the P*n*P problem is to determine the EO of a sensor, given the IO parameters of the sensor and a set of n 2D↔3D correspondences between 3D points defined on a reference object and their 2D mapping.

Solutions to the P*n*P problem can be categorised into linear and non-linear. Within this

[1] Coplanar points are three or more points that lie in the same plane.

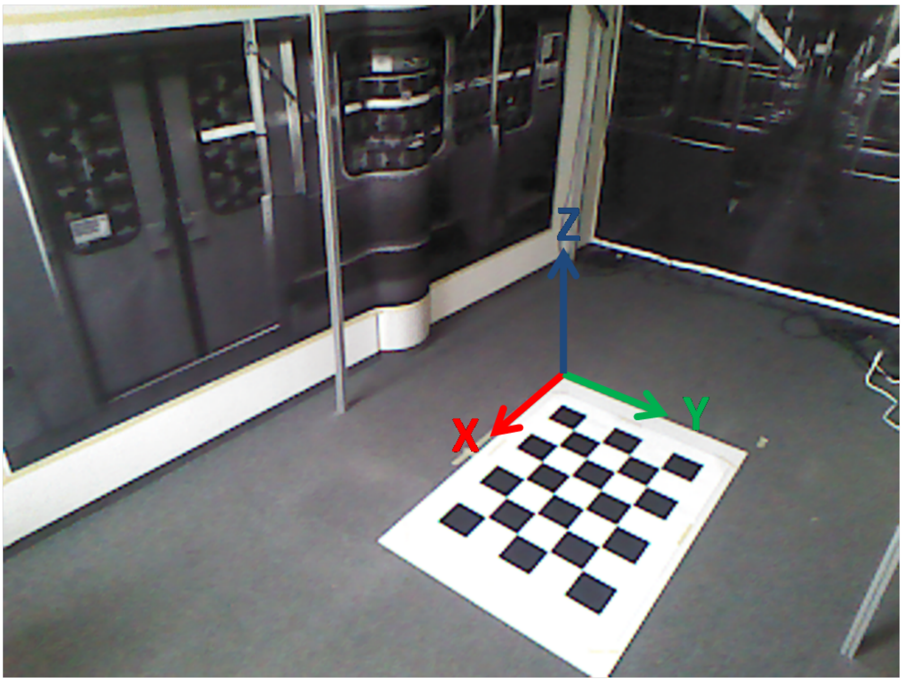


Figure 6.5: World coordinate system defined on a regular chessboard pattern. A large chessboard is placed at the center of the scene, with its upper left corner defined as the origin of the world coordinate system. (**Best viewed in colour**)

context, taking into consideration the planarity of the 3D coordinate system, two non-linear approaches were considered: PnP with Levenberg-Marquardt optimisation and the EPnP (Lepetit et al. [105]). The PnP-LM algorithm tries to find the EO parameters of the sensor by minimising the reprojection error expressed by Eq. 6.15. Likewise, the EPnP algorithm can provide a unique solution for both planar and non-planar cases iif the number of 2D $\leftrightarrow$ 3D correspondences is ≥ 4. The main idea of EPnP is to express the n number of 3D points as a weighted sum of four virtual control points.

The (X, Y) coordinates of the GCPs were generated by multiplying the current index of the rows and columns with the corresponding horizontal and vertical pattern size. Having the XY plane coinciding with the surface of the chessboard, the Z coordinates were set to zero, converting the *full* GCPs into *horizontal* GCPs. Figure 6.6 shows the optimised 2D projections of the corresponding GCPs after applying the EPnP algorithm (PnP-LM produces similar visual results). At first, the image points of all internal corners on the chessboard were found and refined by a subpixel gradient-based corner optimiser implemented in OpenCV [106]. Setting the IO parameters of the Kinect sensors to be fixed during the minimisation process, both the PnP-LM and EPnP algorithms were applied for finding the best EO parameters for all sensors. The corresponding translation and rotation values are provided in Tab. 6.1. Results showed that even though the difference between the EO parameters deduced from both methods are approximately in the same range, the main difference occurs in the RMS error. The PnP-LM algorithm is within the range of a quarter of a pixel whereas the EPnP algorithm is in the range of half a pixel. Therefore, one can perceive the sensitivity and importance of the initial chessboard 2D corner points in the minimisation process for achieving a low RMS error.

In the next step of the evaluation pipeline, a bundle adjustment was carried out for

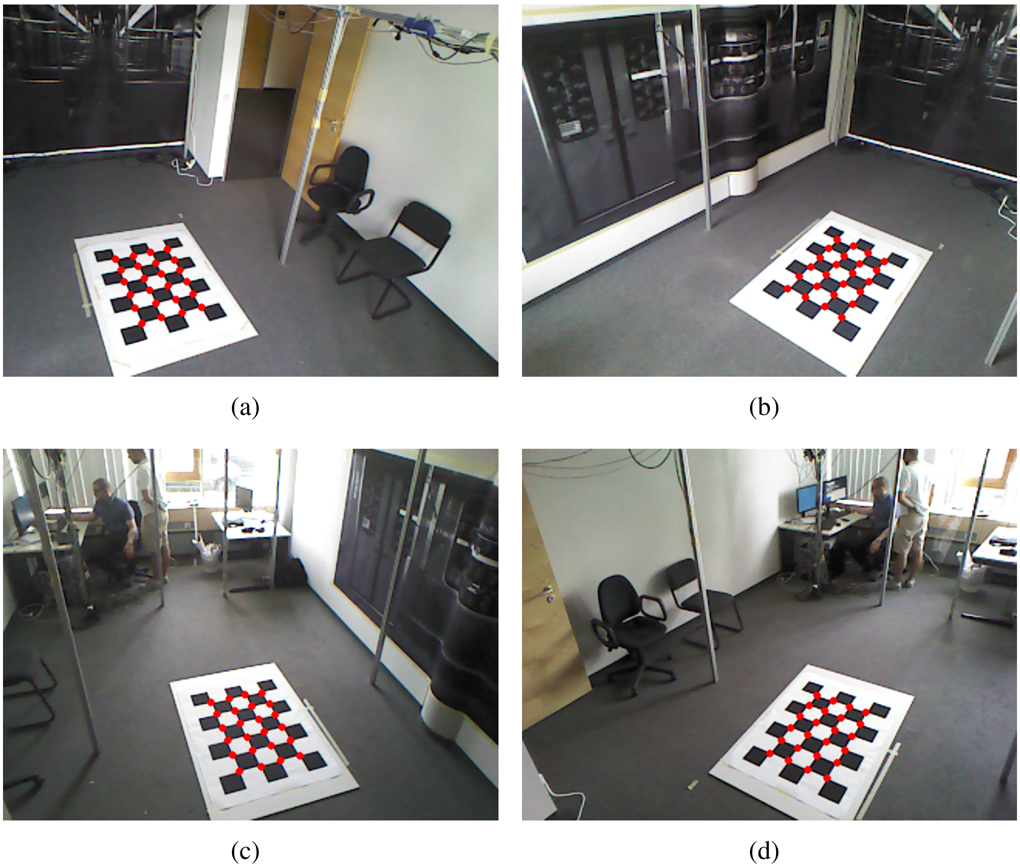


Figure 6.6: Mapped chessboard points of the corresponding ground control points. (**Best viewed in colour**)

optimising the EO of all four Kinect sensors, holding the IO parameters fixed and providing as initial values for the EO parameters the EO results from the previous step (see 6.1). The advantage of running a bundle process is that all observations and unknowns are linear dependent from each other as expressed in the Jacobean matrix $\mathbf{J}$ (see Sect. 6.3). Thus, the amount of correction in the correction vector $\mathbf{d}\hat{\mathbf{x}}$ is subject to this dependency. The minimisation condition 6.15 was set to 0.01 px and the maximum number of iterations to 20. The bundle converged after 8 and 14 iterations for the P*n*P-LM and EP*n*P algorithm respectively, resulting a standard deviation of the unit weight of $\sigma_0^{PnP-LM} = 0.211$ px and $\sigma_0^{EPnP} = 0.342$ px respectively. The small number of iterations, along with the very small standard deviation of the unit weights indicate the robustness of both the P*n*P algorithms. Another interpretation of the solution could be that the GCPs have a fixed location and geometry in 3D space and thus the mapping or projection of these points in the images could accurately be found using a corner detector.

6.6 Sequential Point Cloud Registration

Results from bundle adjustment were given as an input in the current step of the evaluation workflow for refining the 3D geometry of the scene. This task was achieved by performing a sequential registration of all point clouds of the same time stamp into a single cloud

	Sen.	$\mathbf{T_x}$ [m]	$\mathbf{T_y}$ [m]	$\mathbf{T_z}$ [m]	$\mathbf{R_x}$ [deg]	$\mathbf{R_y}$ [deg]	$\mathbf{R_z}$ [deg]	RMS [px]
P*n*P-LM	**A**	2.748	−0.135	2.175	119.232	1.368	−76.261	0.240
	B	2.690	0.768	2.128	124.905	−1.972	−91.559	0.247
	C	−1.576	1.139	2.146	120.264	1.428	98.205	0.255
	D	−1.575	−0.122	2.165	119.634	−0.818	85.313	0.231
EP*n*P	**A**	2.722	−0.161	2.345	121.639	1.828	−75.974	0.463
	B	2.649	0.729	2.259	127.082	−0.881	−91.444	0.432
	C	−1.565	1.148	2.246	121.639	1.250	98.377	0.389
	D	−1.547	−0.077	2.238	120.951	0.199	85.428	0.372

Table 6.1: Exterior orientation parameters of all Kinect sensors. Exterior orientation parameters and corresponding RMS error of all Kinect sensors defined within the chessboard coordinate system.

representing the complete scene. Moreover, it should be stressed that optimising the exterior orientation of the sensors is independent from the ICP process. Thus, bundle adjustment was applied for optimising the EO parameters of all Kinect sensors, whereas ICP was used to minimise the geometric error between pairs of point clouds. The relative rigid transformation derived from the EO parameters of the sensors was used as an initial guess matrix for the registration process. After refinement, the resulting matrix remained fixed during the complete sequence, preserving the relative transformation of the initial state of the scene.

Starting from sensor A (see Fig. 6.6), registration was performed following a counter-clockwise orientation. Given a pair of sensors, for example A and B, let $\mathbf{R}_{B,A}$ be the 3×3 rotation matrix expressing the relative rotation between the sensors and $\mathbf{T_{B,A}}$ the relative translation offset. The initial guess matrix $[\mathbf{R_{BA}} \mid \mathbf{T_{BA}}]$ was computed by the following relations:

$$\begin{aligned} \mathbf{R_{B,A}} &= \mathbf{R_A R_B^T} \\ \mathbf{T_{B,A}} &= -\mathbf{R_A R_B^T T_B} + \mathbf{T_A} \end{aligned} \tag{6.23}$$

Proof of the relations can be found in Appx. E.

The complete evaluation workflow, involving the orientation of the sensors (Sect. 6.5) and the sequential registration process proposed in the current section is outlined in Algo. 7. For the convergence of the ICP algorithm, two termination criteria were set: Maximum number of iterations imposed by the user (30) and the difference between the previous transformation and the current estimated transformation to be smaller than a predefined value (10^{-3}). Figure 6.7 shows the minimisation of the fitness score for every pair of clouds in relation to its number of iterations. It is apparent from the figure that the initial fitness scores for pairs (A,B) and (C,D) are much lower than the (CD,AB) pair. The reason for this is because the overlapping region between the first two pairs is much larger than the last pair, which is critical for performing a good registration. Furthermore, as the number of iterations increases, the amount of correction is significantly reduced. For pairs (A,B) and (C,D) the fitness scores are improved but with slower rates compare to the

Algorithm 7 Proposed approach for sequentially aligning a set of four point clouds.

```
Require: A set of GCPs denoted as $\mathbf{X_{GCP}}$ and their corresponding set of 2D projected
    points $\{\mathbf{x_A}, \ldots, \mathbf{x_D}\}$ acquired from four Kinect sensors $(\mathrm{A}, \ldots, \mathrm{D})$.
1: Compute the EO parameters for every Kinect sensor with respect to the chessboard
   world coordinate system using the PnP-LM or EPnP algorithms. Spatial resection is
   performed given a set of 2D↔3D correspondences, in this case defined by:
       $\{\{\mathbf{x_A} \leftrightarrow \mathbf{X_{GCP}}\}, \ldots, \{\mathbf{x_D} \leftrightarrow \mathbf{X_{GCP}}\}\}$
   The output from the aforementioned pose algorithms is a set of rotation matrices and
   translation vectors provided in the following way:
       $\{[\mathbf{R_A} \mid \mathbf{T_A}], \ldots, [\mathbf{R_D} \mid \mathbf{T_D}]\}$
2: Carry out a bundle adjustment using the EO values from Step 1 as initial values for
   the unknowns (IO parameters are considered fixed).
3: Compute the relative rigid transformation matrices between pairs of point clouds given
   the following order:
       $\mathbf{R_{A,B}} = \mathbf{R_A}\mathbf{R_B^T}$        $\mathbf{T_{A,B}} = -\mathbf{R_A}\mathbf{R_B^T}\mathbf{T_B} + \mathbf{T_A}$
       $\mathbf{R_{C,D}} = \mathbf{R_C}\mathbf{R_D^T}$        $\mathbf{T_{C,D}} = -\mathbf{R_C}\mathbf{R_D^T}\mathbf{T_D} + \mathbf{T_C}$
       $\mathbf{R_{CD,AB}} = \mathbf{R_{CD}}\mathbf{R_{AB}^T}$    $\mathbf{T_{CD,AB}} = -\mathbf{R_{CD}}\mathbf{R_{AB}^T}\mathbf{T_{AB}} + \mathbf{T_{CD}}$
4: Use the relative transformations from step 3 as initial guess matrices for performing
   the registration task through the ICP algorithm.
return Optimised relative transformation matrix for every pair of point clouds.
```

(CD,AB) pair, which rapidly converges after a few iterations. This is due to the minimum overlapping area between these clouds caused by the diametrically opposed configuration of the corresponding sensors. Furthermore, results showed that the ICP algorithm is able to register two clouds with significant overlapping without getting effected by the error of depth measurement.

Figure 6.12 illustrates the best case of the problem, where the noisy part for the registration is restricted towards the end of the scene. A severe case of the problem is depicted in Fig. 6.13. On can observe that the error introduced by the increasing distance to the sensor does not share similar geometries between the two point clouds, causing the registration algorithm to fail. Therefore, this leads to the conclusion that for diametrically opposed point clouds the registration should not be based on similar global structures in the scene but rather on local structures such as normals and curvature.

To this end, the normal-based ICP approach of Holz et al. [107] was used for registering a a pair of point clouds by comparing their normals. Results showed that even though the registration process didn't converge, there were small but stable movements compare to the normal ICP algorithm, which means that surface normals are more stable for comparing local appearances with respect to global scene structures. Nevertheless, due to the sensor dependent random depth measurement error, the surface normals of the ground plane may

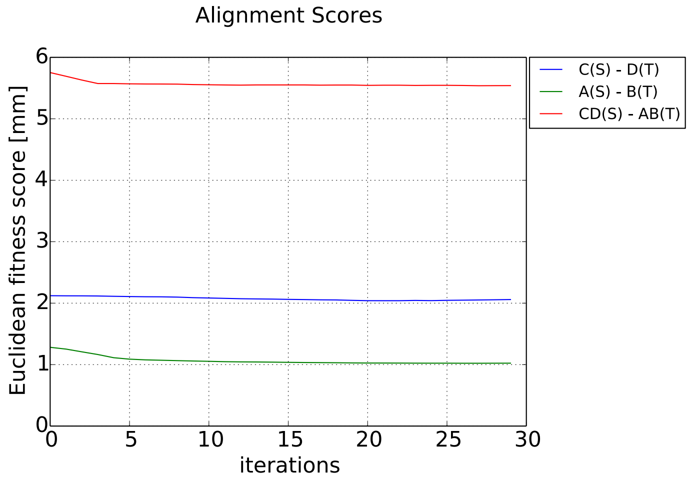


Figure 6.7: Fitness scores for all pairs of point clouds. It is clear that the initial fitness scores for pairs (A,B) and (C,D) are much lower than the (CD,AB) pair. The reason for this is because the overlapping region between the first two pairs is much larger than the last pair, which is critical for performing a good registration. Furthermore, as the number of iterations increases, the amount of correction is significantly reduced. For pairs (A,B) and (C,D) the fitness scores are improved but with slower rates compare to the (CD,AB) pair, which rapidly converges after a few iterations (≈ 3). (**Best viewed in colour**)

vary from sensor to sensor, which also affects the registration process.

6.7 Conclusions

In this chapter, a three-step evaluation workflow was presented for assessing the reliability of registering point clouds generated from multiple Kinect sensors. In the first part of the workflow, all point clouds were transformed to a world coordinate system defined by a regular chessboard object. The EO orientation of the sensors was found using the P*n*P-LM and EP*n*P algorithms. Both approaches provided very accurate EO parameters in the range of half and a quarter of a pixel respectively.

In the second step, the results from the P*n*P algorithms were used as initial values to a bundle adjustment system for optimising the EO of the sensors. Convergence in this step was achieved after a very few iterations, proving the robustness of the pose estimators. Thus, it should be stressed that if a chessboard patter is used as a world coordinate reference system, the bundle adjustment step could be omitted iif the sensors are oriented using one of the tested P*n*P algorithms. The reason for this is because the amount of correction in the EO parameters is in the range of millimetres and therefore does not provide a significant improvement of the overall accuracy of the system.

In the last step of the proposed workflow, all point clouds of the same time stamp were sequentially registered in a counter-clockwise manner using the ICP algorithm. The relative orientation for a pair of point clouds was replaced by the relative orientation between the corresponding sensors. Results showed that for point clouds looking in the same direction in the scene and have a large amount of overlapping, the registration is accurately performed where is for diametrically opposed point clouds the registration

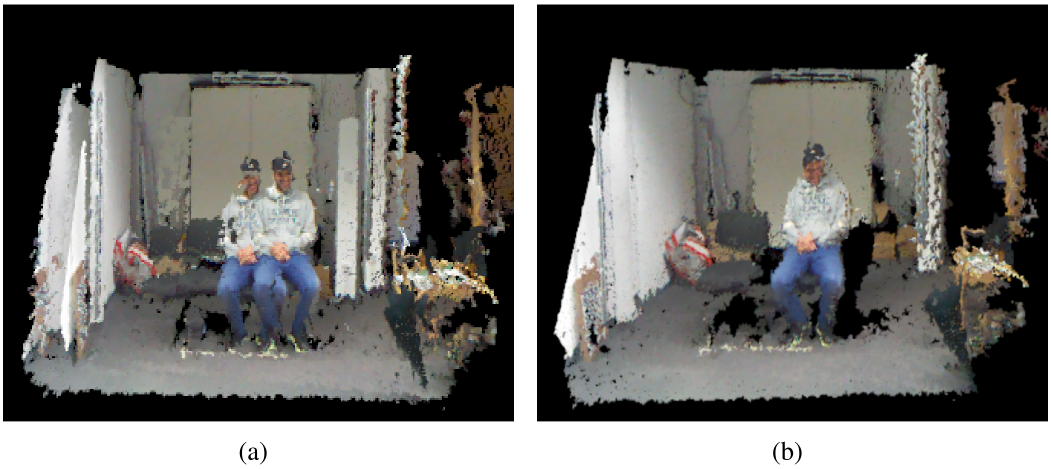


Figure 6.8: Registration of a pair of point clouds capturing a person sitting on a chair. Before registration (a) and after registration (b). (**Best viewed in colour**)

would fail. Therefore, this form of configuration should provide additional hints to the ICP algorithm, in the form of comparing similar local features between the two scenes.

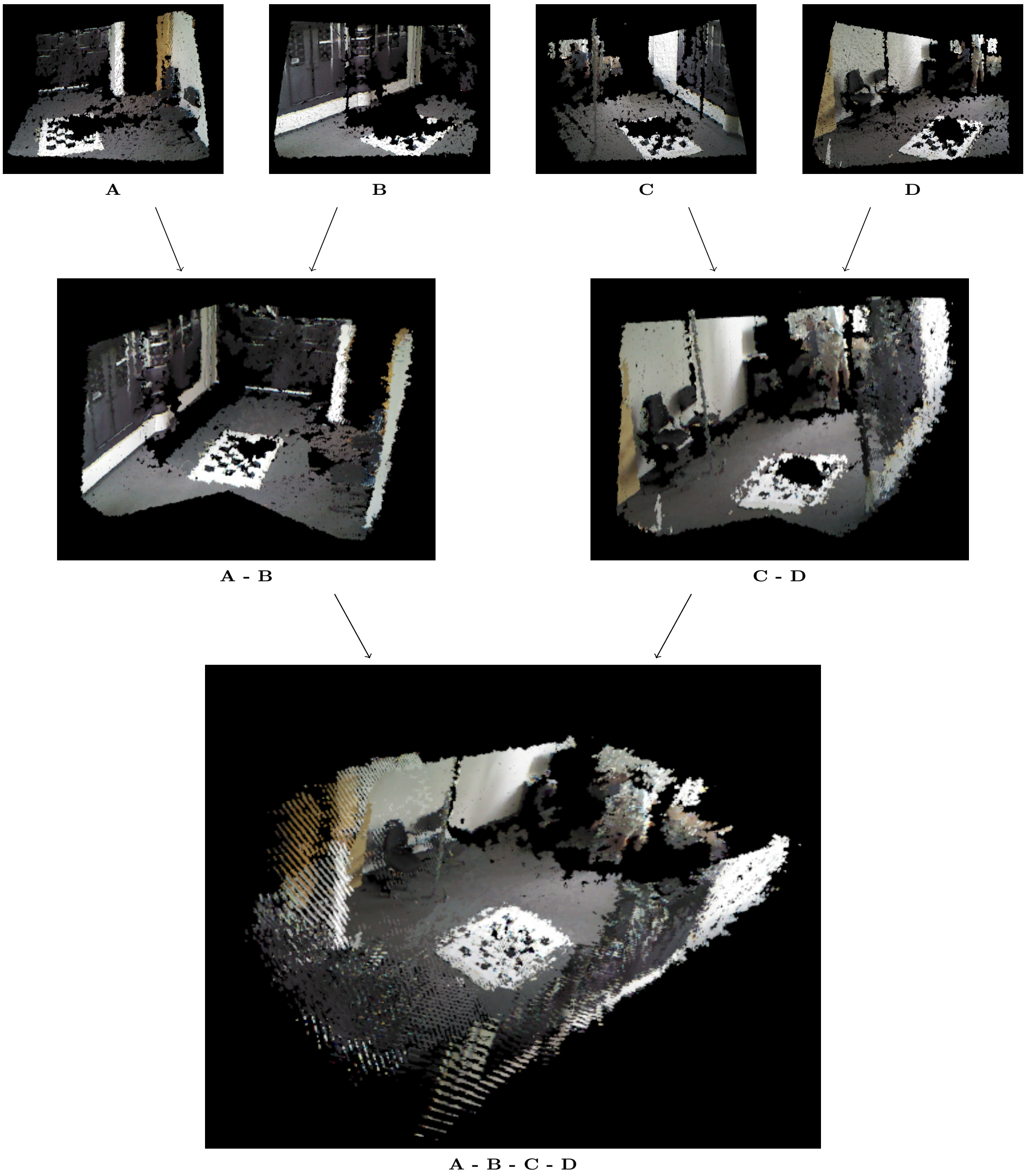
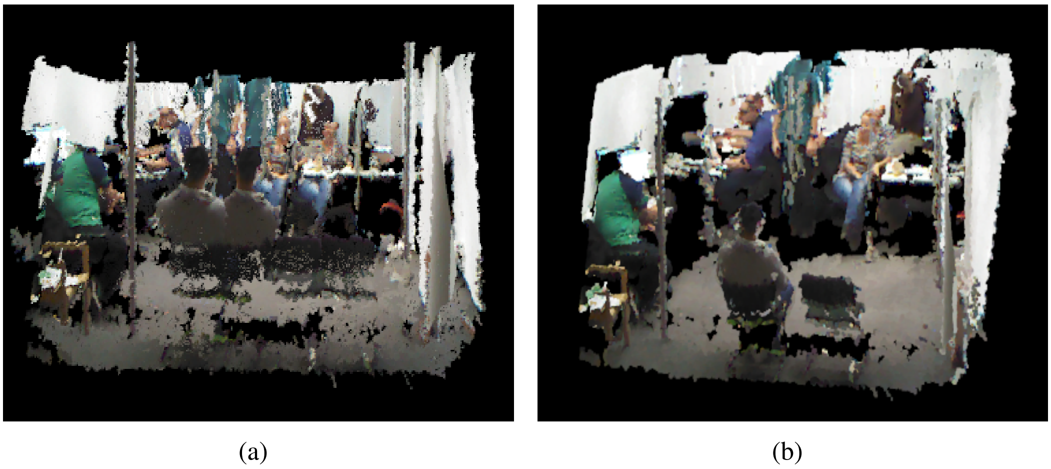
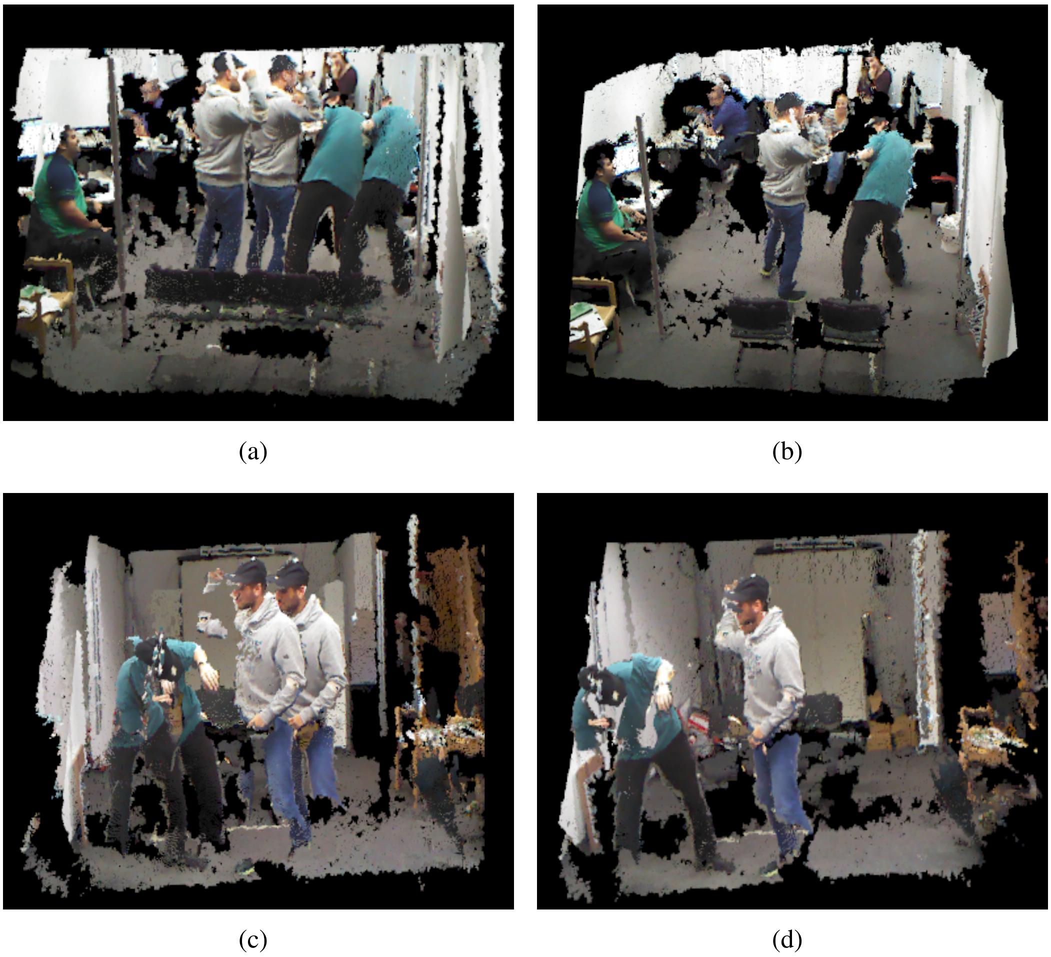


Figure 6.9: Sequential registration of point clouds. The registration procedure follows the pipeline introduced by Algo.7. (**Best viewed in colour**)


(a)

(b)

Figure 6.10: Registration of a pair of point clouds capturing people standing and sitting. Before registration (a) and after registration (b). (**Best viewed in colour**)


(a)

(b)

(c)

(d)

Figure 6.11: Registration of a pair of point clouds capturing people fighting. Left column: Before registration; Right column: After registration. (**Best viewed in colour**)

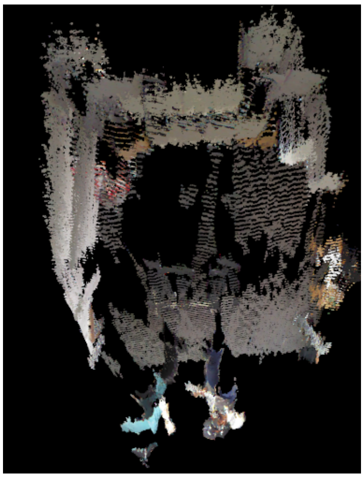
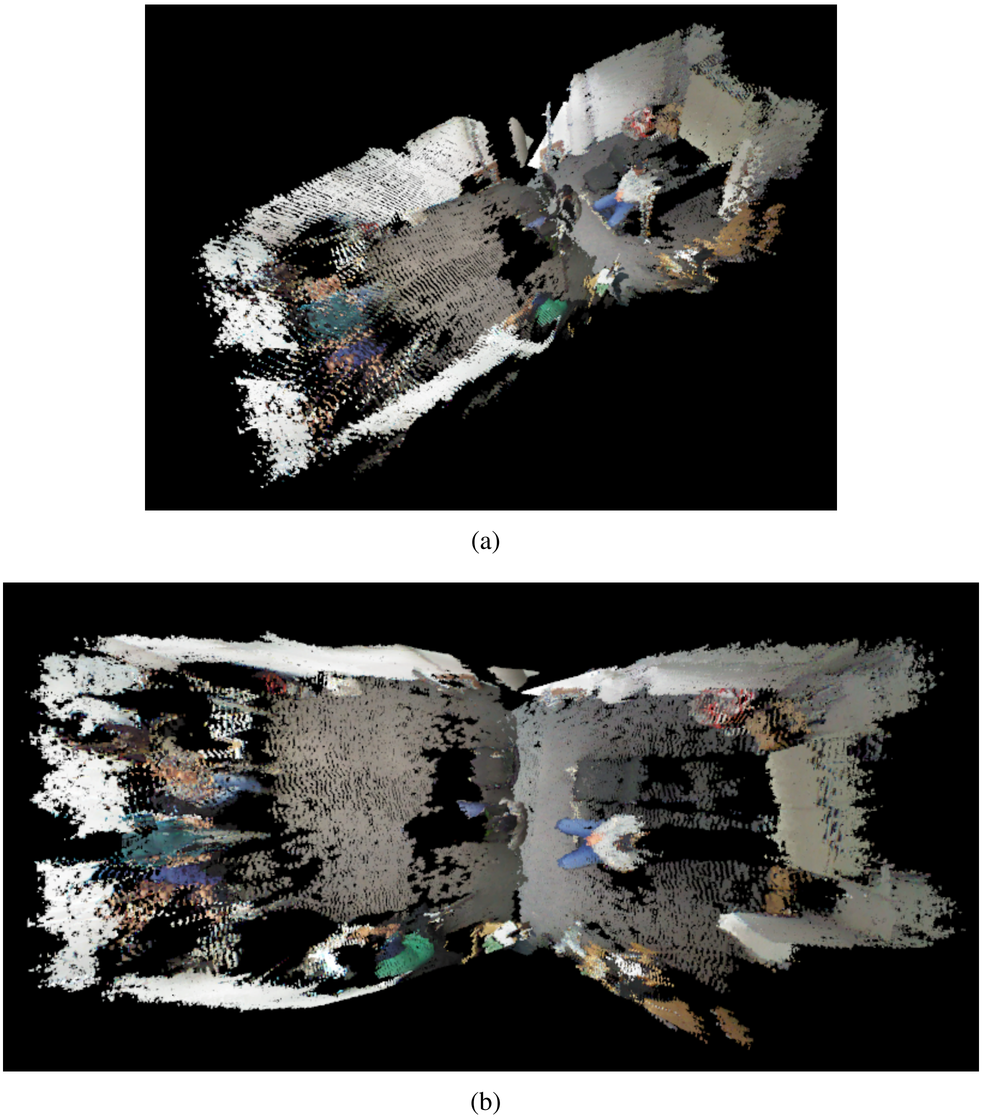


Figure 6.12: Registration problem (Best case).(**Best viewed in colour**)



(a)

(b)

Figure 6.13: Registration problem (Difficult case). Difficult case of registration (perspective view (a) and top view (b)). (**Best viewed in colour**)

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The human brain performs object recognition effortlessly and instantaneously based on visually collected data. It is a result of a life learning process, where visual information is constantly updated in the inferior temporal cortex of the brain. This enables the rapid recognition of objects despite their substantial appearance variations. Hence, the task of object recognition is not just the identification of an object at a particular moment in time but also tracing the way it evolves in time. The human eyes work stereoscopically, which means that the data processed by the visual human system is a real time reconstruction of the scene. Therefore, the purpose of the current dissertation was to develop algorithms that can recognise objects in three-dimensional space and capture their temporal shape variations by processing data similar to the ones obtained by the human visual system. Such data were acquired by a Kinect sensor, a structure light technology that provides real time RGB and depth information of the scene.

The results of my work can be concluded as follows: For the first part, the accuracy of the segmentation depends strongly on the accuracy of the detection box. After evaluating the performance of both detectors, the precision and recall for Dalal and Triggs algorithm was higher than Dubout's algorithm. However, Dubout's approach had a higher IOU accuracy compare to Dalal and Triggs, which means that DPM approaches provide a better localisation and approximate size determination of a person in a scene. Furthermore, in the segmentation part results showed that using different edge potentials does not significantly affect the segmentation accuracy. This means that the unary potentials have a larger influence for the graph cut algorithm compare to the edge potentials. Comparing the segmentation approach to the CRF-CNN approach (Zheng et al. [92]) showed improved results mostly for extreme poses. It is believed that using a larger amount of training data with a larger variety of poses could improve the weights of the classifier and thus achieve better segmentation results.

Articulated objects experience time-dependent shape variations, which are also recognised by the human visual perception system. In the second part of the dissertation, a system was proposed for capturing and tracking these variations in three-dimensional space by using the information deduced by the minimum volume enclosing ellipsoid (MVEE) algorithm introduced by Moshtagh [13]. Results have shown that the reliability of the ellipsoid information depends mainly on the quality of the extracted foreground mask of the person.

In the last part, an workflow was established for evaluating the accuracy of merging point cloud data from multiple Kinect sensors. In the first part the results showed that the P*n*P-LM and EP*n*P approaches are well suited for initially orienting all sensors with respect to a global coordinate system. This was proven in the second part of the workflow, where a bundle adjustment was performed for optimising the exterior orientation of the sensors using as initial values the results provided in the previous step. Convergence was reached after a few iterations, proving the robustness of the P*n*P approaches. However, the accuracy of the orientation of the sensors is independent from the quality of the generated point clouds and thus, a geometric correction was performed using the Iterative Closest Point (ICP) algorithm. For sensors that capture approximately the same area in the scene, registration would converge after few iterations whereas for diametrical opposed views the registration process would make very small corrections due to the limited overlapping information. Applying the surface-normal ICP approach didn't improve the registration result, which means that the orientation of the surface normals is strongly affected by the quality of the point cloud. I believe that the results presented in Chap. 6 could be useful for future object recognition approaches, where accurate combination of 3D dataset will be required.

7.2 Future Work

The work presented in the current dissertation could be extended as follows:

In Chap. 4, a Conditional Random Field pairwise energy function was presented, for the task of segmenting human instances in RGBD space. One direction of future work could be to replace graph cuts (Boykov and Kolmogorov [68]) with dynamic graph cuts (Kohli et al. [108]) for reducing the computation time of the cut. Although graph cuts are well known for their low polynomial time complexity, dynamic graph cuts require time which is proportional to the total amount of change in the edge weights between two graph instances. Considering only sub-modular energy functions, MAP inference can be found in less time, significantly improving the performance of the complete recognition task.
Another direction of future work could be to extend the proposed energy function 4.3 by incorporating a higher order potential term $\psi_c(\boldsymbol{x_c})$, which adds an additional constraint that all pixels constituting a segment should be part of the same object. Spatial consistency for pixels corresponding to an object can be encouraged by giving a high score to pixels taking a different label than the correct one. Absence of partial inconsistency will lead to a hard penalisation assuming that all or none of the pixels should take the correct label. To prevent this, one could use the Robust P^n Potts model (see Kohli et al. [109]), which gives a cost that is modelled by a linear truncated function and depends on the number of inconsistent pixels. Partial inconsistency also depends on the number of pixels disagreeing with the dominant label. This requires an a-priori knowledge of the object's shape. Ladický et al. [14] employed a local colour model using the pixel information within the detection box to distinguish between foreground-background pixels. While colour is not a discriminative feature, foreground/background pixels sharing similar colours could lead to undependable results. This could significantly be improved by utilising the depth information from the Kinect sensor and clustering voxels with similar depth. A simple 3D connected-component could be used for this purpose.

Higher order potentials can efficiently be solved using *move making* algorithms such as the α-expansion and $\alpha\beta$ swapping. While these methods have shown impressive results for multi-label image classification tasks (see Kohli et al. [109], [61]), their computation performance depends highly on the size of the label set. For a binary label set, a higher order submodular energy function could efficiently be solved using swapping or expansion move algorithm in lower computation time.

Apart from the segmentation task, the quality of the detection box could be improved by learning human representations jointly from RGB and depth information such as the Combo-HOD (Spinello et al. [110]) or the depth-based sub-clustering approach for detecting people in groups introduced by Munaro et al. [111].

A third line of research which arises from Chap. 5 is to develop an approach that can "learn" the relation between the parameters of the ellipsoid *i.e.* the variations in the angles or the size of the ellipsoid indication of a predefined motion. Furthermore, the quality of the ellipsoid is highly depended on the accuracy of the foreground. Thus, although the proposed pipeline in Algo. 2 removes all noisy blobs resulted from Kammerl's [12] foreground estimation approach, one could seek improving the existing method in terms of computational performance.

Finally, the most direct extension of the proposed work in Chap. 6 should focus on developing approaches that can solve the error introduced by the Kinect sensor as a function of the distance between the sensor itself and the object/-s placed in the center of scene. This could be improved or eliminated by learning a depth multiplier image (Teichman et al. [112]). Furthermore, the registration accuracy between the point clouds could be further evaluated by combining different geometrical information such as comparison between similar normals, or feature points.

Appendix A

Submodular Energy Functions

Submodular energy functions have been extensively used in the computer vision field, especially for image segmentation tasks (Kohli et al. [61], [109], [108]) as they can be minimised in polynomial time (Kolmogorov and Zabih [67]). The definition presented here follows the notation introduced in Chap. 3, Sects. 3.2 and 3.3 respectively.

If every random variable $(X_i)_{i \in \mathcal{V}}$ is assigned a value x_i from its configuration space $\mathcal{X}_i$ $(x_i \in \mathcal{X}_i)$, a pairwise MRF energy function is said to be *submodular*, if its pairwise term $\phi(x_i, x_j), \forall\,(i,j) \in \mathcal{E}$ satisfies:

$$\begin{aligned} \forall\, x_i^1, x_i^2 \in \mathcal{X}_i, \quad \text{s.t.} \quad x_i^1 \leq x_i^2 \\ \forall\, x_j^1, x_j^2 \in \mathcal{X}_j, \quad \text{s.t.} \quad x_j^1 \leq x_j^2 \end{aligned} \tag{A.1}$$

Based on A.1, the following condition should hold:

$$\phi_{ij}(x_i^1, x_j^1) + \phi_{ij}(x_i^2, x_j^2) \leq \phi_{ij}(x_i^1, x_j^2) + \phi_{ij}(x_i^2, x_j^1) \tag{A.2}$$

In case of binary energy functions, where $X_i \in \{0,1\}, \forall i \in \mathcal{V}$, condition A.2 is given by:

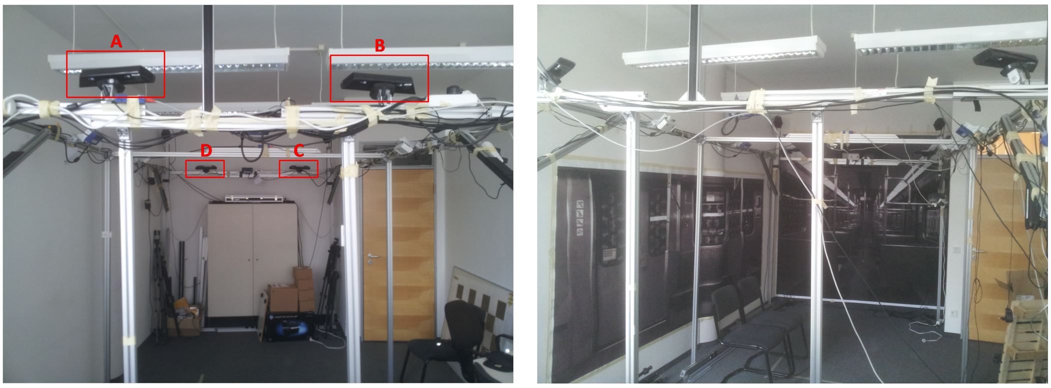
$$\phi_{ij}(0,0) + \phi_{ij}(1,1) \leq \phi_{ij}(0,1) + \phi_{ij}(1,0) \tag{A.3}$$

Potential functions with one binary variable are always submodular.

Appendix B

Environment Setup

The proposed algorithms in the dissertation were tested and evaluated in a simulated indoor environment (see Fig. B.1) that emulates the internal part of a train wagon. In order to record the complete FOV of the scene, four Kinect sensors were mounted on an aluminium construction as depicted in the images below. Every Kinect sensor was assigned a unique ID, which was used throughout the dissertation. Depending on the lighting conditions and texturing of the scene, point clouds of varying qualities were generated. Within this context, the word quality refers to the density of the point cloud, which depends highly on the texturing, lighting and structure of the scene.



(a)

(b)

Figure B.1: Multi sensor environment setup. Sensors setup emulating the internal part of a train wagon. Every Kinect sensor was assigned a unique ID for consistency. From left to right: The environment shown in Fig. B.1(a) was dynamically reconfigured, allowing different texturing and lighting conditions in the scene (Fig. B.1(b)). This was important for testing and evaluating the proposed algorithms on point clouds generated from different conditions in the scene.

The aluminium construction covers an area of ≈ 4.5 m $\times$ 2.2 m $\times$ 2.3 m depth, width and height respectively. Different scenarios were recorded in parallel from all sensors with an acquisition rate of ≈ 19 FPS. One of the main drawbacks of using multiple speckle-based structured light sensors, is the drastic reduction in the quality of the depth images due to the interference of the near-infrared light of different sensors. To eliminate this effect, all sensors where oriented towards the lower part of the scene, restricting the amount of interference on the ground area of the FOV. Synchronisation was performed in a multithreaded fashion (using the Boost library [89]), overcoming the problem caused by the hardware latency of the USB bus.

Appendix C

Kinect calibration

There is a variety of calibration tools that have been developed within several open source projects, such as ROS [113], MPRT [114] and OpenCV [106]. While working on 3D applications, the quality of the generated point cloud depends highly on the quality of the calibration parameters (also known as intrinsic or interior parameters (see Sect. 6.2.1)). Within the dissertation, a pinhole camera model is employed. The image distortion is assumed to be expressed by the Brown model [97], which contains a set of 10 parameters [1]. The manufacturer of Kinect provides a set of default calibration values that are well suited for large scale applications but not accurate enough for modelling local 3D geometry. One main advantage of the Kinect sensor is that it uses low distortion lenses, producing faintly apparent displacement errors around the corners of the image. This is also depicted by the radial distortion curves in Fig. C.2. Inadequate calibration generates a systematic error in the 3D coordinates of individual points. Also, misalignment between the IR and RGB coordinate systems may lead to a inaccurate assignment of the RGB values to the 3D points. As a consequence, these sources of errors may or may not be taken into account, depending on the application and its requirements.

In recent years, several studies have presented a list of factors influencing the geometric accuracy and quality of the depth data. To the best of my knowledge, Khoshelham [115] was the first to perform a theoretical error analysis on the depth data and the impact of the error in indoor mapping applications (Khoshelham et al. [116]). His results showed that the error of depth measurement increases quadratically with the increasing distance to the sensor, ranging from a few millimetres up to ≈ 4 cm at the maximum distance of the sensor (≈ 5 m). Although this is a sensor-dependent random error it was not considered in this work, due to the fact that achieving maximum accuracy of the generated point cloud is not a prerequisite for performing object recognition tasks.

To determine the interior parameters of the IR and RGB cameras, a standard calibration approach was employed. A chessboard pattern of 5 cm grid size and dimensions of 5×7 rows and columns respectively was used as a reference pattern for both camera systems. The inner corners of the grid served as GCPs and the corresponding image points were computed using a sub-pixel accuracy corner detector implemented in the OpenCV [106] library. Since the IR and RGB cameras are not able to acquire data simultaneously due to hardware restrictions, acquisition was performed by enabling/disabling the camera shatters every 500 ms. Due to the observation of the emitted speckles by the IR camera, the quality

[1] The Brown model consists of the following parameters: c, x_0, y_0, k_1, k_2, k_3, p_1, p_2, b_0, b_1.

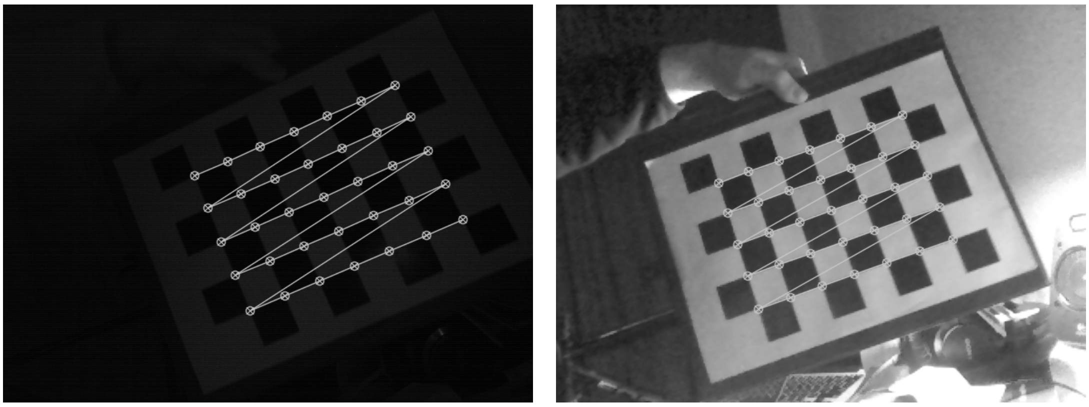


Figure C.1: Example of a chessboard pattern acquired during the calibration process from both the IR and RGB cameras of a Kinect sensor. Fig. C.1(a) shows the chessboard points detected from the IR camera and Fig. C.1(b) from the RGB camera respectively (The RGB image is shown in grey scale because the chessboard corner detector in OpenCV [106] only works on grey scale images. (**Best viewed in colour**)

of the chessboard corners was highly noisy. Thus, to avoid any disturbances caused by the encoded speckles in the IR camera, the infrared emitter was covered with opaque tape. The light from the emitter was replaced by an external halogen lamp (also known as tungsten halogen) for improving the visibility of the chessboard corners and helping to acquire approximately the same chessboard images from the IR and RGB cameras. This step was essential for finding the relative transformation between the camera poses. The parameters for every sensor are provided in Tab. C.4.

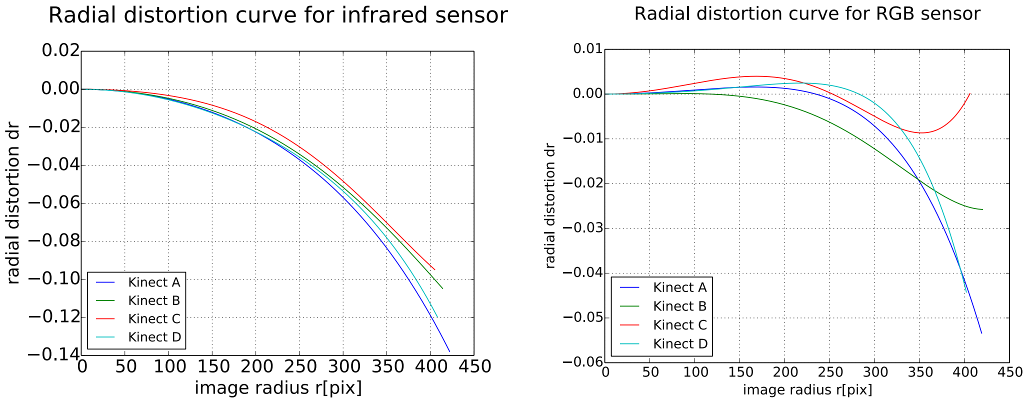
A total of 100 chessboard images in different poses were recorded from each sensor for performing the calibration task. Every pool was split in 10 different random sets of 24 images using a random selection algorithm. Each set was calibrated separately using the calibration algorithm proposed by Zhang[2] [118]. Tables C.2 and C.3 provide the interior parameters values for every Kinect sensor.

The RMS error for the IR and RGB cameras is given in Tab. C.1. It is well known from literature (Khoshelham [115], Zhang et al. [119], Herrera et al. [120]) that the IR camera uses lower distortion lenses compare to the RGB camera, leading to a smaller RMS error as it is evident from the results in Tab. C.1.

Approach	**Sensor ID**			
	A	**B**	**C**	**D**
Infrared	0.171	0.217	0.208	0.216
RGB	0.228	0.241	0.216	0.336

Table C.1: Average reprojection error (in pixels)

[2] An alternative approach could be the one of Herrera et. al [117]. The method is not based on depth discontinuities but uses a set of planar images recorded from various poses



(a)

(b)

Figure C.2: Radial distortion curves for the IR and RGB cameras of the Kinect sensors. The IR curves follow of a barrel-like distortion curve, while the RGB radial distortion curves follow also a more moustache type representation. (**Best viewed in colour**)

Sensor	**ID**	c_x	c_y	x_0	y_0
Infrared	A	569.833	569.117	335.038	259.227
	B	569.220	569.198	323.839	260.600
	C	586.786	585.971	327.520	240.808
	D	591.379	591.084	330.742	241.404
RGB	A	511.908	510.640	330.890	259.780
	B	497.488	497.568	326.354	267.325
	C	516.212	516.059	324.874	246.470
	D	521.307	520.709	323.213	242.130

Table C.2: IR and RGB intrinsic camera parameters (in pixel unit)

Sensor	**ID**	k_1	k_2	p_1	p_2	k_3
Infrared	A	-0.1648	0.7673	0.0069	0.0064	-1.3135
	B	-0.1480	0.6950	0.0079	0.0017	-1.2501
	C	-0.0992	0.2648	0.0022	-0.0036	-0.2833
	D	-0.1933	1.1570	0.0002	0.0039	-2.4079
RGB	A	0.0288	-0.1249	0.0070	0.0056	-0.0506
	B	0.0093	-0.1718	0.0124	0.0048	0.1517
	C	0.0803	-0.4595	0.0072	-0.0008	0.5304
	D	0.0187	0.0259	-0.0009	0.0030	-0.2990

Table C.3: IR and RGB lens distortion parameters.

Moreover, the mean square error for all sensors was 1/4 of a pixel, which is acceptable if one would consider the low distortion of the lenses. Figure C.2 shows the radial symmetric curves of the IR and RGB cameras expressed by the principal point. Specifically, for the

ID	T_x [m]	T_y [m]	T_z [m]	R_x [deg]	R_y [deg]	R_z [deg]
A	−2.160	0.010	0.780	0.937	0.280	−0.386
B	−2.680	0.410	−0.630	0.842	0.049	0.100
C	−1.840	−0.730	−0.090	0.536	−2.676	−0.282
D	−2.010	0.110	−0.190	−0.118	0.483	0.294

Table C.4: Translation and rotation parameters between the IR and RGB cameras coordinate systems. These parameters define the relative rigid transformation between the two camera poses.

IR cameras, it is clear that all sensors have the form of a barrel distortion curve, achieving a maximum displacement error of 0.14 pixels in the extreme regions of the image. For the RGB cameras, the radial distortion error is considered negligible, even though the form of the curve does not strictly follow a barrel distortion form but rather a moustache type representation.

Appendix D

XML Structure of the Ellipsoid

According to Sect. 5.2.5, an ellipsoid can be mathematically represented by its variance-covariance matrix. After applying principal component analysis (PCA) on the matrix, the extracted information is stored in an XML file for performing post-processing tasks. The structure and description of the elements of the XML file are given in Fig. D.1 and Tab. D.1 respectively. If no ellipsoid is found in the current scene, the corresponding XML elements are set to zero.

The XML file was given as an input to a tracking software for visualising the variations of the ellipsoid in time. The results from the software was provided to psychologists in the anthropology and disaster management field for classifying the behaviour of the people according to their shape deformations.

Element Name	**Description**
SemiMajorAxis	The length of the semi-major axis a, b and c of the ellipsoid
Center	Position of the center of the ellipsoid
UpperPoint	Upper vertex position
LowerPoint	Lower vertex position
FrontPoint	Front vertex position
Volume	Volume of the ellipsoid expressed in m^3
Rotations	Angles defined between every semi-major axis and their corresponding predefined reference axis of the fictitious coordinate system.
HumanInfo	Approximated values for the height, width and depth of a person

Table D.1: A description of the XML elements containing the ellipsoid information.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Person ID="">
  <EllipsoidInfo>
    <SemiMajorAxis>
      <a></a>
      <b></b>
      <c></c>
    </SemiMajorAxis>
    <Center>
      <X></X>
      <Y></Y>
      <Z></Z>
    </Center>
    <UpperPoint>
      <X></X>
      <Y></Y>
      <Z></Z>
    </UpperPoint>
    <LowerPoint>
      <X></X>
      <Y></Y>
      <Z></Z>
    </LowerPoint>
    <FrontPoint>
      <X></X>
      <Y></Y>
      <Z></Z>
    </FrontPoint>
    <Volume></Volume>
    <Rotations>
      <Omega></Omega>
      <Phi></Phi>
      <Kappa></Kappa>
    </Rotations>
  </EllipsoidInfo>
  <HumanInfo>
    <Height></Height>
    <Width></Width>
    <Depth></Depth>
  </HumanInfo>
</Person>
```

Figure D.1: XML structure containing the ellipsoid data. The XML tree starts at a parent node (also known as root element) corresponding to the person's ID and branches into several child elements representing the attributes of the ellipsoid. The numerical values of the ellipsoid were registered within the sub-elements of the corresponding child elements.

Appendix E

Relative 3D Transformation

The Iterative Closest Point (ICP) algorithm is one the most known approaches for the geometric alignment of a pair of point clouds. However, this alignment procedure requires an initial estimate of the relative pose between the point clouds, expressed by a rigid transformation (rotation + translation) matrix. The purpose of this chapter is to derive the mathematical relation that expresses the relative orientation between the source (s) and target $\mathbf{t}$ dataset, utilising the information from the sensor poses.

Specifically, let $[\mathbf{R_{rel}} \mid \mathbf{T_{rel}}]$ correspond to the relative transformation matrix representing the relative rotation $\mathbf{R_{rel}}$ and relative translation $\mathbf{T_{rel}}$ of the source to the target point cloud. Given the exterior orientation of the Kinect sensors, computed by the Perspective-*n*-Point algorithms (see Sect. 6.5), one can approximate $[\mathbf{R_{rel}} \mid \mathbf{T_{rel}}]$, by finding the relative transformation matrix between the exterior orientations of the sensors. Even though the accuracy of the sensors poses differs from the accuracy of the point clouds, the relative transformation between the two sensor systems could be considered as a good approximation for the true initial guess.

Setting the upper left corner of a chessboard as the origin of a world coordinate system (see Fig. 6.5), let $[\mathbf{R_s} \mid \mathbf{T_s}]$ and $[\mathbf{R_t} \mid \mathbf{T_t}]$ represent the relative transformation matrices of the left Kinect sensor ($\mathbf{t}$ cloud) and the right sensor (s cloud) with respect to that system. Every spatial point $\mathbf{X_W}$ corresponding to a horizontal GCP on the chessboard surface can be transformed to both coordinate systems from the following relations:

$$\mathbf{X_t} = \mathbf{R_t}\mathbf{X_W} + \mathbf{T_t} \tag{E.1a}$$

$$\mathbf{X_s} = \mathbf{R_s}\mathbf{X_W} + \mathbf{T_s} \tag{E.1b}$$

Solving E.1b for $\mathbf{X_W}$, it becomes:

$$\mathbf{X_W} = \mathbf{R_s^T}\mathbf{X_s} - \mathbf{R_s^T}\mathbf{T_s} \tag{E.2}$$

Inserting E.2 in E.1a:

$$\begin{aligned} \mathbf{X_t} &= \mathbf{R_t}(\mathbf{R_s^T}\mathbf{X_s} - \mathbf{R_s^T}\mathbf{T_s}) + \mathbf{T_t} \\ &= \mathbf{R_t}\mathbf{R_s^T}\mathbf{X_s} - \mathbf{R_t}\mathbf{R_s^T}\mathbf{T_s} + \mathbf{T_t} \end{aligned} \tag{E.3}$$

where,

$$\mathbf{R_{rel}} = \mathbf{R_t R_s^T} \tag{E.4a}$$

$$\mathbf{T_{rel}} = -\mathbf{R_t R_s^T T_s} + \mathbf{T_t} \tag{E.4b}$$

Bibliography

[1] A. Andreopoulos and J. K. Tsotsos, "50 years of object recognition: Directions forward." *Computer Vision and Image Understanding*, vol. 117, no. 8, pp. 827–891, 2013.

[2] L. G. Roberts, *Machine Perception of Three-Dimensional Solids*, ser. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York, 1963.

[3] D. G. Lowe and T. O. Binford, "Perceptual organization as a basis for visual recognition." in *Association for the Advancement of Artificial Intelligence (AAAI)*, 1983, pp. 255–260.

[4] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychological Review*, vol. 94, pp. 115–147, 1987.

[5] C. Hough and V. Paul, "Method and means for recognizing complex patterns," December 1962, uS Patent 3,069,654.

[6] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, January 1972.

[7] A. Desolneux, L. Moisan, and J.-M. Morel, *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, 1st ed. Springer Publishing Company, Incorporated, 2007.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TRAMI)*, vol. 32, no. 9, pp. 1627–1645, 2010.

[10] C. Dubout and F. Fleuret, "Deformable part models with individual part scaling," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2013, Poster, pp. 28.1–28.10.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *ASME Journal of Basic Engineering*, 1960.

[12] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, "Real-time compression of point cloud streams," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, Conference Proceedings, pp. 778–785.

[13] N. Moshtagh, "Minimum volume enclosing ellipsoid," University of Pennsylvania, Tech. Rep., 2005.

[14] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. S. Torr, "What, where and how many? combining object detectors and crfs." in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, vol. 6314. Springer, 2010, pp. 424–437.

[15] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the 18th International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289.

[16] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 23, no. 11, pp. 1222–1239, November 2001.

[17] L. L. Vibhav Vineet, Jonathan Warrell and P. Torr, "Human instance segmentation from video using detector-based conditional random fields," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2011, pp. 80.1–80.11.

[18] G. Shu, A. Dehghan, and M. Shah, "Improving an object detector and extracting regions using superpixels," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 3721–3727.

[19] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[20] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[21] K. Lai, L. Bo, X. Ren, and D. Fox, "Detection-based object labeling in 3d scenes," in *IEEE International Conference on on Robotics and Automation*, 2012.

[22] A. Teichman, J. T. Lussier, and S. Thrun, "Learning to segment and track in rgbd," *IEEE Transactions on Automation Science and Engineering*, pp. 841–852, 2013.

[23] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[24] L. Chen, H. Wei, and J. Ferryman, "A survey of human motion analysis using depth imagery," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 1995–2006, November 2013.

[25] J. Aggarwal and M. Ryoo, “Human activity analysis: A review,” *ACM Computing Surveys*, vol. 43, no. 3, pp. 16:1–16:43, April 2011.

[26] W. Li, Z. Zhang, and Z. Liu, “Action recognition based on a bag of 3d points,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 9–14, 2010.

[27] O. Oreifej and Z. Liu, “Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 716–723, 2013.

[28] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, and M. F. M. Campos, “STOP: space-time occupancy patterns for 3d action recognition from depth map sequences,” in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - 17th Iberoamerican Congress, CIARP 2012, Buenos Aires, Argentina*, 2012, pp. 252–259.

[29] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, “Robust 3d action recognition with random occupancy patterns,” in *Proceedings of the 12th European Conference on Computer Vision (CVPR) - Volume Part II*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 872–885.

[30] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, “Efficient human pose estimation from single depth images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2821–2840, 2013.

[31] U. Rafi, J. Gall, and B. Leibe, “A semantic occlusion model for human pose estimation from a single depth image,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2015, pp. 67–74.

[32] K. Buys, C. Cagniart, A. Baksheev, T. De Laet, J. De Schutter, and C. Pantofaru, “An adaptable system for rgb-d based human body detection and pose estimation,” *The Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 39–52, 2014.

[33] F. Hegger, N. Hochgeschwender, G. Kraetzschmar, and P. Ploeger, *People Detection in 3d Point Clouds Using Local Surface Normals*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 7500, book section 15, pp. 154–165.

[34] M. Sigalas, M. Pateraki, I. Oikonomidis, and P. Trahanias, “Robust model-based 3d torso pose estimation in rgb-d sequences,” in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, June 2013.

[35] J. Ziegler, K. Nickel, and R. Stiefelhagen, “Tracking of the articulated upper body on multi-view stereo image sequences.” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2006, pp. 774–781.

[36] M. Baum and U. D. Hanebeck, "Extended object tracking with random hypersurface models," *Computing Research Repository (CoRR)*, 2013.

[37] Y. Schröder, A. Scholz, K. Berger, K. Ruhl, S. Guthe, and M. Magnor, "Multiple kinect studies," Computer Graphics Lab, Technische Universität Darmstadt, Tech. Rep. 09-15, October 2011.

[38] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3d full human bodies using kinects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 4, pp. 643–650, April 2012.

[39] F. Faion, M. Baum, and U. D. Hanebeck, "Tracking 3d shapes in noisy point clouds with random hypersurface models," in *15th International Conference on Information Fusion, FUSION 2012*, Singapore, July 2012, pp. 2230–2235.

[40] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in *Proceedings of the IEEE*, 2004, pp. 401–422.

[41] E. Almazán and G. Jones, "Tracking people across multiple non-overlapping rgb-d sensors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2013, pp. 831–837.

[42] D. Michel, C. Panagiotakis, and A. A. Argyros, "Tracking the articulated motion of the human body with two rgbd cameras," *Machine Vision and Applications*, vol. 26, no. 1, pp. 41–54, January 2015.

[43] J. M. Hammersley and P. E. Clifford, "Markov random fields on finite graphs and lattices," Unpublished manuscript, 1971.

[44] C. Wang, N. Komodakis, and N. Paragios, "Markov random field modeling, inference & learning in computer vision & image understanding: A survey," *Computer Vision Image Understanding*, vol. 117, no. 11, pp. 1610–1627, November 2013.

[45] Z. Kato and T. Pong, "A multi-layer MRF model for video object segmentation," in *7th Asian Conference on Computer Vision (ACCV)*, Hyderabad, India, January 2006, pp. 953–962.

[46] D. Shi and Y. Han, "An algorithm for moving objects segmentation based on mrf," in *4th International Congress on Image and Signal Processing (ICISP)*, vol. 3, October 2011, pp. 1396–1399.

[47] M. Björkman, N. Bergström, and D. Kragic, "Detecting, segmenting and tracking unknown objects using multi-label MRF inference," *Computer Vision and Image Understanding (CVIU)*, vol. 118, pp. 111–127, 2014.

[48] Z. Yin and R. T. Collins, "Belief propagation in a 3D spatio-temporal MRF for moving object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2007.

[49] C. S. Lea and J. J. Corso, "Efficient hierarchical markov random fields for object detection on a mobile robot," *Computing Research Repository (CoRR)*, 2011.

[50] A. Ghosh, A. Mondal, and S. Ghosh, “Moving object detection using markov random field and distributed differential evolution,” *Applied Soft Computing*, vol. 15, pp. 121–136, 2014.

[51] S. M. Choi, J. E. Lee, J. Kim, and M. H. Kim, “Volumetric object reconstruction using the 3d-mrf model-based segmentation [magnetic resonance imaging],” *IEEE Transactions on Medical Imaging*, vol. 16, no. 6, pp. 887–892, December 1997.

[52] P. Li, R. K. Gunnewiek, and P. H. N. de With, “Scene reconstruction using MRF optimization with image content adaptive energy functions,” in *10th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, Juan-les-Pins, France, October 2008, pp. 872–882.

[53] Y. Pan, M. Zhou, Y. Fan, D. Zhang, and X. Zheng, “A weighted color mrf model for 3d reconstruction from a single image,” in *International Conference on Virtual Reality and Visualization (ICVRV)*, September 2013, pp. 21–28.

[54] J. Sun, N.-N. Zheng, and H.-Y. Shum, “Stereo matching using belief propagation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 25, no. 7, pp. 787–800, July 2003.

[55] L. Zhang and S. M. Seitz, “Parameter estimation for mrf stereo,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, June 2005, pp. 288–295.

[56] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun, “Continuous markov random fields for robust stereo estimation,” in *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, vol. 5. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 45–58.

[57] Y. Boykov and G. Funka-Lea, “Graph cuts and efficient n-d image segmentation,” *International Journal of Computer Vision (IJCV)*, vol. 70, no. 2, pp. 109–131, November 2006.

[58] C. Rother, V. Kolmogorov, and A. Blake, “"grabcut": Interactive foreground extraction using iterated graph cuts,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[59] R. B. Potts, “Some generalized order-disorder transformations,” *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 48, pp. 106–109, January 1952.

[60] E. Ising, “Beitrag zur theorie des ferromagnetismus,” *Zeitschrift für Physik*, vol. 31, no. 1, pp. 253–258, 1925.

[61] P. Kohli, L. Ladický, and P. H. Torr, “Robust higher order potentials for enforcing label consistency,” *International Journal of Computer Vision (IJCV)*, vol. 82, no. 3, pp. 302–324, May 2009.

[62] D. Yan and M. Yongzhuang, “Image restoration using graph cuts with adaptive smoothing,” in *International Conference on Information Acquisition (ICIA)*, July 2007, pp. 152–156.

[63] V. Kolmogorov and R. Zabih, “Multi-camera scene reconstruction via graph cuts,” in *Proceedings of the 7th European Conference on Computer Vision (ECCV)*. London, UK: Springer-Verlag, 2002, pp. 82–96.

[64] D. Wang and K. B. Lim, “Obtaining depth map from segment-based stereo matching using graph cuts,” *Journal of Visual Communication and Image Representation*, vol. 22, no. 4, pp. 325–331, 2011.

[65] D. A. Altantawy, M. Obbaya, and S. Kishk, “A fast non-local based stereo matching algorithm using graph cuts,” in *9th International Conference on Computer Engineering Systems (ICCES)*, December 2014, pp. 130–135.

[66] D. M. Greig, B. T. Porteous, and A. H. Seheult, “Exact maximum a posteriori estimation for binary images,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 51, no. 2, pp. 271–279, 1989.

[67] V. Kolmogorov and R. Zabih, “What energy functions can be minimized via graph cuts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 26, pp. 65–81, 2004.

[68] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 26, no. 38, pp. 1124–1137, 2004.

[69] C. Russell, P. H. S. Torr, and P. Kohli, “Associative hierarchical crfs for object class image segmentation,” in *International Conference on Computer Vision (ICCV)*, 2009.

[70] Q. Huang, M. Han, B. Wu, and S. Ioffe, “A hierarchical conditional random field model for labeling and images of street scenes,” in *International Conference on Computer Vision and Pattern Recognition (ICCVPR)*, 2011.

[71] P. Kohli, “Minimizing dynamic and higher order energy functions using graph cuts, oxford brookes university, oxford, united kingtom,” Ph.D. dissertation, Oxford Brookes University, November 2007.

[72] V. Kolmogorov, “Convergent tree-reweighted message passing for energy minimization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 28, no. 10, pp. 1568–1583, October 2006.

[73] J. Weston, O. Chapelle, A. Elisseeff, B. Schölkopf, and V. Vapnik, “Kernel Dependency Estimation,” in *Neural Information Processing System (NIPS)*. MIT Press, 2002, pp. 873–880.

[74] M. Johnson, “Pcfg models of linguistic tree representations,” *Computational Linguistics*, vol. 24, pp. 613–632, 1998.

[75] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[76] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, “Support vector machine learning for interdependent and structured output spaces,” in *Proceedings of the 21st International Conference on Machine Learning (ICML)*. New York, NY, USA: ACM, 2004, pp. 104–112.

[77] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, “Large margin methods for structured and interdependent output variables,” *Journal of Machine Learning Research (JMLR)*, vol. 6, pp. 1453–1484, 2005.

[78] B. Taskar, C. Guestrin, and D. Koller, “Max-margin markov networks,” in *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2004, winner of the Best Student Paper Award.

[79] C. Sutton and A. McCallum, “An introduction to conditional random fields,” *Foundations and Trends in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.

[80] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Incorporation, 2006.

[81] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. New York, NY, USA: Cambridge University Press, 2002.

[82] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[83] J. C. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” in *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.

[84] R. Hänsch, “Generic object categorization in polsar images - and beyond, technische universität berlin, germany,” Ph.D. dissertation, Technische Universität Berlin, 2014.

[85] M. Szummer, P. Kohli, and D. Hoiem, “Learning crfs using graph cuts,” in *European Conference on Computer Vision*, October 2008.

[86] T. Joachims, T. Finley, and C.-N. J. Yu, “Cutting-plane training of structural svms,” *Machine Learning Journal*, vol. 77, pp. 27–59, 2009.

[87] R. Cottle, J. Pang, and R. Stone, *The Linear Complementarity Problem*. Society for Industrial and Applied Mathematics, 2009.

[88] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[89] B. Schling, *The Boost C++ Libraries*. XML Press, 2011.

[90] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” *International Journal of Computer Vision (IJCV)*, vol. 111, no. 38, pp. 98–136, January 2015.

[91] S. Suzuki and K. Abe, “Topological structural analysis of digitized binary images by border following.” *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[92] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr, “Conditional random fields as recurrent neural networks,” in *International Conference on Computer Vision (ICCV)*, 2015.

[93] D. Meagher, “Geometric modelling using octree encoding,” *Computer Graphics and Image Processing*, vol. 19, no. 2, pp. 129–147, January 1982.

[94] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[95] M. J. Todd and E. A. Yildirim, “On khachiyan’s algorithm for the computation of minimum-volume enclosing ellipsoids,” *Journal of Discrete Applied Mathematics*, vol. 155, no. 13, pp. 1731–1744, August 2007.

[96] OpenNI, *OpenNI User Guide*, Open Natural Interaction, November 2010.

[97] D. C. Brown, “Close-range camera calibration,” *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1971.

[98] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment - a modern synthesis,” in *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ser. International Conference on Computer Vision ’99. London, UK, UK: Springer-Verlag, 2000, pp. 298–372.

[99] K. Ni, D. Steedly, and F. Dellaert, “Out-of-core bundle adjustment for large-scale 3d reconstruction,” in *IEEE 11th International Conference on Computer Vision (ICCV)*, October 2007, pp. 1–8.

[100] S. Agarwal, N. Snavely, S. Seitz, and R. Szeliski, “Bundle adjustment in the large,” in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6312, pp. 29–42.

[101] R. Maier, J. Sturm, and D. Cremers, “Submap-based bundle adjustment for 3d reconstruction from rgb-d data,” in *German Conference on Pattern Recognition (GCPR)*, Münster, Germany, September 2014.

[102] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[103] M. A. Fischler, Bolles, and R. C., “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[104] J. McGlone, E. Mikhail, J. Bethel, and R. Mullen, *Manual of Photogrammetry*. American Society for Photogrammetry and Remote Sensing, 2004.

[105] V. Lepetit, F.Moreno-Noguer, and P.Fua, “Epnp: An accurate o(n) solution to the pnp problem,” *International Journal of Computer Vision (IJCV)*, vol. 81, no. 2, 2009.

[106] D. G. R. Bradski and A. Kaehler, *Learning OpenCV*, 1st ed. O’Reilly Media, Incorporation, 2008.

[107] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, “Registration with the point cloud library: A modular framework for aligning in 3-d,” *IEEE Robotics and Automation Magazine (RAM)*, vol. 22, no. 4, pp. 110–124, 2015.

[108] P. Kohli and P. H. S. Torr, “Dynamic graph cuts for efficient inference in markov random fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 29, no. 12, pp. 2079–2088, December 2007.

[109] P. Kohli, M. Pawan, K. Philip, and H. S. Torr, “P3 and beyond: Solving energies with higher order cliques,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[110] L. Spinello and K. O. Arras, “People detection in rgb-d data.” in *Proceedings of The International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[111] M. Munaro, F. Basso, and E. Menegatti, “Tracking people within groups with rgb-d data.” in *International Conference on Intelligent Robots and Systems (IROS)*. Institute of Electrical and Electronics Engineers, 2012, pp. 2101–2107.

[112] A. Teichman, S. Miller, and S. Thrun, “Unsupervised intrinsic calibration of depth sensors via SLAM,” in *Robotics: Science and Systems*, 2013.

[113] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *International Conference on Robotics and Automation (ICRA) Workshop on Open Source Software*, 2009.

[114] J. Blanco, “The mobile robot programming toolkit (mrpt),” 2009.

[115] K. Khoshelham, “Accuracy analysis of kinect depth data,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS)*, vol. XXXVIII-5/W12, pp. 133–138, 2011.

[116] K. Khoshelham and S. O. Elberink, “Accuracy and resolution of kinect depth data for indoor mapping applications,” *Sensors*, vol. 12, pp. 1437–1454, 2012.

[117] H. C. Daniel, J. Kannala, and J. Heikkila, “Joint depth and color camera calibration with distortion correction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, no. 10, pp. 2058–2064, October 2012.

[118] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 22, pp. 1330–1334, 1998.

[119] C. Zhang and Z. Zhang, “Calibration between depth and color sensors for commodity depth cameras,” in *Proceedings of the 2011 IEEE International Conference on Multimedia and Expo (ICME)*. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1–6.

[120] C. D. Herrera, J. Kannala, and J. Heikkilä, “Accurate and practical calibration of a depth and color camera pair,” in *Proceedings of the 14th International Conference on Computer Analysis of Images and Patterns - Volume Part II*, vol. 2nd. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 437–445.

Publication List

Human Recognition in RGBD combining Object Detectors and Conditional Random Fields
Amplianitis, K., Hänsch, R., and Reulke, R.
International Conference on Computer Vision Theory and Applications
Rome, Italy, 2016

Towards a 3D Pipeline for Monitoring and Tracking People in an Indoor Scenario using multiple RGBD Sensors
Amplianitis, K., Adduci, M., and Reulke, R.
International Conference on Computer Vision Theory and Applications
Berlin, Germany, 2015

A Quality Evaluation of Single and Multiple Camera Calibration Approaches for an Indoor Multi Camera Tracking System
Adduci,* M., Amplianitis,* K., and Reulke, R. (*equal contribution)
International Society for Photogrammetry and Remote Sensing
Riva del Garda, Italy, 2014

Calibration of a Multiple Stereo and RGBD Camera System for 3D Human Tracking
Amplianitis, K., Adduci, M., and Reulke, R.
International Society for Photogrammetry and Remote Sensing
Barcelona, Spain, 2014

3D Detection and Tracking in an Indoor Environment
Amplianitis, K., Adduci, M., and Reulke, R.
3D - NordOst, Application-oriented Workshop on Measuring, Modelling, Processing and Analysis of 3D - Data
Berlin, Germany, 2014

3D Personenerkennung und Verfolgung mit Stereo und RGBD Kameras
Adduci, M., Amplianitis, K., Misgaiski-Haß, M., and Reulke, R.
3D - NordOst, Application-oriented Workshop on Measuring, Modelling, Processing and Analysis of 3D - Data
Berlin, Germany, 2013

Selbständigkeitserklärung

Ich erkläre, dass ich die Dissertation selbständig und nur unter Verwendung der von mir gemäß § 7 Abs. 3 der Promotionsordnung der Mathematisch-Naturwissenschaftlichen Fakultät, veröffentlicht im Amtlichen Mitteilungsblatt der Humboldt-Universität zu Berlin Nr. 126/2014 am 18.11.2014 angegebenen Hilfsmittel angefertigt habe.

Berlin, den ____________________

Konstantinos Amplianitis