AUTOMATED MULTIVIEW SAFETY ANALYSIS AT COMPLEX ROAD INTERSECTIONS

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Supervisors

Prof. Reinhard Klette

Assosciate Prof. Wei Qi Yan

Dr Hsiang-Jen (Johnny) Chien

31 January 2020

By

Zahra Moayed

School of Engineering, Computer and Mathematical Sciences

Abstract

Keywords*: Multiview, calibration, traffic safety, intersection, deep learning*

The safety of pedestrians and vehicles at traffic intersections is a major concern for transport practitioners these days due to the high number of reported accidents and fatalities. Computer vision as an essential part involved in intelligent transport systems that can take advantage of infrastructure-based recordings, such as surveillance cameras to assess events and analyse participants' safety.

Previous studies have revealed that the safety factors are investigated solely, and there is a demand to have an automated safety analyser, which considers the interaction among all participants at intersections. Due to variations in traffic scenes in terms of weather conditions and time of day, further research is still needed to achieve robustness. Most monitoring systems are designed to work in controlled environments, so the analysis might not be a good sample of real traffic intersections. Furthermore, the restricted camera view leads to having an incomplete analysis.

In order to resolve these issues, an automatic vision-based system is proposed that is used to understand traffic patterns and to analyse participants' safety at intersections. The major novelty of this thesis is to present a robust safety analyser using four calibrated cameras at a real intersection. Understanding object locations in world coordinates from different cameras helps to improve the tracking and to address the occlusion problem. Also, it yields a larger field of view for covering more areas.

To inspect safety, the characteristics of the participants are extracted; detection,

classification, and tracking use a fusion of appearance-based and motion-based methods. Deep learning proves its ability to take part at this stage, handling tradeoffs among accuracy, time sufficiency, and robustness, while being associated with motion parameters.

The study is further extended to consider the past and future movements, together with safety measurements and interaction risk factors to analyse the potential risks for each participant in the form of a single value. As a result of this study, some attributes and distributions can be extracted for deriving an understanding of the road intersection for further design and planning to mitigate traffic risks.

This research will provide a more cost-effective and reliable approach for informing participants about possible risks from the infrastructure side due to the low cost of the cameras, hence it also aligns with future technologies such as autonomous vehicles.

Regarding showing effectiveness and robustness in practice, a busy road intersection at Auckland, New Zealand, is selected as the ultimate goal for monitoring.

Contents

List of Tables

List of Figures

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Signature of candidate

Publications

Al-Sarayreh, M., Moayed, Z., Bollard-Breen, B., Ramond, J., & Klette, R. (2016). Detection and spatial analysis of fairy circles. 2016 International Conference on Image and Vision Computing New Zealand (IVCNZ), 1-6.

Moayed, Z., Griffin, A., & Klette, R. (2017). Traffic intersection monitoring using fusion of GMM-based deep learning classification and geometric warping. 2017 International Conference on Image and Vision Computing New Zealand (IVCNZ), 1-5.

Sabokrou, M., Fathy, M., Moayed, Z., & Klette, R. (2017). Fast and accurate detection and localization of abnormal behavior in crowded scenes. Machine Vision and Applications, 28(8), 965-985.

Sabokrou, M., Fayyaz, M., Fathy, M., Moayed, Z., & Klette, R. (2018). Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. Computer Vision and Image Understanding, 172, 88-97.

Zhu, Y., Moayed, Z., Bollard-Breen, B., Doshi, A., Ramond, J., & Klette, R. (2018). Detection of fairy circles in UAV images using deep learning. 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 1-6.

Moayed, Z., Chien, H., Zhang, D., & Klette, R. (2019). Surveillance-based collision-time analysis of road-crossing pedestrians. 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2361-2366.

Chien, H., Moayed, Z., Zhu, Y., Zhang, D., & Klette, R. (2019). On improving bounding box regression towards accurate object detection and tracking. 2019 International Conference on Image and Vision Computing New Zealand (IVCNZ), 1-5, 1-8.

Moayed, Z., Chien, H., Zhang, D., & Klette, R. (2020). On Resolving Occlusion Problem Using Improved Multi-View Object Tracking. 2020 International Journal of Advances in Information Technology (JAIT), 11(1), 1-8.

Acknowledgements

I would like to express my sincere appreciation to many people, who so generously contributed to the work presented in this thesis.
Special mention goes to my enthusiastic supervisor, late Prof. Reinhard Klette. My PhD had been a fantastic experience, and I thank him, not only for his tremendous academic support but also for giving me so many wonderful opportunities. I appreciate his supports especially during his difficult time while being under medical care. Unfortunately, this thesis was defended while he had passed away and I would like to commemorate him for his supports.
I would like to thank my third supervisor, Dr Johnny Chien who is also my friend and my colleague, for his endless supports in all the challenging stages of this Journey; his knowledge in the computer vision and Multiview cameras have deeply inspired me.
I am incredibly grateful to my colleagues at Auckland Transport. When I started to work at AT, I was just a second-year PhD student and surely, without the support of my managers, especially Dr Derek Zhang and Ginny Nayler, I had to choose between studying and working. I always appreciate and never forget their full supports.
My sincere thanks also go to my parents for supporting me spiritually throughout writing this thesis.
Finally, but by no means least, thanks go to my beloved husband, Mahdi for the almost unbelievable support. Mahdi has been extremely supportive of me throughout this entire process and has made countless sacrifices to help me get to this point.

Zahra Moayed
Auckland
25 June 2020

Dedication

I am dedicating this thesis to my beloved husband who has meant and continue to mean so much to me.

Chapter 1

Introduction

1.1 Abstract

This introductory chapter informs about reasons why computer vision-based safety analysis in the transport industry is becoming an important topic these days and how the research reported in this thesis contributes to this issue. Also, the contributions of the thesis and the thesis structure are discussed.

1.2 Computer Vision for Traffic Safety

Nowadays, with extensive developments in the transport industry, from smart cities to autonomous vehicles, safety becomes an advanced challenge for governments and transport practitioners. Smart cities and how to deal with safety-related issues are crucial since the number of incidents increases while people may still fail to get along with the new technologies. During the proposal time of smart cities, safety was mainly ignored when people talked about *intelligent transport systems* (ITS),while smart cities should tend to provide a safe and sustainable environment for all users (Lacinák & Ristvej, 2017). Thus, in recent years, safety has become more and more important

in ITS research, and every attempt to build a smart city includes safety studies and considerations.

Future traffic systems will include various types of communication such *as vehicle to vehicle* (V2V), *vehicle to infrastructure* (V2I), or *vehicle to roadside* (V2R). Vehicles will increasingly be able to drive autonomously, and these communications will support road safety. Video surveillance at road intersections would be an important component for creating data to be communicated to vehicles. In particular, it is of interest not only to track the traffic participants (cars, trucks, pedestrians, bicycles, etc.) at a road intersection but also to understand the traffic flow with respect to potential future risks. Pedestrians' behaviour is an essential element of understanding traffic flow.

This research aims at investigating three key questions:

1. How will deep learning contribute to the analysis of traffic safety factors for road intersection monitoring, considering a trade-off between accuracy and time?

2. What will be the contribution of calibrated multiview tracking in traffic safety?

3. Which safety parameters can be extracted in an automatic manner and used, especially, for dynamic objects (i.e. which safety parameters can be communicated to traffic participants for informing about potential risks)?

1.3 Motivation

The focus of this research is to provide safety-related factors to individual traffic users in a robust system based on automation. However, safety analysis at roads intersections deals with more challenges than the normal road such as possible interactions or crowd motion by road users.

The safety of road participants, including drivers and pedestrians, has been of high concern for transport practitioners in recent years. According to traffic studies in the

U.S.A., in 2002, 50% of reported crashes and 22% of total fatalities were intersection-related (Salim, Loke, Rakotonirainy, Srinivasan & Krishnaswamy, 2007). The New Zealand Transport Agency also revealed an upward trend in the number of deaths and injuries between 2013 and 2017 (New Zealand Transport Agency, 2019). Auckland Transport's 10 year target plan is to reduce this number by 60% (Auckland Transport, 2019).

To achieve this, the behavioural analysis of traffic participants plays a critical role and should be investigated for further decisions of traffic organisations.

Besides, the increasing level of technologies in automated vehicles is expected to make dramatic impacts on transportation. Automated vehicles are highly dependent on receiving information from their environment based on the fact that they can perform the driving tasks without or with the least help from humans. One of the major factors that transportation sectors consider toward automated vehicles is the possibility of various human factors and behaviours that lead to fatal crashes; 94% of the crashes in the U.S.A. was related to wrong human choices and behaviours in 2015 (The U.S. Department of Transportation's Federal Automated Vehicles Policy, 2016).

Thus, this project aligns with the increasing trend of automated vehicles in the world. New Zealand is a country where government encourages and financially supports conducting research and testing of these kinds of modern vehicles (New Zealand Ministry of Transport, 2016).

Intelligent transport systems (ITS) is the term used by transport researchers and sectors which define the way information, data processing, communication and sensor technologies are utilised to provide benefits and safety to transport users and infrastructure (Taylor, 2001; New Zealand Ministry of Transport, 2014). ITS tends to resolve the social issues caused by modern changes in transportation areas (Hanai, 2013). Computer vision, as a major part of artificial intelligence, plays an essential role in the development of intelligent transport systems, thanks to the increasing number of

infrastructure-based recording devices such as surveillance cameras (Buch, Velastin & Orwell, 2011). Among other data collection methods, such as manual observation and sensors including LiDAR and radar, surveillance cameras are the best solution in terms of accessibility and low costs of installation, especially by using existing ones to assess the behaviours of traffic participants and analyse their safety.

The goal of this research is to detect and analyse the behaviour and safety factors of different participants at a complex traffic intersection based on recorded video data from multiple camera. The reason for choosing intersections in urban areas as the research target is that traffic analysis has to deal here with many challenges. In these fields, we have a large and dynamic variety of road users, including pedestrians, bikes, and different types of vehicles like trucks and buses, which are more difficult to analyse compared to highways. In addition, a high number of accidents at intersections show the significant impacts of this research project on ITS.

To analyse safety, first we need to determine the parameters involved in traffic safety. Detection of pedestrians' movements is already investigated in existing research. However, considering pedestrians' behaviour at intersections as one of the major contributing factors that influence safe decisions of drivers is still a challenging field to become a novel topic in computer vision.

Despite extensive research achievements on monitoring busy intersections, current systems face many problems including time inefficiency and inaccuracy, therefore the safety at intersections still needs to be strengthened for all traffic participants (Shirazi & Morris, 2017). To overcome these weaknesses of existing approaches, deep learning is used in this research. Deep learning frameworks prove their ability in terms of high accuracy, though the time efficiency still is a concern.

Furthermore, the existing frameworks do not address the safety of different traffic users in a single system as opposed to each other. As the state-of-the-art research shows, computer vision is used to address a single type of user.

1.4 Open Problems and Contributions

The main objective of this research is to propose a robust vision-based safety analysis system that can automatically detect, classify, and analyse the safety of individual traffic users while providing useful information for intersection design and planning.

As far as the study revealed by the time of writing this thesis, the computer vision-based monitoring tools for intersections provide separate data analysis for different user types. In this research, the type of user is a contributing factor to safety risk parameters.

The contributions of this research are as follows:

i. The proposed safety analyser for intersection monitoring benefits from accurate detection and tracking due to improving *convolutional neural network* (CNN) detectors while considering the time efficiency.

ii. Existing systems do not focus on a combination of safety factors of the individual participant and interaction factors imposed by surrounding users. However, this research aims at considering both to generate one value output for each participant which is called *risk factor*.

iii. To implement and test the robustness of the proposed system, data to be used in training is captured in different environmental conditions. However, existing methods typically work for a limited time or for a particular test bed.

iv. Multiview analysis using four calibrated cameras at a real intersection leads to a practical analysis of actual safety data. Connecting four cameras leads to a better understanding of an intersection due to the major extension of the field of view

v. The point correspondence analysis proposed in this research reduces the endeavour of searching the entire line to a small region on the line and using collaborative track-first approach, the association of multiview objects are established.

vi. The proposed solution is customised to be used at Auckland intersections and can be extended to follow the New Zealand road code .

1.5 Structure of Thesis

The outline of this thesis is as follows: In Chapter 2, the existing literature and related work are reviewed.

Chapter 3 discusses multiview road intersection recording and analysis. In this chapter, the relationship between intersection geometry and camera locations is explained.

The proposed methods on how to detect, classify and track the objects are investigated in Chapter 4. The improvement of tracking methods along with the proposed and used multiview approach to address the occlusion problem is also expressed in this chapter.

Chapter 5 discusses the extraction and analysis of safety parameters to achieve the extraction of risk factors. Due to many different steps and algorithms in this research, and to simplify the review, the results for each part are provided after describing the method in the same section.

Chapter 6 concludes the thesis.

Chapter 2

Literature Review

2.1 Abstract

The proposed research aims at providing a complete package that analyzes the safety behaviours of different participants at traffic intersections considering the approaches of computer vision. For concluding the research gaps, the existing literature is reviewed to ensure the high accuracy of the proposed research work. In order to analyze the related state-of-the-art methods, this section is categorized into four subsections: Section 2.2 discusses recent work in moving object detection, and tracking will be briefly discussed in Section 2.3. The state-of-the-art methods which involves multiview object detection and tracking are mentioned in Section 2.4. In Section 2.5, the approaches in analyzing the safety of traffic participants are explained and Section 2.6 discusses the existing gap in the literature.

2.2 Single-view Object Detection

Video-based object detection is utilized in a lot of industries not limited to video surveillance, vision, and robotics. In particular, object detection in videos, sometimes

referred as motion segmentation, is performed mainly by using motion and appearance of the objects, targeting the spatial and temporal transformations in the sequence of video frames. Detecting moving objects provides a clear focus on further analysis due to restricting the region of interest (Weiming Hu, Tieniu Tan, Liang Wang & Maybank, 2004). However, in some applications, static objects are considered to be the main target. There are mainly two general types of object detection and tracking approaches: model-based methods and Data Driven-based Methods. The latter consists of data-driven methods which mainly focus on deep learning approaches.

2.2.1 Model-based Methods

Model-based object detection methods can be categorized broadly into two main approaches: motion-based methods and appearance-based methods.

Motion-based Methods

In traffic analysis, motion detection is of interest because the movement of each traffic participant leads to the safety analysis. In particular, motion-based methods detect the objects by separating the moving areas from the static background. Here, motion-based methods are classified into background subtraction and optical flows. In this section, only the most important approaches to detect traffic-related objects from the static cameras which are still used by researchers are explained.

Background subtraction

Background subtraction consists of several approaches that all aim at estimating the background image from the current video frames. These methods work well in the situation that videos are captured using static cameras. On the other hand, they are extremely sensitive to the environmental changes such as shadows and lighting. For the purpose of clarification of different methods, we here assume that whenever the

background is estimated, the foreground objects are also detected as outliers of the background. There are several approaches that differ in the detection of moving objects, resulting in different levels of image quality and computational complexity.

Frame differencing is the simplest yet the most common approach for motion segmentation. It detects moving objects in a pixel-by-pixel manner by comparing the values of those pixels between two consecutive frames. This value then goes through the process of thresholding in order to detect the foreground mask. This approach is fast, yet the drawbacks are its inability to handle noise, abrupt changes, and periodic movements.

The method proposed in (Abdelli & Ho-Jin Choi, 2017) improves the idea of frame differencing by considering a pixel-level algorithm on four frames to detect objects in wide area surveillance.

In the averaging methods, a period of frames is summed up. A weight specifies the relations between the new frames and the averaged background to detect the moving objects. This approach is considered cost effective, while it creates a sign of movement in periodic frames. To improve the robustness of averaging, a single Gaussian pixel distribution can be used. In this method, pixels' variance over the sequence of the frame is also calculated. As a result, a variance image and mean image are generated. The pixels on a new frame are classified according to their position in the Gaussian distribution.

In (Zheng, Wang, Nihan & Hallenbeck, 2006),a mode-based approach is proposed to discriminate background and fore- ground. It analyses the pixels' colour values from a sequence of frames and then assigns the mode of the series as the background image. This algorithm cannot perform well on crowded scenes and fails on static objects.

Gaussian Mixture Model (GMM) as a detection method for moving objects was proposed in (Stauffer & Grimson, 2000, 1999). Each pixel is modelled temporarily as a mixture of multiple Gaussian with being able to be updated online. The stability

of Gaussian distribution discriminates the pixels in a way to show they belong to the stable background or moving foreground; a threshold is set to evaluate and decide for the stability. Although this method deals with high computational complexity, it outperforms other background subtraction methods in terms of lighting changes. Many researchers have utilized GMM in order to detect the traffic participants. Kim et al. uses GMM as their base method to extract the potential moving objects and then apply CNN to classify them (C. Kim, Lee, Han & Kim, 2018). In (Moayed, Griffin & Klette, 2017), the authors proposed a technique to replace GMM instead of selective search (Uijlings, van de Sande, Gevers & Smeulders, 2013) to find the interesting objects to be detected and classified.

Optical Flow

Optical flow is the apparent changes in brightness of the objects in a sequence of the frame. It matches those pixels in the consecutive frames using temporal and spatial information. Even though the optical flow results in higher computational cost, it is less susceptible to occlusion problems. Optical flow is also a famous base approach in object tracking which will be further investigated in Section 2.3.

Appearance-based Methods

The visual appearance of the objects is traditionally used to detect the objects in the images. These methods later are extended to be implemented in the video to detect moving and static objects. Gradient, pattern, shape and colour features are four categories of model-based approaches. Gradient features methods are those which construct the histogram from the gradient of the image. Pattern features methods are based on the relations between neighbouring pixels in a sub-region. Shape description and its contours are discussed in the shape feature methods. Colour features methods deal with the representation of the objects according to the colour or intensity information. Appearance-based methods are still widely used in object detection in a vast variety of

applications.

2.2.2 Data Driven-based Methods - Deep Learning

In comparison to the model-based approaches, in data-driven methods, there is no need of a specific engineering customization for this particular case and the feature representation is automatically generated from a huge dataset. Traditional machine learning algorithms strongly rely on feature representation. Thus, to have an effective machine learning algorithm, many efforts should be performed in pre-processing and transformation of data (Bengio, 2009) while in data- driven approaches, the machine learning algorithms are less likely to be the subject of feature engineering. In general, deep learning has two main capabilities that define its discrimination: automatic feature extraction, which is sometimes called feature learning; and scalability (Goodfellow, Bengio & Courville, 2016; LeCun, Bengio & Hinton, 2015).

Deep learning allows the computers to learn complex features from simple ones. In other words, deep learning algorithms utilize from the unknown structure in order to extract the effective features in multiple levels (Bengio, 2012). Scalability of deep learning shows the importance of big data in a way with the larger neural network, which is trained with large data, the performance continues to increase, while other traditional machine learning algorithms reach a plateau in performance (Ng, 2015).

Deep learning, as a data-driven based approach, received more attention from 2012 after the current enhancements in hardware capabilities for processing huge amounts of data using complex algorithms (Hinton, 2012). This is due to the fact that a backpropagation algorithm was proposed in 1986 (Rumelhart, Hinton & Williams, 1985) as an optimizer for neural networks. It was first applied to deep neural networks by LeCun et al. (LeCun et al., 1989, 1990) which was a highly complex algorithm. With the extent of GPU-enabled systems, the implementation time of weeks is brought

back to days. Three main reasons for the popularity of deep learning in recent years are technology improvements to have fast computers, large datasets and selecting the accurate initial weights (Hinton & Salakhutdinov, 2006).

In particular, the high trend of using deep learning will continue; some researchers believe it is the primary and the most important approach to Artificial Intelligence as the ultimate goal of machine learning (Perez, 2017).There are different types of deep architecture. This review aims to utilize the convolutional neural network (CNN) which is designed mainly for image classification and detection.

Convolutional Neural Networks for Classification

CNNs are a type of deep feedforward neural network that is currently widely adopted in the computer vision community due to their structure. It is proposed to handle the data in the form of multiple arrays. Similar to other deep learning models, the architecture of CNNs is composed of multiple stages. However, the difference is that the first few stages are of two types: the convolutional layer and the pooling layer. The convolution layer attempts to derive the relationship between pixels using small squares of input data; whereas the pooling layer results in the most useful feature representation.

LeNet (LeCun, Bottou, Bengio & Haffner, 1998) was the very first CNN in the field of deep learning. With its fundamental architecture of having the convolution layer and the pooling which concentrate on pixel correlation, several models are then investigated. At the time of the LeNet proposal, no powerful processing tools such as GPU were invented. Therefore, training took long enough to not attract many researchers to focus on the idea. AlexNet (Krizhevsky, Sutskever & Hinton, 2012), proposed in 2012 when the usage of GPU became more common, was a deeper and wider version of LeNet. It uses rectified linear units (ReLU) as nonlinearity while LeNet used the tanh or sigmoid function. AlexNet was 10 times faster and became a revolutionary in large neural networks. An improvement of AlexNet, Overfeat (Sermanet et al., 2013), was proposed

as an extension. It also proposed learning bounding boxes to localize the objects.

VGGNet (Simonyan & Zisserman, 2014) contributes to show that the depth of the network has critical impacts on the performance. By a having larger and deeper network, features that are more complex are extracted. VGGNet suffers from computational time. Network in Network (NiN) (Lin, Chen & Yan, 2013) uses a 1×1 filter for convolution, then a spatial MLP layer is used after each convolution. This convolution helps to combine convolutional features in a way that is more effective.

GoogleNet (Szegedy et al., 2015) utilized the concepts of bottleneck layer, in which high performance is achieved using 10 times fewer operations compared to AlexNet. This model is based on inception module that uses 1×1 convolutional blocks to reduce the number of features before expensive processing. In 2015, batch normalized Inception, also called Inception V2 was proposed by the same group (Szegedy, Vanhoucke, Ioffe, Shlens & Wojna, 2016). Batch normalization normalizes the response to mean and standard deviation of all feature maps at the output of each layer.

ResNet (He, Zhang, Ren & Sun, 2016) takes advantage of the residual net in which the input and the outputs of two convolutionals are fed to the next layer. In spite of extremely deep residual nets in comparison to plain CNNs, the training error in ResNet is lower when the depth increases due to easy optimization. Moreover, ResNet outperforms in accuracy by gaining from increased depth.

In the proposed DenseNet (Huang, Liu, Van Der Maaten & Weinberger, 2017), each layer has directly connected to all other layers in the network. The authors claimed several compelling benefits such as reducing the number of parameters. Thus, less memory and computations are needed.

In recent years, researchers performed a lot of improvements in extending the layers and optimizing the functions to meet their requirements. However, the main focus on deep learning changed to object detection rather than just classification.

Convolutional Neural Networks for Detection

The CNN was mainly proposed for object classification. CNNs are considered to be slow and computationally expensive to detect objects by a sliding window detector.

R-CNN (R. B. Girshick, Donahue, Darrell & Malik, 2013) is proposed to address this issue by finding the potential objects using Selective Search (Uijlings et al., 2013) which decreases the number of bounding boxes which are fed to the CNN followed by SVM to predict the classes of the patches. Then the optimization is performed by training bounding box regression.

To overcome the disadvantages of R-CNN such as time efficiency, the same researchers proposed a faster object detection algorithm called Fast R-CNN (R. Girshick, 2015). Unlike R-CNN where region proposals are fed into the CNN, in Fast R-CNN, the input image is fed to generate a convolutional feature map and consequently, the region proposals are identified. Using the softmax layer, the class of the proposed regions and offset values are defined.

Faster R-CNN (Ren, He, Girshick & Sun, 2015) eliminates the need for a slow region proposal, instead it allows the network to learn it. The difference between Fast R-CNN and Faster R-CNN starts after the identifying convolutional feature map. A separate network is used to predict the region proposal.

In Single Shot Detector, known as SSD (Liu et al., 2016), a convolutional network is run on input image only once and a feature map is generated. Then, a 3×3 convolutional kernel is used to predict the localisation and probability of the object. SSD also uses the concept of the anchor boxes at different aspect ratios.

You Look Only Once or YOLO (Redmon, Divvala, Girshick & Farhadi, 2016) divides the image into a grid of $s \times s$ and each grid predicts N bounding boxes and class prediction. The confidence for each bounding box confirms if it contains an object or not. Therefore, for each image, $s \times s \times N$ possibility is measured that by setting up a

threshold, the majority of the bounding boxes is removed. Both YOLO and SSD feed the entire image to CNN only once.

YOLOv2 (Redmon & Farhadi, 2017) has significant improvements compared to YOLO. Batch normalisation which is used on all convolutional layers, results in a 2% improvement in mAP. Furthermore, YOLOv2 uses a 2 times larger image for fine-tuning the classification network. But the major improvement of this version is due to using the anchor box to predict bounding boxes which increases the recall significantly by 7%. YOLOv2 applies K-means clustering (Lloyd, 1982) on the ground truth bounding boxes to find the most suitable size for the dataset. It restricts the location of the bounding box to be close to the original grid location using logistic activation. YOLOv2 also supports multi-scale training for every 10 batches. In general, YOLOv2 is much faster and more accurate than YOLO. Handling the trade-offs between accuracy and speed, YOLOv2 outperforms all aforementioned detectors.

YOLOv3 (Redmon & Farhadi, 2018) is the latest version of YOLO. The abjectness score is predicted using logistic regression to differentiate the overlapping bounding boxes in non-exclusive classes. Also, they use new feature extractor as their based network.

2.3 Single-view Object Tracking

Object tracking in a sequence of video frames is a challenging task in behaviour analysis because of accuracy and time efficiency. Object tracking involves finding the objects in a sequence of frames. Traditional object tracking algorithms have a strong relation to object detection, sometimes called tracking by detection methods. Therefore, discrimination between moving object detection, which was discussed in the previous section, and object tracking cannot be clearly defined; thus, in this context, we just focus on how to analyze the movement of already detected objects.

To investigate the behaviour of moving objects, the motion information should be extracted. Motion interpretation plays a critical role in the vision-based surveillance system. There are generally two approaches to extract motion information in a video: optical flow and trajectory-based tracking.

2.3.1 Optical Flow

Optical flow is commonly used to extract the spatiotemporal information rather than detection of objects. It approximates the motion pattern of objects caused by relative motion in a sequence of frames. Optical flow, in general, is one of the main tools for tracking.

Shi and Tomasi (Shi & Tomasi, 1993) define an algorithm that tracks the good features based on the carefully chosen corner points according to the basic optical flow idea for all neighbouring pixels (Lucas & Kanade, 1981). The features are those having large spatial gradients in two orthogonal directions. In general, the combination of the approach is called a Kanade–Lucas–Tomasi (KLT) feature tracker that directs the search for the best-detected feature. The combination of the KLT tracker with other methods to improve the performance is also used in many research and applications. Rabaud and Belongie used clusturing the feature points and the KLT tracker to address the counting of objects in the crowd (Rabaud & Belongie, 2006). In (Jabar, Farokhi & Sheikh, 2015),the KLT is used to track the objects in UAV-based applications. The gradient weighted optical flow (GWOF) is proposed to improve the KLT tracker to detect and track multiple objects (Buddubariki, Tulluri & Mukherjee, 2015). Dense optical flow for smoothing terms in distortion-robust spherical camera motion estimation is utilised in (Pathak, Moro, Fujii, Yamashita & Asama, 2018), where epipolar geometry helps to estimate 6 DoF camera motion. In spite of being notorious in the research, the computation time should be considered for the deployment.

2.3.2 Trajectory-based Tracking

Tracking can identify the path of a detected object in the background plane. Trajectory-based tracking methods can be classified into three main categories: point-based tracking, kernel-based tracking, and silhouette-based tracking (Yilmaz, Javed & Shah, 2006). In this section, each category is briefly explained and some research, which uses the method, is cited.

Point-based Trajectory

Point based trajectory tracking shares a similar concept with the tracking of the features; moving objects are tracked using their feature points. Indeed, the drawbacks of the methods in this group are handling the occlusions and false detection of the objects to track. To put it briefly, the accuracy of these methods strongly relates to the object detection: if the objects are detected correctly, the possibility of true tracking is high.

The Kalman filter is a set of equations that estimate the past, present, and future states even when the nature of the modelled system is unknown. A Kalman filter defines the states of a linear system where the state is assumed to have a Gaussian distribution. Object tracking (based on a Kalman filter) is performed by predicting the object localization from the previous state and verifying the existence of the object at the predicted location. In addition, some image sequences are used to train the motion model before tracking starts. Several works use a Kalman filter for tracking from 1986 and the 1990s (Beymer & Konolige, 1999; Rosales & Sclaroff, 1999) to recent years.

One disadvantage of a Kalman filter is the assumption that the state variables follow a Gaussian distribution. A particle filter (Del Moral, 1996) was proposed to overcome this drawback. In a particle filter, a state density is composed of a set of samples (particles) with weights, which defines the importance of the particle. After selection, prediction, and correction, the new samples are generated, having the ability to estimate

the new object location. A particle filter still attracts many researchers to use or improve it for their object tracking. The drawbacks of traditional particle filters are they are relatively computationally expensive, and lack diversity.

Before delving into kernel-based trajectory, we should mention that the Kalman filter, particle filter and Bayesian filter are instances of many methods available and proposed every day in this category, yet they are still famous.

Kernel-based Trajectory

Kernel tracking is the motion tracker in which the movement of the object is defined frame by frame using the region of the object. The algorithms in this context diverge in how appearance is presented, the motion estimation algorithm and number of tracked objects. Similar to point-based trajectory, here, only the most to overcome the illumination changes common approaches are discussed.

Template matching techniques are widely used for two main reasons: (1) They are relatively simple, and (2) they benefit from easy computation. They search the region (in our application, video frame) to find the trajectories. To generate the templates, image colour, intensity, and image gradient are usually used to overcome the illumination changes. The template matching methods follow the brute force search, so the computational cost for localizing the template in each frame is high. In order to reduce this time complexity, the search is limited to the area of the object position in the previous frame. Detecting the objects using other object representation in the specific regions is considered to be in this category as well. Improvement of template matching methods, handling the trade-offs among time complexity and accuracy are still of interest to many researchers. In (Mercier, Trottier, Giguere & Chaib-draa, 2017), the objects are detected by a pre-processing by template matching and deep learning classification.

Tracking the objects using different views tackles issues related to the appearance

of the objects. Thus, the different view can be learned offline and used for tracking. The affine transformation which minimizes the difference between the target image and its projected image is used in (M. J. Black & Jepson, 1998). Support vector machine (SVM) is the main classifier for tracking in (Avidan, 2003). SVM classifies the image into positive and negative examples by giving a score to each object. Negative means background and positive defines the objects which need to be tracked.

A diversity of research work attempts to detect objects in a frame using kernel-based approaches, in which the object regions are tracked in consecutive frames.

Geometric shapes (kernels) around the objects to be tracked are a general drawback of the kernel-based trajectory tracker; part of the objects reside outside the kernel while part of the background can be inside it. To resolve this issue, one approach is to force the kernel to reside inside the object or assign weights to the pixels inside the kernel according to their feature representations.

Silhouette-based Trajectory

This category relates to the detection of those objects which have complex shapes while they cannot be described using simple shapes. This algorithm tracks the accurate object region by the model generated in the previous frames.

Shape matching techniques search the frame to find the objects in the current frame. It is similar to the aforementioned template matching, where the object model is searched in the frame. The tracking is done by finding the similarity of the object model in the current frame and in the next frame, and the object model is reinitialized to tackle the issues related to viewpoint and brightness changes.

There is another approach in which the contours evolve to track the silhouette. Contours can be tracked using temporal gradient or appearance statistics. In the first case (temporal gradient) (Bertalmio, Sapiro & Randall, 2000), optical flow parameters are computed for the position of each contour and then energy function is used to

Table 2.1: Analysis of trajectory-based tracking methods

Category	Advantages	Disadvantages
Point-based	• Deals with entering and existing objects	• Expensive computation • Cannot handle occlusion
Kernel-based	• Includes orientation tracking • Discriminate object and motion • Can handle occlusions	• Can be computationally expensive • Need kernel customization
Silhouette-based	• Flexibility to handle different shapes • Complete object tracking • Resistant to noise • Can handle occlusions	• Expensive computation • Time Complexity

determine the speed of the tracked contours. Optical flow based silhouette tracking is also used in other research such as the method proposed in (Mansouri, 2002) by slightly modifying the way the calculations of energy function. In addition to the temporal gradient, the consistency of the statistics inside and outside of the object can be used to track the contours; the contours have to be initialized and localized in the current frame (Yilmaz, Li & Shah, 2004).It is worth mentioning that one of the papers has been published in the area related to traffic scenes (Koller, Weber & Malik, 1994), in which the 3D contours of the vehicles are tracked. Table 2.1 summarizes the general advantages and disadvantages of the trajectory-based methods.

2.4 Multiview Object Detection and Tracking

The achievements in multiple view geometry in recent decades are the result of the enhancement in theoretical understanding and estimating mathematical objects from

images, linked to the powerful search and matching algorithms (Hartley & Zisserman, 2003).

The algorithm of multiview object detection and tracking can be grouped into three categories: track-first, fuse-first and manifold-based approaches (Taj & Cavallaro, 2010). The first approach tracks the object in each camera view separately and then matches the results to other camera views. In the second approach, the objects are detected in each camera and then projected in the common occupancy map, then the tracking is applied to those detected objects in that common view. This method is called fuse-first. Manifold-based approaches are used when the camera calibration is not available, or one cannot decide about the planer surface. Since the manifold-based approaches are out of the scope of this thesis due to availability of calibration parameters, only the first two approaches are investigated in more detail in Section 2.4.1 and Section 2.4.2.

2.4.1 Track-first Approaches

In track-first multiview tracking algorithms, the objects are first tracked and then the tracks are projected on a common plane for merging. This tracking can be performed independently in each camera or collaboratively among cameras, while, in collaborative track-first algorithms, tracking results from different views are improved in each camera. The object and corresponding tracks association is the major problem to be solved.

In *independent tracking*, the projection of localisation and tracking of the objects is projected with correspondence to another camera view or on a top-down/bird's eye view.

A single object is tracked in 2D image coordinates and 3D world coordinates using a Kalman filter and then independent tracks are projected on the common view in(J. Black, Ellis & Rosin, 2002). In multi-object tracking using multiple views, all separate tracks

for each object are projected on a common plane for later fusion. However, spatio-temporal correspondence of the same object is the challenge. This association between objects can be established by considering target data such as the height or ratio of the interval to measure the similarity (Wang, He & Velipasalar, 2010).

The probability hypothesis density (PHD) filter is utilised to find the feature correspondence and to handle the initialization of new tracks in (Houssineau, Clark, Ivekovic, Lee & Franco, 2016). A Gausuian Mixture phd filter (GMPHD) is applied on each camera view and also the top-down view. Tracking of the objects is then performed by assigning the label to each Gaussian component. A position, size and colour histogram is used in object matching (Anjum & Cavallaro, 2009).

The main challenge for independent tracking is handling the occlusion because the tracking is done separately on each view and they are not collaborating to improve each other. Collaborative approaches are designed to address this issue.

Single-view tracking helps in improvement of the tracking on the other view in *collaborative tracking*. The objects are tracked using particle filters and the particles are then projected onto the common plane in (Du & Piater, 2007). To improve the localisation accuracy on the common plane, the principle axis (vertical line from feet to the head) is also projected on the plane and the intersection of the principle axes are the target location.

Generally, track-first approaches involve many steps to confirm object association between views. Fuse-first approaches may reduce this complexity, however, the objects' information will be discarded.

2.4.2 Fuse-first Approaches

Collaborative track-first approaches may introduce some estimation error as they involve multiple steps to improve the tracking across the views. If the objects are tracked on the

common view by considering the information received from the multiple views, this estimation error might be reduced.

The algorithms in this category can be categorised by the used features and also how the common plane is computed. Different features are used to meet the requirements of the applications such as a person's feet (K. Kim & Davis, 2006), the object centre points (Focken & Stiefelhagen, 2002) or the foreground moving objects (Lopez, Canton-Ferrer & Casas, 2007; Fleuret, Lengagne & Fua, 2005). In (Liu, Xu, Zhu & Mu, 2018), the authors use hierarchical representation to leverage discriminative human attributes such as the accessories, gender or speed and, by using attribute grammar, parse graphs from videos are constructed.

The main difference between track-first and fuse-first is the number of tracking steps. While fuse-first approaches only perform tracking once on the common plane, they should deal with detection parts. Detection can be established before projecting the object in the common plane or after projection. Therefore, this research uses simultaneous object detection and tracking.

2.5 Traffic Safety Analysis at Intersections

In the area of traffic safety, intersections always attract engineers. This is due to the fact that statistics show the number of incidents at intersections are considerably high. In New Zealand from 2008 to 2018, 34.93% of the accidents were intersection-related in which 19% of fatalities happened. Thus, this leads to further investigation and analysis of the behaviours of traffic users at intersections.

Intelligent surveillance has been mainly used for the security and public safety for years. In surveillance, the events can be categorized into at least two aspects: spatial and temporal, where an incident can happen instantaneously or lasts over a period of time (Yan, 2016). Multi sensor data fusion is commonly used in the area of intelligent

surveillance, however, this research aims to consider cameras as the only means of providing data.

To analyze the safety at traffic intersections, different factors need to be considered, ranging from participants' behaviour assessments to predict for the conflicts and crashes. This analysis is challenging as it deals with humans which usually show many unexpected behaviours; therefore, many researchers aim to provide a package that defines the risk factors, which are analyzed based on the driver's intentions, recorded inside a vehicle. Irrespective of what happens inside a vehicle, this research concentrates on the safety features captured by surveillance cameras in traffic intersections. To achieve this goal, the understanding of inter-participant risks is compulsory. By inter-participant risks, we mean those risk factors that involve risks between different traffic participants. Section 2.5.1 defines the literature in behaviour analysis, while Section 2.5.2 describes the safety factors.

2.5.1 Behaviour Analysis

In computer vision, behaviour analysis helps to understand the actions of specific objects and the possible future outcomes of these actions. This analysis can provide useful information for transportation engineers who are involved in intersection planning in order to design considering the safety of participants.

Measuring the vehicle speed and acceleration is one major area of research by detecting the vehicle trajectories with Kalman or particle filters. Detection of speed in the field of view of the camera according to the approximate height of the vehicle is discussed in (Kumar, Ranganath, Weimin & Sengupta, 2005). As measuring the speed and acceleration needs wider FOVs, microscopic real data are utilized to analyze and compare their distribution near stop signs (Viti, Hoogendoorn, van Zuylen, Wilmink & van Arem, 2008). In (Moayed et al., 2017; Moayed, Chien, Zhang & Klette, 2019),the

speed is measured by projecting the object in two frames onto a top-down view using the transformation matrix.

Detection and prediction of vehicles' trajectories, including going straight or turning behaviours is another significant factor for analyzing the risks. To predict the trajectory, a model is built, and the extracted trajectory is compared with the model. The probability of different trajectories of two vehicles at intersections is investigated in (Käfer, Hermes, Wöhler, Ritter & Kummert, 2010); in which quaternion-based rotationally invariant longest common subsequence (QRLCS) is used to predict the trajectories ahead in the 2-4 seconds time frame. The decision for the trajectory future path is also discussed in (Tran & Firl, 2014),), where the Gaussian regression model and particle filters are used to show the predicted path. A semantic model for automatic traffic event detection is proposed based on a combination of both mathematical and region-based paths to predict the trajectory (Yu, Zhang, Tian & Liang, 2012).

To analyze the vehicle's behaviours, the bigger view of the intersection leads to a better and more accurate analysis of the movement of the vehicle. In addition, most crashes happen in a short amount of time without following a rule to be predicted. Therefore, different learning approaches are proposed to learn different driving models and accidents (Kumar, Perrollaz, Lefevre & Laugier, 2013; Hülnhagen, Dengler, Tamke, Dang & Breuel, 2010; Schreier, Willert & Adamy, 2016).

A combination of SVM and the Bayesian filter is used to discriminate the violating and normal drivers according to the speed, acceleration, and distance to the stop signal (Aoude, Desaraju, Stephens & How, 2011). Due to their simplicity to model stochastic data while not having large datasets, Bayesian and HMM are widely used to analyze driver's behaviours (Lefèvre, Laugier & Ibañez-Guzmán, 2011; Streubel & Hoffmann, 2014).

Dilemma zone, a range in which a vehicle approaches during the yellow phase, is one of the main factors of risk analysis. The time between of the onset of the yellow

phase and activation of the vehicle's brake is called perception-reaction time (PRT). In fact, the critical PRT is calculated as one second which is bigger than 85% of PRT cumulative distribution (Goh & Wong, 2004). Considering PRT and the dilemma zone in safety and behaviour analysis deals with different issues. They are dependent on a driver's biologic, and their occurrence according to the onset time of the yellow phase. There are several ways to consider the dilemma zone in collision avoidance. However, it is not the target of this research project, so we do not go into more detail.

Pedestrians' motion prediction is one way of analyzing their behaviour. Since this task strongly involves human decisions and external situations, there is not a complete approach, which assures correct prediction. Pedestrians' motion (Abramson & Steux, 2004) is detected and predicted from a mounted camera, combining a motion model with a particle filter. However, when the pedestrians are standing, the accuracy of the motion model decreases. Four states of a Markov chain, "standing still", "walking", "jogging", and "running", related to pedestrian motion speed are used as a motion analysis in (Wakim, Capperon & Oksman, 2004), despite that it cannot investigate the "crossing" state when pedestrians are crossing an intersection. A pedestrian path prediction in stereo-mounted cameras on vehicles is presented in (Keller & Gavrila, 2013). Features extracted from dense optical flow are used in Gaussian dynamical models and probabilistic hierarchical trajectory matching. Together with Kalman filtering, they discriminate a pedestrian's motion as stopping and crossing.

Rather than motion prediction of pedestrians, there are many contributions involved in the behaviour analysis, including walking and crossing speed, age, gender and group size (Montufar, Arango, Porter & Nakagawa, 2007). Many studies investigate these factors while handling high data collection costs. Thus, automatic data extraction is an appealing task. Generally speaking, pedestrian motion analysis at intersections deals with two main issues (Shirazi & Morris, 2016):

1. The pedestrians' motion at traffic intersections is inconsistent because traffic signals force them to stop. These signals have impacts on the estimation of waiting time as well.

2. To analyze the behaviours, the individual pedestrian is of interest in this context. As they are crossing the intersections in a group, to derive analysis based on an individual person is a challenging task.

However, though analyzing based on manual data sources such as observation and surveys may result in optimal outcomes, there is still a gap for automatic vision-based approaches, which have the ability to handle the tradeoffs among accuracy, time, and cost. The pedestrian is detected using a fusion of technologies to estimate the waiting time and crossing count yet is incompatible with a large number of pedestrians crossing the intersection.

2.5.2 Safety Analysis

Safety analysis refers to the measurement of safety at intersections and prediction of conflicts from those safety measurements and datasets. By safety measures, we mean the assessment process undertaken by humans involved in the decision-making process at an intersection to avoid accidents. These assessments are gap, risk, and threat.

Gap refers to available time/space for a maneuver or between two approaching vehicles. Before passing or crossing intersections, pedestrians and drivers check the available gap. An accident can be avoided when the accepted gap is large enough to give the users more time to decide. Although there are some methods estimating the accepted gap size (mainly using probability distribution function and regression models), it is not feasible to implement a gap inference in real traffic observation as other external factors such as age and gender should be involved. In addition, the speed of vehicles plays an important role in gap size. The higher the vehicle speeds, the smaller the accepted gap.

Threats show the possibility of a collision imposed by other vehicles in the near future. Time to intersection (T2I) and distance to intersection (D2I) are contributing warning factors, which need to be considered jointly. These factors (Chan, Marco & Misener, 2004) are used in a way that warning is exposed when the values of T2I and D2I are greater than a threshold. A combination of intention predictors based on a support vector machine with a threat assessor based on random trees is proposed in (Aoude, Luders, Lee, Levine & How, 2010). To identify threat level, trajectories should be detected in a real-time manner.

In the context of safety analysis at intersections, the risk is the uncertain level of danger for a specific vehicle under a certain maneuver introduced by other participants. The difference between risk and threat is that risk assessment detects the dangerous situation that might occur by driver error, while the threat is the imminent possibility of collision. One way to assess the risk is to predict and evaluate a driver's intentions by comparing against the expected model. The joint motion of the vehicles includes the layout of the intersection in a probabilistic framework (Lefèvre, Laugier & Ibañez-Guzmán, 2012a). In addition, the path can be compared with all possible paths. A Bayesian Network is used in (Lefèvre, Laugier & Ibañez-Guzmán, 2012b). Risk analysis for pedestrians is also investigated in some studies. Illegal crossing during the stop phase and crossing away from the traffic signal (King, Soole & Ghafourian, 2009), pedestrian's risk as a function of time (Tiwari, Bangdiwala, Saraswat & Gaurav, 2007), the correlation of a vehicle's flow and a pedestrian's risk (Leden, 2002) are examples of pedestrian's risk analysis.

To analyze the risk, a pyramid, illustrated in Figure 2.1, is used for the severity of the conflict, ranging from undisturbed passages to accidents (Svensson & Hydén, 2006). This pyramid shows how the events are classified according to their safety quantification.

To have a quantification analysis, some important safety measurements (called

Figure 2.1: Traffic safety pyramid

surrogate safety measurements) are used. Here, we have a simple definition of the most commonly used ones:

- *Time to Collision (TTC):* The time specified for two traffic participants to collide if their path and speed continue to be unchanged.
- *Distance to Intersection (DTI):* The distance until a vehicle reaches the stop sign, while the speed is unchanged.
- *Time to Intersection (TTI):* The time until a vehicle reaches the stop sign, while the speed is unchanged.
- *Time Headway:* The time difference between when a vehicle arrives at a specific point and the time that following vehicle reaches the same point.
- *Post-Encroachment Time (PET):* The elapsed time between the end of the encroachment of a vehicle and the time the road user arrives at the potential collision point.

The severity index, ranging from 0 to 1, is one of the measures; it correlates with PET and TTC using Equation 2.1,

$$SI = exp(-\frac{TTC^2}{2(PET)^2}) \tag{2.1}$$

Departure headway is another measure, which is specifically used at intersections. It shows the intersection capacity and time of traffic signals. It is defined as the time between following vehicles when they start crossing an intersection in the green phase. In general, surrogate safety measurements are widely used in automated vision-based safety analysis.

There are two major ways to use them. First, it is to utilize the measurements directly in a region of interest. The problem with this approach arises when the system encounters crowded scenes with many traffic participants or stopped vehicles. In this case, the accuracy of the method degrades, and the result of safety is not precise. The second way is to cluster the trajectories to learn a model using probabilistic frameworks. HMM is used to cluster trajectories for safety analysis and generate a model for conflicting trajectories. Similar to other model-driven approaches, this approach suffers from the availability of a large amount of data.

2.6 Discussion

As far as the current existing literature revealed, there is a gap between having a robust vision- based traffic safety analysis package and a complete package that reduces the data collection cost by using only available surveillance data captured by cameras instead of the sensors such as Lidar and radar. There exist many types of research on traffic safety analysis and pedestrians' behaviour while they deal with high data collection costs when using surveys and observations. Therefore, automated vision-based data extraction can

lead to a wider understanding of what is happening at intersections in order to consider those values for further planning, design, and analysis. Moreover, those studies which deal with computer vision consider safety factors as separate parameters, therefore a combination of safety parameters belongs to each participant and interactive parameters imposed by others is still a new challenge.

Newly proposed approaches such as deep learning can affect the robustness and accuracy of the system in a way that it can handle the moving and the stopped participants in different environmental conditions if the algorithm is fed with a large amount of data. The accuracy of the systems dealing with human safety plays an important role, thus multiview approaches can increase the field of view of the region of interest and improve object tracking. Traffic safety using multiview approaches can effectively provide a robust solution to traffic engineers.

Chapter 3

Multiview Road Intersection Recording and Analysis

3.1 Abstract

The wide-angle surveillance cameras used in smart cities are mainly capable of capturing up to 80 degrees field of view (FOV) (Olivia, 2018). To cover a wider area, one may need to use multiple cameras; however, to have an understanding of the scene, the relationships between cameras need to be investigated.

3.2 Camera Setup and Calibration

Camera resectioning or camera calibration refers to the process of estimation of the camera parameters. The purpose of the camera calibration, in general, is to reduce the lens distortion and to determine the size and location of the objects in the real world.

To estimate the camera parameters to relate the 3D world points to their corresponding 2D points, a checkerboard, designed for camera calibration is used. Figure 3.1 shows the design of the checker calibration board that we employ in this study. The size

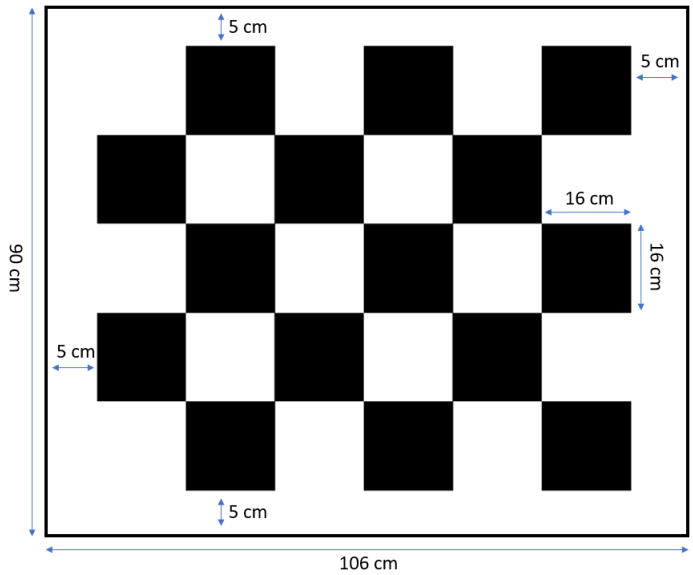


Figure 3.1: Design of the checkerboard used in this study

of the squares are $16cm \times 16cm$ each and the material is designed to get the highest accuracy with a sticker on $6mm$ thick Acrylic board. The decision on the design and material was made to make sure that the board can be used in the outdoor environment with the difficulties of having wind, rain and intense sunshine. The sticker is smooth non-reflective, and the acrylic material is strong enough not to bend.

The main objective of this research is to analyse the safety of traffic participants in a complicated intersection. The modelling of the multiview monitoring system at first step depends on the management policy, the infrastructure network type and the coverage of the network.

One intersection in Auckland CBD is chosen as the desired area for this study. This intersection is located between two major Auckland universities, AUT and UoA, and this causes the intersection to be busy with students as pedestrians together with vehicles and buses. Four cameras are chosen to be used in the calibration process. The cameras are selected based on having a partial overlapping field of view. The design of the intersections and the cameras are illustrated in Figure 3.2.

To calibrate multiple cameras, the intrinsic parameters of each should be identified.

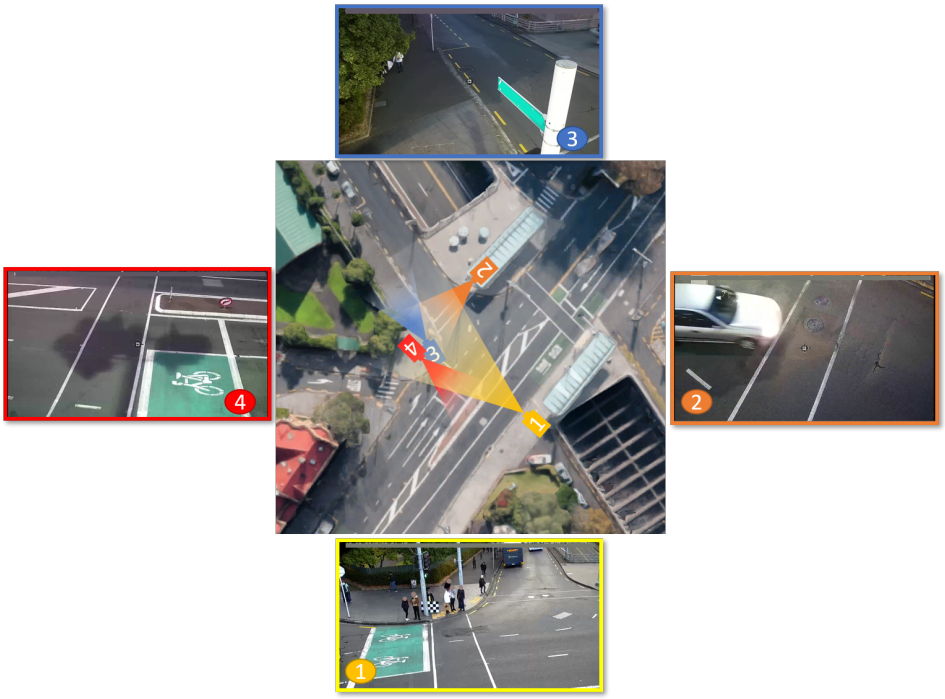


Figure 3.2: The top-down view of the camera location

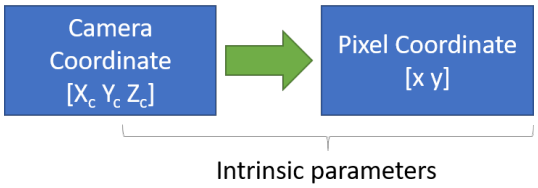


Figure 3.3: Intrinsic parameters

Intrinsic parameters of a camera represent a projective transformation from the camera's coordinates into the 2D image coordinates. The intrinsic parameters matrix is composed of a sequence of scaling, shear and transformation. These transformations represent the focal length, axis skew and principal point offset, respectively.

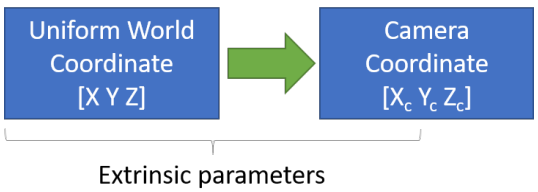


Figure 3.4: Exterinsic parameters

However, extrinsic parameters transform the camera coordinates into a world coordinate. It consist of a rotation R and a translation t. Having these vectors, (x, y, z) is first projected into image plane at $z = 1$ by:

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & t \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{3.1}$$

The distorted pixels $(\breve{u}, \breve{v})$ is found by non-linear distortion:

$$\begin{pmatrix} \breve{u} \\ \breve{v} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} u' & 2u'v' & r^2 + 2u'^2 & 0 \\ v' & r^2 + 2v'^2 & 2u'v' & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 + k_1 r^2 + k_2 r^4 + k_3 r^6 \\ p_1 \\ p_2 \\ 1 \end{pmatrix} \tag{3.2}$$

where $r^2 = u'^2 + v'^2$ is the radial distance. k and p control radial and tangential distortion. Distorted pixels are then projected to pixel coordinates by:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \breve{u} \\ \breve{v} \\ 1 \end{pmatrix} = K \begin{pmatrix} \breve{u} \\ \breve{v} \\ 1 \end{pmatrix} \tag{3.3}$$

(f_u, f_v) are the focal length in pixels in longitude and latitude direction, and (u_c, v_c) is the optical center.

To prepare the appropriate data for calibration, we went to the selected site holding the calibration board in different positions and directions. Later from the synchronised video footage, we selected multiple images with a variety of patterns for each camera, including those frames that the checkerboard is located at the corners to capture the lens distortion.

To simplify the notation, the cameras are defined as C_n where $n \in \{1, 2, 3, 4\}$ for the rest of the thesis.

In order to calibrate 4 cameras according to the design in Figure 3.2, C_1 and C_2 are calibrated together. Also, C_1 and C_3, C_1 and C_4 are paired together.

For this study, we calibrate four cameras. Therefore, we need four sets of intrinsic parameters for each camera. To find these parameters, MATLAB Single-Camera Calibrator (Scaramuzza, Martinelli & Siegwart, 2006) is used. Multiple images for each camera are added into the application. The standard cameras are refined to have a fixed focus and not auto-focused, and the resolution of each camera are set to 1080×720 pixels.

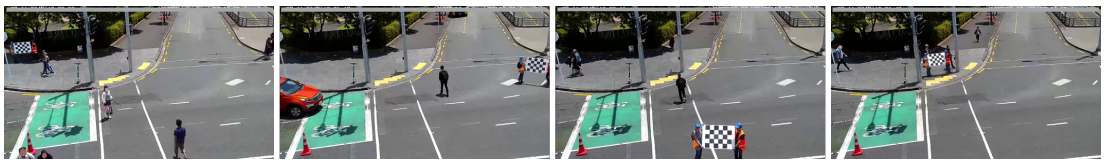
Figure 3.5 shows some sample frames from the cameras used in extracting the camera parameters and stereo parameters. To achieve the highest accuracy, we collect the appropriate images through multiple site visits.

3.3 Uniform World Coordinates

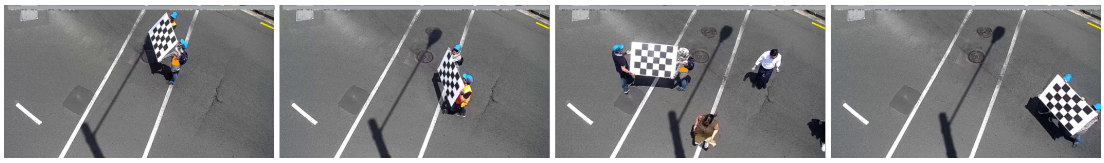
Understanding and modelling of geometry of multiple cameras sharing the same field of view are the challenges that researchers are investigating for years (Hartley & Zisserman, 2003), however, finding the camera internal parameters and how they relate to each other is still effective and useful in many applications.

One of the main motivations to calibrate multiple cameras for this study is to address the issue that for analysing traffic safety at the intersection, a single camera is not enough to provide the sufficient information of the entire intersection. Thus by having multiple cameras, we have extended the field of view to a much greater extent.

Moreover, majority of the calibration experiments in the existing work for single and multiple cameras are done in restricted testbed or limited areas, so this study aims to implement the multiple view calibration with the real CCTV cameras located at the

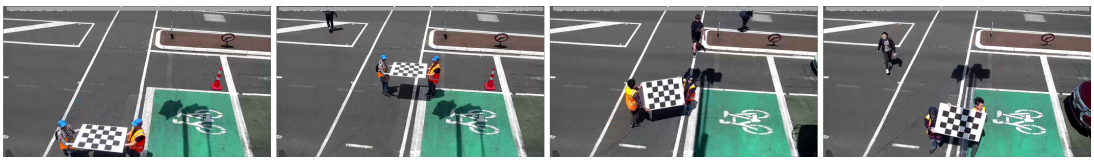


(a) Camera 1



(b) Camera 2

(c) Camera 3



(d) Camera 4

Figure 3.5: Some samples of images used in calibration for each camera

intersection, considering all the challenges that CCTVs face. As a result of this study, multiview safety monitoring can be extended to other real intersections.

As mentioned before, Single-Camera Calibrator APP in MATLAB© 2019a is used to calibrate each camera to extract mainly the intrinsic parameters.

For each camera, the maximum numbers of the frames which show the best and the most distinctive checkerboard locations are captured from multiple 30 minutes videos that the site visit was held.

According to different camera positions and their fields of views, the number of accepted images by calibration App is different. Table 3.1 shows the number of accepted image for each calibration and the average of reprojection error in pixels for each camera after calibration. Reprojection Error (RPE) measures the distance between the keypoint

Table 3.1: Parameters and calibration error in single camera calibration

Camera Number	Number of patterns	Av. RPE (pixel)
1	75	0.19
2	25	0.88
3	42	0.12
4	40	0.18

detected in the calibration image on the checkerboard and a corresponding projected world point of the same image.

As can be observed in Table 3.1 and Figure 3.6, the average of overall RPE for Camera 2 is the highest. The reason is that the field of view of this camera is much limited compared to other cameras and the calibration board on the frame is bigger in size, so the relative error in pixel can also be more significant. Also, the camera position restricts our movement to have many distinctive images. In contrast, wider FOV of Cameras 1, 3 and 4 lead to many images to be useful in the calibration process.

The main reason for calibrating every single camera is to extract the intrinsic parameters.

After extracting the intrinsic parameters using single-camera calibration process, MATLAB© Stereo-Camera Calibration application based on (Zhang, 2000) is used to find the relationships between C_1 and any other cameras; C_1 is the only camera that has overlapping FOV with any other cameras (Please refer to Figure 3.2).

For understanding the geometry of the intersection, stereo vision is applied to allow us to have virtually two identical copies of the same camera (Klette, 2014)

To calibrate two cameras, the matching image pairs should be selected from both cameras. As the footage are from the actual real-time streaming CCTV cameras at an intersection, the synchronised images are later selected in the lab. One significant

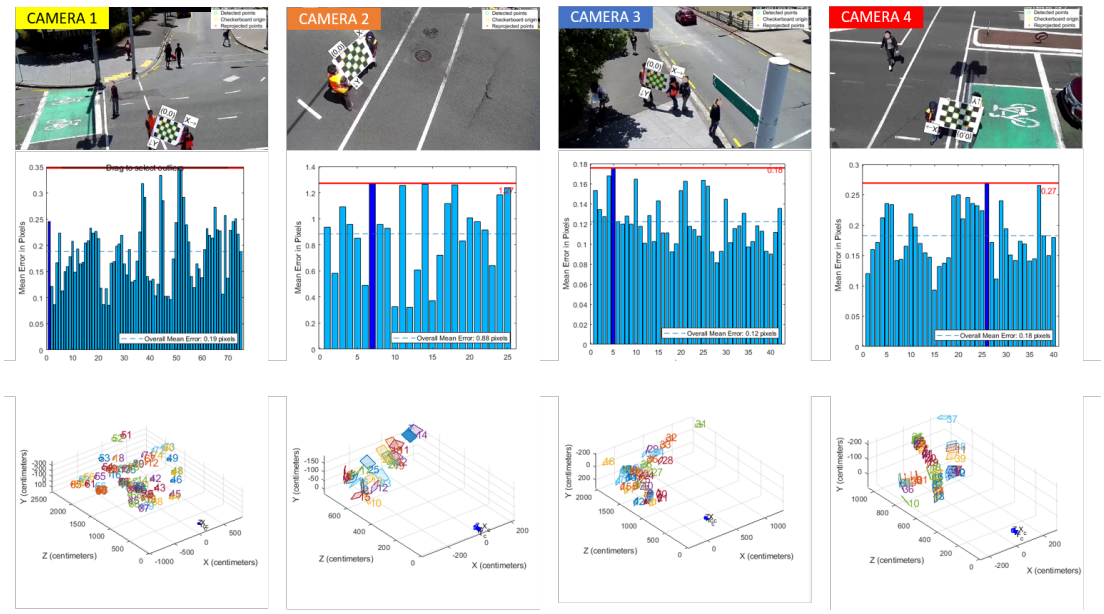


Figure 3.6: Camera calibration for each single camera

difficulty of the image selection process is finding the exact synchronised frames, especially for this specific intersection where we only had 20 seconds to move in a roughly 20 meters intersection wide; the traffic lights allow only 20 seconds for pedestrians to cross, and the width of intersection is around 20 meters. Hence, achieving more synchronised image pairs was somehow a very challenging task.

The average reprojection error in pixels for image pairs of C_1 and other cameras are shown in Table 3.2. Due to a sharp camera angle and long distance between C_1 and C_21, achieving image pairs that can be used for calibration process faces some challenges, and it forced us to visit the site multiple times. However, among 15 image pairs, only 3 with lower RPE were chosen.

Figures 3.7, 3.8 and 3.9 demonstrate the RPE for each pattern and also the cameras-centric view after stereo calibration for each different camera pairs.

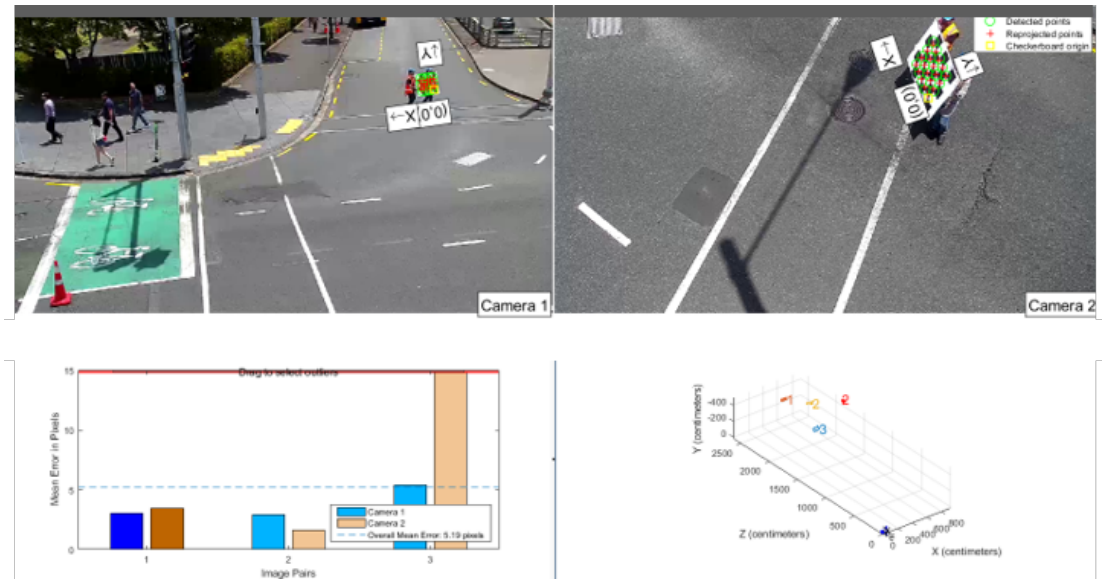


Figure 3.7: Stereo calibration for Camera 1 and Camera 2

Figure 3.8: Stereo calibration for Camera 1 and Camera 3

Table 3.2: Parameters and Calibration error in stereo camera calibration

Stereo Cameras Match	Number of pairs	Av. RPE (pixel)
1-2	3	5.19
1-3	7	0.76
1-4	6	1.26

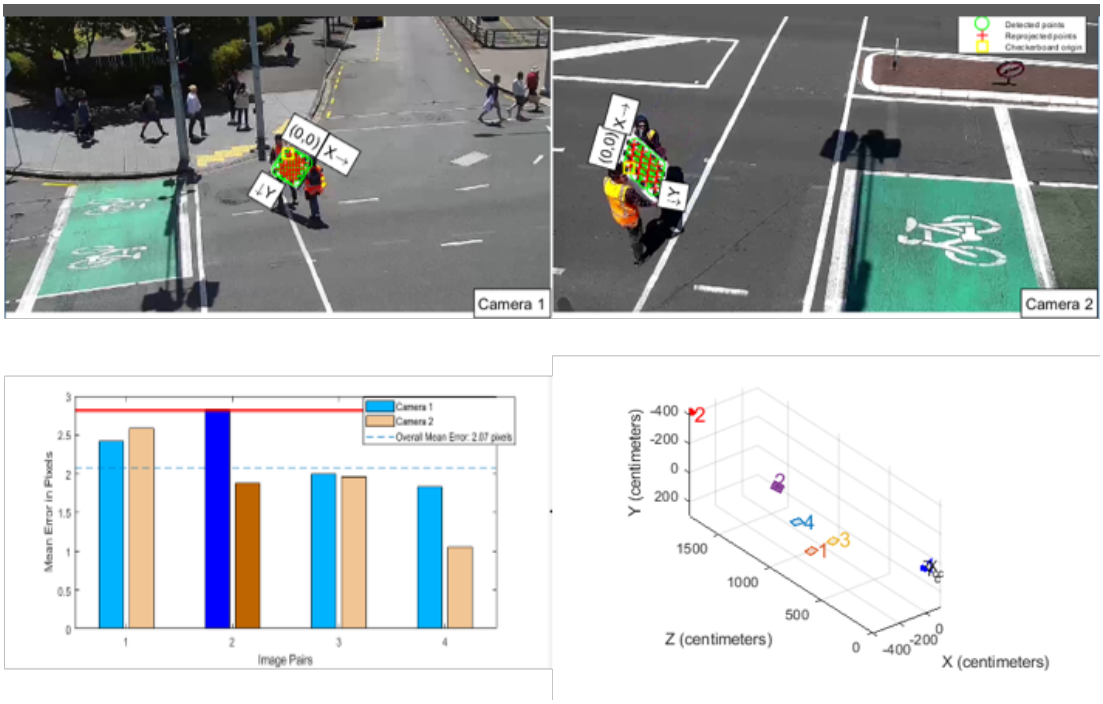


Figure 3.9: Stereo calibration for Camera 1 and Camera 4

3.4 Extraction of the corresponding points

In this section, we generalise the algorithms and steps that are used for calibrating and finding the relationship of the cameras in order to be later used in object association and tracking in Section 4.4.

We assume $C1$ and $C2$ are camera parameters containing intrinsic, extrinsic, and lens distortion parameters for C_1 and any other camera, respectively and $S_1 2$ is the stereo parameter of both cameras containing their geometric relationship.

If $p = (u, v)^T$ and $P = (X, Y, Z)^T$ denote the 2D points in the image and 3D point in homogeneous coordinates, then:

$$p = KR[I| - c] * P, \tag{3.4}$$

where c is the center of the camera in global coordinate system derived from transformation matrix, R is the rotation matrix between world coordinate to camera coordinate, K is the camera calibration matrix which is defined by the intrinsic parameters.

3.4.1 Epipolar Geometry

Epipolar geometry of stereo vision defines a way to relate the cameras, 3D locations of undistorted matching points and corresponding observations.

In general, there are two ways to extract the 3D structure from a set of matching points: calibrated and uncalibrated routes. In the first way, the cameras are calibrated with respect to some world coordinate system and using epipolar geometry, the essential matrix is calculated. However, in the second route, the fundamental matrix is needed to determine the projective 3-dimensional structure of the image scene.

In both ways, triangulation is the initial approach. The 3D location of any visible object point should lie on a straight line which passes through the centre of the projection and projection of the point on the image. To determine the world coordinate of the object X by triangulation, the location of the object point in one image should match the same object point on another image. This correspondence needs an exhaustive search through the whole image, but the epipolar line reduces the efforts to a single line on the second image.

Figure 3.10 shows the triangulation principle and how the epipolar line corresponds to the object point in the first image. Epipolar line is the straight line which intersect the epipolar plane and image plane.

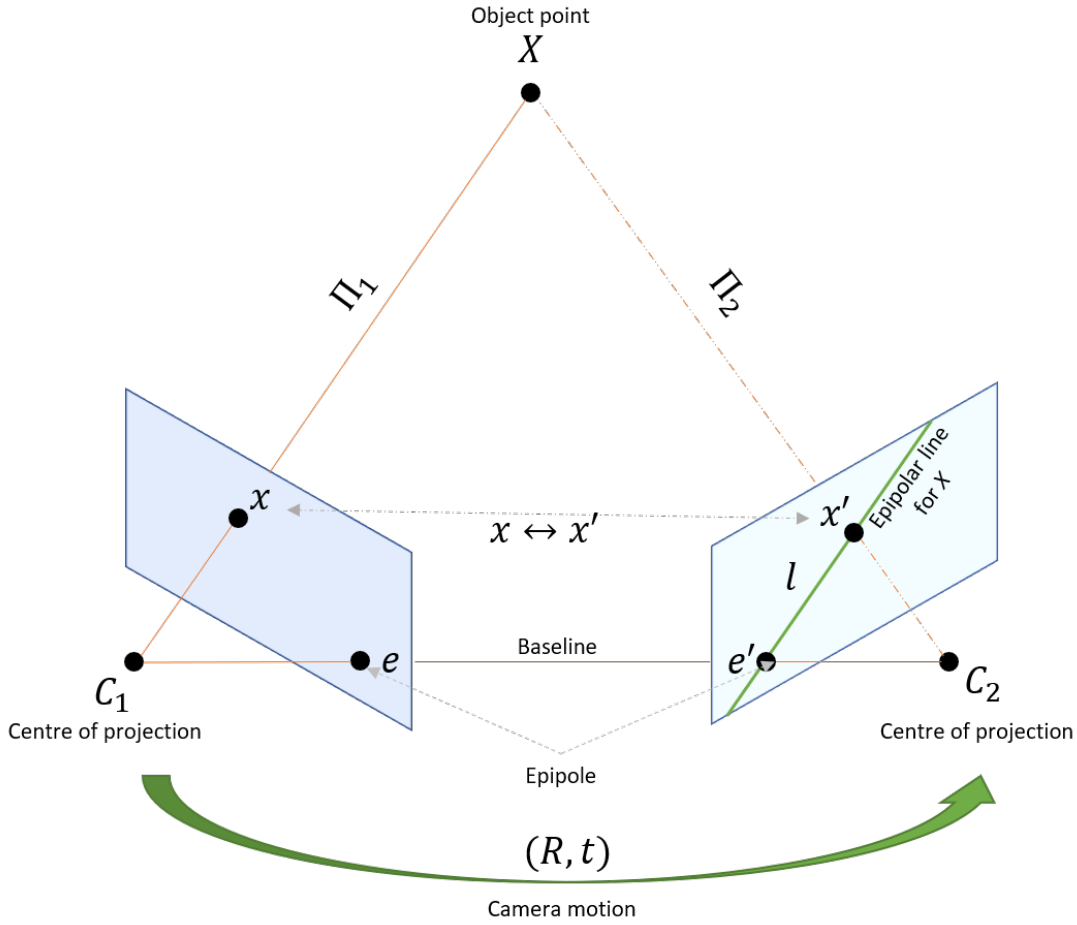


Figure 3.10: The principal of triangulation and epipolar line

Let $x = (u_1, v_1)^T$ and $x' = (u_2, v_2)^T$ be the observations of 3D point $X = (X, Y, Z)^T$ by the projection functions π_1 and π_2, respectively and R and t are rotation matrix and translation vector from the first camera view to the second one. Assuming that the projection functions and camera motion are known, x and x' are denoted as:

$$\begin{aligned} x &= \pi_1(X) \\ x' &= \pi_2(RX + t) \end{aligned} \tag{3.5}$$

To recover the motion and relationships between cameras in epipolar geometry, fundamental and essential matrix are derived in order to find 3D coordinates.

Given the matching undistorted points $x_i \leftrightarrow x_i'$, a 3×3 fundamental matrix F is

denoted by:

$$x'^T F x = 0 \tag{3.6}$$

The eight-point algorithm (Chojnacki & Brooks, 2003) is an infamous algorithm mainly to calculate the fundamental matrix using at least eight point correspondence.

Essential matrix E can be recovered as follows:

$$x'^T K^{-T} E K^{-1} x = 0, \tag{3.7}$$

where K and K' are the camera matrices which control the camera distortions for both cameras.

The epipolar line $l : Ax + By + C = 0$ in image of Camera 2 can be measured by:

$$l = E^T x, \tag{3.8}$$

where x is the undistorted point in the image of Camera 1 and E is the essential matrix.

The purpose of the research is to use the triangulation and multiview camera association in object detection and tracking for traffic safety. Therefore, the corresponding points detected in C_1 should match to the one in C_2. In this section, how the corresponding points between two camera view are detected is discussed and in Chapter 4, extended multiview tracking will be covered.

The epipolar line can restrict the search of corresponding point x on image of C_2 to a single line. However, searching the entire line for detection and tracking application is time and resource consuming. Hence, a homography-based method is proposed to find the best match to the point X on the line.

3.4.2 Establishing Point correspondence

According to the unknown parameters in multiview geometry, different strategies can be applied (Chien, 2018). In this study, camera motion (R, t) are estimated during the stereo calibration and x, the 2D point on C_1 is known. However, the image correspondence is still undisclosed. Hence, the unbounded 1D search on epipolar line should be performed and then, using triangulation, 3D coordinates of the scene point X can be recovered.

Rectification and feature matching are the standard approaches to establish the correspondence to find the direct matching between x and x', however in case of cross camera view where the objects might look differently, standard photometric matching algorithms fail in setting up the correspondence. In the existing literature, the cameras' rotation and transformation matrix are too limited, hence extracting the matching points deals with fewer difficulties. Generally, the extended overlapping field of view of the cameras results in more robust point and object correspondence.

Given the four corresponding points from the cameras' images and the scaled location of the top-down view map, the transformation matrix is calculated.

Figure 3.11 shows the location of two cross-view cameras looking at the same region and how the corresponding points of a rectangle are selected for measuring the transformation matrix. The red boxes on the images show the polygons that are selected from each camera FOV and the projected rectangle on the warped image. In this section, these two cameras are chosen for explanation, because extraction of the same feature of these opposite FOVs is sophisticated. As seen in Figure 3.2, the objects in C_1 and those in C_2 and C_3 shared more feature points compared to ones in C_4.

The concentration of this section is to provide a solution for a single point analysis for later safety analysis.

Firstly, the transformation homography matrix is calculated for both cameras to

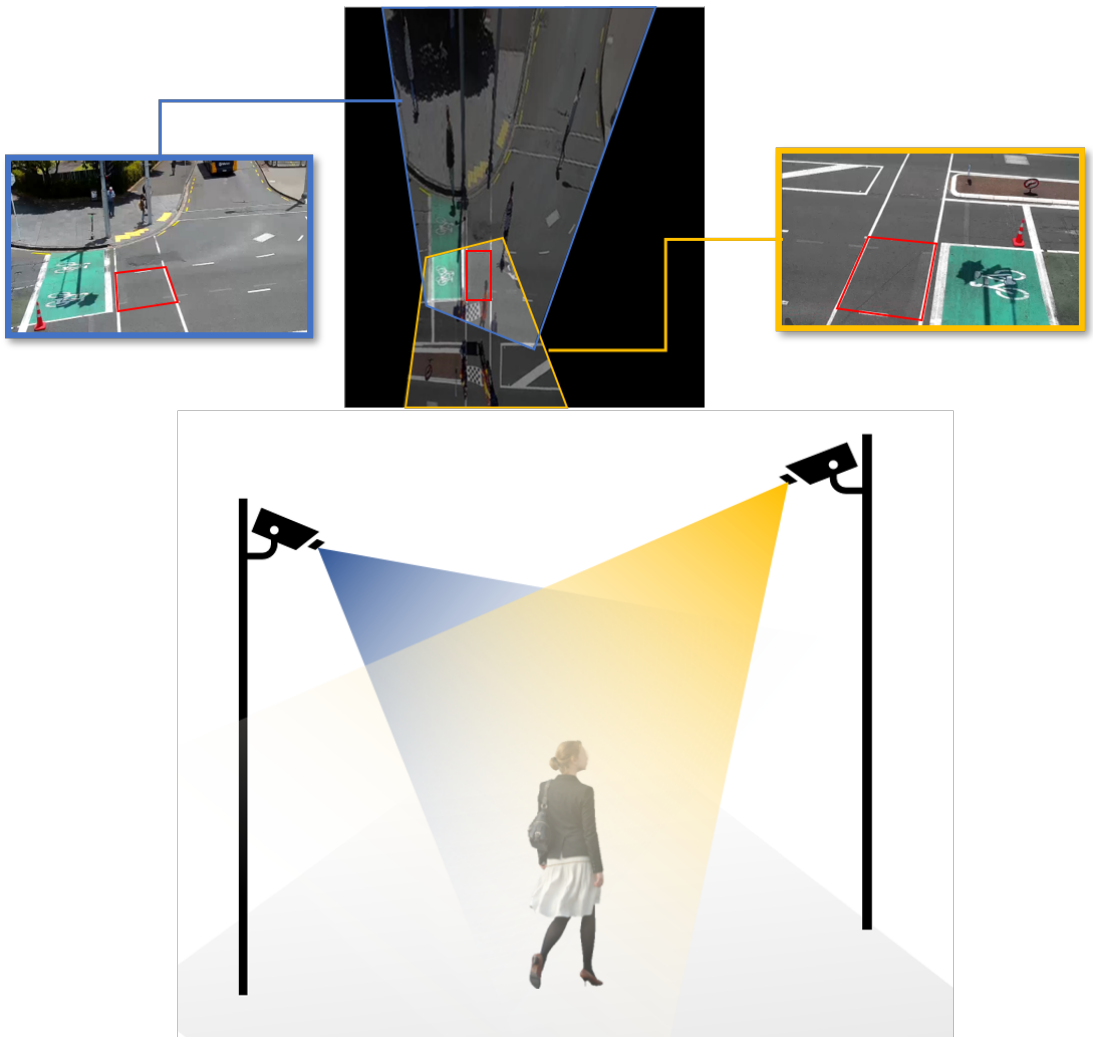


Figure 3.11: Position of the camera 1 and 4 (cross FOV) and homograpgy estimation

transform the images into the same top-down plane (the method will be discussed in Section 5.2.1). The undistorted decision point x (here, the centre bottom point of the bounding box of the object) is transformed by the transformation matrix H_1. The calculated epipolar line, corresponding to x is also warped into the plane using the transformation matrix of the second camera H_2. $\hat{x}$ and $\hat{l}$ are the warped corresponding point of x and l into the same top-down view plane.

The shortest Euclidean distance d between the warped undistorted point $\hat{x}$ and warped epipolar line $\hat{l}$ are calculated. Considering the reprojection and synchronisation errors, a threshold θ is defined. θ is defined to control the search area according to the

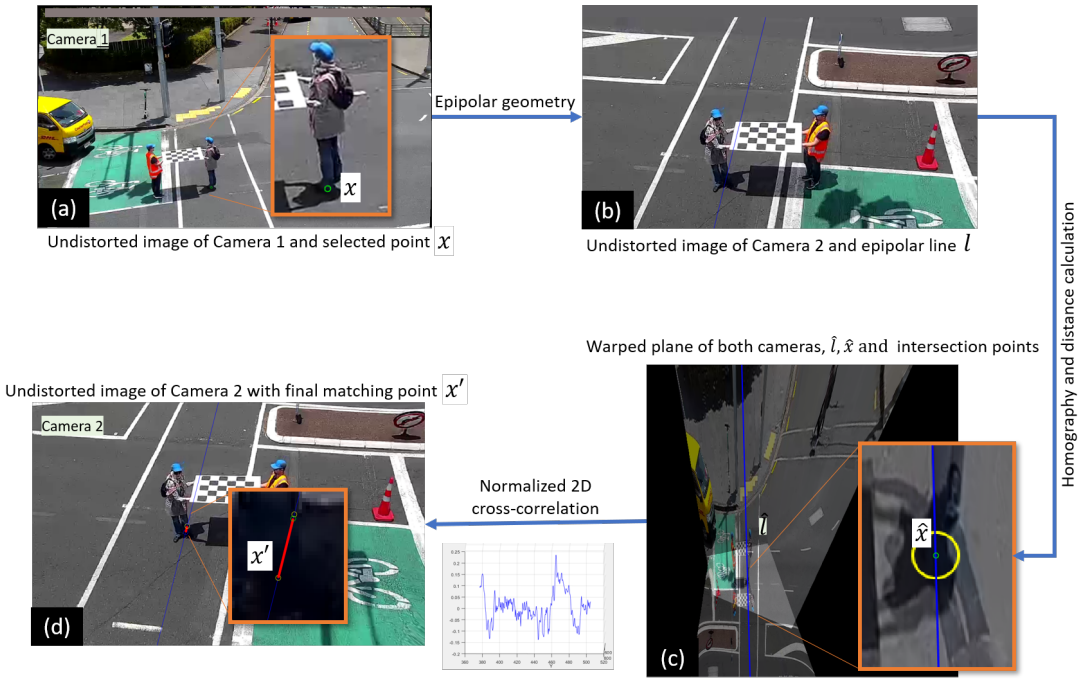


Figure 3.12: The process of finding the matching points by searching epipolar line

camera and the object class of x.

Given a circle with radius $d \times \theta$ and centre point $\hat{x}$, the intersection points of circle and $\hat{l}$ are calculated. These two points are later inversely transformed by the same transformation matrix of the second camera H_2 to the original undistorted image.

Normalised 2D cross-correlation (Yoo & Han, 2009) is applied to search the epipolar line on second camera's image. For this purpose, a square template of size $\alpha \times \alpha$ is selected with x being the centre point. The maximum correlation coefficient along the epipolar line segment which is restricted by two intersection points specifying the spatial coordinates of the endpoints of the line segments is selected as the matching point of x in the second camera. As a result, the undistorted matching point x' is detected in C_2.

The procedure as mentioned earlier, is illustrated in Figure 3.12. Two synchronised frames are selected from two cameras. Point x that is the foot position, is selected for the illustration purpose.

The decision on a point that represents the object is essential especially because the process of detecting x' relates to the 2D position form the bird's eye view. For this study, according to cameras' FOVs, for all object type, we choose the bottom centre points, which is close to the plane.

Using the epipolar geometry, epipolar line is measured, which gives the desired search area for the corresponding point in C_2. Both images are later warped to the same 2D plane. The yellow circle in (c) centred by $\hat{x}$ (plotted in green) and radius of $\theta \times d$ is used to define the search area on epipolar line. For this example, $\hat{x}$ is so close to the line by 0.3197 and for clarity, we choose θ = 50. The intersection points of the yellow circle and blue line are then back transformed to the original image of C_2. Template Matching algorithm searches the epipolar line segment (showed by red line and restricted by blue points in (d)) for the highest correlation coefficient on the designated area. This point is considered as the matching point of $\hat{x}$ in C_2, denoted by $\hat{x'}$.

A stereo or 2D point triangulation function finds the 3D coordinates X of a scene points where $x \leftrightarrow x'$ are the corresponding points in C_1 and C_2.

To this end, the pose of the second camera with respect to the first one by (R, t) and the projection function π_1 and π_1 are assumed to be known. The image correspondence of $x \leftrightarrow x'$ generates some error and ideally, the back-projection of x' never meet the actual point X in 3D space. Therefore, the direct linear transform is used to minimise the error of the back-projected $X'$and X.

3.5 Experimental Results

Figures 3.13, 3.14 and 3.15 show some examples for the proposed point correspondence algorithm between C_1 and other cameras for a pedestrian and a vehicle.

The left part of each figure shows the central bottom point from a detected bounding box, and following the process as mentioned earlier, the corresponding points are

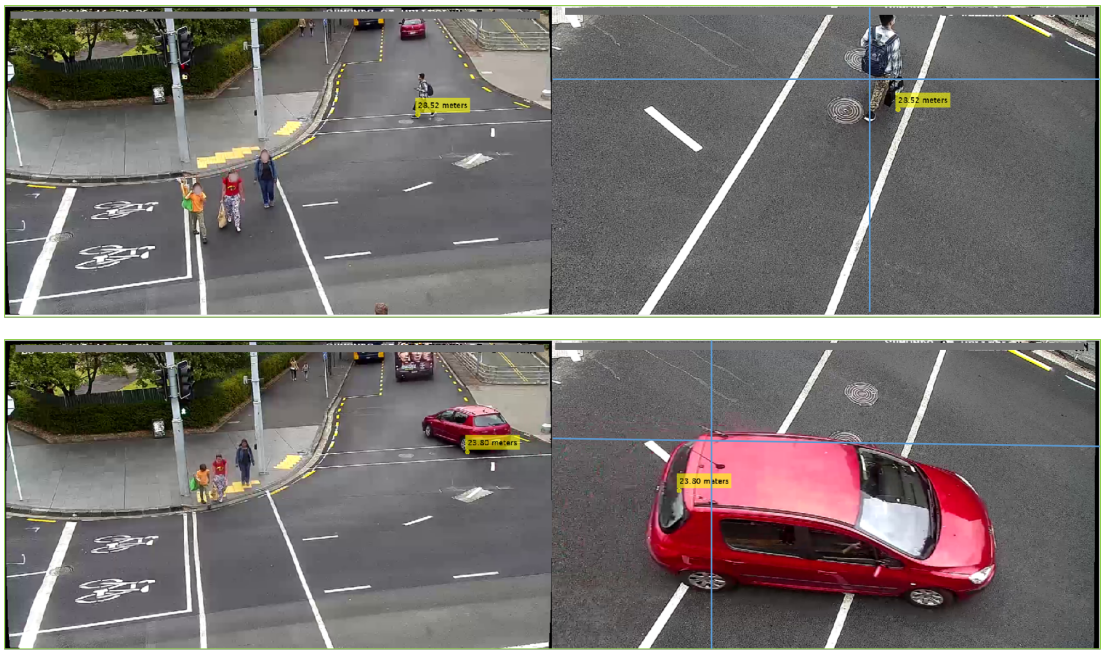


Figure 3.13: The points correspondence between C_1 and C_2

estimated in other cameras, displayed in the right part.

For more straightforward analysis, the world coordinates distance in meters are calculated from the corresponding points to C_1 through the triangulation method. The intersection of the blue lines shows the generated ground truth data for the actual points correspondence for qualitative and quantitative analysis.

As can be observed, the generated point x' for a pedestrian is closer to the corresponding ground truth, comparing if the object is a vehicle. It is due to the fact that the proposed point correspondence solution depends on the top-down view. In particular, when the vehicles are converted into a 2D top-down view, their features expand on the plane; therefore, the single point cannot represent the vehicle robustly. However, in the context of the safety and decision point analysis in Chapter 5, the selected points are useful to establish the parameters when the movements of the objects are of concern.

Table 3.3 compares the accuracy of the point correspondence for pedestrian and vehicle class. Note that this value is based on the image size 1280×720 pixels.

The average reprojection error (RPE) in pixels is calculated to find the accuracy

Figure 3.14: The points correspondence between C_1 and C_3

of the proposed point correspondence in two object classes. In order to have a better experiments, the ground truth data for 10 synchronised frames for each camera pair are defined to be compared after establishing the $x \leftrightarrow x'$ correspondence. Therefore, x for the objects in C_1 and their corresponding points in another camera are selected subjectively, and the Euclidean distance between x' and ground truth are measured.

The experimental results suggest that for all camera pairs, the point correspondence is more accurate when the object is being detected as pedestrian. It is because of the vertical shape of the pedestrians versus vehicles. Furthermore, as expected from Table 3.2 which shows higher RPE in C_2 stereo calibration, the average RPE for both vehicles and pedestrians are higher in this camera, however, the result is promising for the tracking and the same object movement as will be described in next chapter.

In an attempt to start the multiview tracking, the value θ, which was a threshold to define the search area, plays a critical role. Therefore, the optimum value for different object class is retrieved by multiple trials.

The suggested values of θ for each object class per camera is shown in Table 3.4.

Figure 3.15: The points correspondence between C_1 and C_4

Table 3.3: The average re-projection error for the proposed point correspondence algorithm

Camera Number	Av. RPE (Vehicle)	Av. RPE (Pedestrian)
2	58.61	43.02
3	8.93	6.76
4	23.45	12.45

The optimal value of θ is selected according to the camera FOV and the object type. In particular, if the object is large, the search area on the epipolar line should be expanded to cover more area of the object. However, this value should be set up to confirm that the search area is inside the exact cameras. In Chapter 4, the decision on cameras will be explained.

In general, as C_1 and C_3 view any object from the same perspective with the lowest average RPE, the variety of value θ can be robust. However, to consider the complexity, this value is set to be minimum. And this value for C_2 is chosen to handle the trade-offs

Table 3.4: The optimum value for θ per camera

Camera Number	Vehicle	Pedestrian	Heavy Vehicle	Cyclist
2	20	10	20	12
3	15	5	30	10
4	25	10	30	15

between the camera field of view and accuracy.

Chapter 4

Multiview Detection and Tracking

4.1 Abstract

The general overview of the front-end multiview detection and tracking approach for this study is illustrated in Figure 4.1.

The improved multiview detection and tracking methods are described in three sections. In Section 4.2, I explain the data preparations and how the videos are labeled for deep learning training.

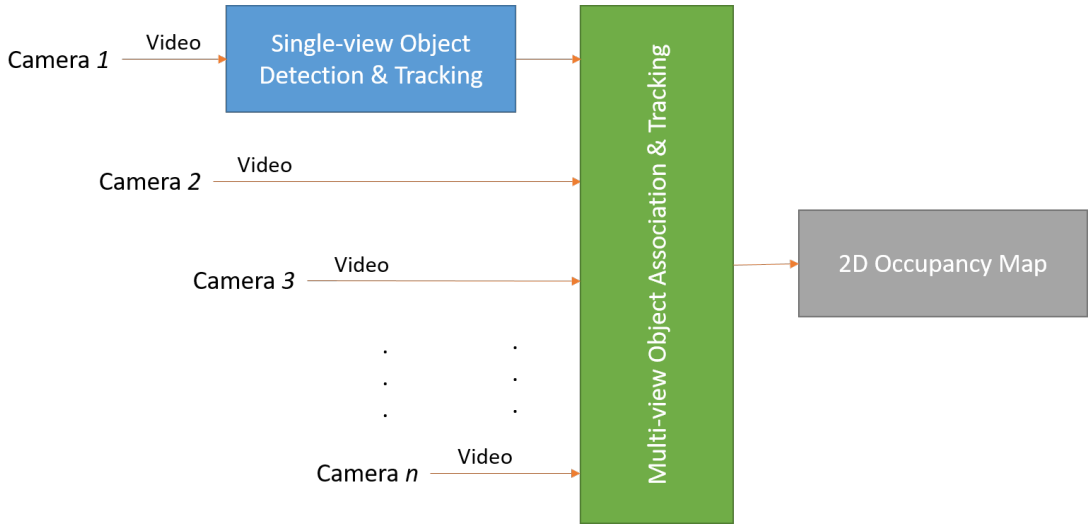


Figure 4.1: The overall framework of multiview detection and tracking

Section 4.3 summarizes detection and tracking in single view and Section 4.4 describes the object association from multiple view into a single occupancy map.

4.2 Dataset Preparation

To have an extensive analysis of safety issues for different traffic users, multiple sequences are used for labelling. The existing and available datasets do not focus on all of the objects we need. Mainly they focus on single object type whether it is a pedestrian, vehicle or special vehicles such as a bus or heavy truck and they do not include the scenarios where multiple object types are involved. For different contexts of this thesis, data from multiple roads are used as sample data for training purposes.

One of the sets of videos is collected from an intersection in Jinan, China. Size of the original videos are 720×1080 which are downsized to 480×720.

The majority of data are generated from the cameras located in Auckland, New Zealand. To have a better understanding of user behaviours, some are chosen in urban while some are located in rural areas. Since the outcome of this study focuses on the intersection that we can implement multiview tracking, some ground truth data are generated from the mentioned cameras too.

Figure 4.2 illustrates some sample frames that I used in this study for training the network.

Two types of datasets are created for this study. One is for re-training the ResNet-50 in which the input of the network is the cropped RGB images containing only desired objects, and one for localisation of the objects in the image.

For the first dataset, a combination of Gaussian Mixture Model (GMM) (Stauffer & Grimson, 2000), blob analysis and morphological operations are applied on each video frames, and then potential objects are nominated. Then the cropped objects are sent to a pre-trained network (Krizhevsky et al., 2012) to classify the object. As the

Figure 4.2: Samples of the video frames used for training

network is not trained on the same dataset, the images need manual observation after the process of classification. After the potential images are collected and categorised, the new regression network inspired by ResNet-50 (He et al., 2016) is trained which will be explained in Section4.3.

The labelling for CNN-detector, YOLOV2 , is also done in a semi-automated manner. The moving objects are first detected by means of a GMM and a pre-trained Aggregated Channel Features (ACF) (Dollár, Appel, Belongie & Perona, 2014). The automatically labelled objects are further refined by manual labelling. The pre-trained ACF only can detect the people and vehicles; however, further class refinement is needed for differentiating heavy vehicles. Also, cyclists have to be differentiated with people.

4.3 Single-view Detection and Tracking

The general detection and tracking framework is illustrated in Figure 4.3. Initially, the locations of the objects are estimated by a CNN-based algorithm. The ROI refinement process, which is inspired by our paper (Chien, Moayed, Zhu, Zhang & Klette, 2019), is modified to be used in the point-based tracker. In case of losing an object in consecutive frames, feature matching is applied to re-locate the lost objects. Note that this process is only applied on a single camera FOV and multiview object association will be explained in the next section.

The proposed approach is a fusion of tracking by detection and detection by tracking. Notably, we take advantage of detection to initialize the object position in frame j. Also, detection and classification confirm the tracking accuracy in Frame $j + k$ where $k \in \mathbb{N}_{>0}$ and it shows the initialization duration in frame numbers to preserve the tracking accuracy.

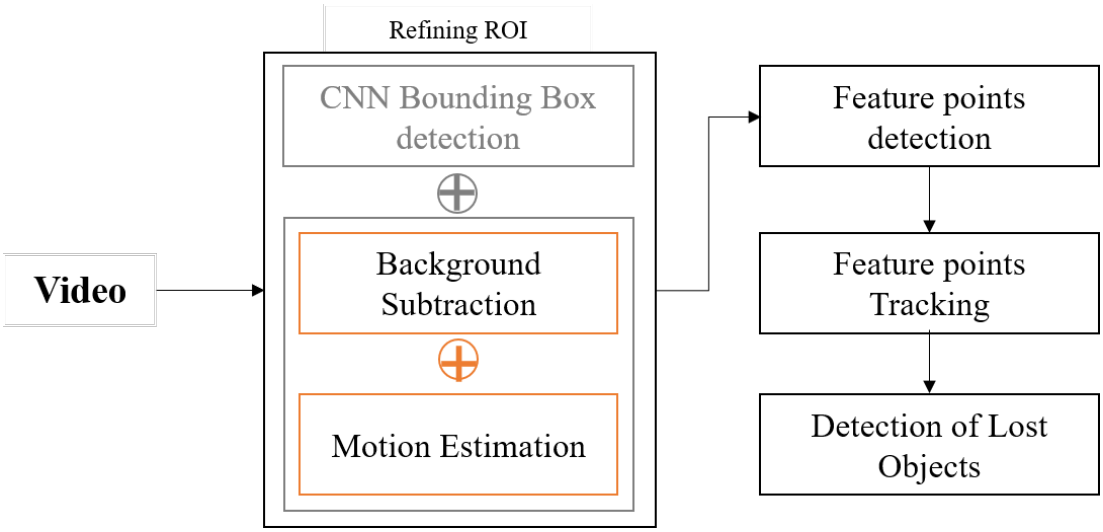


Figure 4.3: Single-view tracking framework

4.3.1 ROI Creation

Current object detection and classification algorithms using convolutional neural networks strongly depend on the number of data. Thus, preparing a dataset plays a critical

Figure 4.4: The YOLO V2 network with ResNet-50 as base network and Activation-Relu 40 as feature extraction layer

role in having an accurate algorithm. Dataset preparation is explained in Section 4.2.

Particularly, the input of a CNN-based object detector is in the form of RGB images, whereas the ground truth data contains the bounding box information and label of the objects.

Figure 4.4 illustrates the structure of a CNN model that is used in this study. The initial convolutional layers are used to extract the features followed by fully connected layers which determine object probabilities and bounding box coordinates.

The network architecture used to extract the features in this study is inspired by ResNet (He et al., 2016) of depth 50 blocks. Each block in ResNet-50 consists of 3 layers while ResNet blocks with lower depth usually consist of 2 layers (Figure 4.5)

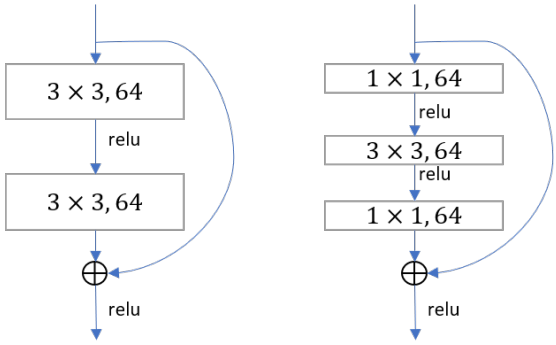


Figure 4.5: The building blocks of ResNet network. Right: 2 layer block, Left: 3 layer blocks used in ResNet-50

To detect and classify multiple objects, a pre-trained ResNet-50 network on ImageNet (Deng et al., 2009) is reused for transfer learning. The last fully connected layers and the final classification layer are replaced with new layers. The new fully connected layer is set to include the required number of the classes which is 4 in our contexts. The classes used in this research are:

$Class = \{People, Vehicle, HeavyVehicle, Cyclist\}$, where motorcyclists are also considered as cyclists.

The cropped objects from the videos are resized to fit the input size of the network. Data augmentation is applied to resize the images to $224 \times 224 \times 3$ and rotate the images

to prevent the network from overfitting.

The re-trained ResNet-50 is then used as the base network for YOLO V2. The ResNet activation RELU-40 is selected as the feature extraction layer. The ground truth data undergo some pre-processing steps, mainly resizing to fit the network. The anchor box sizes are then estimated using *K-means* clustering (Lloyd, 1982) algorithm with intersection over union (IoU) distance metric. The use of an anchor box in YOLO structure improves the computational speed and efficiency.

Experimental Results - Object Classification and Detection

The detection experiment is carried out on a system equipped with Nvidia GeForce X1060 © GPU.

Two experiments have been done to illustrate our testing results: the computation time and accuracy for the feature extractor and the classification networks, and detection accuracy of YOLO V2.

Table 4.1 shows the number of images we have used to train the ResNet-50 using transfer learning. For this experiment, we used 70% of the total images for training and 30% for validation. Stochastic gradient descent with momentum (SGDM)(Qian, 1999) optimizer is used for this training.

Table 4.2 shows the comparison of two different training options that we used to train our classifier.

Table 4.1: Number of images used for training ResNet-50 network

Class	# of training Images	# of validation Images
Person	560	240
Vehicle	480	144
Cyclist	480	144
Heavy vehicle	144	60

Table 4.2: Accuracy and speed results of training Resnet-50

Training Options	Training Time	Training Accuracy
Number of epoch= 8 Learning rate= 0.0001 Iteration= 312 Iteration per epoch = 39	30 min 29 sec	92.81%
Number of epoch= 32 Learning rate= 0.0001 Iteration= 1568 Iteration per epoch = 46	144 min 15 sec	95.21%

Figures 4.6 and 4.7 show the training progress in terms of accuracy and loss per iteration for two different training options. The goal of all CNN algorithms is to minimise the loss functions over each iteration. The loss function for Softmax and classification layer is calculated as

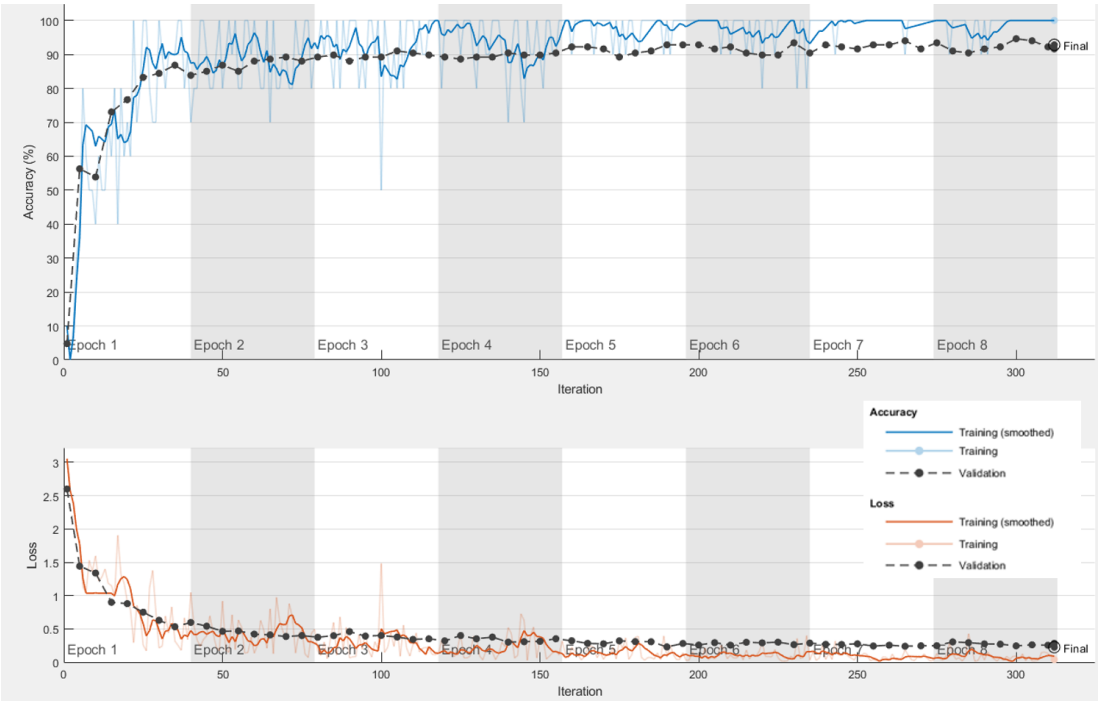


Figure 4.6: Training loss and accuracy at every iteration using option 1

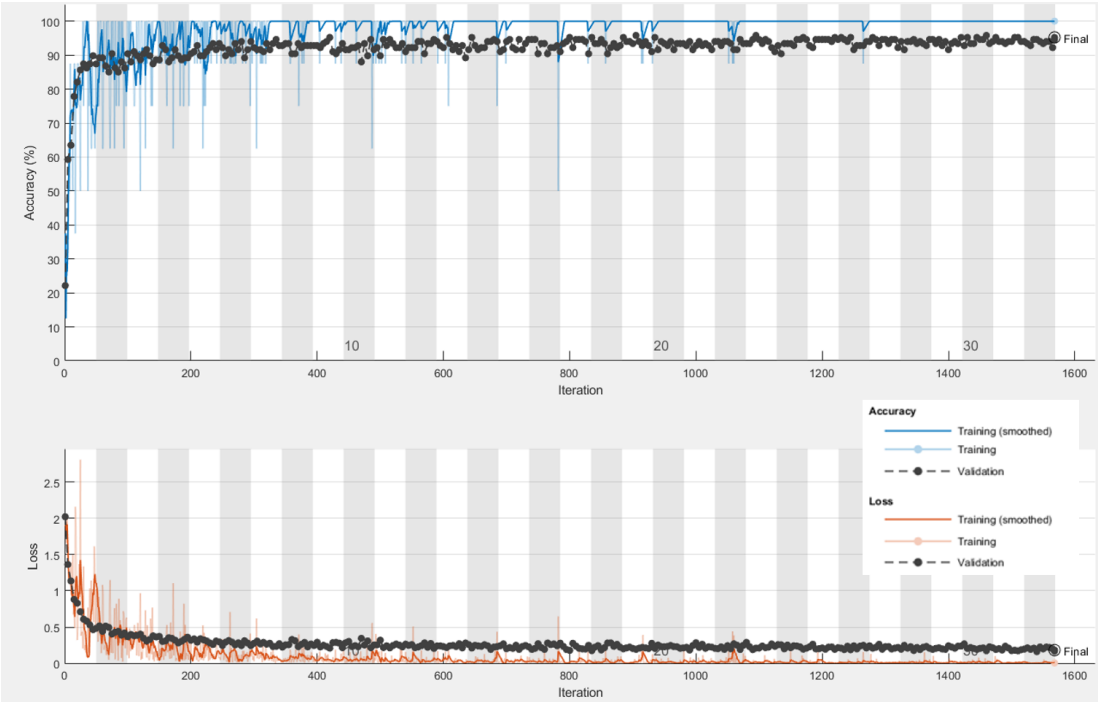


Figure 4.7: Training loss and accuracy at every iteration using option 2

$$loss = -\sum_{i=1}^{N}\sum_{j=1}^{K}\lambda_{ij}\log y_{ij} \tag{4.1}$$

where N and K are the number of samples and classes, respectively and $K = 4$ in this study. $\lambda_{ij} = 1$ if and only if sample i belongs to class j and y_{ij} is the output probability.

The training options used in the second trial shows that despite taking more time to train the classifier, the training accuracy is higher by 2.4%.

The main objective of this thesis is to provide accurate tracking to be used for multiview traffic safety analysis. Therefore, for the estimation of the accuracy and time complexity of the object detector used in this study, we only consider the training accuracy and elapsed time. This is because objects tracking which will be discussed later firmly depends on object detection.

To train YOLO V2 for this study, as details are mentioned in Section 4.2, 500 frames are semi-manually labelled.

4.3.2 CNN-based Tracker

Let $y_{i,j} = f(x_i, I_j)$ be the observation of object x_i in frame j and $I_j : \Omega \rightarrow \mathbb{R}^C$ is the image of frame j in domain Ω with C channels.

Given a set of $y_{i,j-\Delta t}$ and $I_{j-\Delta t}$ where $\Delta t \in \mathbb{N}_{>0}$, an observation function f as well as a binary function $z_{i,j} \in \{0, 1\}$ that yields 1 when object x_i is visible in frame j and 0 otherwise, are estimated. Therefore, the validity of the observation $y_{i,j}$ in the current frame j is identified. In case $\Delta t \in \{1\}$, only the most recent frame I_{j-1} is used, which is the simplest form of the problem.

Observation $y_{i,j}$ contains some feature descriptors of object x_i extracted from I_j, among which a commonly used one is a rectangle enclosing object x_i in Ω of frame j.

A favourable choice of such rectangles aligns with the axes of Ω is known as axis-aligned bounding box (AABB in Figure 4.8). In computer vision, AABB is widely accepted as the choice of bounding box-based detection; however, as it is axis-aligned, the fitness of the objects in the bounding box is not necessarily achievable.

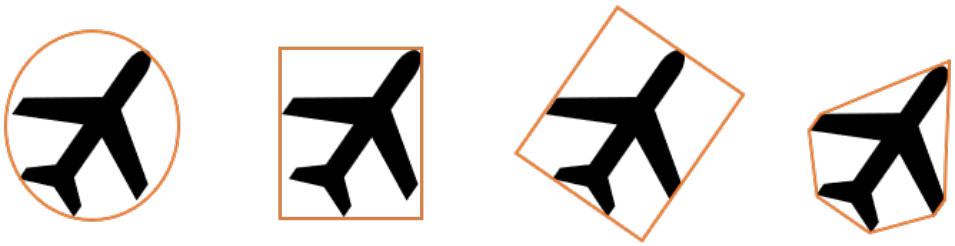


Figure 4.8: Left to right: sphere, axis-aligned bounding box (AABB), oriented bounding box (OBB) and convex hull. CNN object detection methods yield AABB style of bounding box

The details of CNN bounding box detection and classification are explained in 4.3.1, hence, we only consider the output of the detector in Frame I_j as $\beta(y_{i,j})$.

The extraction of bounding box from $y_{i,j}$ can be symbolised by $\beta(y_{i,j}) = (u, v, w, h)^\intercal$, where $(u, v) \in \Omega$ denotes the upper-left corner of the box, and $w, h \in \mathbb{N}_{>0}$ are box's

(a) Best-fit box (b) Over-sized box (c) Under-sized box (d) Missed box

Figure 4.9: Different conditions of detected bounding box.

width and the height, respectively.

Given S, the point set that contains pixels of object x_i, the best-fit bounding box (u, v, w, h) satisfies

$$\forall (p, q) \in S, p \in [u, u + w) \ \wedge \ q \in [v, v + h) \tag{4.2}$$

while minimising $\mathscr{A} = w \cdot h$, the area of the bounding box.

Decision on the conditions of the bounding box is made as follow:

In Equation 4.2, the closure condition is met When $\mathscr{A}$ is not minimised and the bounding box is over-sized. When the condition is partially met; the bounding box is under-sized and the object is cropped. In case the condition is not met at all, the detection is missed or a false positive.

Figure 4.9 shows the example of each condition.

The proposed approach enhances bounding box estimation, employing multi-frame background modelling and motion analysis. The initial estimates are completed by applying a CNN-based object detection network on the image of the current frame I_j. The estimated bounding boxes are then authenticated by an adaptively learned foreground image, following a motion-based outlier rejection algorithm.

The hybrid tracker is a fusion of tracking by detection and detection by tracking. Notably, we take advantage of detection to initialize the object position in frame j. Also, detection and classification confirm the tracking accuracy in Frame $j + k$ which $k \in \mathbb{N}_{>0}$ and it shows the initialization duration by frame numbers to preserve the tracking

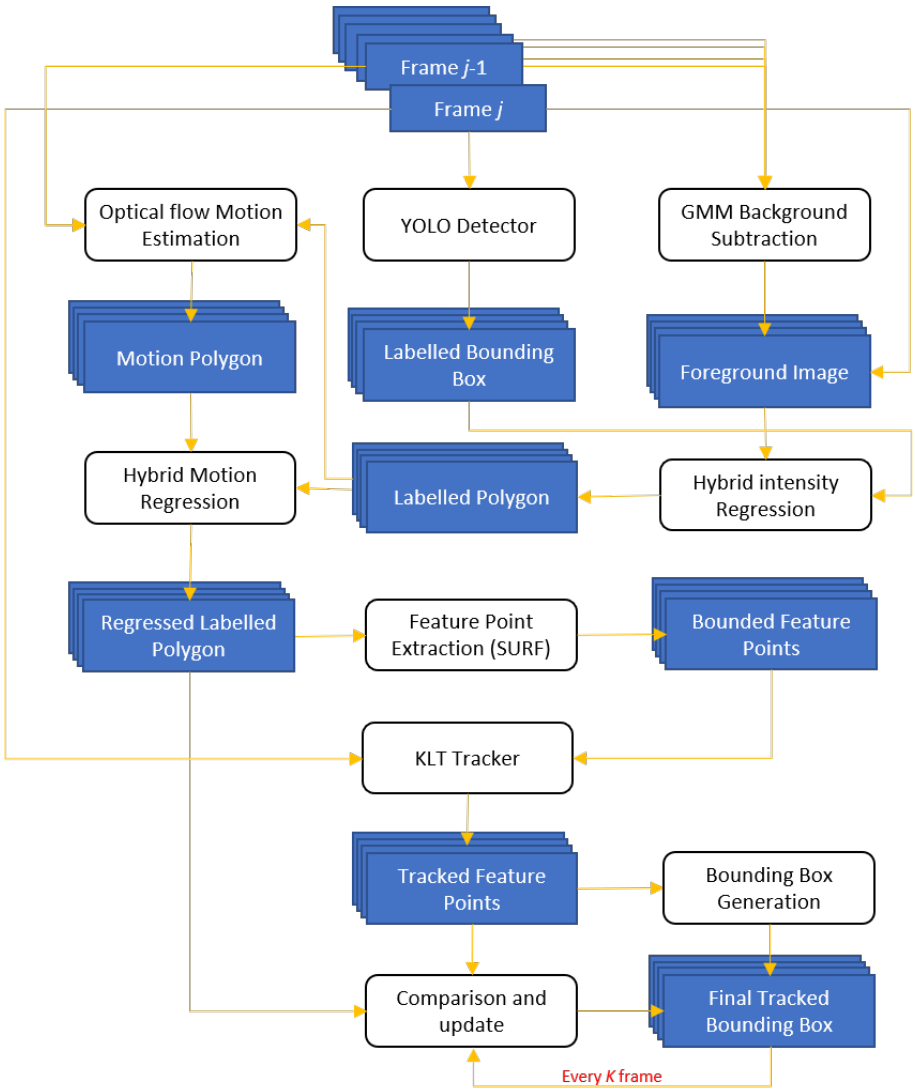


Figure 4.10: Proposed Tracker

accuracy.

Figure 4.10 illustrated the overall flow of the tracking algorithms with a more details which will be described in the following sections.

Bounding Box Refinement and Motion Analysis

The results of YOLO Regressor are objects enclosed in the bounding boxes. The main issue that CNN-based trackers face is the initial location of the object is not accurate for further tracking. The majority of the output bounding boxes generated by YOLO

are either over-sized or under-sized. Also, in Multi-object detection, the possibility of having missed object or false positives are considerable.

The tracker that we used in this study takes advantage of the improved initialisation. For that, we address the issue of location initialisation in CNN trackers which is mainly ignored by other approaches. CNN-detectors detect the object by providing bounding box, and since they are based on sliding windows or anchor boxes with specific sizes, the localisation accuracy is not acceptable enough to be used as the sole main features. This leads to incorrect tracking in the following frames, especially in a complex background.

The improvement of the proposed method relies on accurate shape descriptor estimation for each object of interest. The reason that we segment the object in each bounding box is to refine the tracking algorithm. Therefore, as we are detecting the objects in the temporal domain, the motion descriptors can provide the most dominant features. Also, this thesis aims at analysing the safety of traffic users in a complex intersection in which, all objects are moving.

For the first stage of the detection framework, the region proposals based on the change of image intensity is investigated. To achieve better accuracy, we approach each region proposal from per-pixel analysis first, which then forms bounding polygon and eventually the best-fit bounding box.

Background subtraction is still considered as an effective way to distinguish the moving pixels. A multi-channel variation of the Gaussian mixture model (GMM) (Stauffer & Grimson, 2000) is adopted to distinguish object pixels from a constantly updated background image robustly. In RGB colour space, the assumption for GMM is that red, green and blue components of each pixel are independently distributed.

$$N_{p,c}^{k} = (\mu_{p,c}^{k}, \sigma_{p,c}^{k}) \tag{4.3}$$

where $N_{p,c}^{k}$ is the k-th distribution of channel $c \leq C$ at pixel $p \in \Omega$, the pixel is assigned

to the model, providing a new frame I_j if and only if

$$|I_j^c(p) - \mu_{p,c}^k| < n \cdot \sigma_{p,c}^k \tag{4.4}$$

where I_j^c is the c-th slice of image I_j and $n > 0$ is a constant threshold.

Pixel p is a background pixel in I_j^c if the model achieved the closest Gaussian distance to $I_j^c(p)$ normalised by its standard deviation. Otherwise, p is considered to be a foreground pixel.

Blob analysis is then applied to enhance the results generated by GMM, followed by morphological operations to achieve large enough, yet as distinct as possible foreground objects. Let Φ_j be the binary conversion of the background and foreground pixels, in which 0 and 1 represent the background and foreground pixels, respectively. Here, we define the foreground mask as $\varphi_{i,j} \in \Phi_j$ which shows the foreground mask i in Frame j.

YOLO regression network is applied to I_j to retrieve a set of observations, denoted by $\hat{Y}_j$. For each $\hat{y} \in \hat{Y}_{j-1}$ the best match candidate is found from previous observations Y_{j-1}, following

$$\underset{y_i \in Y_{j-1}}{\arg\max} \{v_{i,j-1} \cdot \Big(\mathcal{I}_{j-1}(y_i) \star \mathcal{I}_j(\hat{y})\Big)\} \tag{4.5}$$

where $v_{i,j}$ is a binary term indicating if object x_i is observed in j-th frame, $\mathcal{I}(y)$ is the normalised zero-mean 1-σ image patch scoped by $\beta(y)$ the bounding box of y, and $\star$ denotes cross-correlation operator. The association is then verified by introducing a distance check. If y_i is too far from $\hat{y}$ in Ω, then the correspondence $y_i \leftrightarrow \hat{y}$ is rejected.

For those $\hat{y}$ which do not have any positive match from previously tracked objects, new entries are created and appended to Y_{j-1}, forming a new observation set Y_j.

Φ_j and Y_j are acquired by two independent procedure. To take more advantage of motion analysis, further post-processing is applied using the concept of optical flow (Horn & Schunck, 1981).

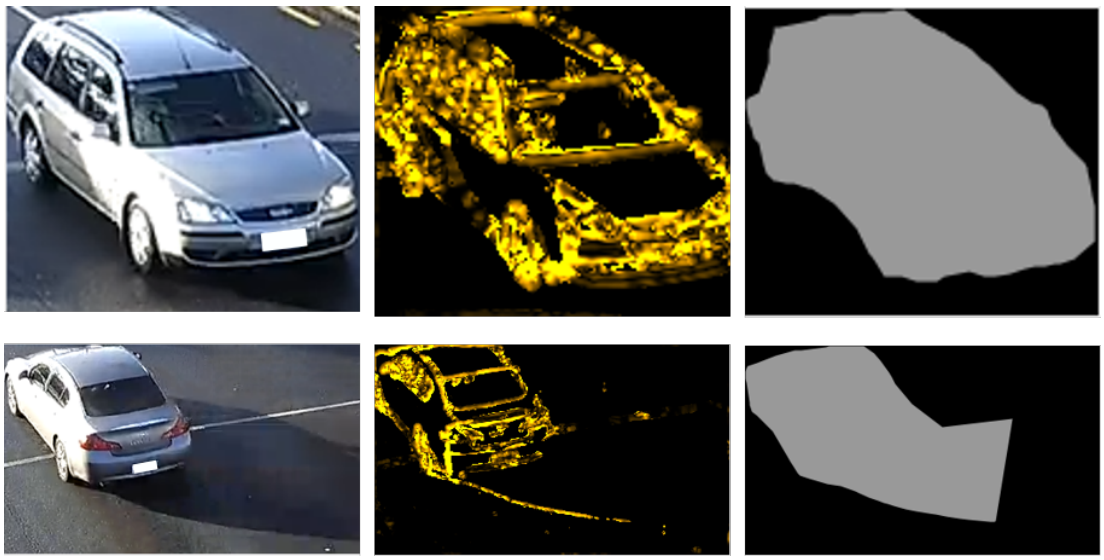


Figure 4.11: Left: RGB Foreground object, centre: Magnitude of pixel movement of the foreground object, Right: Refined foreground polygon

The bounding polygons of φ are again refined to include more motion analysis. Pixels with significant movement are selected to form a point set, on which an α-shape algorithm (Edelsbrunner, Kirkpatrick & Seidel, 1983) is deployed to shape a bounding polygon. Figure 4.11 shows two examples of the motion-based refinement of foreground regions.

As seen in Figure 4.11, relying on the motion parameters in the pixel level is not appealing for all scenarios such as the object in the second row. The background subtraction and optical flow consider the variation of the pixels in each frame; thus any moving pixel is considered as the nominated regions. To be more precise in that example, since the shadow is moving, all pixels on that region are considered to be changed. Hence they are considered as the foreground objects.

The advantage of using YOLO or any CNN object detectors in tracking is that the system works on the object level. The fusion of the bounding polygon in pixel level and object level can increase the probability of accurate extraction of the feature points.

For every observed $y \in Y_j$, a bounding box refinement is confirmed by considering

(a) People

(b) Vehicles

Figure 4.12: Results of bounding box refinement for tracking. Red box and its corresponding polygon is used for tracking

the overlapping region proposal $\varphi \in \Phi_i$ by

$$\arg\max_{\varphi \in \Phi_j} \{\mathscr{A}\Big(\beta(\varphi), \beta(y)\Big)\} \tag{4.6}$$

where $\mathscr{A}(B, B')$ calculates overlapped area of bounding boxes B and B'. If the overlapping area is larger than a threshold, the correspondence $y \leftrightarrow \varphi$ is accepted, and the bounding box of y will be updated by φ.

The outcome of the aforementioned steps is illustrated separately in Figure 4.12. The magenta bounding box represents the results of trained YOLO for person and vehicle. Blue and Cyan polygons denote the foreground masks and motion polygon from the Optical flow, respectively. And the desired final box is shown with red colour. In general, we restricted the "the good features to track"(Jianbo Shi & Tomasi, 1994) to

be selected on the most desired areas.

Upgrading the Tracker

In general, motion-based object detection algorithms such as GMM and optical flow cannot handle the occlusion and unwanted object movement such as shadows. Also, they do not classify the objects. On the other hand, a CNN-based object detector suffers from step-wise detection, which leads to inaccurate detection. With merging the output, the accuracy of the tracker is estimated to outperform the other independent algorithms.

Point-based tracking algorithms mainly rely on updating the position of feature descriptors of an object. The more accurate the feature descriptors are, the more accurate tracking algorithm is.

The region Z that feature points are extracted from is defined as:

$$\zeta_{i,j} = \underset{\varphi \epsilon \Phi_j, y \epsilon Y_j}{\arg\max} \{\mathscr{A}(\varphi, y)\} \tag{4.7}$$

where $\mathscr{A}(\varphi, y)$ is the overlapping pixels of CNN detector y and the motion-refined foreground φ.

Let $(p, q) \in \lambda$ be the point set that contains the most distinctive pixels of object ζ_i in frame j.

Although ORB (Oriented FAST and Rotated BRIEF) (Rublee, Rabaud, Konolige & Bradski, 2011) is the fastest among infamous feature extraction algorithms, in this study, we use SURF (Speeded Up Robust Features) (Bay, Tuytelaars & Van Gool, 2006) because of two main reasons:

1. There is size restriction in the ROI used in the implementation of ORB.

2. SURF is more robust to scale changes for feature matching and tracking

Algorithm 1 describes more details of the tracker, including the region refinement.

The video is proceeded frame by frame until the object $x_{i,j}$ is detected using YOLO. Since the process of ROI creation is considered as time-consuming to be applied in each frame, we perform the process of ROI refinement at every k frames. The value of k is selected based on the scene occlusion level and the required speed of the tracker.

After objects are detected, and potential ROIs are selected, the SURF feature points, denoted by $\lambda_{i,j}$, are generated inside the refined area for each object and each selected feature points are tracked using improved KLT (Tomasi & Kanade, 1991) tracking method. The tracker in frame $j + 1$ is updated by the bounding box, containing all tracked feature points $\beta(\lambda_{i,j+1})$.

For each frame, each object with its new representative, which is the refined ROI is fed into the multi-object tracker. Intersection over Union (IoU) function (Equation 4.8) performs the matching to confirm an object is a new object to be tracked or it belongs to an existing one. This process also refines the detection algorithm, especially in the first frames when multiple CNN-based bounding boxes represent the same object, thus only a single object is being tracked.

$$IOU(A, B) = \frac{(A \cap B)|}{|(A \cup B)|} \tag{4.8}$$

where $|A|$ is the cardinality of set A.

A different identifier is assigned to a different object. This incremental number can also identify the number of objects in the video by frame j.

4.3.3 Re-Detection of Lost Objects

The lost object can be re-detected using the same feature points that we used for our tracking. The decision of the lost object can be made by considering two assumptions:

1. The bounding box of the object in the last detected frame was not located close to the frame edge.

Algorithm 1: Proposed tracker algorithm

Result: $\beta(T_j)$, Tracked bounding boxes in Frame j

θ; /* θ is the threshold of IOU */
$T_j = \{\}$;
$r = 1$; /* r is object id */
while *Frame j exists* **do**
  instructions;
  if $\left(j \bmod k = 0, k \in \mathbb{N} > 0 \middle| k = 1\right)$ **then**
    $Z_j = \arg\max\{\mathscr{A}(\Phi_j, Y_j)\}$;
    d; /* d is the number of detected objects in Z_j */
    for $i \leftarrow 1$ **to** $d - 1$ **do**
      if $IOU(\zeta_{i,j}, \zeta_{i+1,j} > \theta)$ **then**
        $r \leftarrow r$;
      else
        $r \leftarrow r + 1$;
      end
      $\lambda i, j \leftarrow \{(p, q) | (p, q) \in \zeta_{i,j}\}$;
      $t_{i,j} = \lambda i, j$;
      $T_j \leftarrow t_{i,j}$;
    end
  else
    d; /* d is the number of detected objects in T_{j-1} */
    for $i \leftarrow 1$ **to** $d - 1$ **do**
      $\lambda i, j \leftarrow \{(p, q) | (p, q) \in t_{j-1}\}$;
      $t_{i,j} = \lambda i, j$;
      $T_j \leftarrow t_{i,j}$;
    end
  end
  $\beta(T_j)$
end

2. After k frame, the CNN-detector detects the object but with different class identifier.

Therefore in case of losing an object in the middle of the ROI, the updated version of the point matching algorithm (Muja & Lowe, 2009) is applied on Frames I_{j-l} and I_j to find the missing object in I_j. The lost object in I_j is re-detected using feature points of x_i by a bounded template matching over $R_{i,j} \subseteq \Omega$, where $R_{i,j}$ is a region derived from $y_{i,j-1}$ the last known observation of x_i.

If CNN-detector changes the object identifier after k frames, the possibility and the history of the detection are considered to confirm the class, however, with the distinctive classes that we have in this study, this situation is not encountered.

4.3.4 Experimental Results

Real-time video data captured from the available CCTV cameras in Auckland is used for our detection and tracking accuracy.

As the proposed tracker utilises the motion estimation, it can only work on static cameras such as surveillance cameras. The reason is that when the camera moves and shakes, foreground object detectors and optical flows are not able to detect the pixels of interest. Two types of experiments are done in this study. One is the quantitative assessments using the standard metrics used in object detection and tracking, and the other one is the qualitative assessment to define how much the bounding boxes are refined.

All experiments are carried out using MATLAB© 2019a with a machine equipped with Nvidia GeForce© GTX 1060 graphics card.

Quantitative assessment for object detection and tracking

In the context of traffic safety at road intersections rather than the classification accuracy, which discusses in Section 4.3, the correct localisation plays an essential role.

In order to compare the accuracy of the proposed approach using the refined polygon with state of the art methods, the ground truth data are generated semi-manually using the algorithms discussed in 4.3 by detecting the area of moving objects in 200 frames of the video.

To do the experiments on how the detected location of the classified objects are

Table 4.3: Detection accuracy in terms of IOU, MR and FPPI

	Dissimilarity Measure ψ	MR	FPPI	Ave. Time
Proposed	0.6	**0.1**	**0.07**	**0.43**
YOLO V2	**0.54**	0.18	0.07	0.45
Method in (Moayed et al., 2017)	0.75	0.21	0.09	0.68
Fast R-CNN & GMM	0.88	0.52	0.1	0.54

similar in the number of pixels to the available ground truth images, a *dissimilarity measure* is used, in which A is the detected refined bounding boxes in each frame and B is the bounding boxes in the ground truth image. The dissimilarity measure $\psi(A, B)$ is calculated as follow:

$$\psi(A, B) = \frac{|(A \cup B) \backslash (A \cap B)|}{|(A \cup B)|} \tag{4.9}$$

where $|A|$ is the cardinality of set A. It is a common used metric particularly when the location is of concern (Klette, Koschan & Schlüns, 2013).[1]

Table 4.3 shows the comparison of four different methods in terms of average values of dissimilarity measure, Miss Rate (MR) and False Positive Per Image (FPPI), compared with ground truth data. The lower value of the dissimilarity measure indicates that the number of overlapping pixels compared with ground truth frames are better by 0.06 in YOLOV2 with IOU-based tracking. The reason is that detection in our proposed method is based on the bounding box, which is formed after tracking the SURF points, and the CNN-detector is applied on every k frames. So the size of the bounding box in each frame may shrink in each track.

[1]The commonly used measure *intersection over union*,

$$\mathrm{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.10}$$

with $\psi(A, B) = 1 - \mathrm{IoU}(A, B)$ measures similarity; ψ is a metric (i.e. distance measure in the strict mathematical sense), but IoU is not.

Figure 4.13: The tracking results of two objects in 5 consecutive frames, where k = 4. From left to right, the tracked objects and their corresponding feature points to be tracked are shown and the right-most image illustrates the updated bounding boxes after CNN-based YOLO V2

Figure 4.13 illustrates the bounding boxes of two objects generated from SURF feature points in Frame I_j after tracking the points in I_{j-1}. As can be seen, the tracked bounding boxes became under-fit to include the essential points to track.

Miss Rate (MR) and False Positive Per Image (FFPI) are the other metrics that we use in the object-level to find the localisation accuracy.

One of the main objectives of this study is to deduct the number of missing objects in consecutive frames due to the sensitivity of the system dealing with humans' safety. Therefore, the MR is an appropriate metric to show the accuracy in terms of missing objects; the MR is defined as the ratio of false negatives compared to all true positives in the ground truth images. Likewise, we calculate the FPPI as the ratio of false positives compared to all detected objects of the frame in the object level. Both measures are calculated when the bounding boxes overlap by 70%in order to emphasise the localisation accuracy.

The lower values of both MR and FPPI represent our proposed tracking algorithm outperforms the other methods in terms of the number of missed objects and extra detection. The lower MR rate of our method is due to the fact that tracking of the objects helps to keep the objects in consecutive frames while the methods based on sole detection are not able to detect all the objects in all frames. The value of FPPI in the proposed algorithm is almost the same as YOLOV2, and the reason is that the incorrect detection is eliminated when the object is not tracked when the tracking score becomes lower than the threshold. Increasing the value of k has a positive impact on this value.

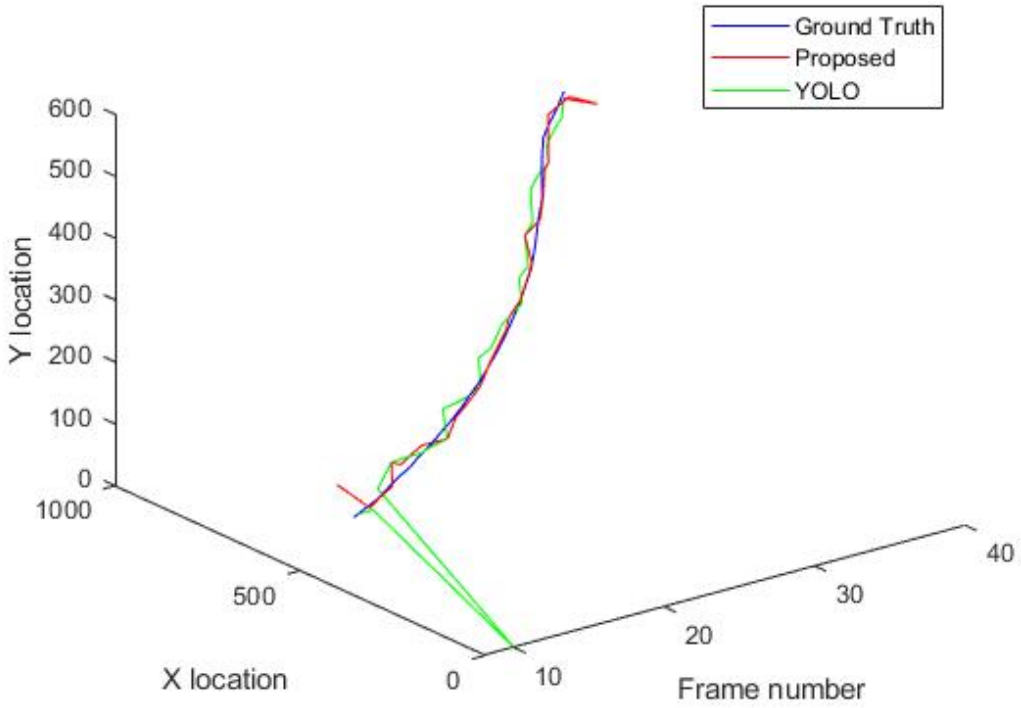


Figure 4.14: Comparison of the trajectory of an object in ground truth images, proposed tracker and YOLO.

However, it may lead to higher MR and dissimilarity measure.

One experiment is also done to compare the bottom centre point of the bounding box in some consecutive frames, comparing the ground truth object with the corresponding one in proposed tracker and YOLO. For this purpose, one object is considered, and the centre point of the detected box in each frame is generated in ground truth data, our proposed method and tracking by YOLOV2 detection. Despite lower IOU in pixel-level when comparing the bounding boxes, the average Euclidean distance between the selected centre points to those in ground truth data is smaller by 7 pixels in our proposed method. The experiment highlights the accuracy when the trajectory is of concern in the application; the proposed method outperforms the ones with CNN-detector in each frame. Figure 4.14 depicts the trajectory of the centre points of an object from when

it appears in the scene to its disappearance. As can be seen, YOLO detection fails to detect the object in frame 10. Best matching of proposed and ground truth curves in the middle section is due to tracking improvement that happens compared to YOLO detection.

One of the main objectives of the proposed method is to design an efficient tracker that is accurate yet fast enough. Although the region refinement is a time-consuming task in the proposed tracker, considering $k = 4$ the improved method is still 0.02 seconds faster comparing with YOLOV2. By increasing the value of k, we can achieve faster processing speed. To examine the effects of value k in the processing time, the algorithm is run on 100 frames of a busy scene against the different value of k. The results, displayed in Figure 4.15, suggest that by increasing the value of k, the processing time decreases significantly from 8.5 to 0.5 seconds when $k = 20$. Since the proposed algorithm tracks the objects while is being detected, the trade-off between missing the objects and processing time should be handled. Particularly, the object is being tracked while it is detected. Therefore, if k is too large, the object will be detected after CNN tracker detects the bounding box. However, for the tested video, the processing time reach a plateau for $k > 12$.

Qualitative assessment for Bounding Box refinement

To measure how the refinement process improves the detection accuracy, three video sequences, each 15 minutes recorded at 25 FPS at different time of a day are used.

The manual qualitative assessment of the bounding box accuracy is performed. To this extent, more than 900 frames have been selected randomly from 8 sub-sequences to evaluate the accuracy of the proposed single-view detection and tracking algorithm compared with the IOU-based CNN-detector.

The sequences are selected to cover complicated object movements, including occlusion of multiple objects moving to a different direction, as well as a variety of

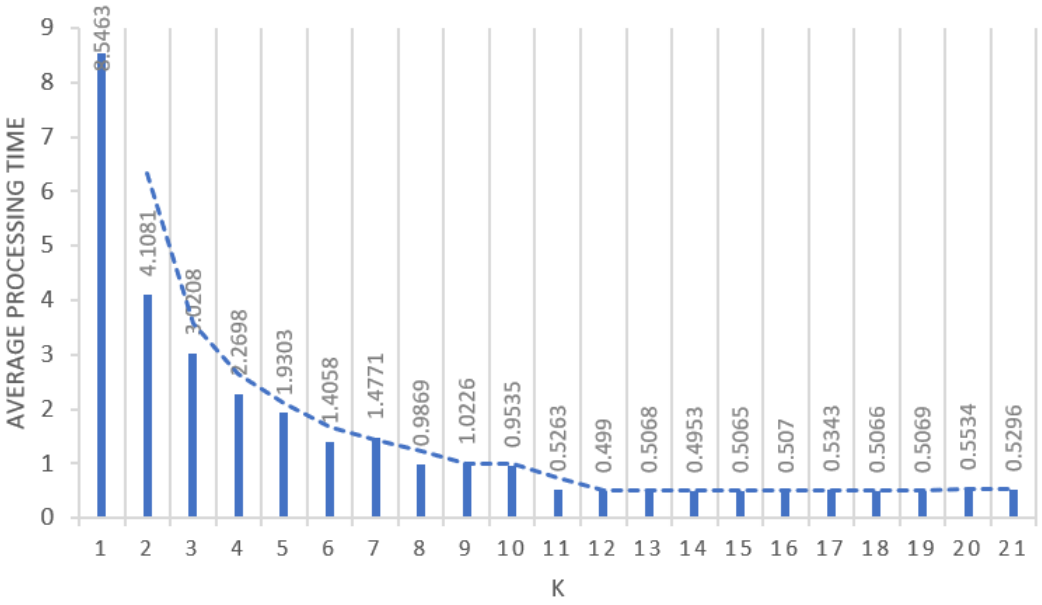


Figure 4.15: The average processing time compared in seconds to value of k

illuminations. The experimental results are tabulated in Table. 4.4.

Table 4.4: Qualitative experimental result for single-view object tracking

	Frames	Object Detector				Proposed Method			
		Fit	Over-sized	Under-sized	Missed	Fit	Over-sized	Under-sized	Missed
Seq. 1	104	0	0	92	12	101	3	0	0
Seq. 2	143	1	0	142	0	117	10	16	0
Seq. 3	123	3	0	99	21	103	10	5	5
Seq. 4	101	15	11	53	22	80	13	3	5
Seq. 5	205	7	0	198	0	149	17	39	0
Seq. 6	97	33	0	57	7	41	0	42	14
Seq. 7	84	2	3	52	27	84	0	0	0
Seq. 8	103	3	1	99	0	90	3	10	0
Total	960	6.67%	1.56%	82.50%	9.27%	76.69%	5.83%	11.98%	2.50%

The results suggest that the proposed method outperforms the CNN-tracker in the reduction of the number of under-estimated bounding boxes, from 82% to 12%. Also, the number of best-fit estimates is increased by more than 70%, from 6.7% to 77%. This increment is because of pixel-level analysis, which refines the regression of bounding boxes.

The proposed method also achieves better tracking of an object, given only 2.5% miss rate compared to the detector-only network, which missed 9.27% of objects in the

tested frames.

However, side-effects have been introduced by the motion analysis when detection of multiple objects are merged into one region proposal which results in the over-sized bounding boxes increasing the error by 5%. The advantages in terms of performance in best-fit estimates far outweigh the disadvantages with regard to generating 5% increase in over-sized estimates.

4.4 Multiview Object Tracking

Multiview trackers use different camera views to address the issues of single-view trackers. When the same object is viewed by multiple cameras, finding the correspondence between the object in different cameras is difficult due to significant changes in the shape, lighting conditions, occlusion with the similar objects and massive difference in size and scale (Nassar, Lefevre & Wegner, 2019).

As will be discussed in Section 5.2.1, the decision points of the bounding boxes depend on the object class and most importantly, the camera field of views. In the case of this study, the chosen point for all object type is selected as the centre bottom point of bounding boxes. Although this point might not be the best option for wide objects such as cars and buses, the multiview association perform better on this point in general.

The triangulation and camera relationships were discussed in Chapter 3. By that stage, a potential point x' on C_2 that corresponds to the point x in C_1 was retrieved as $x \leftrightarrow x'$ correspondence analysis.

The object detection and tracking algorithm described in Section 4.3.2 applies to C_1, in which they are posed according to Figure 3.2. C_2, C_3 and C_4 have the overlapping field of views with only C_1, so the stereo calibration and data association between C_1 and others are established.

There are several methods to establish the consistent labelling of the objects in

multiview tracking. The existing methods are categorised into three main approaches according to the available information such as calibration parameters, corresponding features or trajectory information (Khan & Shah, 2003).

The advantages and disadvantages of each type are generally described in Table 4.5. This table consists of general approaches that exist in associating the objects during multiview tracking.

The most straightforward approach is to use features to associate the objects in different views. These features vary from feature point descriptors such as SURF and SIFT to the colour and edge information. Colour matching is also widely used in literature; however, it is not reliable when the disparity in location and object is significant. Feature correspondence approach across multiple views, if they are solely used, can generate inaccurate matching results. In recent years, several methods feed the entire frames of different views into the CNN networks to find the object matching. In (Nassar et al., 2019), geometric metadata which contains camera's latitude and longitude and height is fed into SSD (Liu et al., 2016) and then Geo Regression Net. The system proposed in (Weber, Volkert, Hubschneider & Zöllner, 2019) uses a multitask CNN architecture and detection as well as an association within the network without geometric scene knowledge or information about camera calibration parameters. The authors tested their method on a non-busy scene generated by virtual reality.

In case of availability of camera calibration parameters and 3D environment model, labelling the same object can be accomplished by projecting the location of the 3D object to the same world coordinates. However, the association of the objects, especially those are close to each other or may produce occlusion, is a challenge.

Several methods fuse different approaches. In (Chang & Gong, 2001), geometry-based modalities and recognition-based modalities are grouped where the former modalities include epipolar geometry and homography while the latter is based on height and colour of the objects. The intersection of the principal access of a human body and

Table 4.5: Multiple cameras tracking approaches

Category	Advantages	Disadvantages
Feature Matching	• Efficient in simple scene • Multiple features can be used together	• Not reliable in huge disparity • Camera FOV dependant
Alignment	• Can work under non-overlapping FOV • Can handle occlusions	• Works under small disparity • Association after single-view tracking • Temporal movement dependent
3D Matching	• robust in controlled environment	• Camera calibration is needed • Needs expert intervention

the line obtained by homography from another view is selected as a feature to associate the objects in (Weiming Hu et al., 2006).

In this study, we have two major difficulties during tracking association:

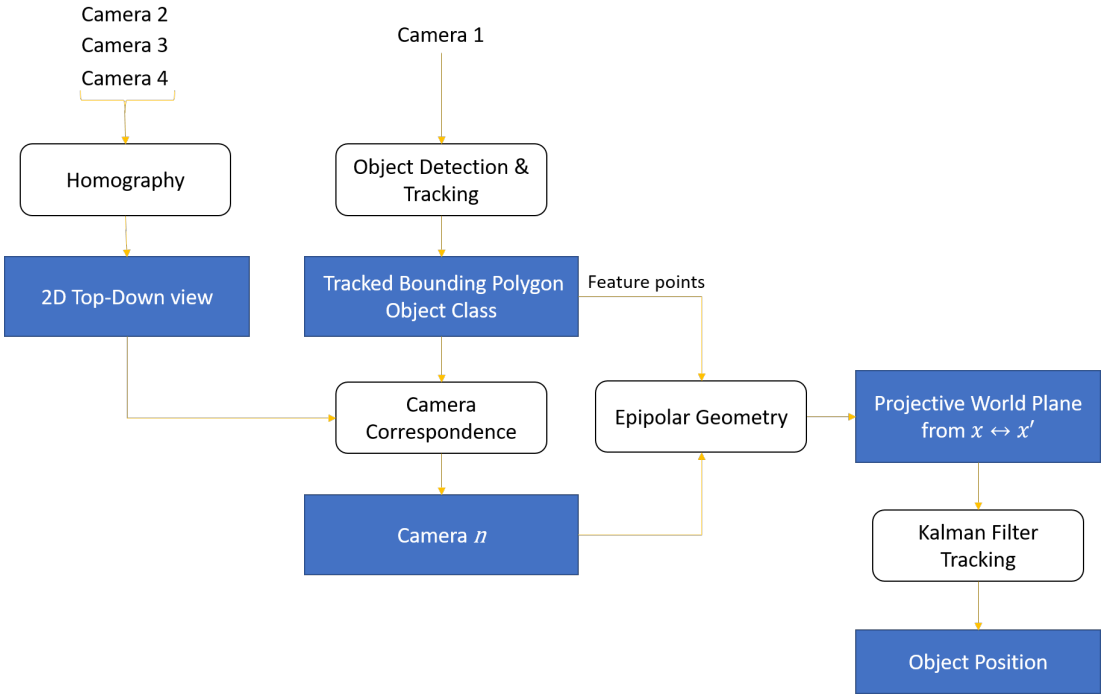


Figure 4.16: The overview of multiview object association

1. The scene is a real intersection with the complexity of the outdoor environment and large camera disparity

2. The objects in different cameras vary in size and shape, so feature matching needs to be carefully revised and tested.

Therefore, to establish the multiview object tracking in our scenario, multiple approaches are tested, and the most robust way is used in further analysis.

The final framework to generate robust tracking is illustrated in Figure 4.16. Firstly, single-view object detection and tracking are performed on C_1. From the 2D top-down view, retrieved during homography process, the central bottom point of the bounding box is located to find the camera correspondence. The feature points are then used to locate the objects in 3D world coordinates through triangulation in epipolar geometry. Finally, Kalman filter is used to re-locate the object in case it was missed during the tracking.

The aforementioned framework is finalised following the confirmation that the object appearance features cannot robustly represent the matching between different views in our camera design. Therefore, Section 4.4.1 explains the standard way that is mainly used in many research and in Section 4.4.2, the proposed multiview tracking is described.

4.4.1 Multiview Object Association by Bounding Box Matching

This approach is based on detection and tracking the objects in all cameras and then find the association by matching features.

Let ν_n^i be the undistorted central bottom point of the bounding box of object i belongs to C_n, where $n \in \mathbb{N}_{=\{1,2,3,4\}}$ in this scenario.

For each ν_n^i, generated during point correspondence analysis according to the 2D bird's eye view, there might be a matching point in another camera as described in

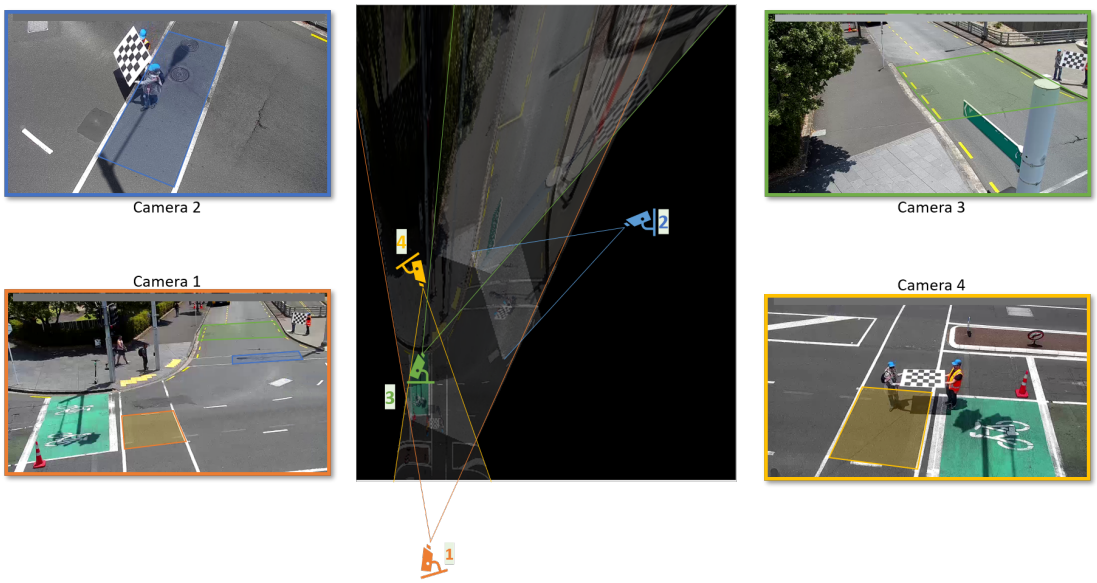


Figure 4.17: Relationships among cameras after homography

Chapter 3. Therefore, at this stage, we have a set of the bounding boxes and their corresponding undistorted points which mapped into the matching points on another camera through triangulation. Therefore, object association plays a role to identify the object correspondence from a different view.

The relationship between cameras, according to their known location, is illustrated in Figure 4.17. As can be observed, C_1, denoted by orange colour, is considered to be the primary sensor to relate the other cameras since this is the only one that has overlapping FOV with other three cameras. C_2 and C_3 have a minimal overlapping area that can be ignored because when there is a detected object in that area, the centre bottom point of the bounding box is not located at the other camera. Note that homography is only used to find the 2D association between cameras with respect to C_1.

Algorithm 2 describes the steps for multiview object tracking and association by object matching.

The reference camera in the proposed design is Camera 1, denoted by C_1. The object detection and tracking are applied to all cameras in synchronised sequences.

Once the objects are being detected in C_1, for any object restricted by a bounding box, the 2D camera association analysis is performed to establish $C_1 \leftrightarrow C_{p \in n \neq 1}$, where C_p represent the camera that holds the transformed point after homography. The details of this process will be explained in the next section.

After the corresponding camera is found, the epipolar line is calculated, and the point correspondence analysis defines the corresponding point of x_i on C_p, represented by ν_p^i for object i.

The process of object matching starts by finding the Euclidean distance between the newly generated point ν_p^i with the centre bottom point of the bounding box $\beta(T_p^j)$ and accepts those objects within a reasonable range. This process attempts to reduce the comparison of the bounding boxes that are far from each other.

According to the camera and its field of view with respect to C_1, the comparison between the bounding boxes is performed by matching algorithms.

The decision is made according to the object association. If the bounding box $\beta(T_p^j)$ of object j in Camera C_p match with the bounding box $\beta(T_1^i)$ of object i in C_1, then the object j is assumed to be the same object i. As the bounding box may have been changed in size during single-view tracking, experiments suggest to improve the accuracy by minimising the distance of the two points by using the mid-point of the ν_p^i and x_p^j). This point will be projected on occupancy map as the representation of object i.

There are two situations that the object correspondence cannot be established. C_1 has wide FOV, so they might be some objects that can only be detected on C_1. The other scenario may happen when the object is detected in other cameras while there is no point correspondence in C_1. In both cases, no point correspondence between two cameras can be generated. Therefore, the representing points are projected on occupancy map, considering the certainty value $\sigma_i = 0$. σ_i is used to denote the confidence of the multiview tracking. In following frames, the object i on occupancy map will be updated

and the $\sigma_i = 1$ if the object matches after correspondence.

Algorithm 2: Multiview object association and tracking

Result: 2D Occupancy Map M

$\beta(T_i^n) \rightarrow x_n^i$, Tracked bounding box i in Camera $n \in \mathbb{N}_{=\{1,2,3,4\}}$;
$M = \{\}$ /* M is occupancy map */
θ /* θ is the threshold of the acceptable distance */
if $\beta(T_1^i)$ *exists* **then**
  for $i \leftarrow 1$ **to** d /* d is the number of the objects in C^1 */
  do
    Search 2D Homography plane for camera correspondence $C_1 \leftrightarrow C_{p \in n \neq 1}$;
    Find epipolar line l^i on C_p & establish $x_1^i \leftrightarrow \nu_p^i$;
    while $\beta(T_p^j) : d(x_p^j, \nu_p^i) \leq \theta$ **do**
      Compare $\beta(T_p^j)$ & $\beta(T_1^i)$;
      if $\beta(T_p^j) \leftrightarrow \beta(T_1^i)$ **then**
        $j = i$;
        $M \leftarrow \mu^i = 1/2(\nu_p^i + x_p^j)$ & $\sigma_i = 1$;
      end
      else
        $M \leftarrow \mu^j = x_p^j$ & $\sigma_{j \neq i} = 0$;
      end
    end
  end
  else
    if $\beta(T_{p \neq 1}^j)$ *exists* **then**
      $M \leftarrow \mu_j = x_p^j$ & $\sigma_{j \neq i} = 0$;
    end
    else
      Move to next frame ;
    end
  end
end

Bounding Box Matching Results

The complexity of the camera design in this study causes the standard object matching techniques to face some difficulties. According to the camera design, among three cameras whose bounding boxes are compared to the reference camera, only the object

in one camera is similar to the reference camera in the appearance on a different scale. Therefore multiple approaches are used to find the object correspondence.

Experiments are done to test how the common feature matching methods work. Figure 4.18 are the original bounding boxes which are detected in the C_1 and C_4 with cross opposite view, resized for display purposes. For each set, Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) (Wang, Bovik, Sheikh, Simoncelli et al., 2004), 2D cross-correlation (Stoica, Moses et al., 2005) and Mean-squared error which are the widely used approaches for image comparison, are measured. To fit the requirement, two alternative metrics based on the colour comparison is also investigated. These are based on the fact that the colour similarity of objects in the top half of both bounding boxes should be similar. So the bounding boxes are divided into two for a better comparison.

Averaging Colour finds the mean value of each colour channel of the object parts first, and then the standard deviation of each channel is compared in two segments to estimate the variation. The average value is considered as the colour similarities.

To have a better understanding of colour comparison, the bounding box parts are segmented using 2D superpixel over-segmentation (Achanta et al., 2012). To improve the accuracy, the background colour, which is mainly grey, is ignored and then Averaging Colour method applies to each segment. The mean value of the standard deviation of each segment is measured and averaged again.

As the experiments from Table 4.6 revealed, nor standard image comparison metrics, neither colour similarity-based methods achieve a robust and reliable way for the cross multiview object association. Therefore, the importance of the location of the objects and local correlation is essential, although it generates challenging complexity.

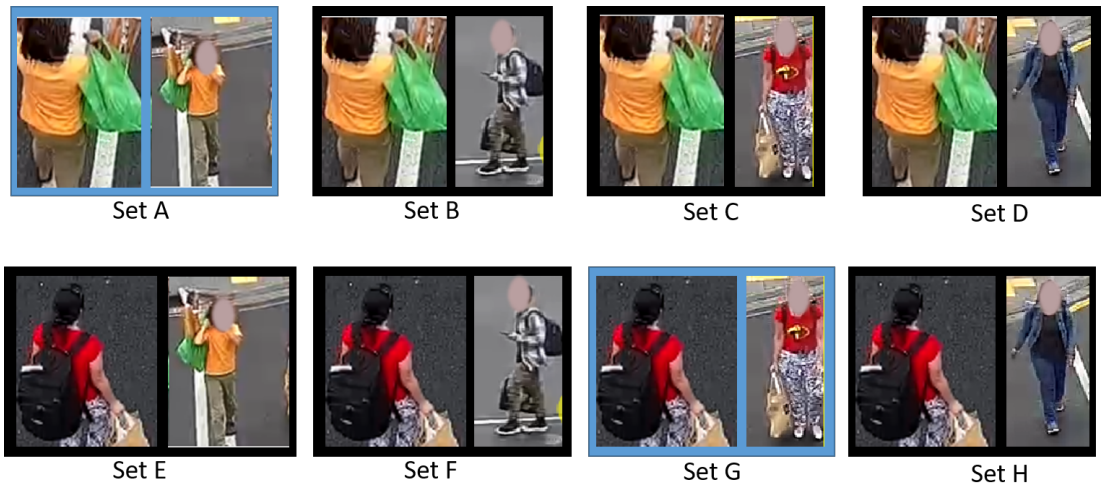


Figure 4.18: The combination sets used for experiments

Table 4.6: Object matching using different methods

	PSNR	SSIM	2D X-Corr	MSE	Ave. Colour	Seg. Colour
Set A	**10.21**	0.14	**0.01**	**6190**	**6.57**	45.64
Set B	9.65	0.1	-0.1	7040	12.57	**44.51**
Set C	9.13	0.09	-0.08	7936	10.13	51.91
Set D	8.97	**0.17**	-0.17	8243	17.52	44.62
Set E	9.58	**0.13**	-0.13	7147	37.17	44.51
Set F	9.54	**0.13**	-0.14	7222	37.15	**40.89**
Set G	8.94	0.06	-0.13	8287	38.72	49.36
Set H	**10.19**	0.11	**-0.06**	**6221**	**25.06**	48.53

4.4.2 Multiview Tracking By Projective World Coordinates

This approach is proposed to generate the occupancy map from the real world point coordinates, projected on the x axis. In this way, multiview tracking yields a robust tracking using a second tracking of 3D world points.

As described in the previous section, the object association across multiple cameras in our scenario deals with many challenges that cannot be addressed by object matching algorithm. Also, the camera design restricts us from performing object matching using the third view.

The single-view object detection and tracking method proposed in Section 4.3.2 is applied on C_1 as the primary camera that covers the majority of the scene view.

Figure 4.17 shows the relationship of the cameras when converting to a top-down view. To establish the camera correspondence, ν_1^i, which is the undistorted central bottom point of the bounding box of object i in C_1 is utilized. Using transformation matrix H_1, ν_1^i is transformed to generate $\bar{\nu}^i$, where $\bar{\nu}^i = H_1 * \nu_1^i$.

In particular, $\bar{\nu}^i$ if viewed by C_p, if the value of $\bar{\nu}_p^i = H_p^{-1} * \bar{\nu}^i$ exists. To establish the camera correspondence, C_p is selected if and only if there is a value assigned to $\bar{\nu}_p^i$ on top-down view. According to the camera design, it is possible that the object only locate in C_1, where Kalman filter predicts the location till it associate with another camera.

To simplify and formulate the problem from now, we reduce the camera dimension to two, so we assume the camera correspondence yields the association between C_1 and C_2.

Furthermore, to reduce the complexity of the epipolar line search, the prior estimation of the bounding box and homography is used. $\beta(T_1^j)$ is the detected bounding box in C_1 for object j. Hence, the size of the bounding box can be used as a hint to reduce the efforts of searching in the entire image of C_2.

Using homography, the bottom centre point $\hat{x}_1^b$ and top centre point $\hat{x}_1^h$ of each bounding box in the image of C_1 is transformed by transformation matrix H_1 into bird's eye view. Then the Euclidean distance d between the two points are estimated, and the intersection between the transformed epipolar lines of the object j and the circle with centre $\hat{x}_b^1$ and radius of d is used for searching.

Let us assume that $\hat{X}_1^j = \{\hat{x}_1^j, \hat{x}_2^j, \ldots \hat{x}_n^j\}$ are feature points detected in bounding box j in C_1, and we want to establish the correspondence between the objects in two camera views. According to the calibration parameters, the epipolar lines l_f^j is generated for every feature point f for the object j and the corresponding object's location in C_2 should be established.

Normalized 2-D cross-correlation (Yoo & Han, 2009) is applied using a template ρ, centered in $\hat{x}_n^j$ with the size of $2\alpha \times 2\alpha$ on the synchronised image of C_2.

To achieve this goal, first the Equation 4.11 computes the normalized cross-correlation between template ρ, generated from image of C_1 on image of C_2.

$$\gamma(u,v) = \frac{\sum_{x,y}[f^2(x,y) - \overline{f^2}_{u,v}][\rho(x-u,y-v) - \overline{\rho}]}{\left\{\sum_{x,y}[f^2(x,y) - \overline{f^2}_{u,v}]^2 \sum_{x,y}[\rho(x-u,y-v) - \overline{\rho}]^2\right\}^{0.5}} \tag{4.11}$$

, where f^2 is the synchronised image of C_2, $\overline{\rho}$ and $\overline{f}_{u,v}$ are the mean of template and $f^2(x,y)$ in the region under template t. γ represents the correlation coefficient in the range of $[-1,1]$.

To select the values along the epipolar line l_f^j, which corresponds to $\hat{x}_f^j$ on C_1, the intensity of cross-correlation value γ along epipolar line is used.

Using stereo parameters of C_1 and C_2, coordinates of points in the image of C_1 and the points of the epipolar lines are utilized to find the 3D locations of undistorted matching line. The $\hat{x}_f^j$ which is of size 1×2 is repeated to shape the similar size of the M pixels on the line ($M \times 2$) and then using triangulation, the 3D undistorted points on the matching line $l_f^{'j}$ is identified.

To reconstruct the 3D projection matrix P of a plane,

$$Q = K^{-1} * H_1, \tag{4.12}$$

where K is the intrinsic matrix and H_1 is the 3×3 transformation matrix of C_1 is used.

The 3×4 3D projection matrix P of a plane is

$$P = [Q_{1,*}, Q_{2,*}, Q_{1,*} \times Q_{2,*}, Q'_{3,*}], \tag{4.13}$$

where $Q_{1,*}$ and $Q_{2,*}$ are the first and second rows of matrix Q, normalized by their

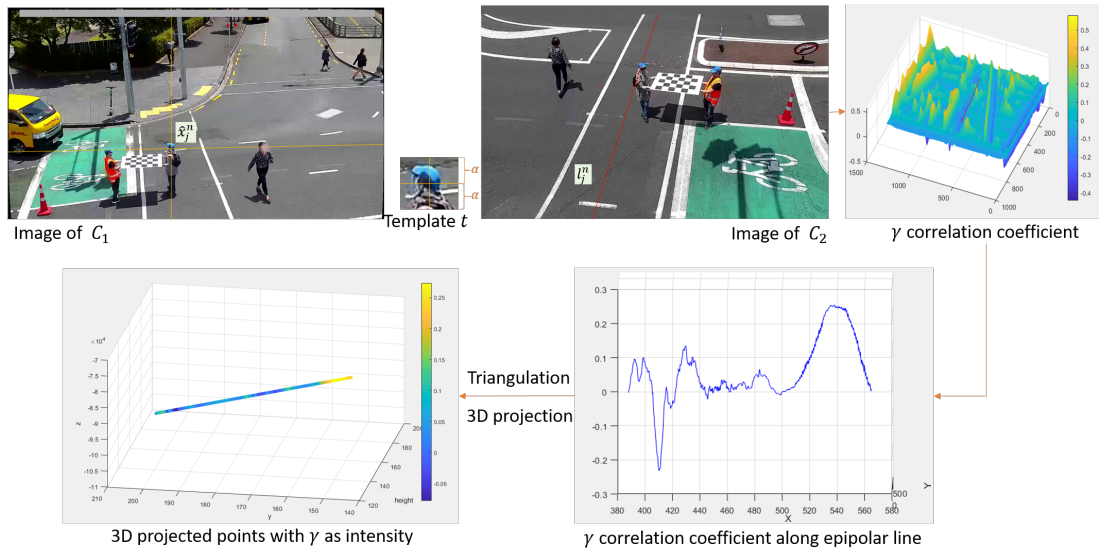


Figure 4.19: The preparation of the object correspondence for a single feature point

norm.

Therefore, the 3D projection matrix using the transformation matrix and camera intrinsic parameters are now compounded as the first three columns of P represent the rotation vectors while the 4-th column is the translation vector

$$\grave{l}_f^j = l_f'^j \star P'; \tag{4.14}$$

Figure 4.19 illustrates the steps prior to object correspondence for a single feature point. Firstly, the epipolar line corresponds to the selected feature point is generated and to reduce the search efforts, homography is used to restrict the range of the point location. Template ρ from C_1 is generated, and template matching is performed on the epipolar line on the image of C_2. The epipolar line is transformed into a 3D line using triangulation and later by a projection matrix P, the projected 3D world coordinates of the pixels on 3D line are generated.

The 3D world-coordinate lines $\grave{l}_f^j$ for all the feature points of the object j are later concatenated into one, preserving the correlation coefficient as the intensity, following

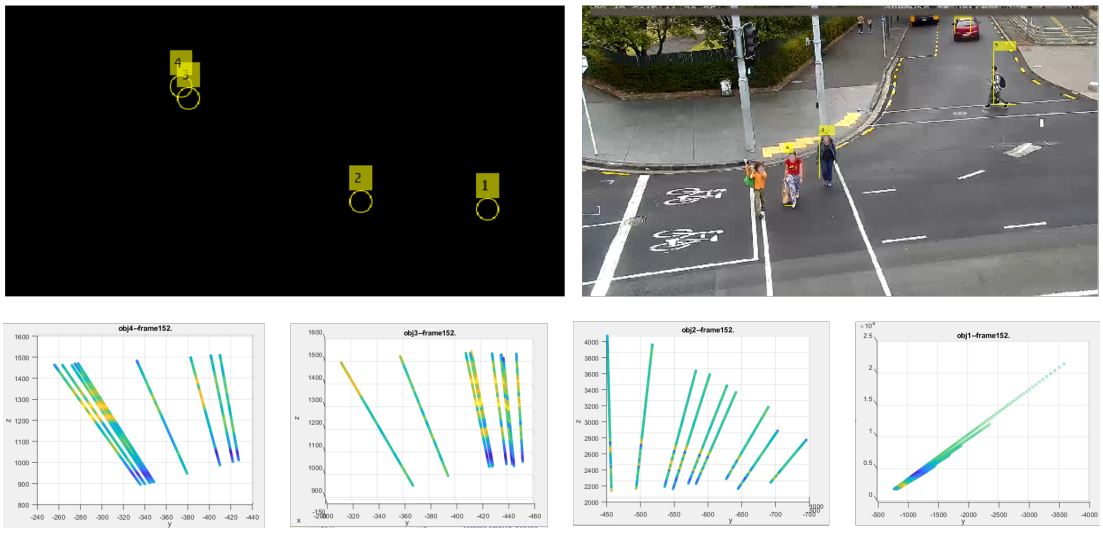


Figure 4.20: Generated 3D projected lines for objects and accumulated projected coordinates in a frame

the position values are transformed using matrix P.

Since the projection of the object on a plane is of concern, the accumulation is applied to the 2D position of the line, excluding the height of the object on x-axis. The accumulation of the array is done on the normalised value of (y, z) of value γ. The matrix A collects all elements of γ that have an identical position in (y, z) and stores their sum in the location of A corresponding to (y, z).

The final projected location of the objects in C_1 are reconstructed in occupancy map by applying value γ into the position by multiplying the value of (y, z) by γ and then normalise it.

Figure 4.20 shows the final locations of detected bounding boxes detected in C_1 on a rotated occupancy map with their corresponding label identifier which was the results of single-view tracking.

Double tracking using Kalmn Filter

As a result of the previous section, we have an occupancy map showing the object position with their corresponding label. Double tracking using a Kalman filter is used to

track the labelled objects and predict their location in case of being lost. Using projected 3D location retrieved from stereo calibration and triangulation also may propose some false negative especially when the object is detected only in the vicinity of C_1 where $C_1 \leftrightarrow C_{p \in n \neq 1}$ cannot be established.

Experimental Results - Multiview Object Tracking

An experimental comparative study is performed on three sets of videos captured in different time and weather condition. Recorded video data from four calibrated cameras, recorded at 25 frames per seconds (fps) are used in this study, however, for a better comparison, the frame rates are reduced to 10 fps.

For a better analysis of the strength and the weakness of the tracking algorithms, the sequences are categorised according to the four scene attributes which are a typical sunny day, rainy day, night time and peak time with many occlusions. The sequences that are used for this experiment and the number of frames with new frame rate for each are shown in Table 4.7. The number of objects is the total number of road users with different class in each sequence.

Table 4.7: The sequence used for multiview tracking experiment

	Attribute	Number of Frames	Number of Objects
Seq. 1	Sunny Day	100	6
Seq. 2	Rainy Day	100	4
Seq. 3	Night Time	150	4
Seq. 4	Peak Time	100	17

There are no other approaches with ground truth data that allow us to compare with this proposed method; therefore, a quantitative analysis is performed to evaluate the final double-tracking approach. Furthermore, the evaluation of the localisation accuracy of the objects on the occupancy map is not possible as the formulation of the ground

truth requires an extensive analysis which is out of the scope of this study.

In general, for single-view multiple object tracking problems, quantitative performance is mainly evaluated using two fundamental evaluation metrics, i.e. multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) (Milan, Leal-Taixé, Reid, Roth & Schindler, 2016; Bernardin & Stiefelhagen, 2008).

MOTA is the most widely used metric to evaluate the performance of the tracking approaches. It takes into account three sources of errors made by the tracker (i.e. false-positives, misses, number of identifier mismatches), averaged over all frames:

$$MOTA = 1 - \frac{\sum_j (FN_j + FP_j + IDSW_j)}{\sum_j GT_j}, \tag{4.15}$$

where j is the frame number, FN, FP and $IDSW$ are false negatives, false positives and ID switches, respectively. Although, in the original Equation 4.15, GT is the number of ground truth objects, in this study, we consider the value of GT to be the total number of the frame that an object is visible and has to be detected.

MOTP is the average of dissimilarity between true positives and their ground truth, however, in the context of projected 3D points on a occupancy map or road manifold, generating the ground truth is not feasible for the multiview approach.

MR and FPPI have already been used in Section 4.3.4 and here their definitions and their relations to Precision (PR) and Recall (RC) are identified. Lower MR and FPPI and higher value of PR and RC show the algorithm works better.

$$
\begin{aligned}
PR &= \frac{TP}{TP + FP} \\
RC &= \frac{TP}{TP + FN} \\
MR &= \frac{FN}{TP + FN} = 1 - RC \\
FPPI &= \frac{FP}{TP + FP} = 1 - PR
\end{aligned}
\tag{4.16}
$$

For evaluation of the proposed method, we define the value of GT to be the total number of the frames that an object is visible. TP in our context are the number of the correctly detected objects while FP is the extra false detection. FN is the number of frames that a specific object is lost.

Quantitative comparison results have been summarised in Table 4.8, where the proposed method is applied against four different sequences, each with the different number of objects.

The values of GT, TP, FP, FN and $IDSW$ are the accumulated results of all the objects in the sequence. For instance, in Seq. 4, the total number of the objects entering and exiting in 100 frames are 17, so the $TP = 1032$ represents the sum of the frames that all objects were detected correctly. However, the values in Equation 4.15 and 4.16 are averaged for all objects in a sequence. It is important to emphasise that customised version of MOTA that we used have a different range in $[0, 1]$.

As expected, the accuracy of the double multiview tracker in Seq.4 shows lower accuracy. This is because the video shows a peak time where many pedestrians are moving in different directions, and the system has to deal with the occlusion. The results also suggest that in Seq. 4, the number of switching identifiers are higher as a result of occlusion. In total, there are 63 frames that objects travel with the ID of their

Table 4.8: Quantitative comparison results for three sequences using double multiview tracking and Single-view Tracking algorithms

	3D Multiview Tracker				Single-view Tracker			
	Seq. 1	Seq.2	Seq.3	Seq.4	Seq. 1	Seq.2	Seq.3	Seq.4
Frame	100	100	150	100	100	100	150	100
GT	420	255	381	1198	420	255	381	1198
TP	383	234	351	1032	359	201	306	970
FP	3	15	21	31	3	15	21	31
FN	37	21	30	136	61	54	75	228
IDSW	4	0	4	63	7	0	7	67
PR	0.98	0.94	0.95	0.97	0.98	0.93	0.94	0.96
FPPI	0.01	0.05	0.04	0.02	0.01	0.06	0.05	0.03
RC	0.92	0.91	0.92	0.86	0.86	0.77	0.81	0.8
MR	0.07	0.08	0.07	0.13	0.13	0.22	0.18	0.19
MOTA	0.89	0.85	0.86	0.8	0.83	0.72	0.72	0.72

neighbouring objects. In other sequences, this value is at the minimum.

Although the comparison between the proposed single-view tracker and 3D extended multiview tracking, which uses the single-view as the base is not feasible, some experiments are done to highlight the improvements when double multiview tracking is used.

The significant enhancement after considering multiview parameters is achieved by reducing the number of missing objects by 48 objects per frame on average. Although the possibility of losing the objects may be increased in the area that there is no overlapping between C_1 and other cameras, however double-tracking using a Kalman filter predicts the movement, thus considering the 3D double-tracking method outperforms the improved single-view tracking itself.

To conclude this chapter, Figure 4.21 shows the comparative trajectories of 2D histogram of bird's eye view generated from uniform world coordinate and single-view tracking of Seq.1. In Seq. 1 , there are six objects in a sequence of 100 frames. As can be observed, single-view tracker fails to track the object when they merged together

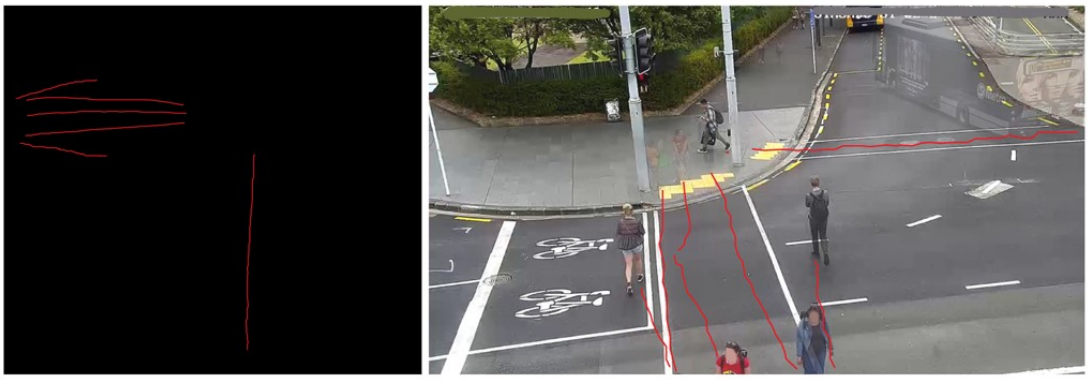


Figure 4.21: 2D histogram from uniform world coordinate and single-view tracker

and when the object occluded by road infrastructure. However, in the 2D histogram where 3D projected coordinates are used, the point correspondences and Kalman filter compensate the drawback of missing objects.

Note that the 2D trajectory preserve the rotation of the projective occupancy map to match Figure 4.20. Also, the background image of the single-view tracker is the transparent fusion of frame 1 and 100.

Chapter 5

Multiview Safety Analysis

5.1 Abstract

The general overview of the traffic safety monitoring system is illustrated in Figure 5.1. We discuss the sole safety parameter extraction in Section 5.2, and Section 5.3 explores the extracted features that can be used for further decision making.

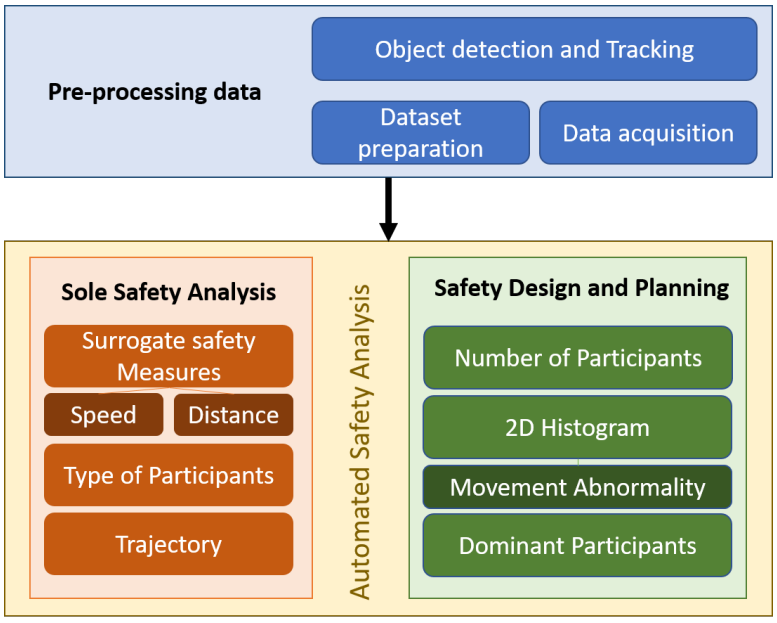


Figure 5.1: General framework of vision-based traffic monitoring system

5.2 Individual Safety Analysis

In Chapters 3 and 4, the process of detecting, classifying and tracking different traffic users and their corresponding decision points were explained. In this section, the risks involved in the safety of a participant are analyzed.

As a result of training the classifier, different types of participants are recorded to be vehicles, heavy vehicles, cyclists and pedestrians. Thus, the type of traffic users is later used in finding the interaction between different types.

5.2.1 Surrogate Safety Measures

Surrogate Safety Measures are widely used metrics to quantify the risks involved in the safety of road participants. As discussed in Chapter 2, these measures are mainly used by transport engineers where manual observation is involved; however, automatic vision-based safety extraction is always a valuable task and reduces many human efforts. For this study, the distance between moving objects, speed of each user, TTC and PET are investigated. Severity Index (SI) or risk factor is later proposed to measure the final risks corresponding to each object type.

Distance and Speed

To investigate the safety of traffic users, two main parameters play the most critical roles: distance and speed.

In order to extract these two factors, the object locations should be converted into a meaningful coordinate; thus, for this study, two ways are implemented to measure speed and distance. One approach is to use the homography information, while the second way is to utilise the 3D location of the objects retrieved from the previous steps.

Object localisation by homoraphy

The first method is based on homography, and it is applicable when the calibration parameter is not available. It can be prevalent practice because, on many roads, people have no opportunity or accessibility to go on-site and do the calibration.

In order to map the object location into a 2D top-down view, the real-world coordinates of a test frame should be known. This bird's eye homography matrix H_b is defined by finding four points in one of the frames and their corresponding points on the image captured in Google © or Apple © Maps, as shown in Fig. 5.2, knowing the actual distance between the points.

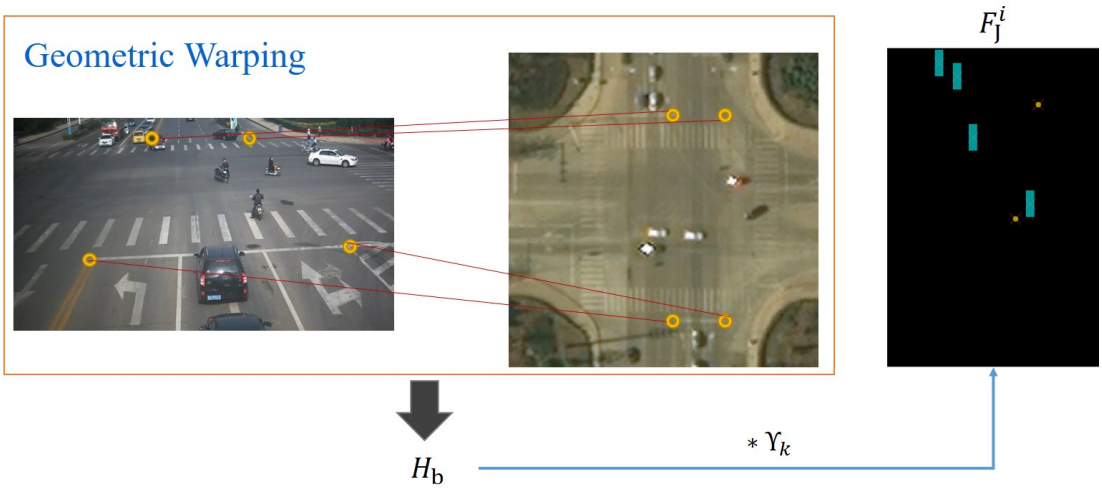


Figure 5.2: The homography matrix H_b is defined by affine transformation of four points and then applied to the bounding boxes' locations

In order to place more emphasis on the object's exact position for the different object type, the decision differs according to the camera field of view, road or surface conditions and object type.

Figures 5.3 and 5.4 show centre points of the bounding boxes for vehicles and pedestrians, respectively for different fields of view by a yellow point. As can be observed in Figure 5.3, when the camera view is closer to the top-down view, centre points can represent the objects and can be a good point for further tracking. Otherwise, the point which is more suitable to represent the objects should place more emphasis on

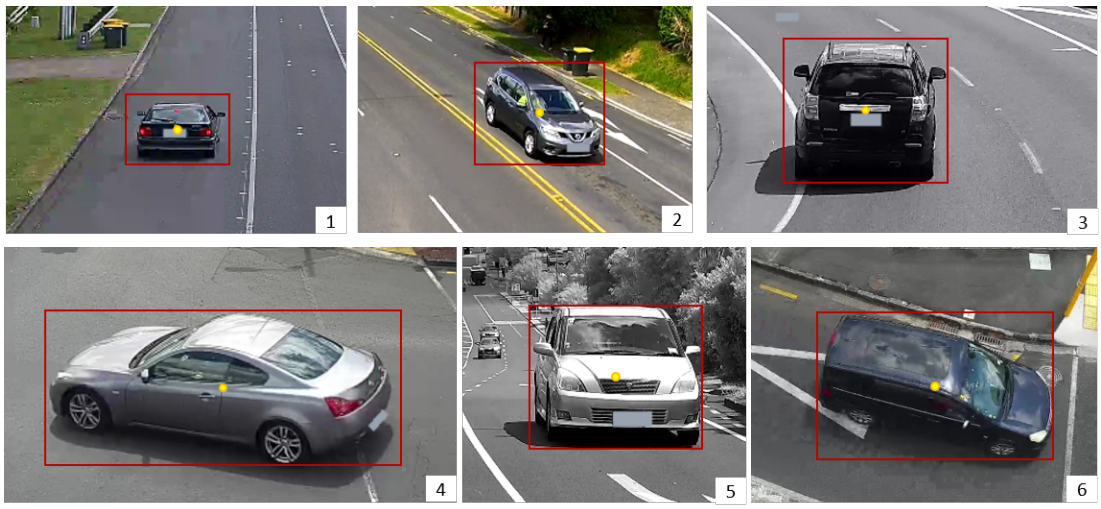


Figure 5.3: Centre points of the vehicles in different camera views

vertical or horizontal location depending on the camera FOV.

For pedestrians, the decision can be made easier comparing with cars because pedestrians are considered to be more vertical. Except for the case that the camera is mounted to have the top-down view (which centre points show the best depiction of the object as seen in sub-images 2 and 3), the suitable point representing the object is located in the central bottom part of the bounding box.

In general, for any types of objects, the decision on the best representation point is crucial, yet it is a challenging task depending on many parameters. Also, changing the shape of the objects, especially cars in a video sequence, may change this decision. Centre points are selected as the base in many kinds of research.

For homography, we assume $\beta(T_j)$ is the tracked bounding box of object j at frame I.

$$\beta(T_j) = [x_j^{\min}, x_j^{\max}, y_j^{\min}, y_j^{\max}] \tag{5.1}$$

where $x_j^{\min}$ and $x_j^{\max}$ are the minimum and maximum vertical offsets of the bounding box, respectively and $y_j^{\min}$ and $y_j^{\max}$ are the same points for the horizontal offset.

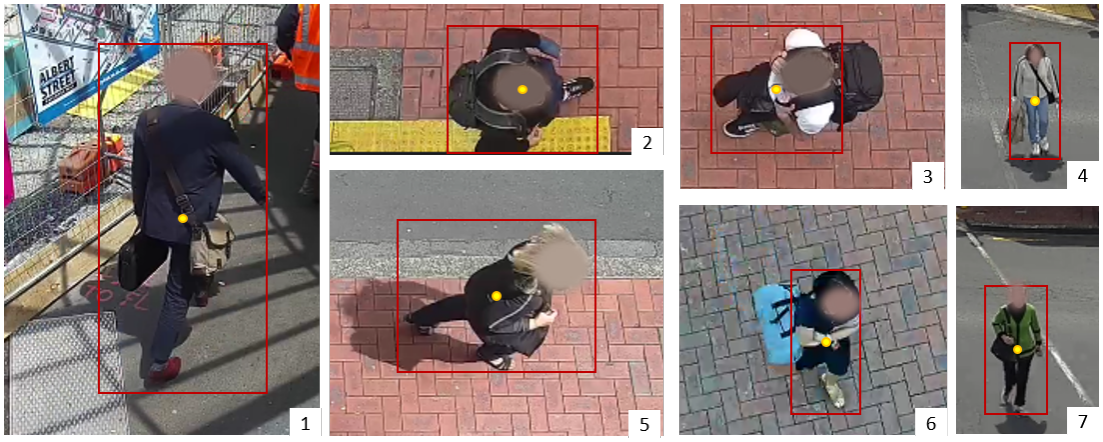


Figure 5.4: Centre points of the pedestrians in different camera views

We use the geometrically warped bounding box base point Υ_j, defined as

$$\Upsilon_j = \left[\frac{x_j^{\max} + x_j^{\min}}{\alpha} \quad \frac{y_j^{\max} + y_j^{\min}}{\beta} \right]. \tag{5.2}$$

where α and β are the decided fraction constant to locate the best point of the bounding box. In the proposed safety approach in (Moayed et al., 2017), $2/3$ of the bounding box's height was selected as the decision points according to the camera FOV, where $\alpha = 2, \beta = 3$. However, we found that the central bottom points can generalize the solution for our multiview safety monitoring system in this study$\alpha = 1, \beta = 2$.

The warped points are then projected into the 2D drop-down view as

$$\Upsilon_{j,I} = H_b \Upsilon_j. \tag{5.3}$$

Object localisation using camera calibration

Obtaining the real-world location of the objects from camera calibration is the second approach. The details of the camera calibration process in single and stereo cameras are

discussed in Chapter 3, thus here only the output of the process is used. The camera's intrinsic and extrinsic parameters are extracted during the calibration process, and the tracking and matching objects are explained in Chapter 4.

While the process of multiview object detection and tracking helps to preserve the robust tracking with fewer missing objects, the establishing of point correspondence $x \leftrightarrow x'$, mainly discussed in Chapter 3, is utilised to measure the speed. Following the triangulation, 3D locations of the undistorted central bottom point of the bounding box are retrieved, which shows the 3D location in the camera view. The Euclidean norm distance between the 3D points $x_1^{j,t}$ and $x_1^{j,t+k}$ is used to measure the speed of object j in k consecutive frames. Although the final occupancy map generated for a robust tracking can be used for finding the travelling distance of the object in the k frame, the triangulation approach of the central bottom point of the bounding box is vigorous enough when a single object is of concern. Figure 5.5 shows examples of two objects traveling in one second and how the intra-distance is measured using the 3D points of the same objects.

On the other hand, for the inter-object distance, where the distance between the objects is important, the relative Euclidean distance is measured using the 3D projective occupancy map.

Measuring the distance and speed

Regardless of the approaches that map the points to be understandable, speed refers to the average rate of an object travelling from one point to another over a period of time.

Let's assume Δd_j and Δt_j are the distance and the time duration which the object j

Figure 5.5: Intra-distance of the objects travelling in $K = 25$ frame

travels. By these definitions, the speed of object j is measured by:

$$S_j = \frac{\Delta d_j}{\Delta t_j} \tag{5.4}$$

Considering the frame rate and the decision on the frequency of the calculation, duration Δt_j can be considered. For example, if one wants to detect the speed of objects in every second in a video with 25 FPS, the position of the object should be identified in Frames i and $i + 25$. If the sensitivity of the system needs to measure the speed for every frame, the position of the objects has to be considered in every frame while $\Delta t_j = 1$. The lower Δt_j we consider, the more frequent results we can achieve. Note that if the first way of homography is used, the units conversion from pixels to metres should be applied.

In Figure 5.5, the speed of the vehicle and pedestrian are estimated as 15.94 KM/h and 10.29 KM/h, respectively. The low speed of the vehicle is because it started to accelerate after the traffic light had just changed to green. On the contrary, the pedestrian

ran to reach to the safe side of the road because he was passing when the light was red for pedestrians.

Time to Collision and Post-Encroachment Time

To have a better analysis of the traffic participants' behaviours, we use computer vision techniques to measure two infamous metrics for traffic engineers automatically. They are Time to Collision (TTC) and Post-Encroachment Time (PET) (Archer, 2004). TTC investigates the elapsed time that a user has to avoid a crash.

The overall process of extracting these safety parameters is illustrated in Figure 5.6. In the detection, classification and tracking process, the location of the object in camera coordinates is found. Considering 2D top-down view or 3D projected occupancy map, by any of the methods described previously, the corresponding locations of the objects are measured.

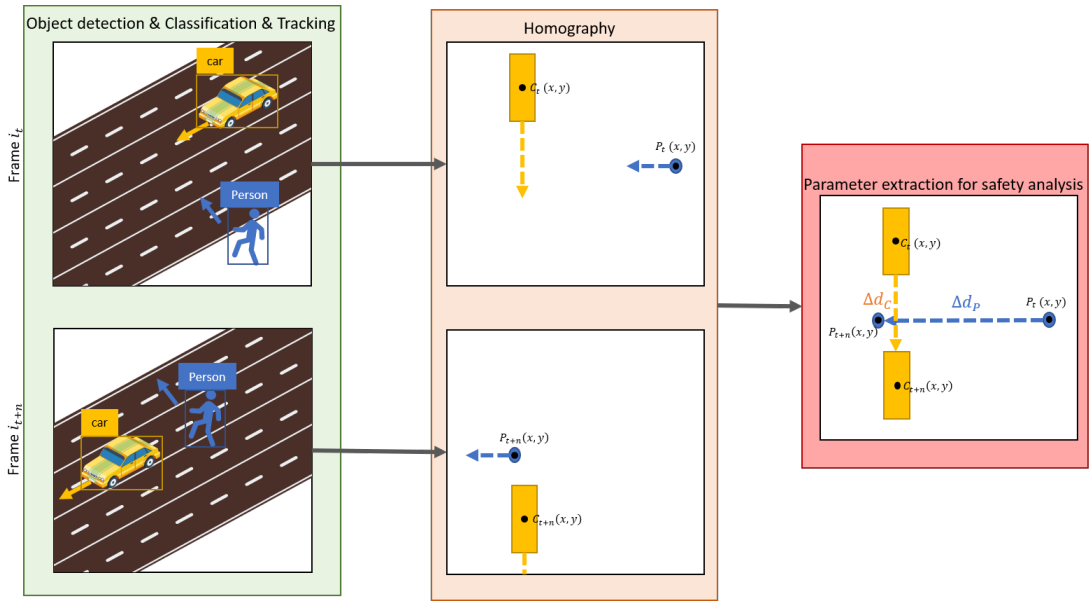


Figure 5.6: The process of extracting safety parameters for an object

To measure TTC followed by PET, the collision point or the point of intersection should be determined. It is worth mentioning that the collision here does not refer to

the actual accident or crash; however, it is a safety parameter considering the possibility of a crash if the users do not change their behaviour or only for measuring the time that a user has to avoid the crash area.

Figure 5.7 shows the general framework of how to find the collision point. Having the locations of two traffic users (particularly a pedestrian and a vehicle) in two frames, we find the point of intersection, regardless of the time, and considering the objects follow the straight path in the following frames.

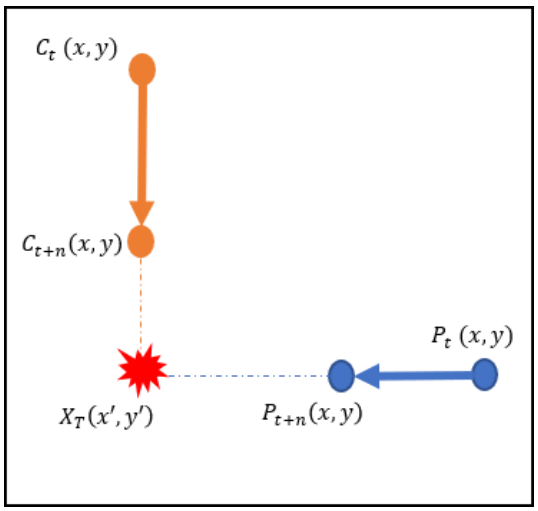


Figure 5.7: Detection of the collision points by knowing the location of objects in two frames

To have a better analysis, if Frame t t is selected close to the intersection time and k is a small, yet substantial enough amount to ignore the detection error, the outcome is closer to the reality.

Once the intersection point is detected, the Equation 5.4 is utilised to find out the elapsed time between the object's final position and the intersection point.

By revising the speed equation, the *Time to Collision* is calculated for object j at time t by:

$$TTC_j = \frac{\|X^{\hat{t}} - x^{j,t}\|_2}{S_j} \tag{5.5}$$

where S_j is the speed of object j, $X^{\hat{t}}$ and $x^{j,t}$ are the intersection point and position of

Table 5.1: Safety risk parameters extracted from 10 scenarios

Scenario	Pedestrian		Vehicles		PET
	Speed	Time to collision	Speed	Time to collision	
1	4.2	8.5	50.8	11.3	2.8
2	4.8	4.3	62.4	13.3	9.02
		9.1	45.4	14.9	5.7
3	4.2	6.5	49.5	9.8	3.3
4	4.8	4.9	78.7	18.1	13.2
5	7.5	3.2	62.1	16.7	13.4
6	1.9	2.9	30.8	10.003	7.05
	2.7	18.1	44.5	21.8	3.6
8	4.2	3.4	62.3	6.5	3.1
9	8.2	5.8	52.8	10.9	5.06
10	3.08	7.3	57.01	8.8	1.5

object j at time t. $||.||_2$ donates the Euclidean distance. Note that the calculation is done after transformation to the world coordinates or 2D plane.

Calculation of TTC and PET between a pedestrian and an approaching vehicle is tested in a collection of 50 sequences captured in an Auckland street using the proposed single-view tracking due to unavailability of camera parameters. Pedestrian safety is one of the desirable projects that transport practitioners are looking for to investigate and measure the risks for pedestrians, which later may lead to some investment decisions. The higher the risk for pedestrians crossing a site, the more focus is needed to propose a solution that reduces that risk. Therefore, this approach is used to provide this information using an automated calculation of TTC and PET.

The detailed calculations for 10 scenarios are depicted in Table 5.1. In each case, the speed of both users is estimated and converted into kilometres per hour for ease of understanding. Then the crash point followed by TTC is measured.

PET is used to measure the situations where two road users pass the point of intersection with the temporal elapsed seconds.

Mainly, PET in this context is the time interval in seconds that a pedestrian left the crash or intersection point before the approaching vehicles reach. In Scenarios 2

and 6, two values are calculated because the situation happened in both lanes, while in Scenario 6, a pedestrian stopped at the flush medians, therefore, the speed was different when confronting each course.

In the 50 scenarios that we analysed, the average speed of pedestrians and vehicles were 4.5km/h and 55km/h, respectively, while the average PET was 6.3s. Particularly, pedestrians have 6.3s time on average to avoid a collision in this specific area.

Transport engineers can define the predefined risk threshold of PET in each area and according to that threshold, they can decide to consider the area to be risky or not.

Notably, these safety values have a considerable variation in different scenarios depending on the urban and rural area, how busy the road is, vehicle type and the age and gender of the pedestrian and the driver. For example, scenario 6 is an elderly adult who tries to cross the road, and the first vehicle coming towards him is a bus with a slower speed compared to standard vehicles. However, PET for both directions shows that there is not a risk for him due to not being in peak times.

Individual Risk factor

To better determine the severity of the risks for each road user, a combination of the metrics are defined to fuse the effects of different parameters.

As discussed in Section 2.5.2, Severity Index, ranging from 0 to 10, is used to provide quantitative measurement for the severity of the risk for each traffic participant. The definition of this index varies for different contexts and different users, and it is customized to fit the best scenario.

However, in the context of general safety for different users at traffic intersections, here we consider these already-extracted parameters: speed (s_j), the contributing weighted distance to surrounding objects (d_j), type of user (ω_j), TTC (τ_j) and PET (p_j). ω_j shows the object class number. In case the object is detected as a pedestrian, this value is 4, for a cyclist, vehicle and heavy vehicle the value of ω_j are set to be

3, 2 and 1, respectively. TTC and PET only contribute to the risk if there is a vehicle approaching to the pedestrians and their trajectories collide in the region of interest, otherwise, $\tau_j, p_j = 0$.

The contributing weighted distance to surrounding objects d_j takes into account the type of the surrounding object ω_ρ, j and their distance to object j:

$$d_j = \frac{\rho}{\sum_{1}^{s} dist_{\rho-j} \times \left(\frac{1}{|5 - \omega_{\rho,j}|}\right)} \tag{5.6}$$

where ρ is the number of objects in the frame, $dist_{\rho-j}$ shows the Euclidean distance between object j and its surrounding object and $\omega_{\rho,j}$ is the type of surrounding object, where they impose more risk if they are heavy vehicles. If the distance between objects is far from other objects, the risk is lower.

The risk factor Φ_t for all users at time t is defined as:

$$\Phi_t = \Gamma_t \Theta_t \tag{5.7}$$

To break each matrix into more details:

$$\begin{pmatrix} \phi_{j_1} \\ \phi_{j_2} \\ \phi_{j_3} \\ \vdots \\ \phi_{j_n} \end{pmatrix}_t \leftarrow \begin{pmatrix} s_{j_1} & d_{j_1} & \omega_{j_1} & \tau j_1 & \cdots & p_{j_1} \\ s_{j_2} & d_{j_2} & \omega_{j_2} & \tau_{j_2} & \cdots & p_{j_2} \\ s_{j_3} & d_{j_3} & \omega_{j_3} & \tau_{j_3} & \cdots & p_{j_3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{j_n} & d_{j_n} & \omega_{j_n} & \tau_{j_n} & \cdots & p_{j_n} \end{pmatrix}_t \begin{pmatrix} \theta_s \\ \theta_d \\ \theta_\omega \\ \theta_\tau \\ \vdots \\ \theta_p \end{pmatrix}_t \tag{5.8}$$

Where $\theta_x = (0, 1]$ is the severity contribution of parameter x and can be optimised by the importance of each parameters. In this study, value of θ_s is set to be 0.5, while the

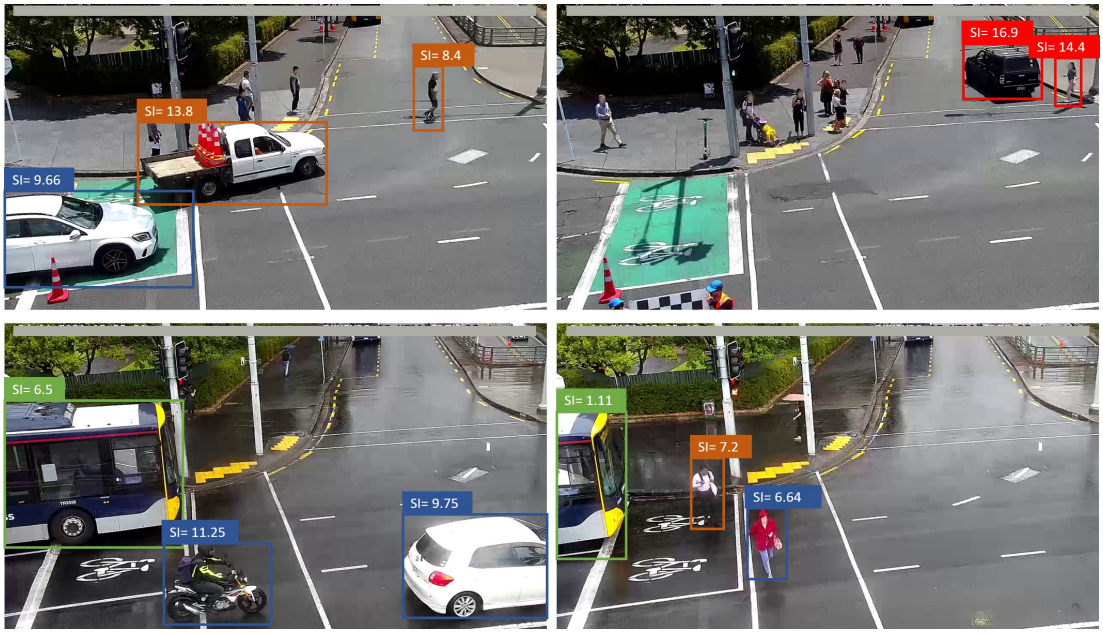


Figure 5.8: Examples of the calculated risk factor for different objects using Equation 5.8

rest are set to be 1. According to the system and the desired range of the final value, these parameters can be optimised.

Figure 5.8 demonstrates the measured Risk Factor for different objects in four frames. For simplicity, the value and the objects are highlighted by different colours according to the risk pyramid in Figure 5.9.

Table 5.2 shows the division of the imposed risks for different objects. Equation 5.8 is designed to consider the object relationship, in which the risk from heavy vehicles is more compared to a vehicle. However, as the speed is a positive contributing factor and pedestrians move slower than other types, Φ_t usually has the smaller value. Therefore, as the final results, we consider the risks for different objects in a range of the risk pyramid.

Notably, the calculated values and range in this study are based on TTC and PET in seconds, speed in kilometers per hour and distance in metres.

Remarkably, many other contributing factors are crucial to analyse the risk, such as weather conditions, if the pedestrians are walking in their own area and the possibility

Table 5.2: Assignment of Φ_t to the risk pyramid in this study

Object Type	No Risk	Moderate Risk	High Risk	Fatal/Injury
Pedestrian	<7	7-10	10-14	>14
Cyclist	<9	9-12	12-16	>16
Vehicle	<9	9-12	12-16	>16
Heavy Vehicle	<9	9-12	12-16	>16

of a crash in the region, which are not considered in this research.

5.3 Intersection Design and Planning

For this study, temporal information such as signal and red-light timing is not considered for safety analysis. Hence, it leads to more generalized traffic safety analysis, where traffic signals are designed to show more flexibility for future analysis.

Transport sectors are always interested in some real data from each intersection for their future design and planning. Information such as how many pedestrians and vehicles are crossing an intersection, or the leading and dominant participants leads to

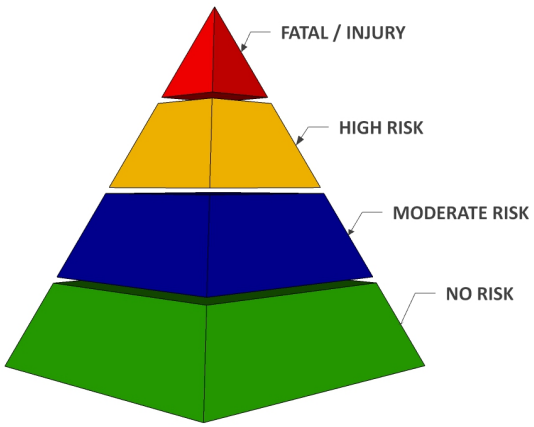


Figure 5.9: The safety risk pyramid, customised version of Figure 2.1

a better decision for design. As a result of this study, the number of different types of road users is generated through improved detection and tracking.

Also, the distribution of some selected movements or attributes can be utilized in the further design and safety analysis. The probabilistic framework can determine the frequency of particular movements, which take place in the intersection area via 2D histograms or heat-maps. These heat-maps can be designed to show the intersection area in terms of pixel values: how frequent each pixel is used by traffic users.

In other words, heat-maps or trajectories are generated to show the usage of road intersections by traffic participants based on a defined occupancy grid with a frequency analysis of passing participants for each cell in this grid. This can also be narrowed down to the type of participants.

For instance, the most frequent location when pedestrians start changing their states brings on a better management strategy to secure them from possible risks. Lastly, the location of the most recurrent accidents may show the need for any improvement that decreases potential accidents. Figure 5.10 shows an example of the movement of the objects in a 3-minute video using improved single-view object tracking and detection. The trajectories of pedestrians are depicted in red, while yellow, green and blue trajectories are for vehicles, heavy vehicles and cyclists, respectively.

Figure 5.11 also illustrates the approximate 1×1 metre grid map of the same video, converted by transformation matrix H_n, where n represents the camera number. To generate this 2D homography grid map, the grids are selected by the results of trajectories from the single-view tracking approach when performed on pedestrians, and the occupancy rate is normalised after all trajectories are confirmed. Three ranges of colours are chosen to depict the intensity value, where the darker red shows that the grid was used the most and the lighter colour shows the 1×1 metre grid use the least. This grid is generated after the tracking is done for 3-minute video using single-view tracking using homography approach.

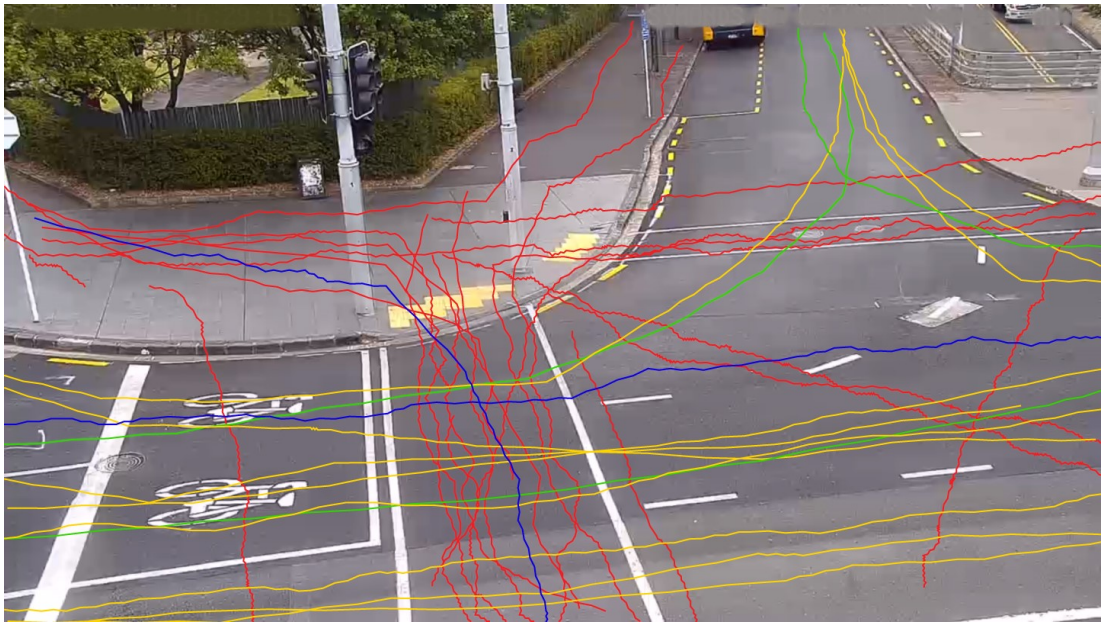


Figure 5.10: Trajectories generated from single-view object detection and tracking

This occupancy grid provides paramount and essential information to transport engineers on users' movement. As can be observed, the people are moving on the designed crossing area in the mentioned video.

Figure 5.11: 2D grid occupancy map to show the pedestrians' movement

In conclusion, this chapter introduces an automatic safety risk value generation for each traffic participant. Also, the multiview object detection and tracking can be used in many traffic safety analyses where the movement of the objects is the concern.

Chapter 6

Conclusion

6.1 Abstract

In recent years, traffic safety has been of primary concern for transport practitioners and with the extensive improvements in hardware devices to implement the complex algorithms, this goal can be achieved more easily.

In this thesis, a multiview vision-based intersection monitoring system was proposed in order to provide the individual and general understanding of safety for different traffic users.

The system composed of object detection and tracking using an enhanced version of CNN-based detector and point tracker in order to provide the robust tracking algorithm. The occlusion problem was addressed by multiple approaches in the analysis of single and multi-cameras. Then, the safety of traffic participants was analysed, considering the interactions between different users.

6.2 Accomplishments

The limited field of view of standard surveillance cameras restricts the area of an intersection to provide a general overview of incidents. Therefore, the main objective of this thesis was to convince the industry about the importance of using multiple cameras for the traffic safety of the individual user or further decisions on intersection design and planning. For this purpose, four cameras were calibrated individually to obtain the intrinsic parameters, and then a mutual one-to-one calibration process was done to ensure the lower reprojection error, achieved for extrinsic parameters.

The thesis focuses on object detection and tracking on existing traffic cameras to provide safety solutions to traffic organizations. For this, infamous CNN-based object classification and detection such as YOLOV2 was significantly improved to fit the purpose of being used in multiview tracking. The result of YOLO detection was fused with a motion-based feature extractor to achieve the best dominant segments to be tracked. To address the occlusion, two approaches were used to address the occlusion problem. The first one was based on bounding box matching and the second approach was to expand the region to a 3D projected road manifold or occupancy map. The former approach proved that with the complex camera design, the object matching may fail in many cases. However, the latter, combined with a Kalman filter, showed a robust tracking method in the reduction of false negatives.

Transport engineers mainly investigate the area of traffic safety while their data is usually collected by manual observations. This research aimed to provide data automatically by developing algorithms to use existing infrastructure-based traffic cameras. The safety information is provided individually for each traffic user, including vehicles, heavy vehicles, pedestrians and cyclists. Moreover, the process of classification and tracking provide some information for intersection design and planning, such as a path heatmap from the trajectories for different objects and dominant users of particular

intersections.

6.3 Future Works

As the existing literature revealed, there is a considerable gap between what computer vision can provide and what transport engineers are looking for. In other words, since traffic cameras are everywhere, computer vision can play a significant role in the automatic extraction of what transport engineers and practitioners want. Hence the proposed traffic safety extraction in this thesis can be significantly improved in future work.

6.3.1 Enhancing the Real-time Extraction

Despite the CNN-based detector used in this research being among the fastest algorithms for object detection and classification, merging with motion-based feature extraction to be used in a point-based tracker resulted in a higher processing time. Although, when the scene was not busy, and the tracking threshold was set to process the frame faster, there will always be an open research topic for improvement. The object tracking can be strengthened by using the features extracted from the lower levels of the network, while they provide the information needed for faster tracking.

The single-view object tracking is further improved in a multiview algorithm to achieve a robust tracker; however, again with sacrificing the processing time. The complexity and multi-layer steps of the proposed multiview tracker should later be improved to handle the trade-offs between robustness and time complexity for real-time analysis.

6.3.2 Improvement of Multiview Performance

Although, using the existing and real surveillance cameras is one of the advantageous parts and novelty of this thesis, being involved in camera design and installation can lead to much higher calibration accuracy and then multiview tracking. The four cameras used in this research were already installed by CCTV contractors for different projects, and they were selected for this research because they have some overlapping fields of view in the intersection.

Despite the camera design restricted us in developing many common-sense approaches, the outcome shows satisfactory results and highlights the fact that the solutions, mainly proposed in the restricted lab or university environments, may not work in industry. Therefore, if the camera design were intended to be used in a multiview monitoring system, the complexity and processing time could be improved to a great extent.

Intersection monitoring can be significantly enhanced by properly designing the camera locations, the field of view and overlapping FOVs.

6.3.3 Extending Multiview Dataset with Ground Truth Data

One of the significant challenges in this study was to compare the proposed multiview tracker on the available dataset and with other approaches.

The standard evaluation metrics for multiple objects tracking in MOT challenges could not merely be re-used in the multiview application due to their different plane. The idea is to generate the path from homography which is the most similar to the occupancy map or road manifold. Although the path from 2D homography would be different from the 3D projected map, the new metric can be proposed to investigate the similarity.

6.3.4 Extending Safety Parameters

This research aimed to mainly analyse the safety parameters for each traffic user while providing some useful information for transport practitioners. However, object detection and tracking can be extensively customised for different approaches. The outcome of this system can provide some safety parameters such as time to the intersection (TTI) and distance to the intersection (DTI), which are both examples of different essential factors for later decisions and behavioural analysis. Also, by providing the long-term analysis of the intersection, the behaviours of different users at different times of the day, weather conditions, before and after any changes will be valuable data for transport decision-makers.

There are many risk factors which can contribute to the severity index calculation. Parameters such as if the pedestrians are on the safe side of the road, turning vehicles and weather conditions could customise the final safety risk factor value for each participant. Also, the overall intersection safety parameters could be extensively used in individual safety parameters. By overall intersection safety parameters, the final individual values will be affected in such conditions as where there is a risky behaviour at the intersection such as red-light running, stationary vehicles, detected debris or a crash.

References

Abdelli, A. & Ho-Jin Choi. (2017, Feb). A four-frames differencing technique for moving objects detection in wide area surveillance. In *2017 ieee international conference on big data and smart computing (bigcomp)* (p. 210-214).

Abramson, Y. & Steux, B. (2004). Hardware-friendly pedestrian detection and impact prediction. In *Ieee intelligent vehicles symposium, 2004* (pp. 590–595).

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. & Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, *34*(11), 2274–2282.

Anjum, N. & Cavallaro, A. (2009, Sep.). Trajectory association and fusion across partially overlapping cameras. In *2009 sixth ieee international conference on advanced video and signal based surveillance* (p. 201-206).

Aoude, G. S., Desaraju, V. R., Stephens, L. H. & How, J. P. (2011). Behavior classification algorithms at intersections and validation using naturalistic data. In *2011 ieee intelligent vehicles symposium (iv)* (pp. 601–606).

Aoude, G. S., Luders, B. D., Lee, K. K., Levine, D. S. & How, J. P. (2010). Threat assessment design for driver assistance system at intersections. In *13th international ieee conference on intelligent transportation systems* (pp. 1855–1862).

Archer, J. (2004). Methods for the assessment and prediction of traffic safety at urban intersections and their application in micro-simulation modelling. *Royal institute of technology*.

Auckland Transport. (2019). *Intersections.* `https://at.govt.nz/driving-parking/road-safety/intersections/`. (Accessed: 2019-04-30)

Avidan, S. (2003). Subset selection for efficient svm tracking. In *2003 ieee computer society conference on computer vision and pattern recognition, 2003. proceedings.* (Vol. 1, pp. I–I).

Bay, H., Tuytelaars, T. & Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404–417).

Bengio, Y. (2009, January). Learning deep architectures for ai. *Found. Trends Mach. Learn.*, *2*(1), 1–127. Retrieved from `http://dx.doi.org/10.1561/2200000006` doi: 10.1561/2200000006

Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. In *Proceedings of icml workshop on unsupervised and transfer learning* (pp. 17–36).

Bernardin, K. & Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, *2008*, 1–10.

Bertalmio, M., Sapiro, G. & Randall, G. (2000). Morphing active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(7), 733–737.

Beymer, D. & Konolige, K. (1999). Real-time tracking of multiple people using continuous detection. In *Ieee frame rate workshop* (pp. 1–8).

Black, J., Ellis, T. & Rosin, P. (2002). Multi view image surveillance and tracking. In *Workshop on motion and video computing, 2002. proceedings.* (pp. 169–174).

Black, M. J. & Jepson, A. D. (1998). Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, *26*(1), 63–84.

Buch, N., Velastin, S. A. & Orwell, J. (2011, Sep.). A review of computer vision techniques for the analysis of urban traffic. *IEEE Transactions on Intelligent Transportation Systems*, *12*(3), 920-939.

Buddubariki, V., Tulluri, S. G. & Mukherjee, S. (2015). Multiple object tracking by improved klt tracker over surf features. In *2015 fifth national conference on computer vision, pattern recognition, image processing and graphics (ncvpripg)* (pp. 1–4).

Chan, C.-Y., Marco, D. & Misener, J. (2004). Threat assessment of traffic moving toward a controlled intersection. In *Ieee intelligent vehicles symposium, 2004* (pp. 931–936).

Chang, T.-H. & Gong, S. (2001). Tracking multiple people with a multi-camera system. In *Proceedings 2001 ieee workshop on multi-object tracking* (pp. 19–26).

Chien, H.-J. (2018). *Egomotion estimation and multi-run depth data integration for 3d reconstruction of street scenes* (Unpublished doctoral dissertation). Auckland University of Technology.

Chien, H.-J., Moayed, Z., Zhu, Y., Zhang, Y. & Klette, R. (2019). On improving bounding box regression towards accurate object detection and tracking. In *Ieee image and vision computing new zealand (ivcnz 2019)* (p. 1-6).

Chojnacki, W. & Brooks, M. J. (2003). Revisiting hartley's normalized eight-point algorithm. *IEEE transactions on pattern analysis and machine intelligence*, *25*(9), 1172–1177.

Del Moral, P. (1996). Non-linear filtering: interacting particle resolution. *Markov processes and related fields*, *2*(4), 555–581.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *Cvpr09.*

Dollár, P., Appel, R., Belongie, S. & Perona, P. (2014). Fast feature pyramids for object detection. *IEEE transactions on pattern analysis and machine intelligence*, *36*(8), 1532–1545.

Du, W. & Piater, J. (2007). Multi-camera people tracking by collaborative particle filters and principal axis-based integration. In Y. Yagi, S. B. Kang, I. S. Kweon & H. Zha (Eds.), *Computer vision – accv 2007* (pp. 365–374). Berlin, Heidelberg: Springer Berlin Heidelberg.

Edelsbrunner, H., Kirkpatrick, D. & Seidel, R. (1983, July). On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, *29*(4), 551-559. doi: 10.1109/TIT.1983.1056714

Fleuret, F., Lengagne, R. & Fua, P. (2005, Oct). Fixed point probability field for complex occlusion handling. In *Tenth ieee international conference on computer vision (iccv'05) volume 1* (Vol. 1, p. 694-700).

Focken, D. & Stiefelhagen, R. (2002, Oct). Towards vision-based 3-d people tracking in a smart room. In *Proceedings. fourth ieee international conference on multimodal interfaces* (p. 400-405). doi: 10.1109/ICMI.2002.1167028

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the ieee international conference on computer vision* (pp. 1440–1448).

Girshick, R. B., Donahue, J., Darrell, T. & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, *abs/1311.2524*. Retrieved from `http://arxiv.org/abs/1311.2524`

Goh, P.-K. & Wong, Y.-D. (2004). Driver perception response time during the signal change interval. *Applied health economics and health policy*, *3*(1), 9–15.

Goodfellow, I. J., Bengio, Y. & Courville, A. (2016). *Deep learning*. Cambridge, MA, USA: MIT Press. (`http://www.deeplearningbook.org`)

Hanai, T. (2013). Intelligent transport systems. *Society of Automotive Engineers of Japan*.

Hartley, R. & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. , 770–778.

Hinton, G. E. (2012). *Introduction to deep learning & deep belief net.* University of California.

Hinton, G. E. & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504–507.

Horn, B. K. & Schunck, B. G. (1981). Determining optical flow. *Artificial intelligence*, *17*(1-3), 185–203.

Houssineau, J., Clark, D. E., Ivekovic, S., Lee, C. S. & Franco, J. (2016, June). A unified approach for multi-object triangulation, tracking and camera calibration. *IEEE Transactions on Signal Processing*, *64*(11), 2934-2948.

Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4700–4708).

Hülnhagen, T., Dengler, I., Tamke, A., Dang, T. & Breuel, G. (2010). Maneuver recognition using probabilistic finite-state machines and fuzzy logic. In *2010 ieee intelligent vehicles symposium* (pp. 65–70).

Jabar, F., Farokhi, S. & Sheikh, U. (2015). Object tracking using sift and klt tracker for uav-based applications. In *2015 ieee international symposium on robotics and intelligent sensors (iris)* (pp. 65–68).

Jianbo Shi & Tomasi. (1994, June). Good features to track. In *1994 proceedings of ieee conference on computer vision and pattern recognition* (p. 593-600). doi:

10.1109/CVPR.1994.323794

Käfer, E., Hermes, C., Wöhler, C., Ritter, H. & Kummert, F. (2010). Recognition of situation classes at road intersections. In *2010 ieee international conference on robotics and automation* (pp. 3960–3965).

Keller, C. G. & Gavrila, D. M. (2013). Will the pedestrian cross? a study on pedestrian path prediction. *IEEE Transactions on Intelligent Transportation Systems*, *15*(2), 494–506.

Khan, S. & Shah, M. (2003, Oct). Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(10), 1355-1360.

Kim, C., Lee, J., Han, T. & Kim, Y.-M. (2018, 10 Jul). A hybrid framework combining background subtraction and deep neural networks for rapid person detection. *Journal of Big Data*, *5*(1), 22.

Kim, K. & Davis, L. S. (2006). Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In A. Leonardis, H. Bischof & A. Pinz (Eds.), *Computer vision – eccv 2006* (pp. 98–109). Berlin, Heidelberg: Springer Berlin Heidelberg.

King, M. J., Soole, D. & Ghafourian, A. (2009). Illegal pedestrian crossing at signalised intersections: incidence and relative risk. *Accident Analysis & Prevention*, *41*(3), 485–490.

Klette, R. (2014). *Concise computer vision*. Springer.

Klette, R., Koschan, A. & Schlüns, K. (2013). *Computer vision: Räumliche information aus digitalen bildern*. Springer-Verlag.

Koller, D., Weber, J. & Malik, J. (1994). Robust multiple car tracking with occlusion reasoning. In *European conference on computer vision* (pp. 189–196).

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kumar, P., Perrollaz, M., Lefevre, S. & Laugier, C. (2013). Learning-based approach for online lane change intention prediction. In *2013 ieee intelligent vehicles symposium (iv)* (pp. 797–802).

Kumar, P., Ranganath, S., Weimin, H. & Sengupta, K. (2005). Framework for real-time behavior interpretation from traffic video. *IEEE Transactions on Intelligent Transportation Systems*, *6*(1), 43–53.

Lacinák, M. & Ristvej, J. (2017). Smart city, safety and security. *Procedia Engineering*, *192*, 522 - 527.

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541–551.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E. & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems* (pp. 396–404).

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Leden, L. (2002). Pedestrian risk decrease with pedestrian flow. a case study based on data from signalized intersections in hamilton, ontario. *Accident Analysis & Prevention*, *34*(4), 457–464.

Lefèvre, S., Laugier, C. & Ibañez-Guzmán, J. (2011). Exploiting map information for driver intention estimation at road intersections. In *2011 ieee intelligent vehicles symposium (iv)* (pp. 583–588).

Lefèvre, S., Laugier, C. & Ibañez-Guzmán, J. (2012a). Evaluating risk at road intersections by detecting conflicting intentions. In *2012 ieee/rsj international conference on intelligent robots and systems* (pp. 4841–4846).

Lefèvre, S., Laugier, C. & Ibañez-Guzmán, J. (2012b). Risk assessment at road intersections: Comparing intention and expectation. In *2012 ieee intelligent vehicles symposium* (pp. 165–171).

Lin, M., Chen, Q. & Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21–37).

Liu, X., Xu, Y., Zhu, L. & Mu, Y. (2018, Oct). A stochastic attribute grammar for robust cross-view human tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(10), 2884-2895.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, *28*(2), 129–137.

Lopez, A., Canton-Ferrer, C. & Casas, J. R. (2007, April). Multi-person 3d tracking with particle filters on voxels. In *2007 ieee international conference on acoustics, speech and signal processing - icassp '07* (Vol. 1, p. I-913-I-916).

Lucas, B. D. & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision.

Mansouri, A.-R. (2002). Region tracking via level set pdes without motion computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 947–961.

Mercier, J.-P., Trottier, L., Giguere, P. & Chaib-draa, B. (2017). Deep object ranking for template matching. In *2017 ieee winter conference on applications of computer vision (wacv)* (pp. 734–742).

Milan, A., Leal-Taixé, L., Reid, I., Roth, S. & Schindler, K. (2016). Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*.

Moayed, Z., Chien, H., Zhang, D. & Klette, R. (2019, Oct). Surveillance-based collision-time analysis of road-crossing pedestrians. In *2019 ieee intelligent transportation systems conference (itsc)* (p. 2361-2366). doi: 10.1109/ITSC.2019.8917448

Moayed, Z., Griffin, A. & Klette, R. (2017, Dec). Traffic intersection monitoring using fusion of gmm-based deep learning classification and geometric warping. In *2017 international conference on image and vision computing new zealand (ivcnz)* (p. 1-5).

Montufar, J., Arango, J., Porter, M. & Nakagawa, S. (2007). Pedestrians' normal walking speed and speed when crossing a street. *Transportation Research Record*, *2002*(1), 90–97.

Muja, M. & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *In visapp international conference on computer vision theory and applications* (pp. 331–340).

Nassar, A. S., Lefevre, S. & Wegner, J. D. (2019). Simultaneous multi-view instance detection with learned geometric soft-constraints. In *Proceedings of the ieee international conference on computer vision* (pp. 6559–6568).

New Zealand Ministry of Transport. (2014). *Intelligent transport systems technology action plan 2014-18* (Tech. Rep.).

New Zealand Ministry of Transport. (2016). *Testing autonomous vehicles in new zealand* (Tech. Rep.).

New Zealand Transport Agency. (2019). *Disaggregated crash data* (Tech. Rep.).

Ng, A. (2015). *What data scientists should know about deep learning.*

Olivia. (2018, Oct). *Best wide-angle security cameras (systems) buying guide – reolink blog.* Reolink. Retrieved from `https://reolink.com/wide-angle-security-cameras/`

Pathak, S., Moro, A., Fujii, H., Yamashita, A. & Asama, H. (2018, Oct). Distortion-robust spherical camera motion estimation via dense optical flow. In *2018 25th ieee international conference on image processing (icip).*

Perez, C. E. (2017). *10 deep learning trends and predictions for 2017.* Retrieved from `https://medium.com/intuitionmachine/10-deep-learning-trends-and-predictions-for-2017-f28ca0666669`

Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural networks*, *12*(1), 145–151.

Rabaud, V. & Belongie, S. (2006). Counting crowded moving objects. In *2006 ieee computer society conference on computer vision and pattern recognition (cvpr'06)* (Vol. 1, pp. 705–711).

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016, June). You only look once: Unified, real-time object detection. In *2016 ieee conference on computer vision and pattern recognition (cvpr)* (p. 779-788).

Redmon, J. & Farhadi, A. (2017). Yolo9000: better, faster, stronger. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 7263–7271).

Redmon, J. & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R. & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).

Rosales, R. & Sclaroff, S. (1999). 3d trajectory recovery for tracking multiple objects and trajectory guided recognition of actions. In *Proceedings. 1999 ieee computer society conference on computer vision and pattern recognition (cat. no pr00149)* (Vol. 2, pp. 117–123).

Rublee, E., Rabaud, V., Konolige, K. & Bradski, G. (2011, Nov). Orb: An efficient alternative to sift or surf. In *2011 international conference on computer vision* (p. 2564-2571). doi: 10.1109/ICCV.2011.6126544

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1985). *Learning internal representations by error propagation* (Tech. Rep.). California Univ San Diego La Jolla Inst for Cognitive Science.

Salim, F. D., Loke, S. W., Rakotonirainy, A., Srinivasan, B. & Krishnaswamy, S. (2007, September). Collision pattern modeling and real-time collision detection at road intersections. In *2007 ieee intelligent transportation systems conference* (p. 161-166).

Scaramuzza, D., Martinelli, A. & Siegwart, R. (2006). A toolbox for easily calibrating omnidirectional cameras. In *2006 ieee/rsj international conference on intelligent robots and systems* (pp. 5695–5701).

Schreier, M., Willert, V. & Adamy, J. (2016). An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments. *IEEE Transactions on Intelligent Transportation Systems*, *17*(10), 2751–2766.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Shi, J. & Tomasi, C. (1993). *Good features to track* (Tech. Rep.). Cornell University.

Shirazi, M. S. & Morris, B. (2016). Vision-based pedestrian monitoring at intersections including behavior & crossing count. In *2016 ieee intelligent vehicles symposium (iv)* (pp. 1022–1027).

Shirazi, M. S. & Morris, B. T. (2017, Jan). Looking at intersections: A survey of intersection monitoring, behavior and safety analysis of recent studies. *IEEE Transactions on Intelligent Transportation Systems*, *18*(1), 4-24.

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Stauffer, C. & Grimson, W. E. L. (1999, June). Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 ieee computer society conference on computer vision and pattern recognition (cat. no pr00149)* (Vol. 2, p. 246-252 Vol. 2).

Stauffer, C. & Grimson, W. E. L. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(8), 747–757.

Stauffer, C. & Grimson, W. E. L. (2000, Aug). Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 747-757.

Stoica, P., Moses, R. L. et al. (2005). Spectral analysis of signals.

Streubel, T. & Hoffmann, K. H. (2014). Prediction of driver intended path at intersections. In *2014 ieee intelligent vehicles symposium proceedings* (pp. 134–139).

Svensson, Å. & Hydén, C. (2006). Estimating the severity of safety related behaviour. *Accident Analysis & Prevention*, *38*(2), 379–385.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A.

(2015). Going deeper with convolutions. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1–9).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2818–2826).

Taj, M. & Cavallaro, A. (2010). Multi-view multi-object detection and tracking. In R. Cipolla, S. Battiato & G. M. Farinella (Eds.), *Computer vision: Detection, recognition and reconstruction* (pp. 263–280). Berlin, Heidelberg: Springer Berlin Heidelberg.

Taylor, M. A. (2001). Intelligent transport systems. In *Handbook of transport systems and traffic control* (p. 461-475). Retrieved from `https://www.emeraldinsight.com/doi/abs/10.1108/9781615832460-031` doi: 10.1108/9781615832460-031

The U.S. Department of Transportation's Federal Automated Vehicles Policy. (2016, September). *Federal automated vehicles policy - september 2016* (Tech. Rep.).

Tiwari, G., Bangdiwala, S., Saraswat, A. & Gaurav, S. (2007). Survival analysis: Pedestrian risk exposure at signalized intersections. *Transportation research part F: traffic psychology and behaviour*, *10*(2), 77–89.

Tomasi, C. & Kanade, T. (1991). *Detection and tracking of point features* (Tech. Rep.). International Journal of Computer Vision.

Tran, Q. & Firl, J. (2014). Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression. In *2014 ieee intelligent vehicles symposium proceedings* (pp. 918–923).

Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T. & Smeulders, A. W. M. (2013, 01 Sep). Selective search for object recognition. *International Journal of Computer Vision*, *104*(2), 154–171.

Viti, F., Hoogendoorn, S. P., van Zuylen, H. J., Wilmink, I. R. & van Arem, B. (2008). Speed and acceleration distributions at a traffic signal analyzed from microscopic real and simulated data. In *2008 11th international ieee conference on intelligent transportation systems* (pp. 651–656).

Wakim, C. F., Capperon, S. & Oksman, J. (2004). A markovian model of pedestrian behavior. In *2004 ieee international conference on systems, man and cybernetics (ieee cat. no. 04ch37583)* (Vol. 4, pp. 4028–4033).

Wang, Y., He, L. & Velipasalar, S. (2010, Sep.). Real-time distributed tracking with non-overlapping cameras. In *2010 ieee international conference on image processing* (p. 697-700).

Wang, Z., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P. et al. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, *13*(4), 600–612.

Weber, M., Volkert, L., Hubschneider, C. & Zöllner, J. M. (2019, Oct). Cnn based multi-view object detection and association. In *2019 ieee intelligent transportation systems conference (itsc)* (p. 73-78). doi: 10.1109/ITSC.2019.8917092

Weiming Hu, Min Hu, Xue Zhou, Tieniu Tan, Jianguang Lou & Maybank, S. (2006, April). Principal axis-based correspondence between multiple cameras for people

tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(4), 663-671.

Weiming Hu, Tieniu Tan, Liang Wang & Maybank, S. (2004, Aug). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *34*(3), 334-352.

Yan, W. Q. (2016). *Introduction to intelligent surveillance*. Springer.

Yilmaz, A., Javed, O. & Shah, M. (2006). Object tracking: A survey. *Acm computing surveys (CSUR)*, *38*(4), 13.

Yilmaz, A., Li, X. & Shah, M. (2004). Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on pattern analysis and machine intelligence*, *26*(11), 1531–1536.

Yoo, J.-C. & Han, T. H. (2009). Fast normalized cross-correlation. *Circuits, systems and signal processing*, *28*(6), 819.

Yu, C., Zhang, C., Tian, G. & Liang, L. (2012). Vehicle trajectory description for traffic events detection. In *Advances on digital television and wireless multimedia communications* (pp. 228–235). Springer.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, *22*.

Zheng, J., Wang, Y., Nihan, N. L. & Hallenbeck, M. E. (2006). Extracting roadway background image: Mode-based approach. *Transportation Research Record*, *1944*(1), 82-88.

Index

Appendix A

Summery of datasets

In this part, the summery of the datasets, that are used in this research, are notified. However, the characteristics of each dataset are different and was used for different purposes. To have a better categorisation, Table A.1 summarise different datasets and the purpose of each.

Table A.1: Description of different datasets used in this research

Dataset Name	Characteristics	Description
Jinan Dataset	• 720×1080 pixels downsized to 480×720 • 25 fps • View of an intersection in China	• Objects were semi-automatically cropped for classification • Labeling and annotation for regression network
Auckland Urban and Rural Dataset	• Multiple 15 minutes footage from different streets in Auckland • 720×1080 or 480×720 pixels • All 25 fps	• Objects were semi-automatically cropped for classification • Labeling and annotation for regression network • Refer to Figure 4.2
Auckland Street Dataset	• 480×720 pixels • 25 fps • Objects were detected as cars	• Tracker was tested in terms of time efficiency and localisation accuracy
Auckland Indoor Dataset	• Three videos, 15 minutes each • 480×720 pixels • 25 fps • Objects were mainly persons	• Used for qualitative assessments of the proposed tracker
Auckland Rural Crossing Dataset	• 50 videos, each shows a person crossing a busy road • 480×720 pixels • 25 fps	• Used for extraction of safety parameters • Validation of TTC and PET
Multiview cameras Dataset	• Multiple footage of four overlapping cameras, recorded synchronously at different times of a day • 480×720 pixels • 25 fps	• Used for the main part of the thesis • Calibration were done to find the intrinsic and extrinsic parameters • Homography matrix was calculated