Sensing shared space using 3D stereo vision

Rishabh Mittal

TU Delft ARUP

This page is intentionally left blank.

SENSING SHARED SPACE USING 3D STEREO VISION

MSc THESIS

by

RISHABH MITTAL

Defended on

20th January 2020

Thesis submitted in partial fulfilment of the requirements for the degree of

Master of Science at Delft University of Technology,

Faculty of Civil Engineering and Geosciences,

Department of Transport and Planning.

Student number:	4721772	
Graduation Committee :	Prof. dr. ir S.P. Hoogendoorn	committee chair, TU Delft
	Dr. ir. W. Daamen	supervisor, TU Delft
	Dr. ir. Y. Yuan	daily supervisor, TU Delft
	Dr. ir. S.C. van der Spek	external supervisor, TU Delft
	Ir. Laurens Tait	company supervisor, Arup
	Ir. Thomas Paul	company supervisor, Arup

This page is intentionally left blank.

Acknowledgements

My time at TU Delft has transformed me as a person and has allowed me to discover the researcher inside me. There have been a number of people who stood by me during the testing times and always made sure I was in a positive frame of mind. I would like to take this opportunity to acknowledge their role during this transformational journey.

I would like to thank Dr. ir. Winnie Daamen and Dr. ir. Yufei Yuan for their rigorous feedback and continuous guidance which helped me learn, improve and enhance my capabilities as a researcher.

I thank Prof. dr. ir. S.P.Hoogendoorn and Dr. ir. S.C.van der Spek for providing me with their valuable insights and encouraging me to investigate this research topic from a different point of view.

I would also like to thank my company supervisors Ir. Laurens Tait and Ir. Thomas Paul for constantly cheering me up and providing me with a broader perspective of this thesis work. Thomas and I frequently had long and in-depth discussions which provided me with much needed encouragement during my research. Special thanks to Senno Kaasjager who helped me all along during the software development phase of my thesis. My fellow colleagues at Arup also boosted my morale by providing a nurturing atmosphere.

I thank Edwin Scharp and Peter van Oossanen for providing their technical support and showing utmost patience and professionalism while conducting the data collection experiments.

I thank Tim Jonathan for permitting me to conduct the experiment within the Green Village premises. I am grateful to all the volunteers who took part in this experiment and agreed to walk and bike even during the scorching summer heat.

My sincere gratitude to Riaaz Bhailal from the Municipality of Amsterdam for arranging the permissions needed to perform the field experiment near the Amsterdam Central Station.

All this would never have been possible without the unwavering support and encouragement from my loving parents. I also thank Shivanjali for her faithful love and I am grateful to my housemate Kashyap and all my friends who have filled this journey with memories to be cherished for life.

Rishabh Mittal
Delft, January 2020

This page is intentionally left blank.

Summary

Background and Research objective

There is a recent trend of rising need and popularity of non-motorised modes of transport as part of urban travel constituting mainly of pedestrians and bicycles. In the Netherlands, there is an increasing number of areas where both pedestrians and cyclists share the same space in the absence of user segregation by any road markings. Such spaces are said to enhance safety, resolve spatial limitations and provide a sustainable solution to the overall traffic problem. The increasing number of such shared spaces has created a practical need for in-depth analysis of the user's behaviour within these spaces.

Many studies focusing on shared space interactions are limited by the current methods of data collection and data extraction. Other studies on people movement also face similar issues. The real-world data collections of visual data require the cameras to be installed in a tilted orientation which makes it difficult for the researcher to determine the agent's location on the ground plane. The use of a 3D-camera to collect three-dimensional data of the scene can be of value to overcome this challenge. Thus, to improve the data collection process, the stereo-vision technology is investigated. The current real-world data processing also relies heavily on manual labour as it is challenging for the traditional frameworks to detect agents appearing in such dynamic ways. To avoid such labour intensive processes, automating the data extraction collected in a controlled environment using certain rule-based approaches have increasingly been adopted by the researchers. This is creating a gap between the behaviours studied by the researchers and the real world behaviours. In order to fill this gap and encourage studies on real-world datasets, this research aims to enhance the current data collection methods and automate the data extraction process for real-world datasets. To achieve this research objective, the following research question is proposed:

***W*hat can be a data gathering and processing framework to automatically extract trajectories of cyclists and pedestrians in a shared space environment using a 3D-stereo vision camera?**

In order to deliver on the research objective, this research is broken down into five stages, (i) Stereo camera selection, (ii) Data collection, (iii) Agent detection, (iv) Ground plane representation and (v) Agent tracking. The last three stages are part of the data processing framework.

Stereo camera selection

The features identified for comparing different cameras available in the market were: (i) sensor's type, (ii) depth sensing range, (iii) recording quality and (iv) software support. The cameras available in the market were compared and scored based on these features and the best possible camera for this research was selected. The Zed camera by Stereolabs was selected and was used to record with 720p resolution at 60fps. The camera's software is able to correct the recorded images for distortion and calculate the 3D coordinates of each pixel in the image. This data is stored in the form of X, Y, Z value w.r.t. the camera's coordinate system. This camera was used to collect data during this thesis.

Data collections

Two data collections were performed, first in a controlled setting to test the camera setup and to use the collected data to develop the data processing framework in the later stages. This experiment was performed in the Green Village area of TU Delft. It consisted of 16 volunteers and three different scenarios based on the modes of travel. The second data collection was conducted in a real-world shared space area. This space was selected such that it was mainly shared between pedestrians and cyclists, traffic flow was multi-directional and, allowed to collect data for different crowding conditions. The shared space behind Amsterdam Central station which is also near to the ferry terminal was selected. This dataset was used to test the developed framework on real-world dataset.

Data processing framework

The data processing framework can be divided into three stages, (i) agent detection, (ii) depth extraction and ground representation and, (iii) agent tracking. The output from these two stages was then used as an input in the third stage of tracking. Following is a brief explanation of each stage:

- *Agent detection:* A neural network based detection model was selected out of many models based on literature and first-hand comparison. Faster R-CNN detection model was used in this research [1]. The object type to be detected was set at people which includes both pedestrians and cyclists. The detections were performed on the visual images recorded by the camera and the output was in the form of bounding boxes around the agent for every image in the video.
- *Depth extraction and ground representation:* This stage combines the bounding boxes from the previous step and the depth information calculated by the camera to project the agent on the ground. First, the combination of reference box and statistical operation is used to refer to the agent's 3D-coordinate w.r.t. the tilted camera axis. As the extracted agent coordinated are on a tilted coordinate system, the coordinates are rotated using the camera's angle of tilt. The resultant coordinate system has one of its planes parallel to the ground plane. Agent coordinates on this plane are used to project it on the ground plane. During the design process, various parameter values were tuned using a small sample from the Green Village dataset. Similar to detections, the detected agent's 2D-points on the ground plane are also obtained for every image of the video.

- *Agent tracking:* The projected points on the ground plane are anonymous, i.e. they are not linked with one another across the video. This information of the agent's location is the only data available for tracking. Moreover, in cases when there are multiple agents available in a single frame, there exist multiple points which need to be correctly associated with one another across the frames. For this, the SORT tracking model was used in this research [2]. The parameters of the tracking model were identified and tuned on the sample dataset from the Green Village recordings. In the final output, the anonymous points are assigned with an id number which can be extracted and plotted to derive the agent's trajectories on the ground plane.

Results & Recommendations

The detections were sensitive to three main factors (i) occlusion, (ii) total number of agents and (iii) agent's shadows. Occlusion leads to the loss of the agent's visual information used by the model to detect them. The inability of this model to detect agents even when they perfectly visible was exhibited in case of high crowding conditions ($>$ 15 agents). Both of these sensitivity parameters frequently occur in high crowding scenarios than in the lower scenarios which makes the detection model ineffective in such scenarios. This can be a result of biases generated during the training of this model and can be improved by providing a more representative training dataset. For lower crowding conditions, the model was able to provide stable and accurate detections for all the visible agents. Some rare moments of wrong (false-positive) detections were given by the model while detecting people's shadows or their reflections in the glass as true detections. While most of the error was in terms of missed(false-negative) detection only.

The depth extraction and ground representation process was largely dependent on the number and quality of detection boxes provided by the previous stage. When the detections were stable and continuous, the projected points also maintained their stability and continuity. Small vibrations and other anomalies in the projected points were observed due to subtle changes in the agent's depth information. Thus, the camera was able to measure even small changes in the observed area. For larger distances ($>$12 meters), the camera was unable to estimate the depth of the observed area accurately. Exploring better ways to use the camera and investigating multi-camera setups can help improve the accuracy and reliability of depth estimation for larger distances.

The tracking model was able to account for some of the errors (false-positives, false-negatives) passed on by the detection model in terms of missed and wrong detections. In moments of brief occlusions, when the detections are only missed for brief time periods but are available otherwise, the tracking model was able to assign ids across the missed detections also. Such situations mostly occurred during low crowding scenario. In case of higher crowds, as the detections were mostly unavailable, the tracking model was not able to provide continuous trajectories. More information (such as agent's visual information) can be included in the tracking model to improve its performance.

Main Contributions

This thesis analysed the working of Zed stereo camera and provided a methodology to collect real-world data of a shared-space environment using this camera . The data collected during the two data collection experiments is also a contribution which can be used to further develop and improve the proposed framework. This project has integrated various different state-of-the-art processes into one framework and has automated the data extraction processes. As automation reduces the cost of data extraction, it will allow researchers to include larger real-world datasets into there studies. Lower costs will also encourage municipalities and private consultancies to undertake more of such studies themselves and help design better infrastructure.

Future work

To further encourage such studies on people movement and 3D-data collections, the camera's supporting hardware needs to be much lighter, and easy to install on-site. As of now, the processing framework is very top-down in nature and lacks a feedback loop between different processes. Sharing of information between different data processing stages are interesting to explore and can enhance the performance of this framework. To overcome the problem of occlusion, multi-camera setups is one of the possible solutions to be investigated which will also increase the depth range and accuracy of the camera setups. Lastly, an automatic mode based differentiation and tracking of people can further advance this framework towards automating the data processing framework in the future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As the non-motorised modes of transport become popular and their use is encouraged by governments around the world, its is creating a need for urban planners to better understand the movement patterns of these modes. Designing an efficient, robust and reliable data collection and processing framework to gather information on people's movement is the first step in understanding the people's movements. This thesis works on developing such a framework using a 3D stereo vision camera. The introduction chapter elaborates on the research objectives, questions and the approach used to design the data collection and processing framework. The last section lays out the report structure followed in this thesis.

1.1 Background and Motivation

In Amsterdam, the share of active modes is as high as 61% out of which 29% of the trips are on foot and 32% are by bicycle [5]. This rise in popularity of active modes has resulted in the presence of three to four modes (cars, buses, bicycles and pedestrians) on the urban transport infrastructure. Providing all these modes with a sufficient right-of-way is creating a design challenge amongst traffic engineers. Other than the spatial constraints, safety of people on the roads is another concern of the governments around the world. The incidences involving conflict between motorised and non-motorised conflict have also been increasing [6]. As the active modes (cyclists and pedestrians) have low speeds of travel, it creates an opportunity for urban planners to facilitate a shared infrastructure for the non-motorised traffic [7]. Within Netherlands, such spaces can be observed near shopping areas, public transport hubs and sometimes near commercial and educational institutions. Apart from providing safety and solving the spatial constrains, these shared space are said to increase livability, encourage local economic activities, reduce emissions and improve air quality of the surrounding [8, 9]. In such spaces, people exhibit interesting and complex interactions as different modes need to negotiate for space and balance their priorities. The road users tend to adapt to each others behaviour much more dynamically than in a uni-modal environment [10]. The presence of multiple modes without any lane discipline makes the modelling of such interactions more challenging. Most of the early modelling approaches have been motivated on the basis of feeling and intuition and lacked enough data based support [11, 12]. Some of the more recent modelling approaches which make use of the real-world datasets were limited by the current data collection and data extraction approaches and used a small dataset [13].

Collecting data on people movement within these spaces will provide deeper insights into the behaviour and movement patterns of people. Moreover, the movement data collected can also be used to better evaluate the safety, efficiency, walkability and usability of the shared space infrastructure itself. Supporting the behavioural and modelling studies with a larger real-world dataset will enhance the credibility and reliability of the future researches. To facilitate this, there is a need to identify and improve upon the challenges faced during data collection and data extraction.

Recent studies on people movement in real-world spaces rely on manual data extraction processes by clicking on each and every person in the image thorough all the frames of a video [10]. These repetitive manual tasks are boring, costly, and end up consuming a lot of valuable research time. For perspective, if a ten minute video was to be analysed (consisting of 5 people in every frame recorded at 60 frames per second), it will amount to a total of 180,000 clicks. Such cumbersome processes are inefficient and limits the researcher's ability to use large real-world datasets. As a result, some of the researches have also resorted towards collecting data on people movement under a controlled environment [14, 15]. Here, mostly a rule-based methodology (e.g. based on specific colour) was used to automate the data extraction. Such rule-based criteria are very limited as they can be applied under very specific and controlled conditions. Thus, it limits the researchers to these custom datasets and they are unable to study people movement in real-world environments. This lack of sufficient real-world datasets in the behavioural and modelling studies on people movement suggests that there is a need for a better data extracting approach.

Most of the researches use a monocular video camera to collect data in a shared space environment [16, 17, 18]. This also limits the kind of information that can be extracted from such data. Parts of the data extraction process can also be affected by the quality and the approach of data collection. When collecting data using a video camera, they can be installed in two orientations, overhead orientation which is vertically above the area of interest or tilted orientation when the line of sight is at an angle to the horizontal ground plane. The overhead orientation helps in tracking the motion in the horizontal ground plane but reduces the viewing area and limits the data collecting to roofed environments only. The tilted orientation increases the viewing area and allows the camera to be mounted on any vertical structure enabling the data collection in open spaces where the majority of shared spaces exist. But, this tilted orientation introduces another challenge of representing the agent on the horizontal ground plane (such that the camera was installed overhead). Here, a new spatial axis representing the depth of each pixel in the image is introduced which needs to be estimated. The perception of depth is a limitation for monocular video cameras but can be overcome by using the recent advancements in camera technology (such as depth sensing 3D-cameras).

Based on the motivation and some research gaps identified in this section, the next section formulates the research object and questions. This is followed by an explanation into the research approach taken during this thesis. The last section provides the structure of this thesis report.

1.2 Research objective and questions

This research investigates the possibilities of integrating the data output from the 3D-stereo camera with the existing agent detection techniques to provide outputs necessary for studying people's movement. The developed data processing framework should be applicable in the real-world environments with a focus on shared spaces environments. The data extraction process should be automated to allow processing of larger datasets. From this research objective and the above research background, the following research question is formulated:

What can be a data gathering and processing framework to automatically extract trajectories of cyclists and pedestrians in a shared space environment using a 3D-stereo vision camera?

In this research question, the term 'data gathering and processing framework' focuses on developing a data collection and data extraction framework. The question builds this research around three main areas which are: the observing technology, the interacting agents and the type of environment. The 3D-stereo vision is the observing technology, the pedestrians and cyclists are the interacting agents and the shared-space is the type of environment where the interactions happen. The main research question is broken down to formulate the following sub-questions:

- Sub-question (1): *Which features of the stereo-vision camera are to be considered for recording the movement of people in shared spaces?*
  This research investigates on using stereo vision technology to gather data in shared space environments. This question helps to identify the important feature of the stereo-camera which leads to acquiring one of the many cameras available in the market.

- Sub-question (2): *Which factors are considered while selecting the location and designing of the data collection experiments?*
  This question helps to design the data collection experiments and develop a data gathering framework within real-world shared-spaces using the acquired stereo-vision camera.

- Sub-question (3): *From the data recorded using the stereo camera, how can the agents (pedestrians and cyclists) be identified and localised on the ground plane?*
  This question focuses on developing a data processing framework to perform two tasks which are agent identification and agent localisation using the recorded data. The agent is represented on the ground plane with its respective 2D-coordinates.

- Sub-question (4): *Which of the existing object tracking frameworks can be integrated with the extracted agent coordinates to provide trajectories on the ground plane?*
  After obtaining the agent's location on the ground plane, this question investigates the application of this data to derive agent's trajectory.

The next section outlines the research approach to fulfil the research objective by elaborating on each sub-question one-by-one.

1.3 Research Approach

The above research questions and research objectives are broken down into five stages namely, (i) stereo camera selection, (ii) data collection, (iii) agent detection (iv) ground plane representation (v) agent tracking. Each of the following paragraphs explains every stage in detail.

- **Stereo camera selection:** First and the foremost step in this research is to identify the key features needed in a stereo-vision camera to collect the data in real-world public spaces. During the data collection, this camera is to be installed at a certain height to capture movement of both pedestrians and cyclists. Thus, the selected camera should have sufficient depth sensing range and frame-rate to capture outdoor scenarios. When working with such new camera technologies and devices, its usability depends largely on the software support provided by the manufacturer and its penetrations into the online, open-source community. The later stages of data collection and agent localisation methodology is directly affected by the quality and usability of the recorded data. More of such features are identified from the literature and market survey. After comparing different stereo cameras available in the market, the camera which provides the best features for this research was selected.

- **Data Collection:** After acquiring the stereo camera, the next step is to collect the data using this camera. The aim of the data collection was to test the camera setup in real-world environments and to use the collected data in the later stages. Two data collection experiments were performed, one in a controlled environment and another in a real-world environment. For controlled environment, the data collection needs to be designed in terms of the camera position and the movement patterns of the observed agents (both pedestrians and cyclists). Later, this data is then used to develop the data processing framework to localise and track the agents. For data collection in a real-world shared-space environment, the site characteristics and the camera position are to be considered beforehand. As these spaces are public, permissions from the local authorities largely govern the decisions made during this experiment. This data is used to apply the data processing framework developed on the controlled dataset and assess its applicability in a real-world scenario.

- **Agent detection:** To enable the use of large real-world datasets, the process of data extraction from the recorded data needs to be automated. Thus, the process of identifying and localising agents should be automatic and reliable. The first step is to review some of the existing state-of-the-art object detection techniques. In a real-world scenario, there are multiple object types other than just people which are moving (or not moving) at different speeds and directions. Many of such objects move along with the people like pet animals, suitcases, bikes, scooters, etc. People themselves can appear in different shapes and sizes depending on there outfit. These visual characteristics constantly change as they move through the observed space. Different detection approaches are assessed based on their ability to overcome such challenges using the literature. Following this, one of the detection approach is selected for further investigation. The selected detection method is then investigated further into different detection models proposed within this approach. In this research, the shared-space environment will consist of people as both pedestrians and cyclists. The possibility to distinguish these two modes is also examined.

- **Ground plane representation:** The selected detection model is then integrated with the stereo camera's output data to obtain the desired localisation of agent on the ground plane. Here, the depth data is used such that even when the camera records the data from a tilted orientation, the proposed framework is able to view the agents on the horizontal ground plane as they were observed from an overhead position.
- **Agent tracking:** Again, to enable the use of large datasets, the process of agent tracking also needs to be automated. Two main challenges facing the automation of this step are the possibility of discontinuous points and the existence of multiple points on the ground plane. Unlike the manual localisation approaches, the automation of agent localisation in the previous step will result in discontinuous and fragmented set of ground plane points. Also, there will be multiple points representing each agents for every time-step. These points need to be correctly differentiated and associated with each other throughout the recorded data. Literature review into the existing tracking algorithms is performed and an algorithm is selected such that it is able to overcome both these challenges using only the agent's location information. Using such an algorithm also enabled this research to qualitatively comment on the performance of the previous steps of agent localisation.

1.4 Report structure

After the introduction, the next chapter provides the literature review of the existing methodologies applied by pedestrians and shared-space studies to extract data from visual videos and on the related works done in the field of computer vision and people tracking. Chapter 3 provides an overview of the methodology used in this research. Chapter 4 answers the first sub-question by identifying the camera features and selecting the stereo-camera to be used in this research. The next chapter, chapter 5 describes the data collection process using the selected camera. Here, two data collections were performed each explained in section 5.1 and section 5.2 respectively. One of the data collected was used to develop the data processing framework in chapters 6, 7 & 8. The third sub-question regarding agent localisation on the ground plane is addressed in chapters 6 & 7. After localising the agent, chapter 8 deals with the tracking of agent. The final framework consisting of all the design choices and the tuned parameters is implemented on the real-world dataset in chapter 9. Each of the above chapters includes its own discussion and summary sections for better readability. The report is concluded in chapter 10 which provides an overall discussion, answer to research questions, main contributions and a note on future possibilities of this study.

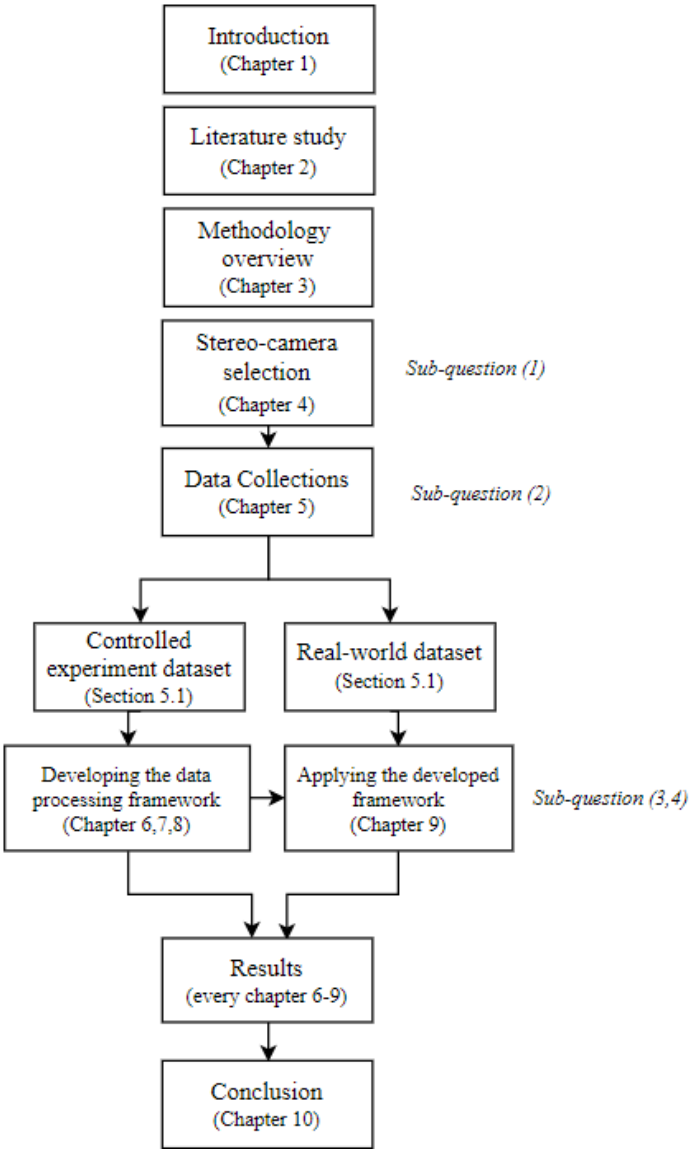


Figure 1.1: Thesis report structure

Chapter 2

Literature Study

This chapter is dedicated to the literature study of the previous works done in the field of people identification, people tracking and stereo-vision camera. The paragraphs follow a timeline starting from some of the early works and moves on to the more recent state-of-the-art approaches in the field of computer vision. For every time period, a relation between the advancements in the field of computer vision and the approach of data extraction in the studies published on people movement has been made. Identifying such relationships helps to justify and analyse the impact of this study in the field of computer vision. In the process, the challenges faced by previous studies while tracking people are identified which guides the development of data collection and processing framework in this research. Much of these literature have been revisited in the later chapters to inform various design decisions.

People tracking using stereo-vision - early works and challenges

Some of the early studies published in the field of people identification and tracking using a stereo camera setup largely focus on the stereo camera setup itself [19, 20, 21, 22]. Due to the lack of readily available stereo camera setups, all of the researches during this time used a self-made stereo camera setup. Thus, the researchers were limited by the camera capabilities at the time and were focusing more on the camera's depth sensing abilities rather than the identifying people. Darrell et. al. used a combination of colour, depth and pattern detection to identify pixels belonging to people [21]. Here, the features of people were hand-coded into the detection program which limited its large scale applicability. These early self-made camera setups were also limited by their depth sensing range [23]. Due to these limitations in data gathering, most of the very early modelling approaches were motivated by the researcher's intuition and experience [11, 12, 24]. Thus, they lacked the supporting of large, real-world data on movement of people.

Stereo vision - recent developments

Since then, the field of computer vision has come a long way. Nowadays, there are readily available stereo-camera setups in the market which come with a generic software support [25, 26, 27, 28, 29, 30]. All of these cameras provide both depth information and the visual information (i.e. normal video recording) of the scene. These cameras can be differentiated based on their depth sensing range, video quality, software support and its market price. Some of the studies compare different cameras based on their sensor type [31, 32, 33]. Such

comparative studies are used to guide the camera selection process in the later in this report (section 4.1). As the visual information of the scene is available from the camera, the following paragraphs review some of the work done on the people detection using visual images.

Data collection

From a camera's perspective, some of the variables identified during the data collection are lighting/shadows and internal variables like angle of observation, height from the ground level and its recording configurations(like frame rate, resolution) [34, 35]. These camera variables directly affect the quality of data that was to be collected and processed by the software. Thus, impacting the performance of object detection, depth estimation and tracking of agents. These variables are further discussed in the data collection chapter (chapter 5).

People detection and tracking - current practices and challenges

The availability of good quality, affordable, easy to use camera setups has certainly encouraged the researchers to include datasets on people movement within their researches. As described in the previous paragraph, the automation of data extraction is still limited. As a result, these recent studies resort to either manual approaches or other rule-based approaches to extract data from videos [36, 15, 37]. In cases of real-world datasets, these researches use manual data extraction approach by clicking on each of the observed agent throughout all the frames of the video. An example of a rule-based approach is colour based detection and tracking of people in a controlled environments [35, 15, 38]. Such criteria are limited in there application as they can only be used when the data is collected in a controlled experimental setup. Recent studies on shared space modelling also rely on similar approaches to extract data from the video footage [10, 7, 14]. All of the above researches collectively identify the challenges faced during the automation of data extraction from visual images. In the real-world, people are seen in many different heights, body types, dressing styles and so on. Moreover, as people move through space, their body posture and orientation also changes constantly [34]. All this makes the problem of detecting people automatically very challenging.

In year 2001, the first real-time face detector was released by Viola et.al., which came to be know as Viola-Jones algorithm [39]. Later in year 2005, Navneet Dala and Bill Triggs proposed a new methodology describing Histogram of Oriented Gradients (HOG) feature extraction [40]. Since then, there has been an increasing number of studies to automate the agent detection in a real-world environment. Some of the survey papers shortlist the methodologies used in such approaches and identify the challenges faced by them [41, 42]. All these studies hand-code most of the object features to detect them automatically on an image. This limits the number of features which were considered while detecting an object. Thus, limiting the accuracy and versatility of such rule-based algorithms.

People detection and tracking - recent developments

In year 2012, one of the first neural-network based image classifier was published which outperformed all of the traditional rule-based models [43]. A survey paper by Brunetti et.al. on pedestrian detection and tracking shows an emerging trend towards neural-network based object detection in the field of computer vision [44]. These detection models are being applied in various fields from agriculture [45, 46], healthcare [47], manufacturing [13], construction

[48] and many more. Such applications of neural-network based detection models motivates this research to investigate further into this detection approach.

The more recent object detectors not just classify an image but are capable of categorising and localising multiple objects within an image [49]. One of the main disadvantages of this approach is that the detection models need to be trained using large and labelled datasets [44]. Thus, this research investigates the pre-trained versions of these models available online. Some of the most used object detection models are Single shot detector (SSD) [50], You Only Look Once (YOLO) [51] and Region based CNNs (R-CNN) [52] due to there overall performance. Each of these detection models have different neural network architecture which leads to variation in the way information is processed within these models. Thus, each model has its own set of advantages and disadvantages based on its training and performance parameters [44]. Pre-trained versions of these detection models (capable of detecting upto ninety different types of objects) are readily available online for use [53]. As output, these detection models provide object category, bounding box coordinates and its confidence score. This output format remains the same across all the models mentioned above [54]. In 2017, Google published a study comparing the performance and speed of a few of these detection models [49]. This study does not focus on people detection specifically but on overall object detection by these models. An elaborate selection process used in this study is provided in section 6.1. Within literature, there is a scarcity of such comparative studies mainly due to a lack of standardisation in the labelled datasets [55, 49, 44].

For tracking using visual images, the earlier approaches were largely focused on minimising the errors caused by the ineffective detection algorithms [56]. After the improvement and increased reliability of the detection models, there has been increasing number of proposed methodologies for tracking object using the detection information. One of the approaches for tracking is to only use the object's location in the image [2]. Another approach towards tracking has been to extract and use the visual information of the object from the image together with the object's location information to perform tracking[57, 58, 59, 60]. Section 8.1 elaborates more on the selection process of these tracking models for this research.

Summary and Conclusion

People's detection and tracking is challenging mainly due to the large variety in people's appearances which is also dynamically changing as people move through space. The traditional frameworks of detecting people in such dynamic real-world environments have proven to be ineffective as the rules and features of detections were hand-coded. As a result, the researchers either use manual processes to extract data from a real-world footage or resort to collecting data in a controlled setup. Recent developments in the field of stereo-vision cameras and people detection show promising results as reported in the literature. As the stereo-vision cameras record both visual data and depth data of the scene, these data can be integrated with the neural-network based detection models to automate the people detection and tracking processes. Later chapters revisit the literature mentioned in this chapter to inform the selection of camera, the detection model and the tracking model.

Chapter 3

Methodology Overview

This chapter provides an overview into the methodology adopted in this research to fulfil the research objective and answer all the research questions. Table 3.1 provides a summary of all the points discussed below.

Stereo camera selection

The very first step in this research was to select the 3D-camera (with its supporting hardware). The reason behind using such a camera was to investigate its value over the traditional monocular cameras which are generally used for data collections. The step deals with the first sub-research question by identifying the factors affecting the camera selection process using the available literature and by surveying the camera's market. Various cameras in the market were scored based on their performance in each of the identified selection criteria. This process helped in shortlisting and acquiring the best possible camera for this research.

Data collection

The next step was to collect data on people's movement using the selected camera. This step was aimed at testing the camera setup in real-world conditions, designing a data collection framework and using the collected data to design/test the data processing framework. This is addressed by the second sub-question which helps in selecting and designing the data collection experiments. As a result, two data collection experiments were planned, one in a controlled environment while another in a real-world shared space environment. The data from the first experiment was used to mainly design the data processing framework. After identifying and rectifying some of the shortfalls during this experiment, the second experiment was performed. The real-world dataset was used to apply the data processing framework and to qualitatively assess its performance.

Designing the data processing framework

The third and the fourth sub-research questions deal with developing the data processing framework to obtain the trajectories on the ground plane. The data processing framework was divided into three stages, (i) agent detection, (ii) depth extraction with ground representation and, (iii) agent tracking. Each of these stages was designed using the data from the controlled

experiment.

Agent detection: The third sub-question is answered in the first two data processing stages. The agent detection stage uses the visual video recorded by the camera and processes it using a neural-network based object detection model. Out of the many available detections models, one model was chosen using literature and first-hand analysis. As this research was processing videos offline, accuracy of results were more valued than the model's processing speed. The first hand comparison of three detection models was done on the controlled dataset due to the scarcity of such comparative literature studies. The results obtained from the detection model was in the form of bounding boxes around every detected agent for each frame of the video.

Depth extraction and ground representation: In the depth extraction stage, the depth map calculated by the camera was combined with the detection box's image coordinates to extract the agent's 3D-coordinates. A combination of reference box and statistical approach was formulated to obtain stable and reliable results. The design parameters for these operations were optimised using the calculated coordinates of an agent across some sample frames. As the stereo camera was tilted w.r.t the ground plane, the extracted 3D-points of the agents were to be transformed. This process was done in the ground representation stage. Thus, the original coordinate system was transformed (or rotated) such that one of its plane is parallel to the ground plane. The 2D coordinates of the agents along this plane were used for ground plane representation.

Agent tracking: The agent's projected points on the ground plane were anonymous. As there are multiple agents in each frame, these anonymous points need to be automatically associated with each other to form a trajectory. In the agent tracking stage, literature survey was performed to select a tracking model capable of this association using the available 2D coordinates. The parameters of the selected model were identified and tuned based on their performance.

Framework application - Amsterdam dataset

As the main research question aims at developing a framework for real-world datasets, thus chapter 9 applies the designed framework on the Amsterdam Central dataset. This helps to identify the advantages and challenges of the proposed framework in a real-world scenario. The results obtained during each step of this research are analysed and discussed in every chapter of this report. The conclusion chapter summarises the main discussion points and makes recommendations to overcome some of the challenges identified in the process.

Table 3.1: Methodology overview for each research question by summarising its objective, reasoning and the process used to answer the respective questions.

Research question	**Objective**	**Reasoning**	**Process**	**Chapter**
Sub-question(1)	Camera selection based on its features for recording shared spaces.	Investigate the stereo camera for real-world use.	Stereo-camera selection	Chapter 4
Sub-question(2)	Identify features for selecting & designing the data collection using the selected camera.	• Design the data collection framework, • Use the collected data to design/test the data processing framework, • Test the camera setup.	Performing two data collections: • Controlled experiment, • Real-world experiment	Chapter 5
Sub-question(3)	Identify and localise the agents using the recorded data.	To automatically extract ground coordinates of agents with every time-step.	• Agent detection using visual video. • Represent agent on ground using the 3D-depth data.	Chapter 6 & Chapter 7
Sub-question(4)	Obtain agent trajectories using their location information.	Automatically derive trajectories using the agent's ground plane coordinates.	Agent tracking	Chapter 8
Main question	Applying the final framework on a real-world data set.	To identify the advantages and challenges of the proposed framework in a real-world scenario.	Framework application - Amsterdam dataset	Chapter 9

Chapter 4

Stereo-camera selection and use

The main component of the hardware setup while collecting data was the depth camera sensor. This chapter focuses on the selection and usage of the stereo depth camera. This camera was later used to collect and extract 3D-information of the observed scene.

The first section elaborates on the selection process of identifying suitable 3D-camera for sensing shared spaces. Here, the features affecting the camera selection process were identified and different cameras available in the market were scored based on these features. Later sections describe the processing of using this 3D-camera and the output provide by its generic software. It also outlines some of the hardware configuration required to use the camera at its full potential.

4.1 Camera selection

The features considered while comparing and selecting a depth camera were identified based on the literature comparing different stereo-camera technologies and on preliminary market survey. These features were: the type of depth sensor, its depth sensing range, image resolution, frame rate of recording, software support provided by the manufacturer and the market price of the camera (figure 4.1). The type of depth sensor was determined by its method of collecting depth data from its surroundings (active or passive). The sensor's type also affects the camera's ability to work in outdoor conditions (under direct sunlight) and the computational support needed for video processing. Depth sensing range was the maximum distance up to which a camera can sense depth. This factor was important as the camera was to be installed in public spaces at a certain height to observe large areas. Frame rate and resolution of the recorded coloured images and the depth map directly impacts the quality of collected data. The level of software support was determined by the combination of generic software provided by the manufacturer and the online support provided by the camera's user community. Software support was an essential factor as it will determine the overall ease of using the depth camera and the methodology of data extraction and processing during this research. Lastly, the price of the depth camera was also a factor in the camera selection process.

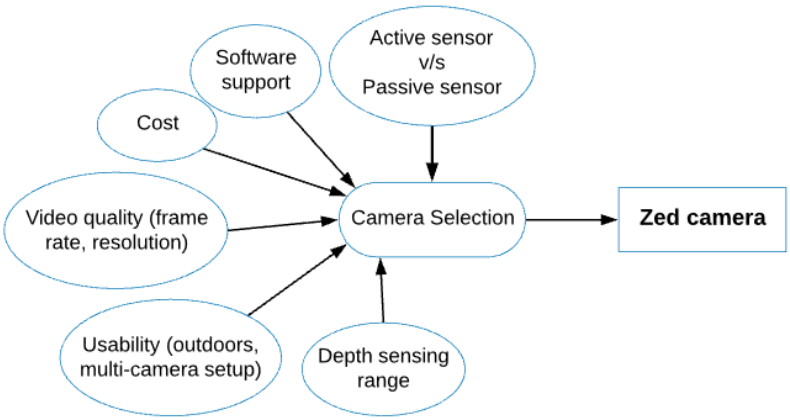


Figure 4.1: Features affecting stereo-camera selection

Based on the method of sensing depth by the camera, they can be divided into two categories, active sensor and passive sensor. Active sensors make use of a self-emitted structured energy (such as infrared radiations) which gets reflected by the surrounding objects and is detected by the sensor. This difference in time and intensity of the detected radiations is used to create a depth map of the surrounding [31]. Generally, these cameras have an additional RGB camera to capture visual images along with the depth map. Examples of such camera sensors are Intel's RealSense D400 series [25], Orbbec's Astra series [26] and Asus's Xtion series [27]. Generally, active depth cameras are computationally light and are considered ideal for indoor, short-range depth sensing applications. However, when faced with an outdoor environment, the infrared radiations from the sun interferes with the camera's detector making the observations unreliable [31]. On the other hand, passive sensors have two RGB cameras which capture the scene with two slightly different perspectives. This slight difference in perspective between two images of the same scene is used to create the depth map which is similar to human binocular vision [31]. Examples of such camera devices are Stereolab's Zed camera [28], Carnegie robotics's Multisense S7[30] and Flir's Bumblebee series[29]. These sensors can work in outdoor conditions even under direct sunlight [32]. As these cameras rely on visual information for their depth estimation, they are not suitable for low-light conditions.

Table 4.1 compares and scores the depth cameras based on their features as provided by the manufacturer on their official websites respectively. The number of plus signs represent a greater value of that feature. Minus sign represents that the feature is not provided in the camera. Whenever information on a feature was not provided by the manufacturer, the score of this feature is labelled as not available (NA) in the table 4.1. Cameras providing best overall features were Intel's Realsense camera and Stereolab's Zed camera. They provide very similar level of image resolution and software support. However, the Realsense camera had a lower depth sensing range and frame rate. Also, as the Realsense camera was a type of active sensor, it was mainly limited to indoor, short-range applications [31]. A study by Deris et. al. was about scanning historical structures using the Zed camera in outdoor environments [32]. They reported no problems faced while using the Zed camera even under direct sunlight. These set of better performance features in the Zed camera comes with a higher price tag.

Table 4.1: Comparing and scoring of different depth cameras

Camera model	Manufacturer	Sensor type	Features				
			Max. Range	Resolution	Frame rate	Software support	Price
Realsense D435i[25]	Intel	Active	++	+++	++	+++	++
Astra Pro[26]	Orbbec	Active	+	++	+	++	+
Xtion Pro[27]	Asus	Active	+	++	+	+	NA
Zed camera[28]	**Stereolabs**	**Passive**	**+++**	**+++**	**+++**	**+++**	**+++**
Multisense S7[30]	Carnegie Robotics	Passive	NA	+	+	+	NA
Bumblebee 2[29]	Flir	Passive	NA	++	+	-	NA

Considering the reliable outdoor performance, extensive depth sensing range (20m) with high frame rate and resolution (720p @ 60fps), Zed camera was selected to be used in this research which can be seen in figure 4.2. The manufacturer of this camera also provided software support to record, process and use the recorded data in different ways. This camera also had a large programming community second only to Intel's Realsense camera which can be very helpful when working with such new devices. Certain limitations to this camera were its huge data generation rate (250MB/s) and its poor performance in low light conditions. Such operational limitations were not known for other cameras as they were not investigated first-hand.

Figure 4.2: Zed camera by Stereolabs

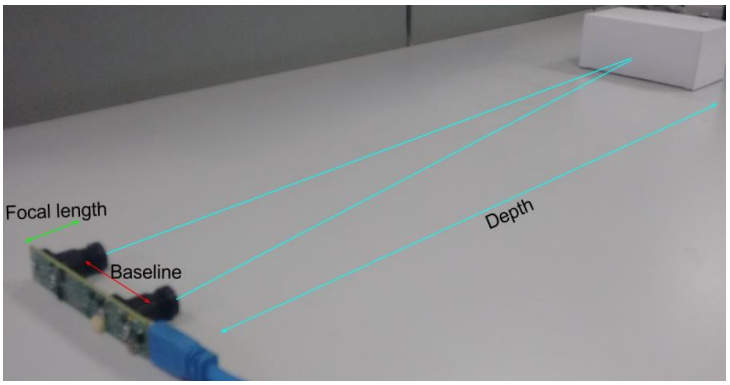


Figure 4.3: Working principle of stereo vision camera[3]

4.2 Working of Zed camera

The Zed camera is a stereo camera with two monocular lenses. It mimics the principles of human vision by using the binocular vision and extracting the depth information from the two images [33]. As shown in figure 4.3, the two monocular cameras are placed at a distance while observing the same scene at a slightly different angle. This slight shift in perspective between both the individual cameras is termed as disparity. The disparity is higher for objects nearer to the cameras compared to objects placed further away. Using this disparity and the principle of triangulation, the camera's generic software was able to calculate the depth of each pixel in the image. A detailed list of camera features and the camera's support hardware (mini-computer, data cables) used in this research during the data collection are provided in

appendix A.1.

4.3 Using the camera

In this research, the Zed camera was used at the resolution of 720p with the frame rate of 60fps. Referring to the study by Ortiz et. al., the above configuration was the optimum trade-off between the quality of collected data and its processing time [33]. This camera generates a massive amount of raw data while recording (0.9TB per hour). With the correct computer configurations (GPU memory>4GB), the generic software was able to compress the raw visual data into a 20 times smaller file size in real-time without any loss in image quality. This feature allowed the data collections over a more significant period without running out of storage memory.

The Zed camera recorded and stored the footage in a proprietary SVO format which can only be read using its generic software (ZED SDK). The software was able to correct the original image for distortion as can be seen when comparing figure 4.4a & 4.4b. Using the corrected visual images from both the lenses, the depth information for each pixel was calculated as represented in figure 4.4c. The generated depth maps were perfectly aligned with the visual images for every pixel. This means that the pixel coordinates from the visual images can directly be used to refer to its depth in the depth map without any modifications. The depth information was expressed as the XYZ coordinate value w.r.t the left lens of the camera also called as point-cloud data. This information can again be extracted using the pixel coordinates to represent that pixel in the 3D-space w.r.t the camera's coordinate axes.

4.4 Discussion

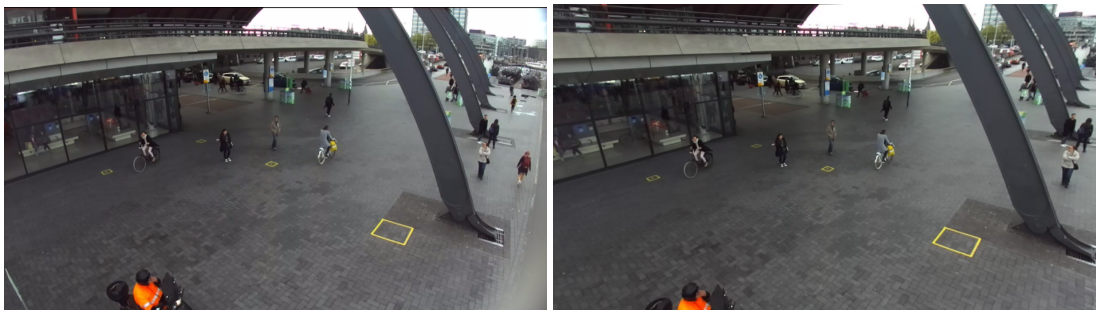
When exploring and working with such new technologies, the online programming communities can provide valuable support and thus was considered as one of the factors in the camera selection process.

As the Zed camera relies solely on visual information for its depth calculations, apart from just the image resolution and distance from the camera, the lighting conditions can also be a factor affecting the accuracy of depth estimation. Literature covering this aspect of depth estimation was unavailable and thus can be investigated in the future.

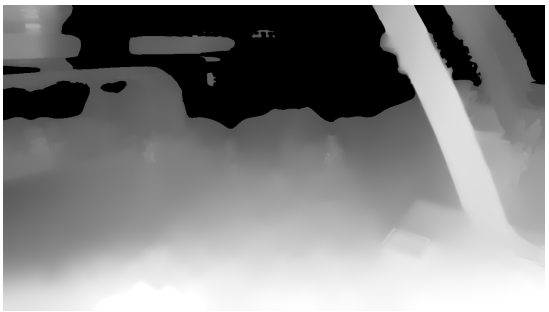
The supporting hardware with correct configurations was necessary to use the Zed camera at its full potential. The camera's generic software was freely available to be downloaded and used by anyone with the correct computer configuration. This allows the recorded data in this research to be easily shared amongst peers for future use. In future, further investigations in to the use of Zed camera in a multi-camera setup can also be explored.

4.5 Summary and Conclusion

The Zed camera was selected based on factors such as depth sensor's type, depth sensing range, video quality, software support and its market price. This camera was able to work in outdoor environments with an extensive depth sensing range, and record the data at good resolution and frame rate. However, it came with a heavy computational requirement and was unable to perform in low-light conditions. The camera's generic software was able to correct the raw images for distortion and provided with a perfectly aligned depth map with the visual images. The hardware setup used in this research enabled live compression of the raw video footage which increased the available time for data collection. Such features eased the handling and processing of data in later stages.



(a) Original image (distorted) (b) Corrected image



(c) Depth map

Figure 4.4: Representing the recorded image with lens distortion, corrected image and the depth maps provided by the camera's generic software

Chapter 5

Data Collections

For this study, there was a need to collect first-hand data which was later analysed to develop a methodology in extracting trajectories of agents. Before collecting the data, the process of identifying the factors to be tested and designing the experimental setup helped in developing a data collection framework for the real-world environments.

For this study, two data collection experiments were performed at two different locations. The first experiment was conducted in the Green Village area within the campus of TU Delft. This experiment was aimed at testing the camera setup in the real-world conditions and to use the collected data to develop the data processing framework. The second data collection was performed behind the Amsterdam Central station. This experiment was mainly aimed at testing the developed data processing framework on such real-world dataset. In the real-world environments, the traffic cameras are generally installed at an angle due to the lack of proper overhead camera installing locations. Thus, to replicate this situation, the camera in both the data collection experiments were installed with a certain angle of tilt.

This chapter provides a detailed explanation into both the data collection setups. At last, some of the sensitivity parameters for this data collection setup are also identified.

5.1 Green Village experiment

The first experiment was aimed at testing the hardware setup in real-world conditions and developing the data processing framework for trajectory extraction. The hardware setup was tested for its sensitivity to external factors (like heat, sunlight), internal factors (like data transfer and storage) and the reliability of supporting devices (like mini-computer, data cables, power supply). The movement patterns in this experiment were controlled to help in developing the data processing framework. For planning movements in the observation area, control variables were identified considering the challenges faced in people tracking from the literature. These control variables guided the design process of different scenarios as explained in the following paragraph.

Scenario Design

In the literature, one of the main challenges mentioned in case of people tracking was occlusion [38, 35, 61]. The reasons and the nature of occlusion in public spaces can be linked to many number of factors. In this controlled experiment, agent-to-agent occlusion was studied by varying the direction of approach and crowding levels in the interaction area. As this study focuses on two modes, namely pedestrians and cyclists in a shared space, this experiment includes both the modes.

Based on the three control variables namely, crowding, direction of approach and differentiation of modes, three scenarios were designed. Table 5.1 provides an overview of the scenarios. Based on the mode of travel, these scenarios can be divided into pedestrian only, mixed flows and cyclists only scenario. Each scenario was performed with varying level of crowding densities. The initial densities were low with 1-2 agents while gradually increasing the number of participants in the experimental space. This variation in the crowding levels helped in observing its effect on the software's performance while also helped in evenly distributing the agents throughout the experimental space. The three scenarios are as follws:

- Scenario 1 was a pedestrian only scenario with bi-directional flow of agents as seen in figure 5.1a. The movement of agents was horizontal w.r.t. the camera. This scenario was aimed at testing the sensitivity of occlusion as a result of crowding to the software's people detection and tracking abilities. The participants entered the space in phases to maintain the observed crowding levels. The first phase starts with two pedestrians from either side of the waiting area which builds up to the last phase with all the 16 pedestrians.
- Scenario 2 had mixed traffic consisting of both pedestrians and cyclists in the interaction area. This scenario was primarily aimed at testing the abilities of the software to categorise different modes of transport. The detailed layout can be seen in figure 5.1b. The total of 16 participants were divided into 6 cyclists and 10 pedestrians. The route was predefined for both cyclists pedestrians as shown in the figure. Similar to the previous scenario, this scenario also had participants entering the experiment one-by-one in phases.
- Scenario 3 was a cyclists only scenario with eight cyclists travelling in a bidirectional flow. The layout and route directions for this scenario were similar to that of scenario 2 but without pedestrians. The aim here was to study the detection and tracking of cyclists independent of pedestrians.

Table 5.1: Overview of different scenarios in Green Village experiment

Scenario	Modes	Flow	Participants
1	Pedestrian only	Bi-directional flows	16 pedestrians
2	Mixed flows	Crossing flows	6 cyclists, 10 pedestrians
3	Cyclists only	Bi-directional flows	6 cyclists

Initially, while selecting the control variables and designing different scenarios, much more control variables were identified and the scenarios were performed than those described above. These data were later discarded and not included further into this research. An elaborate reasoning to discard these data and the lessons learned are provided in the discussion section of this chapter (section 5.3). Each of the above scenario was also performed two times, one with participants wearing red caps and one without red caps. This data was an additional contribution of this project as it can be utilised later to derive agent trajectories using Moving object detection and tracking tool [62] and compare it with the proposed framework in this research. Red-cap plays no role and has no effect on the data extraction process developed in this research.

Data Collection

This controlled experiment was performed in the TUDelft's Green village area on Wednesday, 26^{th} June 2019 from 14:00 until 16:00. The experimental layout can be seen in figure 5.2. Considering the availability of space and the camera's viewing area, the area of interaction was decided to be 4m x 6m. The Zed camera was installed at a height of 6 meter from the ground and at a distance of 4 meter from the start of the interaction area. Traffic cones were placed around the interaction area to make it visually identifiable.

In total, 16 volunteers took part in this experiment. The participants were informed about the aim of this experiment, the risks involved and the usage of the collected data in future. The raw video of 30 minutes was recorded at 720p, 60fps (108,000 frames).

Observations

On the day of data collection, bright sunlight and high temperatures (32 °C) were observed. This challenged the camera's hardware as it was heating under direct sunlight which resulted in unstable connections. As seen in figure 5.3, the camera also recorded shadows of the participants which were casted towards the camera. This posed a challenge during the agent detection process while determining their location as explained later in this report. The effort to distribute the crowds by allowing them to enter in phases was effective for single-mode scenarios (Scenario 1 & 3) as seen in figure 5.3a & 5.3c but not for mixed flows. Figure 5.3b shows that in scenario 2 a frequent grouping was observed due to the need for waiting before crossing the interaction area between pedestrians and cyclists.

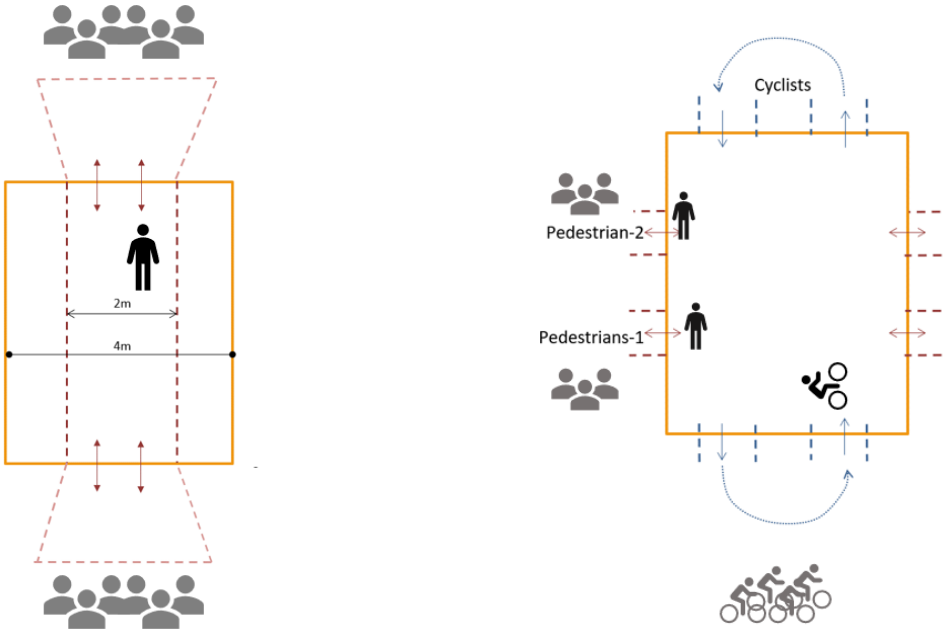
Reflection

During the Green Village experiment, the Zed camera along with the mini-computer were unable to withstand high temperatures under direct sunlight. However, the sunlight in the interaction area did not impact the camera's ability to calculate its depth information. The GPU-enabled computer was able to compress the raw data from the camera in real-time without any loss in image quality.

Before conducting the experiment, recruiting volunteers, acquiring permissions from the Green Village authorities and planning for scenarios were the main tasks. Other than personal connections of friends and colleagues, the timing of the experiment plays a crucial role while recruiting volunteers. While designing the experiment, emphasis on including and testing many control variables lead to over-engineering the scenarios. When such controlled

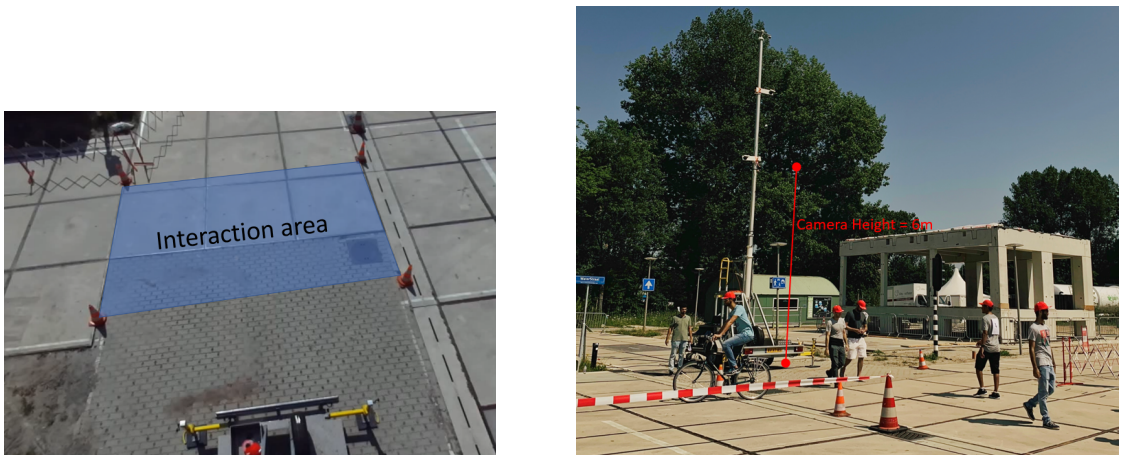
experiments are preformed with a limited purpose (in this case, to test the camera's working and developing the data processing framework), much simpler and easy to execute scenarios can be designed.

While conducting this experiment, the time of preparation of 1.5 hours was observed to be insufficient as it required preparing the site and briefing the participants about the experiment. It was easy to explain the different scenarios and movement patterns to participants with graphical images and on-site demonstrations. Planning the experiment in sub-phases further helped the participants to watch the initial participants and then follow them with a similar movement pattern.



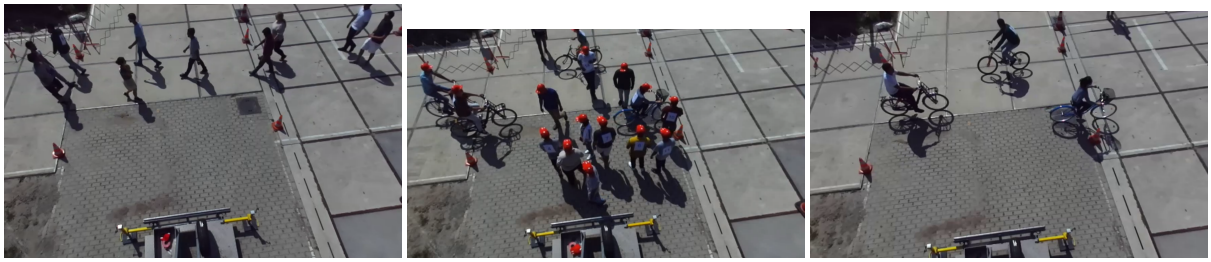
(a) Scenario 1 - Pedestrian only (b) Scenario 2 - Mixed flow

Figure 5.1: Layout of the interaction area in each scenario



(a) Camera's view (b) On-site view

Figure 5.2: Pictures from the Green Village experiment



(a) Scenario 1 (Pedestrian only) (b) Scenario 2 (Mixed Flows) (c) Scenario 3 (Cyclists Only)

Figure 5.3: Snapshots of visual recordings from Green Village experiment

5.2 Amsterdam Experiment

The second data collection was aimed at capturing real-world interactions in a shared space environment. This data was later used to test the developed data processing framework(in chapter 9). Out of all the shared-spaces consisting of pedestrians and cyclists, the area behind Amsterdam Central station was used for data collection. This location was selected based on three main criteria: (1) This space was majorly shared between pedestrians and cyclists with a few motorbikes but no vehicular traffic (cars, buses). (2) The flow of traffic for both the modes was multi-directional. (3) This space was used by numerous people daily allowing to test the setup for high-density situations.

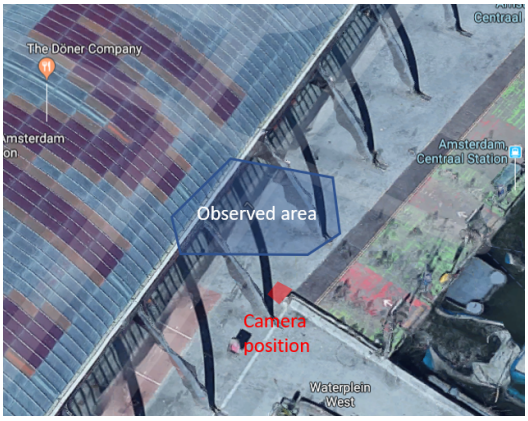
The data collection was performed on Thursday, 17^{th} October 2019 from 14:00 until 17:00. Figure 5.4 shows the on-site setup, camera's location and the camera's view of the observed area. The camera's pole was positioned near the ferry terminal behind Amsterdam Central station. The height of the camera was 5.26 meters from the ground. Three reference points (A, B & C) were marked on-ground as seen in figure 5.4c. Relative distances between the reference points and, from the camera pole's footing were measured. These measured distance were later compared with the distances estimated by the Zed camera to asses the camera's accuracy. The on-site measurements of these reference points are given in table 5.2 where camera's footing is represented as 'O'. In total, approximately 29GB of raw footage (after the live compression) was recorded during two hours of data collection.

When comparing this experiment with the controlled experiment, the real world conditions were observed to be much more dynamic in terms of direction/speed of movement, physical appearances and the accessories people were carrying (such as suitcases, dogs, bags, hats, etc.). People walking and cycling through the space in groups of two or more were also observed during this experiment. Due to the absence of direct sunlight, agents were forming no shadows on the ground surface. This data can be divided into two main scenarios, normal conditions and time of ferry arrival. Higher crowding levels were observed when the ferries docked and its passengers started egressing through the shared space area. Its passengers consisted of both pedestrians and cyclists. While during normal conditions (i.e. without the ferry passengers), the crowding levels were notably lower. Regarding the hardware,camera and all its supporting hardware worked smoothly without any issues.

Reflection

Learning from the shortfalls of the Green Village experiment, multiple on-site visits and intensive hardware testing were conducted before the Amsterdam experiment. Also, the day of Amsterdam experiment was much cooler than the Green Village experiment. Such extensive preparation prior to the actual experiment helped in performing the on-site experiment smoothly without any issues. All the formalities and permissions from the Municipality of Amsterdam were also secured beforehand.

(a) On-site setup



(b) Site location

(c) Camera's view & reference points

Figure 5.4: Pictures from the data collection in Amsterdam

Table 5.2: On-site measurements of reference points

Sides	Distance(m)
OA	14.0
OB	16.6
OC	12.9
AB	5.64
BC	5.15

5.3 Discussion

In addition to the three scenarios explained in the Green Village experiment, two more scenarios with multi-directional flow of pedestrians and a camera testing scenario was planned. The data for multi-directional scenario was recorded but not used as the waiting areas during the experiment were within the recorded frame which made the data unusable for analysis(see figure 5.5). This made the data unfit for further analysis. The camera testing scenario was not performed as the hardware setup for the camera installed at different orientations failed and did not record any data. The shortfalls in the experimental setup were identified (issue with connecting cables) which were then rectified before the Amsterdam experiment. The data for all scenarios during the Green Village experiment was also recorded using a monocular camera at the height of 10 meters. This is an additional contribution of this project as this data can be used later to extract agent trajectories using the existing tools [62] and compare it with the proposed methodology in this research.

Compared to the controlled experiment, the Amsterdam experiment was much hassle-free as it involved only setting up the hardware and there was nothing to control over the agent's movement patters. Extensive preparation beforehand and multiple equipment testing ensured a successful real-world experiment. Overall, this setup needed an intensive on-site support regarding power supply and GPU enabled computer for video storage. Thus, further research in building a lighter, more reliable data collection setup while using the Zed camera is needed. This should provide an easy on-site installation and expand the 3D-data collection possibilities using the Zed camera.

Figure 5.5: Snapshot for multi-directional scenario showing waiting areas within the frame

Chapter 6

Agent Detection

One of the research objectives of this research was to identify and localise the agents in the observed area. In this agent detection stage, the visual images from the Zed camera was used to fulfil the research objective. Only after obtaining the location of agents on the image, the 3D-data from the Zed camera can be used to locate the detected agent on the ground plane.

As the aim was to automate the data extraction process, the first section of this chapter selects a neural network based detection model capable to detect agents in an image. This section uses both literature and first-hand analysis to select the best detection model for this study. Following this, a brief explanation into the working of selected object detection model is mentioned. Lastly, the implementation process and obtained results from the detection model are discussed in detail. This section also explores the possibility to automatically distinguish between two modes of travel (pedestrians and cyclists) using the detection models based on the visual information only.

6.1 Model selection

Selecting the right detection model depends upon the nature of application which can vary for each user. In the literature, two main criteria for comparing these models are the model's video processing speed and the accuracy of output detections. In this research, the visual videos were to be processed offline. Moreover, the performance of all the later processes was also dependant upon the quality of detections extracted in this stage. Thus, the detection model's accuracy was decided to be the main criteria of selection for this project rather than the processing speed.

This section elaborates on the selection process of the object detection model which was applied in this research. Initially, the comparison is done referring to two literature studies. Due to the scarcity of such comparative studies, a brief first-hand analysis of different detection models was performed on the data from Green Village experiment. Figure 6.1 provides an overview of this selection process and the criteria of model selection.

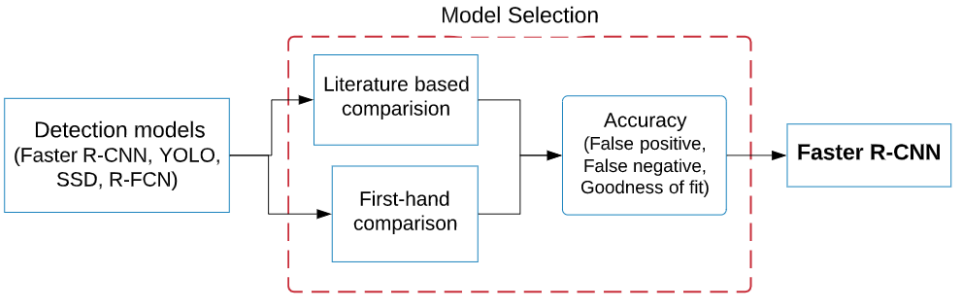


Figure 6.1: Process of selecting the detection model

6.1.1 Literature based comparison

In year 2017, a study published by Google which extensively compares the performance of three different object detections algorithms namely SSD, R-FCN and Faster R-CNN under controlled parameters [49]. This study used the model's architecture as was originally proposed in their respective papers. The same data used for training and testing the models which created a level playing field for further comparison. These comparisons between models were done based on speed and accuracy of detection. Speed was expressed as the processing speed of the input video in frames per millisecond (or GPU time). The accuracy was determined based on the mean average precision(mAP) which includes type-I and type-II errors i.e. the number of false-positive and false-negative detections respectively. Figure 6.2a represents these results for each detection model. This study concluded that SSD and R-FCN were significantly faster than Faster R-CNN (approx. 3 times faster) while Faster R-CNN provided the highest detection accuracy.

Another online article by Jonathan Hui compares the performance of SSD, YOLO, R-FCN and Faster R-CNN by aggregating and comparing the results as were reported in their respective release papers [55]. As for speed, YOLO and SSD were reported to perform much faster than the Faster R-CNN algorithm. While for accuracy, Faster R-CNN outperforms YOLO and SSD by a slight margin. These conclusions were subject to different run-time parameters both during the training phase and the processing of visual video for output.

6.1.2 First-hand comparison

Due to the scarcity of such comparative studies, a first-hand comparison of pre-trained SSD, YOLOv3 and Faster R-CNN models was performed. The accuracy of the detection boxes can assessed based on three criteria, number of false positives and false negatives detections and, the goodness of fit of the detection box to its agent. False positive detections are the detections which are considered valid by the model but are not truly valid. False negatives are the missed detections by the model. The goodness of fit refers to the ability of the detection model to draw a bounding box around the agent with minimum background. Pre-trained model files were downloaded from Tensorflow model zoo [53] and YOLO's official website [63] and were integrated into the video processing python code. This code was inspired and build upon the original code provided by Stereolabs [64, 65]. A minute of video consisting of 3600 frames from each of the three scenarios (performed during the Green Village experiment) was extracted as sample dataset for this comparison.

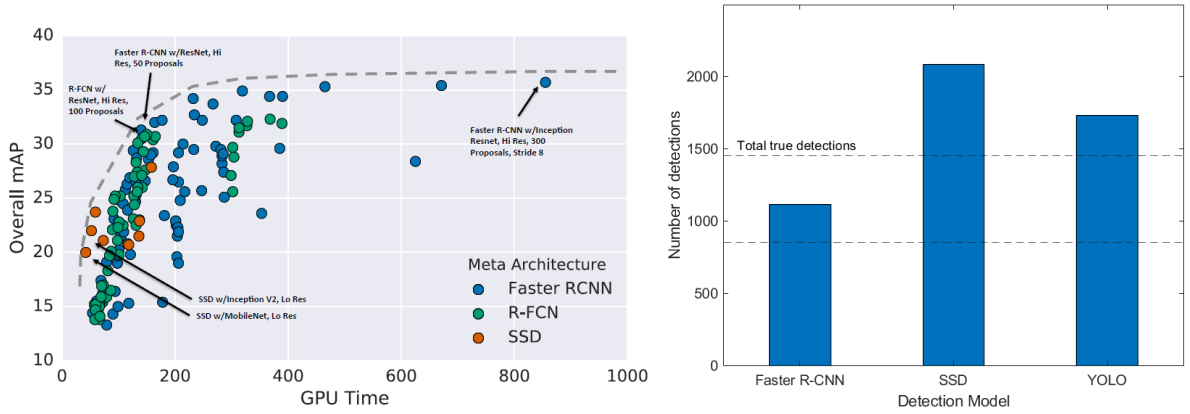
Some snapshots of the output provided by the three models is represented in figure 6.3. Based on the qualitative assessment of the detection boxes, the table 6.1 scores these models based on the accuracy of detections and the speed of processing the video. In this table, the number of plus signs compare the relative performances of these models with higher the number of plus signs, better is the models performance for that criteria. With the given computer configurations, SSD and YOLO processed the visual video at a similar speed of about 15 to 20 fps while Faster R-CNN was much slower at about 0.5fps. From the figure 6.3, the accuracy of different models can be assess qualitatively. Both SSD and YOLO models frequently categorised people's shadows and traffic cones as a person too. They also missed many valid detections or provided one detection for multiple people standing close by. All this lead to a large number of false-positives and false-negatives by these two models. Dominant shadows proved to be a challenge in determining the exact boundary of the participants these models. The visual comparison of the detections from all the three models showed that Faster R-CNN provided bounding boxes with a better fit than the other two models. i.e it was better in excluding the background from the object of interest. Whereas both SSD and YOLO frequently included shadows and the background into the bounding boxes.

To quantify some of these results, a 10 second video (600 frames) of bidirectional pedestrian scenario with an average of 2.4 people in every frame was taken. This dataset was selected for such comparison as it was one of least challenging dataset recorded during the Green Village experiment. The total number of true detections for these frames were manually counted to be 1450 out of which 850 detections were excluding the directly overhead detections. The results of total number of detections provided by the three models on this sample dataset are represented in figure 6.2. It was observed that all three models faced difficulty in detecting the agents from a directly overhead position. Thus, the Faster R-CNN falls short of the 1450 detection mark in the figure 6.2. But, Faster R-CNN largely detected agents with minimum false positive and false negative detections. SSD model had the highest number of detection due to many false positive detections and was the worst performing model out of the three models. YOLO was a much better model and provided a good number of true positive detections. Even though SSD and YOLO can provide same or higher number of detections, the accuracy of detection boxes is unmatched to those provided by Faster R-CNN.

Faster R-CNN's superiority in excluding the background from the bounding box should help in better depth estimation of the agent while detections with lower errors should provide stable trajectories. Thus, Faster R-CNN's pre-trained model was chosen over the other two models to perform detections and extract bounding boxes from the recorded data.

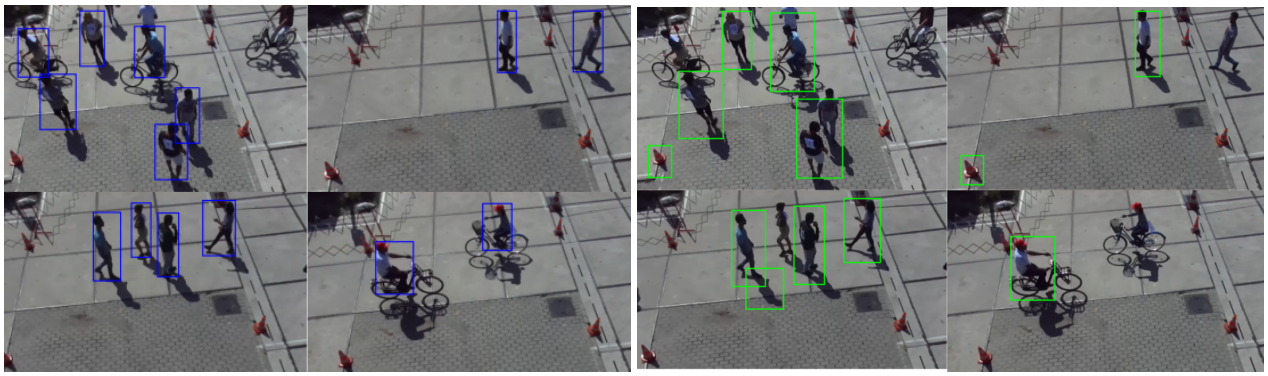
Table 6.1: Qualitative assessment of observed speed and accuracy levels for detection models

Detection Models	**Accuracy of detection boxes**	**Processing Speed (frames per second)**
Faster R-CNN	++++	+
SSD	+	++++
YOLO	++	++++

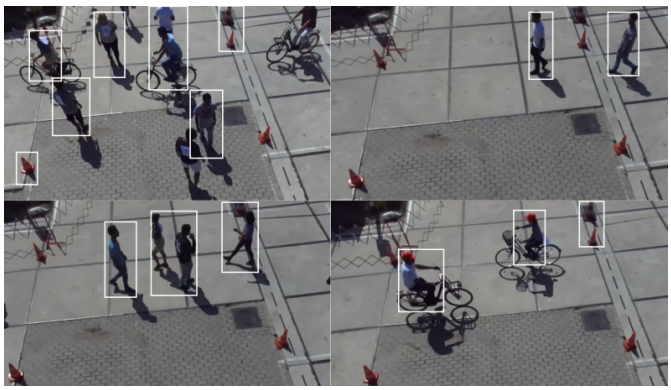


(a) Accuracy vs processing time [49] (b) First-hand comparison

Figure 6.2: Accuracy of detections for different detection models



(a) Faster R-CNN (b) SSD



(c) YOLO

Figure 6.3: Bounding boxes obtained using different detection algorithms

6.2 Working of Faster R-CNN

A computer interprets an image as a collection of tiny pixels with each pixel storing some information of the image. This information expresses the colour of that pixel and is encoded in the RGB format. RGB format represents the intensity of red, green and blue colour respectively in the pixel as seen in figure 6.4.

For image processing, the type of neural network used are called as convolutional neural networks(CNNs). A neural network is a collection of neurons with each neuron performing a dedicated function. All these neurons are arranged in the form of layers as shown in figure 6.5. The first layer takes the input as the image which is then processed by the neurons in the hidden layers. In case of Faster R-CNN, these hidden layers can be divided into three categories based on its operation: feature extraction network, region proposals network and detection network. An overview of these networks can be seen in figure 6.6. The first stage of feature extraction, a raw image undergoes various matrix operations to provide a feature map. These feature maps retains all the spatial and structural information of an image and filters out the irrelevant information. The second stage proposes a number of regions on the feature map to look for an object. These regions are then analysed in the last stage to classify the regions based on its object type. To obtain a detailed understanding into the working of Faster-RCNN's object detection framework, following literature [52][1] [66] [67], Stanford university's online lectures [68] and some online articles [54] [69] [70] can be referred.

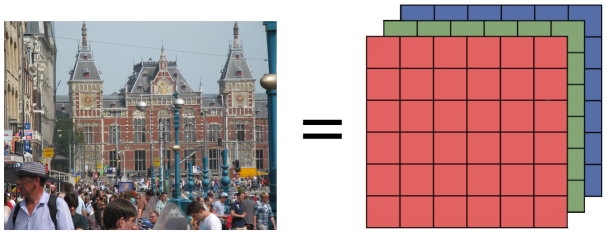


Figure 6.4: Image as RGB matrix

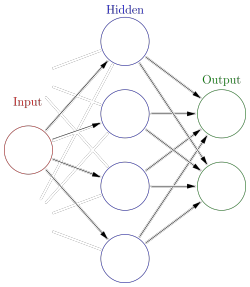


Figure 6.5: Neural networks

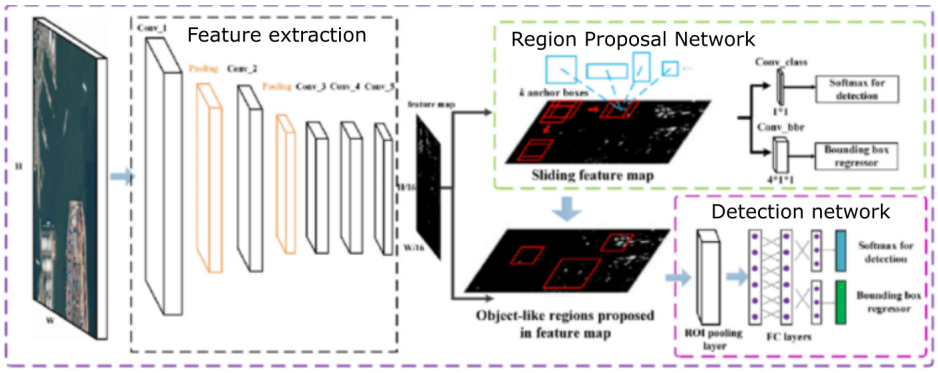


Figure 6.6: Network architecture of Faster R-CNN [4]

6.3 Using the detection model

Faster R-CNN detection model used is already pre-trained to detect 90 different object classes [53]. The original python code was modified to customarily set the object class as need. As observed during the model selection process (section 6.1), the Faster R-CNN model was able to detect people which included both pedestrians and cyclists. As this chapter also wanted to investigate the possibility to differentiate the two modes, the Faster R-CNN model was also used to detect bicycles (and not the cyclists or the rider itself) which is a different object class. The outputs were filtered based on the detection's confidence score. A higher confidence threshold provided more accurate detections and less number of false-positive detections. But this also reduced the total number of detection and increased the false-negative error. Figure 6.8 represents the distribution of total number of detection at different confidence scores. This distribution was obtained for person only detection on a mixed-flow scenario dataset recorded during the Green Village experiment. The nature of this distribution suggests that this detection model provided the majority of its detection at a high confidence score (approx. 75% of the detections were with scores > 0.95). A similar trend was observed across all the three scenarios and for both detections types (people and bicycles). This suggests that in case of Faster R-CNN, the detections obtained were not much sensitive to the confidence threshold value. Based on the best practice as reported in the literature, the threshold value of 0.75 was used in this research [49][1].

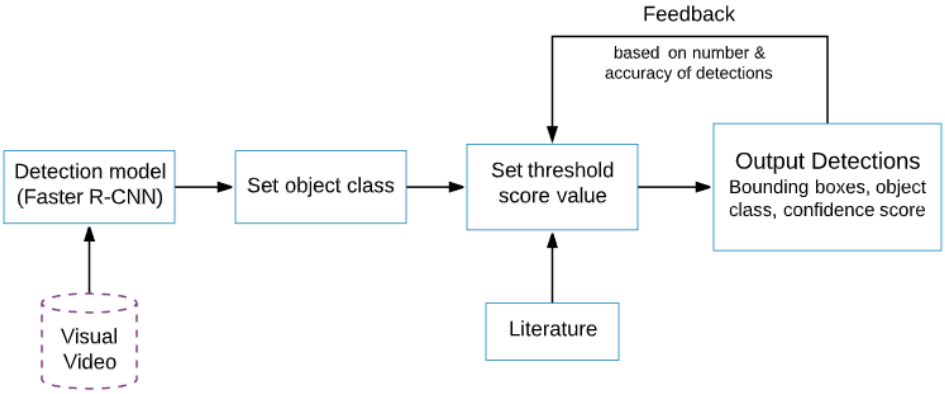


Figure 6.7: Framework for using the detection model

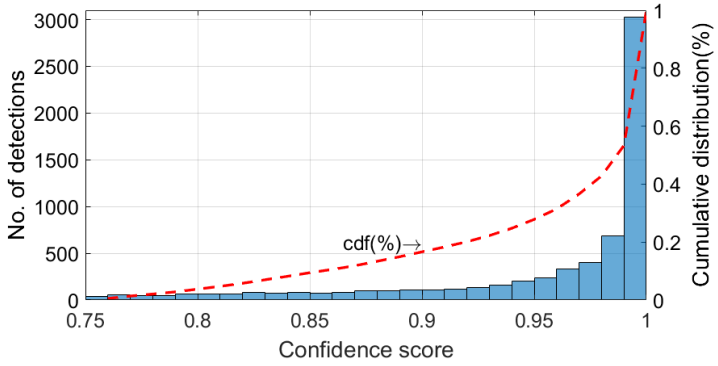


Figure 6.8: Number of detections with each confidence score

6.4 Results & Analysis

The output obtained from this process was in the form of bounding box coordinates w.r.t the visual image. The detection model was tested to detect both bicycles and people (pedestrians and cyclists) on the Green Village data individually as seen in figure 6.9. As observed, the detection model faced some difficulty in distinguishing between the shadows and the person. But the model performed well in identifying the other background from the true agent and provided boxes with a good fit. Major challenges were faced by the detection model in cases of partial occlusion between agents. As seen in figure 6.9a, the detection model labelled two different agents as a single agent by bounding them in a single box.

The shape, size and orientation of the agents were constantly changing as they moved from the space. To bound these moving agents, the detection boxes were also constantly resizing. For pedestrians moving horizontally across the image, this change in box sizes was due to the swaying of hands and feet while walking. For pedestrians walking vertically in the image, this change was observed as due to their changing size as they change their distance from the camera. For cyclists, this change was observed as they perform a paddling motion. The orientation of agents within the boxes also change as they move from centre of the image to its sides. No change in such orientation was observed when they move from top to bottom of the image. For bicycles, the detections were stable as the change of shape and size for bicycles was very limit. But here, the model was challenged while determining the bicycle's true boundaries as it was difficult to exactly identify where the bicycle ends and the rider starts.

(a) People detection & occlusion

(b) People detection in mixed traffic

(c) Bicycle detections

Figure 6.9: Detection three scenarios of the Green Village experiment

6.5 Discussion

In this study, the first-hand comparisons were done using pre-trained models. Thus, it does not cover the effects of varying the design choices and the quality of training data set provided to the untrained model. There can be many possible design choices during the training phases of the model which largely impacts the performance of the detection model. This high degree of freedom during the training phase makes the true comparison between different detections models difficult. All the neural network based detection model can always be re-trained to better fit a custom dataset and provide better detections on a given data. This is an advantage of such detection models as they can constantly be improved as long as their is representative and labelled dataset available.

Here, the number and quality of detections were not much sensitive to the confidence threshold value due to the model's high accuracy. Whereas, this value should be selected wisely for other models which have lower accuracy levels(SSD, YOLO). In future, analysis into the relationship between the detection accuracy and the characteristics of input images in terms of its aspect ratio, resolution, brightness, contrast and other image properties can be studied further.

6.6 Summary and Conclusion

The detection model was responsible to identify, categorise and locate the agents on the visual image. Different detection models were compared using literature study and first-hand analysis based on the accuracy of the detections. Out of all the models, Faster R-CNN performed the best in terms of accuracy and thus was selected to be used in this research. The detection model was successfully able to distinguish between people and bicycles. In case of identifying people, both pedestrians and bicycle riders were included. The detections were provided in the form of bounding boxes around each detected agent. The output results were stored in the form of image coordinates of these bounding boxes for every image in the video. Certain anomalies were observed in making these boxes around the agent due to shadows and occlusion. A few instances of missed detections were also observed across all the three scenarios. In the next chapter, these detection boxes were used to refer to the depth information obtained using the Zed camera.

Chapter 7

Depth extraction with Ground representation

To obtain the trajectories on the ground plane, the agents first need to be projected on the ground for every time-step (or every frame of the video). The previous process of agent detection localised the agents on the visual images in every frame of the video recorded from a tilted camera position.

This chapter elaborates on the process of using the detections from the previous process, combining it with the depth map calculated by the camera and then representing the detected agents on the ground plane. This data processing framework can be divided into two sub-processes as shown in figure 7.1. First process was to extract the depth information of the detected agent to estimate its 3D-location w.r.t the camera axes. Afterwards, using the camera's angle of tilt, this 3D-point of the detected agent was projected on the 2D ground plane. The first section provides some initial calculation which were used to inform the design processes of depth extraction and ground plane representation. The later sections cover each process individually followed by the results and discussion of the developed framework.

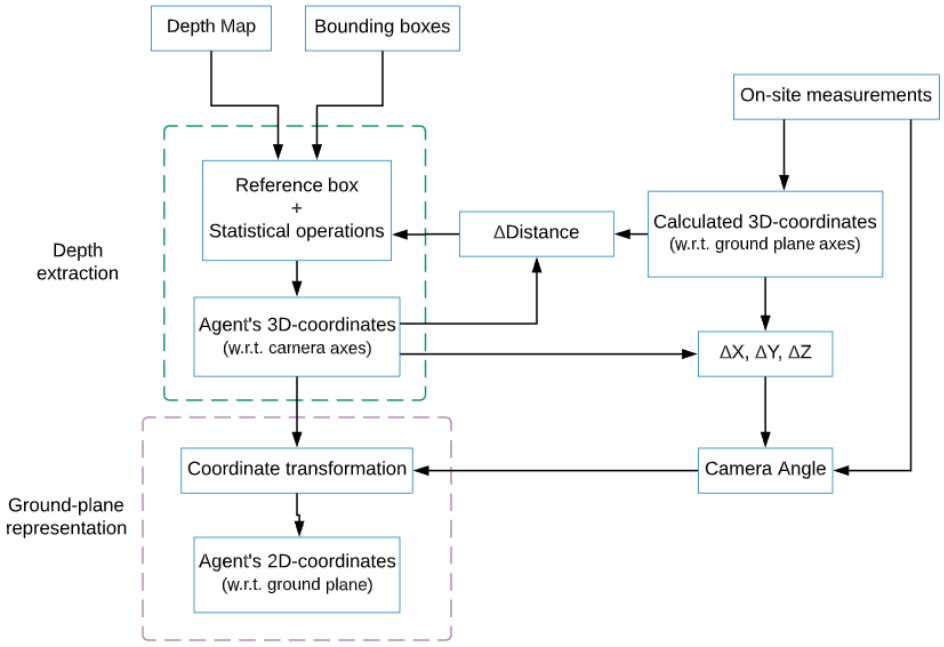


Figure 7.1: Framework for depth extraction and ground representation of agent in Green Village data

7.1 Initial calculations

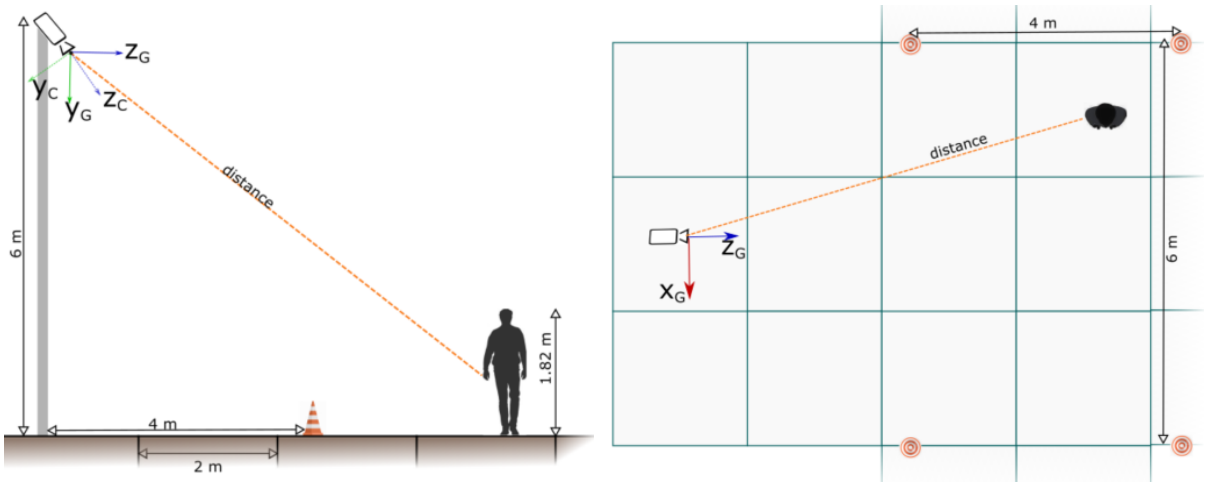
To design and tune the parameters values in the later sections, their should be some ground-truth data which can be used as a reference for comparison. This section calculates the ground truth coordinates of an agent w.r.t the ground plane axes keeping the camera as its origin. For these calculations, this section relies on the on-site measurements (as explained in section 5.1) and visual inspection of the selected pedestrian agent from 10 frames. The process of selecting the frames and the calculation of 3D-coordinates is explained further in this section. These calculated coordinates are treated as the ground truth data and were used in two ways. Firstly, to decide on the statistical operations and the size of the reference box while 3D point estimation of the detected agent. Secondly, to determine the camera's angle of tilt for coordinate transformation and ground plane representation of trajectories.

From the Green Village videos, an agent was selected and its 3D-coordinates comprises of X, Y, Z-coordinates w.r.t the ground plane axes were calculated. The foundation of these calculations were the on-site measurements of camera height, traffic cone distances and the grid structure of the ground plane as can be seen in figure 7.2. In this figure, the x_G, y_G, z_G represent the axes along which calculations of coordinate values were made. The origin of this coordinate system was on the 3D camera's left lens itself with the $X_g - Z_g$ plane parallel to the ground plane. The x_G-axis was laterally aligned, z_G-axis pointed away from the camera while the y_G-axis was downward perpendicular to the ground. For simplicity, all the movement on the ground plane was assumed to have change only on the X_g and Z_g coordinates while Y_g coordinate representing the height component of agent remains unchanged.

For calculating the X_g and Z_g coordinates, measurements done along the ground plane (traffic cone distances, grid tiles on-ground) were used. As these measurements were done by people, they do introduce human error in the further calculations. The X_g-coordinate value

was the lateral distance between the camera and the agent while the Z_g-coordinate was given by the longitudinal distance (or depth) of the agent from the camera. The Y_g-coordinates represent the vertical distance (or height) of the camera from the agent's centre when measured perpendicular to the ground plane. This was calculated by subtracting half the height of the agent from the total height of the camera. This remains unchanged for the selected agent as no movement was along the y_G-axis was assumed.

The 10 frames of the selected agent were equally spaced over the agent's entire motion of 10 seconds (or 600 frames) as it enters from left edge of the image until it exits from image's right edge. This means that one sample frame was selected at every second of the ten second video. This helped to spread the calculated coordinates across the interaction area while including a variety of possible object orientation in the image. Certainly, 10 frames was a very limited sample size but by spreading the frames across the agent's motion and including different agent orientations, this study partially accounts for this limitation. Three image samples out of the ten images are represented in figure 7.3 representing the left most, centre and the right most position of agent respectively. It can be seen that the orientation of agent was slightly tilted w.r.t its position in the image. This orientation is only dependant on the agent's alignment w.r.t the right and left edge of the image and is not affected by the top and bottom edges. Calculated coordinates for the selected agent in all the 10 frames are provided in the appendix table A.2 for reference.



(a) Elevated view (b) Top View

Figure 7.2: Experimental setup and 3D coordinate axis for Green Village Experiment

(a) Agent near the left edge (b) Agent in the centre (c) Agent near the right edge

Figure 7.3: Calculated coordinates of agent for each frame (in meters)

7.2 Depth extraction

The output of the previous detection process provided the bounding box coordinates of the detected agent on a visual image. These box coordinated can be combined with the depth map calculated by the camera to extract depth information of the agent. The depth information was in the form of point cloud data i.e. the X, Y, Z coordinate values (in meters) w.r.t the camera axes. This depth information was calculated by the camera for each pixel in the visual image which can be referred to as a depth map. This depth map was perfectly aligned with the visual images from the camera's left lens as explained in section 4.3.

Initially, there were two possible approaches towards extracting the depth value of the agent. First approach can be to calculate the centre pixel of the bounding box and directly refer to its depth value. This approach relied only on one pixel for its depth value and hence could provide unstable and unreliable results. Moreover, it is also possible that for agents near the edge of the image, the centre point does not lie on the agent itself but in the agent's background. Second approach can be to analyse the depth information of all the pixels in the box and then extract the relevant depth value. This approach introduced a need to apply some statistical operations over depth values from all the pixels in the bounding box. Deciding these statistical operations can be tricky in cases when the box contains pixels of the far-way background, the agent itself and the ground beneath the agent. Especially in case of Green Village data, the ground beneath the agent was frequently included in the detection boxes due to the agent's shadow.

A third possible approach was to calculate the centre of the bounding box and then make a reference box around this centre point to extract the depth information. This will overcome the uncertainty caused by using only one pixel point while also make the selection process of a statistical operation more concrete. This approach also ensures that in most cases, the majority of pixels inside the reference box will belong to the detected agent. Thus, this third approach was taken towards extracting depth information. The distance between the sides of the reference box and the centre point was expressed in the number of pixels and was named as reference distance(R).

The first subsection decides on the statistical operations to be applied on the extracted distribution of X, Y & Z coordinates from the pixels in the reference box. This was to represent the detected agent as a single 3D point in space. The second subsection analyses and selects an optimum size of the reference box to be used. Both of these decisions were informed by the calculations made in the prevision section 7.1 as explained further.

7.2.1 Statistical operation

The depth information extracted from the SVO file was in the form of point cloud data i.e. the X, Y, Z coordinate values (in meters) w.r.t to the left lens of the camera. The reference box refers to a distribution of X, Y & Z coordinates which needed to be analysed to output the true value of the detected agent. Thus, a statistical approach was taken towards extracting the true depth value of an agent. This also helped to filter the values given by the background pixels if it was included in the reference box.

The 3D coordinate distributions w.r.t. the Zed camera were extracted for the same 10 frames as used in the calculations in section 7.1. Out of these ten frames, a sample of three images are shown in figure 7.4 with its X, Y & Z distribution in figures 7.5, 7.6, 7.7 respectively. For this

analysis, the reference box of size R = 20 was taken represented with a solid red line in figure 7.4 containing 1600 pixels in each reference box. The later subsection will focus on the reference box's size optimisation.

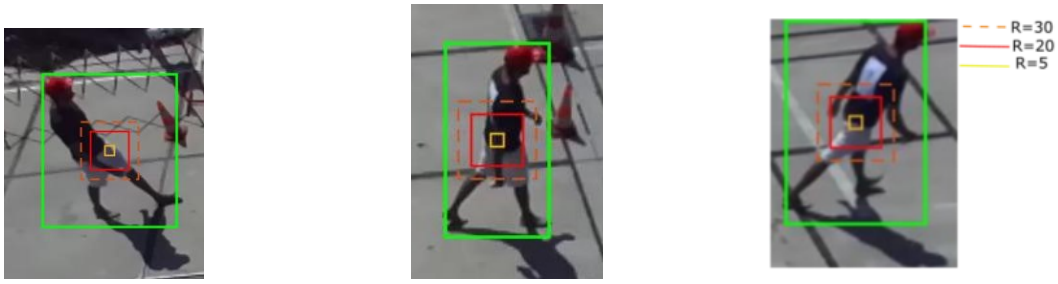
Four statistical operations were investigated to refer to one value from the extracted distribution of depth data using the reference box and the best option was decided. The first choice was to select the value with highest frequency which is the histogram's maximum value of the distribution. This would have filtered all the 3D data from the background pixels only when the majority of pixels inside the reference box were of the agent. This assumption might not be true in cases when the majority of the pixels belong to the background and not to the agent. Also, the calculation of maximum value in the histogram for each 3D data set over all the frames was to be done in parallel to the running of detection algorithm on the SVO file. This was observed to be computationally expensive and slowed the data extraction considerably. This operational challenge and the uncertainty of maximum histogram value representing the true value of the agent made this approach not practical. In future, further investigations in programming this approach more efficiently can be done.

Other statistical operations considered were Lower quartile(Q_1), Median(Q_2) and Upper quartile(Q_3) values of the distribution. Observing the coordinate system and the nature of data distribution, statistical operation were determined for each X, Y, Z value of the 3D data respectively. The aim of these statistical operations was to give values which were closer to the true maximum of the histogram i.e values derived from the agent's pixel. Also, the calculations of quartile values were significantly cheap on computation without loosing much of accuracy given by the maximum histogram approach as seen in table 7.1.

For X distribution, figure 7.5a & 7.5c show that the edge frames had a skewed distribution towards a clear peak value. Whereas, when the agent was in the centre of the image, the distributions were more plateaued with smaller peaks as in figure 7.5b. Also, the peak of the distribution shifts from right to left as the agent moves from left to right in the video. This is due to the fact that X=0 plane passes from the centre of the image. Thus, to account for this change in the nature of distribution, Median value (Q_2) of the X-distribution was used to extract the X_c coordinate of the agent. This value is represented by the red line in figure 7.5.

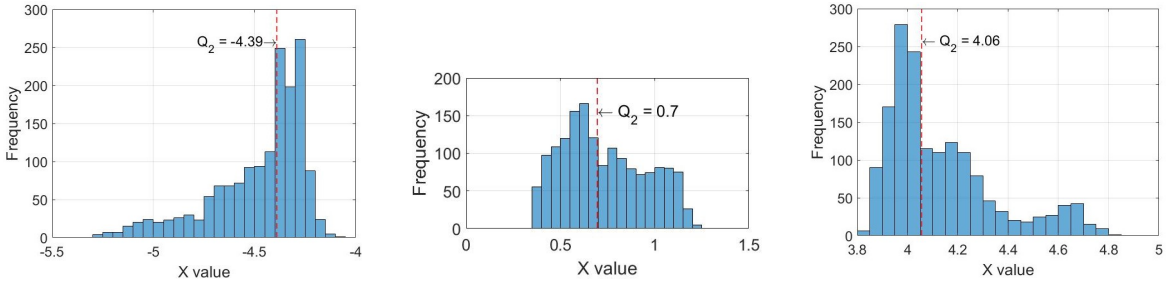
For Y value, the distributions were partially bi-modal in nature i.e. two peaks were observed in their distribution with the latter peak being highly dominant as seen in figure 7.6. When measured along the y_C-axis, the agent's centre was nearer to the camera than the background. Thus, the former peak (or blip) represented the background pixels while the latter peak represented the agent pixels. To obtain values nearer to the latter peak, upper quartile (Q_3) value of the distribution was used to represent the Y-coordinate of the agent.

For Z value also, the distributions are bi-modal in nature. When measured along the z_G-axis, the agent was nearer to the camera that the background. Thus, in figure 7.7, the former pixels were that of the agent while the latter were from the background. This arrangement of agent being nearer to the camera than the background holds true in every scenario (excluding the agent's shadow). Thus, the lower quartile(Q_1) value of the distribution was used to represent the Z_c-coordinate of the agent within the bounding box. For the left edge frame's distribution in figure 7.7a, the latter peak represented the background and has a higher value than the former peak representing the agent. Thus, taking the absolute maximum value of the histogram from the distribution can be risky and the lower quartile value accounts for such distributions also. The quartile values in figure 7.7a, 7.7b are not very close to the maximum histogram values but making these quartile values fit to the sample dataset can lead to over-fitting.



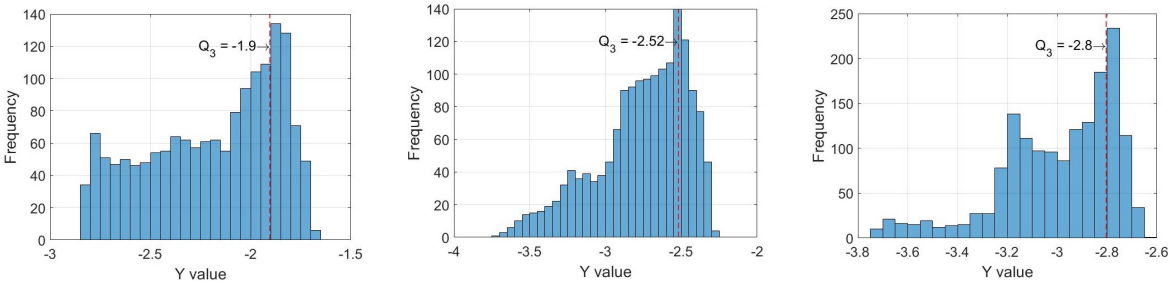
(a) Left edge frame (b) Centre frame (c) Right edge frame

Figure 7.4: Bounding boxes with reference boxes of reference distances 30, 20, 5 pixels.



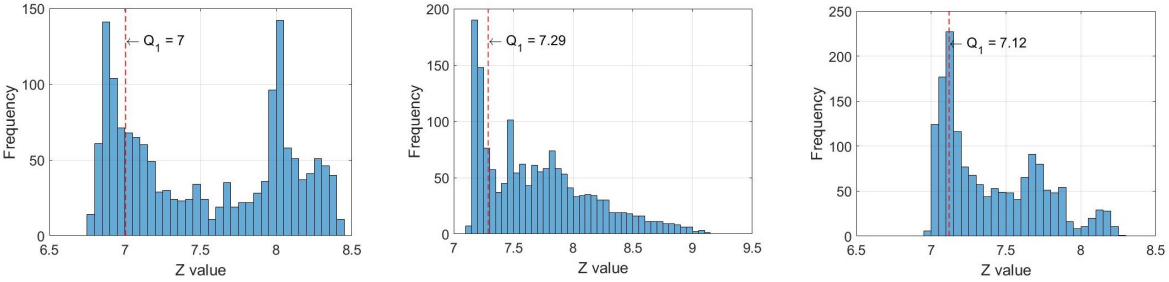
(a) Left edge frame (b) Centre frame (c) Right edge frame

Figure 7.5: X-distribution of agent from the reference box



(a) Left edge frame (b) Centre frame (c) Right edge frame

Figure 7.6: Y-distribution of agent from the reference box



(a) Left edge frame (b) Centre frame (c) Right edge frame

Figure 7.7: Z-distribution of agent from the reference box

For bicycles, the depth data was unstable and inaccurate due to two reasons. Firstly, the bicycles itself are less solid than people (less pixels to refer to) which makes it difficult for the 3D-camera to calculate its depth accurately. Also, the reference box at the centre of the bicycles meant targeting the rider's legs for depth information as seen on figure 7.8. As the legs were in constant motion, the depth information derived using the reference boxes was unstable. This resultant distribution of extracted 3D-coordinates can be seen in figure 7.9 for reference distance of 20 pixels. It shows that there are several peaks in the distribution which made it harder to identify the true depth for a bicycle. Hence, to derive trajectories of bicycles, person only detection was used similar to pedestrians as described above. Person only detection was capable of detecting the bike's rider while giving stable and reliable depth information which was used to represent the bicycle in the observed space.

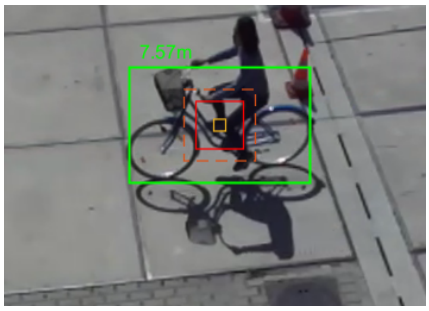
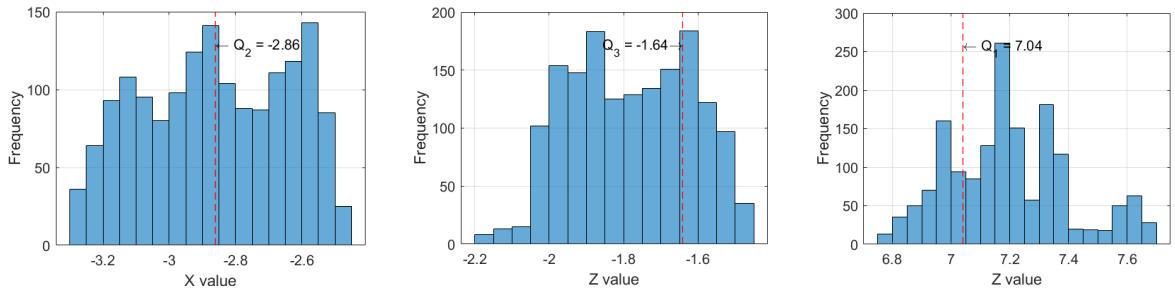


Figure 7.8: Bicycle detection with reference boxes



(a) X-distribution (b) Y-distribution (c) Z-distribution

Figure 7.9: Coordinate distribution using reference box (R=20) in bicycle detection

7.2.2 Reference box selection

After deciding the statistical operations, the next step was to decide the size of the reference box for referring to the agent's depth information. The 3D-coordinates extracted using the statistical operation from the previous section was used to calculate the distance between the camera and the centre of the agent (D = $\sqrt{X^2 + Y^2 + Z^2}$). This calculation was done for five different sizes of the reference boxes at R = 5, 10, 15, 20, 25, 30 pixels respectively over all the ten sample frames. Difference in distances was used as the selection criteria for reference box's size as it enables the use of extracted coordinated from the depth map as it is without any further modifications. This also helped to aggregate the differences caused in all the three coordinates and expressed it in a single value.

For perspective, figure 7.4 shows the detection boxes in green while the reference boxes with reference distance 30, 20 & 5 pixels with dashed orange, solid red and solid orange

box respectively. The percentage difference between the distances using the extracted camera coordinates and the calculated coordinates in section 7.1 was computed. These values can be seen in appendix table A.3. Table 7.1 gives the average value of the percentage difference in the distances over all ten sample frames. As the average distance measured was about 8.5 meters, the ΔD(%) of 4% means that the difference between the calculated distance using on-site measurements and the derived distances using the camera was about ±0.34 meters. This difference aggregates all the possible errors caused in on-site measurements, position of the reference box, the statistical approach and the rounding-off of values during calculations. Figure 7.10 represents the percentage of difference in distances(using the quartile coordinates) for each reference distances with the standard deviation of this difference across the ten frames.

Table 7.1: Average error across frames for each reference box

Reference distance	**Δ $D_{max.hist.}$ (%)**	**Δ $D_{quartile}$ (%)**	**Std. deviation b/w frames**
R = 5	3.44	3.77	2.06
R = 10	4.25	4.35	1.56
R = 15	4.36	4.50	1.32
R = 20	3.63	3.65	1.54
R = 25	3.52	3.40	2.27
R = 30	3.84	3.15	2.50

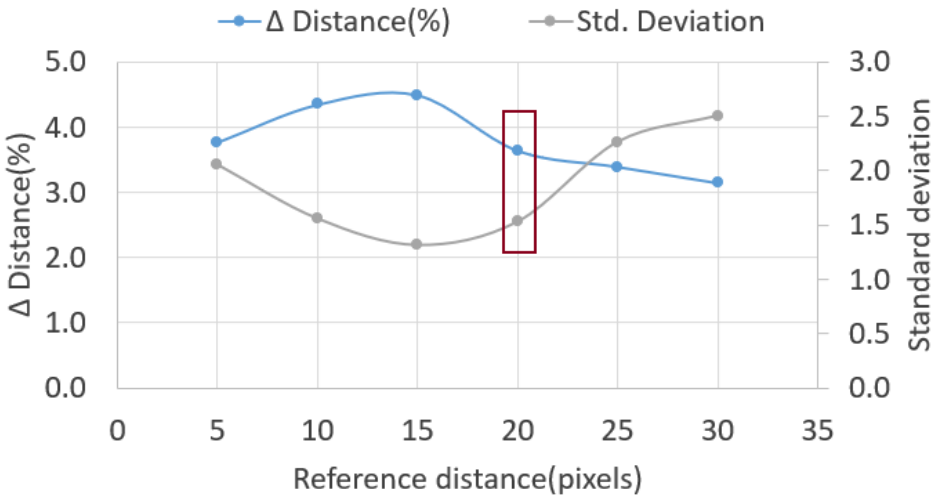


Figure 7.10: Average difference in derived distances w.r.t each reference distance

The reference box with reference distance of 5 pixels provided a good distance estimation for the centre frames but performed poorly for the edge frames having a larger standards deviation. This was due to its small sample size of only 10 x 10 pixels. For reference distance of 10 and 15 pixels, the increase in percentage error shows that pixel values from the background were making the quartile values deviate from their respective true maximums in the histogram. Overall, the reference distance of 20 pixels performed the best with minimum standard deviation in the derived distances across all the sample frames. Thus, the reference distance of 20 pixels was used further in this project.

7.3 Ground plane representation

The camera was installed in a tilted position while recording. Thus, the camera's coordinate axes were tilted w.r.t. the ground plane. This created a need to transform the camera's coordinate system to enable the representation of extracted 3D-point on the ground plane. This section elaborates on this coordinate transformation approach.

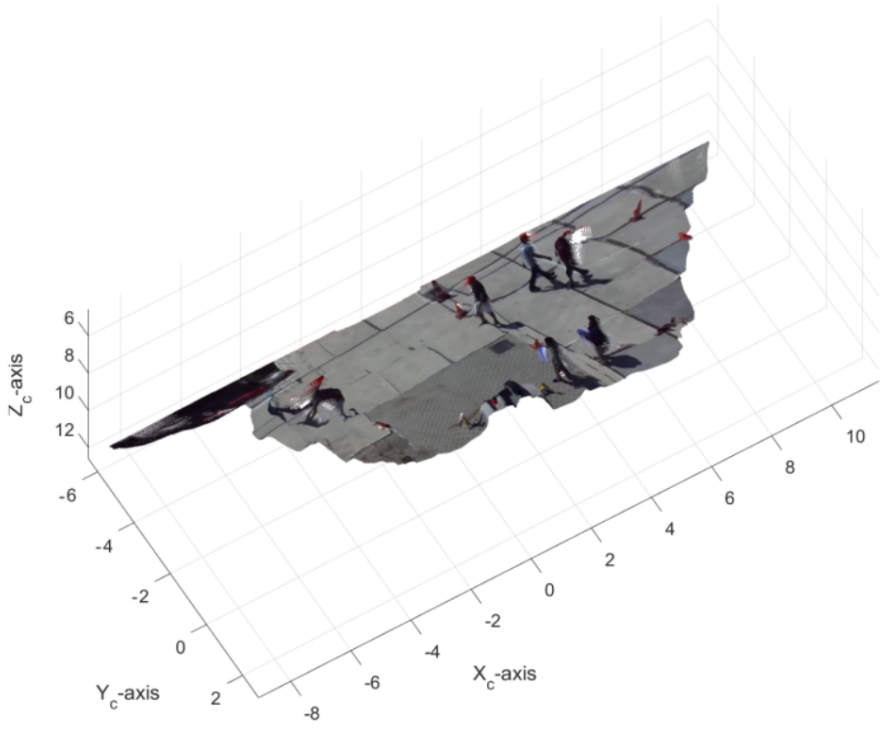
In the following explanation, the X_C, Y_C & Z_C stands for the original camera's axes while X_G, Y_G & Z_G stands for the transformed axes used for ground plane representation of trajectories. Figure 7.11a represents the Green Village scene w.r.t. the original camera axes. This diagram clearly shows that original Y_C & Z_C axes were tilted w.r.t to the ground plane. X_C-axis was also observed to be slightly tilted in the visual footage. As the trajectories were to be represented on the ground plane, the original coordinate system was to be transformed in a way to make the new X_G-Z_G plane being parallel and Y_G-axis being perpendicular to the ground plane. For both the coordinate systems, the point of origin remained unchanged at the camera's left lens.

As there were two set of tilted angles, the coordinate transformation was done in two steps, one step for each angle as shown in figure 7.12. Initially, the Y_C & Z_C axes were rotated anti-clockwise along the X_C-axis to give the Z_G-axis parallel to the ground plane and an intermediate Y'-axis. Finally, the X_G & Y' axes were rotated clockwise along the Z_G-axis to make the X_G-Z_G plane parallel and Y_G-axis perpendicular to the ground. The equation for axis rotation for any point $\mathrm{A}(x_c, y_c, z_c)$ in the original plane to obtain new coordinates $\mathrm{A}(x_g, y_g, z_g)$ in the transformed plane can be expressed as:

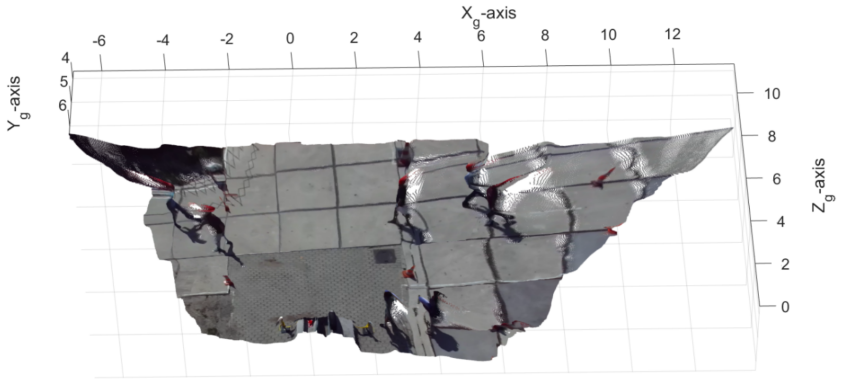
$$\begin{aligned} z_g &= z_c.\cos\theta - y_c.\sin\theta, \\ y' &= z_c.\sin\theta + y_c.\cos\theta \end{aligned} \tag{7.1}$$

$$\begin{aligned} x_g &= x_c\cos\Phi - y'.\sin\Phi, \\ y_g &= x_c.\sin\Phi + y'.\cos\Phi \end{aligned} \tag{7.2}$$

During the Green village experiment, the camera's angle of tilt(θ&Φ) were unknown. Thus, the calculated coordinates of the agent from section 7.1 were used to estimate these angles for coordinate transformation. As these coordinate points were calculated along the ground plane axes, this gave the x_g, y_g, z_g coordinates while the camera provided with x_c, y_c, z_c coordinates of the same agent. By substituting these values in equations 7.1 & 7.2, the unknown angles θ&Φ were calculated as 55.65°and -5.31°anticlockwise respectively. For further reference, the calculation are provided in appendix table A.4. Based on these angles of tilt, all the obtained coordinates of the agent on the tilted camera axes were transformed to obtain the coordinates on the ground plane axes as seen in figure 7.11b. This figure shows that the new X_G-Z_G plane was parallel to the ground plane. Thus, these transformed coordinates were used to represent the detected agents on the ground plane.

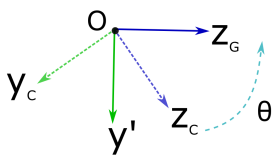


(a) Original camera axes

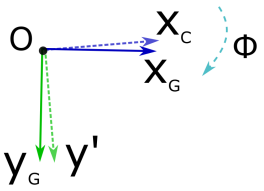


(b) Transformed ground plane axes

Figure 7.11: 3D representation of Green Village scene



(a) $Y_c - Z_c$ axis rotation along X_c-axis



(b) $X_c - Y'$ axis rotation along Z_c-axis

Figure 7.12: Coordinate transformation for Ground plane representation

7.4 Result & Analysis

In practice, the detection, 3D-point estimation and coordinate transformation was done using a single python code over a SVO file video provided by the Zed camera. This code was inspired and build upon the code published by Stereolabs [64]. The outputs were stored in a CSV file with each column representing the frame number, bounding box coordinates w.r.t. the visual image and the 3D-point of the detected agent w.r.t. ground plane axes. A sample data output can be referred in appendix table A.5. Some of the results are visualised in figures 7.13, 7.14. The resultant points on the ground plane are assessed based on the their continuity, stability and the extent of false swaying. To do this, the ground plane points of the agents for all the three scenarios are represented in two ways, (i) as trails in figure 7.13 and (ii) as trajectories for a single agent in figure 7.14. These figures are explained further in this section. While analysis these results, some of the reasons for anomalies in the ground points have been explained and the possibilities of deriving agent's direction of motion and travel speeds from this data have also been discussed.

The visual pictures in figures 7.13a1, 7.13b1, 7.13c1 represent the detections on visual images which are also represented on the ground plane in figure 7.13a2, 7.13b2, 7.13c2 respectively by the red dots. The smaller blue dots represent the trail of detections over the past frames. For perspective, the traffic cones placed around the interaction area were also represented on the graph with a red cross. These trails are a projection of detections on the ground plane. Such representation provides further insights into the performance of detection and the projection of point using the above methodology. It also helps in understanding the speed and direction of motion of the detected agents. These individual points can be selected and joined to obtain continuous trajectories for an agent. As an example, in figure 7.14 one trajectory for each scenario is visualised. This was done by selecting an agent in each of the three scenario (pedestrian only, mixed traffic and cyclists only) and manually extracting its ground-plane points for all the frames in which it exists.

For pedestrians in figure 7.13a, the direction of trail represents the past locations of the agent and the length of the trails represents the distance covered by agents in last 60 frames($\approx$ 1 second). The areas 4, 6, 7 represent discontinuous trails due to lost detections. Area 4 has lost detections in the past frames mainly due to occlusion while areas 6, 7 has lost detections without any occlusion. Sharp swaying was observed as marked in areas 2, 3 can also be seen in the plotted trajectory of pedestrian in figure 7.14a. This was due to two reasons. Firstly, there is always some natural swaying of agents while walking with each step from left to right. Secondly, agents also move their hands back and forth while walking. As the depth was referred using the reference boxes near the agent's waist area, this caused the hand to be a part of depth information in some instances. Thus, a more abrupt and sharp swaying in the extracted trails and trajectories was observed than what might be observed in a natural movement. The diagonal points as marked in area 1 was caused due to changed orientation of agents within the bounding box causing the reference box to extract depth of the background. A similar patter of diagonal movement can be seen in area 5 which was caused by inaccurate detection box bounding two agents in one box.

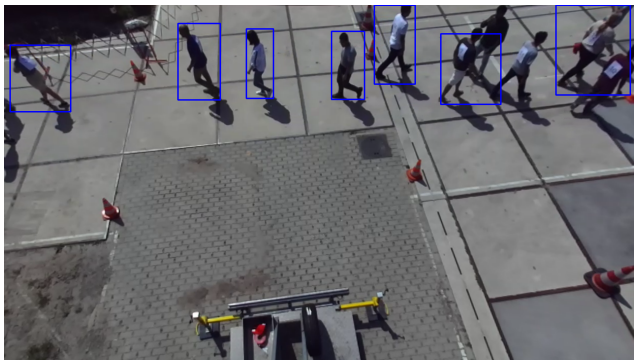
In the mixed traffic scenario, the trails displayed in figure 7.13b2 are for both pedestrians and cyclists for past 100 frames ($\approx$1.67 seconds). Longer trails of cyclists than the pedestrians of the same number of frames convey that cyclists are travelling at a higher speed than pedestrians. The discontinuous points in area 2 represents the lost detections for cyclists. Inclusion of shadows and nearby pedestrians in a single box during detections also caused

error and instability in ground plane projection of detected agents as seen in area 1. In this scenario, as pedestrians were moving vertically in the image, the extracted depth information was more stable and no sharp swaying was observed due to hand movements. Thus, the plotted trajectory of pedestrian in figure 7.14b is smoother and without any abrupt swaying. A slight bump in trajectory was caused while the agent was turning near the camera's pole. During this turning, the camera viewed only the head and shoulder of the agent. Thus, the detection box was only bounding the agent's upper body for which the depth was also extracted and represented in the trajectory. As the agent moved away from the camera, the box resizes to correctly bound the whole agent from head to toe. This caused the ground plane points to shift back to its correct position(near the waist). The change in box size due to changing agent's orientation caused a sense of false movement(or bump) in the plotted trajectory.

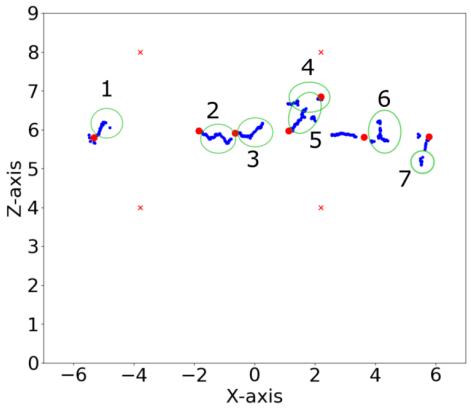
For the cyclists only scenario, the trails represented in figure 7.13c2 are for the past 100 frames. These ground points are much coarsely spaced even in case of continuous detections that what is observed for bicycles in a mixed traffic scenario (figure 7.13b2). This suggests that in cyclists only scenario, the riders were travelling at a higher speed that the mixed flow scenario. The discontinuous points as marked in areas 1,2,3 were due to lost detections. From the bicycle trajectory in figure 7.14c is much smoother than the pedestrian trajectories in figure 7.14a. This suggests that overall, the reference boxes for bicycle riders provided with a stable depth distribution. A gradual change in the direction of motion beyond x=5 was observed as the agent was making room for a U-turn to come back into the interaction area.

As explained in previously in section 6.4, the agent's orientation and the size of the detection box were constantly changing as the shape and size of moving agents change. This constant resizing of the detection boxes has a direct impact on the positioning of reference boxes which does introduce false motion along the Z-coordinate. Some of these false motions were very evident as case of the bump in figure 7.14b. For cyclists, such anomalies were difficult to observe in the obtained trajectories in figure 7.14c. The extent of this false motion along the Z-axis depends on the angle of tilt for the camera. Larger the angle of tilt w.r.t. ground, larger the amplitude of this variation.

Even after the constant change in box sizes and agent's orientation within these boxes, this methodology of 3D-point extraction and ground plane representation provided a largely stable and continuous set of points as in figure 7.13a2, 7.13b2, 7.13c2. These graphs also provide a clear sense of direction and speed of agents which matches the visual observations. While extracting the trajectories for pedestrians and bicycles, the number of frames and the distance travelled was roughly estimated to calculate the average speed of that agent. For instance, the pedestrian trajectory in figure 7.14a existed for 535 frames($\approx$8.91 seconds) to cover a distance of 10.8 meters which gives an average speed of 4.36 km/hr. Similarly, the agent's walking speed during mixed flows was calculated to be 3.5km/hr. This suggests that the agents were walking at higher speeds in the pedestrian only scenario than in the mixed flow environment. This again matches the visuals observations from the video and on-site observations during the data collection experiment.

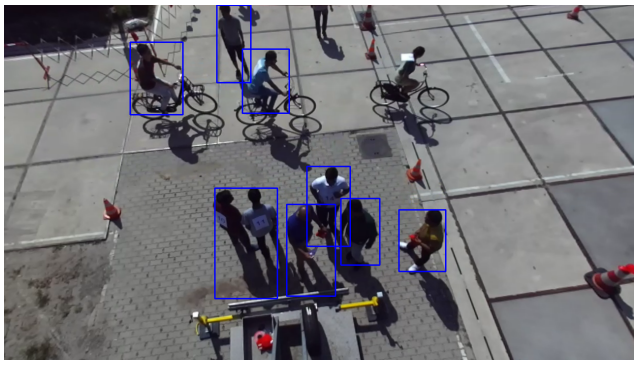


(a1) Detection boxes

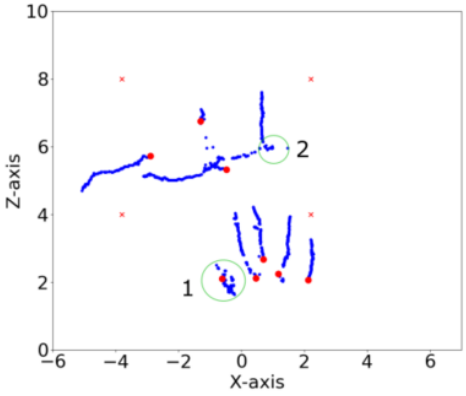


(a2) Trail on ground

(a) Pedestrian only scenario

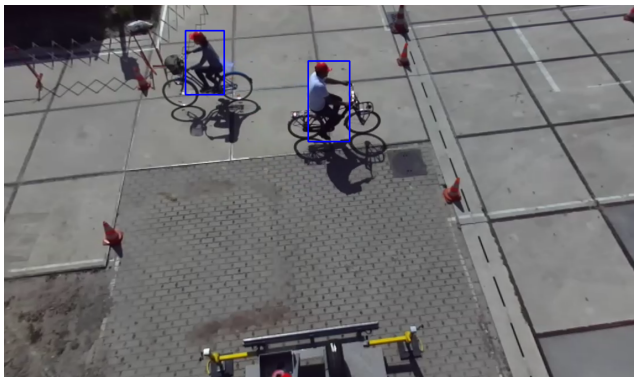


(b1) Detection boxes

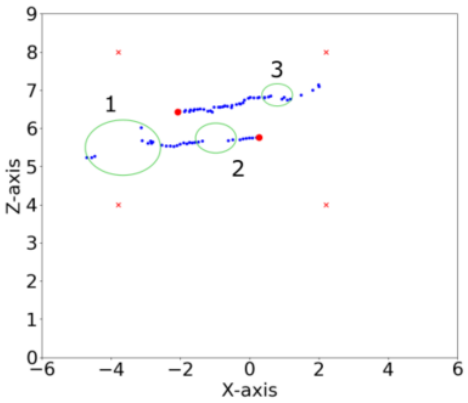


(b2) Trail on ground

(b) Mixed traffic scenario



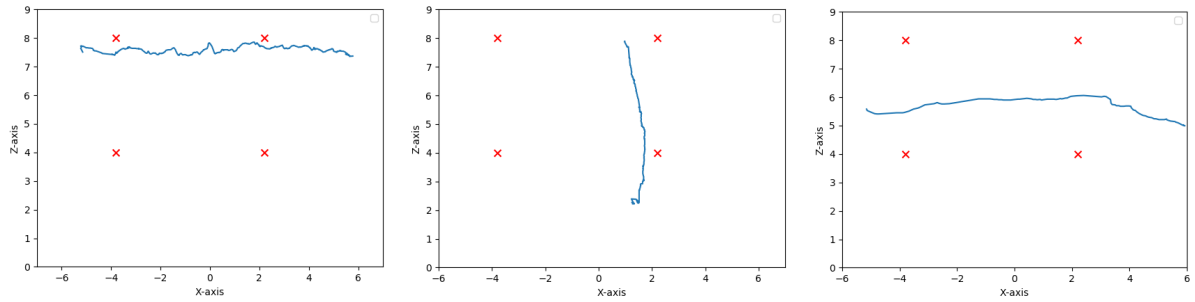
(c1) Detection boxes



(c2) Trail on ground

(c) Cyclists only scenario

Figure 7.13: Green Village detections with trails on ground for all three scenarios



(a) Pedestrian only (b) Mixed flows(pedestrian) (c) Cyclists only

Figure 7.14: Plotted trajectory using extracted points on ground plane

7.5 Discussion

The above approach of using the reference boxes in combination with the statistical operations tries to minimise the inaccuracy passed on by the detection model in its detections and the possible uncertainty in camera's depth estimation. These inaccuracies in terms of missed detections and unstable box sizes can always be minimised by improving the detection model itself. But, the design choices made in the above process can also have many different possibilities which can improve its performance with the existing detections. Such as the shape, size and aspect ratio of the reference box can be varied depending upon the agent's position and its orientation in the image. Similar variation while choosing the statistical approach in selecting the relevant 3D-coordinates of the agent can be possible. A more idealistic approach for selecting a statistical operation can be to take the coordinate with maximum histogram value from the extracted distribution. This approach was computationally expensive as it was performed for each detection in every frame of the video. While designing such approaches, always only a small sample of the total dataset can be analysed and optimised for. This makes it necessary to choose these samples in a way that they are normally distribution over all possible scenario. This can make sure that the design choices made will hold true for a majority of the dataset. Most importantly, there is always a risk of making the design choices such that they might be over-fitting on the sample dataset and should be avoided.

While optimising the reference box's size, the absolute distance from the camera was used. As the agent was to be projected in the ground, another possible approach can be to use the distance measured along the ground instead. To calculate this distance the coordinate transformation needs to be performed beforehand. Thus, this approach might introduce additional errors due to coordinate transformation and might not provide the true measure of camera's coordinate estimation. For the Green Village data, the angle of tilt was not measured during the data collection. This lead to the development of an approach to calculate the angle while using other site measurements and the available 3D-data as reference. This method can also be used to validate the on-site measurements for angle of tilt provided that all the reference data is reliable and accurate.

The existence of anomalies in the projected points of an agent due to some subtle changes in the depth (such as pedestrian's hand movement) suggests that the Zed camera was able to detect these small changes. Thus, the Zed camera is sensitive and is able to observe even minor changes in the depth information.

The projected points on the ground plane are the most basic input into many information

of motion of the agent. For example, these points can be used to obtain instantaneous speeds, direction of movement and individual spacing of agent w.r.t its surroundings. More aggregate information representing the nature of space such as flows, densities, travel times, average speeds, etc can also be calculated using this data. All this data can help in understanding the movement of people in the real-world environment.

7.6 Summary and Conclusion

To summarise the above processes, initially the detection boxes were combined with the depth information from the camera to provide 3D-points of the detected agents. These 3D-points were first obtained on the camera axes which was tilted w.r.t the ground plane. Thus, the obtained coordinated were transformed such as to make on of the planes (X-Z plane) parallel to the ground plane. This enabled the projection of 3D-point of the detected agent on the 2D-ground plane to obtain trajectories on the ground plane.

For extracting the 3D-data of the detected agent, a combination of reference box and statistical operation was used. For people, the extracted distribution of 3D-coordinates using the reference boxes provided a clear distinction between the agent's depth and the background's depth distribution. Thus, the choice of statistical operation to refer to only the agent's coordinates and filtering the background pixels was based on this nature of distribution for each of the X,Y,Z coordinates. For bicycles, the extracted distributions were unstable which did not allow a clear identification of bicycle's true coordinates. Thus, it was decided to detect and project the bicycle's rider instead of the bicycle itself.

After the coordinate transformation of the camera's original axes to the new ground plane axes, all the extracted points were projected on the ground. These results were then visualised as a trail of ground plane points side-by-side with its respective visual image. This helped to identify the anomalies in the projected points and make qualitative conclusions on the performance of the proposed methodology to represent the agent on the ground. The anomalies observed in the projected points on the ground were mostly related to lost detections and irregular boxes which caused a change in reference box's position. Other irregularities like swaying and small vibrations observed in the projected points were also due to small changes in the depth information due to agent's motion. This suggests that the camera is sensitive and can precisely detect these changes in the visual images. Largely, this methodology to project agents on the ground provided with a stable and continuous set of points but was highly dependent on the quality of input detections. The trail of points also helped in understanding the speeds and direction of motion of the detected agent. In the next chapter, these points are used to derive agent trajectories on the ground plane.

Chapter 8

Agent tracking

In the previous chapter, the detections obtained on the visual images were projected on the ground plane using the 3D-data provided by the Zed camera. In this chapter, these points were used to draw agent's trajectories on the ground plane. As this research aims at automating the whole data extraction process, thus the first section investigates into the existing tracking models capable of tracking the agents using the available points. This section discusses the process of model selections followed by a brief description on the working of the selected tracking model. The third section describes the process of using the tracking model including the parameter selection for running the model. The last sections analysis the output results of this tracking approach and provides the main conclusion points of this chapter.

8.1 Model selection

For each frame in the video, the agent was represented as a point when detected. This was the only information available for the tracking to be performed. In case of multiple agents in a single frame, each frame generates multiple points representing every detected agent. To obtain trajectories of detected agents, these projected points need to be uniquely identified and associated over multiple frames for each agent. On the basis of the input data available and the tasks to be performed by the tracking model, two main criteria for selecting a tracking model are identified. (i) The tracking models should be able to track agents using information of detections only. For this research, this information is represented in the form of 2D-coordinates on the ground plane. (ii) As there can be many agents in the same frame, thus the model also needs to distinguish and track multiple objects simultaneously. Following this, the existing tracking models are assessed and one model is selected to be used further in this research.

Most of the multi-object tracking frameworks such as Similar multi-object tracking (SMOT)[57], Deep affinity tracking[58], Markov Decision Processes tracking (MDP)[59], Deep simple online tracking (DeepSORT)[60] uses a combination of agent's location obtained from the detection model and its appearance information from the visual videos. Hence, these models make use of additional information rather than just the 2D-point information which is the only information available in this stage. Also, using such tracking algorithms might perform well in terms of tracking but, will not provide much insights into the performance of the proposed framework of representing agents on the ground plane.

In year 2016, Bewley et. al. proposed a Simple online real-time tracking (SORT) which uses only the object's detection formation to perform its tracking through the video [2]. It uses a simple, rule-based tracking approach and combines prediction and association processes. Using this approach, it is able to track multiple object and provide unique ids to each detection. Using only the detection information for tracking also makes the performance of such models largely dependent on the continuity and stability of these detections. As the SORT model fulfilled both the criteria of selection, this model is used further in this research. This model was originally developed to perform tracking using object's location on the image (using image coordinates) but this projects uses this model to perform tracking using the ground plane points. The next section elaborates on the working of SORT tracking model.

8.2 Working of SORT tracking

SORT is an abbreviation for Simple Online Real-time Tracking. The word 'simple' stands for the simplicity of the tracking approach in this algorithm, word 'online' explains that this tracking framework does not need future frames for its working and relies only on the agent's location data from past frames. The word 'real-time' refers to the speed at which this algorithm is able to process data and assign ids to each detection. The model's source code was in python programming language and was made public by the author [71]. The following explanation of SORT's working was understood from its original release paper [2]. The original paper proposed this model to track agents using the image coordinates, but the below working is explained from the perspective of using it to track agents on the ground plane.

The input data consists of the detected agent's location and its frame numbers. For this research, the location of detected agent on the ground plane was used with each frame as an input. The input data format with some sample rows is provided in appendix table A.6. The tracking process in SORT is divided into two steps, prediction and association of detections. For prediction, first a constant velocity model is used to form the equation of motion of the agent based on its past locations. Kalman filter then uses this equation of motion to predict the agent's location in the next frame. This process of prediction helps to carry-forward the agent's id from past frames to the next frame and narrows down the search area for id-association. These predicted locations of the agent then are associated with the given detected locations using Hungarian algorithm. Here, a small box around the agent's ground plane location is drawn and the area of overlap between the predicted box and the detected box is calculated. This area of overlap is also referred to as Intersection over Union(IoU). As there are multiple agents in the same frame, Hungarian algorithm solves the assignment problem using the overlapping area where IoU_{min} acts as a threshold value for association. During tracking, the detections are treated as the true value and the equation of motion gets updated after association. The Kalman filter constantly tries to minimise the error between its predicted location and the detected location of the agent. The final output assigns the ground plane points with there respective ids. Sample output data can be referred in appendix table A.7. Parameters governing the process of id creation, association and deletion are explained in the following section.

8.3 Parameter Testing

A detailed explanation into the working of these parameters values was missing from the model's release paper (some of the parameters were not even mentioned in the paper). Initially, this section identifies and explains the role of each of the parameters governing the SORT tracking model. On the basis of this, the later paragraphs identify the performance indicators and tune the parameter values for the given dataset using different run-time parameters.

Identifying parameters and their role

The first step to identify the parameter values was to develop a detailed understanding into the model's source code. In process, three main parameters were identified namely Minimum hits(min_{hits}), Maximum age (max_{age}) and Intersection over Union threshold(IoU_{min}). In the next step, various runs with different combination of parameter values were performed to clearly understand the role of each parameter on the obtained output. The role of each parameter is explained in the following paragraph.

Minimum hits (min_{hits}) controls the creation of a new tracker id for the available detection. This value determines the minimum number of consecutive frames for which a new detection needs to be successfully associated with other new detections. If a detection fulfils this condition, then only it is assigned with a new id and if otherwise, the detection is considered invalid and is deleted. This measure helps to check for false-positive errors passed on by the detection model. Maximum age (max_{age}) determines the maximum number of frames for which an id will be kept alive (or active) without any valid associations. If an id is not associated with any detection after max_{age} number of frames, then the ids is deleted. When an id is without any association, the tracking model constantly predicts its possible location until max_{age} number of frames. This values helps to account for some of the false-negative (or missed) detections in the input data for max_{age} frames. If the missed detections are greater than max_{age} frames but the agent is still present in the frame, its id will be falsely deleted. The resultant trajectories are fragmented and discontinuous. IoU threshold(IoU_{min}) determines the ease of association between the predicted location and the actual location of the agent. The effects of choosing different parameter values for the given dataset is explained in the the following paragraphs. Next paragraph explains the dataset used for tuning these parameter values.

Test setup

To tune these parameter values, the data of two scenarios from the Green Village were taken as input. The reason behind taking two different datasets was to investigate if the performance of the parameter values was sensitive to the type of input scenario. One scenario with pedestrian only traffic which had less number of agents and posed a less challenging situation of the tracking model. This scenario had bi-directional flows with 14 individual agents present over all 2500 frames. It had 3-5 agents present in each frame with each agent taking on average of 300 frames ($\approx$ 5 seconds) to cross the interaction area. Another scenario used for testing parameter values was a mixed flow scenario with both pedestrians and cyclists. This scenario posed a more challenging situation for the tracking model and helped to test its parameters in relation to the shared-space conditions. Here, the number of frames were 2000 with a total of 35 agents (20 pedestrians and 11 cyclists). Each frame consisted of 5-7 pedestrians and 2-

3 cyclists. On an average, an agent existed for 180 to 250 frames depending on the mode of travel.

Based on the explanation into the working of each parameter value and its effect on the output, two main indicators of performance were identified for tuning the parameter values. These indicators are (i)number of ids assigned and (ii)their lifespan. Lifespan of an id is expressed in number of frames and it stands for the total number of frames for which that id exists. To better asses the output, four threshold values for the id's lifespan value were decided based on the observed lifespan of the agent in the visual video as describe in previous paragraph. These thresholds were placed at 60 frames, 120 frames, 240 frames and 360 frames. Total number of ids above these threshold values and there average lifespan were used to asses the output from every run. Dividing and representing the results based on these threshold values allows the visualisation of distribution for the ids with there respective lifespans. Less number of total ids with more number of ids existing for a longer lifespan was the desired nature of the output.

Four different combinations of run-time parameters are explained in the following paragraphs with values as given in table 8.1. In practice, there were many runs performed (about 15-20 runs) but this report discusses only four runs to explain the trend and effect of each parameter on the output. The output for both the scenarios w.r.t all the four runs is provided in figure 8.1. The parameter values of run-1 serve as the base values for all the other runs. For this run, the values were selected based on the continuity and stability of the input detections and the understanding of the parameter values from the source code. Following the base run, the value of each parameter was changed (one parameter at a time) to observe and explain the effect of each parameter on the output. After every run, its output is compared with the output from previous runs to explain the trend due to the change of each parameter value. The following paragraph compares the explains each run one-by-one and selects the best performing run-time parameters.

Comparison of outputs from different parameters

Here, the comparison of outputs from different runs is done using the results visualised in figure 8.1 and the total number of ids in table 8.1. The performance indicators on which the outputs are assessed have been explained in the previous subsection. Compared to the base run, in run-2 the value of min_{hits} needed for id generation was increased from 1 frame to 5 frame. This decreases the total number of ids generated as the id generation gets dearer. Thus, more detections were considered as invalid which reduces the number of ids existing above each frame threshold value. As the number of detections being a part of each trajectory decreases, the average lifespan of the ids also reduces. Thus, $min_{hits} = 1$ was fixed for all other runs. In run-3, the IoU_{min} was increased from 0.1 to 0.5 which makes the criterion of associating the predicted locations and the detected locations more stringent. This leads to an increase in the number of unassociated but valid detections which increases the total number of ids generated. A similar trend was observed in run-4 when the max_{age} of the tracker-id was reduced from 30 frames to 5 frames. Both of these changes in parameter values increases the number of ids with shorter life spans. Overall, run-1 provided with the most number of id with higher lifespans than any other run. Thus, the parameters of run-1 were used to assign ids and obtain trajectories in this research.

Table 8.1: Run-time parameters for agent tracking

Runs	Input Parameters			Output (Total ids)	
	IoU threshold IoU_{min}	**Maximum age** max_{age}	**Minimum hits** min_{hits}	**Pedestrian only**	**Mixed flows**
Run 1	**0.1**	**30**	**1**	76	130
Run 2	0.1	30	5	69	122
Run 3	0.5	30	1	162	258
Run 4	0.1	5	1	184	314

(a) Pedestrian only scenario

(b) Mixed flows scenario

Figure 8.1: Number of Ids with their average lifespans for each test run during the parameter tuning for the tracking model

8.4 Results & Analysis

The outputs were stored in a CSV format with each column representing frame number, id number and the ground plane coordinates (X-Z coordinates) of the detected agent. A sample dataset can be seen in appendix table A.7. The ground plane coordinates and there respective frame numbers remains the same as input from the previous process of ground plane representation. The output from the tracking algorithm adds a column of unique ids in this dataset. Using this unique id number, the points belonging to each ids can be extracted to plot the agent's trajectory. Some of the results are visualised in figures 8.3, 8.2.

Figure 8.2 represents the ground plane points for two agents. For perspective, the traffic cones places in the interaction area are represented by red cross. Area 1,2 in trajectory 55 shows discontinuity in the detections. As these missed detections were for less than max_{age} number of frames, the tracking model was successfully able to account for the missed detections (false-negatives) and associate them to the correct ids. While ids 34,43 and part of 54 also represent a single agent but was assigned with multiple ids. In area 3, the number of frames with missed detections were greater than max_{age} which caused the deletion of old id(34) and the creation of new id(43). Here, some of the detections represented by the grey dots were not associated with any id as they were not able to associate itself with any other detections within min_{hits} number of frames. Thus, in case the projected points were too off and away from any other valid detections (false-positives), they were not associated with any other valid detections and deleted from the output. Area 4 was a case of id-switch when the points from two different agents were falsely associated. This occurs due to the closeness between the predicted points of one agent and the projected points of another agent. Thus, in case of unstable and discontinuous detections, a longer prediction periods (or higher max_{age}) increases the chances of false associations and id switches.

Figure 8.3 represents the trajectories obtained for each scenario which exists over 120 frames. The anomalies observed in the trajectories are in the form of false-vibrations, swaying and discontinuous tracks. These anomalies exists in the agent's projected points itself which is already explained in section 7.4. To avoid repetition, please refer to the mentioned section which provides a detailed explanation on the reasons behind such anomalies. For mixed scenario, the trajectories were differentiated based on the model of travel by observing the direction of trajectories. All these trajectories can provide insights into the path followed by agents in each scenario. For pedestrian only scenario, figure 8.3a shows the trajectories to be closely packed to each other and only in the horizontal direction. Thus, the agents were walking in a bidirectional fashion and close to each other. While for the pedestrians in the mixed flow scenario, figure 8.3b suggests that they were walking in vertical direction and were more spread out over the interaction area. When comparing the trajectories of cyclists from figure 8.3c and figure 8.3d, the cycle only scenario has a much smooth and laminar flow while in mixed flow scenario, the cycle trajectories are more spread out and turbulent. This was because the cyclists has to constantly change there directions to accommodate pedestrians in a shared space environment while their was no such need in cycle only scenario. These observations derived from studying these trajectories are coherent with the observations on-site and from the visual videos.

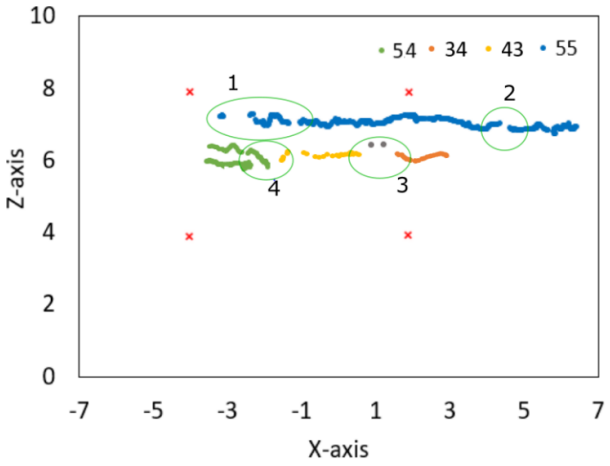
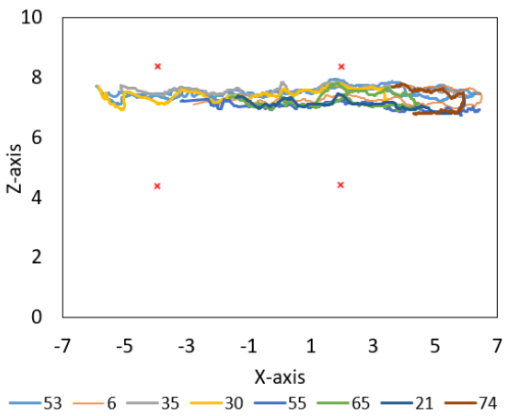
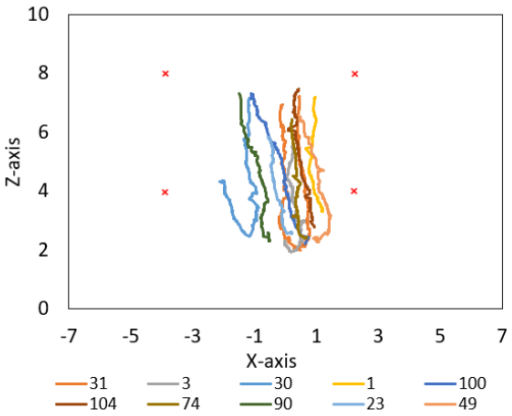


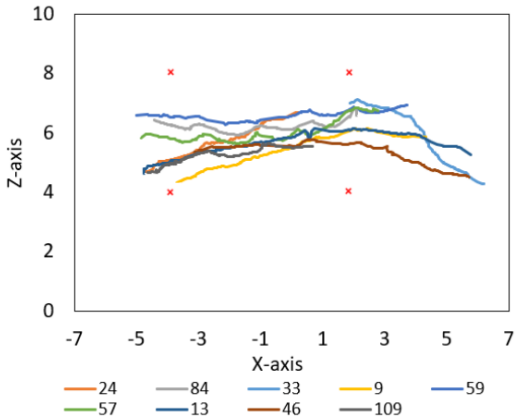
Figure 8.2: Id assignment on the agent's projected points



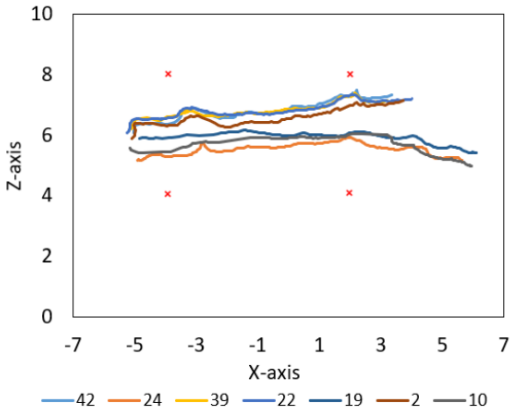
(a) Pedestrian only



(b) Mixed flows(pedestrian)



(c) Mixed flows(cyclists)



(d) Cyclists only

Figure 8.3: Trajectories on ground plane using SORT algorithm

8.5 Discussion

While selecting the parameter values, this research only considers the number of ids and their average lifespans as the performance indicators. These indicators indirectly covers other aspects of output trajectories such as discontinuity and fragmented trajectories but not id switches. In future, these factors can also be included and made more explicit as part of the performance indicators. Much better parameter tuning can be performed if the ground truth trajectories are available as reference data. Also, as both pedestrians and cyclists were identified as people, hence same parameter values were used to track both the modes. Difference in speed and movement patters can lead to different parameter values for each mode. For such mode based tracking, the projected points itself need to be classified based on their travel mode.

Obtaining trajectories from the projected points can be seen as one of the applications of the projected ground plane coordinates of the detected agent. While assigning unique ids to the anonymous detections, the tracking model also tries to account for some of the errors passed on by the previous processes of detection and depth extraction. For instance, false-negative errors in the form of missed detections were accounted for by the tracking process as its was able to assign ids across these missed detections also. False-positive detections were deleted as they were not associated with any other valid detections. In future, the efficiency the tracking model in removing these errors can be quantified.

The selected tracking model was purely rule-based and does not require any training like the detection model. Hence, it can be easily extended to track object other than people. Obviously, this is possible only when the the proposed methodology is able to both detect and project objects categorically on the ground plane. The performance of this tracking approach can be improved by passing more data specific to each detected agent for every frame. One of such data can be agent's visual information which can be included into the tracking framework to obtain better outputs. Few of such tracking models which already make use of such information were mentioned in section 8.1.

8.6 Summary and Conclusion

To get trajectories of agents in the interaction area, the anonymous projected points obtained from the previous processes need to be associated with each other. A multi-object tracking model (SORT [2]) was selected such that it uses only the available points on the ground plane to perform this association. The selected tracking model uses a combination of prediction and association processes to assign ids to every projected point. This model was originally developed to be used for tracking agents using detection coordinated on visual images. In this research, the source code was modified for tracking the detected agents using their ground plane coordinates only.

The model's release paper lacked a detailed explanation and identification of the model's parameter value. Thus, these parameter values were identified by in-depth study of the model's source code. A detailed understanding into the working of these parameter values was developed by performing multiple runs (15$\sim$20 runs) with varying set of values. The output from these runs were also analysed to test and tune the parameters for the given data-sets. The performance indicators used for this comparison were the number of ids generated and their average lifespan. The final parameter values were used to obtain trajectories for all

the three scenarios of the Green Village experiment. Later, these values are also used to obtain trajectories from the Amsterdam dataset.

The trajectories provide the information of path followed by the agents throughout the video. As the tracking model used the series of 2D-points as its main input, hence its performance largely depends on the quality of these points (continuity, stability and other anomalies). The results obtained suggest that the tracking model was able to account for some of the shortfalls (false-negatives, false-positives) of the previous processes of detection and ground plane representation.

Chapter 9

Framework application - Amsterdam Central dataset

In this section, the framework developed in the previous chapters using the Green Village data was implemented on the Amsterdam Central data to extract agent trajectories on the ground plane. This chapter is aimed at implementing and analysing the proposed data processing framework on the real-world dataset to see its working on such datasets. The first section represents the data processing framework developed using the Green Village data. Next section outlines the test setup and the nature of dataset used in the chapter. The next sections analysis the results from each of the data processing stage one-by-one. The last sections analyse the results and discuss on the limitations and conclusions made from this Amsterdam case study.

9.1 Developed Framework

From the previous sections, the developed data processing framework is represented in figure 9.1 with each stage namely: agent detection, depth extraction with ground representation and, agent tracking. For each of these stages, the design choices selected and the parameter values tuned on the Green village dataset are provided in table 9.1. These same values were used on the Amsterdam dataset. Each of the following sections 9.3, 9.4, 9.5 are dedicated to each data processing stage respectively. These sections start with mentioning the finding from the previous chapters and then discuss the results obtained from this dataset .

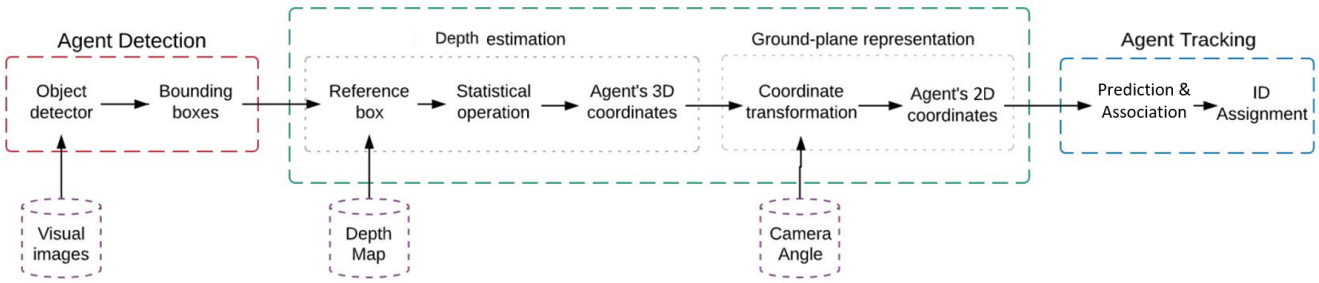


Figure 9.1: Data processing framework for Amsterdam data

Table 9.1: Overview of the design choices and the parameter values of each data processing stage for Amsterdam data

Stage	Design choice	Parameters
Agent detection	Faster R-CNN model	Threshold score = 0.75 Object class = Person
Depth estimation	Reference box	Reference distance = 20 pixels
	Statistical operation	X-distribution = Mean value Y-distribution = Upper quartile value Z-distribution = Lower quartile value
Ground representation	Coordinate transformation	Rotate the original Y-Z plane along X-axis using camera's angle of tilt ($\theta = 22^{\circ}$)
Agent tracking	SORT model	IoU threshold (IoU_{min}) = 0.1 Maximum age (max_{age}) = 30 frames Minimum hits (min_{hits}) = 1 frame

9.2 Test setup

When comparing the Green Village data with the Amsterdam data, the latter had agents with much diverse set of appearances. They were also observed to perform motions in a much more dynamic way and frequently changed their direction/speed of movement. As there was no direct sunlight into the observation, no shadows were formed by the agents. A more detailed discussion on the data collection and on-site observations is provided in section 5.2. As the observed space was near the ferry terminal, higher crowding levels were seen as the ferries docked and its passengers moved through space. Thus, the dataset collected during the Amsterdam dataset was divided into two categories based on the time of ferry arrival which are: (i) Normal conditions (ii) Time of ferry arrival. Normal conditions are times without any traffic of the passengers from the ferry near the shared space. While the other category represents the data collected after the ferry arrival. After the ferry arrival near the observed area, the passengers were observed egressing through the area which created higher crowding levels than the normal conditions. During normal conditions, a single frame of this dataset consisted of 5 to 15 agents in the observed area. While for the other dataset, the average number of agents in a single frame were about 25 to 35 agents. Such division of data allowed the following sections (section 9.3, 9.4, 9.5) to analyse the effect of crowding on the performance of every data processing stage in this proposed framework. These sections also make a comparison between the results obtained on the Green village dataset and the Amsterdam dataset. This provides a better idea on the additional observations and conclusions made while assessing the results from the real-world scenario.

9.3 Agent Detection

For Amsterdam recordings, the people were detected using Faster R-CNN detection model with a threshold score value of 0.75. Object category of people detects both pedestrians and cyclists. This section first discusses different types of detections based on their accuracy which are (i) Good detections, (ii) Inaccurate detections and (iii) False positive detections. Then, it discusses the detections provided in the two different scenarios divided on the basis of ferry arrivals. For better comparison, the number of detections are also quantified on a sample dataset for both the scenarios.

The results visualising different types of detections based on accuracy can be seen in figure 9.2. Compared to the Green village data, the agents in Amsterdam data were captured with a much diverse set of appearances. The bicycle riders were also observed to ride many different types of bikes like cargo bikes, sports bikes and accessorised city bikes. Even after such challenges, the detection model was able to detect agents as shown in figure 9.2a. The absence of shadows in this scene contributed largely in obtaining stable and better fitting detections boxes that the Green Village data. Inaccuracies in the detection boxes due to occlusion was observed similar to Green Village data as seen in figure 9.2b. For agents further away from the camera, the detections were unstable as there object size was small. Sometimes, the reflection of agents passing nearby the glass window was also detected by the model as seen in figure 9.2c. These were the only false-positive which was observed during the Amsterdam experiment.

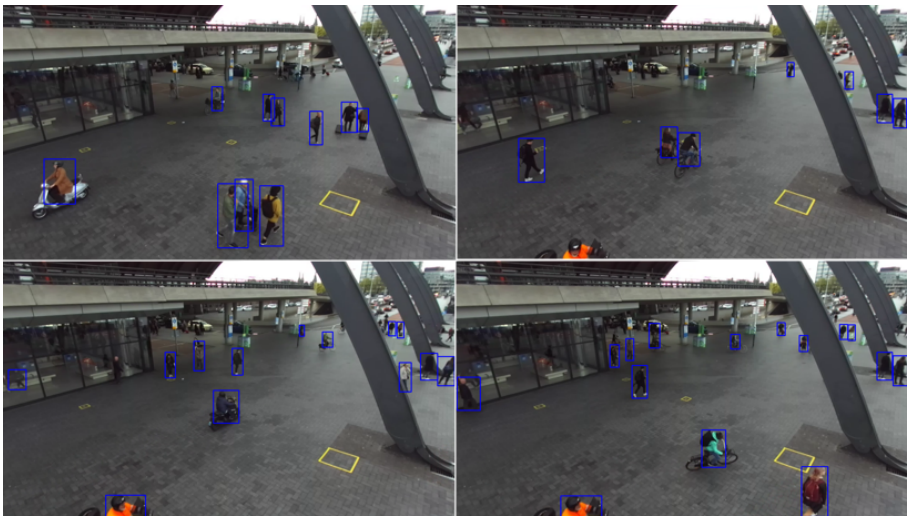
(a) Good detections (b) Inaccurate detections (c) False positive detection

Figure 9.2: Different types of detections based on accuracy for Amsterdam Central data

The snapshots of obtained detection w.r.t. the level of crowding are represented in figures 9.3a & 9.3b. To quality the number of detections in each scenario, a sample of 500 frames was taken and the actual number of agents present in the frame were compared to the number of detections provided by the model. For lower crowding conditions on average, the model provided 5.95 detections against 6 number of people present in each frame. If a single agent was observed for 250 frames in the camera, 235 detections were provided by the model. Here, the agents were being detected even when they were close to each other but not occluding. The main challenge here was agent's occlusion itself. Dynamic appearances of agents did not challenge the detection model at all. Thus, in this scenario the detections were stable and the model was able to detect agents that were visible in the frame. For higher levels of

crowding, on average 12 detections were obtained against 18 visible agents in the frame. When a single agent was observed for a total of 300 frames, only 120 detections across its journey were provided. Here, the occlusion occurred even more frequently as more number of people were observed in the space. But, even in cases when agents were perfectly visible in the image, the model was unable to provide detections for many of such agents. The problem of occlusion was also identified during the detection analysis on the Green Village dataset (section 6.4) but the inability of the model to even detection visible agents is a new observation which leads to the following interpretation of the detection model.

This suggests that the detection model's performance is not directly dependent on the density of people observed but on the level of occlusion and the total number of people visible. So, when the agent is occluded and when their are more people visible, the model tends to miss the detections. But in case of less people, even when they are standing close and not occluding each other, they were detected by this detection model. Such observations could not be made during the controlled experiment due to limited number of volunteers.



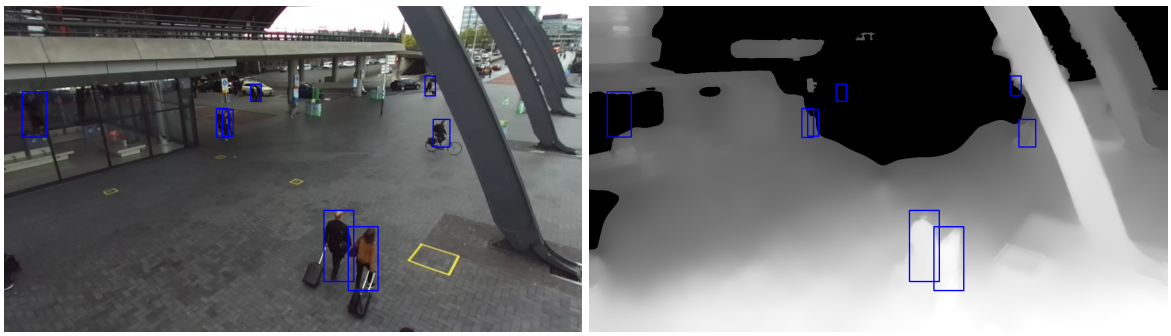
(a) Normal conditions (low crowding)

(b) Time of ferry arrival (high crowding)

Figure 9.3: Detection for the two scenarios based on ferry arrivals for Amsterdam Central data

9.4 Depth extraction and ground representation

After obtaining the coordinates of the bounding boxes w.r.t image coordinates using the detection model, the depth data was extracted using the combination of reference boxes and statistical operations. As proposed, the reference distance was set at 20 pixels and the agent's depth distribution was extracted. To obtain X, Y, Z coordinate value of the agent, statistical operations of mean, upper quartile and lower quartile values was implemented on each distribution respectively. Part of the observed area was out-of-range ($>$ 20 meters) for the camera to calculate its depth information. So, even when the detections were available for people far from the camera, the depth information was unavailable as seen in figure 9.4. Originally, these pixels contained null or no value which resulted in an error and crashed the program while applying statistical operations on these null distributions. In such cases, the depth data was programmed to output zero value for that detection.



(a) Detections on visual image (b) Detections represented on depth map

Figure 9.4: Depth map with detections out-of-range(>20m) for the camera

The ground plane w.r.t camera axis is represented in figure 9.6a. For Amsterdam data, the camera's angle of tilt was measured on-site as 22°. Thus, the Y_c- Z_c axes were rotated along the X_c axis by $\theta = 22°$ to get the transformed axes X_g, Y_g, Z_g for ground plane representation. The ground plane coordinates w.r.t the transformed axes is represented in figure 9.6b. When looked closely, it was seen that the ground plane was not represented as a straight plane but was curved inwards (concavely curved). As the distance from the camera increased, the curvature of the curve increased and formed a bowl-like formation as seen in figure 9.5.

To further support this argument, the 2D-coordinate values of the reference points A,B,C (as describes in data collection section 5.2) were extracted using the Zed camera. The distances between the camera pole's footing, O(0,0) and reference points A(-6.8,11.3), B(-3,14), C(0.9,11.2) as derived from the camera were calculated. The error between the on-site measured distances and the calculated distances is given in table 9.2. This means for Amsterdam dataset, the Zed camera was unable to accurately represent the ground plane as a straight plane. Such curve in the ground plane was not observed in case of the Green Village dataset. This suggests that the Zed camera under the current user settings (resolution, frame rate and other parameters) and experimental setup (like lighting conditions) was unable to accurately estimate the depth of the observed scene.

Table 9.2: Error between the on-site measurements and the calculated distances of the reference points

Points	Measured distances (m)	Calculated distances (m)	RMSE
OA	14.00	13.17	4.74
OB	16.60	14.32	8.39
OC	12.90	11.25	6.31
AB	5.64	4.67	3.16
BC	5.15	4.80	1.86

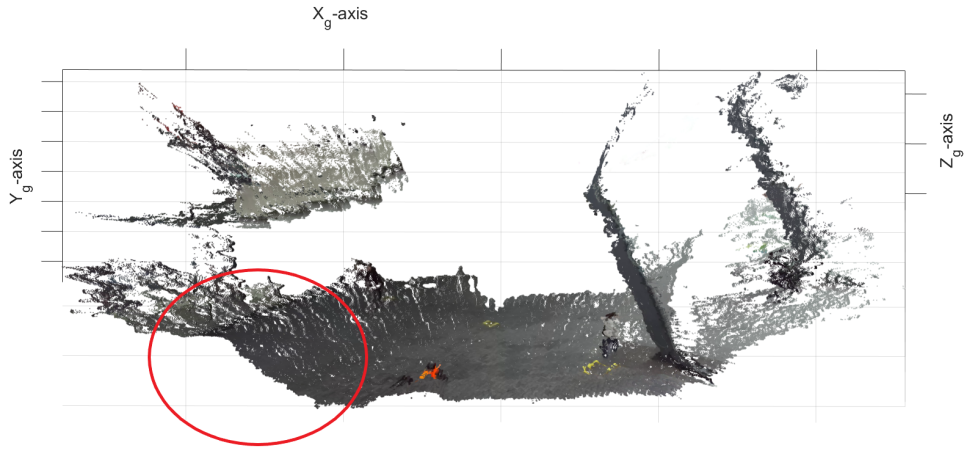
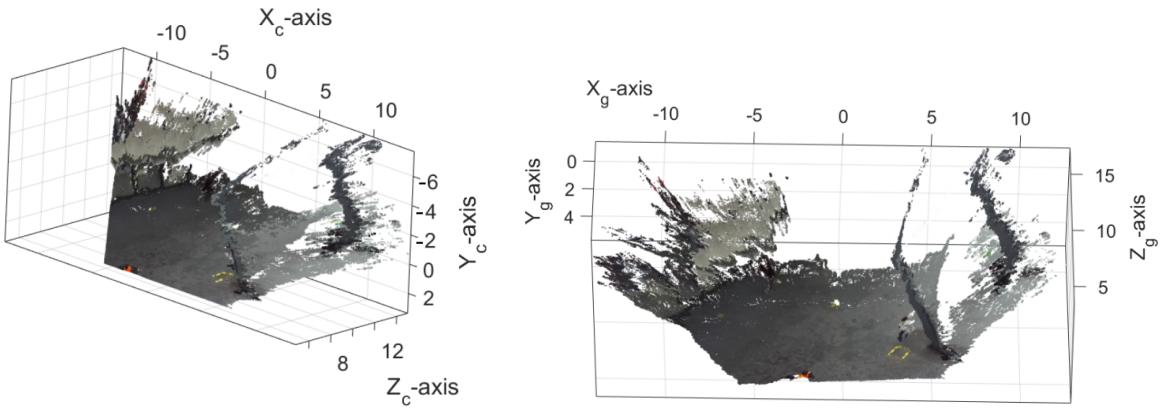


Figure 9.5: Amsterdam's ground plane represented with curved edges



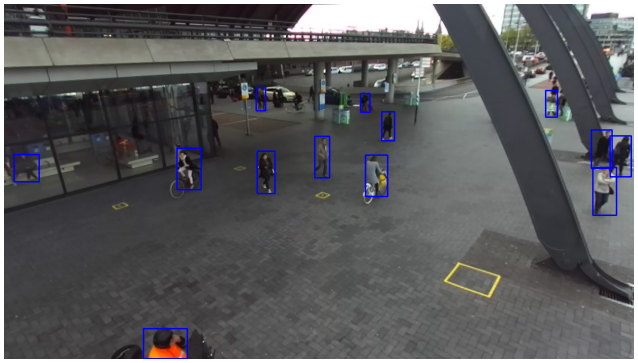
(a) Original camera axes

(b) Transformed axes

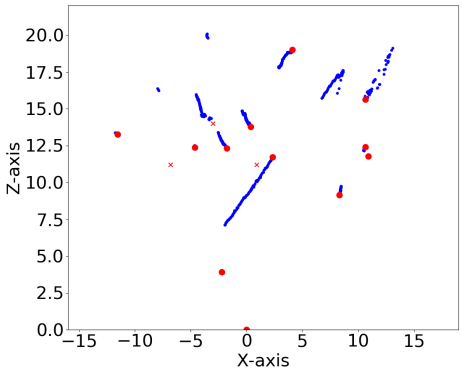
Figure 9.6: 3D-representation and coordinate transformation of Amsterdam scene to obtain horizontal plane

Figure 9.7 represents the detections of the current frame on the visual images and the trails obtained on the ground plane for the 100 frames. Here, the red dot represents the detections on ground plane for the current frame while the smaller blue dots represents the trail of detected agents over the past 100 frames. For each of the graph, the visual image on the left represents the detection boxes on the visual images for the current frame. This is just a

way to visualise the 2D coordinates of agent on the ground plane and it does not affect the data processing framework in any form. For agents standing far from the camera($>$ 20 m), the depth estimations were unstable. This resulted in unstable representation of agents on the ground plane. But when agents were clearly visible, both the detections and the ground plane representation of trajectories were continuous and stable. In cases of good and stable detections(as described in section 9.3), the proposed framework of estimating the agent's 3D location and representing it on the ground plane provided continuous trajectories.

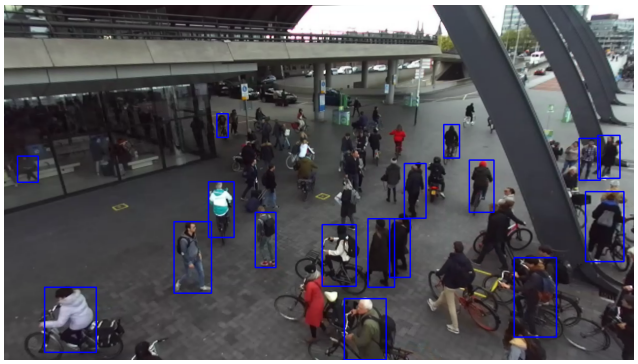


(a1) Detection boxes

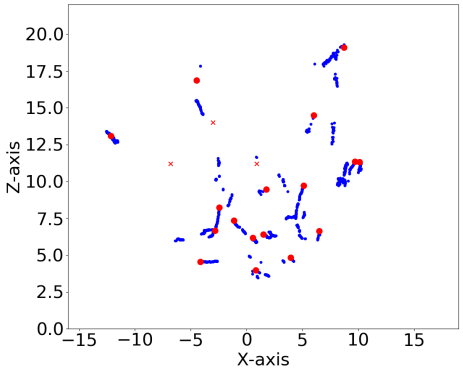


(a2) Detection trails

(a) Normal conditions (low crowding)



(b1) Detection boxes



(b2) Detection trails

(b) Time of ferry arrival (high crowding)

Figure 9.7: Detections with point trails for Amsterdam data

9.5 Agent tracking

For assigning ids to the agent's 2D-points. the tracking model was used with selected parameters (as in section 8.3). The comparison between the results obtained from the two crowding scenarios was done on the basis of number of ids and their lifespans. Compared to the Green Village data, the observation area of the Amsterdam area was larger. So, the agents were in the frame for a larger time period. On average, a pedestrian took 450 frames while a cyclists took 200 frames to cross the observed area. Thus, the threshold values to represent

these results was set at 120 frames, 360 frames and 600 frames. Segregating the outputs based on these threshold values helped to better visualise and understand the distribution of id numbers with there average lifespans.

Graph in figure 9.8 represents the distribution of ids with there average lifespans for both the crowding scenarios. For the low crowding scenario, the number of ids were less but had a longer lifespan due to uninterrupted and continuous detections. In higher crowding conditions, the number of ids generated are much larger but with lower lifespans. This further confirms that quality of input detections does have a direct impact on the working of such tracking-by-detection based models. Figure 9.9 represents some of the trajectories obtained for both cyclists and pedestrians. In Amsterdam, some of the pedestrians were also standing still without moving. The detection model detected such agents on the same location across multiple frames. The tracking model was also able to assign them with a constant id over multiple frames. One such agent is represented with id number 269 in figure 9.9 who was stationary for 750 frames (12.5 seconds). This means that the detection model was able to detect and the tracking model was able to track agents even when they were stationary. In the real-world applications, the value of this framework and its abilities to detect and track people in motion and in stillness depends upon the type space observed.

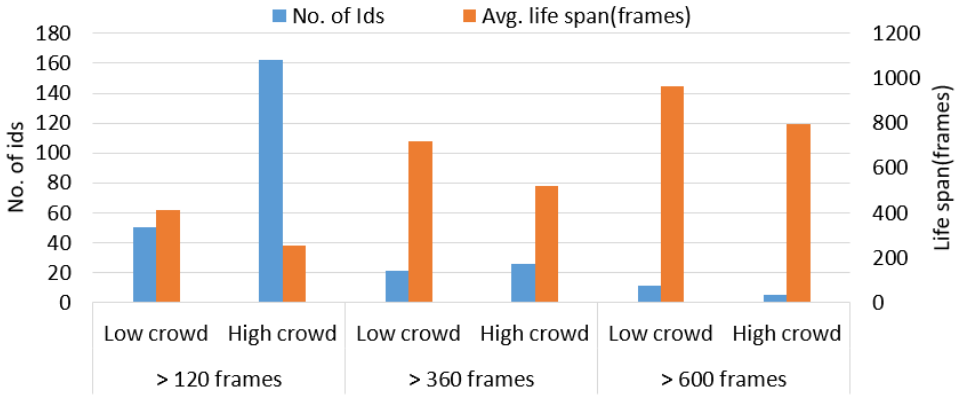


Figure 9.8: Number of Ids with their average lifespans for Amsterdam data

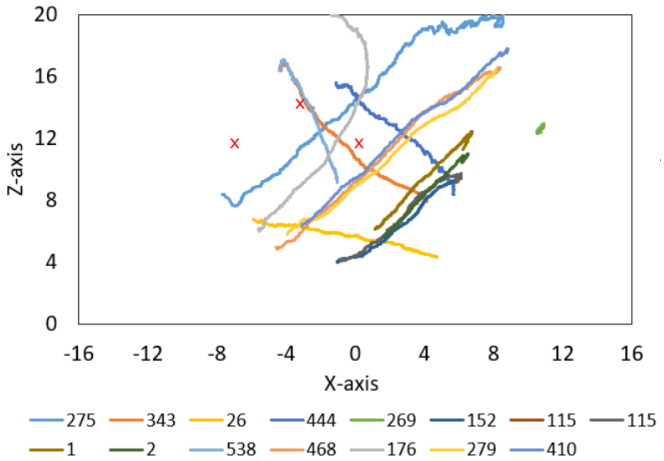


Figure 9.9: Trajectory of agents (pedestrians and cyclists) on the transformed plane

9.6 Discussion

If a comparison was done between the detections observed in figure 9.3a & figure 9.3b, then people further away were easily detected in low crowded scenario but not in case of highly crowded scenario. The model was also struggling to detect nearby objects in case of second scenario. This can be a result of biases developed during the model's training due to a biased training dataset. To elaborate on this issue and to explain the generation of these biases in the detection model, an imaginary example is used. Say the training dataset might have contained images with a range of 10 to 15 people in one image. Thus, when this model will be faced with 30 people in one image, it will just detect 15 people and will miss other detections resulting in false-negative detections. While in cases with fewer people, it might try to make at least 10 detections giving false positive detections. Thus, such detection models are not capable of generalising the detection problem as they are unable to understand that there can be more or less individual people in an image. Such biases can be removed by using a more representative dataset while training the detection model based on the nature of its application.

From figure 9.6 representing the Amsterdam scene on the 3D plane, it can also be seen that as the distance of points from the camera increases, the number of points represented decreases. This reduction in number of observations can also be a factor in reducing the accuracy of depth estimation as reported in the literature [33]. The Zed camera was unable to project the Amsterdam's ground plane as a straight surface. This was a limitation of this camera which can be affected by the observed lighting conditions, nature of the surface observed and the distance of the surface points from the camera. Different camera setting and initial parameters can be tested in the future.

During the implementation of proposed framework on the Amsterdam dataset, the design parameters (such as reference box's size, tracking parameters) were used with similar values as selected in the Green Village data. Due to the smaller size of people in the image and larger area of observation, there are possibilities of tuning these values specific to the Amsterdam dataset. In future, the parameter tuning framework developed and applied on the Green Village dataset can also be used on this dataset.

9.7 Summary and Conclusion

This chapter applies the framework developed in the previous chapters on the Amsterdam data. The dataset was divided into two scenarios, one recorded during the ferry arrival (high crowding) while another recorded without the ferry passengers (low crowding). In this real-world dataset, the agents (both riders and pedestrians) were observed with a large set of varying appearances. Even then, the selected detection model was successfully able to detect and draw bounding boxes around the agents. The stability and fit of the detection boxes were affected by the level of occlusion, total number of visible agents and the object's shadow. The absence of shadows in the Amsterdam dataset largely contributed to the improved fit of the bounding box compared to the Green Village data. In low crowding scenario, occlusion was the main reason for missed detections. In high crowded scenario, frequent occlusion was observed. Here, large number of perfectly visible agents also passed undetected (possibly due to biases in the detection model generated due to biased training dataset). Thus, the performance of this detection model was largely dependent on the occlusion and the total number of people visible but was independent of the density by itself.

During depth extraction, the Zed camera was unable to calculate the depth information of scene at distance greater than 20 meters. Thus, even when the detection were available, the agent's depth could not be extracted. For the ground plane representation of the extracted 3D-point, the measured angle of tilt of 22° was used. The ground plane was projected as a concavely curved plane instead of a straight plane. Hence for larger areas with greater distances, the depth information from the Zed camera was unable to project the ground plane as a straight plane. Thus, the obtained agent coordinates were w.r.t this curved ground plane and not on a straight plane.

During tracking at low densities, the assigned ids were with longer lifespans due to continuous and uninterrupted detections. For brief occlusions, the tracking model was able to perform tracking even across missed detections. At higher crowding levels, the average lifespan was much lower as the input 2D-points were fragmented and unstable. Thus, the model was unable to account for much of the errors passed on by the detection model as they were excessive. The last chapter of conclusion includes a overall discussion (section 10.1) of this framework and elaborates on its limitations and possible improvements.

Chapter 10

Conclusion

This section concludes all the findings of this thesis report by initially discussing the main advantages, challenges and recommendations for the proposed framework. The research questions are answered in section 10.2 followed by the main contributions of this research for the scientific community and society. The last section lays out the future work to further advance the proposed framework.

10.1 Overall discussion and recommendations

Discussion for each of the individual steps during this thesis project has been included in every chapter above. This section links all these different chapters and provides an overall discussion. It mentions the limitation and recommendations for possible improvements upon the proposed framework and the methodology used in this research. It also reiterates some of the important discussion points previously mentioned in the report. A summary of all the discussion points are provided in table 10.1.

Zed camera

Regarding the Zed camera, the software provided by the manufacturer had a good user interface and the online community support was overwhelming. The camera was small, light and easy to carry and install. The advantage of using the Zed camera was that the camera was sensitive to detect small changes in the depth values and it eliminated the need for any on-site camera calibrations. The correction for lens distortion is done using the camera's generic software. However, it needed heavy on-site support it terms of the GPU-enabled computer with a constant power supply. For a smaller depth range (10-12 meters during the Green village experiment), the ground surface was projected as a straight plane. For such distances, the camera also exhibits high sensitivity for even subtle changes in the recorded area. But for larger ranges (15-20 meters in the Amsterdam experiment), the ground plane was projected as a curved plane and not as a straight plane. Thus, the accuracy of camera's depth estimation was dependent upon the distance measured by the camera. It can be improved by further investigating into the camera's depth estimation process and testing different user settings (resolution, frame rate and other parameters). Using multiple Zed cameras to observe the area can also increase the accuracy and range of the camera setup.

Agent detection

The detection model provides one of the main inputs into the proposed framework. It provides the agent's bounding boxes and thus their location with every time-step. The quality of these detections highly determines the output quality of the later processes. The detection models were able to detect agents in real-world conditions even when they were seen with much varied appearances. Two main limitations of this model were (i) occlusion and (ii) higher number of total agents (about $>15 \sim 18$ agents) in the frame. Occlusion makes the information needed for detections scare which can be challenging to resolve using the detection model but can be partially solved by the tracking model as explained later. The limitation of the maximum number of agents present in the frame can be a result of biases generated by the training dataset. This can always be improved by re-training the model on additional datasets consisting of labelled images with a higher number of agents. The ability to train such neural-network based detection models and fit it to a custom dataset can also be seen as one of the main advantages of this framework. Multi-camera setups can observe the scene from different angles and reduce the number of missed detections due to agent occlusion.

Depth extraction and ground representation

While designing the depth extraction process, the sample dataset used during the parameter tuning was minimal (10 frames only). A much larger and more representative sample dataset can be used to obtain better results. The process of tuning should always avoid overfitting as the sample dataset is always limited compared to the overall data size. The reference box's shape and size, and the statistical operation was fixed for all the agents. These operations can also be designed to be a function of agent's location within the image. This might improve the overall depth extraction processes, particularly for agents in the edge frames. For representing agent on the ground plane, the proposed framework needs only the camera's angle of tilt as an on-site measurement for coordinate transformation. The output of depth estimation and ground representation stage was the agent coordinated on the ground plane for every time-step. The main challenge faced this process was the discontinuity of projected points due to missed detections. This challenge can be overcome by improving the detection model. The output of 2D-coordinates with every time-step is the most fundamental data and can be used to calculate many other forms of data such as flows, densities, trajectories, speeds and so on. This research focuses on one of many data processing possibilities which was to obtain agent trajectories on the ground plane.

Agent tracking

In the agent tracking chapter, the model was selected to use only the agent's 2D-points on the ground plane. SORT tracking model was able to track agents by assigning ids to the anonymous 2D-points. Using the id number, all the 2D-points related to the respective id were extracted and plotted to derive the agent's trajectories. In cases when an agent briefly passes undetected by the detection model (mainly due to occlusion), the tracking model was still able to assign ids across those missed detections. Thus, this model was able to account for some of the shortfalls passed on by the detection model. But in cases when detections were not at all continuous and stable (high-crowding), the tracking model failed to provide full and unbroken trajectories. As the detection model itself failed in high crowding scenario, this research was unable to test the tracking model for situations when a large number of closely located points

are inputted into this model continuously. To improve the tracking model, adding other forms of data (like visual information) into the tracking process can enhance its performance and reduce its reliance on the performance of the previous processes. During the parameter tuning process, the performance of different runs was relatively judged. A better tuning framework can be designed using the ground truth trajectories as reference.

Overall approach

While applying the developed framework on the Amsterdam data, parameter values tuned during the design phase were used. There is a possibility to tune these parameters on every custom dataset provided that a reliable ground truth dataset is available. Both Zed camera's depth estimation process and the detection model's agent detection are entirely dependant on the quality of visual information recorded by the Zed camera. Thus, this overall dependence of the proposed framework on the visual data makes its implementation under low-light conditions limited. Each stage of the proposed framework is very much fragmented i.e. there is absence of any feedback loop amongst these individual processes. For example, the detection process is not informed by either the depth data or the tracking algorithm to detect agents in a video. Such frameworks can have its advantages and disadvantages. The current individuality of each process allows future researchers with the flexibility to swap or modify one process without the need to change much of other processes. While integrating these process can provide better performance but at the expense of such modularity. Some of the possibilities of integrating different processes are mentioned in section 10.4 while describing the future work.

Table 10.1: Summary of the main advantages and challenges faced by each stage/process and the recommendations for further improvements.

Stages	Advantages	Challenges	Recommendations
Zed camera	• Small, light, easy to carry and install. • Eliminates the need of on-site camera calibrations. • The camera is sensitive to detect small changes in the depth. • Easy sharing and using of raw data (free software).	• Inaccurate depth estimation for longer ranges. • Heavy on-site support.	• Exploring better ways to use the camera (and its software). • Multi-camera setups for better depth estimation and range.
Agent detection	• Able to detect people in real-world scenarios. • Effective in differentiating between people and other objects (bags, suitcases, bicycles, etc.). • Negligible false positive detection.	• Missed detections due to occlusion of agents. • Missed detections in high crowded situations due to biases. • Sensitive to agent's shadow.	• Multi-camera setups to overcome occlusion. • Re-training the detection model on custom data sets for better performance.
Depth extraction & ground representation	• Reduces on-site measurements (only camera's angle of tilt is needed). • Filters background pixels and errors passed on by the detection process.	• Discontinuous points due to missed detections.	• Making depth extraction process more dynamic and robust. • Improving the detection model for continuous points.
Agent tracking	• Able to overcome missed detections for small time-steps. • Model is purely rule-based and can be extended to track other objects.	• False swaying and vibrations. • Highly dependent on detections.	• Curve smoothening over certain time-steps. • Pass/add more information in the tracking model.

10.2 Answers to research questions

- *Which features of the stereo-vision camera are to be considered for recording the movement of people in shared spaces?*

  The camera features considered while selecting a stereo vision sensor were: (i) sensor's type, (ii) range of depth sensing, (iii) quality of data recording, (iv) software support and (v) market price. The methodology of the sensor to collect depth information of its surrounding divided the depth sensors into two types, active sensors and passive sensors. The distance up to which the camera can sense depth is said to be its depth sensing range. The quality of the recorded data is determined by two factors, its image resolution and its frame rate (frames per second). The software support includes the generic software provided by the manufacturer and the online support available for the selected camera. This feature is particularly important to consider when working with such new technological devices. It determines the ease with which the recorded depth data can be integrated into the data processing framework. Lastly, the camera's buying price was also one of the factors considered during this selection process.

- *Which factors are considered while selecting the location of and designing the data collection experiments?*

  During this research, two data collection experiments were performed, one in a controlled environment while another in a real-world environment. For both the experiments, the Zed camera recorded the data at 720p, 60fps. The factors considered during each of these experiments are explained in the following paragraphs.

  The controlled experiment aimed to test the camera setup for its data collection abilities and to use the collected data to develop the data processing framework. To test the setup, an outdoor location was selected such that the camera was exposed to external factors (heat, sunlight), internal factors (data transfer and storage) and tested the reliability of other supporting devices (mini-computer, data cables, pan-tilt, power supply). As the research was aimed at developing the framework for cyclist and pedestrians, one scenario dedicated to each of these modes was designed. To include the shared-space perspective, a mixed flow scenario involving interactions between both the modes was also designed. The camera's height and the angle of tilt were chosen to mimic the real-world data collection setup.

  For the second data collection, the aim was to collect a real-world recording of shared spaces. This dataset was then used to test the developed framework for real-world use. The location was selected based on three criteria: (i) Shared space consists of mainly pedestrians and cyclist, (ii) multi-directional flow of traffic and (iii) Allow testing for different crowding conditions. The camera setup in terms of its location and the type of setup to be used was largely governed by the nature of permissions provided by the local authorities.

- *From the data recorded using the stereo camera, how can the agents (pedestrians and cyclists) be identified and localised on the ground plane?*

  The proposed data processing framework to localise the agent on the ground plane using the data recorded using the Zed camera was divided into two stages, (i) agent detection, (ii) depth extraction and ground representation. Each paragraph below briefly explains the two processes.

In the agent detection stage, the visual video from the Zed camera is processed using a detection model which was able to locate the agents on every frame of the video. The detection model used in this research was a pre-trained Faster R-CNN model. The model was modified to identify specific object types. It was successfully able to identify and differentiate between a person and a bicycle using the visual images only. Both a pedestrian and a cyclist were categorised as a person. This model was able to locate the object in the image by providing bounding boxes around the detected agent. The output provided by the model consists of the detected agent's image coordinates with their respective frame numbers and the object types. The quality and stability of detections were dependent on the level of occlusion, total number of people in a single frame and people's shadows.

In the depth extraction process, the image coordinates of the agent were used to refer to the depth map calculated by the camera. The depth map consists of 3D-coordinate values (X, Y, Z values) for every pixel of the image w.r.t. the camera's coordinate system. An imaginary reference box of size 40 by 40 pixels was used to refer to the agent's depth information from the depth map. As this provided with a distribution of coordinate values for each agent, a statistical operation was applied to obtain a single coordinate value. Mean, upper quartile and lower quartile value for the X, Y, Z distribution were taken respectively. When designing the framework, the depth estimation for bicycles was unstable and unreliable as they provide less number of pixels for depth estimation. Thus, only people (pedestrians and cyclists) were represented on the ground plane.

As the camera was tilted w.r.t. the horizontal ground surface, the camera's coordinate system was rotated to make one of the planes (X-Z plane) parallel to the ground plane. The resultant coordinate values w.r.t the transformed plane were then used to represent the agent on the ground plane. The angle of rotation was same as the camera's angle of tilt which can be measured during the data collection.

The framework for the above two stages was developed using the Green Village in chapter 6 & 7 respectively. These sections provide a detailed explanation of the design choices and the parameter values chosen for these stages. Sections 9.3 & 9.4 analyse the results obtained from applying this framework on the Amsterdam dataset.

- *Which of the existing object tracking frameworks can be integrated with the extracted agent coordinates to provide trajectories on the ground plane?*

  The trajectories provide the information of path taken by the agent through the observed space. From the previous processes, the agents were already represented on the ground plane for every time-step (as per the video's frame rate). To obtain the trajectories, the tracking model should be able to perform (i) tracking using the agent's 2D-coordinates and, (ii) distinguish and track multiple objects simultaneously. Using these selection criteria, the SORT tracking model was selected and used in this project [2]. This tracking model was originally developed to track the detected objects on an image. But for this research, the model's python code was modified to track agents using the ground plane points.

  The tracking model assigned unique ids to a series of projected points by performing association of individual points throughout the frames. Using the id-number, the series of points can be extracted to plot the trajectory of a single agent. This tracking model was able to provide ids even when the series of points were sometimes discontinuous and thus accounting for the false-negative (or missed) detections. During the process of association, the model also deleted some points which were not being associated with

any valid id-number. This means the model was also able to account for false-positive detections passed on from the previous processes. The parameters of this tracking model were also identified and tuned based on the Green Village dataset in section 8.3.

After answering all the sub research questions above, the answer to the main research question is as follows:

What can be a data gathering and processing framework to automatically extract trajectories of cyclists and pedestrians in a shared space environment using a 3D-stereo vision camera?

The research question focuses on two main research objectives which was to develop a data gathering and a data processing approach. For data gathering, the stereo camera was selected as the Zed camera. Its features were selected considering its further use in observed real-world shared spaces. The two data collections performed during this thesis also allowed the testing of camera's supporting hardware which was important to utilise this sensor at its full potential. During the data collections, the Zed camera was installed in a tilted orientation to allow the development and testing of data processing framework in such real-world situations. For the later data processing stages, only two kinds of data were to be collected on-site, (i) recordings by the Zed camera and (ii) camera's angle of tilt. For validation purposes, certain number of ground measurements can also be collected.

The data processing framework was developed to automate the trajectory extraction using the data collected by the Zed camera. The framework was divided into three stages, (i) agent detection, (ii) depth extraction and ground representation and, (iii) agent tracking. Agent detection uses a neural-network based object detection model. Here, the visual images from the Zed camera are taken as input to output the bounding boxes around the agents for every frame of the video. The depth extraction stage combines the depth data calculated by the camera and the bounding boxes to output 3D-coordinates of the agent. As these coordinates are on tilted camera axes, the ground representation stage rotates the coordinate system to make one of the plane parallel to the ground plane. The agent's coordinate values along this transformed plane is used to represent it on the ground plane. The tracking stage uses these set of ground plane points to assign them with ids automatically. These ids can then be used to extract a set of ground plane points to derive agent trajectories. As overview of the final data processing framework with all the parameter values are provided in section 9.1.

10.3 Main Contributions

This section identifies the key contributions made during this research based on the research gaps identified from the literature review. These contributions are divided into parts, (i) Scientific and (ii) Societal contributions.

10.3.1 Scientific Contributions

This thesis provides a data collection framework by using the Zed stereo-camera and investigates its abilities to record the real-world data from a tilted orientation and then represent the agents on the ground plane. In the process, this thesis also formulated a methodology to derive the Zed camera's angle of tilt which can be used to validate the on-site measurements provided that the depth data recorded is reliable. As this project collects the data first-hand, this collected

data itself is a contribution which can be used for further investigations. Especially, the recordings with agents wearing the red caps can be used to to derive the ground truth and compare the performances of different data extraction methodologies.

While designing the data processing framework, this thesis has integrated various individual processes of data extractions and has tested it in the real-world shared spaces. During the analysis of the results, limitations and biases of the tracking model were identified many of which can be improved to enhance the framework's performance. In case of real-world datasets, lack of automation in the data extraction process was one of the main motivations of this research. This project has investigated automation of data extraction at every stage of its data processing framework and has identified its strengths and weaknesses. Such frameworks focusing on automation and data quality will encourage future researchers to collect and include real-world datasets in their studies.

10.3.2 Societal Contributions

By providing better tools for data collection and automating the data processing, this thesis has contributed towards improving data quality while reducing the efforts for data processing. By limiting (or eliminating) the manual and repetitive tasks performed otherwise during data extractions, the costs of undertaking such detailed studies on gathering data on people's movement certainly reduce. Lower costs will encourage not only academic community by also commercial consulting companies and local governments to perform such studies on people movement. By including such detailed traffic analysis into the design process of public spaces can help improve overall quality and usability of the infrastructure. Moreover, reduced costs of such data-backed analysis can also allow urban planners to understand each space individually and design custom solutions for every scenarios.

10.4 Future Work

This section discusses the possible future directions to advance the work done in this thesis. Some of the recommendations and smaller improvements in the current framework while using the model have been mentioned in previous section 10.1. This sections discusses the research opportunities in the future.

- **Lighter camera setup:** Regarding the data collection setup, the Zed camera itself is very light and easy to carry and install. But, the camera's supporting hardware (mini-computer, power supply) is heavy and needs constant on-site support. Thus, more research is needed in designing the setup to be light, self-sufficient, easy to carry and install on-site. This will help facilitate the use of such data collection setups in the future.
- **Sharing of information between different processes:** The proposed data collection framework is very much top-down as the flow of information is only one-way. In future, integrating different stages and sharing of information between different processes can be investigated. For example, the tracking process can be designed to nudge the detection model to detect agents. It might help reduce the number of missed detections. Similarly, the depth information can also guide the detection model to make better bounding boxes and filter the background pixels.

Different processes can also be integrated into a single process to provide better output quality. For instance, the detection model and the depth information can be integrated to form cuboids enclosing the whole agent in 3D-space. This will truly integrate the two stages and eliminate the need for additional depth extraction process.

- **Multi-camera setups:** Such camera configurations can help overcome the problem of occlusion as the space can be observed from multiple directions and increase the overall depth-sensing range of the setup. Thus, improving the stability, reliability and accuracy of the output data.
- **Mode-based tracking:** Using the ability of the detection model to visually differentiate modes and integrating it with some mode specific movement characteristics (like speed), mode-based detection and tracking can also be explored.
- **Exploring the use of location data:** In this research, the agent's location with every time-step was used to derive agent trajectories. This location data can also be used to derive other information of motion such as densities, speeds, flows and so on.
- **Validation:** This framework is still in its early stages. As this framework matures, an extensive validation study to compare its output with the ground truth can be performed. Before this, a methodology to determine a reliable ground truth data also needs to be developed.
- **Open sourcing the raw data:** Lastly, even though the process of data extraction can be automated and improved, the data collection process remains a manually intensive, costly and cumbersome process. This initial process acts as a threshold and might discourage many researchers from undertaking such projects and advancing this field of computer vision. In future, this threshold can be reduced by making the 3D-dataset publicly available. Moreover, a standardisation framework can be designed to benchmark the performance of different algorithms on the given 3D-dataset. This will allow further advancement into the field of people tracking using 3D-data and help unlock its immense potentials.

Appendix A

Appendix

A.1 Hardware selection

The camera generates a huge amount of raw data while recording which would take up large disk space within short recording periods(0.9TB per hour). To overcome this barrier, a HP Z2 mini workstation with Nvidia Quadro P1000 GPU was used for recording, storing and processing the data. It has 4GB GPU memory while being light and easy to carry on-site during data collection. Other supporting hardware includes a pan tilt and a USB data extension cable. The pan tilt used was Zifon's YT-260 a wireless remote control with a panning angle of 230 degrees. The extension cable used was Lindy's USB 3.0 Active Extension measuring 10 meters. This was to have flexibility with the camera in adjusting its angle and the distance of connections while collecting data.

Table A.1: Technical specifications of Zed camera

Physical characteristics	Dimension: 175x30x33 mm Weight: 159g
Resolution and Frame rate	2x(2208x1242)@15fps 2x(1920x1080)@30fps 2x(1280x720)@60fps 2x(640x480)@100fps
Depth	Range: 1m to 20m Baseline: 120mm
System requirements	Windows or Linux USB 3.0 $>$ 4GB RAM, Nvidia GPU

A.2 Calculation for Green Village data

The distances in table A.2 are calculated using the Euclidean distance formula.

Table A.2: Calculated coordinate values for agent in Bi-directional, less-crowded scenario (in meters) (section 7.1) (Frame 175: left edge frame, Frame 422: centre frame, Frame 590: right edge frame)

Frame no.	**100**	**175**	**245**	**292**	**358**	**422**	**490**	**521**	**590**	**630**
X_g	-5.40	-4.00	-2.70	-1.50	0.00	1.00	2.30	3.30	4.60	5.50
Y_g	5.09	5.09	5.09	5.09	5.09	5.09	5.09	5.09	5.09	5.09
Z_g	6.30	6.20	6.15	6.10	6.10	6.10	6.10	6.00	6.00	6.10
Distance (from camera)	9.73	8.96	8.43	8.09	7.94	8.01	8.27	8.53	9.11	9.66
Ground distance	8.30	7.38	6.72	6.28	6.10	6.18	6.52	6.85	7.56	8.21

Table A.3: Camera's distance estimation (in meters) and the difference in distances (ΔD(%)) from the reference distances (section 7.2)

Frame no.		**100**	**175**	**245**	**292**	**358**	**422**	**490**	**521**	**590**	**630**
Camera's distance estimation	**R = 5**	9.55	8.60	8.29	7.61	7.68	7.67	8.12	8.43	8.53	8.96
	R = 10	9.32	8.56	8.11	7.63	7.62	7.64	8.10	8.40	8.58	8.98
	R = 15	9.29	8.54	8.07	7.60	7.67	7.64	8.10	8.26	8.61	9.00
	R = 20	9.38	8.48	8.12	7.76	7.73	7.74	8.19	8.39	8.66	9.06
	R = 25	9.33	8.56	8.13	7.82	7.86	7.80	8.23	8.46	8.66	8.85
	R = 30	9.49	8.54	8.21	7.83	7.79	7.81	8.29	8.50	8.75	8.75
ΔD(%)	**R = 5**	1.89	4.06	1.64	5.82	3.37	4.23	1.79	1.25	6.43	7.26
	R = 10	4.22	4.51	3.78	5.63	4.13	4.63	2.12	1.59	5.85	7.08
	R = 15	4.53	4.68	4.20	5.95	3.50	4.58	2.05	3.14	5.53	6.82
	R = 20	**3.63**	**5.36**	**3.60**	**4.04**	**2.64**	**3.31**	**0.97**	**1.69**	**4.98**	**6.22**
	R = 25	4.20	4.55	3.50	3.25	1.09	2.60	0.45	0.85	5.02	8.45
	R = 30	2.48	4.72	2.55	3.19	1.93	2.53	0.20	0.35	4.05	9.49
	Average ΔD(%)	3.49	4.64	3.21	4.64	2.78	3.65	1.26	1.48	5.31	7.56

Table A.4: Calculation for unknown angles of tilt for ground plane representation of 3D coordinates (section 7.3)

Frames		**100**	**175**	**245**	**292**	**358**	**422**	**490**	**521**	**590**	**630**
Coordinates w.r.t. camera axes	**x_c**	-5.72	-4.27	-2.97	-1.92	-0.52	0.62	2.07	2.72	3.97	4.67
	y_c	-1.67	-1.87	-2.12	-2.27	-2.37	-2.52	-2.82	-2.72	-2.77	-2.92
	z_c	7.11	7.00	7.24	7.16	7.32	7.28	7.41	7.42	7.11	7.14
Y-Z rotation (θ=55.652)	z_g	5.40	5.50	5.84	5.92	6.10	6.20	6.51	6.44	6.31	6.44
	y′	4.93	4.73	4.79	4.63	4.71	4.59	4.53	4.59	4.31	4.25
X-Y rotation (Φ=-5.319)	x_g	-5.24	-3.82	-2.52	-1.49	-0.09	1.05	2.49	3.14	4.36	5.05
	y_g	5.44	5.10	5.04	4.79	4.74	4.51	4.31	4.32	3.93	3.79

A.3 Output of detection model, depth extraction and ground representation

Table A.5: Sample output for Green Village dataset (mixed flow scenario) with frame number (column 1), bounding box coordinates (column 2-5), prediction score (column 6), object class (column 7), 3D coordinates (column 8-10)

framenr	**ymin**	**xmin**	**ymax**	**xmax**	**score**	**class**	**x**	**y**	**z**
1	0.397363	0.520452	0.585258	0.569599	0.995047	1	0.989893	4.559073	3.11012
1	0.172934	0.407298	0.392473	0.45757	0.991616	1	-0.27639	4.73721	4.972407
1	0.555708	0.375584	0.823067	0.467958	0.986595	1	-0.21559	5.039148	1.976348
:	:	:	:	:	:	:	:	:	:
2	0.395719	0.52064	0.585019	0.569912	0.992535	1	0.988496	4.525624	3.10555
2	0.150732	0.404539	0.357152	0.453859	0.981254	1	-0.3378	4.720756	5.203262
2	0.190645	0.297931	0.391355	0.356465	0.980762	1	-1.569	4.924088	4.967777
:	:	:	:	:	:	:	:	:	:
3	0.395678	0.520655	0.584946	0.56991	0.993081	1	0.991506	4.539735	3.115287
3	0.153027	0.404484	0.377377	0.453528	0.984878	1	-0.33663	4.734605	5.148144
3	0.147485	0.35858	0.324592	0.408581	0.981243	1	-0.89053	4.628875	5.243807

A.4 Input and output for the tracking model

Table A.6: Sample input into tracking model for Green Village dataset (mixed flow scenario) with frame number (column 1), id number (column 2, -1 is garbage value), ground coordinates (column 3-4)

framenr	Id	x	z
1	-1	0.989893	3.11012
1	-1	-0.27639	4.972407
1	-1	-0.21559	1.976348
:	:	:	:
2	-1	0.988496	3.10555
2	-1	-0.3378	5.203262
2	-1	-1.569	4.967777
:	:	:	:
3	-1	0.991506	3.115287
3	-1	-0.33663	5.148144
3	-1	-0.89053	5.243807

Table A.7: Sample output of tracking model for Green Village dataset (mixed flow scenario) with frame number (column 1), id number (column 2), ground coordinates (column 3-4)

framenr	Id	x	z
1	1	0.989893	3.11012
1	2	-0.27639	4.972407
1	3	-0.21559	1.976348
:	:	:	:
2	1	0.988496	3.10555
2	2	-0.3378	5.203262
2	3	-1.569	4.967777
:	:	:	:
3	1	0.991506	3.115287
3	2	-0.33663	5.148144
3	3	-0.89053	5.243807

Bibliography

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[2] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *Proceedings - International Conference on Image Processing, ICIP*, vol. 2016-Augus, pp. 3464–3468, 2016.

[3] P. R., *Theory of Stereo vision system*. Available at `http://prod.optoiq.com/content/dam/VSD/NextGen/5-3D-2.pdf`, [Online; accessed 5-March-2019].

[4] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou, "Multi-scale object detection in remote sensing imagery with convolutional neural networks," *ISPRS journal of photogrammetry and remote sensing*, vol. 145, pp. 3–22, 2018.

[5] Amsterdam Economic Board, "Feiten en cijfers: De toekomst van mobiliteit." `https://www.amsterdameconomicboard.com/nieuws/feiten-en-cijfers-toekomst-mobiliteit`, 2018. Accessed: 2019-02-25.

[6] World Health Organisation, *GLOBAL STATUS REPORT ON ROAD SAFETY 2018*. Available at `https://www.who.int/violence_injury_prevention/road_safety_status/2018/GSRRS2018_Summary_EN.pdf`.

[7] D. Beitel, J. Stipancic, K. Manaugh, and L. Miranda-Moreno, "Assessing safety of shared space using cyclist-pedestrian interactions and automated video conflict analysis," *Transportation Research Part D: Transport and Environment*, vol. 65, pp. 710–724, 2018.

[8] B. Hamilton-Baillie, "Shared space: Reconciling people, places and traffic," *Built environment*, vol. 34, no. 2, pp. 161–181, 2008.

[9] Department of Transport, "Local transport note 1/11 shared space." `https://nacto.org/docs/usdg/shared_space_qualitative_research_dickens.pdf`, 2011.

[10] B. Anvari, M. G. Bell, A. Sivakumar, and W. Y. Ochieng, "Modelling shared space users via rule-based social force model," *Transportation Research Part C: Emerging Technologies*, vol. 51, pp. 83–103, 2015.

[11] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[12] D. Helbing, R. Jiang, and M. Treiber, "Analytical investigation of oscillations in intersecting flows of pedestrian and vehicle traffic," *Physical Review E*, vol. 72, no. 4, p. 046130, 2005.

[13] J. Jiang and W. Wong, "Fundamentals of common computer vision techniques for textile quality control," in *Applications of Computer Vision in Fashion and Textiles*, pp. 3–15, Elsevier, 2018.

[14] Y. Yuan, W. Daamen, B. Goñi-Ros, and S. Hoogendoorn, "Investigating cyclist interaction behavior through a controlled laboratory experiment," *Journal of Transport and Land Use*, vol. 11, no. 1, 2018.

[15] W. Daamen and S. Hoogendoorn, "Capacity of doors during evacuation conditions," *Procedia Engineering*, vol. 3, pp. 53–66, 2010.

[16] S. Zangenehpour, L. F. Miranda-Moreno, and N. Saunier, "Automated classification based on video data at intersections with heavy pedestrian and bicycle traffic: Methodology and application," *Transportation research part C: emerging technologies*, vol. 56, pp. 161–176, 2015.

[17] I. Kaparias, M. G. Bell, J. Greensted, S. Cheng, A. Miri, C. Taylor, and B. Mount, "Development and implementation of a vehicle–pedestrian conflict analysis method: adaptation of a vehicle–vehicle technique," *Transportation research record*, vol. 2198, no. 1, pp. 75–82, 2010.

[18] T. Sayed, M. H. Zaki, and J. Autey, "Automated safety diagnosis of vehicle–bicycle interactions using computer vision analysis," *Safety science*, vol. 59, pp. 163–172, 2013.

[19] 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.

[20] 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.

[21] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated person tracking using stereo, color, and pattern detection," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 175–185, 2000.

[22] M. Bertozzi, E. Binelli, A. Broggi, and M. Rose, "Stereo vision-based approaches for pedestrian detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, pp. 16–16, IEEE, 2005.

[23] D. M. Gavrila, "The visual analysis of human movement: A survey," *Computer vision and image understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[24] R. Jiang and Q.-S. Wu, "Interaction between vehicle and pedestrians in a narrow channel," *Physica A: Statistical Mechanics and its Applications*, vol. 368, no. 1, pp. 239–246, 2006.

[25] Intel, *Intel's RealSense camera*. Available at `https://www.intelrealsense.com/depth-camera-d435i/`, [Online; accessed 5-March-2019].

[26] Orbbec, *Orbbec's Astra pro*. Available at `https://orbbec3d.com/product-astra-pro/`, [Online; accessed 5-March-2019].

[27] Asus, *Asus's Xtion pro*. Available at `https://www.asus.com/3D-Sensor/Xtion_PRO/specifications/`, [Online; accessed 5-March-2019].

[28] Stereolabs, *Stereolabs Zed camera*. Available at `https://www.stereolabs.com/zed/`, [Online; accessed 5-March-2019].

[29] Flir, *Flir's Bumblebee-2*. Available at `https://www.flir.eu/support/products/bumblebee2-firewire#Specifications`, [Online; accessed 5-March-2019].

[30] Carnegie robotics, *Carnegie robotic's Multisense S7*. Available at `https://carnegierobotics.com/multisense-s7`, [Online; accessed 5-March-2019].

[31] D. Beltran and L. Basañez, "A comparison between active and passive 3d vision sensors: Bumblebeexb3 and microsoft kinect," in *Robot2013: First iberian robotics conference*, pp. 725–734, Springer, 2014.

[32] A. Deris, I. Trigonis, A. Aravanis, and E. Stathopoulou, "Depth cameras on uavs: A first approach," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 231, 2017.

[33] L. E. Ortiz, E. V. Cabrera, and L. M. Gonçalves, "Depth data error modeling of the zed 3d vision sensor from stereolabs," *ELCVIA: electronic letters on computer vision and image analysis*, vol. 17, no. 1, pp. 0001–15, 2018.

[34] C. Q. Lai and S. S. Teoh, "A review on pedestrian detection techniques based on histogram of oriented gradient feature," in *2014 IEEE Student Conference on Research and Development*, pp. 1–6, IEEE, 2014.

[35] S. P. Hoogendoorn, W. Daamen, and P. H. Bovy, "Extracting microscopic pedestrian characteristics from video data," in *Transportation Research Board Annual Meeting*, vol. 9, pp. 1–15, Citeseer, 2003.

[36] T.-C. Lee, *An agent-based model to simulate motorcycle behaviour in mixed traffic flow*. PhD thesis, Imperial College London (University of London), 2007.

[37] M. Chraibi, U. Kemloh, A. Schadschneider, and A. Seyfried, "Force-based models of pedestrian dynamics," *Networks & Heterogeneous Media*, vol. 6, no. 3, p. 425, 2011.

[38] S. Saadat and K. Teknomo, "Automation of pedestrian tracking in a crowded situation," in *Pedestrian and Evacuation Dynamics*, pp. 231–239, Springer, 2011.

[39] P. Viola, M. Jones, *et al.*, "Rapid object detection using a boosted cascade of simple features," *CVPR (1)*, vol. 1, no. 511-518, p. 3, 2001.

[40] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *international Conference on computer vision & Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893, IEEE Computer Society, 2005.

[41] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 12, pp. 2179–2195, 2008.

[42] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 1239–1258, 2009.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[44] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17–33, 2018.

[45] M. P. Arakeri *et al.*, "Computer vision based fruit grading system for quality evaluation of tomato in agriculture industry," *Procedia Computer Science*, vol. 79, pp. 426–433, 2016.

[46] S. R. Debats, D. Luo, L. D. Estes, T. J. Fuchs, and K. K. Caylor, "A generalized computer vision approach to mapping crop fields in heterogeneous agricultural landscapes," *Remote Sensing of Environment*, vol. 179, pp. 210–221, 2016.

[47] N. Rieke, F. Tombari, and N. Navab, "Computer vision and machine learning for surgical instrument tracking: Focus: Random forest-based microsurgical tool tracking," in *Computer Vision for Assistive Healthcare*, pp. 105–126, Elsevier, 2018.

[48] H. Luo, C. Xiong, W. Fang, P. E. Love, B. Zhang, and X. Ouyang, "Convolutional neural networks: Computer vision-based workforce activity assessment in construction," *Automation in Construction*, vol. 94, pp. 282–289, 2018.

[49] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 3296–3305, 2017.

[50] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, Springer Verlag, 2016.

[51] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "(YOLO) You Only Look Once," *Cvpr*, 2016.

[52] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[53] Google Inc., "Tensorflow detection model zoo," 2015. Available at `https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md`, [Online; accessed 10-August-2019].

[54] "The Complete Beginners Guide to Deep Learning: Convolutional Neural Networks and Image Classification." Available at `https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb`, [Online; accessed 10-September-2019].

[55] J. Hui, "Object detection: speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3)," 2018. Available at `https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656` [Online; accessed 10-August-2019].

[56] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim, "Multiple object tracking: A literature review," *arXiv preprint arXiv:1409.7618*, 2014.

[57] C. Dicle, O. I. Camps, and M. Sznaier, "The way they move: Tracking multiple targets with similar appearance," in *Proceedings of the IEEE international conference on computer vision*, pp. 2304–2311, 2013.

[58] S. Sun, N. Akhtar, H. Song, A. S. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[59] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE international conference on computer vision*, pp. 4705–4713, 2015.

[60] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, IEEE, 2017.

[61] H. Li, Y. Liu, C. Wang, S. Zhang, and X. Cui, "Tracking algorithm of multiple pedestrians based on particle filters in video sequences," *Computational intelligence and neuroscience*, vol. 2016, p. 13, 2016.

[62] D. C. Duives, W. Daamen, and S. Hoogendoorn, "Trajectory analysis of pedestrian crowd movements at a dutch music festival," in *Pedestrian and Evacuation Dynamics 2012*, pp. 151–166, Springer, 2014.

[63] J. C. Redmon, "YOLO: Real-Time Object Detection." Available at `https://pjreddie.com/darknet/yolo/`, [Online; accessed 10-June-2019].

[64] Stereolabs, "Stereolabs/Zed-Tensorflow," Feb. 2019. Available at `https://github.com/stereolabs/zed-tensorflow`.

[65] A. Alexey, "Stereolabs/Zed-YOLO," Dec. 2018. Available at `https://github.com/stereolabs/zed-yolo/tree/master/zed_python_sample`.

[66] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[67] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 1–9, 2015.

[68] "Stanford University CS231n: Convolutional Neural Networks for Visual Recognition." Available at `http://cs231n.stanford.edu/`,[Online;accessed 10-September-2019].

[69] "A deeper look at how Faster-RCNN works - Subrata Goswami - Medium." Available at `https://medium.com/@whatdhack/a-deeper-look-at-how-faster-rcnn-works-84081284e1cd`, [Online; accessed 10-September-2019].

[70] "Faster R-CNN: Down the rabbit hole of modern object detection — Tryolabs Blog." Available at `https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/`, [Online; accessed 10-September-2019].

[71] Alex Bewley, *Simple online real-time tracking (SORT)*. Available at `https://github.com/abewley/sort`.