Investigation of Tracking Processes Applicable to Adjacent Non-overlapping RGB-D Sensors

Author: Emilio J. Almazán
Director of Studies: Professor Graeme A. Jones

Digital Imaging Research Centre
Faculty of Science, Engineering and Computing
Kingston University
Penrhyn Road, Kingston-upon-Thames
KT1 2EE, London, U.K.

This Thesis is being submitted in partial fulfilment of the requirements of Kingston University for the Degree of
Doctor of Philosophy (Ph.D.)

October 2014

1. External Examiner: Dr. Antonio Sanz Montemayor
Departamento de Ciencias de la Computación
Grupo GAVAB - Linea CAPO
Universidad Rey Juan Carlos
C/Tulipán, S/N,
28933 - Móstoles - MADRID
SPAIN

2. Internal Examiner: Professor Tim Ellis
School of Computing and Information Systems (CIS)
Faculty of Science, Engineering and Computing
Kingston University London
Penrhyn Road, Kingston-upon-Thames,
London, KT1 2EE,
United Kingdom

Day of the defence: 26/09/2014.

Signature from Chair of Ph.D. committee:

Digital Imaging Research Centre (DIRC)
Faculty of Science, Engineering and Computing (SEC)
School of Computing and Information Systems (CIS)
Kingston University London
Penrhyn Road, Kingston-upon-Thames
London, KT1 2EE
United Kingdom

Declaration

This report is submitted as requirement for a Ph.D. Degree in the School of Computing and Information Systems (Faculty of Science, Engineering and Computing) at Kingston University. It is substantially the result of my own work except where explicitly indicated in the text.

No portion of the work referred to in this report has been submitted in support of an application for another degree or qualification of this or any other UK or foreign examination board, university or other institute of learning.

The thesis work was conducted from July 2011 to August 2014 under the supervision of Professor Graeme A. Jones in the Digital Imaging Research Centre (DIRC) of Kingston University in London.

Kingston-upon-Thames, London, United Kingdom.

Copyright Statement

1. The author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright and rights in it (the "Copyright") and he has given to Kingston University certain rights to use such Copyright for any administrative, promotional, educational and/or teaching purposes.

2. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

3. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

4. The report may be freely copied and distributed provided the source is explicitly acknowledged and copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

5. Further information on the conditions under which disclosure, publication, exploitation and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy, in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations and in The University's policy on presentation of Theses.

Abstract

The work presented in this thesis provides a framework for monitoring wide area indoor spaces built from multiple Microsoft Kinect sensors. A large field of coverage is achieved by placing the sensors in a non-overlapping configuration to reduce the interference between the projected structured patterns. A novel procedure is proposed for estimating the geometric calibration between sensors that enables a common representation for all data by providing many corresponding planes in the view volume of each sensor using a "paddle".

Within this framework, an investigation is conducted of different depth-based spaces for people detection and tracking purposes. Kinect v.1 sensors bring a multitude of benefits to surveillance applications, mainly for occlusion reasoning. However, this sensor has important limitations in terms of resolution, noise and range. In particular, data becomes more scattered with distance along the optical axis of the camera resulting in non-homogeneous representations throughout the range. Furthermore, when considering the aggregated view, each camera produces a different orientation of data. The polar coordinate space representation of the common ground plane is proposed that mitigates these limitations and effectively aggregates the data from all sensors.

The use of discriminative appearance models is a chief aspect in order to properly distinguish people from each other, especially where the density of people is high. A multi-part appearance model is presented in this work – the chromogram – which combines colour with the height dimension offering high discriminative capabilities especially during occlusions periods.

A critical stage for multi-target tracking systems is establishing the correct association between targets and measurements; also known as the data association problem. In this context, the data association stage is investigated by evaluating different well known data association methodologies. An alternative tracking approach which does not require a data association process is also analysed – the Mean-Shift tracker. A modified version of the Mean-Shift tracker is proposed for tracking on the ground plane that integrates the use of chromograms that reduces distractions from the background and other targets.

A new challenging dataset is proposed for the evaluation of multi-target tracking algorithms. The tracking methodologies proposed in this work are compared quantitatively in this framework.

Acknowledgments

First and foremost I would like to express my deepest gratitude to my supervisor Graeme A. Jones, whose guidance and constant support throughout these years have been truly invaluable. Without his knowledge, motivation and enthusiasm this study would not have been possible.

Thanks to Dr. Dimitrios Makris, Dr. Gordon Hunter and Dr. Francisco Flórez for their assistance and helpful discussion of ideas specially during the last part of my PhD. I would also like to thank Dr. James Orwell and Dr. Vasilis Argyriou whom I had the pleasure to assist during their teaching, which was a very enjoyable and enriching experience. My gratitude extends to all academics of DIRC who offered me help at some point during my PhD.

I am also very grateful to all my colleagues and friends at Kingston University whom I have worked with over the last three years: Simi Wang, Pau Climent, Charles Mallah, Spyridon Bakas, Victoria Bloom, Mattheos Doulgerakis, Sateesh Patel and so many others. Thank you all for listening, offering me advice, and supporting me through this entire process. The good moments shared with all these people have kept me going during the toughest times.

Last but no means least, this journey would not have been possible without the invaluable support of my wife and family in Spain.

To María.

Contents

List of Figures

List of Tables

Acronyms

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
Blob	Binary large object
CAMSHIFT	Continuously Adaptive Mean Shift
CCTV	Closed Circuit Television
CS	Coordinate System
EKF	Extended Kalman Filter
FN	False Negative
FOV	Field Of View
FP	False Positive
FPR	False Positive Rate
GNN	Global Nearest Neighbour
GT	Ground Truth
HOG	Histogram of Oriented Gradients
IDC	ID change
i-LIDS	Imagery Library for Intelligent Detection Systems
HSV	Hue-Saturation-Value colour space
INN	Iterative Nearest Neighbour
IPS	Image Plane Space
IR	Infra-Red
JPDAF	Joint Probabilistic Data Association Filter
KF	Kalman Filter
MHT	Multi-Hypothesis Tracker

NN	Nearest Neighbour
NNSF	Nearest Neighbour Standard Filter
OpenCV	Open Computer Vision Library
PDF	Probability Density Function
PETS	Performance Evaluation of Tracking and Surveillance
RGB	Red-Green-Blue colour space
RGB-D	Red-Green-Blue-Depth
ROC	Receiver Operating Characteristic
SIFT	Scale-Invariant Feature Transform
SVD	Singular Value Decomposition
TPR	True Positive Rate
SNN	Suboptimal Nearest Neighbour
UKF	Unscented Kalman Filter
VATIC	Video Annotation Tool - UC Irvine
ViPER-GT	Video Performance Evaluation Resource - Groud Truth

Chapter 1

Introduction

Visual surveillance applications are used for monitoring private and public spaces with a wide range of purposes such as identification and prevention of illegal behaviours, facilitating crimes investigations, traffic control, monitoring of patients, homeland security applications, etc.

These systems proved their effectiveness after the Boston marathon bombing of 2013 where the suspects were identified by inspecting the CCTV footage. This incident also revealed the necessity of more intelligent systems capable of detecting threats in real time. Nowadays, a growing need for public security has lead governments and private companies worldwide to invest in the development of more sophisticated surveillance systems. The UK government announced in early 2014 an investment of £1.1bn on high-tech surveillance systems. The US government is currently spending $3.7bn on the development of drones for frontier control. The Danish government has invested DKK 15 million in surveillance solutions to automatically interpret and describe video (Milestone XProtect© 2014)[1].

Visual surveillance applications are present in a wide range of applications in society. Last year Panasonic released a multi-camera in-car system to aid police officers that starts recording when a relevant incident is detected. In the retail and marketing sector an emerging trend is to analyse in-store customer behaviour for video analytic and statistics. The low-cost airline company Easyjet is developping drones to inspect its fleet of Airbus aircrafts. The drones will be used to scan and assess Easyjet planes and report damage back to engineers. Recently Shanghai airport has installed a network of almost 2000 fixed and PTZ cameras for access control, fire detection and luggage handling system. Furthermore, the farming sector is starting to use cameras for monitoring daily operations and watch over feed lanes.

Surveillance applications normally rely on traditional intensity-based cameras. However, these are highly sensitive to illumination conditions and occlusions, and therefore the use of sophisticated algorithms is required in order to mitigate these effects. Another possibility to address these problematic situations is the use of alternative sensors. Since

(a) Hover drone with an integrated camera.

(b) Unmanned Aerial Vehicle (UAV) from the UK armed force.

Figure 1.1: The use of drones and UAVs have great potential for surveillance tasks such as traffic control, monitoring of farms or frontier control. However, there are still much controversy with their use in public spaces due to the lack of regulations and issues related to violation of personal privacy. Fig (a)[1], Fig (b) [2].

the release of the affordable Kinect sensor by Microsoft, the use of RGB-D cameras has become very popular especially in the research community. Apart from colour, these sensors also provide with depth information which is robust to illumination conditions and highly valuable for identifying and resolving occlusions. RGB-D sensors offer the possibility of monitoring crowded indoor environments such as airports, train stations, shopping centres, etc. Additionally, since they do not require an external light source they can be used in dark environments. There are, however some issues associated with these sensors in terms of limited range, resolution and noise that restrict their use to a certain type of applications. Researchers have not investigated in much detail the use of RGB-D sensors beyond their operating range of up to 4-5 m. In this context, this project provides an opportunity to advance the use of RGB-D sensors in the field of visual surveillance.

The present work have plenty of applications in real world scenarios. In particular for monitoring wide area indoor spaces such as airports, museums or parking lots. Additionally, it could be applied for night surveillance – e.g. monitoring of patients at night or in offices outside their opening hours.

1.1 Aims and objectives

In this work an investigation will be conducted on the use of multiple RGB-D cameras for detection and tracking people in large indoor spaces, which is expected to provide

[1]Photography by: Don McCullough, Title: "Drone and Moon"
https://www.flickr.com/photos/69214385@N04/

[2]Photography by: UK Ministry of Defence, Title: "Watchkeeper Air System"
https://www.flickr.com/photos/defenceimages/

Figure 1.2: CCTV cameras in Victoria Station, London (UK)[1]. The city of London has one of the highest number of CCTV cameras of any city in the world. It has been estimated that on average an individual may be recorded by more than 300 cameras in a single day.

multiple benefits to the field of visual surveillance. The following are the main objectives of the thesis:

- The capabilities and limitations of RGB-D sensors will be analysed in terms of maximum range, depth resolution, accuracy and interferences produced between sensors. This study will allow the design of an optimal configuration of multiple RGB-D sensors for monitoring wide area indoor spaces.

- In order to use efficiently the data from multiple cameras, a calibration methodology will be proposed to enable a common representation for the data.

- The depth dimension will be explored aiming to obtain an optimal space that effectively aggregates the data from all sensors, mitigates the main limitations of RGB-D cameras, and allows people segmentations beyond the depth sensor operating range.

- Target tracking methodologies will be investigated to be used in complex situations with multiple people and occlusions. The depth dimension will be introduced to provide an optimal tracking space that minimizes the number of occlusions.

[1]Photography by: Antonio Martínez, Title: "Último día"
https://www.flickr.com/photos/poper

- An important objective will be to produce a discriminative depth-based appearance model that effectively distinguishes people from one another. Such a model will be studied in the context of multi-target tracking particularly during occlusion situations.

- The design of a new and challenging dataset will be investigated aiming to serve as a suitable platform for the evaluation of people segmentation and tracking algorithms. In order to produce a functional and comprehensive dataset an analysis will be conducted to identify the most relevant situations in multi-target tracking.

1.2 Issues

There are specific issues that will need to be considered in this project.

The Microsoft Kinect sensors presents important limitations in terms of resolution and noise that restrict their operating range to 4-5 metres. Due to the nature of the depth sensor based on triangulation the amount of noise increases with distance. Effects such as blurring, pixelation and quantization are expected to introduce additional inaccuracies in the results. Additionally, the depth resolution decreases with distance which means that the gaps between contiguous depth values increase. These issues complicate the use of depth data beyond the operating range of the sensor.

Another critical issue related to the Kinect sensor is the fact that it cannot be used effectively outdoors in presence of direct sunlight or in combination with more Kinect sensors, namely when all sensors work on the same scene simultaneously. The Kinect sensor is in essence a structured light sensor, and in short it works by projecting a fixed infra-red (IR) pattern of dots onto the scene which is captured by an IR camera. The depth of each dot is estimated by comparison with the corresponding dot in a pre-loaded pattern captured at a known distance. When there are external sources of IR light projecting onto the same scene (i.e. sunlight, other sensors), the sensor struggles to identify its own dots resulting in areas with no depth estimation or erroneous values.

An additional issue common to all visual surveillance systems are occlusions. This is probably one of the biggest challenges to resolve in detection and tracking scenarios. Occlusions can be classified as partial occlusion when only some areas of the target get occluded, or total occlusion when the target disappears completely from the scene. They can also be static or dynamic depending on whether the target got occluded by an element from the scene or by other targets, even by the target itself due to rotations or pose changes. The correct identification of targets during and after occlusions is highly difficult to resolve since the appearance of targets inevitably changes.

The colour camera of the Kinect is affected by classical issues related to illumination

conditions. In particular, illumination changes affect directly the appearance model of targets resulting on possibly incorrect identification of targets. This effect is even more noticeable when multiple cameras are employed, where each camera has its own shutter configuration at each time step yielding different colour representations. Furthermore, effects like shadows or cluttered backgrounds could interfere with the targets' appearance model.

A final issue refers to the design of a suitable evaluation framework that allows the effective assessment of the algorithms presented in this work. It will require the design of a dataset that covers the challenging situations intended to solve and the identification of the relevant failure modes of the system.

1.3 Contributions

The main contributions of this thesis are:

- A semi-automatic calibration procedure that estimates the geometric transformations between pairs of non-overlapping range sensors. The proposed calibration methodology uses corresponding planes to derive constraints on rotation and translation.

- A depth-based polar coordinate space representation that mitigates important limitations of RGB-D sensors in terms of range, resolution and noise. It also aggregates effectively the data from all sensors allowing segmentations of people beyond the operating range of the sensor while minimizing the number of occlusions.

- Presenting a discriminative new multi-part target appearance model – the "chromogram" – which combines the height dimension in the 3D space with colour information. This model is especially intended to serve effectively during occlusions.

- A challenging dataset recorded from three non-overlapping RGB-D sensors is presented. Furthermore, the ground truth annotations, relevant failure modes and evaluation metrics are included for a comprehensive evaluation of multi-target tracking algorithms.

1.3.1 Publications to date

- E. J. Almazán and G. A. Jones. Tracking People Across Multiple Non-Overlapping RGB-D Sensors. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on. IEEE, 2013, pp. 831-837.

- E. J. Almazán and G. A. Jones. Multiple Non-Overlapping RGB-D Sensors for Tracking People. In Robotics: Science and Systems, 2013.
- Submitted for a conference workshop: E. J. Almazán and G. A. Jones. A Depth-based Polar Coordinate System for People Segmentation and Tracking with Multiple RGB-D Sensors. In IEEE ISMAR 2014 Workshop on Tracking Methods & Applications 2014.

1.4 Structure of the thesis

This section presents a brief outline of the thesis.

In chapter 2 a review of the state of the art in visual surveillance systems is conducted with especial consideration to detection and tracking methodologies. In particular it is described the necessity of solving the data association problem and the use of discriminative appearance models for multi-target tracking environments. Some of the most recent configurations of multi-camera systems for surveillance purposes are discussed along with popular performance evaluation metrics for detection and tracking applications.

Chapter 3 presents the surveillance framework proposed in this work. First, the RGB-D sensor is analysed individually to assess its capabilities in terms of resolution, noise and maximum range. Second, the design of a non-overlapping configuration of cameras is presented aiming to maximize the field of coverage and minimize the interference between IR sensors. Finally, it is introduced a novel semi-automatic procedure for the calibration of multiple non-overlapping range cameras that enables a common representation for all data.

In chapter 4 the depth dimension is explored in the context of people segmentation. Three alternative depth-based spaces are presented with the main objectives of effectively aggregating the data from all sensors and reducing the number of occlusions. A novel space is introduced that mitigates the main limitations of RGB-D sensors in terms of resolution and noise allowing segmentations beyond the operating range of the sensor.

Two fundamentally different tracking methodologies are explored in chapter 5 – the Kalman filter and the Mean-Shift tracker. The Kalman filter is studied from the perspective of data association where different methodologies are presented. The Mean-Shift approach is discussed and its main limitations in multi-target tracking environments are identified. Some important enhancements are proposed to increase its performance. Additionally, a discriminative appearance model that combines the absolute height of the target and colour information is presented. This model is especially intended to be effective during occlusions and robust to changes in targets' scale.

The apparatus for the evaluation of multi-target trackers is presented in chapter 6. A challenging dataset for segmentation and multi-target tracking is produced along with the ground truth annotations. A study is conducted to identify the relevant failure modes of the system and a set of metrics is discussed to provide meaningful evaluation. The two trackers methodologies introduced in chapter 5 are assessed and compared quantitatively within the proposed evaluation framework.

The final chapter presents a discussion on the main contributions and achievements of this thesis, combined with some conclusions and suggested future research direction.

Chapter 2

Literature Review

Typically, CCTV surveillance systems are used offline in courts as proof to incriminate people, or for online inspections by operators at central monitoring locations. The efficiency of the system then relies upon the operator, who is required to concentrate on monitors for long periods of time, a tedious task highly prone to distractions. The increasing computer power has allowed computer vision techniques to be applied on the footage obtained by CCTV systems. Nowadays, visual surveillance tends towards more intelligent systems where relevant situations e.g. illegal behaviours are detected automatically in real time [2, 3].

There is now an extensive line of research in the use of alternative sensors such as range sensors [4, 5], especially since the release of the affordable RGB-D *Kinect*© camera by Microsoft. These sensors allow the exploration of different modalities that aim to address some of the challenges in video surveillance such as occlusions, varying illumination conditions or shadows.

In general the classical pipeline of video surveillance applications consists of foreground segmentation, data association, tracking and in some cases event detection or action recognition modules. When multiple cameras are employed, a prior step for calibration should be performed to allow the integration of data from all cameras. The stage at which the integration is applied varies depending on the system as depicted in figure 2.1 where two common locations to perform the fusion are shown.

This chapter covers the review of the most relevant aspects and methodologies of video surveillance applications. In section 2.1 some of the most popular people segmentation techniques based on foreground detection are discussed. Multi-target tracking techniques are presented in section 2.2. Section 2.3 covers different multi-camera environments along with calibration techniques. In section 2.4 is discussed the performance evaluation of people detection and multi-target trackers.

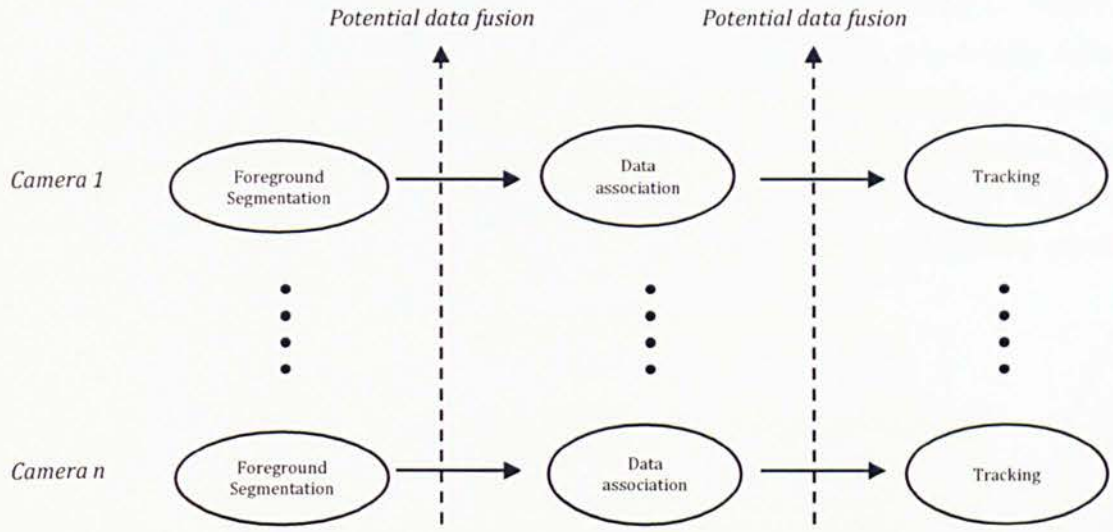


Figure 2.1: Pipeline of a video surveillance application. It includes common locations where the fusion of data from the cameras is performed.

Figure 2.2: Control room[1]. Operators spend long hours looking at surveillance monitors, a tedious task prone to distractions.

2.1 Background Subtraction for People Segmentation

People segmentation is commonly approached in surveillance applications by means of foreground detection techniques. These are in general based on background subtraction,

[1]Photography by: Paul Gorbould, Title: "Bold & Doc" https://www.flickr.com/photos/gorbould/

where the current image is compared with a background model aiming to detect the differences which are identified as foreground regions. This process is normally followed by a step of refinement to reduce noise and group foreground pixels in connected components or blobs – see figure 2.3. In tracking systems people segmentation is used for the automatic initialization of tracks and in many approaches is used as well during the actual tracking of targets. It increases the speed and accuracy of tracking since the search space is reduced.

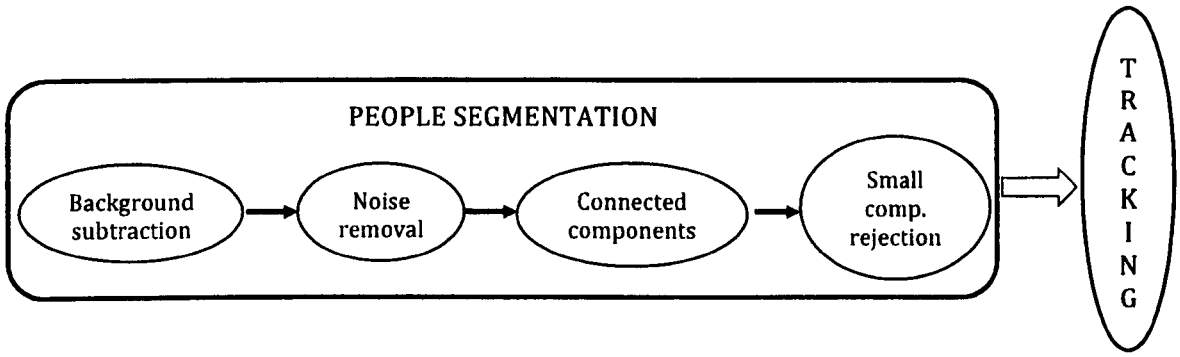


Figure 2.3: People segmentation pipeline.

In the literature, a huge variety of background subtraction approaches can be found. The simplest methods assume static backgrounds [6–8]. The majority of these techniques model the background pixel-wise using simple models such as using the frame before, to more advanced methods that use a number of static background images taken at the beginning of the sequence. Lo and Velastin [9] compute the median of the first N frames. Wren et al. [10] proposed a very effective way to handle illumination changes by modelling the background pixels with single Gaussians.

Dynamic backgrounds

The aforementioned techniques will fail in scenarios with systematic background movements e.g. waving trees, snow, etc. Friedman and Russell [11] presented a traffic surveillance application that models the scene with three different Gaussians; one for the road, one for the shadows and one for the cars. Based on this idea Stauffer and Grimson [12] proposed a general approach using a mixture of K Gaussians (MOG) to model the background. Elgammal et al. [13] relaxed the Gaussian constraint by presenting a non-parametric model using Kernel Density Estimators (KDE). Oliver et al. [14] proposed a method where the background is modelled in the eigenspace. Li et al. [15] use a Bayesian decision rule for background and foreground classification. Kim et al. [16] introduced another non-parametric method with limited memory requirements known as the codebook algorithm. Additionally, the presence of shadows can be particularly problematic and most authors try to mitigate their effect by using different

colour space less sensitive to brightness changes, such as chromaticity [6, 10, 17] or the Hue-Saturation-Value (HSV) colour space [18].

All these methods are considered standard solutions and even though most of them were proposed more than a decade ago, they are still widely used nowadays with minor variations [19–22]. Although they are specifically designed to address issues such as illumination changes, shadows and dynamic backgrounds, some authors have considered combining different features to increase the performance e.g. colour, texture or edges to obtain more reliable results [23–26]. Other approaches contemplate representations not only at a pixel level but at a region and frame level as well [27–29]. Using different model levels, problems like light switching can be handled more accurately.

Depth-based models

The use of information such as texture or edges is still dependent on the data captured by intensity-based cameras, which means that they are sensitive to the same issues as colour-based models (shadows, illumination changes, etc.). Some authors have explored the use of alternative sensors such as Time-of-Flight cameras, stereo systems or RGB-D cameras trying to mitigate these problems [30–34]. Depth is a powerful feature for background subtraction since it has been proven to be invariant to illumination changes and shadows [35–37]. However, depth on its own has some limitations. For instance it fails to segment people that are at the same distance to the background or people at the same depth. In addition, when the segmentation is integrated into a tracking system the identification of people using just depth is problematic.

In summary, the final selection of the background model mainly depends on the application itself and the type of scene. Simple models such as the frame before or the first frame of the sequence [8, 9] might be enough in controlled environments i.e. scenes without illumination changes or background movements. However, other situations such as outdoor scenes with the presence of wind and illumination changes require more sophisticated approaches [12, 13] or even the use of alternative sensors (e.g. RGB-D, Time-of-Flight).

2.2 People tracking

Tracking people consists of identifying consistently people over time. Algorithms for tracking are widely used in video surveillance applications for monitoring public spaces [38], detection and identification of illegal behaviours [39] and even in the sport industry for tracking players and tactical analysis [40].

2.2.1 Tracking methodologies

Generally, tracking methodologies rely on the availability of target detections at every time step which are integrated over time to form complete tracks. The motion and appearance model of targets are used to assist in the integration stage. Kalman filters and particle filters are by far the most popular tracking algorithms. Additionally, it is important to mention a recent tracking methodology proposed by Zdenek et al. [41] called Tracking-Learning-Detection, which has become very popular in recent years. The main novelty of this technique is the online learning stage that increases the performance of the detector over time.

Kalman Filter

The Kalman filter was proposed initially in 1960 by R. E. Kalman [42] as a recursive solution to the discrete data linear filtering problem. Since then, the method has been the subject of numerous investigations due to its great potential and computational efficiency. In particular it is widely used in the context of visual surveillance for tracking people. The method estimates recursively the state of a person (e.g. location, velocity, acceleration, etc.) using a two stage procedure: prediction of the state, and update given the current observation of the person (obtained from the sensor and the segmentation module). The prediction stage employs a motion model that is built upon the history of the target, and the observation refers to the segmentation produced by the people segmentation module at each time step. Additionally, it provides mechanisms to allow certain degree of inaccuracy or noise in the models and observations. Kalman filter is commonly referred to as the "optimal" solution to the state estimation problem in the sense that minimizes the mean square error of the estimated parameters, but only when some conditions are satisfied.

The Kalman filter assumes the target state is a Gaussian distribution, the motion model, and the measurement model[2] are linear, and the inaccuracies of the motion model and the inevitably noise of the measurements can be modelled with Gaussian distributions. In case any of these conditions are not completely satisfied, some authors have proposed different alternatives. The Extended Kalman Filter (EKF), also known as the non-linear version of the Kalman filter, presents a solution for Gaussian non-linear systems. It approximates the non-linear functions using the Taylor's expansion. EKF is especially aimed for systems that can be easily linearised (i.e. near linear), and it will probably diverge for highly non-linear models. EKF is being widely used in surveillance applications [43, 44] and a comprehensive analysis of the technique can be found on the work of Ribeiro [45] and Welch and Bishop [46].

To overcome the limitations of EKF, Julier and Uhlmann [47] proposed the Unscented

[2]The measurement model is used to transform the state into the measurement space.

Kalman Filter (UKF), which is a method designed to handle highly non-linear systems. UKF uses a deterministic approach using samples to obtain the mean and the covariance of the probability density function. A set of samples called "sigma points" are chosen near the mean and are propagated through the non-linear function. The mean and the covariance are recovered in the new sample distribution and Kalman filter is applied normally [48].

EKF and UKF still assume the distributions are Gaussian. For those systems where these constraints are not satisfied there is fortunately another tool for state estimation – the particle filter.

Particle filter

The Particle filter was introduced in 1993 by Gordon et al. [49], where it was first called bootstrap filter. It is considered a generalization of the Kalman filter since it can be applied to any system e.g. non-Gaussian, non-linear. It was used first in a computer vision application by Isard and Blake [50].

The concept of particle filters is to represent the state density of a particular target using a population of samples randomly distributed through the feature space, where the samples represent hypothetical states of the target. This allows an accurate definition of the state distribution as long as enough samples are used. The samples are weighted according to their similarity with the observations received at every time step. In the original paper the particles were updated using a scheme known as Sequential Important Sampling (SIS) based on the motion model and the observations. Further extensions to the original method have been proposed such as the Sampling Importance Resampling (SIR) [51], where at every iteration samples with low weight are replaced avoiding the problem of "sampling impoverishment".

Particle filters are widely used in computer vision applications, in particular in video surveillance systems [52–58]. A further discussion of particle filters is given by Auralampalam et al. [59].

People appearance modelling

Building reliable appearance models is a real challenge in particular for visual surveillance applications due to factors such as illumination changes, occlusions and variation of target poses and orientations. In particular, when tracking people across cameras with non-overlapping views the illumination and appearance of the person might change significantly from camera to camera. The use of discriminative models is essential in multi-target systems for the correct distinction between targets.

Over the years researchers have proposed a wide range of appearance models that aim to deal with these situations. In general appearance models can be classified

between local and global models. Local models capture the local structure of the target and are characterized for being partly robust to illumination changes, partial occlusions and orientation and pose variations such as HOG [60] or SIFT features [61]. However they are normally expensive to compute and might require *a priori* knowledge of the target and a number of samples for training the models [62, 63]. Local features therefore are not normally considered for fast tracking algorithms. On the other hand, global models are simpler and faster to compute but they are more sensitive to illumination and orientation changes, and occlusions. Two of the most popular models within this latter category are templates and histograms.

Templates are structures constructed using directly the raw information of the pixels within the boundaries of the object. They are simple representations that preserve the spatial structure of the target along with their intensity [64–66]. Nonetheless, templates present major problems in varying illumination conditions when targets undergo pose or orientation changes and during occlusions. Many authors have proposed different enhancements by introducing additional information such as edges or texture [67, 68] or even for dealing with scale changes [69].

Histograms are very popular representations that capture the distribution information of the objects. They are widely used in visual surveillance [70, 71] since, unlike templates, can handle scale and orientation changes. However, they are mainly criticized for not preserving the spatial structure of the object. In the literature, histograms have been extended by many authors to mitigate this problem with the use of multi-part histograms [36, 72–74]. The data is divided spatially in regions to improve the discriminative capability of the model as well as make it more suitable for dealing with occlusions. Wren et al. [10] proposed one of the first multi-part histograms, where regions of similar colour within a person were modelled with different Gaussians e.g legs, torso, head, hands, etc. An interesting approach was proposed by Birchfield and Rangarajan [75] who introduced the so-called "spatiograms" which are structures that augment the standard single cue histogram with the spatial distribution of pixels in each bin.

More recently, the use of 3D data is being used to construct more robust models. Muñoz-Salinas et al. [5] build histograms using the data from the torso, which is approximated in the 3D space. Alternatively, Spinello and Arras [76] propose a local feature that uses histograms of oriented depths inspired by the well known HOG descriptors [60].

Data association

In visual tracking applications data association is the process that assigns the correct measurement to every target at every time step. It uses the similarity between ap-

pearance models of targets and measurements aiming to maximize the total similarity of all associations. This is a key stage in tracking since it is directly related to the target update process. In single target trackers when only one target is considered data association is trivial. It is also a relatively easy problem in multi-target trackers when the targets are well separated. The problem becomes complicated when multiple targets are spatially close, occluded, or in the presence of spurious measurements.

To simplify the process, it is common practice to reduce the space of search by analysing only those measurements that are within a region of high probability of containing the correct measurement (i.e. nearby the target) known as a "gate". However, there are still periods of ambiguity when more than one measurement fall within this region and data association still needs to be addressed.

The data association problem has been studied since the early years of visual tracking and it is interesting to note that the early methodologies are still in use nowadays with little modification. The different approaches can be classified between single-hypothesis and multiple-hypothesis techniques.

Single hypothesis

Single-hypothesis techniques refer to those approaches that produce a single set of associations at every time step. They are normally preferred for their simplicity and practicality in real scenarios. The simplest method within this category is the Nearest Neighbour Standard Filter (NNSF). This method chooses the best association for each target with a single scan without considering a global solution. The simplest NN approach is not normally used for multi-target environments since it allows a measurement to be associated with multiple targets. The Iterative Nearest Neighbour (INN) is more suitable for multi-target tracking scenarios where the solution is constrained to forbid multiple associations. The INN is a simple methodology which is easy to implement and does not require much computational load. It has a complexity of $O(n)$[3]. However, it performs poorly in dense target situations since the solution depends on the order of target association, and it may result in one target stealing other target's measurement [77–79]. A slightly variation of the INN is the Suboptimal Nearest Neighbour (SNN) which does not rely on the order of association, instead searches sequentially for the best possible single association, the one that returns the highest similarity between target and measurement. The SNN has a complexity of $O(n^2)$ and in many situations achieves high performance despite the fact that it does not seek specifically for an optimal solution that maximises the total similarity [80–82]. A technique that guarantees an optimal solution is the Global Nearest Neighbour (GNN). This technique is the most widely used NNSF technique and, unlike the aforementioned methods, achieves an

[3] $n = min(N_t, N_m)$ and N_t and N_m are the number of targets and measurements respectively

optimal solution at every time step. The fastest implementation of GNN was proposed by Munkres [83] with a polynomial complexity of $O(n^3)$ [80, 81, 84–86].

An alternative approach within the single-hypothesis category is the Joint Probabilistic Data Association Filter (JPDAF) proposed by Fortmann, Bar-Shalom and Scheffe [87, 88]. JPDAF is conceptually different to the NNSF approach since it allows multiple associations assuming that the real measurement may not be the closest one and that every association is possible with some probability. Consequently, targets are updated using a weighted combination of all possible associations. JPDAF is specifically designed to deal with noisy environments where spurious measurements are frequent. One limitation of the standard JPDAF is the assumption of a fixed and known number of targets. Schulz [89] proposed a sample-based version of JPDAF that relaxes this constraint.

Multiple hypothesis

Multiple-hypothesis techniques, on the other hand, return a set of possibilities at every time step and the solution is delayed until more information is available. They reach the correct solution with high probability at the expense of an increase in the computational load. The Multi-hypothesis tracker (MHT) [90, 91] is the most popular. It computes all possibilities at every time step including the termination and initialization of new tracks. MHT is commonly represented as a tree where each node indicates a different hypothesis[4]. The tree grows exponentially expanding each current hypothesis with a new set of hypotheses every time a new set of measurements is received. As expected MHT is expensive in terms of memory and computational time and therefore to make it practical requires of optimal implementations [80, 92] as well as approximation techniques. Common approximations are clustering [90], merging of tracks [93] and pruning techniques using Murty's algorithm [94, 95] to keep the K best hypothesis in polynomial time [96, 97].

Split and *merged measurements*

An important consideration during the data association process is the presence of split and *merged measurements* which are frequent in real surveillance scenarios. Split measurements arise due to partial occlusions while *merged measurements* appear as a consequence of the limited resolution of the sensor when several objects are in close proximity. From the data association perspective it is critical to identify and manage these situations. In the literature different approximations can be found in this regard. Joo and Chellappa [95] identify these special cases based on the area of measurements. Bose et al. [98] used the number of measurements in the gate area. If more than

[4]An hypothesis consists of a set of feasible associations

one measurement fall within the gate of a single target it is assumed split, and if a measurement falls in the intersection of two gates it is labelled as a *merged measurement*. Once they are identified, one possibility is to join back together split measurements and decompose *merged measurements* [95, 99]. Alternatively, they can be maintained as split and *merged measurements* with continued estimation of their state until they are detected as single measurements again [98].

2.2.2 Alternative tracking methodologies

There is another type of tracking algorithms that do not rely on the detections made at every time step and therefore do not require to solve the data association problem. Instead the appearance model of the targets is constructed in the first frame (automatically or manually) and the subsequent frames are searched looking for the location more similar to the model. These trackers are some time referred to as data-driven trackers since they only use the data obtained from the images at every time step without the aid of almost any high level information. Two common approaches are template tracking and the more sophisticated Mean-Shift tracker.

Template tracking

Template tracking is considered as one of the simplest approximations for tracking. Targets are modelled with the raw pixel-wise intensities of the area defined by the target. The search for the model in the current frame usually starts from the last estimated position or is predicted with a motion model. It continues by matching the template with nearby location looking for the position that produces the best match. Typical matching methods are the sum of squared differences (SSD) or cross-correlation [100, 101].

Although this algorithm may be a good approximation in some restricted situations, it presents major weaknesses in real environments. It is particularly ineffective in situations of illumination changes, rotations or variations in scale of the target. It also presents problems when tracking multiple people with similar appearance. More practical implementations incorporate mechanisms to deal with these situations, for instance a common practice is to update the model at every time step. Nguyen et al. [69] present a warping method that allows scale changes. Beymer and Konolige [102] use a template based on the disparity information from a stereo system, which is less affected by changes in the illumination conditions.

Mean-Shift tracking

The Mean-Shift tracker is the most popular alternative within this category. The search for the model is performed in an optimal way by a gradient ascent procedure i.e. Mean-Shift. The Mean-Shift tracker is also known as a kernel histogram tracker because it makes use of a kernel to weight the pixels for building the appearance model i.e. pixels closer to the object centre are weighted heavier. It was originally used in computer vision for tracking purposes by Comaniciu, Ramesh and Meer [70]. The Mean-Shift tracker has become a popular tracking method in the last decade as it is efficient computationally and easy to implement. However, it presents the following limitations. First, it does not consider properly the change in the object's scale or rotation. Second, it is highly sensitive to similar backgrounds and interferences produced by nearby targets. Third, in the original implementation the appearance of the object is modelled with a simple single-cue colour histogram, which is normally criticised for not being very discriminative.

Most of the extensions of the Mean-Shift tracker proposed in the literature aim to address these issues. To reduce the distractions from the background some authors applied a previous background subtraction which additionally speeds up the tracking process since less data is considered [103, 104]. In order to increase the performance of the tracker alternative appearance models are suggested such as multi-part histograms where colour is combined with some spatial information [75, 105–108]. Regarding the scale change, some methods adapt the kernel size and orientation using the expectation maximisation algorithm [69, 71] or alternatively the moments of the distributions [109, 110].

The Mean-Shift tracker is also commonly combined with other techniques such as in Comaniciu, Ramesh and Meer [111] where the Kalman filter is used to reduce the number of iterations, or in the paper presented by Li [112] where Mean-Shift is used to improve the data association process in a Kalman filter tracker. Alternatively, it has been combined with the particle filter method to reduce the number of particles [113].

2.3 Multi-camera environments

The reduction in the price of sensors and the increase in the computational power of modern computers have allowed the incorporation of additional cameras to aid in computer vision applications. In particular, for surveillance systems, multiple sensors are used to reduce the number of occlusions [79, 114] and to increase the area monitored by the system [115, 116].

In order to use the information from all cameras in an efficient way, it is normally required to perform a prior calibration process to estimate the relative position of the

cameras with respect to a common coordinate system (CS). The calibration involves the estimation of the geometric transformations i.e. rotation and translation, between all cameras and the reference CS.

2.3.1 Calibration of multiple overlapping cameras

The use of multiple overlapping cameras in surveillance scenarios is used to reduce the number of occlusions. The external calibration of cameras facilitates the correct identification of targets in each camera. For traditional RGB systems a multitude of methods have been proposed that involve the use of corresponding features between cameras to estimate the transformations by error minimization [117, 118]. For close range scenes, it is common to use a chessboard pattern viewable by all sensors where the corners are detected in all views. Alternatively, for more complex scenarios where the cameras are so far that it is difficult to detect the corners of the chessboard, Svoboda et al. [119] proposed a system where a moving bright spot, viewed by all cameras in a dark scene, was used to create the correspondences. Lee et al. [115] instead track a common person in all cameras using the position of the person over time for creating point correspondences. Renno et al. [120] presented a calibration methodology where the image to ground plane homography was estimated by accumulation of tracks.

For multiple range sensors, and in particular for structured light sensors such as the Kinect camera, the configuration of the system is more challenging since each sensor emits a fixed infra-red (IR) pattern at the same wavelength. Therefore each sensor can see another sensor's pattern superimposed on its own and will have problems distinguishing the two. Different approaches have been proposed to address this issue. One of the most popular methods was presented by Butler et al. [121], where a mechanical system was used to vibrate a Kinect sensor. Since the IR projector and IR camera of the Kinect vibrate at the same frequency, its own IR pattern is detected normally while the IR patterns from other sensors are blurred avoiding interferences [122, 123]. Another approach is to use a time slot schedule for each sensor. Since the deactivation of sensors by software is relatively slow, Schröder et al. [124] used an external shutter for time multiplexing. A more sophisticated system was presented by Faion et al. [125] with the development of an internal shutter. Alternatively, a cheaper solution was presented by Maimone and Fuchs [126] with a software solution to fill the holes produced by the interferences.

The calibration of multiple overlapping RGB-D sensors normally requires the use of special calibration grids, such as using a chessboard where the black squares are covered by IR deflected material [127], or with a planar calibration grid with retro-reflective dots [128]. Once the point correspondences are established, standard calibration procedures are used [117, 118]. A more practical approach but less accurate for the external

calibration of multiple Kinect cameras is to estimate first the external relations between RGB cameras using standard procedures [118] and then use the depth-RGB registration provided by the Kinect driver [126].

2.3.2 Calibration of multiple non-overlapping cameras

Systems built from non-overlapping cameras aim to monitor large areas where each sensor covers a different region of the scene. When the cameras are highly separated and there are large unobserved regions the external calibration of cameras is not normally required, since it cannot directly assist in the association of targets across cameras. In these situations it is common to use re-identification techniques based on the appearance model of the targets [129–131]. When the intermediate unobserved regions are small, some authors estimate the trajectory of targets in those areas to help in the re-identification task [132] and even to estimate the calibration parameters [133]. Makris et al. [134] proposed a methodology that learns the topology of the cameras using temporal correlation of objects moving across adjacent cameras. For more restrictive scenarios of close range scenes, Kumar et al. [135] presented an interesting calibration method that allows all non-overlapping sensors to see the same calibration grid with the use of mirrors.

2.4 Performance evaluation

The performance evaluation of algorithms in particular for surveillance systems is a crucial stage to determine the progress during the development stage and to obtain quantitative comparisons with other reported work.

The evaluation of detection and tracking applications is a complicated process that in general involves the following three aspects:

1. Designing a suitable dataset.
2. Producing ground truth annotations for the dataset.
3. Defining a proper set of metrics that allows a meaningful evaluation of the algorithm.
4. Setting the optimal values for the evaluation parameters.

2.4.1 Ground truth and dataset

To evaluate the performance of an algorithm a common approach is to compare the algorithm's results with those considered ideal, also known as ground truth. Producing

accurate ground truth annotations is surprisingly challenging. The process for generating ground truth involves the annotation (e.g. bounding boxes, ellipses) of the objects in every frame of the sequence which is a highly tedious task, especially with long sequences and when many objects are present. For some evaluations the task is even more gruesome as pixel-level accuracy is required [38, 136]. Normally the human annotations are assumed to be the perfect values. However they contain many ambiguities since the process is error prone and often requires subjective interpretations of the scene. It is surprising the variability on the annotations depending on the annotator [137, 138]. Additionally, there are complex situations that are difficult to interpret such as occlusions, objects partially cropped by the image edges, objects on pictures (e.g. ad-boards), reflections in mirrors, etc. To assist in the process there are available several semi-automatic tools such as ViPER-GT [139] or VATIC [140] that offer interpolation tools to avoid the need to annotate every single frame. Some authors have considered the creation of synthetic datasets to avoid the ground truth annotation altogether [141].

Ideally, the datasets should be comprehensive enough to cover a wide range of challenging situations e.g. weather conditions, illumination variations, dynamic backgrounds, occlusions, etc. as discussed by Ellis [142] or more recently in the study of Motwani [143]. For RGB multi-target tracking systems some of the most popular and widely used datasets are PETS [144], i-LIDS [145], CAVIAR [146] or ETISEO [147]. Recently, with the increasing use of RGB-D sensors for surveillance purposes, new datasets have arisen. However, the majority of the RGB-D datasets publicly available are specific for identification of objects [148, 149] and human activity recognition [150, 151]. There are still very few for evaluation of multi-target tracking systems. One is recently published by Munaro and Menegatti [86] called Kinect Tracking Precision (KTP), which is a dataset acquired from a mobile robot platform. Alternatively Spinello and Arras [76] made available a dataset recorded from static RGB-D cameras for people tracking purposes.

2.4.2 Evaluation metrics

Ground truth-based metrics are in general computed from the classical true positives (TPs), false negatives (FNs), true negatives (TNs) and false positives (FPs). Some of the most well known metrics used for evaluation of detection and tracking algorithms are true positive rate $\frac{TP}{TP+FN}$, false positive rate $\frac{FP}{FP+TN}$ and specificity $\frac{TN}{TN+FP}$. However, as it was cleverly identified in the work of Lazarevic-McManus et al. [152] the TNs cannot be computed for object-based systems and therefore popular evaluation metrics such as the ROC curve cannot be applied.

Metrics can be classified as global or local. Global metrics present a single value to assess the overall performance of the algorithm which is convenient for comparison

purposes. A popular global metric is the CLEAR MOT metric [153] which comprises two metrics: MOTP for the evaluation of object detections based on the spatial overlapping between ground truth and detected objects, and MOTA which accounts for the spatial and temporal overlapping between ground truth and detected tracks. The VACE metric [154] is widely used as well as a global metric where FPs, FNs, ID-switches and track fragmentations are combined in a single value.

Other authors prefer the use of local metrics to obtain a more comprehensive evaluation of the algorithm. Local metrics are especially useful to identify problems during the development stage of the algorithm, or to determine the strengths and weaknesses of a particular algorithm [153, 155, 156]. Smith et al. [157] proposed a total of nine different metrics divided between detection and tracking purposes. Black et al. [141] included a metric for the evaluation of occlusions (Occlusion Success Rate) which is very convenient for multi-target systems.

Additionally there are some authors that define periods of time or even objects that will not be considered for evaluation since they are out of the scope of the algorithm purpose. For example the segmentation of individuals within a group or during occlusions as in the work of Kasturi et al. [154] where they define the so-called "Don't care frames" and "Don't care objects".

It is worth mentioning the study conducted by Milan et al. [138] where it was noted the variability of results obtained with different implementations of the same set of metrics. The authors also stated that for assuring fair comparisons between tracking algorithms they all should use the same object segmentation module since tracking algorithms rely heavily on the performance of the segmentations.

2.4.3 Evaluation parameters

Special consideration should be given to the evaluation parameters which affect dramatically the measured performance. For detection and tracking algorithms these parameters are used for establishing the required mapping between the ground truth and the output of the algorithm – see figure 2.4.

The evaluation parameters for detection algorithms could refer to the spatial threshold used to define the mapping with the ground truth, which is normally computed based on the spatial overlapping between bounding boxes [157–159] or the Euclidean distance between centroids. For tracking algorithms a double threshold is usually required to account for the spatial and temporal overlapping of tracks [155, 156].

Similarly to the data association stage during the actual tracking, the ground truth mapping can allow multiple mappings with one ground truth object [156, 157, 160] or

[5]Diagram copy from the original work of Lazarevic-McManus et al. [152] with the consent of the authors.

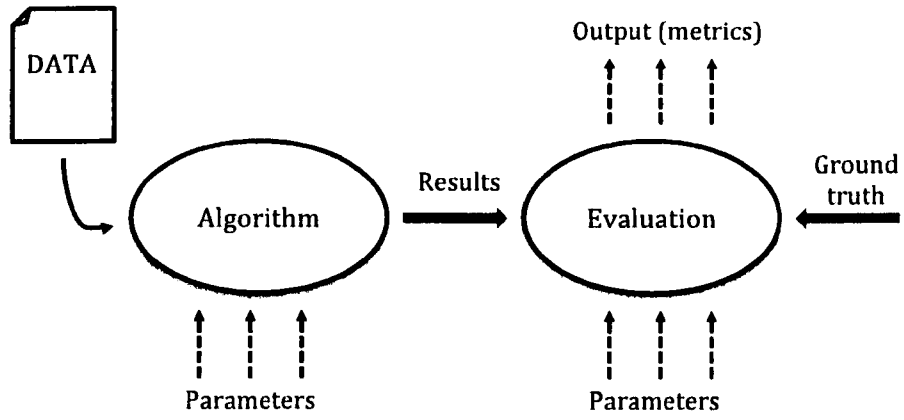


Figure 2.4: Performance evaluation[5].

only permit single mappings using methodologies such as the nearest neighbour [160] or the Hungarian algorithm [153, 161, 162].

Ideally, the evaluation parameters should not favour any technique with respect to others. Some authors propose to include the evaluation parameters in the optimization process of a few well known algorithm and use those as standard values for the evaluation of the rest of algorithms [152].

Chapter 3

Multi RGB-D Sensor Monitoring System

3.1 Introduction

RGB-D cameras are sensors that produce intensity images as well as depth data. These sensors are widely used in the computer vision community since Microsoft released the *Kinect*© camera in November 2010. This sensor was a revolution as it provides reasonably accurate depth data at an affordable price. Afterwards similar sensors were developed such as the Xtion Pro Live camera or the second version of Kinect released in July 2014. In particular, for video surveillance applications the use of RGB-D cameras brings at least two major benefits with respect to classical intensity-based systems in the visual surveillance context: they are robust to illumination changes even allowing the monitoring of dark environments; and the effectiveness for identifying and solving occlusions, which are considered nowadays one of the major challenges in video surveillance.

In this work a multi-sensor device built from non-overlapping RGB-D cameras will be proposed. The system is intended for monitoring wide indoor spaces which maximises the area covered. In order to attain an optimal design an analysis of the capabilities and limitations of the RGB-D sensor will be conducted. The system will need to be calibrated to allow the data from all the RGB-D cameras to be represented in a common coordinate system. The calibration of non-overlapping sensors is a challenging problem and some approaches have been proposed in the past to solve it. However, they tend to be highly complex such as the one proposed by Anjum N. et al. [133] based on trajectory estimation of moving objects during the unobserved regions, or the mirror based method introduced by Kumar R. K. et al. [135] were the sensors see the calibration grid through a mirror. Another approach that requires the relative location of the cameras to be fixed was presented by Lébraly P. et al. [163] where

the calibration parameters are computed by manoeuvring the system through a static scene and estimating the trajectories of the cameras. In this work a novel and simple plane-based calibration methodology is proposed for the calibration of non-overlapping RGB-D sensors.

The remainder of this chapter is organized as follows: In section 3.2 a detailed analysis of the Kinect sensor is presented along the proposed configuration of the combined device. The methodology employed for the calibration of the non-overlapping cameras is described in section 3.3. In section 3.4 the potential issues of the system are identified, and finally section 3.5 provides some conclusions for the chapter.

3.2 System geometry and design

In this section the configuration of a multi-sensor device proposed for surveillance and monitoring purposes is presented. This device is composed of three RGB-D sensors, namely Microsoft *Kinect*© cameras, which are set in a non-overlapping fashion to maximize the covered area and minimize the interferences produced between sensors. The use of the Kinect sensor brings many advantages to surveillance applications especially for detecting and solving occlusions. However, these sensors have limited range and can only be used in indoor environments. In addition, the depth accuracy and resolution decrease with distance. A comprehensive analysis of these issues is required in order to take appropriate measures and obtain the best performance of the sensors.

3.2.1 RGB-D sensors: The Microsoft Kinect

Microsoft's *Kinect*© sensor[1] is a laser-based depth sensor which generates a depth image enabling the 3D locations of points within a room to be located as well as the colour information about these points – essentially a 3D camera.

The affordable price of this camera in comparison with other range sensors has revolutionized the research community. In particular in surveillance applications it has become very popular since it addresses the main challenges of classical intensity-based systems: occlusions and illumination changes.

3.2.1.1 Kinect device: Capabilities

The Kinect device features an infra-red (IR) projector and a monochrome CMOS camera with an IR-pass filter that produces images at approximately 30 frames per second. The original resolution of the sensor is 1280×960 pixels which is downsampled to 640×480 pixels due to limitations on the USB bandwidth. The spatial resolution at 2 m is 3 mm

[1]In this work the Microsoft's *Kinect*© sensor refers to the first version of the sensor designed for the Xbox.

in the horizontal and vertical axes and 1 cm along the depth dimension. The field of view is 58 and 45 degrees on the horizontal and vertical axis respectively. Although the operating range goes from 0.5 to 4 metres, it produces depth data up to 9.7 metres. Further details can be found in the studies produced by Khoshelham and Elberink [164] and Andersen et al. [165].

Additionally, the Kinect device also has a RGB camera and a multi-array microphone. The RGB camera delivers the three basic color components (red, green and blue) at a frequency of 30 Hz with a resolution of 640 × 480 pixels. Regarding the multi-array microphone, it allows voices to be localized in the 3D space and ambient noise to be rejected.

3.2.1.2 Depth estimation

The IR projector sends out a fixed pattern of light and dark speckles (figure 3.1) and depth is calculated by triangulation using a reference pre-recorded IR pattern at a known distance. It works as a structured light sensor. The depth of each IR speckle is estimated based on their displacement with respect to their corresponding point in the pre-loaded pattern. At this stage an operation of cross-correlation is performed between the current and recorded pattern to yield a map of disparities. In figure 3.2[2] the depth z_k of the point k is calculated based on the disparity d between the projection onto the image plane of k, and its corresponding point in the pre-loaded pattern O, using similar triangles as follows:

$$\frac{D}{b} = \frac{z_0 - z_k}{z_0}$$

$$\frac{d}{f} = \frac{D}{z_k}$$

where z_k can be obtained:

$$z_k = \frac{z_0}{1 + \frac{z_0}{fb}d} \tag{3.1}$$

These sensors bring many advantages to numerous computer vision applications. However it is important to be aware of their limitations in order to maximize their performance. They cannot be used in outdoor environments as the sunlight interferes with the IR pattern. The same situation occurs when used in combination with others IR sensors [167]. In addition, they have limited range and resolution especially beyond the operating range. Due to the nature of any sensor based on triangulation, the error increases with distance. The next section outlines some of these issues in order to

[2]Image taken from the work of Khoshelham [166].

Figure 3.1: Infra-red image from Kinect sensor.

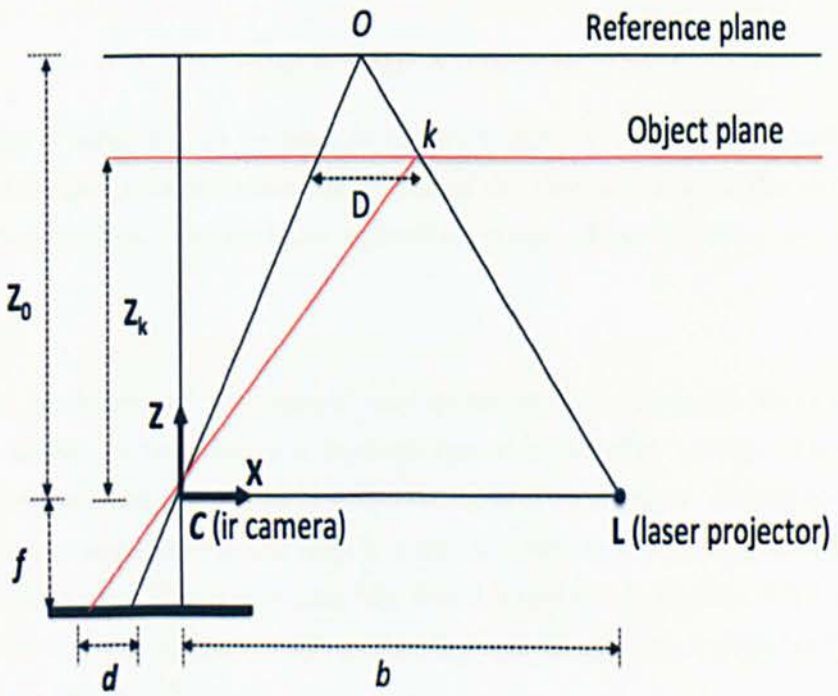


Figure 3.2: Kinect triangulation.

produce a suitable configuration of the multi-sensor device envisaged in this work.

3.2.1.3 Kinect sensor: Accuracy analysis

A theoretical analysis regarding the Kinect error and resolution can be found in the literature in different studies [164, 165, 168]. A series of experiments have been conducted to validate these analyses.

Resolution

The resolution of a depth sensor can be defined as the minimum gap between contiguous depth values, which is produced during the quantization process. Due to the nature of the sensor this gap increases with distance, where larger gaps indicates lower resolutions. To evaluate the actual resolution of the sensor a simple experiment was performed. Samples from a plane perpendicular to the camera axis were taken at different distances. Specifically, the Kinect was mounted on top of a wheeled trolley and the plane was recorded while the trolley was pushed away from it. The range of recording went from 0.5 metres to 4.5 metres – see figure 3.3. To actually calculate the depth resolution, at each distance the values collected from the plane were sorted according to their depth values, and the minimum difference between two adjacent values was taken as the resolution at that distance. Equation 3.2 defines a quadratic function that models the depth resolution of the sensor.

$$\xi(d) = 2.6d^2 + 0.6d - 0.2 \tag{3.2}$$

This model is intended to be used in future stages of this work in particular during the segmentation stage to mitigate the effects of the degradation on the depth resolution and allow segmentations beyond the operating range of the sensor – see section 4.4.

Depth error

Due to effects like blurring[3], pixelation[4] and quantization[5] the depth value of a particular point in the scene varies within a certain range of nearby values. This is normally considered the residual error and is affected by the resolution. Using the same set up as in the previous experiment the depth error is computed with the standard deviation at different distances. The errors and the fitted function (equation 3.3) are plotted in figure 3.4. The increasing gaps between contiguous values affect directly this error as it is illustrated in figure 3.5.

[3]Blurring appears when the light ray of one point affects more than one pixel on the sensor.

[4]Pixelization is produced when considering that the projection of a point in the image lays on the centre of the pixel of the sensor.

[5]Quantization refers to the process that converts the continuous signal capture by the sensor to discrete values.

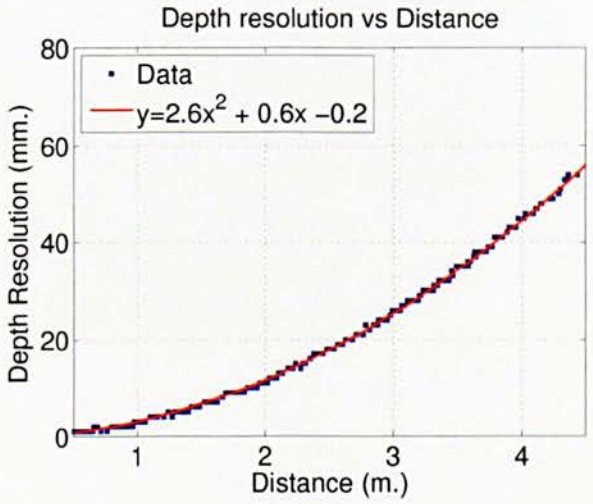


Figure 3.3: Depth resolution of the Kinect with respect to distance.

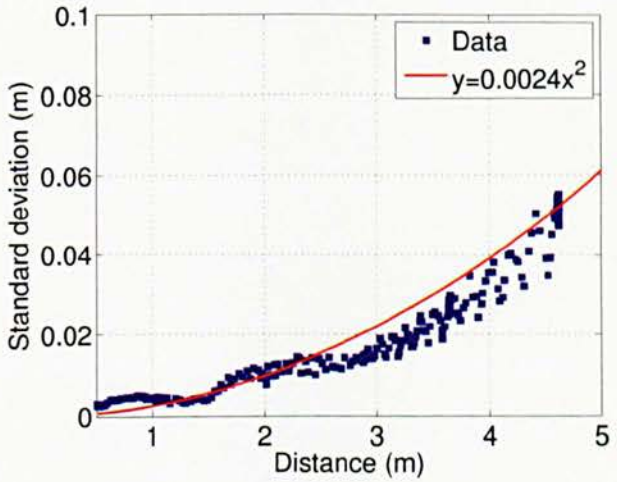


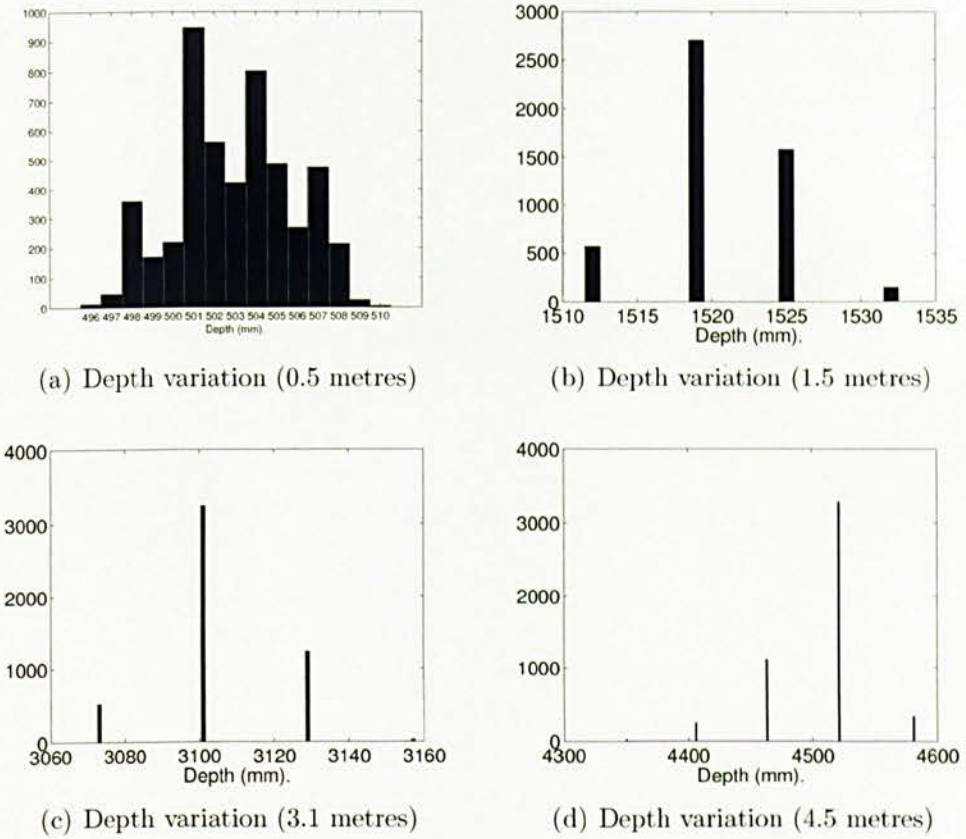
Figure 3.4: Depth error (standard deviation of samples from the same plane).

$$\sigma(d) = 0.0024d^2 \tag{3.3}$$

This model will be applied in this work during a background subtraction process in a future stage to define a depth-based threshold to differentiate foreground pixels. – see section 4.2.1.1.

3.2.2 Proposed design

A device that combines three Kinect sensors in a non-overlapping configuration is proposed as shown in figure 3.6. The benefits of this design are two fold. First it allows wider areas to be monitored since the field of view (FOV) of the overall device is the aggregation of the FOVs of the three sensors. Sceond, this configuration avoids the possible IR interferences between sensors. These interferences have been identified in



(a) Depth variation (0.5 metres)

(b) Depth variation (1.5 metres)

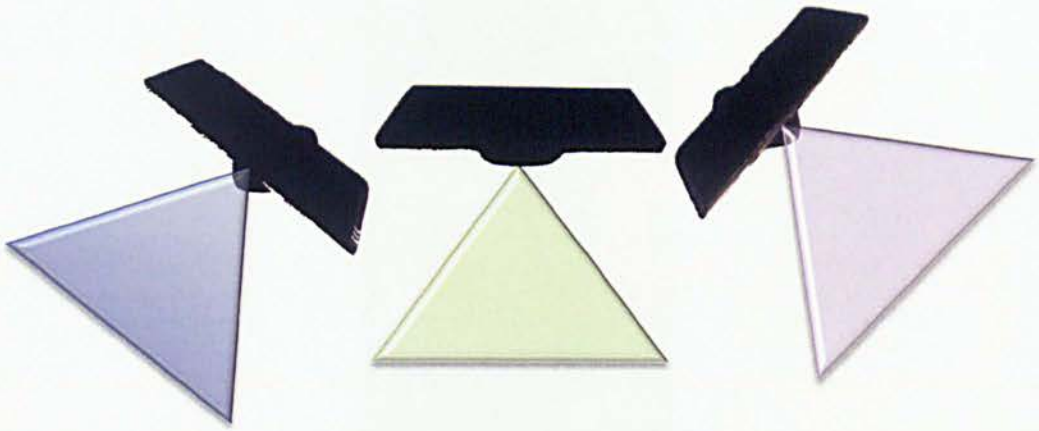
(c) Depth variation (3.1 metres)

(d) Depth variation (4.5 metres)

Figure 3.5: Distribution of depth values from a plane capture at four different distances.

previous works [121, 167] and are produced when several Kinects project simultaneously their IR pattern onto the same region of the scene leading to failures during the point correlation stage.

(a) Front view



(b) Top view

Figure 3.6: Design of the multi Kinect device

The device is mounted on top of a tripod at approximately 2.20 metres high as shown in figure 3.7. The main reason for locating the device in a high location is to minimize the number of occlusions, both static and dynamic. The approximate area covered by the device is 220 m^2, limited in depth by the Kinect range. Note that there exists a blind spot just below the device, which depends on the tilt angle of the cameras.

3.2.3 Issues

The main challenge of the proposed configuration is the external calibration of non-overlapping sensors, which requires the estimation of the geometric transformations between the sensors coordinate systems (CSs) and a common CS. As opposed to systems with overlapping cameras FOVs, in this configuration standard calibration approaches based on corresponding points cannot be employed [119, 123]. In the next section a novel plane-based calibration procedure for non-overlapping range sensors is described.

3.3 System calibration

In multi-camera systems a proper external calibration between sensors is essential in order to manage the data efficiently. This process aims to represent the data from the three Kinects in one reference CS.

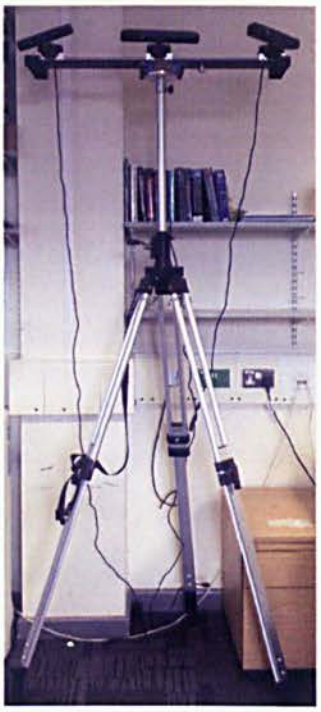


Figure 3.7: Tripod and camera mounting set up.

The data captured by each Kinect sensor generates 3D positions within some local CS. Calibration between two sensors produces the geometric transformation (rotation and translation) between the CSs of both sensors. For simplicity in this work the reference CS of the combined device was established on the middle Kinect CS and therefore, only two calibrations are required as shown in figure 3.8.

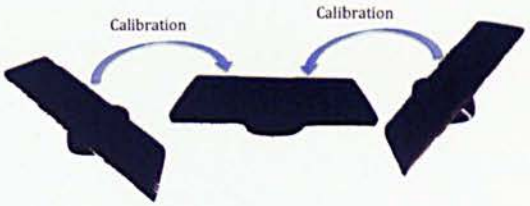


Figure 3.8: Multi calibration required for the whole device

In order to estimate the transformation between two sensors, common features are required. Generally, stereo calibration techniques use a set of corresponding common points in both sensors and obtain the best transformation by error minimization techniques [169]. However, in the configuration proposed, the physical set up of the sensors does not allow the use of corresponding points as their FOVs do not overlap.

A novel calibration technique has been developed to enable the 3D data from different devices to be represented within one CS for the whole monitoring space. The calibration technique presented exploits the depth capability of the Kinect by using planes as common features.

3.3.1 Plane detection

Planes are used for the calibration process and thus must be identified and detected in the images. A plane is detected by a fitting process using a set of 3D points, in a way that the distance from the points to the plane is minimized i.e the plane that best fits the set of points given as shown in figure 3.9.

Figure 3.9: Plane fitting

The fitting process uses the plane equation $Ax + By + C = z$ which can be represented with matrices as follows:

$$\begin{pmatrix} x & y & 1 \end{pmatrix} \cdot \begin{pmatrix} A \\ B \\ C \end{pmatrix} = z \tag{3.4}$$

If equation 3.4 is generalized for a set of points, the following system is obtained:

$$\begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{pmatrix} \cdot \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}$$

which is a system of the form $M\underline{a} = \underline{z}$, where $\underline{a}$ is the plane coefficient matrix $[A, B, C]^T$ and can be calculated as follows $\underline{a} = M^{-1}\underline{z}$.

As described, planes are detected in images from a set of 3D points, which are easily obtained using the depth information provided by the Kinect sensor.

3.3.2 Rotation estimation

The rotation between a pair of Kinects is estimated by using the normal vectors of corresponding planes[6]. The process of estimating the rotation is based on the idea that the transformation between two normal vectors can be represented by a rotation – see figure 3.10. At this point it is important to recall the alternative plane representation $n_x x + n_y y + n_z z = d$, where $[n_x, n_y, n_z]$ is the normal of the plane, $[x, y, z]^T$ is a point in the plane, and d the distance of the plane to the origin of the CS. Therefore, it is required to compute the normal coordinates from the plane coefficients $[A, B, C]$ obtained during the plane detection process. Note that the plane equation $Ax + By + C = z$ is derived from the normal plane equation $n_x x + n_y y + n_z z = d$ as follows:

$$
\begin{aligned}
n_x x + n_y y + n_z z &= d \\
n_x x + n_y y - d &= -n_z z \\
-\frac{n_x}{n_z} x - \frac{n_y}{n_z} y + \frac{d}{n_z} &= z
\end{aligned}
$$

where

$$A = -\frac{n_x}{n_z} \tag{3.5}$$

$$B = -\frac{n_y}{n_z} \tag{3.6}$$

$$C = \frac{d}{n_z} \tag{3.7}$$

In order to recover $[n_x, n_y, n_z]$ from $[A, B, C]$ the normal vector is constrained to be a unit vector, i.e. $\sqrt{n_x^2 + n_y^2 + n_z^2} = 1$. Finally using this constraint and equations 3.5 and 3.6 the normal vector of the plane is calculated as $\hat{n} = [A, B, -1]$. Note that this is not a unit vector.

[6]The expression "corresponding planes" denotes in this context the same plane represented in different CSs

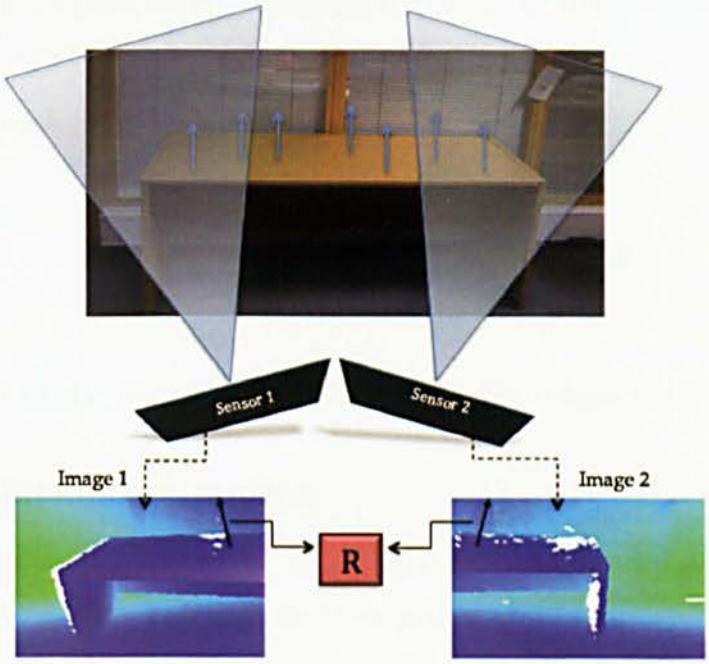


Figure 3.10: Corresponding normal vectors

The rotation between two CSs in three dimensions is represented by a square 3x3 matrix R which must meet the following properties:

- $R^T = R^{-1}$ (i.e. R is orthogonal).
- $det(R) = 1$.
- $||R_{i:3}|| = 1$, where i denotes a column of R (i.e. the columns of R are unit vectors).

The rotation matrix is calculated following the method described by Sorkine [169] which guarantees that all these properties are hold. The steps to calculate the rotation matrix using a set of corresponding normals are summarized as follows:

1. Organize the corresponding normals in two matrices (a matrix for each camera)

$$N_1 = \begin{pmatrix} n_{x,1} & n_{x,2} & \cdots & n_{x,m} \\ n_{y,1} & n_{y,2} & \cdots & n_{y,m} \\ n_{z,1} & n_{z,2} & \cdots & n_{z,m} \end{pmatrix}$$

$$N_2 = \begin{pmatrix} n'_{x,1} & n'_{x,2} & \cdots & n'_{x,m} \\ n'_{y,1} & n'_{y,2} & \cdots & n'_{y,m} \\ n'_{z,1} & n'_{z,2} & \cdots & n'_{z,m} \end{pmatrix}$$

where m is the number of corresponding normals.

2. Calculate the singular value decomposition (SVD) of the product of both matrices $SVD(N_1 N_2^T) = U\Sigma V^T$.

3. Obtain the rotation matrix as follows:

$$R = V \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det(VU^T) \end{pmatrix} \cdot U^T$$

Further details of the technique can be found in the original paper [169].

3.3.3 Translation estimation

Based on the rotation obtained, the translation is estimated by error minimization using a set of corresponding points. How these points are obtained is the key innovation of the method.

For a plane detected in the first CS, a unique point can be identified as the point on the plane closest to the origin, i.e. $\underline{x} = d\hat{\underline{n}}$. This point undergoes an as yet unknown translation $\underline{x}' = d\hat{\underline{n}} + \underline{t}$, to be represented in the second CS. A graphical model is shown in figure 3.11.

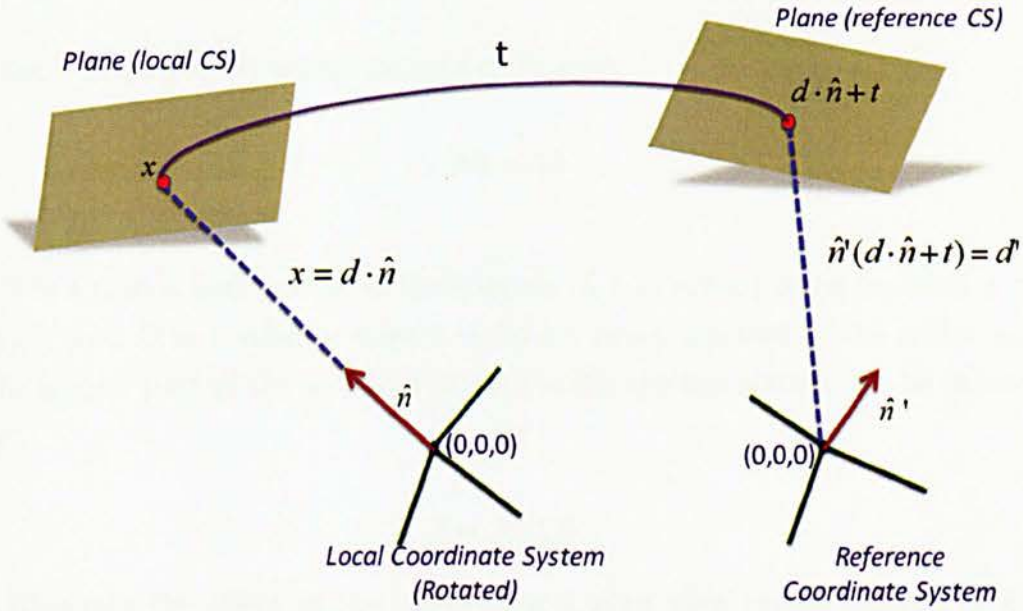


Figure 3.11: Translation estimation using a pair of corresponding point

Since this translated point must lie on the second plane, a constraint on the translation t can be obtained as follows:

$$\hat{\underline{n}}' \left(d\hat{\underline{n}} + \underline{t}\right) = d'$$

$$\hat{\underline{n}}' \cdot \underline{t} = d' - d\,(\hat{\underline{n}}' \cdot \hat{\underline{n}}) \tag{3.8}$$

where d and d' are the distances of the plane to both CS origins (local and reference), $\hat{\underline{n}}$ and $\hat{\underline{n}}'$ denote the normal vectors of the plane in both CSs, and $\underline{t}$ represents the translation vector $[t_x, t_y, t_z]^T$ between the two CSs. Equation (3.8) can be generalized for every plane as follows:

$$\hat{\underline{n}}'_i \cdot \underline{t} = d'_i - d_i\,(\hat{\underline{n}}'_i \cdot \hat{\underline{n}}_i)$$

Therefore, the following system of equations is obtained for m planes:

$$\begin{aligned}
\hat{\underline{n}}'_1 \cdot \underline{t} &= d'_1 - d_1\,(\hat{\underline{n}}'_1 \cdot \hat{\underline{n}}_1) \\
\hat{\underline{n}}'_2 \cdot \underline{t} &= d'_2 - d_2\,(\hat{\underline{n}}'_2 \cdot \hat{\underline{n}}_2) \\
&\vdots \\
\hat{\underline{n}}'_m \cdot \underline{t} &= d'_m - d_m\,(\hat{\underline{n}}'_m \cdot \hat{\underline{n}}_m)
\end{aligned}$$

which can be represented using matrices as follows:

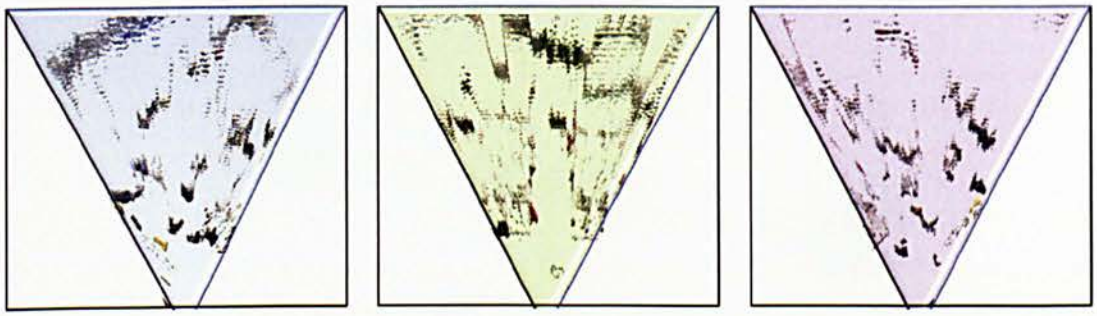
$$N\underline{t} = D \tag{3.9}$$

where N is a matrix that groups all the normals $n'_{1:m}$ in rows, $\underline{t}$ is the translation matrix $[t_x, t_y, t_z]^T$, and D is a column matrix in which every position is the scalar resulted from the second part of the equation (3.8). Finally the translation can be obtained as follows:

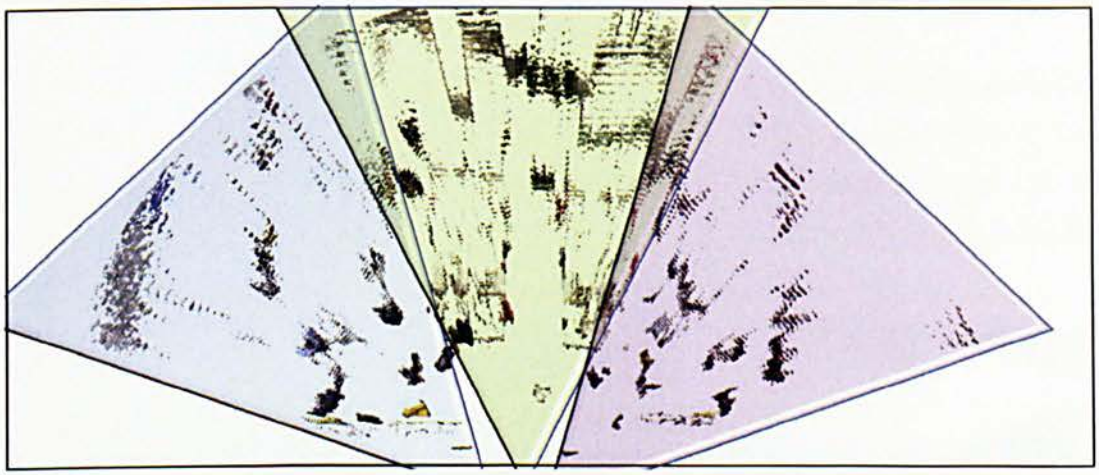
$$\underline{t} = N^{-1}D \tag{3.10}$$

To illustrate the effect of the calibration a plan view representation of a scene captured from the three Kinects is presented before and after calibration in figure 3.12.

(a) RGB images from the three Kinects (left, middle and right)



(b) Plan view images from the three Kinects (left, middle and right)



(c) Calibrated plan view image from the three Kinects

Figure 3.12: Calibration of points from the three Kinects into a common representation (plan view).

3.3.4 Calibration tool

A calibration tool has been built in order to create as many corresponding planes as required.

The estimation of the rotation and translation is based on error minimization. The rotation estimation uses corresponding normal vectors of planes and to estimate the translation corresponding points in the planes are used. The accuracy of these techniques depends mainly upon the number of corresponding features used so that the more corresponding planes, the more accurate the calibration will be. However,

frequent scenarios do not contain many common planes (e.g. floor, ceiling, walls, tables), exhibiting at best 4 or 5 common planes.

In order to increase the number of common planes, and therefore improve the accuracy, a calibration tool has been built. This tool allows the creation of common planes between a pair of cameras. The tool consists of a pole of 1.7 m length, with two boards of 32 × 18 cm attached at both edges of the pole in a way that both boards belong to the same plane – see figure 3.13. The distance between their centroids is 1687 mm.

Figure 3.13: Calibration tool for creating corresponding planes – the "paddle".

The idea is that each board is viewed and detected in a separate camera at the same time. Therefore, a pair of corresponding planes is obtained as shown in figure 3.14, in which the normal of each plane is represented with a red line and the area of the detected plane with a white rectangle (the blue rectangle denotes an initialization area).

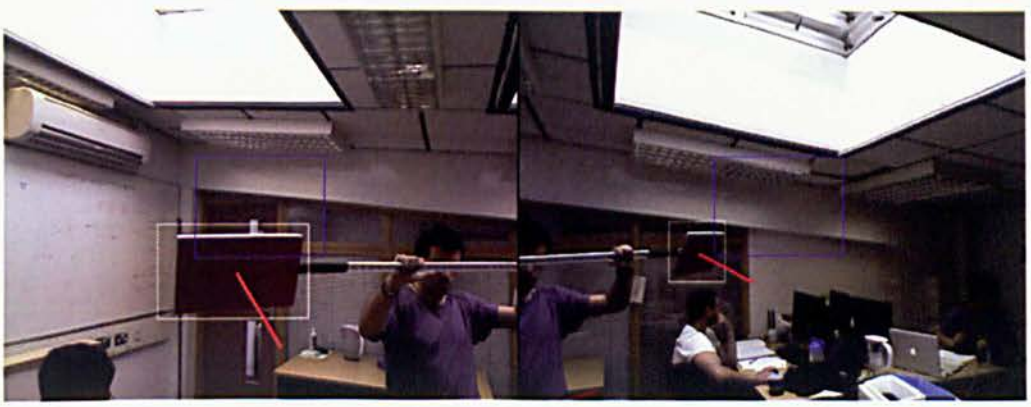


Figure 3.14: Pair of corresponding planes detected in the calibration process

The calibration procedure consists of holding the paddle in front of the two sensors, allowing each board to be viewed by a different sensor. An initialization volume is defined in the centre of the field of view at two metres from the camera as illustrated in figure 3.15. A colour filtered is applied to remove data that does not belong to the board. The remaining data is fitted into a plane using equation 3.4. The planes in the subsequent frames are obtained by considering a neighbour volume around the fitted plane in the previous frame.

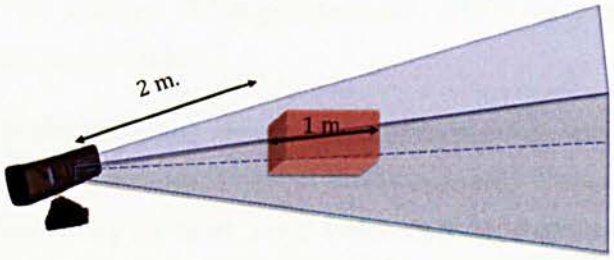


Figure 3.15: Frustum of the Kinect depth sensor. The plane initialization volume is red shaded

3.4 Issues

The calibration of the system does not only refer to the external transformations between cameras. There are two additional calibrations that need to be considered: the internal calibration of each camera to compute the intrinsic parameters i.e. focal length(f), principal point (i_0, j_0) and lens distortion coefficients $(k_1, k_2, ...)$; and the calibration that models the transformation between the IR camera and the RGB camera, which is required to align the two images.

This work uses the default parameters provided by the framework OpenNI[7] [170]. The rigid transformation between the RGB and the IR camera unfortunately are not available as they are encrypted in the code. The default intrinsic parameters are summarized in table 3.1.

f	574 (pixels)
j_0	320
i_0	240

Table 3.1: Kinect depth sensor default intrinsic parameters.

These parameters are reasonably accurate and for the purpose of this work they are acceptable. Although the lens distortion model is not considered, the Kinect features low-distortion lenses ($|k_1| \approx 0.1$) and even at the edges of the image the displacement is not more than a few pixels [171]. However, if more accurate results were required (e.g. action recognition applications) a manual calibration should be performed [118, 172].

In order to obtain accurate calibration results it is important to ensure that every pair of corresponding planes are coplanar. Within this context, two sources of errors were identified associated to the plane detection stage.

- Asynchronization of views: The paddle orientation changes during the moment of

[7]The framework OpenNI is not longer available since it was acquired by Apple Inc.

capture of the two sensors. This problematic situation is mitigated by moving the paddle slowly and steady.

- Misalignment of the paddle: Small errors during the assembly of the paddle produce differences in the orientation of the two boards. In addition, environmental issues such as humidity or heat yield little misalignments of the boards. This situation is addressed by estimating the angle between the two boards in a previous stage and correcting that error angle during the actual capture of the planes.

3.5 Discussion

A multi-sensor device built from non-overlapping RGB-D cameras, namely the Microsoft *Kinect*© sensor, is proposed for monitoring wide area indoor spaces.

The Kinect sensor has been independently analysed in terms of resolution, range and noise, with the objective of maximising the efficiency of the combined device. The proposed design aims to maximise the area covered as well as minimize the interferences between sensors.

For the calibration of the system a novel plane-based procedure is presented for non-overlapping range sensors that allows the data from the three Kinects to be represented in a common CS. The proposed calibration methodology uses corresponding planes to derive constraints on rotation and translation, in particular the rotation is computed from the normal vectors and the translation by using a special point in the planes i.e. the closer point to the origin of the reference CS. A calibration tool was presented to allow the generation of many corresponding planes between a pair of adjacent non-overlapping cameras. Using a plane fitting approach planes were effectively extracted from the range data.

The internal calibration of the sensors and the estimation of the transformation between the IR and RGB cameras was not necessary as the default parameters provided by the framework were accurate enough for the purpose of the system. However, if more accurate results are required a manual calibration should be performed.

Chapter 4

People Segmentation

4.1 Introduction

Segmenting people in video sequences is used in a wide range of applications such as in robotic environments, intelligent cars or for counting people purposes. It is also a key component in higher level systems such as tracking or activity recognition applications.

The segmentation of people is normally implemented on the image plane or intensity image produced by conventional RGB cameras [103, 103, 173]. Many issues are associated with this space such as cluttered backgrounds, illumination changes, shadows and occlusions, which make this space very challenging to work with.

In the past significant effort was employed in the development of sophisticated algorithms that try to overcome these problems [12, 17, 18, 174]. However, in recent years, with the incorporation of alternative sensors and configurations the effort is focused on the exploration of alternative spaces that minimize or eliminate the effect of these issues. For example, the use of multiple cameras aids in managing occlusions [5, 79, 175, 176]. Alternatively, more advanced sensors such as lasers or RGB-D cameras have allowed researchers to investigate the depth dimension and the 3D space. [4, 36, 175, 177–179]

The main objective of this chapter is to propose and evaluate alternative spaces in the context of people segmentation. Three different spaces are presented in this work. First, the typical image plane enhanced with depth to aid in the identification and resolution of occlusions. This space is referred in this work as the Image Plane Space (IPS). Second, a space built over the ground plane that aggregates the data from the three cameras that form the system and is named the Map of Activity (MoA). Third a space constructed over the polar coordinate system (CS).

The remainder of the chapter is organised as follows. In the next section the Image Plane Space along with the segmentation process in this space are explained. In section 4.3, the Map of Activity and how it is built from the aggregation of data from the

three sensors is described. In addition, the main limitations of this space are identified. Section 4.4 offers a comprehensive analysis of the Remapped Polar Space and the segmentation methodology employed. Finally, in section 4.5 the different spaces are evaluated and discussed.

4.2 Image Plane Space

The Image Plane Space (IPS), as defined in this work, refers to the two dimensional digital image returned by the sensor. The IPS is a discrete space where each position i.e. pixel, is identified by the horizontal and vertical coordinates with respect to the origin, which is usually located at the top-left corner of the image. For depth sensors each pixel stores a depth value instead of a value of intensity.

Depth information is a powerful feature to use for segmenting objects. Unlike intensity data, depth is robust to illumination changes, shadows and clutter backgrounds. Nevertheless, the use of depth data has some limitations associated related to resolution and noise that need to be considered in order to get optimal results.

Most of the IPS segmentation techniques proposed in the literature are conceived to be used with intensity images although they can be extrapolated to depth. In this work an approach based on foreground segmentation with depth data is followed for detecting people. This segmentation is applied independently on each depth IPS of the three sensors, and requires a final process to fuse the results into a common representation.

This section describes in detail the proposed technique for segmenting people in the depth IPS. In addition the process for aggregating the results into a common view is presented. Next, the critical issues related to the process and the use of depth data are identified and analysed. Finally, the special measures taken to minimize the effect of these issues are described.

4.2.1 People segmentation

The technique presented in this work for segmenting people in the IPS comprises of the following two stages:

1. Foreground segmentation. In this stage moving pixels are detected in each IPS independently applying a background subtraction technique.

2. People detection. Foreground pixels are grouped in connected components, which first, are filtered to remove noisy components and then analysed to detect occluded people.

4.2.1.1 Foreground segmentation

A very well known technique for segmenting foreground objects in video sequences is background subtraction. Its simplicity and high computational speed make it very popular among researchers within the video surveillance community. The foreground points of a given image are obtained by performing a pixel-wise comparison with a background model of the scene. Those pixels that differ more than a certain threshold with respect to this background are labelled as foreground.

In the literature a huge variety of background subtraction techniques can be found. The majority of these are designed to be used with intensity images, e.g. RGB or grayscale. Important issues that must be considered in these scenarios are shadows, illumination changes and cluttered background. To deal with these issues, sophisticated modifications of the basic technique have been proposed such as the Gaussian Mixture Model [12] or the Kernel Density Estimation [13].

The use of depth data presents at this stage a major advantage with respect to intensity data. Depth is robust to all the aforementioned critical issues, and therefore the use of sophisticated techniques to deal with them is not necessary. High performances in detecting foreground objects using depth are achieved by using basic background subtractions techniques [4, 36] .

The proposed approach in this work uses the depth data captured by the Kinect sensor to perform a basic background subtraction. The implemented algorithm described below is composed of three sub-stages: background modelling, model maintenance and foreground labelling.

- Background modelling. At this stage a representation of the background is built. This representation should only contain the static elements of the scene. In order to get an accurate representation, an initialization period of time free-of-people at the beginning of the sequence is required. A pixel-wise model is built using the depth median value from the whole initialization period as proposed by Lo and Velastin [9]. This approach is robust to possible outliers during the initialization period – see section 4.2.2 for related issues.

- Model Maintenance. This stage plays an important role when working with intensity images as illumination changes are common in real situations. On the contrary, depth data is not affected by light variations and therefore, there is no need of gradual updates of the background. A depth model, however may still experience sudden changes when objects of the background are moved or taken out of the scene.

  The per-pixel model maintenance process proposed uses a selective updating rate based on the foreground and background regions obtained at each time step.

$$B_{i,k+1} = \alpha R_{i,k} + (1 - \alpha) B_{i,k} \tag{4.1}$$

where $B_{i,k}$ and $R_{i,k}$ are the i^{th} pixels of the background model and the current depth image at time k respectively, and α is the learning rate, which has different values for background and foreground pixels. The background regions of the model are updated with a slow rate in case background objects are moved since gradual changes are not expected. On the other hand, the setting of the updating rate for foreground regions is more complex and must be analysed in further detail. Two possible situations that generate foreground regions are identified:

- Background movement. When a background object is reallocated within the scene yields two foreground regions; the region where the object used to be, and the region where the object is currently located.
- Foreground movement. For instance, a person walking in the scene. That person generates one foreground region, which belongs to the current position of the person.

Both situations lead to foreground regions, although the former is considered as a false foreground. Each of these requires different updating rates. In the first scenario, it will be desirable to update the model quickly, so the foreground regions become part of the model faster. In the second situation, the updating rate should be low in order to avoid the inclusion of the person in the model. The main problem comes from the difficulty of discriminating these two situations, which requires higher level interpretations of the scene. The updating rate α in this implementation was selected experimentally as a trade off between both situations as follows:

$$\alpha = \begin{cases} 0.05 & \text{, if } I_t \notin \text{ background} \\ 0.005 & \text{, else.} \end{cases}$$

- Foreground labelling. The foreground detection process refers to the method used for discriminating foreground objects from the background. It is the final step and defines the output of the background subtraction – see figure 4.1. The foreground detection is performed pixel-wise using a threshold as follows:

$$|R_{i,k} - B_{i,k}| > \tau(B_{i,k})$$

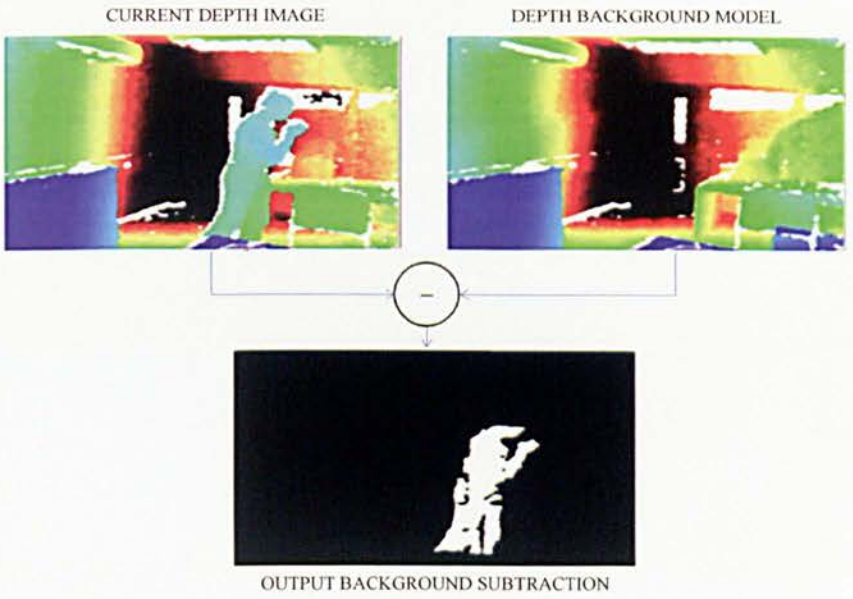


Figure 4.1: Depth-based background subtraction with adaptive threshold

where $\tau(\cdot)$ is a threshold function of depth which has been computed from the analysis of the residual error of the depth sensor described in section 3.2.1.3. Specifically $\tau(\cdot)$ is defined with three standard deviations of the residual error (equation 3.3) covering 99.7 % of the depth variation.

$$\tau(d) = 0.0073d^2 \tag{4.2}$$

There is a possible failure mode when a person is closer to the background than the threshold used. In those cases the person will not be detected. These situations are hard to resolve using uniquely depth data. A possible approximation would be to include extra information such as colour or texture.

The segmented foreground pixels are used in the next step to recover the blobs that represent people.

4.2.1.2 Blob detection

Classical pipeline

The classical procedure for segmenting people after the foreground pixels have been detected is to apply a connected components to group pixels in blobs and then filter out small components assuming they are produced by noise. In general, the identification and analysis of occlusions is deferred to subsequent stages where more information is available i.e. appearance models.

Extended pipeline: occlusion reasoning

Since depth information is available, it is proposed to include and additional module to the pipeline in order to identify and solve occlusions – see diagram 4.2.

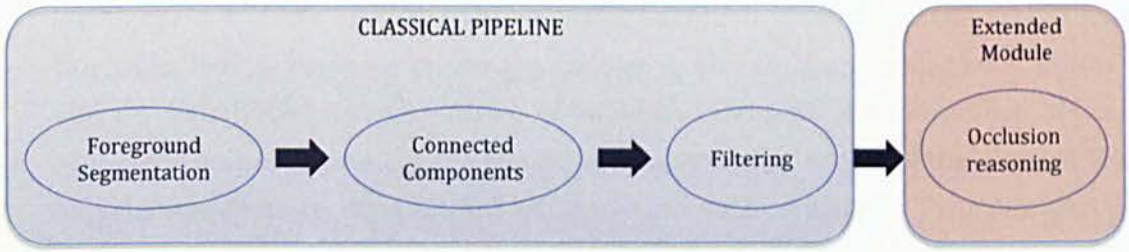
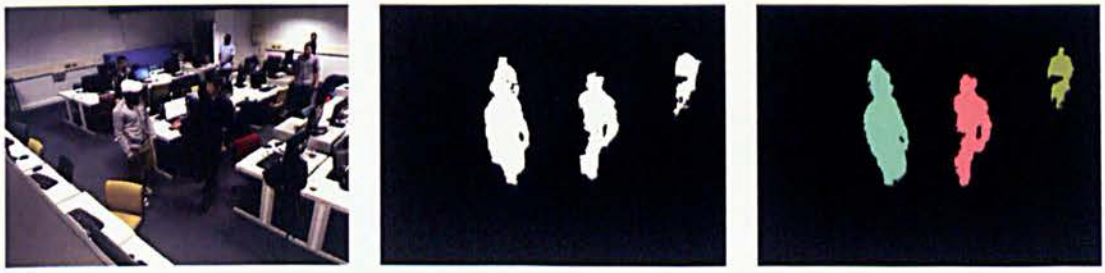


Figure 4.2: Classical people segmentation pipeline with an extended post-processing module to solve occlusions.

The occlusion reasoning module consists of segmenting multiple people projected into a single component (referred here to as a *merged component*). These situations normally occur during occlusions where a more distant person becomes partially occluded by a closer person as illustrated in figure 4.3. Solving *merged components* is one of the most challenging issues for people segmentation, especially when only RGB information is available.



(a) RGB image of a particular instant in a video sequence.

(b) Foreground image. Binary image where the foreground points are represented in white and the background in black.

(c) After the connected components and filtering steps the segmentation produces three blobs.

Figure 4.3: Example of a situation where two people are connected in the image plane in a *merged component*.

In this work, occlusion reasoning is approached by using the depth information of the points in the *merged component*. The intuition is to throw all depth data from the component into a one dimensional histogram expecting the data from different people that are connected in the image plane to become separated in the depth dimension. The process involves the following steps:

1. **Detection of the number of people in the component**. In principle, the number of people included in a *merged component* is unknown. Using the depth

histogram this number can be estimated by counting the number of peaks. In order to identify these peaks, a threshold (equation 4.3) has been set empirically based on training samples, where each sample represents the number of pixels of a detected component at a particular distance i.e. area of the component – see figure 4.4. Note that the threshold is set much lower than the actual fitted function, this is because in the depth histogram all data from the component will be distributed across a range of values. As expected the number of pixels of closer components is higher since they cover larger areas of the camera FOV and decreases in an exponential-like function with distance. Two remarks can be made regarding the plot of figure 4.4. First, there is a significant amount of samples at close distances with lower values than should be expected. This is due to the edge effect where close people are not fully covered by the field of view of the camera. Second, there is a noticeable wide range of values produced by components at the same depth. This is produced by the fact that the samples are obtained from a variety of different components such as *merged components* that produce higher values, or partially occluded components which yields lower values.

$$\tau_{peak}(d) = 3300e^{-0.0006d} \tag{4.3}$$

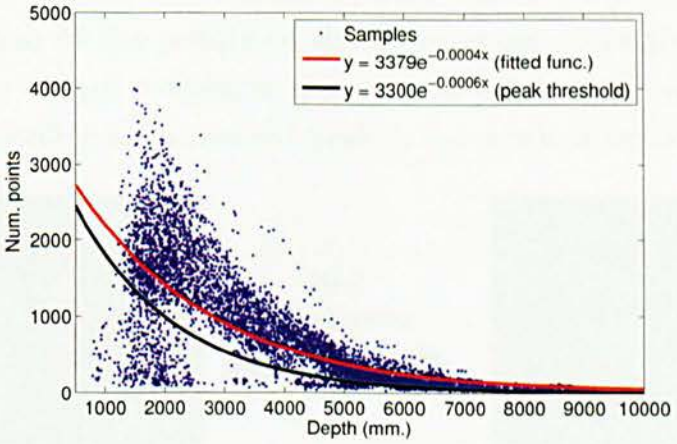


Figure 4.4: Data fitting on a set of training samples where each sample represents the number of points of a component at a particular distance.

Following the example introduced in figure 4.3, the peaks[1] of the three components in the depth dimension are identified as shown in figure 4.5, where one of them is a *merged component* formed by two people (two peaks on the histogram). The detection of peaks only returns the number of people and their approximated

[1]A set of connected bins in the depth histogram that surpass the threshold are referred as one peak

depth position, the physical extent of each person in the component will be computed in the next step.

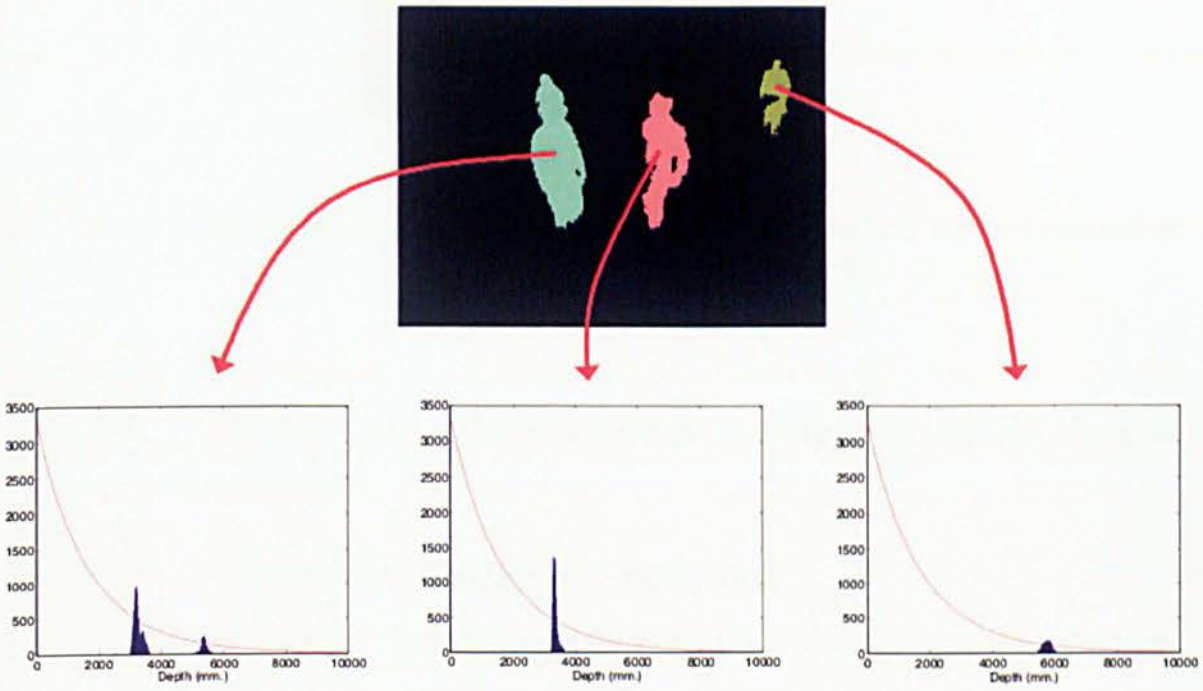


Figure 4.5: Peak detection on the depth dimension. The left histogram shows two peaks, which indicates that the component contains two people.

2. **Classification of the points of the component**. Once the number of people involved in a *merged component* is known, all points in the component are then classified according to the nearest peak in the depth dimension – see figure 4.6.

Figure 4.6: Pixel classification

3. **Filter small components**. The area of the components is calculated and small components are filtered out using equation 4.4 (fitted function from figure 4.4).

$$\tau_{area}(d) = 3300e^{-0.0006d} \tag{4.4}$$

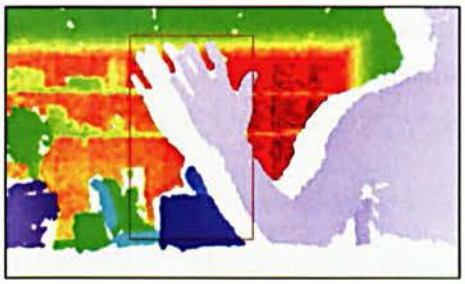


Figure 4.7: Depth image characterized with colours. Null values are represented in white. The red rectangle defines an area with a shadow region

4.2.2 Issues

The following issues associated with the background modelling and occlusion reasoning stages have been identified.

Foreground segmentation: background modelling

When building the model of the background the following problems need to be considered:

Null regions. These are areas where the sensor has not been able to recover any depth information. Three possible reasons are associated with this problem:

- Objects whose texture or colour reflect the IR light with less intensity e.g. black colours reflect the infra-red light with low intensity. If they are far from the camera the resulted noise increases.

- Infra-red interference: Infra-red light from different sources affects the estimation of depth. These sensors calculate the depth based on a correlation of points between the current infra-red image and a pre-recorded pattern as explained in section 3.2.1.2. If infra-red light from a different source interferes in the current image, the correlation of many points will be impossible resulting in null values at those points. For example, they cannot be used outdoors as the sunlight contains infra-red light, or in conjunction with other similar sensors on the same scene.

- Shadow regions. Any object in the scene generates a shadow. The shadow region is larger when the object is closer to the sensor. The reason comes from the fact that the IR camera and the IR projector have different FOVs[2] (see figure 4.7).

[2]The IR camera and the IR projector are separated by a baseline of 7.5 cm approx. There is an area behind any object where the IR light does not reach, as it is blocked by the object itself. However, this area is captured by the IR camera, leading to regions with no depth information or shadows.

Image formation noise. This noise refers to the residual error produced by effects like blurring, pixelation or quantization. A detailed analysis of this error is conducted in section 3.2.1.3 where some experiments were undertaken to model this noise with a quadratic function of depth.

Infra-red laser errors (edges). Minimal variations on the position of the laser projector, illumination factors or even tiny fluctuations on the temperature of the laser lead the IR beam to impact in a slightly different spot. When the impact location is located at the edge of an object these variations cause the laser to impact on a completely different object as illustrated in figure 4.8. This effect results in completely different depth values for a particular pixel.

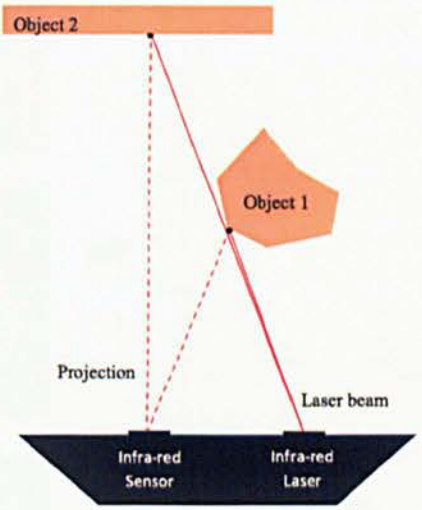


Figure 4.8: Depth error produced at the edges of objects

People detection: occlusion reasoning

Two main reasons of failure have been identified when re-solving occlusions. First, when the people involved in a *merged component* are spatially close, they cannot be discriminated in the depth dimension, they produce one single peak. Figure 4.9 illustrates this situation where a *merged component* is composed of three people, two of them are shaking hands which means they are mostly at the same distance and therefore the system fails to segment them, and the third person who is approximately 2 metres behind, is correctly identified.

The second problem is related to the resolution and noise of the Kinect sensor. At farther distances these factors might lead to one person producing two peaks in the depth histogram. Figure 4.10 illustrates this situation.

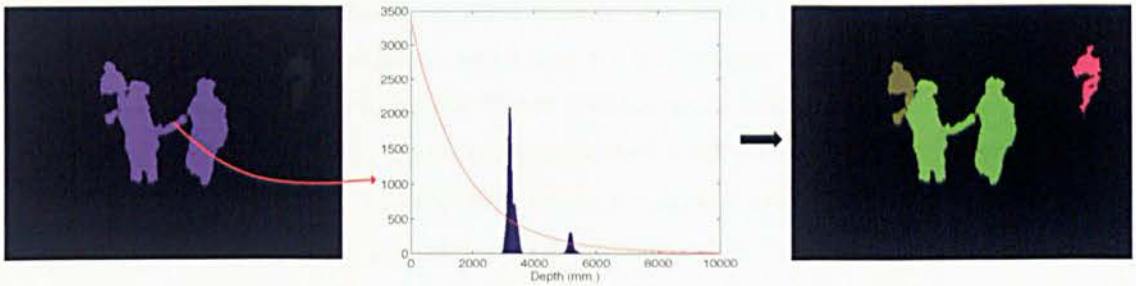


Figure 4.9: *Merged component* formed by three people. Two of them are not distinguished because they are located at the same distance as it is shown in the histogram.

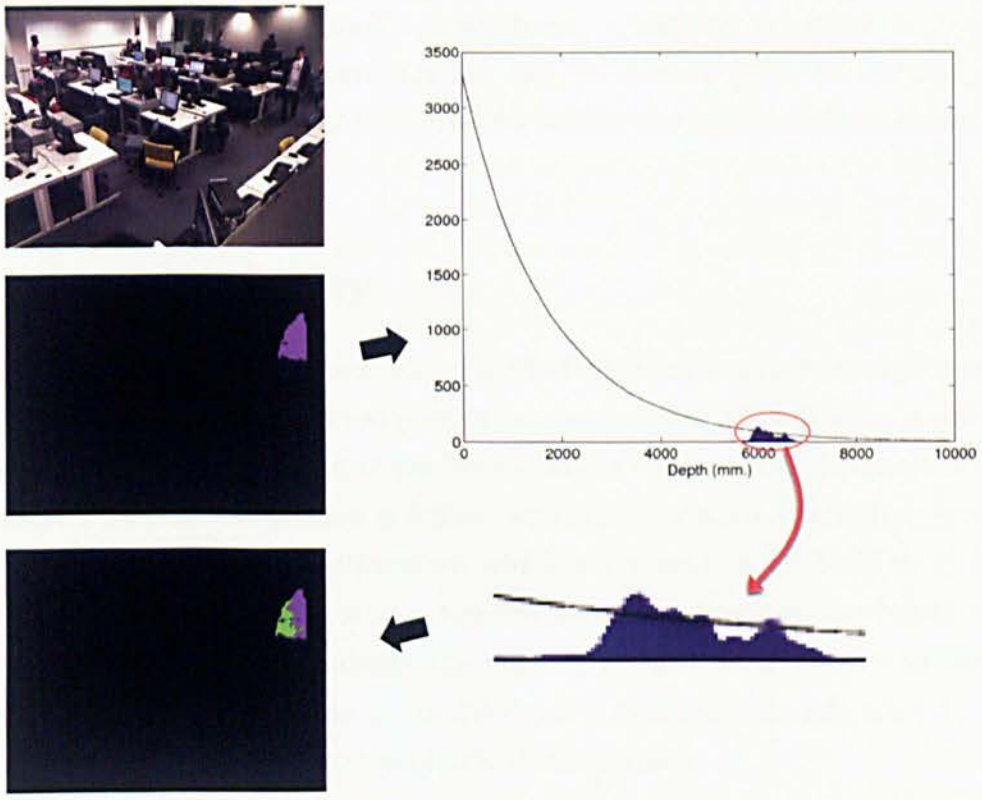


Figure 4.10: Single component which is misinterpreted by the system as being formed by two people.

4.2.3 Discussion

In this section a process for segmenting people using depth data in the IPS for each sensor has been described. This process is based on a foreground segmentation approach followed by a people detection step.

The use of depth information for detecting foreground pixels has many advantages in comparison to intensity data. Shadows, illumination changes and cluttered backgrounds, are all critical issues in intensity images which are avoided by using depth data. However, there are some issues associated with depth sensors that have been identified such as

pixels with noisy depth values, regions where the depth could not be estimated, or the fact that the depth resolution decreases with distance. All these issues have been considered and measures to mitigate these results were taken.

The resulting foreground points are connected and filtered to yield a set of people blobs, one for each camera. In classical intensity-based systems, solving occlusions is one of the biggest challenges i.e. blobs that comprise two or more people. In this work occlusion situations are approached using the depth information. The intuition is that people that are connected in the image plane become well separated along the depth dimension. However, this approach is also associated with some issues. For instance when people are spatially close (e.g. hand shaking or path crossing), depth information is not enough to discriminate them. In addition, occlusions are difficult to solve at farther distances where the resolution is low and the noise is high. In the next section an alternative space is presented that aims to reduce the effect of these issues during the occlusion reasoning step.

4.3 Map of Activity

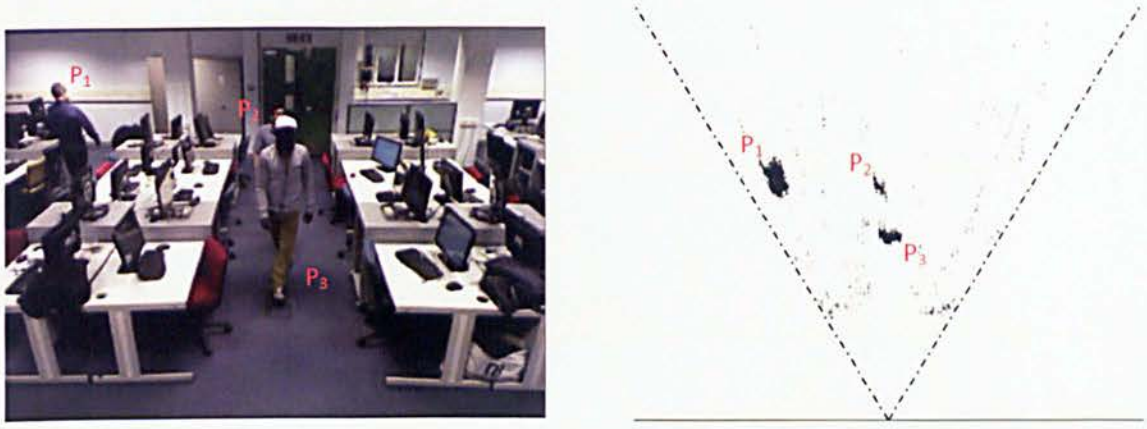
As seen in the previous section, segmenting in the IPS requires a dedicated process based on depth to discriminate connected people i.e. occlusions. This process is associated with some issues when people are at similar distances or at farther distances where the noise is higher and the resolution is lower. In this section an alternative space that handles occlusions naturally is presented, which is referred to as the Map of Activity (MoA). This space can be thought as a top-down view representation where the depth is explicitly represented. To motivate the use of the MoA with respect to the IPS, a visual comparison of an occlusion in the two spaces is displayed in figure 4.11. Unlike the IPS, in the MoA the occlusion is clearly distinguished.

The MoA is built in two steps:

1. Aggregation of data: The foreground pixels from the three cameras are projected into the 3D space, and then transformed (using the calibration parameters) into a single point cloud.

2. Projection of data: The aggregated point cloud is projected orthogonally onto the ground plane where a 2D histogram accumulating the points is built.

In this section, the process for building the MoA is described, and then some relevant issues are identified regarding the people segmentation task in this space.



(a) IPS

(b) Map of Activity

Figure 4.11: Comparison between IPS and MoA with an example of a partial occlusion in both spaces.

4.3.1 Aggregation of data

The aggregation of data is the first step towards the construction of the MoA, where the data from the three sensors is projected into a common CS. This step involves the following two sub-steps:

1. Point back-projection: Using the projective equations 4.5 the foreground pixels obtained in each camera are projected into the three dimensional space. The result is three clouds of 3D points, each of them represented with respect to their camera CS (left-handed CS).

$$X = \frac{Z}{f}(j - j_0), \quad Y = -\frac{Z}{f}(i - i_0) \tag{4.5}$$

   where i and j are the pixel coordinates with respect to the digital image CS, i_0 and j_0 represent the origin of the image plane, X, Y and Z are the 3D coordinates of the points in the space with respect to the camera CS, and f is the focal length of the camera expressed in pixels.

2. Point cloud fusion: The three point clouds are transformed using the extrinsic parameters obtained in the calibration process (see section 3.3) into a common CS. For convenience, the middle sensor CS has been assigned as the reference. Therefore, the transformation is only applied to the data from the two outer sensors.

   The full process of data aggregation is illustrated in figure 4.12.

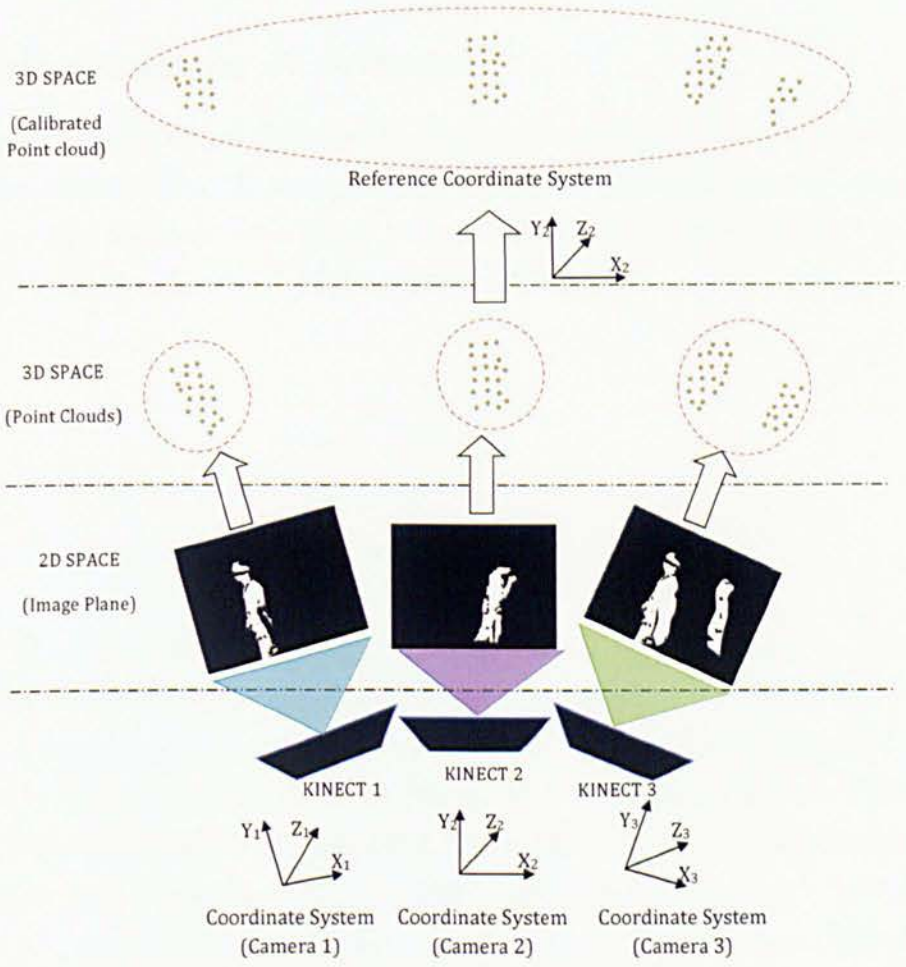


Figure 4.12: Aggregation of data from the three sensors

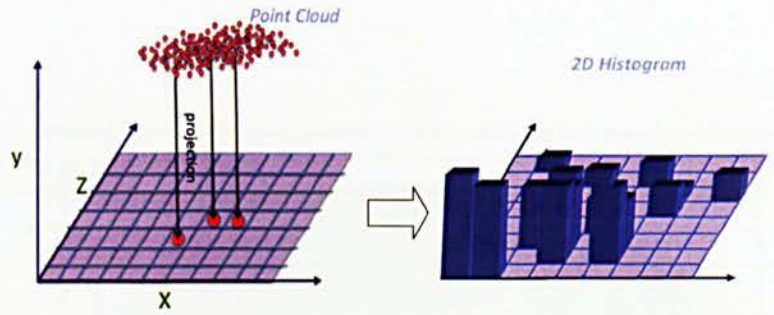


Figure 4.13: 3D foreground points projecting into the ground plane, yielding a 2D histogram of accumulations.

4.3.2 Accumulation of evidence

The aggregated point cloud obtained in the previous step is now projected onto the ground plane over which a 2D histogram is built where each bin stores the accumulation of evidence in that location – see figure 4.13. The bin where a point projects is calculated using the coordinates X and Z of the point as follows.

$$u = \frac{X - \mathcal{M}_X}{b_X}$$
$$v = \frac{Z - \mathcal{M}_Z}{b_Z} \tag{4.6}$$

where $u \in \mathbf{R}$ and $v \in \mathbf{R}$ refer to the bin where the 3D point projects. The variables $\mathcal{M}_X$ and $\mathcal{M}_Z$ are the minimum range of the aggregated FOV in the horizontal and depth axes respectively. Finally, b_X and b_Z define the width and height of the bin in the grid. The histogram is delimited by the range of the combined field of views (FOVs) of the three sensors as depicted in figure 4.14. In this work the minimum range towards the horizontal axis ($\mathcal{M}_X$) was set to -11000 mm., and in the depth axis ($\mathcal{M}_Z$) to 0 mm. The size of the bins were chosen empirically to 20×20 mm. The dimensions of the histogram in terms of number of bins was 1100×500. An example is given in figure 4.15 where the histogram has been converted into a binary image for visualization purposes. This histogram of accumulations is referred in this work as Map of Activity (MoA). In the next section it is assessed whether the MoA is suitable for segmenting people or not.

4.3.3 People segmentation on the MoA: Issues

The MoA is a simple structure where the information from the three sensors is represented in a way where partial occlusions are clearly distinguished. However, the following issues have been identified that complicate the people segmentation task.

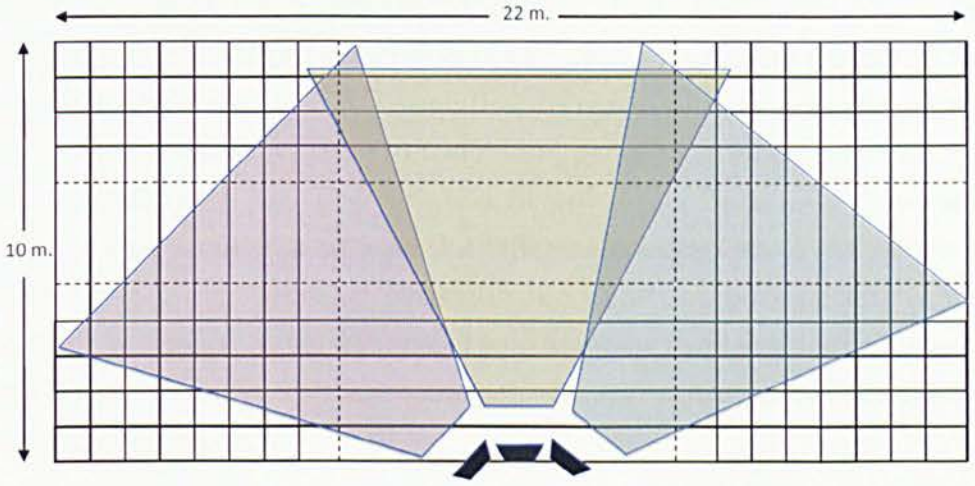


Figure 4.14: 2D histogram that covers the aggregated field of views from the three sensors.

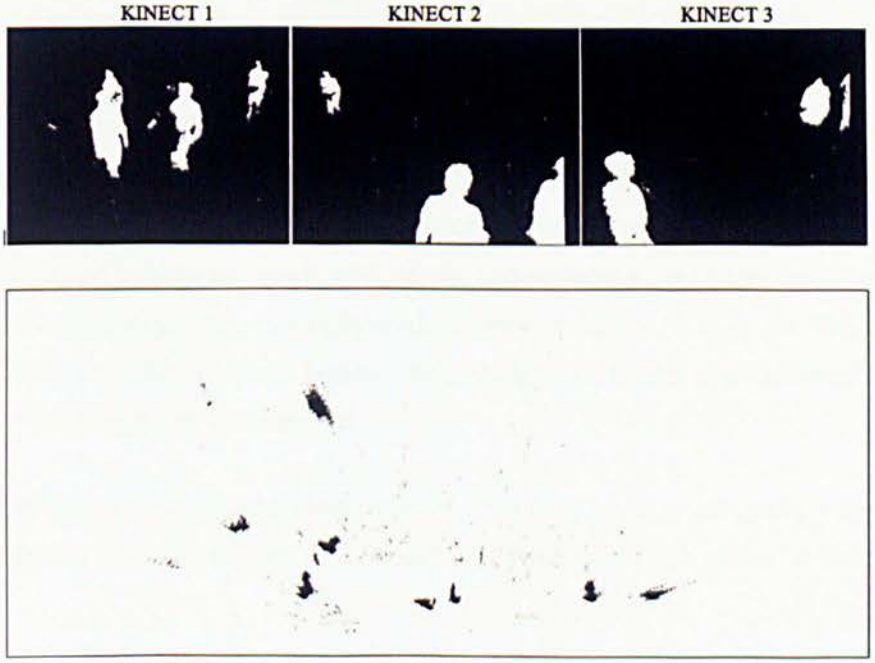


Figure 4.15: Map of Activity built upon the foreground points detected on the three image planes of the sensors. Note that for the purpose of visualization this is a binary image instead of an image of accumulations.

- Non-homogeneous blobs: Points belonging to people project into the MoA forming blobs. Despite the relatively constant size of people, these blobs vary their orientation, width and height throughout the MoA – see figure 4.15. The variation of height (i.e. the depth dimension of the MoA) is an issue related to the depth resolution of the sensor; as the resolution decreases with distance, the gaps between the projected points in the MoA increase, generating larger and more scattered blobs (see figure 4.16). The variation in width (i.e. horizontal dimension of the MoA) is mainly associated with the different orientations of people with respect to the camera e.g. sideways, perpendicular. The non-homogeneity of blobs may result in problems during the smoothing stage as fixed size kernels would not be appropriate. Ideally, in these situation the size of the kernel should vary with respect to the distance.

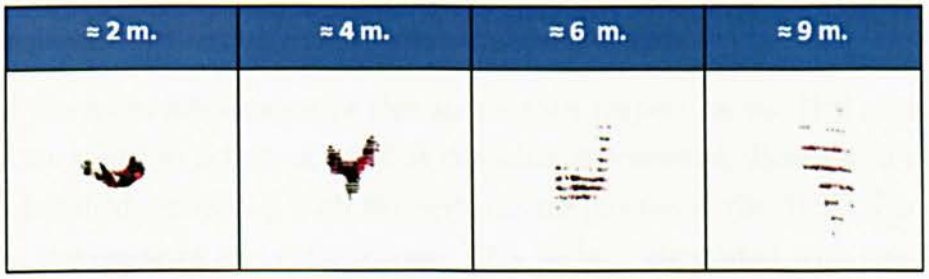


Figure 4.16: Projection of points from a person into the MoA at different distances. Projected points from closer people have higher density than projections form farther distances.

- Varying blob orientations: Blobs in the MoA can be found with three different orientations depending on the camera they are captured from. The reason of this behaviour is associated with the geometry of the cameras and their directions. The points of a blob are scattered along the optical axis of the camera from which they are obtained (see the coloured arrows in figure 4.17). As before this issue affects the smoothing step during the segmentation process, different orientations of kernels should be considered.

These problematic situations make the MoA a less appropriate space for segmenting people. In section 4.4 an alternative space is presented that aims to overcome these difficulties.

4.3.4 Discussion

In this section a common representation (MoA) for the data from the three sensors was described. The MoA is a top-down view of the scene where the foreground objects from the three sensors are projected. The MoA is built in two stages:

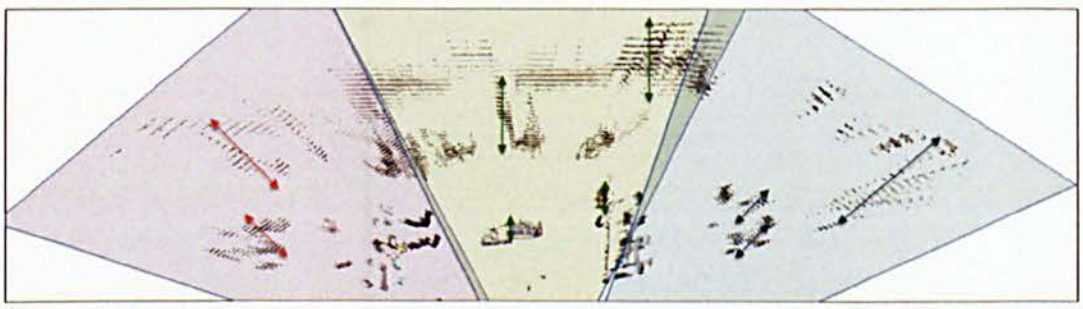


Figure 4.17: Three different orientations of blobs in the MoA, one for each of the three cameras

1. Aggregation of data. The foreground pixels from the three cameras are first projected into the 3D space, and then transformed into a common 3D CS.

2. Accumulation of evidence. The aggregated 3D point cloud is projected into an accumulated histogram defined over the ground plane.

One of the main advantages of this space with respect to the IPS is that partial occlusions are easier to detect as depth is explicitly represented. However, a major issue has been identified associated with the segmenting process in the MoA. The projected blobs have varying sizes and orientations. This issue is associated with the decreasing depth resolution, orientation of people with respect to the camera, and the fact that the three sensors yield blobs orientated in three different ways depending on the camera they are capture from. The segmentation of non-homogeneous blobs requires in general the use of adaptive kernels, where the size of the kernel changes with distance.

An alternative space is presented in the next section that aims to solve or at least minimize the effect of the problem identified above.

4.4 Remapped Polar Space

The Remapped Polar Space (RPS) is an alternative space designed to solve some of the issues that arise in the MoA. Rather than using a Cartesian CS, the points are projected into a polar CS which immediately reduces the problem of different orientations of blobs. In addition, the varying blobs size is mitigated using a mapping function on its radial dimension that aims to normalize the blob height throughout its range. The RPS is built according to the following two steps:

- Cartesian to Polar CS: The aggregated point cloud, which is represented in the Cartesian CS, is transformed to the polar CS where the problem of different blob orientations is diminished.

- Remapping: A transformation is applied directly on the range dimension of the polar CS which aims to reduce the issue of different heights in the projected blobs.

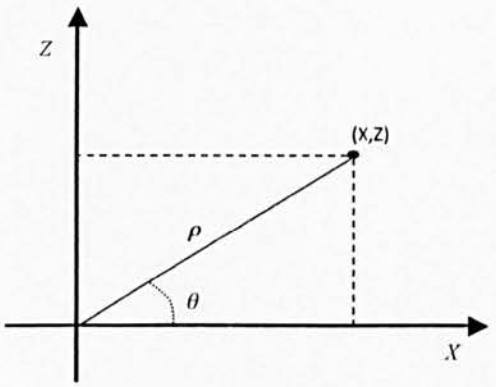


Figure 4.18: Polar CS representation

The RPS is discretised in a 2D histogram of accumulation of evidence (equivalent to the MoA representation), where the segmentation process is applied. The process followed to segment people is equivalent to the one described in the IPS in section 4.2.1.2, with some subtle differences according to the particularities of the space.

This section, first describes the process of building the RPS. Then, the procedure followed for segmenting people as well as some of the issues that still affect the RPS are presented.

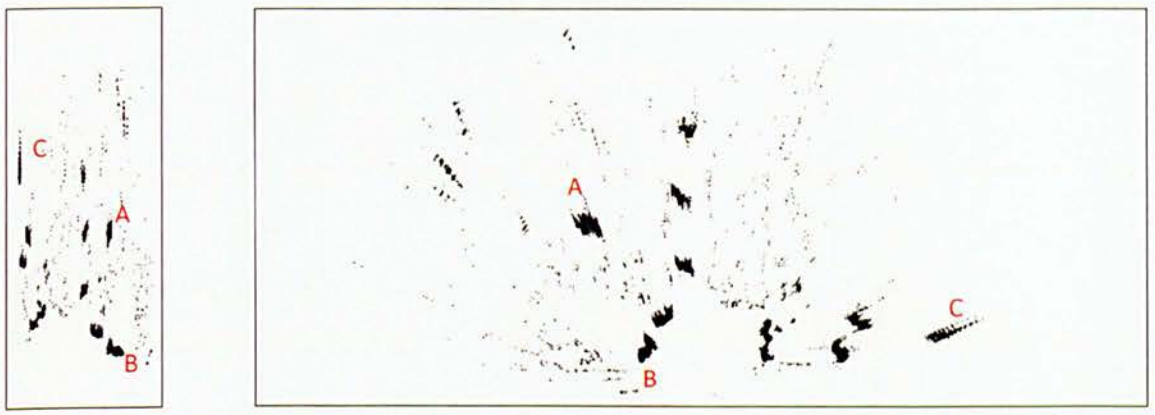
4.4.1 Cartesian to Polar CS

The first stage to build the RPS is to transform the aggregated point cloud from the Cartesian CS to the polar CS. This transformation aims to normalize the different orientations of the blobs. The polar CS is built over a two dimensional space where the data is represented by a distance ρ and an angle θ as shown in figure 4.18. Transforming data from the Cartesian CS to the polar CS is obtained by the following two non-linear equations:

$$\rho = \sqrt{X^2 + Z^2}, \quad \tan\theta = \frac{Z}{X} \tag{4.7}$$

where $Z \geq 0$. Figure 4.19 captures a particular instant of a sequence in both, the Cartesian CS (MoA) and the polar CS, so they can be compared visually.

Although the issue of different blob orientations is addressed in the polar CS, there is still the problem of the size variability. Blobs at farther ranges appear larger than closer blobs. This situation is partially solved by applying a new transformation referred to in this document as **remapping**, which is explained in detail in the following subsection.



(a) Polar CS

(b) Map of Activity

Figure 4.19: Two different representations of the same data: Polar CS and Map of Activity.

4.4.2 Remapping

The remapping is a transformation applied to the ρ dimension of the polar CS to obtain a representation where the radial size of the blobs is homogeneous. This representation is referred in this document as the Remapped Polar Space (RPS).

The intuition behind the remapping is to compress larger blobs at farther distances and stretch smaller blobs at closer distances, bringing a constant blob height throughout the range. At this point it is important to be reminded that the degradation on the depth resolution (see section 3.2.1.3) is the primary reason for the different heights of the blobs. Therefore, the remapping is derived from equation 3.2 which models the depth resolution of the sensor.

The calculation of the remapping function $f(\rho)$ is obtained from the following equality that ensures a constant height of the blobs throughout the whole range:

$$\xi_{(}\rho) * \frac{\partial f}{\partial \rho} = C \tag{4.8}$$

where $\xi_{(}\rho)$ is a function capturing the variation of depth resolution with respect to the range ρ, $\frac{\partial f}{\partial \rho}$ is the derivative of the mapping function, and C defines a constant height for the blobs. The derivative can be obtained re-arranging the terms as follows:

$$\frac{\partial f}{\partial \rho} = \frac{C}{\xi_{(}\rho)} \tag{4.9}$$

If C is set to 1, the derivative of the transformation is just the inverse of the depth resolution function:

$$\frac{\partial f}{\partial \rho} = \frac{1}{\xi_{(}\rho)} = \frac{1}{2.6\rho^2 + 0.6\rho - 0.2} \tag{4.10}$$

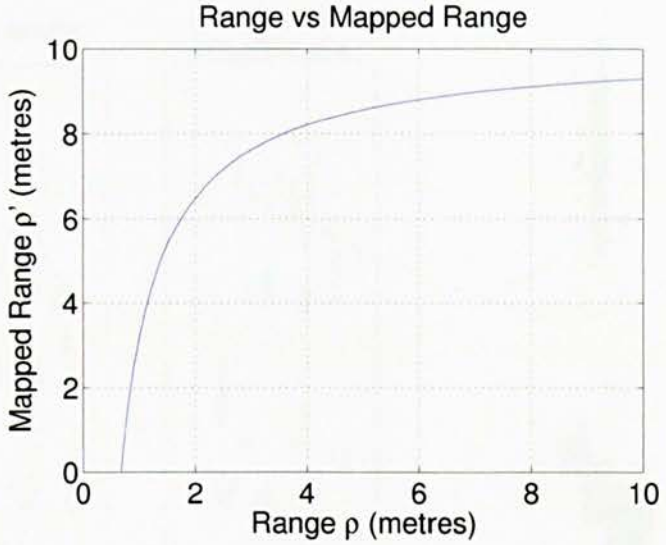


Figure 4.20: Plot of the mapping function $f(\rho)$

where the quadratic in the denominator was estimated in section 3.2.1.3. Finally, the remapping function is obtained by integration (see figure 4.21)

$$\begin{aligned} f(\rho) &= \int \frac{1}{2.6\rho^2 + 0.6\rho - 0.2} d\rho \\ &= 0.6 * (\log(5 - 26\rho) - \log(11 + 26\rho)) + \text{constant} \end{aligned} \tag{4.11}$$

The constant is set to 0.5, so the resulting range is always positive. In addition, the function is multiplied by a scale factor $S = 20$ to normalize the range between 0 and 10 metres.

Using the remapping function 4.11 all data from the polar CS is transformed into the new space (RPS) yielding a two dimensional (θ, ρ') set of points, where the coordinate ρ' derives from the ρ coordinate of the polar CS.

In order to work in the RPS the data is discretised and represented in a 2D histogram of accumulations equivalent to the MoA histogram (section 4.3.2). In this case the dimension of the histogram is 500×180 bins, where the vertical axis represents the remapped range and the horizontal axis defines the angle, and the bin size is set to 2 cm $\times$ 1 degree in the remapped range and angle dimensions respectively. A visual comparison in both spaces, polar CS and RPS is presented in figure 4.21[3]. It also shows two enlarged regions in both representations. The region with the more distant blob is slightly more compressed than its equivalent in the polar CS. On the other hand, the closer blob is expanded in the RPS with respect to the same blob in the polar CS.

[3]For visualization purposes the two images are converted into binary images.

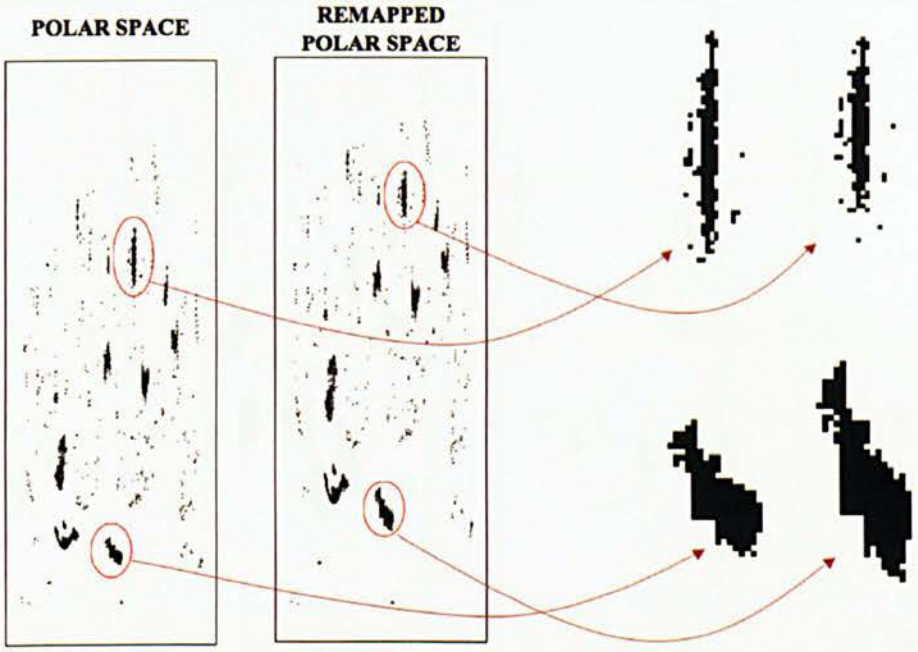


Figure 4.21: People representation in both, the polar CS and in the RPS.

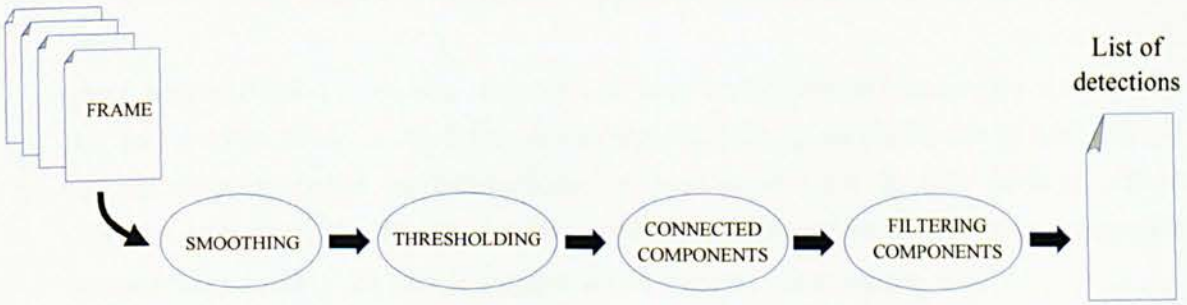
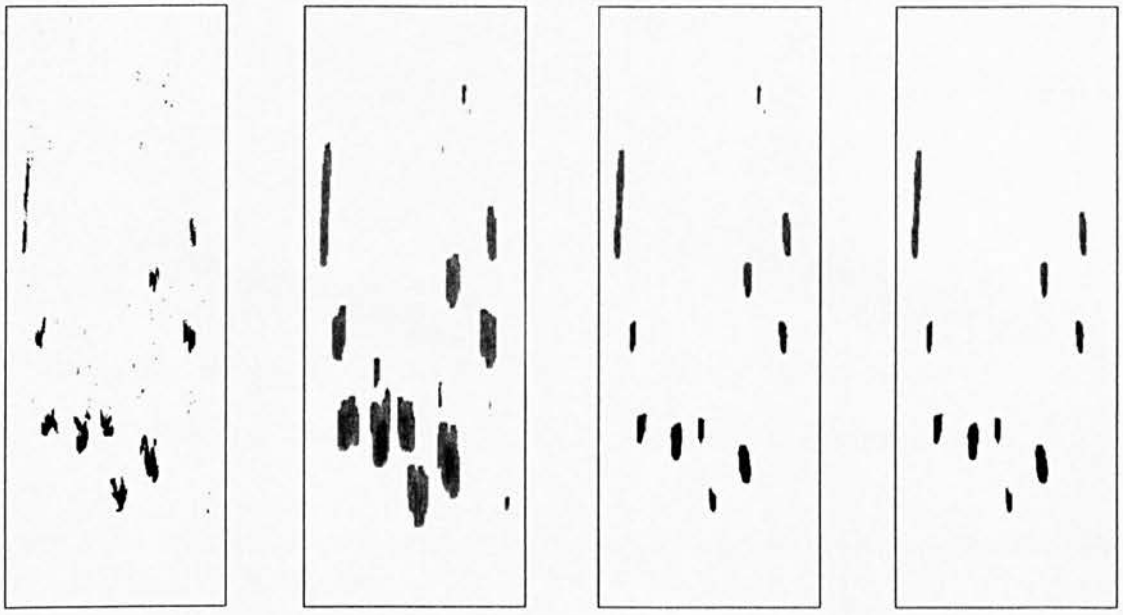


Figure 4.22: Classical people segmentation pipeline: Smoothing, Thresholding, Connected Component and Filtering.

4.4.3 People segmentation on the RPS

The main objective of this chapter is to study and compare the performance of different spaces applied to the people segmentation problem. In order to get a fair comparison, the process of segmenting people should be similar in every space i.e. only the space changes. Therefore the classical segmentation pipeline used in the IPS (with some particularities associated with the RPS) is used. This pipeline includes smoothing, thresholding (using a low threshold), connected components and filtering of components (using a high threshold). The thresholding and filtering stages perform a similar role to hysteresis thresholding as proposed by Canny [180]. Figure 4.22 depicts this segmentation pipeline. These are described below.

- Smoothing: This step aims to reduce the noise and eliminate the gaps within blobs by applying a convolution to the RPS image with a 2D Gaussian kernel. It is important to use an appropriate size for the kernel to avoid under or



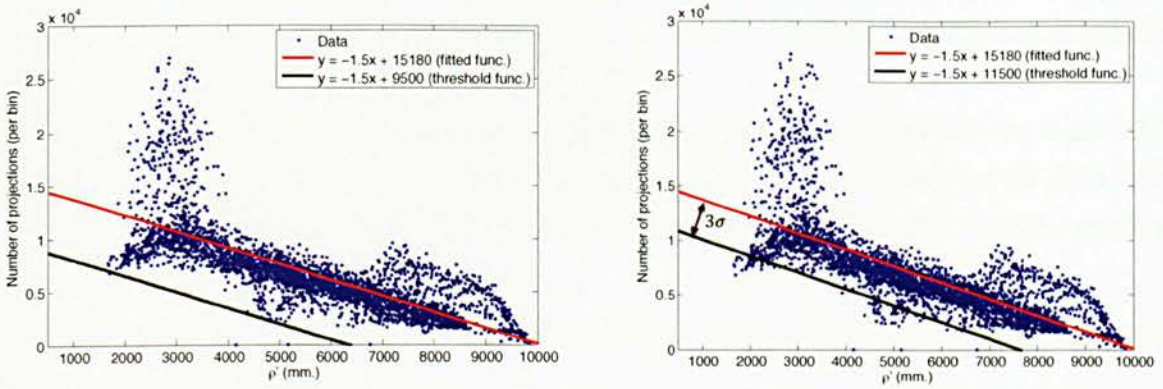
(a) Original (binary) (b) Smoothed (c) Thresholded (d) Filtered

Figure 4.23: Stages of the people segmentation process in the RPS.

over segmentation. Ideally, the size of the kernel should have the size of the typical person blob in the RPS. Although the extent of the blobs is normalized by remapping, other particularities, as explained later in this section, affect the dimension of the blobs. In this work the size of the kernel was estimated empirically as 11 × 25 pixels (sigma 2 × 4 pixels) – see figure 4.23.

- Thresholding (low threshold): The purpose of this process is to reduce the majority of the noise in the image, and prepare the data for the next step – see figure 4.23(c). This thresholding is performed pixel-wise and it is designed to remove small isolated noise peaks. The threshold varies with ρ' and it has been modelled using a set of samples taken from a training sequence. Connected components are extracted from the training sequence and the amount of evidence accumulated on the centre bin of each component is plotted on figure 4.24(a). The centre bin is assumed to contain the higher value of the component. As expected, samples taken from closer distances contain more accumulation of evidence than more distant samples[4]. This behaviour can be approximated with a linear function within this particular range i.e. 0.5 m to 10 m. The threshold for eliminating noise is set empirically lower than the fitted function to avoid filtering out bins that belong to actual people (equation 4.12). There is also a minimum threshold ($\tau_{Low:min} = 100$) to be used where the threshold function does not apply i.e. from 6 metres approx.

[4]Not considering the edge effect produced at closer distances i.e. 0.5 m to 2.5 m



(a) Thresholding (Low threshold)

(b) Filtering of components (High threshold)

Figure 4.24: Number of projections per pixel. Data points collected experimentally from a training video sequence.

$$\tau_{Low}(\rho') = -1.5\rho' + 9500 \tag{4.12}$$

- Connected components: This operation segments blobs of connected points within the binary RPS representation. Foreground points that are spatially 8-connected in the RPS are grouped into the same blob.

- Filtering of components (high threshold): Similarly to the hysteresis threshold [180], components that do not possess any bin with an evidence value higher than a certain threshold are filtered out – see figure 4.23(d). This threshold is computed by lowering the fitted function (computed in the previous threshold stage) three standard deviations to cover most of the data[5] – see figure 4.24(b). As in the thresholding stage a minimum value is used were the threshold function does not apply from 7 metres approx. ($\tau_{High:min} = 200$).

$$\tau_{High}(\rho') = -1.5\rho' + 11500 \tag{4.13}$$

4.4.4 Issues

The different orientations of people with respect to the camera affects the size of the blobs as well. For instance, the blob of a person who is sideways to the sensor is wider along the angular dimension and shorter along the radial dimension. On the contrary,

[5]The training samples were projected onto the perpendicular line to the fitted function. The variance was computed from the Gaussian distribution of points in this one dimensional space.

if the person is perpendicular to the image plane of the sensor, the blob will appear narrower and larger. This is not a specific issue related to the RPS but is a general effect that is present in every space.

Additionally, the segmentation in the RPS requires more computational time (26% of the total time is spent building the RPS histogram). The mapping of all data from the Cartesian CS to the RPS comprises a set of expensive non-linear transformations that must be performed point-wise.

4.4.5 Discussion

In this section an alternative space (RPS) has been presented that aims to solve the issues of different orientations and dimensions of the projected blobs that affect the MoA.

Firstly, the problem of different orientations is solved by representing the data in the polar CS. The second issue, varying dimensions, is approached by remapping the radial dimension of the polar CS into an alternative space where the radial width of objects is made homogeneous. This mapping is derived from the inverse of the depth resolution function of the sensor.

In a similar fashion to the MoA, the RPS is overlaid by a histogram where each bin stores the number of projections. Based on this RPS histogram, the people segmentation is applied following the traditional pipeline: smoothing, thresholding, connected components and filtering.

There are still some outstanding issues such as people with different orientations that yield varying blob sizes and the expensive computational requirements for building the RPS histogram.

4.5 Performance evaluation

In this section the Image Plane Space and the Remapped Polar Space are evaluated and compared with a particular focus on their impact on the performance of the segmentation process.

People segmentation can be used for different tasks such as counting people, tracking, action recognition, etc. Each application has its own requirements. For instance an application that recognizes people actions may require highly accurate results in terms of spatial location and it may not need to detect people who are farther away. This work, on the other hand, aims to monitor larger spaces where one of the requirements is to detect every person in the scene even if they are distant from the sensor. The spatial accuracy is not a priority in this case.

To facilitate the evaluation, the results obtained from the system are compared

using a dataset and a ground truth. This ground truth has been manually annotated in the MoA and therefore the segmentations produced in the IPS and RPS must be transformed into the MoA.

The first two following subsections discuss the relevant failure modes and the chosen metrics. In addition, the parameters involved in the evaluation are identified. Next, the processes for projecting the IPS and RPS segmentations into the MoA are described. After the results obtained from the two spaces are presented, a discussion is conducted based on the results obtained.

4.5.1 Failure modes

The failure modes of a system refers to a hopefully small discrete set of categories of situations where the output of the algorithm is different to what it is expected. Some failures are more relevant than others depending on the application. For that reason it is important to identify the relevant failure modes for each application, in order that the evaluation provides meaningful results. For this application the following two failure modes are identified:

- Misdetection of people: The algorithm fails to segment a person in the scene. Normally this situation occurs when the person is partially occluded or because the depth signal is noisy and there is not enough evidence to support the presence of a person.

- Falsely detected people: The system incorrectly segments a person in a location where in fact there is no person. Noisy environments and incorrect foreground segmentations are normally the responsible for these situations.

The former failure mode is often approached by lowering the threshold of the segmentations, so that less evidence is required to support the presence of a person. However, such a measure typically results in the second failure mode in which noise is incorrectly detected as people.

The failure modes are evaluated on a common dataset for the two different spaces, IPS and RPS. This dataset and its corresponding ground truth are explained in detailed in section 4.5.4.

4.5.2 Metrics

Once the failure modes have been identified the next step is to decide on a set of suitable metrics that account for the failure modes. The computation of the metrics requires a prior step where the ground truth is mapped to the system detections (SD) i.e. detections produced by the system. This mapping consists of associating the ground

truth annotations with the SDs at each frame. It is performed based on the degree of overlap between ground truth and SD, which is measured by the Bhattacharyya coefficient (B_c). A ground truth annotation at a particular frame is mapped onto a SD if their degree of overlap is higher than a certain threshold (τ_o). (For this evaluation the threshold was estimated empirically to 0.6).

This mapping between ground truth and SD allows the computation of the number of true positives (TP), false positives (FP) and false negatives (FN). TPs refer to the number of correctly detected people in the whole sequence, FPs are the number of incorrect detections made by the system, and FNs define the number of people that were not detected by the system.

Based on these values the performance of the system is represented by precision (P) and recall (R) – see equations 4.14. These are simple metrics that cover the failure modes described in section 4.5.1. In addition, the popular F1-score is used to present a single value to describe the overall performance of the system.

$$P = \frac{TP}{TP + FP}$$
$$R = \frac{TP}{TP + FN} \tag{4.14}$$

The F1-score – the harmonic mean of precision and recall – is defined as follows:

$$F_1 = \frac{2 * P * R}{P + R} \tag{4.15}$$

4.5.3 Projection of detections to MoA

The performance evaluation of the system is estimated by comparing the results obtained with a given ground truth which represents the ideal result. In the dataset used for this evaluation the ground truth was manually extracted on the Map of Activity (MoA). This means that the segmentations obtained in the IPS and in the RPS need to be transformed into the MoA in order to be compared with this ground truth.

Projecting IPS detections into MoA

The extracted blobs in the IPS are represented as two dimensional Gaussian PDFs, where the mean and covariance represent the centroid and physical extent of the person

(a) Left sensor. (b) Middle sensor. (c) Right sensor.

Figure 4.25: People segmentation in the IPS of the three Kinect sensors.

in the image plane respectively – see figure 4.25 where the ellipses are computed from the covariance matrix. The process for transforming these detections into the MoA consists of the following steps:

1. Projection into the 3D CS of the camera. All pixels of every detected blob are projected into each corresponding 3D camera CS using equations 4.5.

2. Tilt correction. The tilt angle of the cameras is corrected since this angle is known. The objective is to represent the data in a 3D CS where the Y axis is orthogonal to the ground plane.

3. Computation of Gaussian parameters. The 3D points of each blob are modelled as 3D Gaussian distributions $p \sim N(\mu, \Sigma)$

$$\mu = \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 & \sigma_{xz}^2 \\ \sigma_{xy}^2 & \sigma_y^2 & \sigma_{yz}^2 \\ \sigma_{xz}^2 & \sigma_{yz}^2 & \sigma_z^2 \end{bmatrix} \tag{4.16}$$

4. Transformation of PDFs into a common CS. Using the calibration parameters (R and t) obtained in section 3.3 the mean and covariance are transformed as follows:

$$\mu_c = R\mu + t, \quad \Sigma_c = R^T \Sigma R \tag{4.17}$$

5. Projection of PDFs into the MoA. The parameters of the 3D Gaussian PDF of each blob are projected into the 2D ground plane MoA. The mean μ_c is mapped using equation 4.6, and the covariance Σ_c is projected as follows

$$\Sigma_{MoA} = \mathcal{P}\Sigma_c \mathcal{P}^T \tag{4.18}$$

where $\mathcal{P}^{2\times 3}$ is the projective matrix derived from equation 4.6 as follows:

Figure 4.26: People segmentations in the IPS and transformed into the MoA.

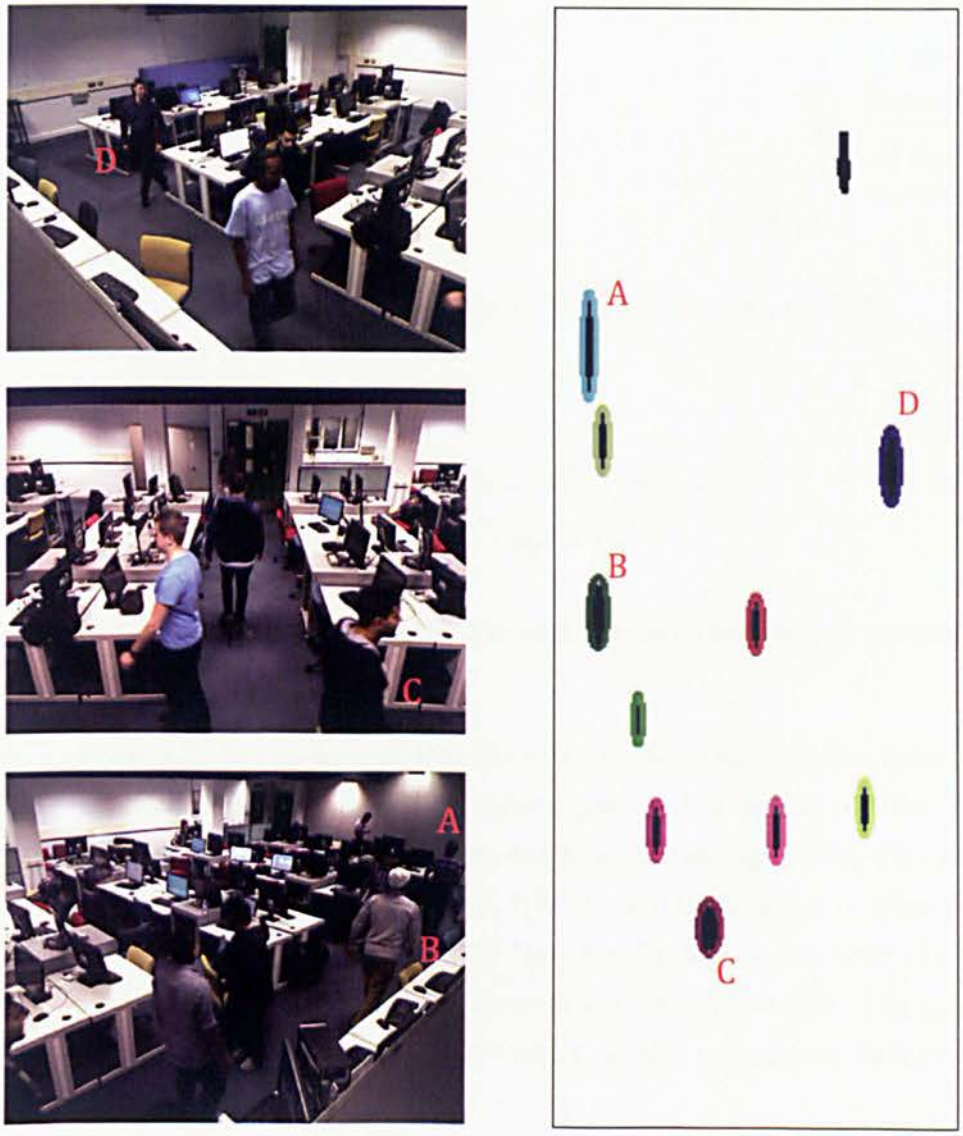
$$\mathcal{P} = \begin{bmatrix} \frac{1}{b_X} & 0 & 0 \\ 0 & 0 & \frac{1}{b_Z} \end{bmatrix} \tag{4.19}$$

Figure 4.26 illustrates a particular instant of a video sequence with the detections represented as ellipses in both spaces, IPS and MoA.

Projecting RPS detections to MoA

The extracted blobs in the Remapped Polar Space are modelled with 2D Gaussian distributions in the same way as in the case of the IPS, where the mean and covariance represent the centroid and scatter of points of a person in the RPS respectively – see figure 4.27. The process of transforming the PDF from the RPS to the MoA is illustrated in figure 4.28 and can be described in the following four steps:

1. RPS(histogram) to RPS. The RPS(histogram) refers to the discrete representation where the segmentation is performed and the RPS defines the continuous Remapped Polar Space. The transformation of the 2D Gaussian from one space to the other is simplified by considering that the angle dimension remains equal and the covariance matrix is diagonal. Therefore, the process is reduced to the transformation of the mean range $\mu_{\rho'_{hist}}$ and the propagation of the variance range $\sigma_{\rho'_{hist}}$ as follows:



(a) RGB images.

(b) RPS.

Figure 4.27: People segmentation in the Remapped Polar Space.

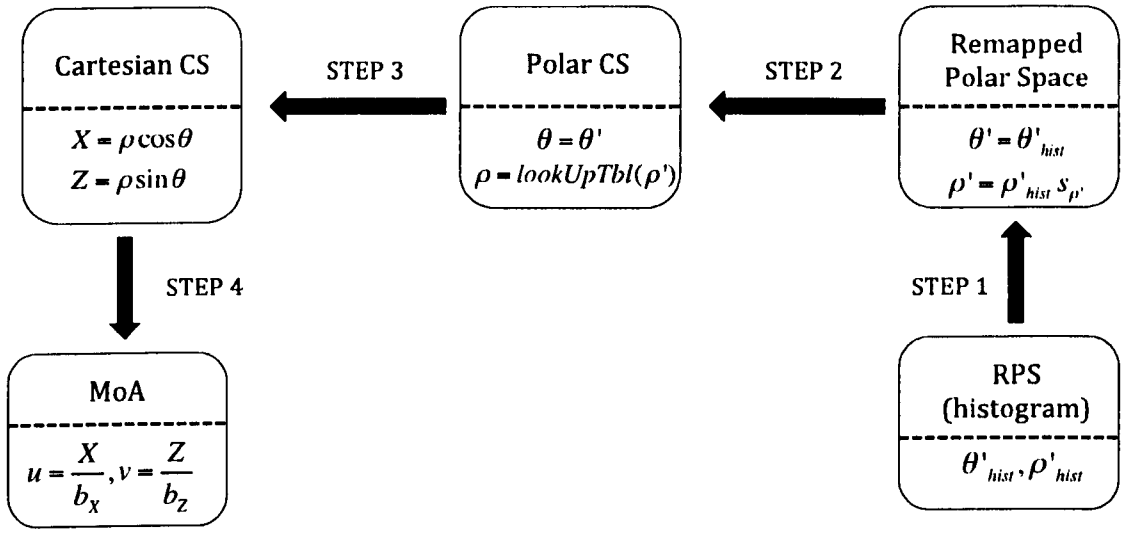


Figure 4.28: Transformation from RPS(histogram) to MoA.

$$\begin{aligned} \mu_{\rho'} &= \mu_{\rho'_{hist}} s_{\rho'}, \quad \sigma^2_{\rho'} = \sigma^2_{\rho'_{hist}} s^2_{\rho'} \\ \mu_{\theta'} &= \mu_{\theta'_{hist}}, \quad \sigma^2_{\theta'} = \sigma^2_{\theta'_{hist}} \end{aligned} \tag{4.20}$$

where $s_{\rho'}$ is the down-sampling factor and defines the level of accuracy in the transformation.

2. RPS to Polar CS. In the second transformation the considerations taken in step 1 can be applied as well i.e. angle remains equal and diagonal covariance matrix. The mean in the polar CS μ_ρ is estimated from a look up table, which was built during the mapping of points into the RPS to avoid the need to invert equation 4.11 ($\mu_\rho = lut(\mu_{\rho'})$). Note here that if the exact range is not found in the table its mapping is interpolated from the nearest values in the table. The propagation of the variance is derived for the derivative of the remapping function 4.11 as follows:

$$\sigma^2_{\rho'} = \sigma^2_\rho \left(\frac{df}{d\rho}\right)^2_{\mu_\rho} \tag{4.21}$$

where $\sigma^2_{\rho'}$ and σ^2_ρ are the variances of the range in the RPS and in the polar CS respectively; and $\left(\frac{df}{d\rho}\right)_{\mu_\rho}$ defines the partial derivative of the remapping function of equation 4.10 evaluated at the mean μ_ρ. From this transformation the variance in the polar CS can be obtained as follows:

$$\sigma_\rho^2 = \frac{\sigma_{\rho'}^2}{\left(\frac{df}{d\rho}\right)^2_{lut(\mu_{\rho'})}} \tag{4.22}$$

3. Step 3: Polar CS to Cartesian CS. The third transformation is defined with the following non-linear functions:

$$X = \rho\cos\theta, \quad Z = \rho\sin\theta \tag{4.23}$$

The mean in the polar CS $\mu_{pcs} = [\mu_\rho, \mu_\theta]^T$ is transformed directly using equations 4.23 to the mean in the Cartesian CS $\mu_{ccs} = [\mu_X, \mu_Z]^T$. The propagation of the covariance Σ_{pcs} requires the use of linear approximations of equations 4.23, which are computed using the first order term of the Taylor expansion (evaluated at the mean μ_{pcs}) i.e. the Jacobian matrix.

$$\Sigma_{pcs} = \begin{bmatrix} \sigma_\rho^2 & \\ & \sigma_\theta^2 \end{bmatrix} \tag{4.24}$$

$$\Sigma_{ccs} = J(\mu_{pcs})\Sigma_{pcs}J(\mu_{pcs})^T \tag{4.25}$$

where $J(\mu_{pcs})$ is the Jacobian matrix evaluated at μ_{pcs} and is defined as:

$$J(\mu_{pcs}) = \begin{bmatrix} J_X \\ J_Z \end{bmatrix} \tag{4.26}$$

where

$$\begin{aligned} J_X &= \left[\frac{dX}{d\rho}, \frac{dX}{d\theta}\right] = [\cos\theta, \ -\rho\sin\theta] \\ J_Z &= \left[\frac{dZ}{d\rho}, \frac{dZ}{d\theta}\right] = [\sin\theta, \ \rho\cos\theta] \end{aligned} \tag{4.27}$$

4. Step 4: Cartesian CS to MoA. The final transformation refers to the mapping into the ground plane MoA. The mean $\mu_{MoA} = [\mu_u, \mu_v]$ is obtained from equation 4.6, and the covariance Σ_{MoA} is computed as follows:

$$\Sigma_{MoA} = \mathcal{P}_{red}\Sigma_{cart}\mathcal{P}_{red}^T \tag{4.28}$$

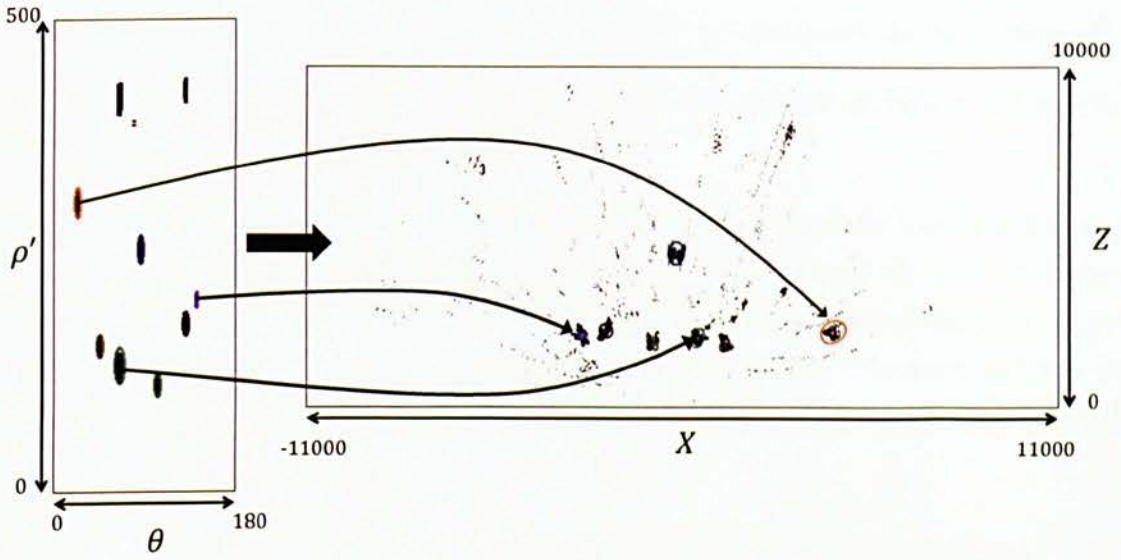


Figure 4.29: Detections in the RPS transformed into the MoA.

where $\mathcal{P}_{red}^{2\times2}$ is the projective matrix derived from equation 4.6 as follows:

$$\mathcal{P}_{red} = \begin{bmatrix} \frac{1}{b_X} & 0 \\ 0 & \frac{1}{b_Z} \end{bmatrix} \tag{4.29}$$

Figure 4.29 shows the detection of people on the RPS and the transformed version in the MoA.

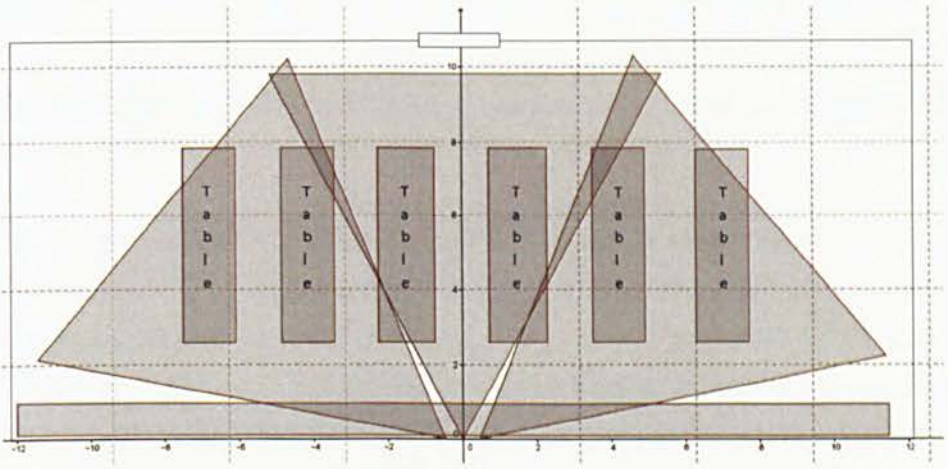
4.5.4 Benchmark dataset and ground truth

Since the release of the Microsoft Kinect different indoor datasets have been created for different types of indoor applications. Most of them are aimed at object recognition tasks [148, 149] and human activity recognition [150, 151]. Few datasets have been found for the evaluation of people segmentation and tracking systems. One of them was recently published by Munaro and Menegatti [86] called Kinect Tracking Precision (KTP), which is a dataset acquired from a mobile robot platform. To the author's best knowledge the only RGB-D dataset recorded from static RGB-D cameras for people segmentation and tracking purposes is the one presented by Spinello and Arras [76]. In their dataset three vertically mounted Kinects are located in a non-overlapping configuration at approximately 1.5 metres high. Although this dataset is close to the purposes of this project, it does not fulfil some essential requirements:

- The location of the sensors must be at a high location. e.g. 2 metres or higher, so the number of occlusions is minimized.

- The cameras should be mounted horizontally to maximize the area covered[6].
- The configuration of the scene should allow the capture of data at its maximum range, 10 metres approximately.

Therefore, a new data set is proposed that aims to satisfy the aforementioned requirements. A set of video sequences was recorded in one of the labs of Kingston University. Three Kinects were horizontally placed in a non-overlapping adjacent configuration, where the area covered was maximised. They were located half way along the largest wall of the room at approximately 2.20 metres high. The area covered by the whole system is around 220 m^2. See figure 4.30.



(a) Diagram of the lab where the recordings took place.

(b) Left camera.

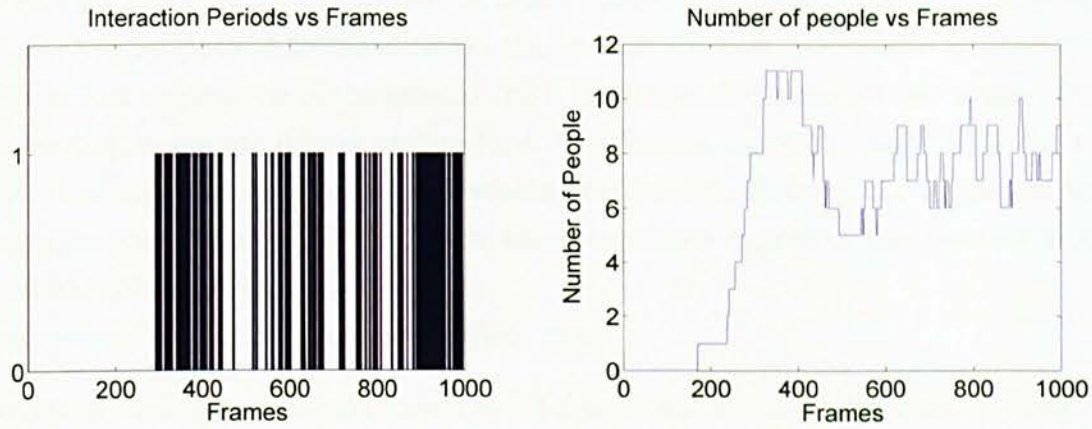
(c) Middle camera.

(d) Right camera.

Figure 4.30: Configuration of the cameras in the lab and the actual views of the three Kinects.

Two sequences of 1000 frames each were captured. One is used for the training of parameters of the algorithm i.e. threshold values, while the other is used for the actual evaluation of the system. The sequences were recorded using the ".oni" format provided by the OpenNI framework. This format combines the colour and depth information. The colour data is stored in a 640x480 array of 8 bit, 3 channel. The depth information is presented in a 640x480 array where each position contains a value between 500 and 9700 mm.

[6]The Kinect depth sensor features an angular field of view of 57° horizontally and 43° vertically.



(a) Plot of the interaction periods in the test sequence

(b) Plot of the number of people present at each frame during the whole sequence

Figure 4.31: Description of the test video sequence in terms of periods of interactions between people and number of people present in the scene.

Up to 15 people appear in the evaluation and training sequence walking in a casual way along the aisles of the lab. Figure 4.31(b) describes the sequence in terms of the number of people present at each frame of the video. The actors leave and re-enter the scene multiple times as a consequence of the limited range of the Kinect sensor. The dataset comprises approximately 140 different people interactions where most of them are short-lived consisting on path crossing between people. There are as well occasional handshaking and grouping interactions with direction and motion changes after it. Figure 4.31(a) plots the frequency and duration of periods where two or more people are part of an interaction. Due to the layout and structure of the scene there are multiple partial static occlusions when people walk behind desks and computers.

These sequences were manually annotated with bounding boxes in the Map of Activity using the open source tool VATIC[7]. The ground truth annotations can be defined as the ideal values that any algorithm aims for. These annotations have to be as objective as possible and not being biased by any other process. In general, human annotations are considered the perfect values, however the results can be subject to minor errors related to the subjective interpretation of the annotator. Moreover, ground truth annotation is a highly tedious and monotonous task where the annotator might get distracted or reduce their concentration at some point, resulting in the introduction of additional errors in the ground truth data. To slightly alleviate the task, the tool VATIC features a linear interpolation capability, so that annotations do not have to be recovered in every single frame; the annotator can just accept the interpolation results. Nevertheless, this is subject to some errors as well, as the annotations might get biased towards the interpolation tool.

[7]http://web.mit.edu/vondrick/vatic/

Although there are several sources of errors involved in the generation of the ground truth, for the purpose of this evaluation the ground truth is assumed to be error-free.

The actual annotation of the ground truth in this work consists of two stages. First, the bounding boxes are drawn on the MoA by a human operator aided with the tool VATIC. Second, the data contained within the bounding boxes are modelled with two-dimensional Gaussian PDFs where the dimensions represent the horizontal and vertical axis of the MoA.
The procedure for annotation follows three rules:

- The bounding box should have the minimum size to cover the whole person.
- During occlusion periods only the visible part is annotated. Note here, that when two or more people are involved in an interaction the process of estimating the limits of each target is quite difficult and might lead to some small accuracy errors in the annotation data.
- When a person leaves the scene and later on re-enters; that person is annotated as a different person.

Those bounding boxes that are spatially connected on the MoA are flagged as *merged measurements*. Note that the spatial detection of the people involved in a *merged measurement* is out of the scope of this evaluation.

4.5.5 Results

Table 4.1 presents the evaluation of the people segmentation in the two spaces, IPS and RPS. For the IPS evaluation two versions are compared, the classical approach of intensity-based systems and the extended version with the occlusion reasoning module (described in section 4.2.1.2).

Space/Metric	**Precision**	**Recall**	**F1-Score**
IPS (classical)	0.5	0.67	0.57
IPS (extended)	0.78	0.83	0.8
RPS	0.95	0.84	0.89

Table 4.1: Performance evaluation of the people segmentation process applied to three different spaces.

Not surprisingly, the evaluation indicates a significant improvement when the extended IPS version is used. The reason is two fold; first, since the classical version does not detect individual people within a *merged component* the number of false negatives is higher; and second the PDF of a *merged component* does not match the ground truth PDFs yielding an increase in the number of false positives too.

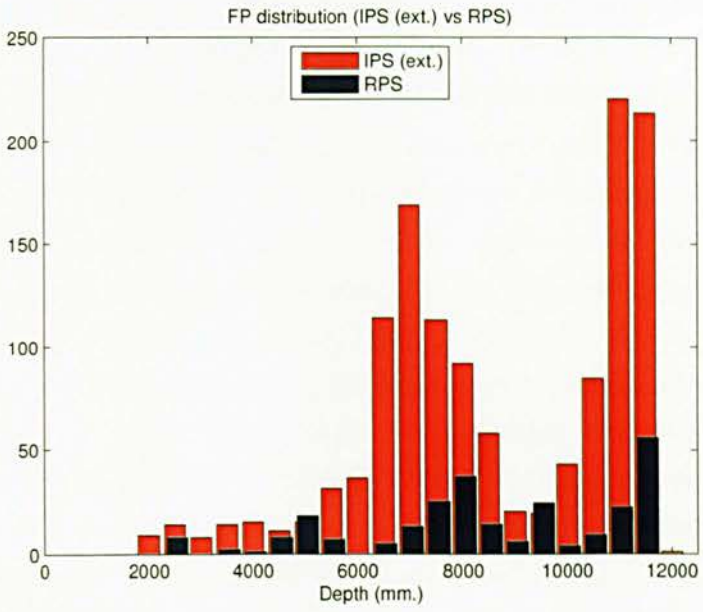


Figure 4.32: Distribution of FPs along the depth dimension in the IPS and RPS

More significantly, the third row of table 4.1 reveals that the RPS outperforms the use of IPS. From this, it can be inferred that occlusion reasoning performed on the ground plane is more effective than in the image plane. It also suggests that the measures taken to mitigate the noise and the decreasing resolution at farther distances are effective. This improvement is presented visually in figure 4.32 with the distribution of FPs obtained in the RPS and in the IPS along the depth dimension. As expected the number of FPs obtained in the IPS increases beyond the operating range of the Kinect sensor. The results obtained in the RPS shows the reliability of the remapping operation. The low values of FPs obtained at 9 metres can be associated with the structure of the scene.

4.6 Discussion

In this section three different spaces have been presented in the context of people segmentation: the Image Plane Space (IPS), the Map of Activity (MoA) and the Remapped Polar Space (RPS). All three exploit in different degrees the depth information provided by the Kinect sensor. The objective of this chapter was to compare the three spaces in the context of people segmentation and identify their weaknesses and strengths. To ensure a fair comparison the people segmentation methodology applied on the three spaces follows the same pipeline: foreground detection, noise filtering, smoothing and connected components (but with variations according to the particularities of the space).

The first space proposed was the Image Plane Space (IPS) which is defined as the

two dimensional digital image produced by the sensor. This space has been widely used over the years for people segmentation with intensity images. Occlusions are the most challenging issue in this space and normally additional information is required for its resolution. In this project an extension to the classical approach has been proposed to deal with occlusions by employing the available depth dimension.

The second space is the Map of Activity (MoA) which is constructed over the ground plane, and serves as a common representation for the data from the three Kinects that constitute the system. Since depth is explicitly represented, the occlusion reasoning is handled naturally. However two important issues were identified in this space. First the blobs projected onto the MoA become increasingly scattered with distance as the depth resolution decreases i.e. varying blob dimensions. Second this scattering occurs along the optical axis of each camera yielding three different orientations of blobs (depending on the camera they were captured from). These issues impact in particular the smoothing stage of the segmentation process as different kernel sizes and orientations would be required.

Finally, a Remapped Polar Space (RPS) is proposed as an alternative space, and aims to solve the issues identified in the MoA. The problem of different orientations is automatically solved by transforming the data into a polar representation. In addition, the effect of varying blob dimensions is mitigated by the use of a remapping operation derived from the resolution function, which aims to normalize the dimension of the blobs.

The evaluation is conducted only on the IPS and RPS which are considered fundamentally different since the RPS is an improved version of the MoA. The results are presented using the precision, recall and F1-score metrics which cover the relevant failure modes identified for this application: misdetections of people and falsely detected people.

The results show that the occlusion reasoning approach applied to the IPS is effective as the performance increases significantly with respect to the classical approach. The use of the RPS, however, represents a significant increase in performance over the use of the IPS. This suggests first that the ground plane is more discriminative for solving occlusions, and second that the actions taken to address the noise and the decreasing resolution of the data are effective.

Chapter 5

Study of Multi-target Tracking Methodologies

5.1 Introduction

Tracking multiple people in crowded scenes is a real challenge mainly because of the number of dynamic occlusions produced between people. It is particularly difficult to correctly re-stablish identities of people after an occlusion. Different methodologies have been proposed in the past to address these situations. However this remains an active research topic.

Traditional approaches use a module for segmenting targets at every time step and rely on a data association process to correctly link the measurements over time. In this context the performance of the data association stage is of the highest importance especially during occlusion situations where the targets' appearance inevitably change. Popular examples of this type of approaches are the Kalman filter and particle filters [5, 56].

Alternative tracking methodologies exist that do not rely on a data association process. The Mean-Shift algorithm is a popular approach within this category. It is mainly used for single target tracking as it is generally highly sensitive to distractions from nearby targets. However, in the past few years some authors have proposed modifications to the original method aiming to make it more suitable for multi-target environments [103, 104].

A chief aspect regardless the tracking algorithm is the model used to describe the appearance of targets. This model should be sufficient discriminant to distinguish people from one another especially during complex situations such as occlusions, illumination changes or variations in the target scale.

In this chapter a traditional tracking methodology, namely the Kalman filter, and the alternative mean-shit approach will be explored in the context of multi-target

tracking. Using the depth dimension provided by the Kinect sensor some enhancements will be proposed to improve the performance specifically during occlusions. Additionally a discriminative appearance model built from the 3D space and the colour dimension will be presented to assist during the tracking process.

The remainder of this chapter is organized as follows. Section 5.2 presents a traditional tracking methodology based on Kalman filter. In particular different data association methodologies are discussed and the new appearance model is proposed. In section 5.3 the Mean-Shift approach is introduced along with some modifications to increase the performance in multi-targets environments. The content of the chapter is discussed in section 5.4.

5.2 Data association strategies applied to tracking

Tracking using the Kalman filter applied to segmented object observations is the most common technique used in visual surveillance systems[181]. It requires a segmentation module to provide people detections at every time step and a data association process to correctly link the detections from frame to frame.

In this context the tracking of a target consists of a recursive process where at every frame its location is predicted using a motion model and then updated with the latest observation. The appearance models of the target and the current observations are compared to find the observation whose model is the most similar to the target's model. In single target tracking when only one observation is detected the association is trivial. However, in multi-target environments the correct solution might become extremely complicated to attain especially in certain situations. For example when targets are in close proximity and have similar appearance, during occlusions when targets disappear temporarily or when spurious observations are detected. This problem has been studied by many authors in the past [86, 88, 91]. However, it is still an unresolved problem. In this work the data association problem is investigated further and a new appearance model is presented, which aims to mitigate some of the uncertainties of the data association process and improve the performance during occlusion situations.

5.2.1 Tracking methodology

In general, the process for tracking people in video sequences consists of labelling people consistently throughout the sequence. It is approached from a recursive perspective with two stages: prediction of the people states from the previous time, and the update of these predictions with the latest measurements. This is commonly known as the estimation problem.

In the context of people tracking two different spaces can be distinguished: the state

space where people are described in terms of location, velocity, acceleration, etc, and the measurement space where observations are represented, normally only with the location. A target moves across the state space over time according to the following function:

$$t_k = f(t_{k-1}, v_{k-1}) \tag{5.1}$$

where $t_k \in \mathbf{R}^{n_x}$ is the vector of n_x dimensions that defines the target state at time k, $f(\cdot)$ is the motion model that predicts the current target state from the last estimation t_{k-1}, and v_{k-1} is a noise signal used to cover any mismodeling issue or unforeseen disturbances. The prediction t_k is updated with the last measurement by converting it from the state space to the measurement space as follows:

$$m_k = h(t_k, w_k) \tag{5.2}$$

where $m_k \in \mathbf{R}^{n_z}$ is the measurement vector, $h(\cdot)$ is the measurement model that converts the target state t_k into the measurement space, and w_k is the measurement noise introduced to cover for the noise of the sensor.

Kalman filter

The Kalman Filter (KF) [42] provides an optimal solution to the estimation problem assuming the target state is Gaussian ($t \sim N(\mu_t, \Sigma_t)$). This assumption implies that the following statements must be true:

- The motion model $f(t_{k-1}, v_{k-1})$ is a linear function
- The measurement model $h(t_k, w_k)$ is a linear function.
- The motion model noise (v_{k-1}) and the measurement noise (w_k) are normally distributed.

Given these considerations the prediction and update stages are:

1. Prediction: The target state is predicted from the last state estimation using the system model as follows:

$$\hat{\mu}_{t,k} = F\mu_{t,k-1}, \qquad \hat{\Sigma}_{t,k} = F\Sigma_{t,k-1}F^T + \Phi \tag{5.3}$$

where $\hat{\mu}_{t,k}$ and $\hat{\Sigma}_{t,k}$ are the predicted mean and covariance of the target at time k, $F \in \mathbf{R}^{n_x \times n_x}$ is the matrix that defines the linear motion model, and Φ is the covariance of the motion model uncertainty, which covers for minor violations of the linearity assumption.

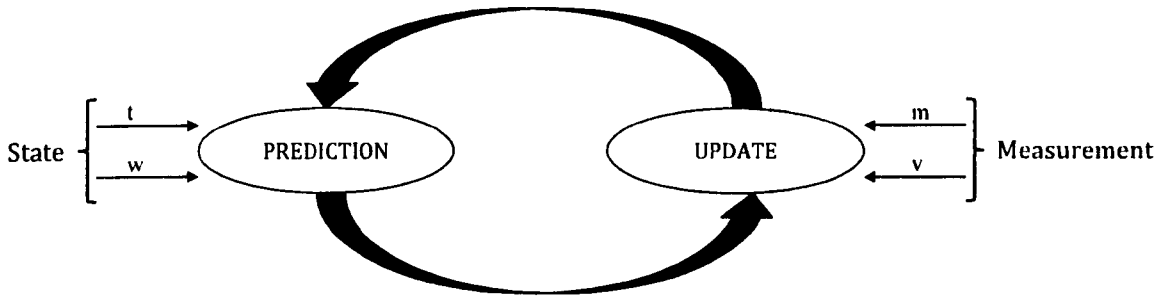


Figure 5.1: Recursive cycle of Kalman Filter: Prediction and Update

2. Update: The predicted mean and covariance at time k are updated using the latest measurement m_k as follows:

$$\mu_{t,k} = \hat{\mu}_{t,k} + K_k(m_k - H\hat{\mu}_{t,k}), \qquad \Sigma_{t,k} = \hat{\Sigma}_{t,k} - K_k S_k K_k^T \tag{5.4}$$

where $H \in \mathbf{R}^{n_z \times n_x}$ is the matrix that defines the linear measurement model, $H\hat{\mu}_{t,k}$ is the predicted measurement, $m_k - H\hat{\mu}_{t,k}$ is often referred as innovation, and S_k is the covariance of the innovation or total uncertainty, which is defined as:

$$S_k = H\hat{\Sigma}_{t,k}H^T + \Lambda \tag{5.5}$$

where Λ is the covariance of the measurement model noise, and K_k is the Kalman gain which is defined as follows:

$$K_k = \hat{\Sigma}_{t,k}H^T S_k^{-1} \tag{5.6}$$

The Kalman gain is used to weight the contribution of the measurement m_k to the final estimation $\mu_{t,k}$. Its value depends on the uncertainty of the prediction ($\hat{\Sigma}_{t,k}$) with respect to the total uncertainty (S_k).

The recursive process of Kalman filter is illustrated in figure (5.1).

5.2.1.1 Design decisions

The most relevant design decisions of the tracker concern the selection of the tracking space, the state and measurement space, the motion model of people and the measurement model.

Tracking space

The tracking space refers to the coordinate system used to represent the physical location of the targets of interest, and in which the tracking takes place. The selection

of a proper tracking space is essential for achieving high performance in tracking. Three different spaces are considered for tracking in this work: Image Plane Space (IPS), Remapped Polar Space (RPS), and the ground plane Map of Activity (MoA). These are the same spaces that were presented in chapter 4 and where evaluated throughout in the context of people segmentation. In the tracking stage, however, such a comprehensive study is not intended. In this section a brief discussion is conducted about the potential performance of the three spaces in the context of a multi-camera multi-target tracking system. The objective of this discussion is to motivate the selection of a suitable tracking space.

Image Plane Space. The IPS is described in detailed in section 4.2. From the three spaces proposed, the IPS has been widely used for tracking people in typical CCTV systems. A multitude of methodologies have been proposed over the years applied to this space [79, 112, 173]. However, for the multi-camera system proposed in this work this space presents the following major disadvantage: re-identification of targets across cameras. For example, when people move from camera to camera ideally their ID should be consistent independently of the camera they were captured from. A solution to this problematic situation would require an external association module. As will be discussed later in the chapter (section 5.2.3), the association problem is not trivial in this context.

Remapped Polar Space. The RPS as described in section 4.4, is a very convenient space for segmenting people, mainly because the size and orientations of targets are homogeneous throughout the space. In the context of tracking systems, it solves the problem of re-identification of targets as the views from the three sensors are aggregated in a common view. However, tracking in a polar CS is not convenient in general, because the motion of people is not linear in the polar system, and therefore more complex tracking solutions are required.

Map of Activity. In section 4.3 the MoA is presented and evaluated in the context of people segmentation. This space was not recommended for segmentation purposes, since target blobs in the MoA are represented with different orientations and varying dimensions. Nevertheless, for tracking purposes this space addresses the problems encounter in the other two spaces.

- The MoA is a common representation for the data from the three sensors. Therefore the problem of re-identification that appears in the IPS does not arise here.
- The motion of people in the MoA can be assumed to be linear, which allows the use of optimal trackers such as Kalman Filter.

Based on this analysis, the MoA is considered more suitable than the IPS and the RPS for tracking multiple people in a multi-camera system.

This analysis was not intended to be exhaustive as the discussion presented in chapter 4 as its only purpose was to justify the use of the MoA for tracking.

State and measurement space

The state of a person is represented in four dimensions $t = (u, v, \dot{u}, \dot{v})^T$, where (u, v) are the two dimensional coordinates that define the location of the person on the MoA, and $(\dot{u}, \dot{v})$ is the velocity in both directions. The measurement space is defined with the two dimensions of the MoA $m = (u, v)^T$.

Motion model

In general, it can be assumed that the motion of people walking has constant velocity i.e. no acceleration. Therefore, using the kinematic equations, the motion model of a person is defined as:

$$F = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.7}$$

Using the prediction equation 5.3 the state of a person at the current time is predicted from the state at the previous time and the motion model as follows:

$$\begin{pmatrix} u_k \\ v_k \\ \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_{k-1} \\ v_{k-1} \\ \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} u_{k-1} + \dot{u}\Delta t \\ v_{k-1} + \dot{v}\Delta t \\ \dot{u} \\ \dot{v} \end{pmatrix} \tag{5.8}$$

Measurement model

The measurement model refers to the function that converts the target state space into the target measurement space and is defined with the matrix $H \in \mathbf{R}^{2 \times 4}$ as follows:

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{5.9}$$

A target state is described in the measurement space as:

$$\begin{pmatrix} u_k \\ v_k \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_k \\ v_k \\ \dot{u} \\ \dot{v} \end{pmatrix} \tag{5.10}$$

This model is used in equation 5.4 during the update stage of KF.

5.2.1.2 Implementation details: Initialization of targets

When a new target is identified from a detected measurement, the following parameters are initialized:

- Target state, $t = (u, v, \dot{u}, \dot{v})^T$.
- Uncertainty of the target state, Σ_t.
- Motion model uncertainty, Φ.
- Measurement model uncertainty, Λ.

Target state

The spatial location of the target state (u, v) is initialized with the location of the associated measurement m. The two dimensions of velocity $(\dot{u}, \dot{v})$ are completely unknown and they are assumed zero.

$$t_{init} = \begin{pmatrix} m_u \\ m_v \\ 0 \\ 0 \end{pmatrix} \tag{5.11}$$

Uncertainty of the target state

The initial uncertainty of the target location is approximated using the 2×2 scattered matrix Σ_m that defines the physical extent of the measurement m. Regarding the uncertainty in velocity an initial value of σ_v is used. Note that this uncertainty will increase every time when adding the motion model uncertainty at the prediction stage (equation 5.3).

$$\Sigma_{t,init} = \begin{pmatrix} \Sigma_m & & \\ & \sigma_v^2 & \\ & & \sigma_v^2 \end{pmatrix} \tag{5.12}$$

where σ_v has been computed from an empirical value of 0.1 m/Δt, which in the MoA yields a standard deviation of 5 bins/Δt since the bin size is 0.2 × 0.2 m.

Motion model uncertainty

The uncertainty of the motion model is set with a constant value σ_m for the location uncertainty and σ_v for the velocity uncertainty that guarantee the recovery of the system in situations where the model differs from the actual motion of the target.

$$\Phi = \begin{pmatrix} \sigma_m^2 & & & \\ & \sigma_m^2 & & \\ & & \sigma_v^2 & \\ & & & \sigma_v^2 \end{pmatrix} \tag{5.13}$$

where the constant value σ_m has been obtained from an estimated uncertainty of 15 cm, which is approximately 50 bins variance in the MoA.

Measurement model uncertainty

The measurement uncertainty is defined over the two dimensions of the MoA as follows

$$\Lambda_{MoA} = \begin{pmatrix} \sigma_u^2 & \sigma_{uv}^2 \\ \sigma_{uv}^2 & \sigma_v^2 \end{pmatrix} \tag{5.14}$$

In order to account for the different orientations of measurements in the MoA, Λ_{MoA} is computed from the uncertainty in the polar CS Λ_{pcs}. The study conducted in section 3.2.1.3, reveals that the accuracy of the Kinect depth sensor decreases with range – see equation 3.3 and figure 3.5. Therefore, each person has a different uncertainty value depending on the distance of that person. The procedure to compute the uncertainty of a particular measurement consists of the following steps:

1. The range ρ_m and angle θ_m of the measurement are computed.
2. The variance in the range dimension σ_ρ^2 is determined from equation 3.3 evaluated at ρ_m.

3. The angle variance σ_θ^2 is set with a constant value, since the accuracy does not vary as a function of the angle. At this point the covariance matrix in the polar CS has the following form:

$$\Lambda_{pcs} = \begin{pmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_\rho^2 \end{pmatrix} \tag{5.15}$$

where σ_θ was set empirically to 7 degrees, which corresponds to 7 bins in the RPS i.e. 1 bin in the RPS accounts for 1 degree.

4. The uncertainty in the polar CS Λ_{pcs} is transformed into the MoA with the following geometric transformation:

$$\Lambda_{MoA} = R_{\theta_m} \Lambda_{pcs} R_{\theta_m}^T \tag{5.16}$$

where R_{θ_m} is the rotation matrix computed at the measurement angle θ_m

$$R_{\theta_m} = \begin{pmatrix} \cos(\theta_m) & -\sin(\theta_m) \\ \sin(\theta_m) & \cos(\theta_m) \end{pmatrix} \tag{5.17}$$

5.2.1.3 Issues: The need for data association

The Kalman Filter assumes that the measurement used during the updating stage is correct. This assumption is challenging to ensure, especially in multi-target tracking applications such as the one proposed in this work. The Kalman filter is not responsible for the correct association of measurements as is illustrated in figure 5.2 where the data association module is located outside the Kalman Filter.

The data association module receives at every time step a set of targets and measurements of unknown origin. The similarity[1] between targets and measurements is computed based on their appearance models obtaining a matrix known as "similarity matrix" that relates targets (rows) with measurements (columns). The objective is, using this matrix, to find the set of disjoint associations that maximizes the overall similarity. At this stage the appearance model employed to describe targets and measurements is decisive for the success of the association process. In the next section two different appearance models are explored.

5.2.2 Appearance modelling

Appearance models are used during the data association stage to compare targets and measurements. For visual tracking the appearance model employed to describe people

[1]Alternatively it could compute the dissimilarity or cost of association.

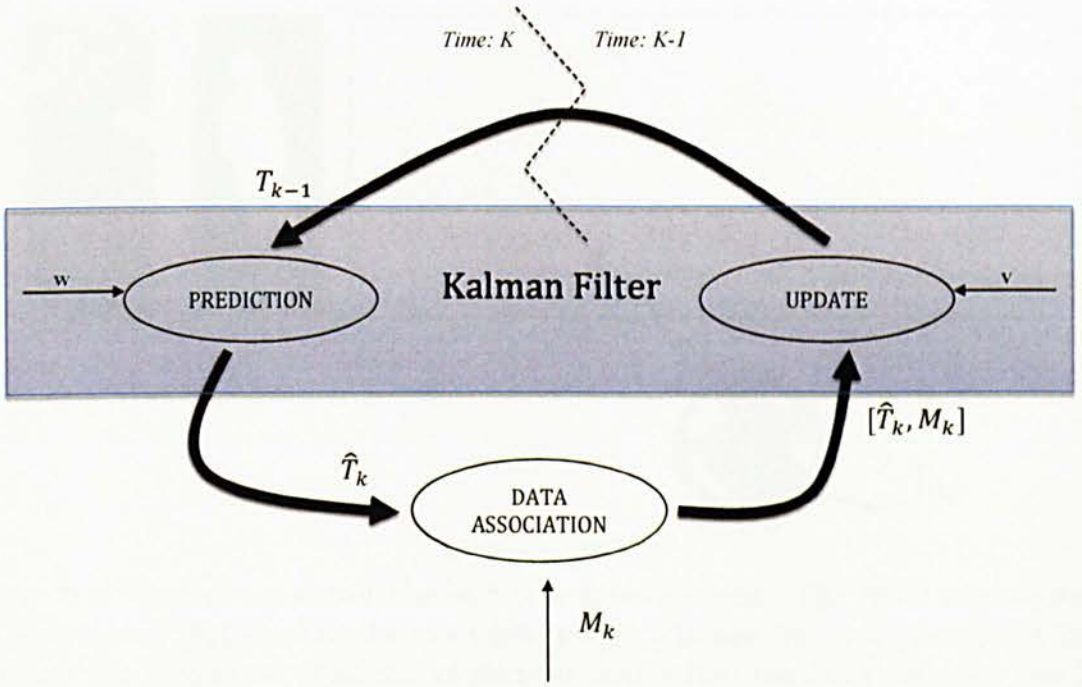


Figure 5.2: Recursive cycle of Kalman Filter: Prediction, Data Association and Update. Note that T and M refer to the set of targets and measurements.

should be capable to readily distinguish one person from another. Moreover, they should facilitate mechanisms for updating and comparison with other people's models. In this section two appearance models of increasing complexity are analysed.

5.2.2.1 Spatial appearance model

The spatial appearance model of a target is built over the same dimensions of the tracking space i.e. MoA. No colour information is used, only the physical extent.

Model construction

The spatial model of a target is defined with a probability density function (PDF), which is built using the projections of the target points. The PDF is modelled with a Gaussian distribution, where the mean refers to the centroid of the projections, and the covariance defines the physical extent of the target in the tracking space – see figure 5.3.

At this point it is essential to differentiate between the target state used in the Kalman filter and the spatial model of the target. Both are defined with a Gaussian distribution to define their location in the tracking space, and both share the same mean position. However, their covariances are conceptually and physically different. As just described, the covariance of the spatial model refers to the physical extent of the points that conform the target, while the covariance of the target state refers to the uncertainty of the mean position.

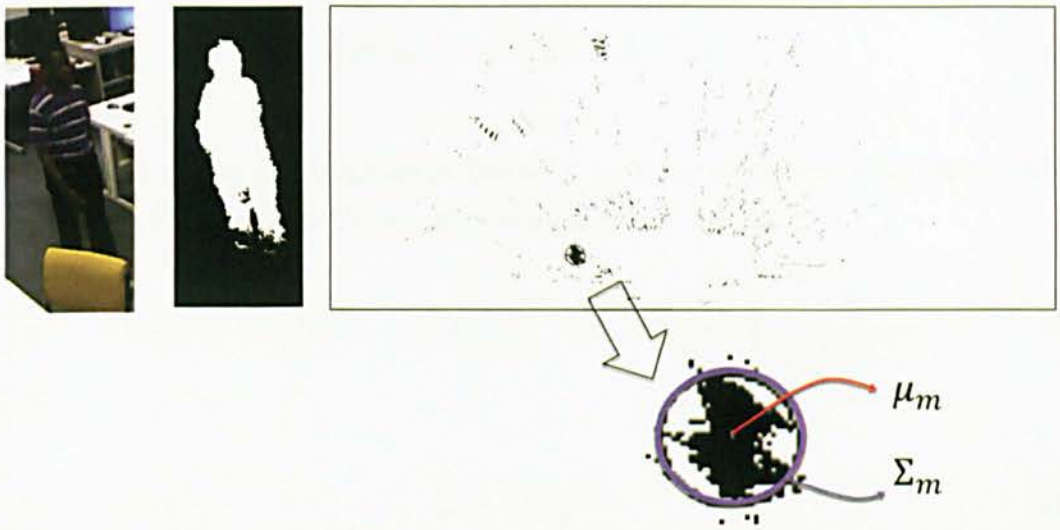


Figure 5.3: Spatial appearance model of a measurement. The RGB representation of the target is displayed in the most left image, followed by its segmentation in the IPS and the projection of all its constituent points into the MoA. Besides, the MoA projection is enlarged to present the spatial model of the measurement (mean and covariance of the projections distribution).

Model update

This model requires to be constantly updated at every time step since the person moves through the tracking space. The update process consists of replacing the target model with the associated measurement model and propagate the mean via the prediction equation of Kalman filter (5.3). Note here that the target might not get associated with any measurement at a particular time. In this case the spatial model only undergoes the prediction of the mean, maintaining the same covariance.

Similarity function

The similarity between two spatial models can be estimated by the distance between their Gaussian distributions. In this work the Bhattacharyya distance is used to assess the similarity between the two models.

The Bhattacharyya distance is a popular measure that generates a value not only in terms of the separation of means but also with respect to their shapes. The general equation for continuous PDFs is as follows:

$$D_B(t, m) = \sqrt{1 - \rho(t, m)} \tag{5.18}$$

where t and m are the Gaussian PDFs of a target and a measurement respectively, and $\rho(t, m)$ is the similarity measure between them defined as

$$\rho(t,m) = \int \sqrt{t(x), m(x)}dx \tag{5.19}$$

The closed form of the Bhattacharyya distance for two multivariate Gaussian distributions $t \sim N(\mu_t, \Sigma'_t)$ and $m \sim N(\mu_m, \Sigma'_m)$ is defined as:

$$D_B(t,m) = \frac{1}{8}(\mu_t - \mu_m)^T \Sigma^{-1}(\mu_t - \mu_m) + \frac{1}{2}\log\left(\frac{|\Sigma|}{\sqrt{|\Sigma'_t||\Sigma'_m|}}\right) \tag{5.20}$$

where

$$\Sigma = \frac{\Sigma'_t + \Sigma'_m}{2} \tag{5.21}$$

$D_B \in [0, \infty)$ is converted into a similarity value normalized between 0 and 1, where 1 indicates maximum similarity as follows:

$$\mathcal{S}(t,m) = e^{-D_B(t,m)} \tag{5.22}$$

5.2.2.2 Multi-part height and colour model: Chromograms

To achieve more discriminative results, a multi-part model defined on the height in the 3D space and colour dimensions is proposed in this section. The target is represented in four dimensions: three dimensions for colour: red, green and blue (R,G,B); and one dimension for the absolute height (h) of the person. The colour information is retrieved from the RGB camera, and the height from the vertical dimension of the target 3D points. It is especially intended to handle occlusion situations and being robust to scale changes. This model is referred to in this work as a **chromogram**.

Model construction

Chromograms consist of a histogram over the height dimension augmented with colour information. The histogram is binned in n equal ranges of height, and each bin stores the number of person's 3D points that fall in that range. In addition, each bin is associated with the colour distribution of its constituent points, which is modelled with a three dimensional Gaussian PDF (R,G,B) defined with the mean and the covariance – see figure 5.4. Chromograms can be thought of as representations that lie half-way between templates [182, 183] and histograms [111, 184]. They combine the advantages of templates maintaining some spatial information (height), and keep, at the same time, the computational requirements low by using a histogram structure. The concept and name of the model are inspired by the work of Birchfield and Rangarajan [75]

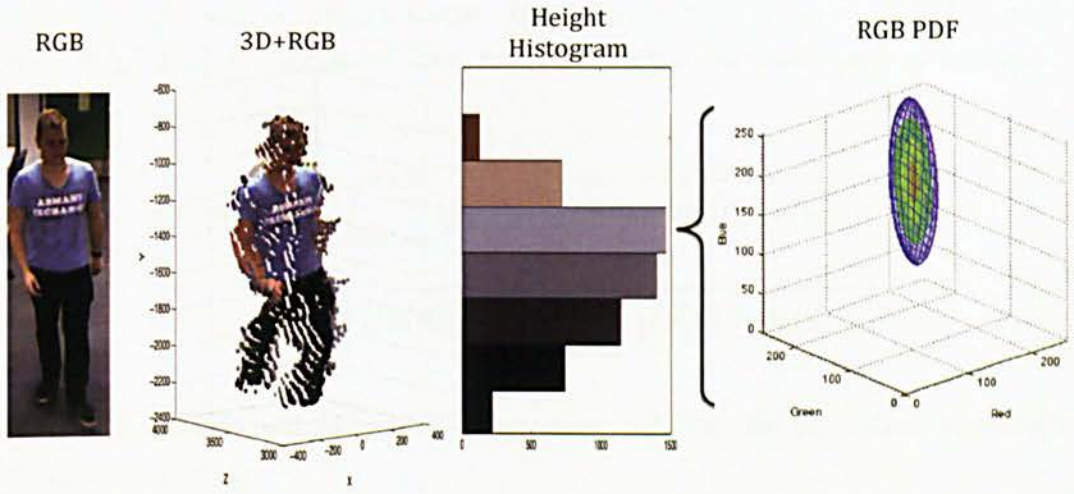


Figure 5.4: Chromogram of a person. From left to right: 2D RGB representation of the person; 3D RGB distribution of the person's points; Height histogram with 8 bins of 25 cm. each. (each bin is coloured with the mean of the RGB distribution at that bin); Three-dimensional Gaussian distribution in the colour space (mean and covariance) of bin fifth. Note that for visual purposes only the colour PDF of one bin is represented.

where they proposed "spatiograms". Their structure is similar to the one presented here, however the histogram is computed over the colour dimension instead, and is augmented with a spatial PDF. Chromograms are expected to be more effective in the presence of occlusions since the division is made on the height dimension. The size of the divisions was set empirically to 25 cm. with a total of 8 bins as a compromise between resolution and computational load.

Model update

Changes are expected in the appearance of people during the sequence, especially when they move between cameras. To cope with these changes and avoid loosing tracks, the targets' chromograms must be updated.

A target's chromogram is updated bin-wise with the associated measurement's chromogram every 10 frames following a simple rule; each bin of the target's chromogram (i.e. height and colour Gaussian PDF) is replaced by the measurement's bin provided that the measurement's chromogram contains data in that bin. In other words, if the measurement does not have any points within the height range of the bin, it is assumed to be temporally occluded and therefore should not be used for updating the target.

Similarity function

The similarity between two chromograms is computed using the metric proposed by Conaire et al. [185]. Originally the metric was intended for spatiograms. However it can be easily adapted for chromograms. Using the original terminology, the similarity

between a target and a measurement chromogram $t = \{n, \phi \sim N(\mu, \Sigma)\}$ and $m = \{n', \phi' \sim N(\mu', \Sigma')\}$ is calculated from the Bhattacharyya coefficient as follows:

$$\begin{aligned} \rho(t,m) &= \sum_{b=1}^{B} \int_{x=-\infty}^{x=\infty} \sqrt{p(\phi_x|n_b)p(n_b)p(\phi'_x|n'_b)p(n'_b)}dx \\ &= \sum_{b=1}^{B} \left[\sqrt{p(n_b)p(n'_b)} \int_{x=-\infty}^{x=\infty} \sqrt{p(\phi_x|n_b)p(\phi'_x|n'_b)}dx \right] \end{aligned} \tag{5.23}$$

Following the original paper [185], equation 5.23 can be simplified yielding to the following closed form:

$$\rho(t,m) = \sum_{b=1}^{B} \sqrt{n_b n'_b} \left[8\pi |\Sigma_b \Sigma'_b|^{\frac{1}{4}} N(\mu_b; \mu'_b, 2(\Sigma_b + \Sigma'_b)) \right] \tag{5.24}$$

where $\rho(t,m) \in [0,1]$, where 1 indicates maximum similarity, and $N(\mu_b; \mu'_b, 2(\Sigma_b + \Sigma'_b))$ is the probability of μ_b with respect to the Gaussian PDF $N(\mu'_b, 2(\Sigma_b + \Sigma'_b))$.

Issue with chromograms

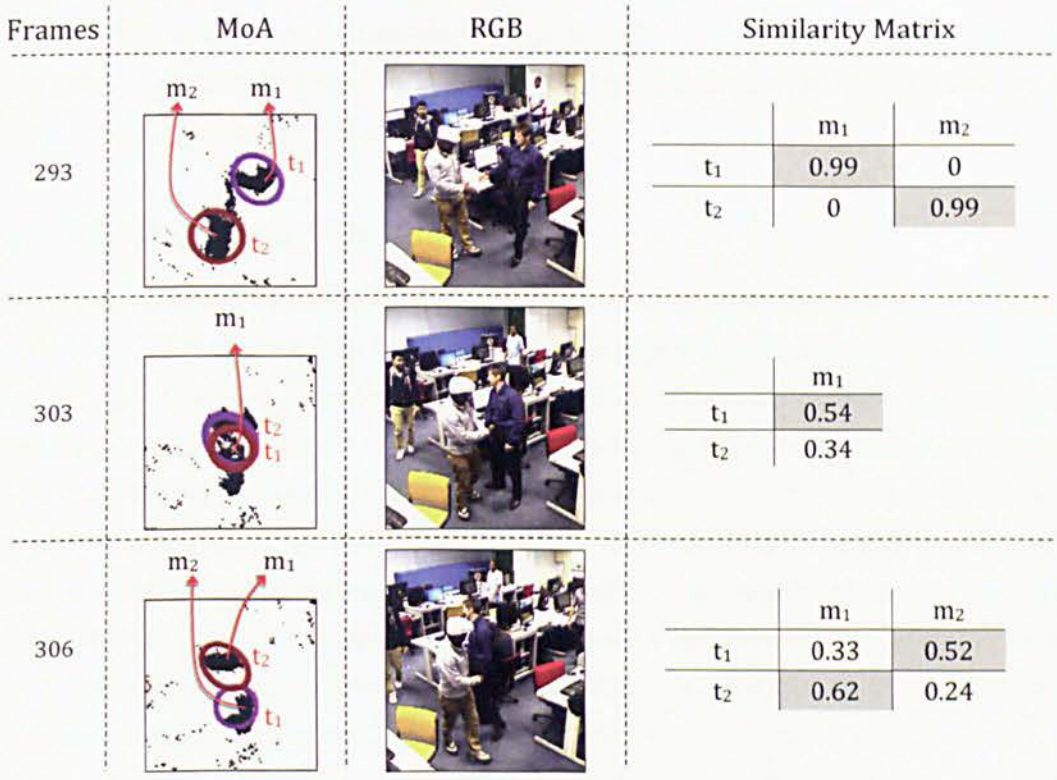
A failure mode has been identified regarding the use of chromograms. When a *merged measurement* is detected, the chromogram of each target involved is compared with the chromogram of the *merged measurement*. This comparison results inevitably in low similarity and erroneous association. As a solution, a mechanism that switches between chromograms and spatial models is proposed. When a *merged measurement* is detected, the similarities between the targets involved and the measurement are computed using only the spatial models. This results in high similarity values, and assures the targets will be associated with the *merged measurement*. Once the *merged measurement* splits, the similarities are computed again using chromograms. Note the fact that as both similarities, spatial model and chromograms, return normalized values between 0 and 1, the switch between models does not affect the association process.

5.2.2.3 Qualitative results

Some qualitative results are presented here which illustrate some failure modes that have been identified for both appearance models.

As expected the spatial model exhibits a poor capacity for discrimination when people are in close proximity. Figure 5.5 presents a case study where two people shake hands, they become merged and then they split again. In this case the spatial model fails to disambiguate the conflict after the *merged measurement*.

The same case is evaluated using chromograms where it is solved correctly as presented in figure 5.6.

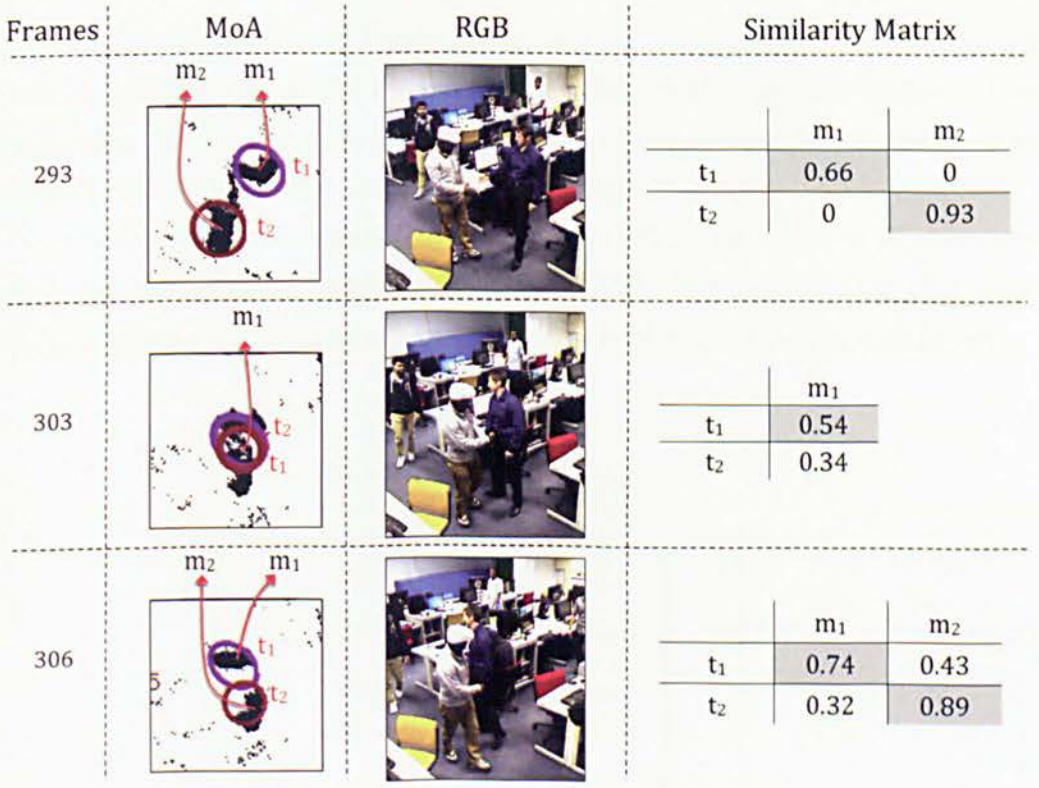


	m_1	m_2
t_1	0.99	0
t_2	0	0.99

	m_1
t_1	0.54
t_2	0.34

	m_1	m_2
t_1	0.33	0.52
t_2	0.62	0.24

Figure 5.5: Key frames of an interaction between two targets. The interaction is resolved incorrectly using the spatial model.



	m_1	m_2
t_1	0.66	0
t_2	0	0.93

	m_1
t_1	0.54
t_2	0.34

	m_1	m_2
t_1	0.74	0.43
t_2	0.32	0.89

Figure 5.6: Key frames of an interaction between two targets. The interaction is resolved correctly using chromograms.

Finally, although chromograms present in general good performance in dense target spaces, they fail with some probability when targets have similar appearance. Figure 5.7 illustrates a situation were the use of chromograms does not discriminate correctly two people wearing similar colours.

5.2.3 Data association

In the previous section two different appearance models were described. In this section those models will be studied in the context of the data association problem.

Multi-target tracking applications require an intermediate process to associate the measurements available at any given time with the active tracks, which is known as the data association problem. The associated measurement will be used to update the target estimation (equation 5.4). Solving the data association problem is not trivial, especially in highly dense target environments, when the number of targets is unknown and variable over time, when spurious measurements are present in the scene, or when there are temporary disappearances of targets due to occlusion. Further uncertainties could appear if split and *merged measurements* are considered.

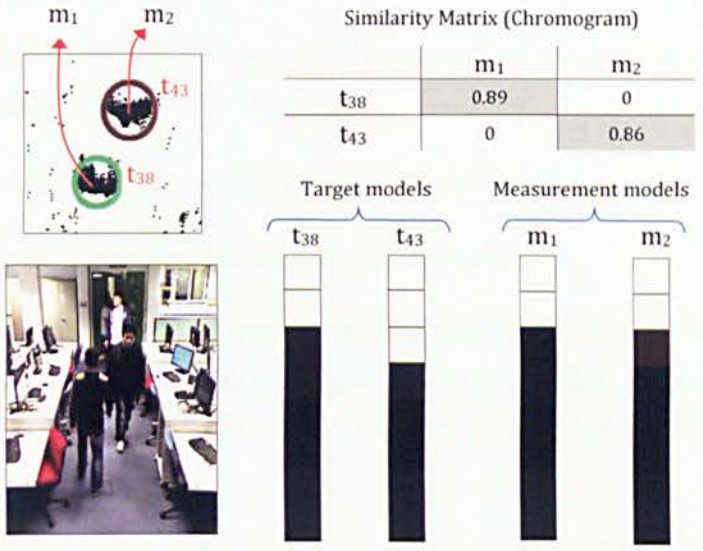
Solving the data association requires the comparison between targets and measurements with a function based on the appearance model of targets and measurements:

$$s_{i,j} = \psi(t_i, m_j) \tag{5.25}$$

where $\psi(t_i, m_j)$ compares the i^{th} target model and the j^{th} measurement model according to the similarity function of the appearance model. Both, the appearance model and the comparative function determine the capacity of the system to discriminate targets, and therefore the performance on the data association process.

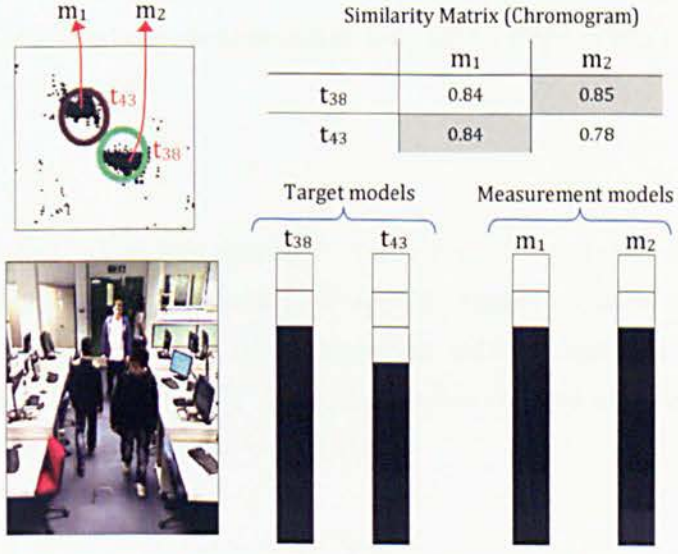
Using equation 5.25 a similarity matrix (Ψ) is built, which relates all measurements (columns) with all targets (rows) - see equation 5.26. The objective is to obtain from this matrix a set of associations where the sum of all similarities is maximised.

$$\Psi = \overbrace{\begin{matrix} 1 & 2 & 3 & \cdots & N_m \end{matrix}}^{\text{measurement (j)}} \begin{pmatrix} s_{1,1} & s_{1,2} & s_{1,3} & \vdots & s_{1,N_m} \\ s_{2,1} & s_{2,2} & s_{2,3} & \vdots & s_{2,m} \\ s_{3,1} & s_{3,2} & s_{3,3} & \vdots & s_{3,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{N_t,1} & s_{N_t,2} & s_{N_t,3} & \vdots & s_{N_t,N_m} \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ \vdots \\ N_t \end{matrix} \Big\} \text{target (i)} \tag{5.26}$$



(a) Frame 779. Target 38 and 43 are similar. Each target produces an independent measurement, m_1 and m_2 respectively.

(b) Frame 782. Targets 38 and 43 produce a single *merged measurement*. Note that appearance similarity between a target and a *merged measurement* does not produce discriminative results. Instead spatial similarity is considered during the merge.



(c) Frame 786. Targets 38 and 43 split from the *merged measurement*. Data association based on chromograms fails due to the high similarity between targets appearance.

Figure 5.7: Sequence of two similar-looking targets crossing each other. The interaction is incorrectly resolved using chromograms.

For some applications the process can be simplified if the following two classical assumptions are made:

- A measurement is generated by one target maximum.
- A target can generate maximum one measurement.

However, in real applications due to limitations in the resolution and quality of the sensors these assumptions do not necessarily hold. In fact, when tracking multiple people in an indoor scene, the following two situations happen with relatively high frequency:

- *Merged measurements*: These measurements are produced when two or more targets are so close that only one measurement is produced for both of them.
- Split measurements: Due to partial occlusions a target produces more than one measurement.

A complex situation arises after a *merged measurement*, when the targets involved separate and the resultant measurements have to be re-associated with their original targets. As these situations are very common in the scenarios envisaged in this work, they are especially treated (see section 5.2.3.5) and independently evaluated.

Under these uncertain conditions, the complexity of the process grows exponentially with the number of targets and measurements involved, therefore approximations need to be considered. One of the most common approaches is to define areas with high probabilities of finding the true measurement for the corresponding target, these regions are often referred to as **gates**.

Gates

In the tracking context, gates are employed to reduce the number of possible combinations between targets and measurements. For every target an area around its predicted measurement is defined and only measurements within that area are considered as possible associations (see figure 5.8). The gate area is defined on the MoΛ based on the square of the Mahalanobis distance as follows:

$$v^T S^{-1} v \leq \gamma \tag{5.27}$$

where S is the innovation uncertainty (equation 5.5), γ is the spatial threshold that defines the gate volume, and v is the innovation term as follows:

$$v = m - \hat{m}_k \tag{5.28}$$

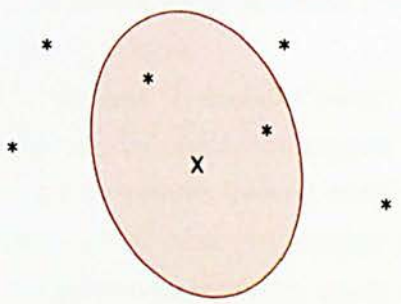


Figure 5.8: Gate area in the MoA. The cross (X) is the predicted measurement; the stars (*) are the available measurements; and the shaded region defines the gate.

where m is the position of the measurement, and $\hat{m}_k$ is the predicted measurement of the target at time k – see equation 5.2.

The threshold γ is defined from the "chi-squared" tables, since the Mahalanobis distance of samples drawn from a Gaussian distribution are chi-squared distributed, with n_z degrees of freedom.

Using gates, the similarity matrix Ψ can be now constructed as follows:

$$s_{i,j} = \begin{cases} -1 & \text{if measurement j is outside the gate of target i.} \\ \psi(t_i, m_j) & \text{otherwise.} \end{cases} \tag{5.29}$$

It is important to notice a potential problem that can arise with gates. When a target is not associated with any measurement (temporally occluded), its uncertainty (and associated gate) starts growing. As a consequence, measurements from nearby targets fall within this large gate occasionally leading to erroneous associations.

To reduce the effect of large gates, one possibility is to limit the maximum size of the gate. Another possibility is to introduce ordering within the association process, prioritising targets with smaller covariances.

Although the use of gates reduces the number of possible combinations, ambiguous situations can still arise. For instance, when two or more measurements fall within the same gating area, or when a single measurement falls in the intersection of two different gates. For those situations association techniques are still required.

Choosing a data association methodology

The problem of data association has received considerable attention in the community and sophisticated techniques such as the joint probabilistic data association filter (JPDAF) [88, 186] and the multi hypothesis tracking (MHT) [90] have been studied

and extended over the years. However these methodologies present some limitations that make them not suitable for this work.

JPDAF was particularly designed to handle noisy environments with spurious measurements, which is not the case in the environment proposed. The segmentation module presented in section 4.4 produces indeed very few spurious measurements. Moreover, JPDAF assumes a known and constant number of targets. MHT is considered the best solution to the data association problem since it explores and maintains all hypotheses. MHT inevitably requires a high computational load and even with the use of approximation techniques (e.g. pruning, clustering) the approach struggles to meet real-time requirements. Furthermore, MHT is a batch method, which means that in the presence of conflicts the decision is delayed in time until more information is available.

A more appropriate data association method for this project is the nearest neighbour standard filter (NNSF). The NNSF is computationally efficient, takes decisions at every time step and its performance has been proven satisfactory in a range of problems [82, 84]. Three variations of increasing complexity of the NNSF are explored in this project.

5.2.3.1 Iterative Nearest Neighbour

Iterative Nearest Neighbour (INN) is one of the simplest methodologies for solving the problem of data association. It is a derivation from the simple nearest neighbour that prohibits a target being associated with multiple measurements [4, 77]. This technique is executed sequentially considering one target at a time.

The procedure consists of the following steps:

1. Build the similarity matrix between targets and measurements – see equation 5.29.

2. Establish an order in which the targets will be associated, e.g. random order, largest first, nearest to the depth sensor first, etc, and choose the first target.

3. Search in the matrix along the corresponding row for the most similar measurement.

4. Eliminate the associated measurement from the similarity matrix (the entire column) to ensure the measurement cannot be associated with another target.

5. If there are still targets available, choose the next target and go back to step 3, otherwise finish the process.

This approach works reasonably well when targets are quite separate from each other. In addition, it requires low computational time and resources. However, when targets

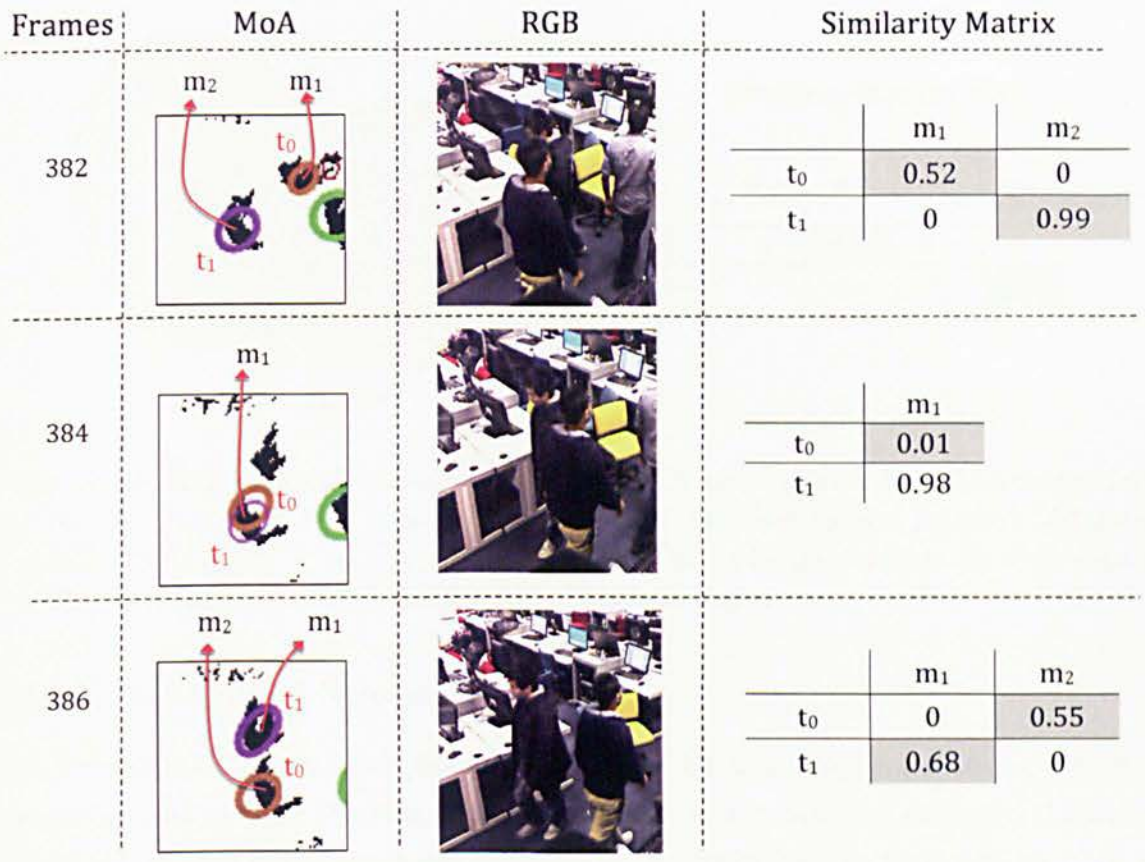
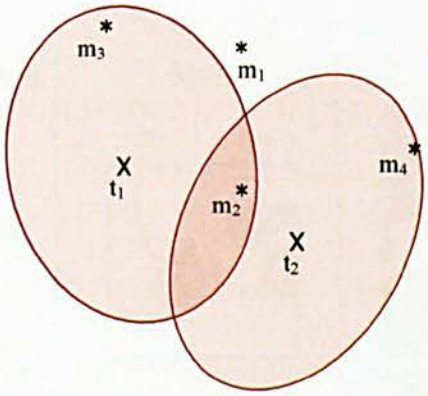


Figure 5.9: Three key frames of an interaction between two targets which is incorrectly resolved using INN. Note that the thinner ellipse of target 1 at frame 384 indicates that the target is being unassociated. The association result at frame 384 is erroneous due to the chosen order in which the two targets are associated (t_0 first and then t_1).

get close this approach does not perform well, mainly because it is highly dependent on the order in which the targets are associated. Figure 5.9 presents a case of study where this technique actually fails. In this example two targets (t_0 and t_1) get involved in an interaction. t_0 is more distant and at some point it gets occluded by t_1, not producing any measurement. In addition, as the two targets are very close, the measurement produced by t_1 falls within the gate of both, allowing the measurement to get associated with either of them. t_0 is first evaluated and becomes incorrectly associated with the measurement, leaving t_1 unassociated. Figure 5.10 shows a hypothetical situation where INN would obtain a non-optimal solution. These cases reveal a clear limitation of the methodology, which is the dependency on the order of association. The performance of INN could improve if a meaningful order of associations is chosen. For example, in this case it could compute first closer targets, assuming they are less likely to be occluded by others.

In the next section a more advanced algorithm for association is presented, which aims to cover the identified weakness of INN.



Similarity Matrix (INN)

	m_1	m_2	m_3	m_4
t_1	0	0.8	0.6	0
t_2	0	0.9	0	0.2

Figure 5.10: Hypothetical situation of 2 targets (t_1 and t_2) and 4 measurements (m_1, m_2, m_3 and m_4). The INN is applied to solve the data association problem obtaining a non-optimal result $r = \{[t_1, m_2], [t_2, m_4]\}$. The optimal solution in this case is $r' = \{[t_1, m_3], [t_2, m_2]\}$ where the total similarity is higher.

5.2.3.2 Suboptimal Nearest Neighbour

The Suboptimal Nearest Neighbour (SNN) is a data association technique that is commonly used in early tracking literature [80–82]. It is considered suboptimal because it does not explicitly recover a global solution i.e. maximize the total similarity of all associations.
The procedure consists of the following steps:

1. Create the similarity matrix between targets and measurements at the current time using equation 5.29.

2. Choose the highest similarity value in the matrix and create the association between the target (row) and measurement (column) involved.

3. Remove the associated target row and measurement column from the similarity matrix.

4. If there are still targets available go back to step 2, otherwise finish the process.

In general, SNN outperforms INN because it is not dependent on the order of the targets. For comparison purposes the same case of study presented in figure 5.9 is analysed again using the SNN method instead (see figure 5.11). This time the interaction is correctly resolved.

However, SNN does not always achieve the correct result primarily because a global solution is not explicitly sought, i.e. it does not aim to maximize the similarity of all associations. SNN is expected to fail in situations such as the one depicted in figure 5.12.

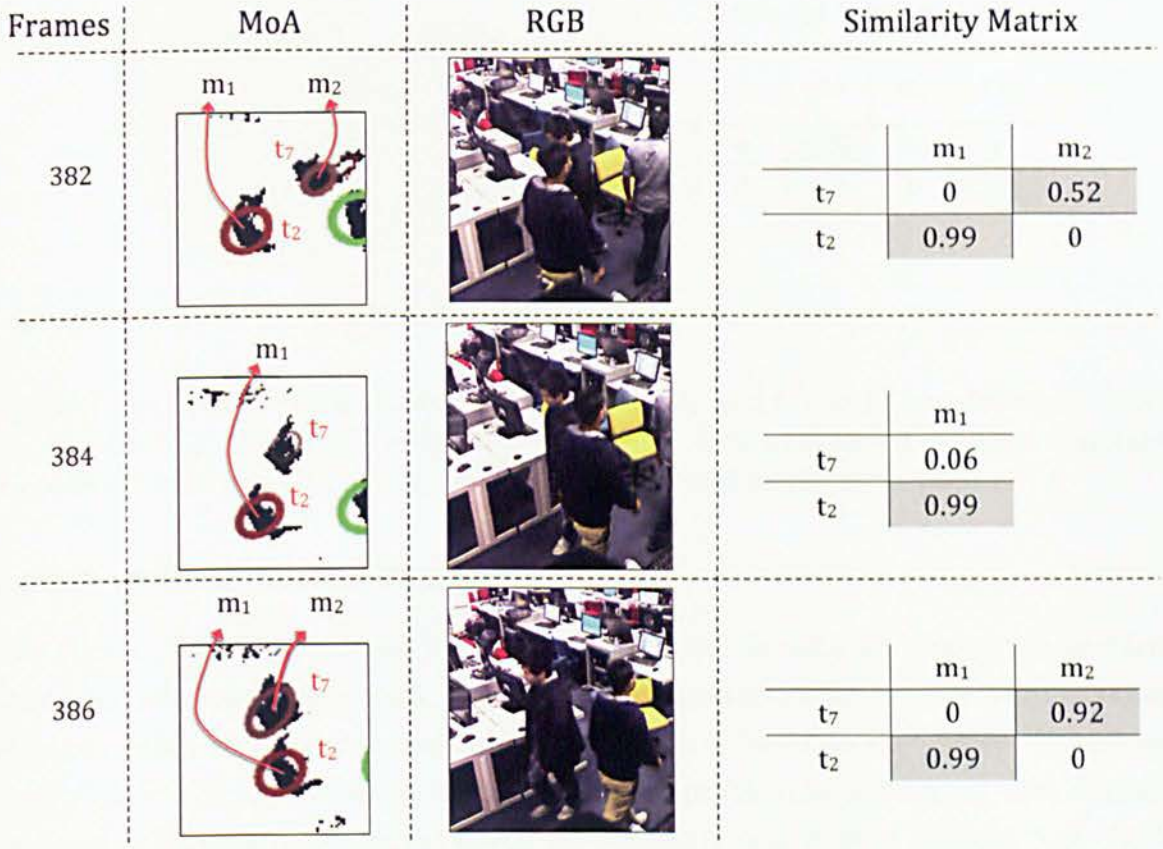
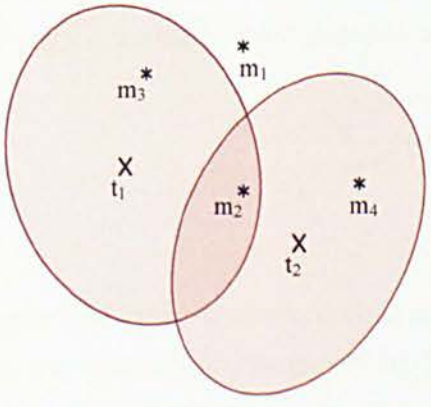


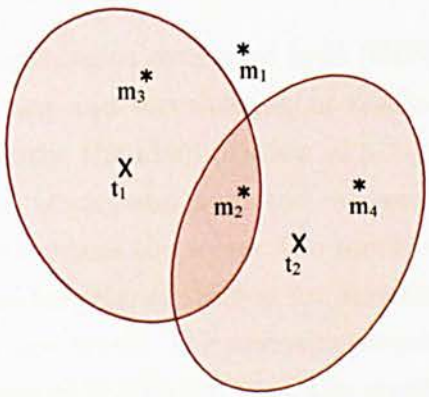
Figure 5.11: Three key frames of an interaction between two targets, which is correctly resolved using SNN.



Similarity Matrix (SNN)

	m_1	m_2	m_3	m_4
t_1	0	0.8	0.2	0
t_2	0	0.9	0	0.8

Figure 5.12: Hypothetical situation of 2 targets (t_1 and t_2) and 4 measurements (m_1, m_2, m_3 and m_4). The SNN is applied to solve the data association problem obtaining a non-optimal result $r = \{[t_1, m_3], [t_2, m_2]\}$. The optimal solution in this case is $r' = \{[t_1, m_2], [t_2, m_4]\}$ where the total similarity is higher.



Similarity Matrix (GNN)

	m_1	m_2	m_3	m_4
t_1	0	0.8	0.2	0
t_2	0	0.9	0	0.8

Figure 5.13: Hypothetical situation of 2 targets (t_1 and t_2) and 4 measurements (m_1, m_2, m_3 and m_4). The GNN is applied to solve the data association problem obtaining a optimal result $r = \{[t_1, m_2], [t_2, m_4]\}$ where the total similarity is higher.

5.2.3.3 Global Nearest Neighbour

The Global Nearest Neighbour (GNN) is an approach to the data association problem that uses the similarities from all targets and measurements to construct a global solution, which is considered optimal. This solution is based on the popular Hungarian algorithm [83, 84, 187] that solves the assignment problem in polynomial time without the need of an exhaustive search. Given the similarity matrix Ψ of equation 5.26, GNN finds the set of associations that maximizes the total similarity as follows:

$$\arg\min_{x} \sum_{i}^{N_t} \sum_{j}^{N_m} s_{i,j} x_{i,j} \tag{5.30}$$

where N_t and N_m are the total number of targets and measurements respectively; and x defines the set of associations, which applies the restriction that a target can only be associated with a single measurement and vice-versa (equation 5.31):

$$\sum_{i}^{N_t} x_{i,j} = \sum_{j}^{N_m} x_{i,j} = 1; \tag{5.31}$$

In general, GNN requires more computational time than SNN. However with the improved implementation proposed by Munkres [83][2], and depending on the cardinality of targets and measurements, the approach achieves similar execution times.

Figure 5.13 presents a hypothetical situation where GNN finds the optimal solution.

[2]The implementation of Munkres improves the performance of the algorithm achieving a complexity of order $O(n^3)$

5.2.3.4 Initialization and termination of tracks

The methodologies presented here (INN, SNN and GNN) do not explicitly handle the initialization and termination of tracks and therefore additional actions need to be taken. Firstly, the identification of potential new tracks is determined attending to the number of tracks and detected measurements at each time step. In general, when a new target enters the scene, the number of detected measurements is larger than the number of targets, signalling an unassociated measurement, which is considered as a potential new track. The opposite occurs when a target leaves the scene, the number of measurements is smaller than the number of targets, and therefore one target is left unassociated, which is labelled as a potential finished track. These potential new and finished tracks are analysed independently:

- A potential new track is promoted to actual track if during a certain amount of time exists evidence to support it. In other words, in order to initialize a new track, the target should be associated with measurements for a minimum period of time. The objective of this action is to reduce the number of false new tracks caused by noise measurements.

- A potential terminated track is terminated if it is not associated with any measurement for a certain period of time. This action reduces the number of falsely terminated tracks that are just temporally occluded.

The time span threshold considered for the initialization and finalization of tracks was defined empirically to 1.5 seconds.

5.2.3.5 Issues: Interaction Periods

Interaction Periods (IP) refer to those situations where two or more people produce a single *merged measurement* on the MoA due to their spatial closeness. These periods are normally the result of events such as grouping, handshakes or even just path crossing. After the *merged measurement* the system is expected to correctly re-identify the targets, i.e. the targets ID after the *merged measurement* should be consistent with their IDs before the merge. This re-identification represents an important challenge for the data association module. Note that for the purpose of this project the independent segmentation of targets during a *merged measurement* is not necessary.

The proposed process for handling interaction periods consists of the following steps:

1. Detection of *merged measurements*. A dedicated module has been implemented to identify these special measurements based on area and proximity of targets

2. Multiple associations. The data association module needs to be adapted to allow the targets involved in the interaction to be associated with their common (merge) measurement

3. Targets' state estimation (update and prediction). Targets are predicted normally. However for the update stage, different policies can be adopted.

   - Normal update. The targets involved update their position with the *merged measurement*. This approach is suitable for long-lasting merges such as grouping where the motion of the targets changes during the merge.
   - Non-update: The targets motion is not updated with the *merged measurement* aiming to preserve the motion model of the targets. This approach to updating is recommended for short interactions such as path crossing events.

4. Continue to step 1 and repeat the entire process until the targets involved in the interaction split. At this point the data association and target estimation are applied normally.

This process is independent from the association methodology and the object modelling used. Nonetheless, its performance relies highly on a correct detection of *merged measurements* at step 1. This detector module is described in further detail next.

Merged measurement detector

The *merged measurement* detector is an external module responsible for the recognition of measurements produced by more than one target. These measurements appear when people get spatially close and their projections on the ground plane become connected in a single blob.

The identification of *merged measurements* is based on two features: area of the measurement and number of close targets. A measurement is labelled as a merge if it satisfies the two following requirements: its area is larger than a defined threshold and more than one tracked target are in close proximity.

Area restriction. The idea of filtering measurements by area is motivated by the assumption that in general, *merged measurements* are larger than single-target measurements. This filtering is performed in the RPS, where the areas of the blobs over the entire range of the space are more homogeneous than in any other space.

The optimal value for the threshold is learned using a training dataset[3] and its

[3]The training dataset is the same dataset used to adjust the parameters for the people detection algorithm.

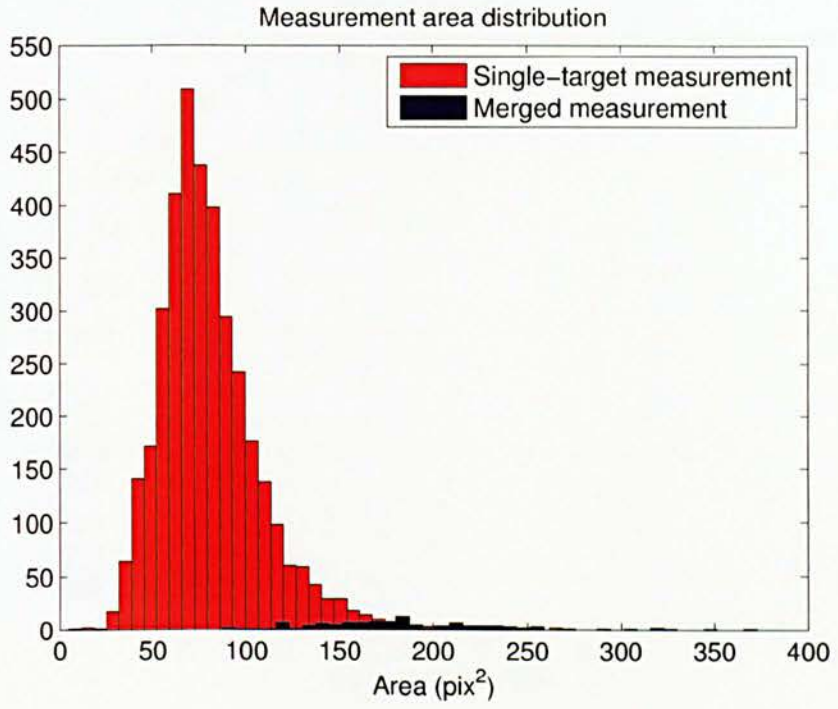


Figure 5.14: Distribution of the areas of *merged measurements* and single-target measurements in the RPS. The samples have been manually labelled from a training sequence on the RPS. In each frame of the sequence the detected measurements (connected components) were annotated with their area and category – i.e. merged or single.

corresponding ground truth of *merged measurements*[4]. Figure 5.14 plots the distribution of the measurement areas labelled by category i.e. single-target and *merged measurement*. From the plot, it is clear that the two classes are completely inseparable using only the area. In addition, the number of samples of *merged measurements* is significantly smaller than the number of samples from the other class i.e. only 0.03% of the samples are *merged measurements*. In order to select an appropriate threshold an empirical approximation based on the popular ROC curve is employed – see figure 5.15. The ROC curve is a visual way to compare the performance of an algorithm for different parameter values. It is represented in a two dimensional plot where the vertical axis defines the true positive rate ($TPR = \frac{TP}{TP+FN}$) and the horizontal axis represents the false positive rate ($FPR = \frac{FP}{FP+TN}$). The idea is to plot the results for a set of different area thresholds and fit a curve to the data. The optimal values are on the most top-left part of the curve, where the ratio between TPR and FPR is maximum.

Proximity of targets. Once a measurement has been defined as larger than the threshold, the next step is to identify the number of nearby targets. If it has more than one, then it is considered a merge.

[4]The *merged measurement* ground truth was manually created by an operator. Every *merged measurement* was labelled and stored in a file along with its area in the RPS (pixels^2)

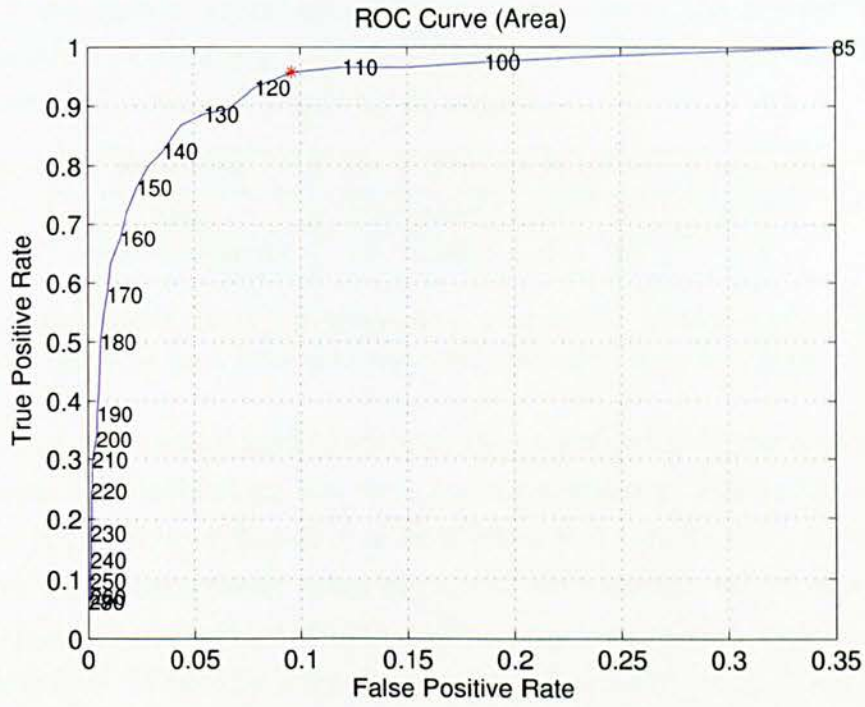


Figure 5.15: ROC curve that presents the evaluation for detecting *merged measurements* with different area thresholds. The optimal value is identified as the point of the curve closest to the top-left corner (115 pixels^2). Note, that the axis are in different scales.

The process followed is similar to the idea of the gates used during the data association stage. The search window in this case is estimated using the scatter matrix of the measurement in the RPS. If more than one target's predicted position falls within this search area, the measurement is labelled as a merge.

Evaluating the *merged measurement* detector

The *merged measurement* module has been independently assessed on the same dataset used for the evaluation of people segmentation – see section 4.5.4. The ground truth in this case identifies the *merged measurements* and the single measurements in the sequence. To evaluate the performance of the system the following metrics are computed:

- True Positives: number of merged measurements correctly identified as merged.
- False Positives: number of single measurements incorrectly labelled as merged.
- True Negatives: number of single measurements correctly labelled as single.
- False Negatives: number of merged measurements incorrectly labelled as single.
- F1-score: harmonic mean of precision and recall.

Table 5.1 presents a comparison of the performance of the *merged measurement* detector described in section 5.2.3.5 using first only the area filter, and second, using the combined filter: area and proximity of targets.

Features	TP	TN	FP	FN	F1-score
Area	109	3338	353	5	0.37
Area+Proximity	84	3681	10	30	0.8

Table 5.1: Comparison of the performance evaluation of the *merged measurement* detector using only the area filter and the combination of area and proximity of targets.

As expected, the overall performance of the combined filter outperforms the area filter. However, it is interesting the fact that the number of FNs is lower in the area-based filter. A possible explanation is that when two targets start to approach and before the *merged measurement* takes place, the more distant target gets occluded by the closer target. The trajectory of the occluded target starts to diverge because it relies only on predictions. When the *merged measurement* actually occurs the diverged target fall outside the measurement search window. As a consequence the targets proximity filter does not hold and the measurement is not identified as a merge.

Failures of the *merged measurement* detector may result in the loss of people in subsequent stages. In particular, the direct consequence of the FPs is that a target's model and location will not be updated with the measurement (depending on the update policy during *merged measurements*). Equivalently, FNs yields only one target to be associated with the *merged measurement*, leaving the rest of the targets involved unassociated or forced to "steal" somebody else's measurement. Although, the combined filter increases the number of FN by a factor of 6, the number of FP are reduced by a factor of approximately 35, which clearly justifies the use of the combined filter.

Modifying the data association algorithm

The data association methodologies introduced in sections 5.2.3.1, 5.2.3.2 and 5.2.3.3 do not allow multiple targets to be associated with the same measurement. Therefore, when a *merged measurement* occurs one or more of the targets involved will not be appropriately associated.

In order to manage these situations an ad-hoc solution is proposed to allow measurements, in this case *merged measurements*, to be associated with all the targets involved. The process consists of the following steps:

1. Extend the similarity matrix Ψ by duplicating the columns that belong to *merged measurements*. The number of duplications is set by the number of targets within the proximity gate.

2. Execute the data association algorithm using the extended similarity matrix Ψ'.

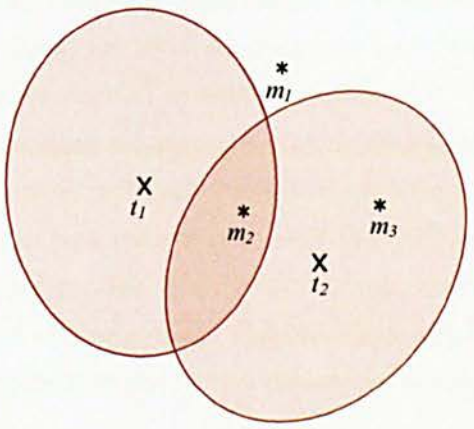


Figure 5.16: Two targets yield a *merged measurement*, m_2.

3. Identify all targets associated with the duplicated columns and associate them with the corresponding measurement.

The situation illustrated in figure 5.16 presents an example involving a *merged measurement*. The process proposed yields the followings similarity matrix (Ψ) and extended similarity matrix (Ψ').

$$\Psi = \begin{array}{c} \overbrace{}^{} \\ \begin{pmatrix} -1 & \psi(t_1, m_2) & -1 \\ -1 & \psi(t_2, m_2) & \psi(t_2, m_3) \end{pmatrix} \end{array} \begin{array}{l} m_1 \quad m_2 \quad m_3 \\ \left.\begin{array}{l} t_1 \\ t_2 \end{array}\right\} \end{array}$$

$$\Psi' = \begin{array}{c} \overbrace{}^{} \\ \begin{pmatrix} -1 & \psi(t_1, m_2) & -1 & \psi(t_1, m_2) \\ -1 & \psi(t_2, m_2) & \psi(t_2, m_3) & \psi(t_2, m_2) \end{pmatrix} \end{array} \begin{array}{l} m_1 \quad m_2 \quad m_3 \quad m_2' \\ \left.\begin{array}{l} t_1 \\ t_2 \end{array}\right\} \end{array}$$

when the data association algorithm is applied on Ψ', targets t_1 and t_2 will be associated with measurements m_2 and m_2' respectively.

A thorough evaluation of the multi-target tracking system discussed in this section is presented in chapter 6.

5.3 The Mean-Shift algorithm applied to tracking

In this section an alternative tracking approach is presented – the Mean-Shift algorithm. This algorithm was first applied for tracking purposes by Comaniciu, Ramesh and Meer

[70]. Unlike typical tracking methodologies such as Kalman filter and particle filters, the Mean-Shift approach does not need to perform any data association, which is an important source of errors as argued in section 5.2.3. For that reason the Mean-Shift approach appears as a promising alternative to the multi-target tracking problem.

The Mean-Shift algorithm is normally classified as a data-driven methodology, since it does not segment objects but rather use only the information retrieved from the data itself to build the target model. Mean-Shift tracks a given target by searching for its model in every image of the sequence. Rather than applying an exhaustive search, the frame is efficiently searched towards the direction of the similarity gradient in the tracking space.

The Mean-Shift tracker is considered a computationally efficient method. However it is frequently associated with some limitations especially in multi-target environments. It is easily affected by the inclusion of background data into the model or the interferences produced by similar targets during interaction periods. Targets are modelled using histograms which have less discriminative power since the spatial information is discarded. The standard Mean-Shift does not support changes in the scale and orientation of the targets over time. In addition, it does not provide mechanisms for the automatic initialization of new targets. Due to these limitations the Mean-Shift algorithm is not normally used for tracking multiple targets. Nonetheless, many authors have proposed different approximations to overcome these issues. For example, Gao and Liu [103] applied a previous background subtraction operation to reduce the number of distractions. Beyan and Temizel [104] used a people segmentation module to allow the automatic initialization of new targets. Leichte et al. [188] improved the appearance model by using multiple color histograms from different views. In this work some modifications are introduced into the Mean-Shift algorithm to address the aforementioned weaknesses.

5.3.1 The standard Mean-Shift approach

Mean-Shift is a non parametric method for climbing density gradients. It is a versatile technique that can be applied in segmentation, clustering or for tracking among other computer vision tasks. The method was initially proposed by Fukunaga [189], but it was not used for computer vision tasks until late 90's. Mean-Shift was originally proposed for tracking purposes by Comaniciu et al. [70] and it has been widely used since then.

The Mean-Shift tracker is presented in this work as an alternative approach to the common and widely used tracking methodology outlined in section 5.2.1. The most relevant feature of the Mean-Shift tracker for the current study is that it does not require data association. The objective here is to evaluate Mean-Shift as an alternative "data association-free" tracker in multi target environments and to compare the results

with those that need a data association module i.e. Kalman Filter.

A model of the target is built in the initial frame and Mean-Shift searches the image in an optimal way looking for the location of maximum similarity with the target model. The outline of the algorithm is described next using the notation of the original paper:

Target model. When the object first appears in the scene it is initialized by building a colour histogram appearance model $\hat{q} = \{\hat{q}_u\}_{u=1...m}$ where m is the total number of bins which sum to unity. i.e. $\sum_{u=1}^{m} \hat{q}_u = 1$. Each bins defines the target probability for that particular colour range.

$$\hat{q}_u = C \sum_{i=1}^{n} k(||x_i||^2)\delta[b(x_i) - u], \quad C = \frac{1}{\sum_{i=1}^{n} k(||x_i||^2)} \tag{5.32}$$

where $\{x_i\}_{i=1...n}$ are the pixel locations of the target; k is a spatial-kernel profile that weights more highly pixels closer to the centre position; $b(\cdot) : \mathbf{R}^2 \rightarrow \mathbf{R}$ is a function that associates a pixel with its corresponding histogram bin with regard to its colour information; δ is the Kronecker delta function; and C is a normalization term that makes the summation of all bins equal to one. The only restriction regarding the kernel is that it must be convex and monotonically decreasing [70].

Candidate model. In the next frame the search for the target starts from the previous location y_0. At that position a candidate model is constructed in a similar fashion $\hat{p}(y_0) = \{\hat{p}_u(y_0)\}_{u=1...m}$

$$\hat{p}_u = C_h \sum_{i=1}^{n_h} k\left(\left|\left|\frac{y_0 - x_i}{h}\right|\right|^2\right)\delta[b(x_i) - u], \quad C_h = \frac{1}{\sum_{i=1}^{n_h} k(\left|\left|\frac{y_0 - x_i}{h}\right|\right|^2)} \tag{5.33}$$

where h is the bandwidth that defines the size of the candidate i.e. window size; $\{x_i\}_{i=1...n_h}$ are the candidate pixels; and C_h is the normalization constant. Candidate and target models are compared using the Bhattacharyya coefficient.

$$\rho[\hat{p}(y_0), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y_0)\hat{q}_u} \tag{5.34}$$

The objective is to find a candidate in the current image frame that minimizes the Bhattacharyya coefficient

$$\arg\min_{y} \rho[\hat{p}(y), \hat{q}]$$

Mode seeking. This global target search strategy is optimized using the Mean-Shift algorithm that seeks iteratively the mode of a probability distribution along the density gradient direction. This distribution has been previously computed from the target's colour histogram. This approach requires the computation of the pixels weights within a search window $\{w_i\}_{i=1...n_h}$, which are obtained based on their probability of belonging to the target model as follows:

$$w_i = \sum_{u=1}^{m} \delta[b(x_i) - u] \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_o)}} \tag{5.35}$$

Then, the new location y^* is located at the mode of this probability space, which is obtained via Mean-Shift equation 5.36:

$$y^* = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left|\left|\frac{y_0 - x_i}{h}\right|\right|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left|\left|\frac{y_0 - x_i}{h}\right|\right|^2\right)}. \tag{5.36}$$

where $g(\cdot) = -k'(\cdot)$.

If the magnitude of the Mean-Shift vector ($\vec{y} = y^* - y_0$) is higher than a certain threshold, then the centre is updated to the new position $y_0 \leftarrow y^*$ and the process is repeated. Otherwise the search finishes and the new target position at the current frame is set at y_0, which represents a local maximum of the PDF.

Although the Mean-Shift tracker is considered to be efficient and robust, there are a number of limitations that reduce its performance on certain situations especially when tracking multiple targets. Those limitations are described next.

5.3.1.1 Limitations of Mean-Shift in multi-target environments

The Mean-Shift tracker was originally designed for single target tracking and its use in multi-target environments is limited due to the following issues:

- It is highly sensitive to distractions produced by other targets during interaction periods or from the background. This issue is normally associated to the fact that rigid primitives e.g. bounding box are used to delimit non-rigid targets e.g people, allowing the inclusion of outliers in the target model.

- The appearance model is not discriminative enough to distinguish people from one another in complex situations. The majority of the Mean-Shift implementations found in the literature as well as in the original paper, model the target with a colour histogram, which is sometimes simplified to a 1D histogram. Histograms are in general very convenient structures to work with due to their simplicity, fast computation and especially because they are robust to rotations and non-rigid transformations. Nonetheless, they are frequently criticized for not preserving

the spatial dimension of the data, which implies a lower discriminative capability [74, 75].

- The original Mean-Shift approach does not provide with adequate mechanisms to deal with changes in the scale of targets. The authors offered an ad-hoc solution far from being ideal. They try three different bandwidth sizes and choose the one that fits the model best. This is an important issue particularly in visual surveillance applications where targets usually move throughout the whole field of view varying their size in the projected image.
- It does not provide with mechanisms for the automatic initialization and termination of tracks. In real surveillance scenarios this is a chief aspect since in general the number of targets is unknown and varies unpredictably over time.

In the next section is presented an enhanced Mean-Shift tracker that aims to deal with all these limitations.

5.3.2 Enhanced Mean-Shift algorithm

In this section the modifications introduced to the original Mean-Shift implementation are presented, which aim to address the aforementioned limitations.

Chromogram appearance model

One of the main weaknesses of the standard Mean-Shift tracker is the use of a poor discriminative model, namely a 1-dimensional histogram. One of the enhancements proposed is the use of the chromogram appearance model as presented in section 5.2.2.2. Many authors noted this weakness before and different models have been introduced in the past. Leichter et al. [188] uses a combination of multiple colour histograms taken from different views. Zhang et al. [190] learn a model based on SURF features. However, most of these models are complex to compute and evaluate. Chromograms on the other hand are simple models based on a histogram structure, but also discriminative and effective during occlusions situations since they are constructed with adjacent parts. In addition, unlike most of the models built on the image plane [70, 75], chromograms are robust to changes in scale.

Clearing target data after evaluation

The main challenge of applying Mean-Shift in a multi target environment is the interference produced by other targets especially during interaction periods i.e. data that belongs to one person is used in the tracking of other people due to their proximity. As a consequence multiple targets end up following the same person. In the literature

some authors perform foreground object extraction, which requires an additional stage for data association. In the work of Beyan and Temizel [104] occlusions, merges and splits are resolved through a data association process. In this work a completely data association-free tracking methodology is presented. To handle these problematic situations the pixels of a person are removed from the tracking space just after being evaluated, so the pixels of that person cannot distract the rest of the people. Using this approximation the order in which people are evaluated is critical. A meaningful order has been established giving priority to people closer to the cameras since they are less likely to be occluded. Figure 5.17 illustrates the tracking order for a particular frame in a video sequence where 8 people are involved.

Tracking space over the ground plane: MoA

Unlike the majority of Mean-Shift tracker implementations that define the tracking space on the image plane [113, 188], in this work the plan view MoA is employed instead since it has been shown (section 4.5) to be more effective for solving occlusions. Typically, each pixel of the image plane is weighted with the probability obtained from the histogram appearance model according to the colour or each pixel [70]. This operation is known as "histogram backprojection". This process has been adapted to be used in the plan view MoA. Since multiple points might contribute to the same position in the MoA, the probability of each position $p_{MoA}(u,v)$ is computed as the sum of the probabilities of all the points that project in that location as follows.

$$p_{MoA}(u,v) = \sum_{i=1}^{n} p(x_i(h))p(x_i(c)|x_i(h)) \tag{5.37}$$

where $\{x_i\}_{i=1...n}$ are all points that project into the same location (u,v) in the MoA; $p(x_i(h))$ is the probability of the i^{th} point in the height dimension of the chromogram; and $p(x_i(c)|x_i(h))$ is the conditional probability of the point colour given its height. This probability is computed using the colour Gaussian PDF $(N_h(\mu,\Sigma))$ associated with the chromogram bin of the pixel height.

To speed up the process, in the actual implementation only an area around the last person position is considered. The dimension of this area is set dynamically every frame to be 50% larger than the estimation size of the last person location. This region should cover any possible displacement of the target from the last time step.

Background exclusion

To further improve the performance of Mean-Shift the background pixels are removed from the scene and only the foreground pixels, which are assumed to belong to people, are considered. A foreground segmentation is performed at each time step using

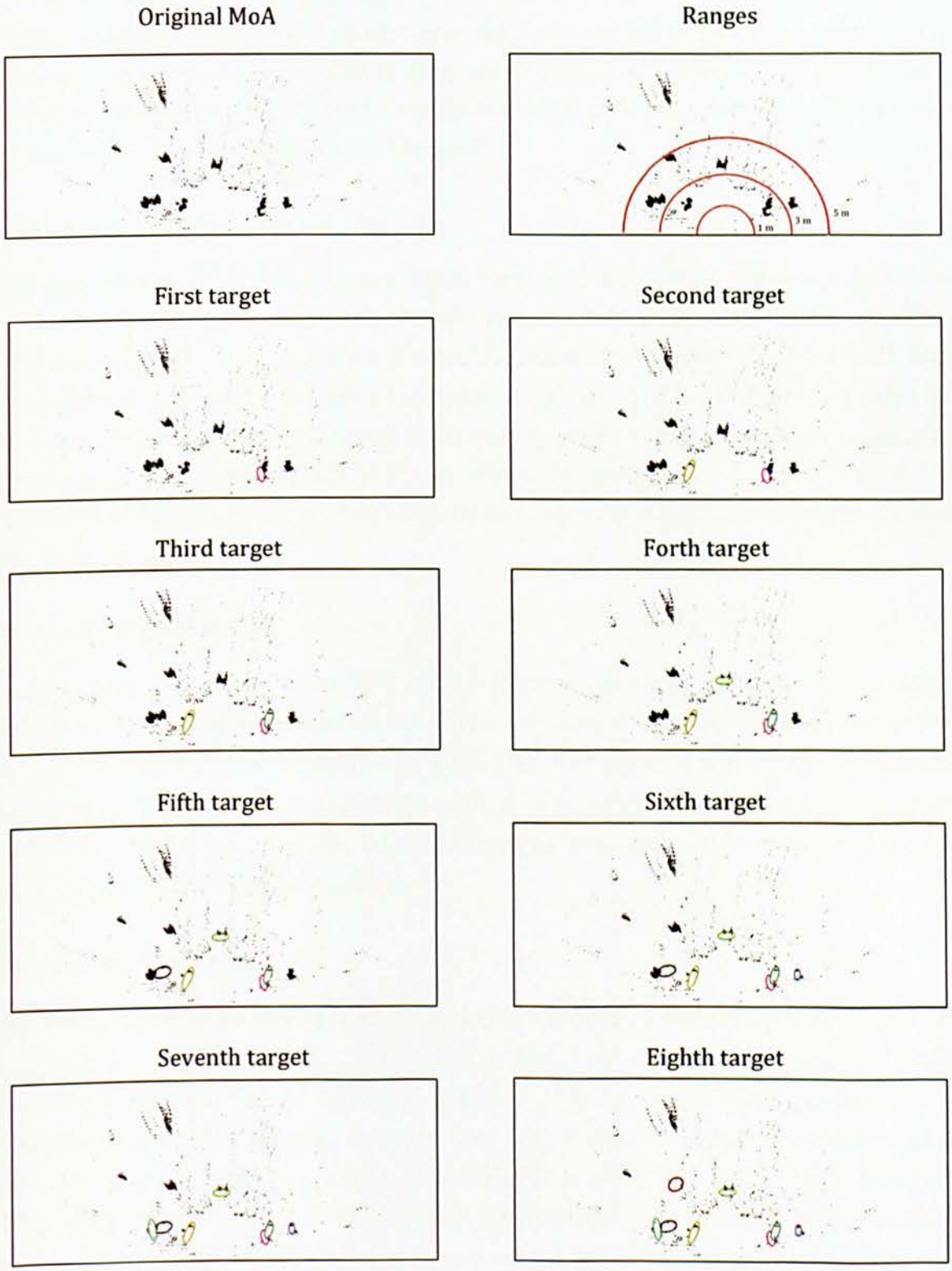


Figure 5.17: Instant of the execution of the Mean-Shift tracker with 8 people involved. The partial results are presented for each evaluation starting from the closer person.

the depth-based background subtraction implementation described in section 4.2.1.1. The objective of this measure is two fold: first, to mitigate the interference added by background data during tracking; and second, to increase the computational speed as the number of pixels to consider is smaller and less iterations are required to converge. The same approximation has been used in other works [103, 104] , however they use intensity images which require sophisticated techniques to deal with illumination conditions such as shadows or gradual illumination changes.

Automatic adaptive kernel size

The projections of targets into the MoA vary their sizes with the range as it was reported in section 4.3.3. In order to handle these variations the CAMSHIFT algorithm [110] is employed[5]. This technique follows the same methodology of Mean-Shift with an additional step at the end after the mode of the distribution is found. At this step the size and orientation of the kernel is estimated using the moments of the underlying distribution. Although CAMSHIFT was originally designed for face tracking, it has been used in other contexts as well and can serve as a good approximation for tracking people in the MoA.

Initialization of tracks

Another limitation of the Mean-Shift tracker in surveillance environments with multiple targets is the lack of a mechanism for initializing new tracks. In this implementation the people segmentation methodology presented in chapter 4 is employed to identify new people. This module is executed at every frame after all active targets have been evaluated. Therefore, only the data that has not been associated with any target is used for the detection of new people.

Termination of tracks

Regarding the identification of terminated tracks a similar procedure to the one presented in section 5.2.3.4 is employed. There is a time threshold where the target is still active even if no evidence for its existence is found. This measure allows recovery from temporarily occluded targets. A target does not produce evidence of existence when the area of the kernel returned by CAMSHIFT is smaller than a certain threshold τ_{area}. This threshold was set experimentally to 100 pixels^2 which corresponds to an area approximately of 400 cm^2 and it was estimated using the projected area of an occluded person at 1 m. from the camera.

[5]In this work is used the CAMSHIFT implementation version included in the computer vision framework OpenCV 2.4.

Summary of the proposed algorithm.

To summarize, the proposed approach consists of the following steps:

1. Sort targets by distance. A list of targets is created where targets closer to the camera are located first.
2. Track first target in the list. Using the target chromogram the tracking space is weighted (equation 5.37) and the CAMSHIFT tracker is applied.
3. All samples that belong to the current target are removed from the tracking space.
4. If there are more targets left in the list go back to step 2, otherwise continue to step 5.
5. Identify and remove terminated targets. If a target is lost for more than a maximum period of time it is eliminated from the list of targets.
6. Detect new people in the scene. The people segmentation module is executed with the remaining data after the tracking of all targets.

The results of the evaluation of this enhanced version of Mean-Shift are presented in the next chapter.

5.4 Discussion

In this chapter two different tracking methodologies have been investigated in the context of multi-target tracking. The standard approach based on object segmentations and data association, namely the Kalman filter, and the Mean-Shift method, an alternative "data association-free" tracker.

Tracking with data association: the Kalman filter

A chief aspect of the Kalman filter in multi-target environments is the selection of the correct measurement for the update of each target. This requires an additional module to evaluate and identify the best set of associations between measurements and targets at each time step. This issue is known as the data association problem. The objective is to find the set of disjoint associations that maximizes the total similarity between targets and measurements, which is computed based on the comparison of their appearance models.

Two different appearance models were presented in this chapter. First a simple spatial model defined with the mean location and area of a person in the MoA . This model is easy to implement and fast to compute but does not discriminate between people

effectively in dense target spaces. To overcome this limitation a more sophisticated model was presented, the so-called chromogram. A chromogram is a novel appearance model that combines the absolute height dimension in the 3D space and the colour dimension. It is constructed in a multi-part fashion structure particularly useful during partial occlusion situations allowing only the observable parts to be considered. Furthermore, it is robust to changes in scale since it uses the absolute height of targets.

Data association is not a trivial process especially in dense target situations. The problem becomes even harder to solve when the number of targets is unknown, spurious measurements are present in the scene, temporal occlusions are frequent and splits or *merged measurements* are considered. The problem of data association has been studied in this chapter by exploring three well known methodologies each of increasing complexity: Iterative Nearest Neighbour (INN), Sub-optimal Nearest Neighbour (SNN) and Global Nearest Neighbour (GNN). The first is widely used because of its simplicity and high speed execution, but is highly dependent on the order of association. SNN is independent of ordering but does not explicitly seek a global solution, unlike GNN.

Alternative tracker: the Mean-Shift approach

The Mean-Shift technique for tracking is considered as an alternative tracker to the traditional tracking methodologies based on data association e.g. Kalman filter. In this work a novel approximation of the Mean-Shift tracker has been presented aiming to increase the performance when tracking multiple targets.

The Mean-Shift tracker as originally proposed by Comaniciu et al. [70] weights the tracking space using the current image and the target model and searches this space for the location of maximum similarity with the target. This search is performed in an optimal way using the Mean-Shift gradient ascent methodology. Although this technique has become very popular in recent years because it is easy to implement and computationally efficient, the following limitations are associated with it:

- It is in general rather sensitive to the interferences produced by background data and similar targets, in particular during interaction periods.
- Targets are modelled with colour histograms, which are robust structures against rotations and non-rigid transformations. However, they lack of spatial information which make them less discriminative in cluttered backgrounds or when multiple targets are nearby.
- Uniquely the translational motion is computed. It does not account for changes in scale or orientation.
- The tracking space is built over the image plane where occlusions are difficult to solve.

- It does not provide a mechanism for the automatic initialization of new targets.

The approach proposed enhanced the standard Mean-Shift tracker to overcome these limitations. The modifications introduced are the followings:

- The chromogram structure presented in section 5.2.2.2 is employed to model the appearance of people. This model is highly discriminative minimizing the distractions produced from nearby targets or from the background.
- The data belonging to a target is removed from the tracking space once that target has been evaluated. This measure reduces the interference produced by nearby targets. However it requires a meaningful order of evaluation. Considering that people closer to the camera are less likely to be occluded, they are evaluated first.
- The tracking space is built over the ground plane MoA with the objective of increasing the performance during occlusion situations.
- The background data is removed from the scene at every time step before the evaluation of the targets. This measures avoid possible interferences of the background with the tracking process and speed up the computations since less data is considered.
- An automatic process to adapt the kernel size at each time step is employed to handle scale changes of targets in the tracking space. In particular it is employed CAMSHIFT, an approach proposed by Bradsky [110] that computes the scale and orientation of the target based on the moments of the distributions.
- The people segmentation module presented in chapter 4 is employed for the initialization of new targets at every time step. This module is executed after all current targets have been evaluated so only the remaining data is analysed.

The proposed approach takes "hard" decisions to determine the origin of the pixels giving priority to those targets closer to the camera, which might lead to incorrect solutions occasionally. A possible line of investigation would be to compute the probabilities of the pixels with respect to each target and assign each pixel to the target with higher probability. Another alternative could be to weight the contributions of the individual pixels with targets using their probabilities in a "soft" way.

Chapter 6

Performance Evaluation: Multi-target Tracking

The evaluation of an algorithm is not only important to compare the results with other people's works, but also to assess progress during its development. This chapter describes in detail the procedure followed for the evaluation of the tracking methodologies presented chapter 5.

The first stage is the design of a dataset and ground truth to serve as a platform for the evaluation. Ideally, the dataset should cover the challenge situations the algorithm is expected to address. For example, in this project the dataset should contain occlusions and interactions between people. The ground truth is considered the perfect solution which all algorithms should aim for. The generation of the ground truth for a tracking application normally requires the manual annotation of all targets throughout the sequence by a human operator. This is a highly tedious task which is error prone in part because of the subjective interpretation needed by the annotator and also by the likely reduction in the concentration of the operator after a long time repeating the same task.

The second stage entails the identification of the relevant failure modes of the system. Depending on the application the failure modes will be different. For instance an application that counts people is more interested in getting the right number of people in each moment rather than the accuracy in their location. Next, a set of metrics needs to be defined to cover all detected failure modes. These metrics should be comprehensive enough to allow the identification of weaknesses and strengths of the algorithms, which is useful during the development stage to assess the progress. Ideally, they should be combined to generate a single global metric to describe the overall performance which simplifies the comparison between different approaches.

This chapter first presents in section 6.1 a study is conducted to identify relevant failure modes in multi-target tracking environments and a set of metrics is proposed to provide meaningful evaluations. Section 6.2 describes the evaluation parameters. The

tracking methodologies presented in chapter 5 are evaluated and compared quantitatively in sections 6.3 and 6.4. Finally, in section 6.5 the conclusions are presented.

6.1 Failure modes and evaluation metrics

As defined in section 4.5.1, the failure modes of a system refers to the situations where the outcome of the algorithm differs from what is expected; in this case the ground truth. The failure modes are application dependent. As an example, for a system that counts people the inaccuracies in the location of people is not a relevant issue. What is important is to obtain the correct number of people in the scene. On the contrary, location inaccuracies might be relevant for an action recognition system. The proper identification of the relevant failure modes in a particular application is a critical step in order to define meaningful metrics for its evaluation.

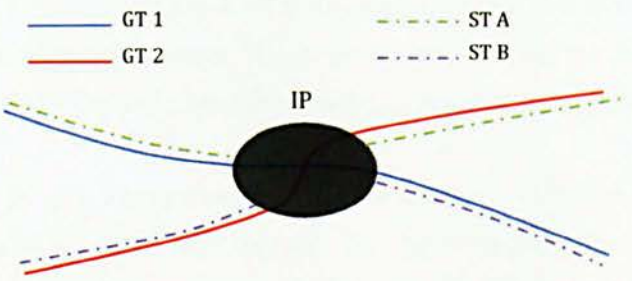
Failure modes

The two following failure modes have been identified as the most relevant failures for the proposed application:

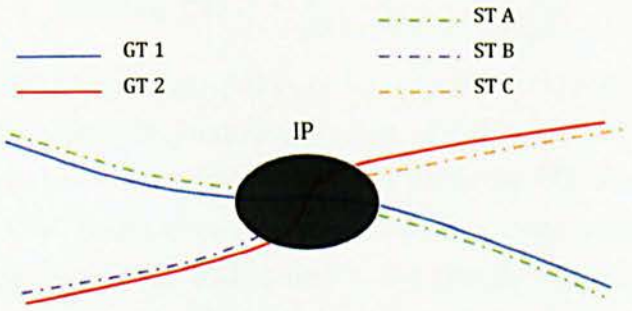
- Cardinality errors. Due to noisy and inaccurate data, discrepancies occur in the number of tracks detected by the System (STs) with respect to the number of Ground truth Tracks (GTs). These errors occur because one or several GTs were not detected by the system (e.g. the target was distant or highly occluded), or because detected STs do not actually belong to any existing GT (e.g. spurious measurements).

- Label inconsistency. When people get into physical interactions of any kind (e.g. grouping, hand shaking, path crossing) they become spatially close and the proper identification of the targets involved becomes harder to resolve. These situations are likely to produce different labels for the same target before and after the interaction. Note that the possible inconsistency of labels during the interaction (i.e. *merged measurement*) is out of the scope of this evaluation. Figure 6.1 depicts the most common scenarios where label inconsistency occurs during Interaction Periods (IP).

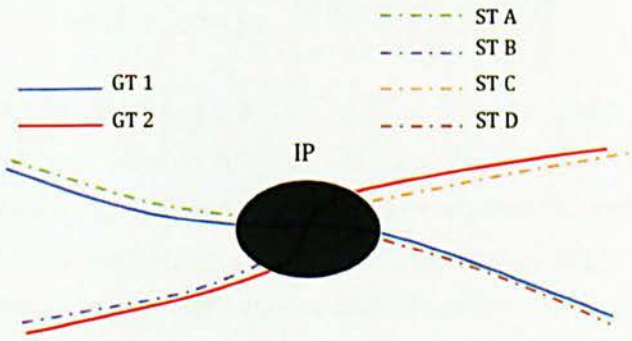
By no means do the aforementioned errors cover the totality of failures that can arise in a multi-target tracking system. However, they define the problematic situations that are intended to be tackled in this work.



(a) Failure mode 1: Two targets swap their IDs after the IP.



(b) Failure mode 2: GT 2 changes its ID after the IP. Note that this failure mode is equivalent to the failure mode where the GT 1 changes its ID instead.



(c) Failure mode 3: Both, GT 1 and GT 2 change their IDs after the IP.

Figure 6.1: Failure modes during an Interaction Period.

Evaluation metrics

Throughout the years different metrics have been proposed aiming to standardize the way multi-target trackers are evaluated. Authors tend to use the most suitable metrics for the evaluation of their algorithm and in many cases they come up with new additional metrics that better describe the particularities or novelties of their approach. Therefore, there does not exist a set of standard metrics for evaluating multi-target tracking systems. For this evaluation, the popular set of metrics proposed by Yin et al. [156] are used since they provide suitable metrics for assessing the failure modes just identified.

As a prior step to the computation of the metrics GTs are mapped with STs based on temporal and spatial overlapping. In the original paper the spatial overlap $A(GT_{i,k}, ST_{j,k})$ at frame k between the j^{th} GT and i^{th} ST is defined as:

$$A(GT_{i,k}, ST_{j,k}) = \frac{\text{Area}(GT_{i,k} \cap ST_{j,k})}{\text{Area}(GT_{i,k} \cup ST_{j,k})} \tag{6.1}$$

where $\text{Area}(GT_{i,k} \cap ST_{j,k})$ and $\text{Area}(GT_{i,k} \cup ST_{j,k})$ refer to the intersection and union region respectively between the bounding boxes of GT_i and ST_j at time k. For this work, the computation has been slightly modified since the GT and ST are represented by Gaussian PDFs. The Bhattacharyya coefficient has been used instead to obtain a value of spatial overlapping as it was done for the people segmentation evaluation in section 4.5.2.

The temporal overlapping is defined as follows:

$$T(GT_i, ST_j) = \frac{\text{Length}(GT_i \cap ST_j)}{\text{Length}(GT_i \cup ST_j)} \tag{6.2}$$

where $\text{Length}(GT_i \cap ST_j)$ and $\text{Length}(GT_i \cup ST_j)$ are the temporal intersection and union respectively between the life span of GT_i and ST_j.

In the original paper nine different metrics are presented, two for the accuracy of detections at frame level and seven for the consistency of trajectories at track level. Not all of them are relevant for this project and therefore only the following sub-set of metrics are adopted:

1. Correct Detected Tracks (CDT): A GT is identified as a CDT if it has sufficient spatial and temporal overlap with at least one ST that has sufficient spatial and temporal overlap.

$$T(GT_i, ST_j) \geq \tau_t, \quad \frac{\sum_{k=1}^{N} A(GT_{i,k}, ST_{j,k})}{N} \geq \tau_a \tag{6.3}$$

where τ_t and τ_a are predefined temporal and spatial thresholds respectively, and N refers to the number of frames that have both GT_i and ST_j.

This metric is an indicator of high performance and therefore is expected to return large values.

2. False Alarm Tracks (FAT): A ST is regarded as a FAT if it does not have enough temporal or spatial overlap with any GT.

$$T(ST_i, GT_j) < \tau_t, \quad \frac{\sum_{k=1}^{N} A(GT_{i,k}, ST_{j,k})}{N} < \tau_a \tag{6.4}$$

The evaluation of this metric should report low values to indicate good performance.

3. Track Detection Failure (TDF): A GT is considered a TDF if it does not have temporal and spatial overlap with any ST.

$$T(GT_i, ST_j) < \tau_t, \quad \frac{\sum_{k=1}^{N} A(GT_{i,k}, ST_{j,k})}{N} < \tau_a \tag{6.5}$$

As with the previous metric TDF are expected to be as low as possible to guarantee high performance.

4. ID Change (IDC): This metric counts the number of label changes for GTs. The actual implementation of this metric, unlike the one proposed by the authors, is computed with respect to GTs instead of STs. In addition, since IDCs mainly occur when targets are spatially close, they are evaluated specifically at the Interaction Periods (IPs) – see section 5.2.3.5. For evaluation purposes an IP is defined over three sub-periods (see figure 6.2):

   - Before *merged measurement*. This consists of a predefined number of frames before the *merged measurement* is detected.
   - *Merged measurement*. Lasts as long as the *merged measurement* is detected.
   - After *merged measurement*. This consists of a predefined number of frames after the actual *merged measurement* is detected.

   An IDC is counted during an IP when a GT is mapped with a particular ST before the *merged measurement* occurs and is mapped with a different ST after the *merged measurement*. Note that the evaluation during the actual *merged measurement* period is out of the scope of this project. For example, attending to the cases of study depicted in figure 6.1, in the first and third case (figures 6.1(a) and 6.1(c)) two ID changes are counted, and only one ID change is computed in the second case (figure 6.1(b)).

   Low values of IDCs indicates a good performance of the algorithm especially in the resolution of occlusions.

Figure 6.2: Sub-periods in an IP.

For more details about these metrics the reader is referred to the original paper [156].

An additional metric is presented that combines the results of CDT, FAT and TDF into a single global result. This metric is the F1-score, which was presented in section 4.5.2 and aims to ease the comparison between different algorithms by representing with one value the overall performance of the system. The F1-score values are normalized between 0 and 1 where 1 indicates the ideal performance.

6.2 Evaluation parameters

This section covers the relevant decisions taken for the actual implementation of the metrics, which allow the reader to replicate the results presented in section 6.3.

Regarding the estimation of the spatial overlap between GTs and STs defined in equation 6.1, the methodology used depends on the primitives employed to delimit the physical extent of the target. – e.g. bounding boxes, ellipses, PDFs. Unlike the original paper, in this work the physical extent of a target is represented with a Gaussian PDF, and therefore a slightly different approach is considered. The spatial overlap in this implementation is based on the Bhattacharyya coefficient (see equation 5.22), and the spatial threshold is set to 0.4.

The temporal overlap is calculated as in the original paper with a threshold set to 0.5.

In this implementation, a GT is only allowed to be mapped to a maximum of one ST. When multiple STs satisfy the spatial and temporal overlap conditions for a particular GT, the mapping is performed using a majority based rule i.e. the GT is mapped to the ST that meets the spatial condition for the largest amount of time.

Finally, for the evaluation of IDCs during interaction periods, the frame span that defines the periods before and after a *merged measurement* is set arbitrarily to 10 frames. In each period an independent GT mapping is performed following the majority based rule. An IDC is accounted if the STs mapped in both periods are different.

6.3 Evaluation of data association strategies

The tracking methodology described in section 5.2 is evaluated wit regard to the following aspects:

- Object modelling. Two different models have been proposed; the spatial model and the chromograms – see section 5.2.2.
- Data association methodology. Three techniques of increasing sophistication have been presented: Iterative Nearest Neighbour (INN), Suboptimal Nearest Neighbour (SNN) and Global Nearest Neighbour (GNN) – see section 5.2.3.
- Update strategy during *merged measurements*. When a *merged measurement* is detected the targets involved can update their location with the *merged measurement* or not. If the update is skipped, the target estimation relies exclusively on the motion model of the target previous to the *merged measurement* – see section 5.2.3.5.

Additionally, a dummy algorithm has been implemented that performs the data association randomly and is used to serve as a benchmark for the rest of approaches. This algorithm is referred to as *Version 0* and it is independent of the object model employed.

In sections 6.3.1, 6.3.2 and 6.3.3 a detailed analysis of the results is given attending to the three aforementioned aspects. These results have been obtained using the dataset presented in section 4.5.4. For completeness the full set of results are presented in section 6.3.4.

6.3.1 Choosing an object model: Spatial vs Chromogram

The first analysis conducted aims to compare the two models proposed: the simple model based only on spatial features and the more discriminative model that combines 3D height and colour dimensions, the so-called chromograms. The two evaluations are conducted using the INN data association methodology and a normal update strategy during occlusions. Table 6.1 presents the results for both appearance models along with the results of Version 0.

	CDT	TDF	FAT	F1-score	IDC
Version 0	23	22	48	0.4	69
Spatial Model	25	20	41	0.45	67
Chromogram	33	12	35	0.58	37

Table 6.1: Object model evaluation results (I).

A visual comparison is given in figure 6.3 where only the two most representative metrics are shown; the F1-score that derives from CDT, TDF and FAT, and the IDC to allow an independent evaluation of the algorithm during interaction periods.

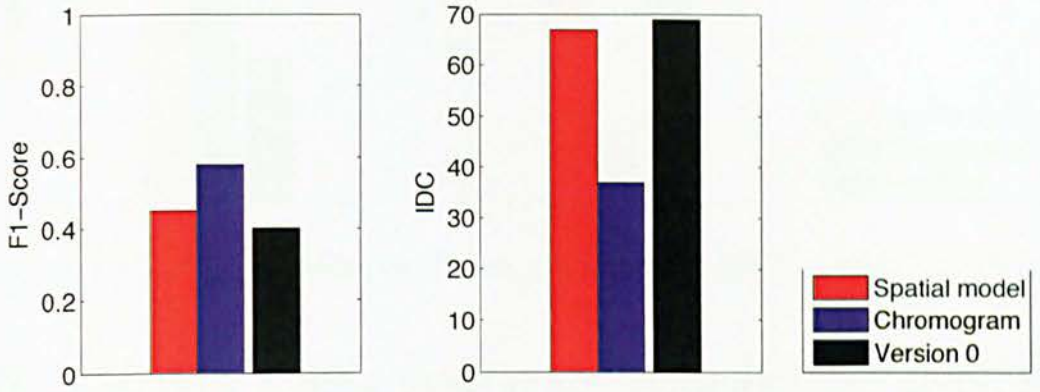


Figure 6.3: Spatial model vs Chromogram model. (*INN, normal update*).

These results indicate a clear improvement in the overall performance of chromograms over spatial models especially in terms of IDC, in fact the spatial model is only slightly better than Version 0. These results are not particularly surprising since chromograms provide with specific mechanisms for dealing with occlusions situations by using only the observable parts. Additionally, the fact that they are built over the absolute height dimension of the 3D space makes them robust to changes in scale.

Interestingly, this increment in the performance is even more significant when used along with SNN or GNN, which reveals that discriminative models are more relevant when used in combination with a sophisticated data association method. Table 6.2 presents the results obtained with the SNN methodology and a normal update strategy.

	CDT	TDF	FAT	F1-score	IDC
Version 0	23	22	48	0.4	69
Spatial Model	28	17	34	0.52	57
Chromogram	40	5	19	0.77	15

Table 6.2: Object model evaluation results (II).

Attending to the F1-score and IDC metrics figure 6.4 shows a visual comparison of both appearance models.

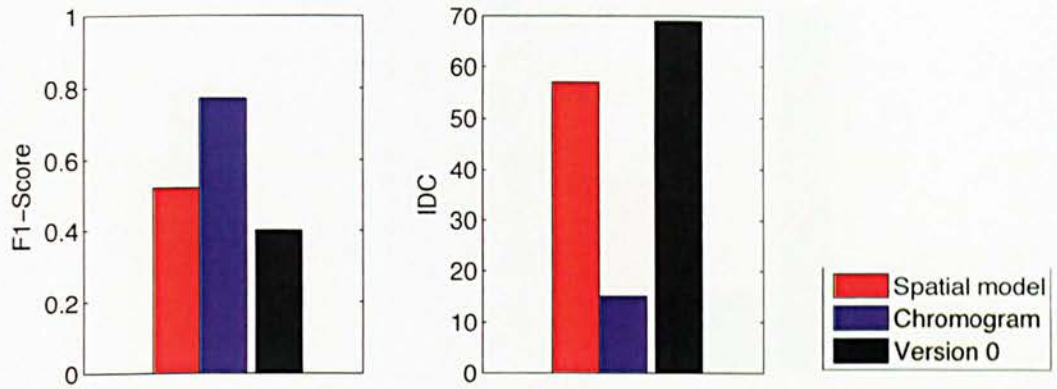


Figure 6.4: Spatial model vs Chromogram model. (*SNN, normal update*).

6.3.2 Choosing a data association methodology: INN, SNN, GNN

The evaluation of three data association approaches is presented in this section in detail. These are the simple Iterative Nearest Neighbour which relies heavily in the association order; the more advanced Suboptimal Nearest Neighbour; and the optimal Global Nearest Neighbour method. For this evaluation the chromogram appearance model is used with a normal update strategy during occlusions. Table 6.3 presents the results obtained for the three data association methods and Version 0.

	CDT	TDF	FAT	F1-score	IDC
Version 0	23	22	48	0.4	69
INN	33	12	35	0.58	37
SNN	40	5	19	0.77	15
GNN	40	5	19	0.77	15

Table 6.3: Data association evaluation.

A closer look is presented in figure 6.5 where the F1-score is used to compare the global performance of the three methods and the IDC metric is used to assess their performance specifically during interaction periods.

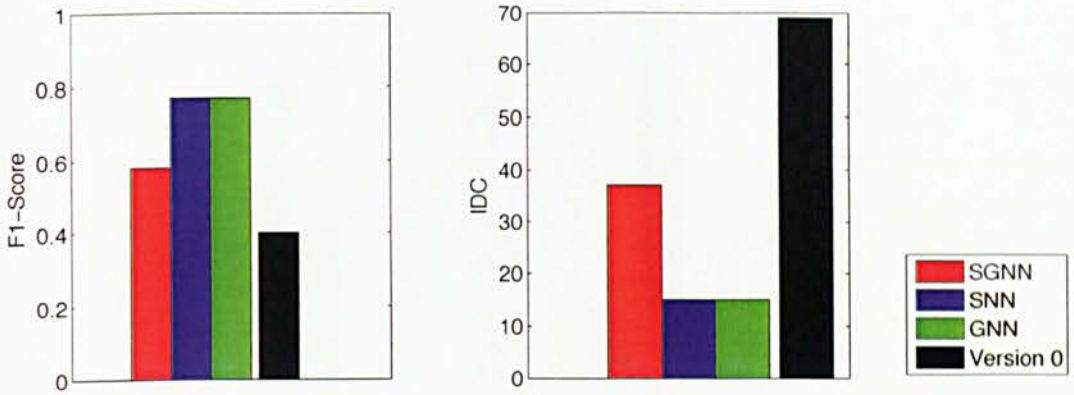


Figure 6.5: INN vs SNN vs GNN. (*Chromogram, normal update*).

The results of the F1-score metric reveal a significant increase in the overall performance of the two more sophisticated methods, SNN and GNN with respect to the simpler INN. INN still outperforms Version 0 by a factor of 1.5. Regarding the number of IDCs, INN reduces them by half with respect to Version 0. GNN and SNN obtained about 2.4 times less number of IDCs than INN. An interesting result is the fact that SNN and GNN behave similarly. A possible explanation might be that for these evaluation parameters the sub-optimal results obtained with SNN happen to be the optimal.

6.3.3 Choosing the update strategy during occlusions: Normal update vs Non-update

A final discussion refers to the update strategy followed during *merged measurements*. Two possible options are presented: normal update where targets update their position with the *merged measurement* or non-update. For this evaluation chromograms are used as appearance models and the GNN methodology is employed to resolve the data association problem. Table 6.4 presents the results obtained for the two update strategies and Version 0.

	CDT	TDF	FAT	F1-score	IDC
Version 0	23	22	48	0.4	69
Normal update	40	5	19	0.77	15
Non-update	43	2	18	0.81	12

Table 6.4: Update strategy evaluation (I).

Figure 6.6 presents the visual comparison of the overall performance with the F1-score and the more detailed evaluation during interaction periods with the IDC metric.

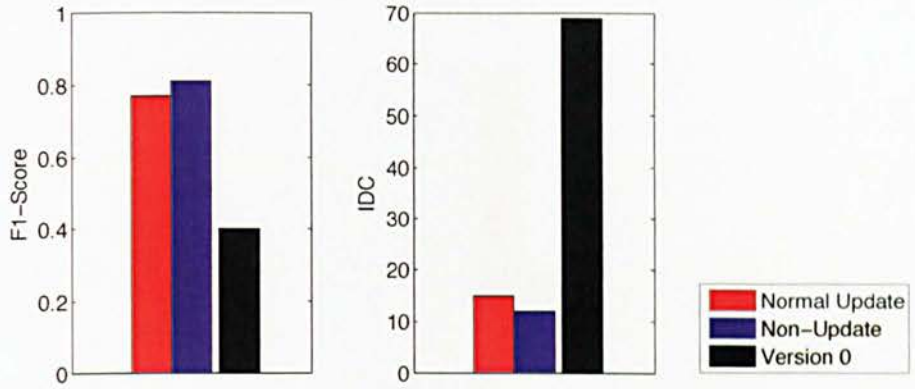


Figure 6.6: Update vs Non-Update strategy during mergings. (*Chromogram, GNN*)

The F1-score results show a similar performance between the two update strategies improving Version 0 by a factor of 2 approximately. Both strategies obtain comparable numbers of IDCs which is not surprising since the similarity between chromograms does not employ location features. Furthermore the numbers of IDCs are reduced about seven times with respect to Version 0.

An additional evaluation was conducted to assess the performance of the spatial model with respect to the update strategy utilized. Table 6.5 presents the comparison between the two update strategies using the spatial model and the GNN data association method.

	CDT	TDF	FAT	F1-score	IDC
Version 0	23	22	48	0.4	69
Normal update	27	18	35	0.5	60
Non-update	41	4	21	0.77	20

Table 6.5: Update strategy evaluation (II).

It is interesting the significant improvement when the non-update strategy is combined with spatial models as illustrated in figure 6.7. The overall evaluation is presented with the F1-score and IDCs show the specific performance of the algorithm during occlusions.

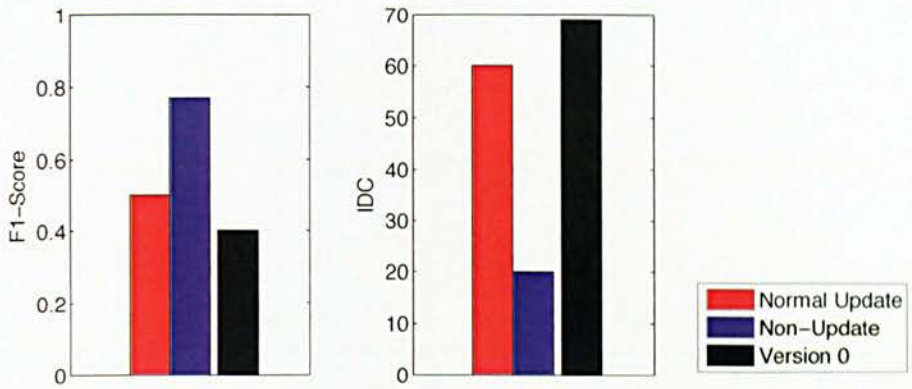


Figure 6.7: Update vs Non-Update strategy during mergings. (*Spatial model, GNN*)

These results suggest that spatial models, which in general are considered poor discriminative models when targets are close, are in fact adequate models when combined with a non-update strategy during *merged measurements*. However, this interpretation must be taken with caution as it depends on the accuracy of the targets' motion model. If during a *merged measurement* the targets involved modify their velocity or direction, the motion model prior to the *merged measurement* is not longer accurate and relying on it results inevitably in failures. As an example, figure 6.8 presents a situation where two targets become merged and the motion model of one of them changes. The two update strategies are compared during this scenario using different appearance models. The first approach utilizes the spatial model, INN for data association and a non-update strategy during the occlusion. The second approach uses the chromogram appearance model, GNN and a normal update strategy. As expected the first approach fails to resolve the occlusion since it relies on the motion model of the target. The second approach, on the other hand, succeeds because it does not use any motion estimation and relies exclusively on the performance of chromograms, highly discriminative appearance models.

6.3.4 Complete set of evaluation results

In this section the results obtained from all the evaluations conducted are presented. For visual purposes the results are divided in three tables. Table 6.6 include the benchmark results obtained with the Version 0 algorithm. Table 6.7 presents the results obtained with a normal update strategy during occlusions. Finally table6.8 shows the evaluations with a non-update strategy during occlusions.

	CDT	TDF	FAT	F1-Score	IDC
Version 0	23	22	48	0.40	69

Table 6.6: Results obtained from a random process of associations (Version 0).

Figure 6.8: Three key frames of an interaction period between two targets. Top row: Non-update strategy combined with spatial models and INN. This approach fails as the motion of one of the targets slightly changes during the interaction. Middle row: Colour images of the key frames of the interaction. Bottom row: Update strategy combined with chromograms and GNN. In this case the association is resolved successfully.

			CDT	TDF	FAT	F1-score	IDC
Normal Update	Spatial Model	INN	25	20	41	0.45	67
		SNN	28	17	34	0.52	57
		GNN	27	18	35	0.5	60
	Chromogram	INN	33	12	35	0.58	37
		SNN	40	5	19	0.77	15
		GNN	40	5	19	0.77	15

Table 6.7: Set of results for a normal update strategy during occlusions.

			CDT	TDF	FAT	F1-score	IDC
Non-Update	Spatial Model	INN	37	8	29	0.67	28
		SNN	41	4	21	0.77	20
		GNN	41	4	21	0.77	20
	Chromogram	INN	35	10	35	0.61	41
		SNN	40	5	24	0.73	18
		GNN	43	2	18	0.81	12

Table 6.8: Set of results for a non-update strategy during occlusions.

6.4 Evaluation of the enhanced Mean-Shift methodology

In this section the Mean-Shift approach for multi-target tracking proposed in section 5.3.2 is evaluated quantitatively. In addition, it is compared with the tracking methodology based on Kalman filter described in section 5.2; in particular with the version that obtained the best results in the evaluation of section 6.3 i.e. KF with the Global Nearest Neighbour for data association and the chromogram appearance model. The results are summarized in table 6.9.

	CDT	TDF	FAT	F1-Score	IDC
Enhanced Mean-Shift	36	9	105	0.39	35
KF + GNN + Chromogram	43	2	18	0.81	12

Table 6.9: Performance evaluation of the enhanced version of the Mean-Shift approach and the tracking methodology based on Kalman filter, GNN and chromograms

A closer detail of the most significant results is presented in figure 6.9.

With regard to the number of CDT, the results of Mean-Shift are comparable with those obtained with the traditional tracker based on KF and data association. In terms

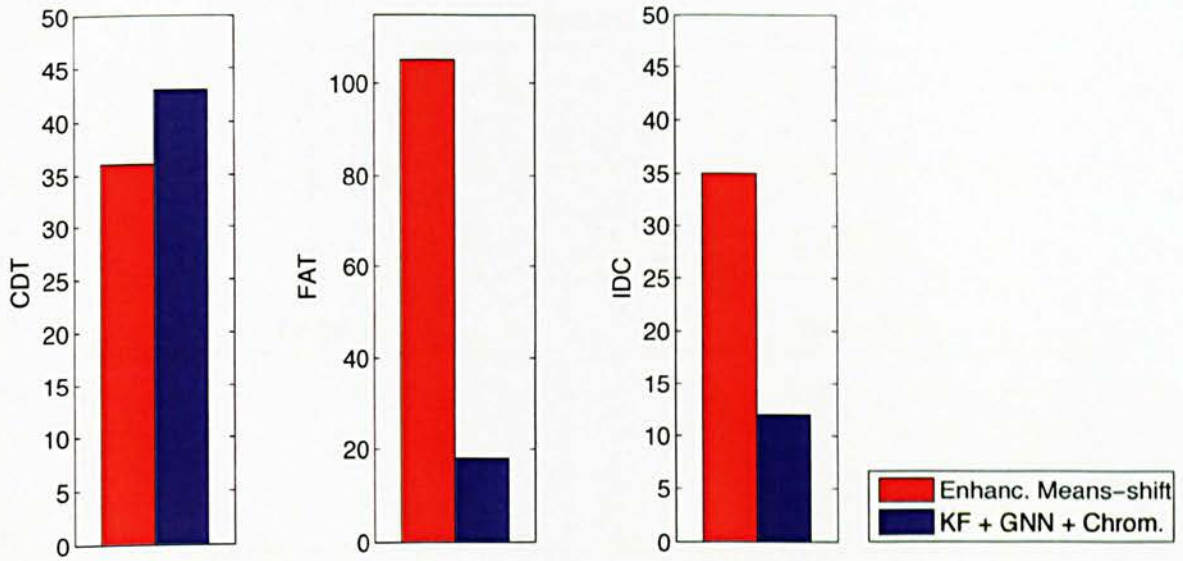


Figure 6.9: Performance evaluation in terms of CDT, FAT and IDC – Enhanced Mean-Shift algorithm and KF-based tracking with GNN and chromograms.

of IDCs, Mean-Shift obtains more than double the number produced by the traditional approach. The more significant result is obtained with the number of FATs. The Mean-Shift approach is especially poor in the large amount of false alarms that are produced. A possible explanation for this behaviour may be the inaccuracy of the people dimension estimated by CAMSHIFT. It is been observed that the estimations are slightly smaller than the actual extension of people in MoA. This results in regions of people data falling outside the estimation area which are not removed from the feature space and, as a consequence, being detected as new people. Figure 6.10 shows an example of this behaviour in a particular instant of the video sequence.

A possible solution to this situation is to explore different approaches for computing the size of the estimation such as the SOAMST algorithm [109], or the method proposed by Zivkovic and Krose [71] based on the EM algorithm. Alternatively, a restriction could be added to the location where new people are detected which prevents the creation of new targets at the edges of the MoA. However, more research on this issue needs to be undertaken.

6.5 Discussion

In this section the different tracking methodologies presented in chapter 5 were evaluated quantitatively. For this evaluation a dataset was specifically created which consists of two video sequences: one for training (i.e. parameters setting) and another for the actual evaluation. They were recorded from a set of three Kinect sensors strategically mounted on a non-overlapping configuration at a high location. The content of the videos consists of people walking in a lab with constant interactions e.g. path crossing.

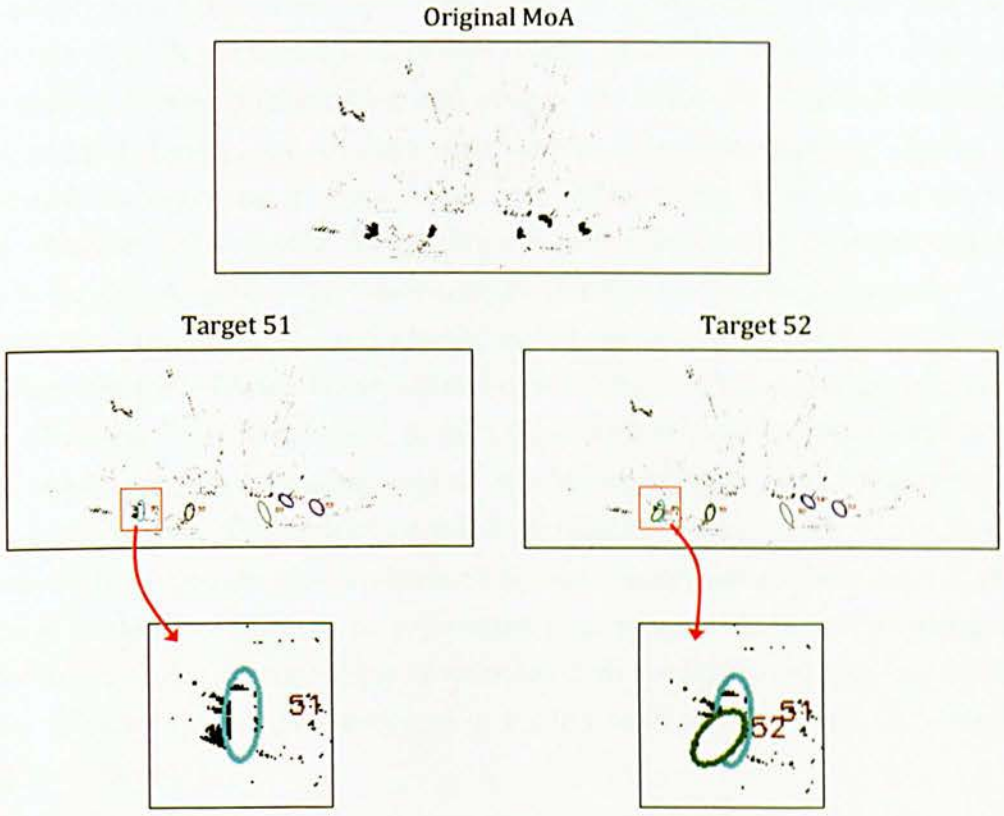


Figure 6.10: CAMSHIFT estimates a smaller region for target 51 leading the remaining points to be detected as a new target.

The dataset was manually annotated with bounding boxes by a human operator and the assistance of the semi-automatic tool VATIC. Using these annotations the final ground truth was generated by modelling the data within the bounding boxes with Gaussian PDFs.

A proper evaluation requires the identification of the relevant failure modes for each application. In the proposed system two main failure modes were recognized: cardinality errors that occur when the number of ground truth tracks differs from the number of tracks detected by the system; and the inconsistency of labels that happens during interaction periods. For the assessment of these failure modes some of the metrics proposed by Yin et al. [156] were employed.

First it was evaluated the performance of the tracking methodology based on Kalman filter and data association. Three relevant aspects were specifically considered: the object model; the data association methodology; and the update strategy during *merged measurements*. As expected, discriminative models such as chromograms perform better than simple models based solely on spatial features. In particular the difference is more significant when used in conjunction with an optimal data association technique. The multi-part structure of chromograms combined with the absolute height dimension in the 3D space have been proven successful for resolving occlusions. The performance of the simple data association method INN is not surprisingly outperformed by the

more sophisticated approaches SNN and GNN. It is interesting to note the fact that the outcome of SNN is comparable to the results obtained with GNN. This suggests that for this particular configuration and design, the sub-optimal association coincides with an optimal. Lastly, the results revealed that the update strategy during *merged measurements* is in general irrelevant when the object model does not contain location features. However, when spatial models are used the performance increases significantly as long as targets do not modify their motion during *merged measurements*.

Second, the enhanced Mean-Shift algorithm for tracking was evaluated and the results were compared with those obtained with the traditional approach based on Kalman filter and data association; in particular with the version that combines GNN for data association and chromograms for modelling people since it was proven to be the best combination. The results revealed an inferior performance of the Mean-Shift approach with respect to the traditional tracker especially in the number of FATs. The reason could be attributed to a systematic underestimation in the size of targets. It is concluded that further enhancements need to be introduced in the Mean-Shift approach in order to achieve comparable results to those obtained with traditional tracking methodologies.

Chapter 7

Conclusions and Future Work

7.1 Outcome

This thesis aimed to investigate the use of the popular Kinect RGB-D sensor for surveillance purposes. A framework was proposed for the integration of multiple non-overlapping RGB-D cameras to allow the monitoring of large area indoor spaces. New techniques were developed that fully exploit the capabilities of RGB-D sensors. This study advanced towards new workspaces that are expected to serve as the basis for further study within the research community.

7.2 Contributions

In this section the main contributions of this work are summarized.

7.2.1 Calibration of non-overlapping RGB-D cameras

The surveillance framework proposed in this work is formed by three non-overlapping Microsoft *Kinect*© sensors. This configuration maximizes the area covered and minimizes the interference between sensors. To efficiently use the data from all sensors it is required to calibrate the sensors with respect to a common coordinate system.

7.2.1.1 Issues

The external calibration of non-overlapping cameras is always a challenging task, especially because standard procedures based on corresponding points cannot be applied. Additionally, issues related to the depth resolution and noise of the depth sensor inevitably result in inaccurate calibration parameters. Finally, a reference coordinate system for the entire device should be chosen carefully to allow simple calibration procedures and serve as a useful representation for segmentation and tracking tasks.

7.2.1.2 Solutions

A novel plane-based procedure is proposed for the calibration of non-overlapping RGB-D sensors. The method uses corresponding planes to derive constraints on rotation and translation. The normal vector of the planes are used to estimate the rotations, while the translation is computed by using the closest point of the planes to the origin of the reference CS. In order to obtain many corresponding planes between pairs of adjacent non-overlapping cameras a calibration tool was presented – the "paddle". This tool features two coplanar boards attached to both ends of a pole to be detected by adjacent cameras simultaneously. Using a plane fitting approach planes were effectively extracted from the range data. Finally, for practical reasons the middle Kinect CS was selected as the reference CS which minimizes the required number of calibrations.

7.2.1.3 Outstanding problems

The proposed solution for the calibration of non-overlapping range cameras requires some manual intervention for holding the paddle in different positions in front of the cameras. Ideally, the procedure would be fully automatic. A possible line of investigation is the use of accumulation of trajectories to estimate automatically the geometric calibration between sensors.

7.2.2 Depth-based polar coordinate system for people segmentation

For segmenting people in the proposed framework it requires the use of a common representation that aggregates the data from all sensors. Different depth-based spaces have been explored in order to obtain a representation that achieves high performances in the context of people segmentation, especially during occlusion situations.

7.2.2.1 Issues

Due to the nature of RGB-D sensors based on triangulation the depth resolution decreases with distance while the amount of noise increases. These issues result in people data appearing increasingly scattered with distance along the optical axis of the camera. Furthermore, when considering the aggregated view from all sensors, each camera produces a different orientation of data.

7.2.2.2 Solutions

A depth-based polar coordinate system is proposed to effectively aggregate the data from all sensors – Remapped Polar Space (RPS). In this space the problem of different orientations is automatically solved by transforming the data into a polar representation.

In addition, the effect of increasing scattering of data with distance is mitigated by the use of a remapping operation. This operation compresses distant data and enlarges closer data resulting in a homogeneous representation of data throughout the range. The proposed space allows segmentations of people at distances beyond the operating range of the sensor.

7.2.2.3 Outstanding problems

The RPS involves an increase of approximately 26% on the computation time. The mapping of all data from the 3D Cartesian CS to the RPS comprises a set of non-linear transformations that must be performed point-wise. A future version could consider a parallelized implementation in order to speed up the process.

For tracking purposes the RPS presents some limitations since the motion of people cannot be assumed linear. This entails the use of more complex tracking algorithms such as particle filters, which in general are computationally intensive.

7.2.3 Chromogram appearance models

Traditional tracking methodologies rely on the correct identification of observations over time (i.e. data association). In this context it is required the use of appearance models that can effectively distinguish people from each other.

7.2.3.1 Issues

The issues related to appearance models are associated with variations in the target representation over time. Several problematic situations that produce changes on the targets' appearance are identified:

- Occlusions: The correct identification of people during and after occlusions is a real challenge since their appearance inevitably change. Occlusions are very frequent in situations of high density of people.
- Illumination changes: Appearance models based on colour information are highly sensitive to illumination conditions, e.g. weather, switching on/off lights, etc. Additionally, in multi-camera systems the specific configuration of each camera e.g. camera shutter, results in different colour representations.
- Scale changes: When appearance models are built over the image plane people scale varies according to the distance to the camera.

7.2.3.2 Solutions

A novel discriminative multi-part appearance model was presented that combines the height from the 3D space and colour information. It was specifically designed to be effective in the presence of occlusions, the multi-part structure allows only the observable parts to be considered. It is also robust to changes in scale since it is built from the absolute height of targets.

7.2.3.3 Outstanding problems

Chromograms have proven to serve well in dense target situations. However they generally fail to distinguish people dressed in similar colours. It would be interesting to assess the effect of using extra information such as texture.

7.2.4 RGB-D dataset for people segmentation and multi-target tracking

For the evaluation and comparison of the different algorithms presented in this work it is required the use of a benchmark dataset.

7.2.4.1 Issues

The design of a proper evaluation platform for multi-target tracking algorithms is a highly complex task. First, it requires the design of a suitable dataset e.g. definition of routes and behaviour of actors, type of interactions, etc. Second, to produce the ground truth annotations, which is in general subject to different interpretations and requires the definition of certain rules e.g. how to annotate occluded people, what label assigned to people re-entering the scene, etc. Third, the identification of the relevant failure modes of the application. Finally, the definition of a set of metrics that provides meaningful evaluation.

7.2.4.2 Solutions

A new dataset was presented for the evaluation of people segmentation and tracking algorithms. This dataset was recorded with the combined device proposed in this work covering an area of approximately 220 m^2. Up to 15 people appear in the evaluation sequence performing normal behaviours such as walking through the scene in a casual way and showing frequent short-lived interactions between them. It comprises approximately 140 different people interactions where occlusions, dynamic and static, are highly frequent. The dataset was manually annotated with bounding boxes by a human operator and the assist of the semi-automatic tool VATIC. Two relevant failure modes were identified in the context of multi-target tracking: *cardinality errors*

that occur when the number of ground truth tracks differs from the number of tracks detected by the system; and the *inconsistency of labels* frequently during occlusions. For the assessment of these failure modes some of the metrics proposed by Yin et al. [156] were employed along with the F1-score metric.

7.2.4.3 Outstanding problems

The majority of occlusions appearing on the dataset are reduced to brief interactions such as handshaking or path crossing. A future extension should include new scenarios with more challenging occlusions such as grouping of people, changes of directions and velocity, etc.

7.2.5 Additional contributions

In this section some additional contributions that were proposed to assist in the progress of this work are presented.

7.2.5.1 Enhanced Mean-Shift algorithm for tracking

A modified version of the Mean-Shift tracker was proposed aiming to improve the performance in multi-target tracking environments. The main modifications are, first, the integration with chromogram appearance models to increase the discriminative capacity during occlusions. Second, the use of a ground plane tracking space to minimize occlusions. Third, the segmentation of foreground data to reduce distractions from the background. Finally, the use of a priority-based target evaluation strategy to minimize the interferences between targets.

In the proposed version, targets closer to the camera have priority for using pixels located in the intersection area with other targets which might lead to incorrect solutions occasionally. A possible line of investigation would be to use the probabilities for the pixels with respect to each target. For instance a pixel is used with the target with highest probability. Alternatively, it could be used in a soft way by weighting the contribution of the individual pixels with targets using their probabilities. To reduce the inevitable increase in the computation time, parallelized implementations could be considered.

7.2.5.2 Depth-based foreground detection

A depth-based background subtraction approach is proposed for foreground segmentation that mitigates the low resolution and increasing noise introduced by RGB-D sensors at far distances. The main contribution in this context is the use of an adaptive

depth threshold derived from a characterization of the error. This approach effectively segments foreground data while minimizing the noise.

A failure mode has been identified in the proposed method. When a person is near the background, that person will not be detected if the depth difference between the person and the background is smaller than the threshold used at that distance. A natural progression of this work would be the use of additional information such as colour to assist in the segmentation at critical regions of the scene.

7.2.5.3 *Merged measurement* detector

A detector of *merged measurements* is proposed in this work, which is a module responsible for the recognition of measurements produced by more than one target. These measurements appear when people are in close proximity and the sensor, due to its limited resolution, cannot separate their signals yielding a single measurement that combines them all. The correct detection of these measurements is of critical importance since the results obtained from this module are used by the tracker to apply different update strategies, depending on whether the measurement is merged or not.

The proposed approach labels a measurement as a merge if it satisfies the two following requirements: its area is larger than a defined threshold and more than one tracked target are in close proximity.

It has been identified a significant failure mode that produces a certain number of false negatives. The reason seems to be the misdetection of partially occluded people during interaction periods. As a consequence the target proximity requirement is not satisfied. Future research might explore a soft approach where instead of making a hard decision whether a measurement is merged or not, it could return probabilities to be used for weighting the subsequent actions accordingly. Another possibility could be to take special actions when these situations are likely to occur; for instance by lowering the detection threshold in that region to reduce the probability of misdetections.

References

[1] Milestone xprotect 2014. [Online]. Available: http://www.milestonesys.com/ 1

[2] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," Proceedings of the IEEE, vol. 89, no. 10, pp. 1456–1477, 2001. 8

[3] M. Valera and S. Velastin, "Intelligent distributed surveillance systems: a review," in IEE Proceedings on Vision, Image and Signal Processing, vol. 152, no. 2. IET, 2005, pp. 192–204. 8

[4] C. Stahlschmidt, A. Gavriilidis, J. Velten, and A. Kummert, "People detection and tracking from a top-view position using a time-of-flight camera," in Multimedia Communications, Services and Security. Springer, 2013, pp. 213–223. 8, 42, 44, 99

[5] R. Muñoz-Salinas, E. Aguirre, and M. García-Silvente, "People detection and tracking using stereo vision and color," Image and Vision Computing, vol. 25, no. 6, pp. 995–1007, 2007. 8, 15, 42, 80

[6] T. Horprasert, D. Harwood, and L. S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in Proceedings of 7th IEEE International Conference on Computer Vision (ICCV'99), vol. 99, 1999, pp. 1–19. 10, 11

[7] M. Karaman, L. Goldmann, D. Yu, and T. Sikora, "Comparison of static background segmentation methods," in Visual Communications and Image Processing, 2005. International Society for Optics and Photonics, 2005, pp. 596069–596069.

[8] R. Rodriguez-Gomez, E. J. Fernandez-Sanchez, J. Diaz, and E. Ros, "Fpga implementation for real-time background subtraction based on horprasert model," Sensors, vol. 12, no. 1, pp. 585–611, 2012. 10, 11

[9] B. Lo and S. Velastin, "Automatic congestion detection system for underground platforms," in Proceedings of International Symposium on Intelligent Multimedia, Video and Speech Processing. IEEE, 2001, pp. 158–161. 10, 11, 44

[10] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 780–785, 1997. 10, 11, 14

[11] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," in Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc., 1997, pp. 175–181. 10

[12] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), vol. 2. IEEE, 1999. 10, 11, 42, 44

[13] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in European Conference on Computer Vision (ECCV'00). Springer, 2000, pp. 751–767. 10, 11, 44

[14] N. Oliver, B. Rosario, and A. Pentland, "A bayesian computer vision system for modeling human interactions," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 831–843, 2000. 10

[15] L. Li, W. Huang, I. Y. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in Proceedings of 11th ACM International Conference on Multimedia. ACM, 2003, pp. 2–10. 10

[16] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," Real-time Imaging, vol. 11, no. 3, pp. 172–185, 2005. 10

[17] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in Procedings of 2nd European Workshop on Advanced Video Based Surveillance Systems, vol. 25, 2001, pp. 1–5. 11, 42

[18] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 10, pp. 1337–1342, 2003. 11, 42

[19] P. D. Z. Varcheie, M. Sills-Lavoie, and G.-A. Bilodeau, "A multiscale region-based motion detection and background subtraction algorithm," Sensors, vol. 10, no. 2, pp. 1041–1061, 2010. 11

[20] A. Hernández-Vela, M. Reyes, V. Ponce, and S. Escalera, "Grabcut-based human segmentation in video sequences," Sensors, vol. 12, no. 11, pp. 15 376–15 393, 2012.

[21] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04), vol. 2. IEEE, 2004, pp. II–302.

[22] I. Bravo, M. Mazo, J. L. Lázaro, A. Gardel, P. Jiménez, and D. Pizarro, "An intelligent architecture based on field programmable gate arrays designed to detect moving objects by using principal component analysis," Sensors, vol. 10, no. 10, pp. 9232–9251, 2010. 11

[23] B. Zhong, X. Hong, H. Yao, S. Shan, X. Chen, and W. Gao, "Texture and motion pattern fusion for background subtraction," in Proceedings of 11th Joint Conference on Information Sciences, 2008, pp. 15–20. 11

[24] B. Zhang, B. Zhong, and Y. Cao, "Complex background modeling based on texture pattern flow with adaptive threshold propagation," Journal of Visual Communication and Image Representation, vol. 22, no. 6, pp. 516–521, 2011.

[25] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," IEEE transactions on pattern analysis and machine intelligence, vol. 28, no. 4, pp. 657–662, 2006.

[26] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld, "Detection and location of people in video images using adaptive fusion of color and edge information," in Proceedings 15th International Conference on Pattern Recognition (ICPR'00), vol. 4. IEEE, 2000, pp. 627–630. 11

[27] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in Proceedings of 7th IEEE International Conference on Computer Vision (ICCV'99), vol. 1. Ieee, 1999, pp. 255–261. 11

[28] X. Fang, W. Xiong, B. Hu, and L. Wang, "A moving object detection algorithm based on color information," in Journal of Physics: Conference Series, vol. 48. IOP Publishing, 2006, p. 384.

[29] D. Pokrajac and L. Latecki, "Spatiotemporal blocks-based moving objects identification and tracking," IEEE Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS'03), pp. 70–77, 2003. 11

[30] Y. Ivanov, A. Bobick, and J. Liu, "Fast lighting independent background subtraction," International Journal of Computer Vision, vol. 37, no. 2, pp. 199–207, 2000. 11

[31] G. Gordon, T. Darrell, M. Harville, and J. Woodfill, "Background estimation and removal based on range and color," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), vol. 2. IEEE, 1999.

[32] R. Crabb, C. Tracey, A. Puranik, and J. Davis, "Real-time foreground segmentation via range and color imaging," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'08). IEEE, 2008, pp. 1–5.

[33] J. Zhu, M. Liao, R. Yang, and Z. Pan, "Joint depth and alpha matte optimization via fusion of stereo and time-of-flight sensor," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09). IEEE, 2009, pp. 453–460.

[34] I. Schiller and R. Koch, "Improved video segmentation by adaptive combination of depth keying and mixture-of-gaussians," in Image Analysis. Springer, 2011, pp. 59–68. 11

[35] L. Wang, K. L. Chan, and G. Wang, "Human detection with occlusion handling by over-segmentation and clustering on foreground regions," in 12th Asian Conference on Computer Vision Workshops (ACCV'12). Springer, 2013, pp. 197–208. 11

[36] J. Han, E. J. Pauwels, P. M. de Zeeuw, and P. H. de With, "Employing a rgb-d sensor for real-time tracking of humans across multiple re-entries in a smart environment," IEEE Transactions on Consumer Electronics, vol. 58, no. 2, pp. 255–263, 2012. 14, 42, 44

[37] D. W. Hansen, M. S. Hansen, M. Kirschmeyer, R. Larsen, and D. Silvestre, "Cluster tracking with time-of-flight cameras," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'08). IEEE, 2008, pp. 1–6. 11

[38] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12). IEEE, 2012, pp. 1815–1821. 11, 21

[39] N. D. Bird, O. Masoud, N. P. Papanikolopoulos, and A. Isaacs, "Detection of loitering individuals in public transportation areas," IEEE Transactions on Intelligent Transportation Systems, vol. 6, no. 2, pp. 167–177, 2005. 11

[40] J. Martínez-del Rincón, E. Herrero-Jaraba, J. R. Gómez, C. Orrite-Uruñuela, C. Medrano, and M. A. Montañés-Laborda, "Multicamera sport player tracking with bayesian estimation of measurements," Optical Engineering, vol. 48, no. 4, pp. 047201–047201, 2009. 11

[41] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1409–1422, 2012. 12

[42] R. E. Kalman, "A new approach to linear filtering and prediction problems," Journal of Basic Engineering, vol. 82, no. 1, pp. 35–45, 1960. 12, 82

[43] J. Lou, Q. Liu, T. Tan, and W. HU, "3-d model based visual traffic surveillance," Acta Automatic Sinica, vol. 29, no. 3, pp. 434–449, 2003. 13

[44] C. Rossi, M. Abderrahim, and J. C. Díaz, "Tracking moving optima using kalman-based predictions," Evolutionary Computation, vol. 16, no. 1, pp. 1–30, 2008. 13

[45] M. Ribeiro, "Kalman and extended kalman filters: Concept, derivation and properties," Institute for Systems and Robotics, p. 43, 2004. 13

[46] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, Chapel Hill, NC, vol. 7, no. 1, 1995. 13

[47] S. J. Julier and J. K. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in Int. symp. aerospace/defense sensing, simul. and controls, vol. 3, no. 26. Orlando, FL, 1997, pp. 3–2. 13

[48] G. Terejanu, "Unscented kalman filter tutorial," in Workshop on Large-Scale Quantification of Uncertainty, Sandia National Laboratories, 2009, pp. 1 6. 13

[49] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in IEE Proceedings on Radar and Signal Processing, vol. 140, no. 2. IET, 1993, pp. 107–113. 13

[50] M. Isard and A. Blake, "Condensationconditional density propagation for visual tracking," International Journal of Computer Vision, vol. 29, no. 1, pp. 5 28, 1998. 13

[51] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," Statistics and Computing, vol. 10, no. 3, pp. 197–208, 2000. 13

[52] K. Kim and L. Davis, "Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering," European Conference on Computer Vision (ECCV'06), pp. 98–109, 2006. 13

[53] H. Charif and S. McKenna, "Tracking the activity of participants in a meeting," Machine Vision and Applications, vol. 17, no. 2, pp. 83–93, 2006.

[54] J. J. Pantrigo, J. Hernández, and A. Sánchez, "Multiple and variable target visual tracking for video-surveillance applications," Pattern Recognition Letters, vol. 31, no. 12, pp. 1577–1590, 2010.

[55] J. Wang, Y. Ma, C. Li, H. Wang, and J. Liu, "An efficient multi-object tracking method using multiple particle filters," in World Congress on Computer Science and Information Engineering, vol. 6. IEEE, 2009, pp. 568–572.

[56] A. S. Montemayor, J. J. Pantrigo, and J. Hernández, "A memory-based particle filter for visual tracking through occlusions," in Bioinspired Applications in Artificial and Natural Computation. Springer, 2009, pp. 274–283. 80

[57] L. Jing and P. Vadakkepat, "Interacting mcmc particle filter for tracking maneuvering target," Digital Signal Processing, vol. 20, no. 2, pp. 561–574, 2010.

[58] R. Cabido, A. S. Montemayor, J. J. Pantrigo, and B. R. Payne, "Multiscale and local search methods for real time region tracking with particle filters: local search driven by adaptive scale estimation on gpus," Machine Vision and Applications, vol. 21, no. 1, pp. 43–58, 2009. 13

[59] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 174–188, 2002. 13

[60] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1. IEEE, 2005, pp. 886–893. 14, 15

[61] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004. 14

[62] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in European Conference on Computer Vision (ECCV'06). Springer, 2006, pp. 404–417. 14

[63] Z. Kim, "Real time object tracking based on dynamic feature grouping with background subtraction," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08). IEEE, 2008, pp. 1–8. 14

[64] J. Wen, X. Li, X. Gao, and D. Tao, "Incremental learning of weighted tensor subspace for visual tracking," in IEEE International Conference on Systems, Man and Cybernetics (SMC 2009). IEEE, 2009, pp. 3688–3693. 14

[65] W. Hu, X. Li, X. Zhang, X. Shi, S. Maybank, and Z. Zhang, "Incremental tensor subspace learning and its applications to foreground segmentation and tracking," International Journal of Computer Vision, vol. 91, no. 3, pp. 303–327, 2011.

[66] D. Greenhill, J. Renno, J. Orwell, and G. A. Jones, "Occlusion analysis: Learning and utilising depth maps in object tracking," Image and Vision Computing, vol. 26, no. 3, pp. 430–441, 2008. 14

[67] H. Wang, D. Suter, K. Schindler, and C. Shen, "Adaptive object tracking based on an effective appearance filter," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 9, pp. 1661–1667, 2007. 14

[68] M. S. Allili and D. Ziou, "Object of interest segmentation and tracking by using feature selection and active contours," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07). IEEE, 2007, pp. 1–8. 14

[69] H. T. Nguyen, M. Worring, and R. Van Den Boomgaard, "Occlusion robust adaptive template tracking," in Proceedings of 8th IEEE International Conference on Computer Vision (ICCV'01), vol. 1. IEEE, 2001, pp. 678–683. 14, 18

[70] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00), vol. 2. IEEE, 2000, pp. 142–149. 14, 18, 110, 111, 113, 114, 118

[71] Z. Zivkovic and B. Krose, "An em-like algorithm for color-histogram-based object tracking," in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04), vol. 1. IEEE, 2004, pp. I–798. 14, 18, 134

[72] F. Porikli, "Integral histogram: A fast way to extract histograms in cartesian spaces," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1. IEEE, 2005, pp. 829–836. 14

[73] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in European Conference on Computer Vision (ECCV'04). Springer, 2004, pp. 28–39.

[74] C. Yang, R. Duraiswami, and L. Davis, "Efficient mean-shift tracking via a new similarity measure," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1. IEEE, 2005, pp. 176–183. 14, 113

[75] S. T. Birchfield and S. Rangarajan, "Spatiograms versus histograms for region-based tracking," in IEEE Computer Society Conference on Computer Vision and

Pattern Recognition (CVPR'05), vol. 2. IEEE, 2005, pp. 1158–1163. 14, 18, 91, 113

[76] L. Spinello and K. O. Arras, "People detection in rgb-d data," in IEEE International Conference on Intelligent Robots and Systems (IROS'11). IEEE, 2011, pp. 3838–3843. 15, 21, 74

[77] J. L. Crowley, P. Stelmaszyk, and C. Discours, "Measuring image flow by tracking edge-lines," in 2nd International Conference on Computer Vision (ICCV'88). IEEE, 1988, pp. 658–664. 16, 99

[78] Z. Yin and R. Collins, "Belief propagation in a 3d spatio-temporal mrf for moving object detection," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07). IEEE, 2007, pp. 1–8.

[79] J. Black and T. Ellis, "Multi camera image tracking," Image and Vision Computing, vol. 24, no. 11, pp. 1256–1267, 2006. 16, 19, 42, 84

[80] S. S. Blackman, "Multiple-target tracking with radar applications," Dedham, MA, Artech House, Inc., vol. 1, 1986. 16, 101

[81] P. Konstantinova, A. Udvarev, and T. Semerdjiev, "A study of a target tracking algorithm using global nearest neighbor approach," in Proceedings of International Conference on Computer Systems and Technologies (CompSysTech03), 2003. 16

[82] N. Suganuma, "Clustering and tracking of obstacles using stereo vision system," in ICCAS-SICE International Joint Conference. IEEE, 2009, pp. 4623–4628. 16, 99, 101

[83] J. Munkres, "Algorithms for the assignment and transportation problems," Journal of the Society for Industrial & Applied Mathematics, vol. 5, no. 1, pp. 32–38, 1957. 16, 103

[84] Y.-C. Lim, C.-H. Lee, S. Kwon, and J.-h. Lee, "A fusion method of data association and virtual detection for minimizing track loss and false track," in 4th IEEE Intelligent Vehicles Symposium. IEEE, 2010, pp. 301–306. 16, 99, 103

[85] A. M. Li and P. S. Park, "Long term multi-target tracking based on detection and data association."

[86] M. Munaro and E. Menegatti, "Fast rgb-d people tracking for service robots," Autonomous Robots, pp. 1–16, 2014. 16, 21, 74, 81

[87] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-target tracking using joint probabilistic data association," in 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes, vol. 19. IEEE, 1980, pp. 807–812. 16

[88] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," IEEE Journal of Oceanic Engineering, vol. 8, no. 3, pp. 173–184, 1983. 16, 81, 98

[89] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in Proceedings of IEEE International Conference on Robotics and Automation (ICRA'01), vol. 2. IEEE, 2001, pp. 1665–1670. 16

[90] D. B. Reid, "An algorithm for tracking multiple targets," IEEE Transactions on Automatic Control, vol. 24, no. 6, pp. 843–854, 1979. 16, 98

[91] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," IEEE Aerospace and Electronic Systems Magazine, vol. 19, no. 1, pp. 5–18, 2004. 16, 81

[92] R. W. Sittler, "An optimal data association problem in surveillance theory," IEEE Transactions on Military Electronics, vol. 8, no. 2, pp. 125–139, 1964. 16

[93] A. Amditis, G. Thomaidis, P. Maroudis, P. Lytrivis, and G. Karaseitanidis, "Multiple hypothesis tracking implementation." 16

[94] K. G. Murty, "An algorithm for ranking all the assignments in order of increasing cost," Operations Research, vol. 16, no. 3, pp. 682–687, 1968. 16

[95] S.-W. Joo and R. Chellappa, "A multiple-hypothesis approach for multiobject visual tracking," IEEE Transactions on Image Processing, vol. 16, no. 11, pp. 2849–2854, 2007. 16, 17

[96] I. J. Cox and S. L. Hingorani, "An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 2, pp. 138–150, 1996. 16

[97] M. Luber, L. Spinello, and K. O. Arras, "People tracking in rgb-d data with on-line boosted target models," in IEEE International Conference on Intelligent Robots and Systems (IROS'11). IEEE, 2011, pp. 3844–3849. 16

[98] B. Bose, X. Wang, and E. Grimson, "Multi-class object tracking algorithm that handles fragmentation and grouping," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07). IEEE, 2007, pp. 1–8. 17

[99] T. Stephan and M. Grinberg, "Probabilistic handling of merged detections in multi target tracking," in 9th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS'12). IEEE, 2012, pp. 355–361. 17

[100] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 8, pp. 993–1008, 2003. 17

[101] K. G. Derpanis, "Characterizing image motion," Citeseer, Tech. Rep., 2006. 17

[102] D. Beymer and K. Konolige, "Real-time tracking of multiple people using continuous detection," in IEEE Frame Rate Workshop, 1999. 18

[103] M. Gao and D. Liu, "Multi-object tracking based on improved mean shift," in International Conference on Information Science and Technology (ICIST'13). IEEE, 2013, pp. 1588–1592. 18, 42, 80, 110, 116

[104] C. Beyan and A. Temizel, "Adaptive mean-shift for automated multi object tracking," Computer Vision, IET, vol. 6, no. 1, pp. 1–12, 2012. 18, 80, 110, 114, 116

[105] D. Xu, Y. Wang, and J. An, "Applying a new spatial color histogram in mean-shift based tracking algorithm," in Image and Vision Computing New Zealand, 2005. 18

[106] E. Maggio and A. Cavallaro, "Multi-part target representation for color tracking," in IEEE International Conference on Image Processing (ICIP'05), vol. 1. IEEE, 2005, pp. I–729.

[107] Z. Fan, Y. Wu, and M. Yang, "Multiple collaborative kernel tracking," in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2. IEEE, 2005, pp. 502–509.

[108] A. P. Leung and S. Gong, "Mean-shift tracking with random sampling." in British Machine Vision Conference (BMVC'06), 2006, pp. 729–738. 18

[109] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Scale and orientation adaptive mean shift tracking," Computer Vision, IET, vol. 6, no. 1, pp. 52–61, 2012. 18, 134

[110] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," 1998. 18, 116, 119

[111] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564–577, 2003. 18, 91

[112] X. Li, T. Zhang, X. Shen, and J. Sun, "Object tracking using an adaptive kalman filter combined with mean shift," Optical Engineering, vol. 49, no. 2, pp. 020 503–020 503, 2010. 18, 84

[113] Z. H. Khan, I. Y.-H. Gu, and A. G. Backhouse, "Robust visual object tracking using multi-mode anisotropic mean shift and particle filters," IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 1, pp. 74–87, 2011. 18, 114

[114] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L. Van Gool, "Color-based object tracking in multi-camera environments," in Pattern Recognition. Springer, 2003, pp. 591–599. 19

[115] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: Establishing a common coordinate frame," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 758–767, 2000. 19

[116] J. Ren, M. Xu, J. Orwell, and G. A. Jones, "Multi-camera video surveillance for real-time analysis and reconstruction of soccer games," Machine Vision and Applications, vol. 21, no. 6, pp. 855–863, 2010. 19

[117] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," IEEE Journal of Robotics and Automation, vol. 3, no. 4, pp. 323–344, 1987. 19, 20

[118] Z. Zhang, "A flexible new technique for camera calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330–1334, 2000. 19, 20, 40

[119] T. Svoboda, D. Martinec, and T. Pajdla, "A convenient multicamera self-calibration for virtual environments," PRESENCE: Teleoperators and Virtual Environments, vol. 14, no. 4, pp. 407–422, 2005. 19, 31

[120] J. Renno, J. Orwell, and G. A. Jones, "Learning surveillance tracking models for the self-calibrated ground plane," in British Machine Vision Conference (BMVC'02), 2002, pp. 1–10. 19

[121] D. A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges, and D. Kim, "Shake'n'sense: reducing interference for overlapping structured light depth cameras," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2012, pp. 1933–1936. 19, 30

[122] A. Maimone and H. Fuchs, "Reducing interference between multiple structured light depth sensors using motion," in IEEE Virtual Reality Short Papers and Posters (VRW'12). IEEE, 2012, pp. 51–54. 19

[123] W. Lemkens, P. Kaur, K. Buys, P. Slaets, T. Tuytelaars, and J. De Schutter, "Multi rgb-d camera setup for generating large 3d point clouds," in IEEE International Conference on Intelligent Robots and Systems (IROS'13). IEEE, 2013, pp. 1092–1099. 19, 31

[124] Y. Schröder, A. Scholz, K. Berger, K. Ruhl, S. Guthe, and M. Magnor, "Multiple kinect studies," Computer Graphics, vol. 2, no. 4, p. 6, 2011. 19

[125] F. Faion, S. Friedberger, A. Zea, and U. D. Hanebeck, "Intelligent sensor-scheduling for multi-kinect-tracking," in IEEE International Conference on Intelligent Robots and Systems (IROS'12). IEEE, 2012, pp. 3993–3999. 20

[126] A. Maimone and H. Fuchs, "Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras," in 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR'11). IEEE, 2011, pp. 137–146. 20

[127] K. Berger, K. Ruhl, Y. Schroeder, C. Bruemmer, A. Scholz, and M. A. Magnor, "Markerless motion capture using multiple color-depth sensors." in VMV, 2011, pp. 317–324. 20

[128] A. D. Wilson and H. Benko, "Combining multiple depth cameras and projectors for interactions on, above and between surfaces," in Proceedings of 23rd annual ACM Symposium on User Interface Software and Technology. ACM, 2010, pp. 273–282. 20

[129] M. Quaritsch, M. Kreuzthaler, B. Rinner, H. Bischof, and B. Strobl, "Autonomous multicamera tracking on embedded smart cameras," EURASIP Journal on Embedded Systems, vol. 2007, no. 1, pp. 35–35, 2007. 20

[130] R. Mohedano, C. R. del Blanco, F. Jaureguizar, L. Salgado, and N. García, "Robust 3d people tracking and positioning system in a semi-overlapped multi-camera environment," in 15th IEEE International Conference on Image Processing (ICIP'08). IEEE, 2008, pp. 2656–2659.

[131] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian, "Local fisher discriminant analysis for pedestrian re-identification," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13). IEEE, 2013, pp. 3318–3325. 20

[132] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking across multiple cameras with disjoint views," in Proceedings of 9th IEEE International Conference on Computer Vision (ICCV'03). IEEE, 2003, pp. 952–957. 20

[133] N. Anjum, M. Taj, and A. Cavallaro, "Relative position estimation of non-overlapping cameras," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'07), vol. 2. IEEE, 2007, pp. II–281. 20, 24

[134] D. Makris, T. Ellis, and J. Black, "Bridging the gaps between cameras," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04), vol. 2. IEEE, 2004, pp. II–205. 20

[135] R. K. Kumar, A. Ilie, J.-M. Frahm, and M. Pollefeys, "Simple calibration of non-overlapping cameras with a mirror," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08). IEEE, 2008, pp. 1–7. 20, 24

[136] E. Horbert, K. Rematas, and B. Leibe, "Level-set person segmentation and tracking with multi-region appearance models and top-down shape information," in IEEE International Conference on Computer Vision (ICCV'11). IEEE, 2011, pp. 1871–1878. 21

[137] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," International Journal of Computer Vision, vol. 77, no. 1-3, pp. 157–173, 2008. 21

[138] A. Milan, K. Schindler, and S. Roth, "Challenges of ground truth evaluation of multi-target tracking," in IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'13). IEEE, 2013, pp. 735–742. 21, 22

[139] Viper-gt. [Online]. Available: http://viper-toolkit.sourceforge.net 21

[140] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation," International Journal of Computer Vision, vol. 101, no. 1, pp. 184–204, 2013. 21

[141] J. Black, T. Ellis, and P. Rosin, "A novel method for video tracking performance evaluation," in International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (PETS'03), 2003, pp. 125–132. 21, 22

[142] T. Ellis, "Performance metrics and methods for tracking in surveillance," in Proceedings of 3rd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS'02). Citeseer, 2002, pp. 26–31. 21

[143] M. Motwani, N. Tirpankar, R. Motwani, M. Nicolescu, and F. Harris, "Towards benchmarking of video motion tracking algorithms," in International Conference on Signal Acquisition and Processing (ICSAP'10). IEEE, 2010, pp. 215–219. 21

[144] J. Ferryman and A. Ellis, "Pets2010: Dataset and challenge," in 7th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'10). IEEE, 2010, pp. 143–150. 21

[145] i-lids. the image library for intelligent detection systems. [Online]. Available: http://www.ilids.co.uk 21

[146] Caviar context aware vision using image-based active recognition. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/ 21

[147] A. T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin, "Etiseo, performance evaluation for video surveillance systems," in IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'07). IEEE, 2007, pp. 476–481. 21

[148] S. Song and J. Xiao, "Tracking revisited using rgbd camera: Unified benchmark and baselines." ICCV, 2013. 21, 74

[149] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in IEEE International Conference on Robotics and Automation (ICRA'11). IEEE, 2011, pp. 1817–1824. 21, 74

[150] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from rgbd images," in IEEE International Conference on Robotics and Automation (ICRA'12). IEEE, 2012, pp. 842–849. 21, 74

[151] H. Zhang and L. E. Parker, "4-dimensional local spatio-temporal features for human activity recognition," in IEEE International Conference on Intelligent Robots and Systems (IROS'11). IEEE, 2011, pp. 2044–2049. 21, 74

[152] N. Lazarevic-McManus, J. Renno, D. Makris, and G. A. Jones, "An object-based comparative methodology for motion detection based on the f-measure," Computer Vision and Image Understanding, vol. 111, no. 1, pp. 74–85, 2008. 22, 23

[153] B. Keni and S. Rainer, "Evaluating multiple object tracking performance: the clear mot metrics," EURASIP Journal on Image and Video Processing, vol. 2008, 2008. 22, 23

[154] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, M. Boonstra, and V. Korzhova, "Performance evaluation protocol for face, person and vehicle detection & tracking in video analysis and content extraction (vace-ii)," Computer Science & Engineering University of South Florida, Tampa, 2006. 22

[155] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors," International Journal of Computer Vision, vol. 75, no. 2, pp. 247–266, 2007. 22, 23

[156] F. Yin, D. Makris, and S. A. Velastin, "Performance evaluation of object tracking algorithms," in 10th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS2007), 2007. 22, 23, 123, 125, 135, 141

[157] K. Smith, D. Gatica-Perez, J.-M. Odobez, and S. Ba, "Evaluating multi-object tracking," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'05). IEEE, 2005, pp. 36 36. 22, 23

[158] D. Hall, J. Nascimento, P. Ribeiro, E. Andrade, P. Moreno, S. Pesnel, T. List, R. Emonet, R. Fisher, J. Santos-Victor et al., "Comparison of target detection algorithms using adaptive background models," in 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (PETS'05). IEEE, 2005, pp. 113–120.

[159] J. Nascimento and J. S. Marques, "New performance evaluation metrics for object detection algorithms," in Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS'04), 2004. 23

[160] S. Pingali and J. Segen, "Performance evaluation of people tracking systems," in Proceedings 3rd IEEE Workshop on Applications of Computer Vision (WACV'96). IEEE, 1996, pp. 33–38. 23

[161] B. Yang and R. Nevatia, "An online learned crf model for multi-target tracking," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12). IEEE, 2012, pp. 2034–2041. 23

[162] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, pp. 319–336, 2009. 23

[163] P. Lébraly, E. Royer, O. Ait-Aider, C. Deymier, and M. Dhome, "Fast calibration of embedded non-overlapping cameras," in IEEE International Conference on Robotics and Automation (ICRA'11). IEEE, 2011, pp. 221 227. 24

[164] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," Sensors, vol. 12, no. 2, pp. 1437–1454, 2012. 26, 28

[165] M. R. Andersen, T. Jensen, P. Lisouski, A. K. Mortensen, M. K. Hansen, T. Gregersen, and P. Ahrendt, "Kinect depth sensor evaluation for computer vision applications," Århus Universitet, Tech. Rep., 2012. 26, 28

[166] K. Khoshelham, "Accuracy analysis of kinect depth data," in ISPRS Workshop Laser Scanning, vol. 38, no. 5, 2011, p. W12. 26

[167] T. Mallick, P. Das, and A. Majumdar, "Characterizations of noise in kinect depth images," 2014. 26, 30

[168] J. Smisek, M. Jancosek, and T. Pajdla, "3d with kinect," in Consumer Depth Cameras for Computer Vision. Springer, 2013, pp. 3–25. 28

[169] O. Sorkine, "Least-squares rigid motion using svd," Technical Notes, vol. 120, p. 3, 2009. 32, 35, 36

[170] (2014, Jun.) Openni. [Online]. Available: https://github.com/OpenNI/OpenNI 40

[171] W. Garage. (2014, Jun.) Robot operating system. [Online]. Available: http://www.ros.org/ 40

[172] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in ICRA workshop on Open Source Software, vol. 3, no. 3.2, 2009, p. 5. 40

[173] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle, "Appearance models for occlusion handling," Image and Vision Computing, vol. 24, no. 11, pp. 1233–1243, 2006. 42, 84

[174] M. M. Khan, T. W. Awan, I. Kim, and Y. Soh, "Tracking occluded objects using kalman filter and color information." 42

[175] K. Hayashi, M. Hashimoto, K. Sumi, and K. Sasakawa, "Multiple-person tracker with a fixed slanting stereo camera," in Proceedings of 6th IEEE International Conference on Automatic Face and Gesture Recognition. IEEE, 2004, pp. 681–686. 42

[176] M. Harville, "Stereo person tracking with adaptive plan-view templates of height and occupancy statistics," Image and Vision Computing, vol. 22, no. 2, pp. 127–142, 2004. 42

[177] W. Qia, Y. Yangb, M. Yi, Y. Li, Z. Pizloc, and L. J. Lateckib, "Robust object tracking based on rgb-d camera." 42

[178] W. Choi, C. Pantofaru, and S. Savarese, "Detecting and tracking people using an rgb-d camera via multiple detector fusion," in IEEE International Conference on Computer Vision Workshops (ICCV'11 Workshops). IEEE, 2011, pp. 1076–1083.

[179] L. Xia, C.-C. Chen, and J. Aggarwal, "Human detection using depth information by kinect," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'11). IEEE, 2011, pp. 15–22. 42

[180] J. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, no. 6, pp. 679–698, 1986. 63, 65

[181] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, Bayesian multiple target tracking. Artech House, 2013. 81

[182] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," International Journal of Computer Vision, vol. 77, no. 1-3, pp. 125–141, 2008. 91

[183] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, and J. Cheng, "Visual tracking via incremental log-euclidean riemannian subspace learning," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08). IEEE, 2008, pp. 1–8. 91

[184] Q. Zhao, Z. Yang, and H. Tao, "Differential earth mover's distance with its applications to visual tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 2, pp. 274–287, 2010. 91

[185] C. Ó. Conaire, N. E. O'Connor, and A. F. Smeaton, "An improved spatiogram similarity measure for robust object localisation," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'07), vol. 1. IEEE, 2007, pp. I–1069. 92, 93

[186] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," IEEE Control Systems, vol. 29, no. 6, pp. 82–100, 2009. 98

[187] H. W. Kuhn, "The hungarian method for the assignment problem," Naval research logistics quarterly, vol. 2, no. 1-2, pp. 83–97, 1955. 103

[188] I. Leichter, M. Lindenbaum, and E. Rivlin, "Mean shift tracking with multiple reference color histograms," Computer Vision and Image Understanding, vol. 114, no. 3, pp. 400–408, 2010. 110, 113, 114

[189] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," IEEE Transactions on Information Theory, vol. 21, no. 1, pp. 32–40, 1975. 110

[190] J. Zhang, J. Fang, and J. Lu, "Mean-shift algorithm integrating with surf for tracking," in 7th International Conference on Natural Computation (ICNC'11), vol. 2. IEEE, 2011, pp. 960–963. 113