國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

針對安全攸關之嵌入式即時系統的異質性資訊融合

Heterogeneous Sensing Fusion for Safety Critical Embedded Real-time Systems

古君葳

Chun-Wei Ku

指導教授：施吉昇 博士

Advisor: Chi-Sheng Shih, Ph.D.

中華民國 107 年 1 月

January, 2018

doi:10.6342/NTU201800542

國立臺灣大學碩士學位論文
口試委員會審定書

針對安全攸關之嵌入式即時系統的異質性資訊融合

Heterogeneous Sensing Fusion for Safety Critical Embedded Real-time Systems

本論文係古君葳君（學號 R04922133）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 106 年 7 月 7 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

施吉昇
（指導教授）

逄愛君　　[illegible]

系 主 任　趙坤茂

致謝

很高興自己可以完成這一份碩士論文，很感謝身邊的人兩年來的教導以及陪伴讓我可以完成這篇論文。

首先我要感謝我的指導老師施吉昇教授，感謝教授兩年來教導我如何尋找問題，如何以嚴謹的方式去思考與論述，實是幫助我完成論文的最大幫手。

再來非常感謝 Elastic Team 的同學們以及叢培貴博士，感謝培貴總是不吝於提供許多建議，也感謝 Elastic Team 這段時間的陪伴，在撰寫論文卡關時，和同學們的討論總是可以讓我從中激發出許多想法。

還要感謝許多其他實驗室的同學們以及我的室友，雖然研究領域各不相同，但是在同樣的道路上前進，一起扶持前進總比一個人獨自闖關有動力多了。

最後感謝我的媽媽和姊姊，感謝她們的一路支持和養育，希望可以順利畢業好好回報她們的付出。

再次的感謝所有幫助過我的人，謝謝你們。

台北, 夏, 2017

古君葳

doi:10.6342/NTU201800542

摘要

現今有許多車禍的原因都是出自於駕駛的疏忽或是駕駛時分心。先進駕駛輔助系統即是為了輔助駕駛者的行車安全而發展的系統。其中，在先進駕駛輔助系統裡，緊急煞車輔助系統是一個重要的議題，至今有許多的研究專注於此。 電腦視覺、雷達、光達都是近年來常被應用在緊急煞車輔助系統上的技術。由於長距離偵測之雷達和光達的價格非常昂貴，因此我們希望僅使用影像感測器在緊急煞車輔助系統上。相較於只使用一個影像感測器的系統，我們提出了一個可以結合不同視野範圍的影像感測器之系統。此系統透過結合不同視野範圍的影像感測器，達到比使用一個影像感測器的系統更好的準確率。同時，我們也降低了物體偵測時誤認的機率。最後，由於嵌入式即時系統的運算能力是有限的，因此我們也必須加速感測器資訊整合方法。使用了我們提出的系統之後，相較於僅使用一個影像感測器的系統可以提升至多10%的準確率。

關鍵字 - 異質性資訊融合，異質性影像感測器系統，緊急煞車輔助系統，物體偵測

doi:10.6342/NTU201800542

Abstract

Nowadays, the bulk of these road collisions is caused by human unawareness or distraction. Since the most important thing is your safety and the safety of others, ADAS is developed to support enhanced vehicle system for safety and better driving. AEBS as an important part of the ADAS has become a hot research topic. Computer vision, together with Radar and Lidar, is at the forefront of technologies that enable the evolution of AEBS. Since the cost of long range radar and lidar is very high, we want to use camera-based system to construct AEBS. Instead of using a single monocular camera, we propose a heterogeneous camera-based system to use sensor fusion to combine the strengths of all the difference FoV cameras. Also, We use a heuristic false positive removal method to decrease the false positive rate that caused by the sensor fusion method. We optimize the sensor fusion method Because of the the limitation of computing resource on embedded system. As a result, the recall of YOLO can be increased up to 10% through our heterogeneous camera-based system.

Keywords - Heterogeneous Sensing Fusion, Heterogeneous Camera-Based Sytem, Tri-focal camera, AEBS, Object Detection

doi:10.6342/NTU201800542

Contents

doi:10.6342/NTU201800542

doi:10.6342/NTU201800542

List of Figures

doi:10.6342/NTU201800542

doi:10.6342/NTU201800542

List of Tables

doi:10.6342/NTU201800542

Chapter 1

Introduction

1.1 Motivation

Vehicle accidents are unfortunately very common for a long time. The bulk of these road collisions is caused by human unawareness or distraction. The National Highway Traffic Safety Administration (NHTSA) issued that 35,092 people died in vehicle crashes in 2015. Research shows that 94% of crashes were tied to human choices or error [1]. Since the most important thing is the safety of passenger and driver, advanced driver-assistance systems (ADAS) is developed to support enhanced vehicle system for safety and better driving. The purpose of ADAS is to enhance traffic safety and efficiency. For example, ADAS are composed of autonomous emergency braking system (AEBS), adaptive cruise control (ACC), lane departure warning (LDW), speed limit monitoring (SLM), rear cross traffic alert (RCTA), blind spot detection system (BST).

These subsytems provide different effects for better driving. Lane departure warning warns the driver if ADAS detects that the vehicle is departing from the current lane. Adaptive cruise control is used to handle the vehicle speed adaptively based on the distance between your vehicle and leading vehicle, your current speed, the road condition, and prediction of the leading vehicle's speed change. Automatic emergency braking system is a system that automatically detect a potential forward collision and trigger the vehicle braking system to decelerate the vehicle with the purpose of avoiding or mitigating

doi:10.6342/NTU201800542

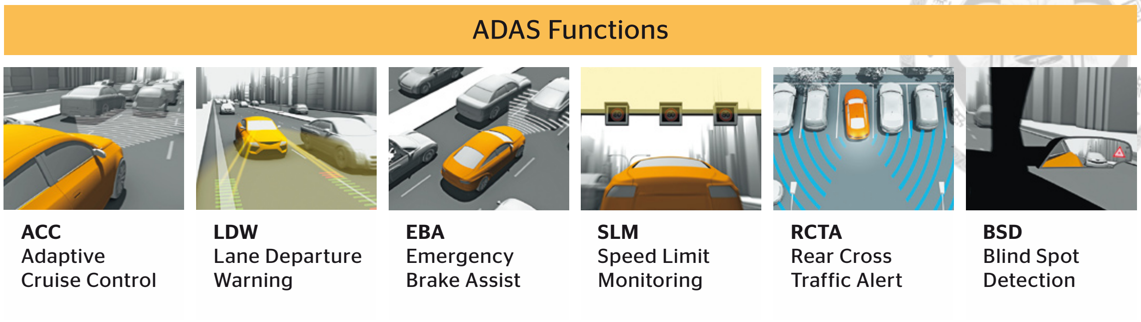


Figure 1.1: ADAS functions[1]

a collision if the driver fails to react to an emergency situation.

Nowadays, computer vision, together with radar and Lidar, is at the forefront of technologies that enable the evolution of AEBS. Figure 1.2 shows an example of ADAS sensors. Radar offers some advantages, such as long detection range (up to 200 m) and capability to operate under extreme weather conditions. However, it acts awful on false positives, especially around road curves, because it is not able to recognize objects. Lidar, which is commonly spelled LiDAR and also known as LADAR or laser altimetry, is an acronym for light detection and ranging. It is a surveying method that measures distance to a target by illuminating that target with a pulsed laser light, and measuring the reflected pulses with a sensor. Although both radar and lidar are precise and have long range, lidar has more resolution than radar sensors. So lidar is popularly used to make high-resolution maps in these days [2] [3] [4]. For example, Google's self-driving car [5] relies on lidar to provide it with a 360 degree of what is happening around the vehicle. It has a lidar sensor attached to the top of a car where it spins and shoots out lasers to create high-resolution maps of the car's surroundings. Camera-based systems also have their own limitations. They are very sensitivity to weather conditions, and they are not as reliable as radar when obtaining depth information. On the other hand, they have a wider field of view, and more importantly, they can recognize and categorize objects.

However, the cost of long range radar or lidar is very high. Lidar is the most expensive

[1]Source: http://www.conti-engineering.com/CMSPages/GetFile.aspx?guid=cf6d6925-8148-46e9-b59a-6eed8c23a0f6

[2]Source: Advanced Driver Assistant System - Intel, https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/advanced-driver-assistant-system-paper.pdf

doi:10.6342/NTU201800542

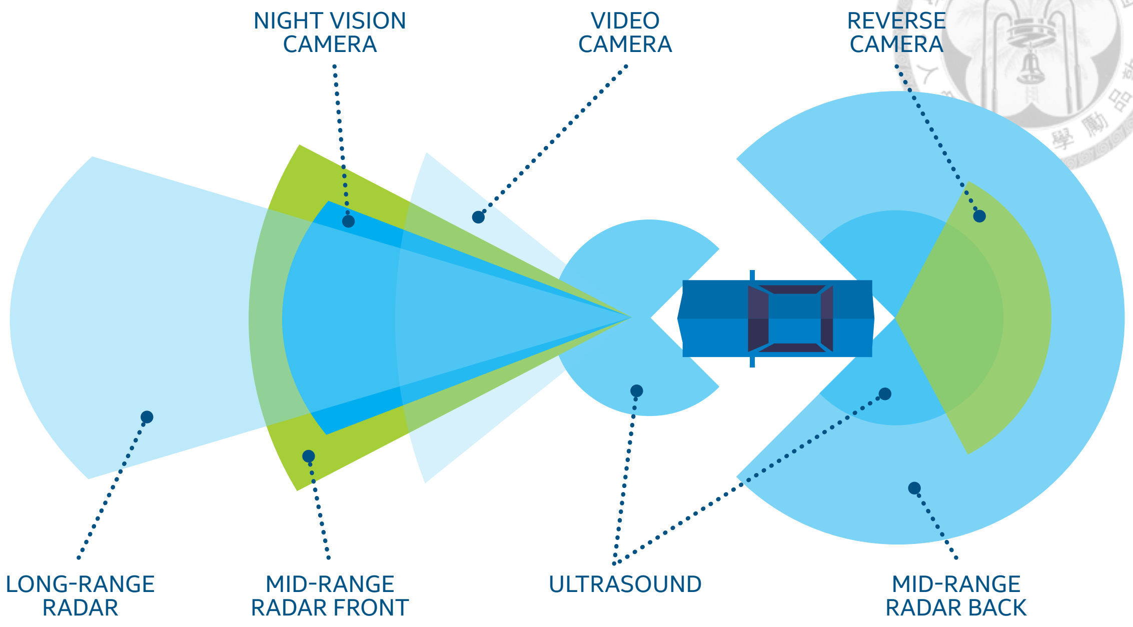


Figure 1.2: An example of ADAS sensors.[2]

component in these sensors. A single unit of lidar cost 75,000 US dollars just a few years ago. Although Google's self-driving car project-turned-spinoff company claim to slash the price of lidar by 90%, it stills cost 7,500 US dollars per vehicle [6]. Therefore, we want to study by using only camera-based system to construct AEBS system. There are many kinds of camera now, like short/long focal length, digital/optical zooming...etc. Choosing a suitable camera plays an important role in our camera-based system. The wide-angle camera (wide field of view) eventually reaching the super wide-angle range, which capture an even broader scope. On the other hand, the object will be very small when it is far from the camera. The telephoto camera restricts the angle of view, but it is capable to capture faraway objects at a larger size. If we use only wide-angle camera to detect the objects, we may miss some vehicles which are far from us. On the other hand, we may miss vehicles in a traffic jam (only the part of vehicles in the FoV) if we choose telephoto camera.

For all these reasons, our proposed heterogeneous camera-based system on AEBS will use sensor fusion to combine the strengths of all the difference field of view (FoV) cameras. There are some purposes to fuse information from the heterogeneous camera sensors

doi:10.6342/NTU201800542

Figure 1.3: An example of different FoV camera. (a) wide-angle camera. (b) focal length between wide-angle camera and telephoto camera. (c) telephoto camera.

of ADAS. First, to make content of some information more complete. Second, to improve the accuracy of the sensor-detecting information. Final, to improve the robustness of the sensor-detecting information. Although the true positive (shown in Figure 1.4(a)) of the AEBS system will be increased by using sensor fusion to fuse all the different FoV cameras, however, the false positive will be increased as well. As all the ADAS systems are computer-based and depend on sensor technology, data fusion, and image analysis. False positive instances (false alarms) or error of the system are unavoidable. Since the false positives may cause a fatal malfunction of a vehicle, and it may result in dangerous accidents. It is important to reduce the occurrences of false positives to improve the robustness and reliability of the system. For example, Figure 1.4(c) shows a false positive detection. In this case, the "car" rectangle is detected by vehicle detection algorithm. AEBS will automatically stop the car to avoid the accident because of this false positive detection. However, this unnecessary emergency brake activation doesn't make sense since there is no cars ahead. Further, this reaction may cause a rear-end accidents.

Because of the above mentioned, our work will develop a heterogeneous camera-based system on AEBS, which has not only the advantage of all the different FoV cameras, but also the lower false positive rate.

doi:10.6342/NTU201800542

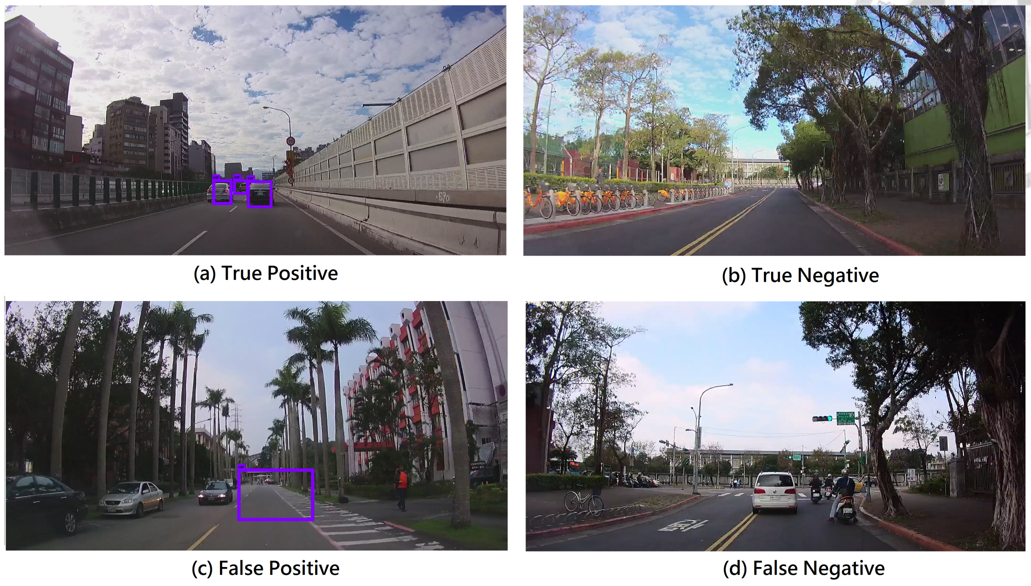


Figure 1.4: An example of true positive, true negative, false positive, false negative.

1.2 Contribution

In this thesis, we propose a heterogeneous camera-based system for AEBS. Compare to others camera-radar based system, the AEBS can be set up in a lower cost by using our system. Also, the proposed system guarantees higher accuracy and lower false positive rate than single monocular camera system. At last, we optimize the proposed sensor fusion method because of the the limitation of computing resource on embedded system.

1.3 Thesis Organization

The remainder of the thesis is organized as follows. In Chapter 2, we present previous existing fusion functions for heterogeneous sensors and related works. In Chapter 3, we present our system architecture. In Chapter 4, we present our algorithm and we use R-tree to reduce the search space. In Chapter 5, we evaluate our algorithm and present the results of our experiments. Chapter 6 conclude our work in this thesis.

doi:10.6342/NTU201800542

Chapter 2

Background and Related Work

2.1 Autonomous Emergency Braking System

In recent years, Europe, America and other developed countries have spent enormous efforts in developing ADAS, AEBS as an important part of the ADAS has become a popular research topic. To ensure safety, UNECE R131 [7] defines the functional requirements of AEBS. Here are the definition in UNECE R131. The subject vehicle refers to the tested vehicle which is the category M3 vehicle. The target refers to a high volume series production passenger car of category M1 AA saloon[1] or in the case of a soft target. Time to collision (TTC) refers to the interval of time obtained by dividing the distance between the subject vehicle and the target by the relative speed of the subject vehicle and the target, at an instant in time.

The subject vehicle shall approach the stationary target in a straight line. The functional part of the test shall start when the subject vehicle is traveling at a speed of 80 ± 2 km/h and is at a distance of at least 120m from the stationary target. First, at least one warning shall be issued no later than 1.4s before the start of emergency braking phase. Second, at least two warnings shall be provided no later than 0.8s before the start of emergency braking phase. Final, the emergency braking phase shall not start before the TTC is equal to or less than 3.0 seconds.

[1]As defined in the Consolidated Resolution on the Construction of Vehicles (R.E.3.), document ECE/-TRANS/WP.29/78/Rev.2, para. 2.

doi:10.6342/NTU201800542

Since AEBS should have at least one warning no later than 1.4s before the start of emergency braking phase and the emergency braking phase shall not start before the TTC is equal to or less than 3.0 seconds by the regulation, AEBS should issue at least one warning when TTC is equal to or less than 4.4s. In the same way, AEBS should issue at least two warnings when TTC is equal to or less than 3.8s. As a result of above, we need to detect if the stationary target vehicle exists when TTC is equal to or less than 4.4s and TTC is equal to or less than 3.8s.

By the formula of constant velocity motion ($s = v * t$), we can obtain that the distance between subject vehicle and stationary target is 97.78 meters when the speed of subject vehicle is 80km/h and TTC is equal to or less than 4.4s.

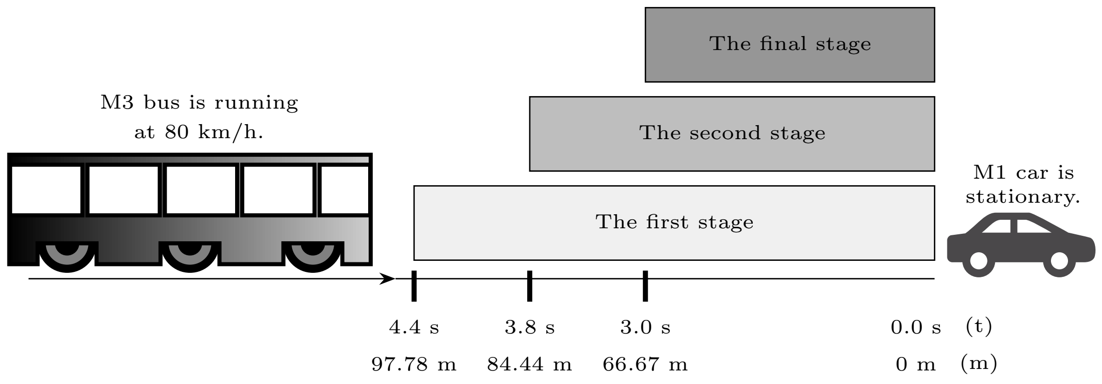


Figure 2.1: AEBS-TTC testing. (a) The first stage: at least one warning mode shall be provided. (b) The second stage: at least two warning mode shall be provided. (c) The final stage: the emergency braking phase shall start.

2.2 Vision-Based Vehicle Detection - YOLO

A large amount of researches which are computer vision based have been conducted on vehicle detection from over the years. Many of them applied traditional methods, such as background subtraction, frame difference, optical flow, etc. In [8], they applied a median-based background subtraction method to detect vehicles. In [9], they proposed a frame difference method to detect moving vehicles. However, our research targets a

doi:10.6342/NTU201800542

dynamic background, which is collected by a camera, meaning that traditional methods such as frame difference, background subtraction cannot be used directly.

Since deep learning has become a powerful technology in image recognition, gaming, information retrieval, and many other areas that need intelligent data processing. There are several machine learning based approaches have been proposed on vehicle detection in last few years. You Only Look Once (YOLO) [10] [11], a machine learning based algorithm which is proposed by Joseph Redmon at 2016, is a new and effective method to detect objects based on regression instead of classifying. Different from others machine learning algorithm (like R-CNN), the image is only fed into the YOLO network just for once and the network can output all the detect results. The workflow of YOLO detection system is shown in Figure 2.2. First, the system resizes the input image to 416 × 416 pixels. Second, it runs a single convolutional network on the image. At last, it filters the resulting detections by the model's confidence. Compared to traditional methods of object detection, processing images with YOLO is simple and straightforward. Also, YOLO is extremely fast and it is capable to detect a wide variety of object classes. It can detect over 9000 object categories and it runs at 67 frame per second (FPS) on a Geforce GTX Titan X [11].

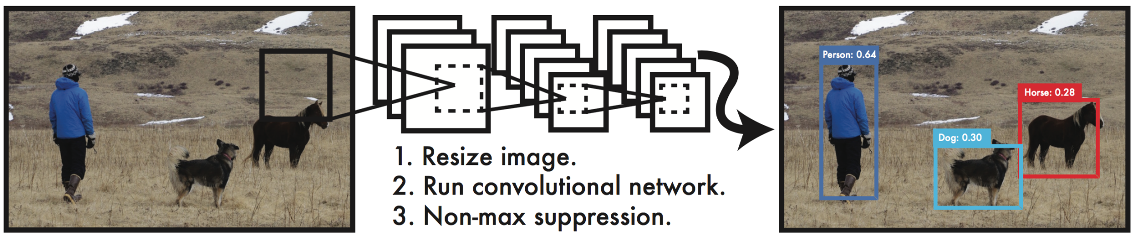


Figure 2.2: The YOLO detection system[2]

[2]Source: J. Redmon and A. Farhadi., "Yolo9000: Better, faster, stronger." in Computer Vision and Pattern Recognition, 2016.

doi:10.6342/NTU201800542

2.3 R-tree

R-tree is a hierarchical data structure proposed by Antonin Guttman [12] in 1984. It is based on B-tree and can be used to search spatial objects efficiently. Similar to the B-tree, the R-tree is also a balanced search tree, within which all leaf nodes are at the same height. R-tree is a well known spatial indexing technique that has been widely used in many geospatial applications, like indexing 2D or higher dimensional data. A common real-world usage for R-tree is to store spatial objects such as restaurant locations, streets, buildings informations, and then find answers to query such as "Find all restaurant within 1 km of my current location".

The main idea of R-tree is to group nearby objects and represent them by their minimum bounding rectangle (MBR) in the next higher level of the tree. The R-tree data structure consists of intermediate nodes and leaf nodes, and each node consists of several entries. Data objects are stored in leaf nodes and intermediate nodes are built by grouping rectangles at the lower level. Each entry of intermediate node is associated with MBR within which some rectangle completely encloses all rectangles that correspond to lower level nodes. Intermediate nodes contain entries of the form *(Rect, child-ptr)* where child-ptr is a pointer to a child node in the R-tree; Rect is the MBR that covers all rectangles of the child node. Leaf nodes contain entries of the form *(Rect, tuple-identifier)* where tuple-identifier is a pointer to the object description, and Rect is the MBR of the object. The main innovation in the R-tree is that parent nodes are allowed to overlap. This way, the R-tree can guarantee at least 50% space utilization and remain balanced [13]. In R-tree, the MBR of root node covers all rectangles and the leaf nodes store the information of data objects. Let us assume that M is the maximum number of entries that can fit in a leaf or intermediate node, m is the minimum number of entries that must fit in an intermediate node. The R-tree has the following properties:

(i) The root node has at least two children unless it is a leaf.

(ii) The entries number of entries on intermediate node should be no less than m and no greater than M.

(iii) All leave nodes have the same depth level.

doi:10.6342/NTU201800542

The first step of constructing R-tree is to generate MBR for the data rectangles. The next step is to group nearby rectangles and group them into a new MBR that is large enough to cover the rectangles in the next higher level of the tree. This process will continue until the root of R-tree is found, which is the rectangle covering all data rectangles. Figure 2.3 shows an example set of data rectangles. For example, at the first step, each data rectangle (gray rectangle) generates its MBR. At the next step, the data rectangles are divided into several groups, and the new MBRs (blue rectangles) are generated to cover each group of data rectangles. In this case, R_5 is the MBR which is created in this step and it covers the data rectangle R_{11} and R_{12}. The process will continue until the root of R-tree is found, which are red rectangles in this case. The final result of this construction is shown in Figure 2.4, which is the corresponding R-tree built on these data rectangles (assuming a maximum branching number $M = 3$ and minimum branching number $m = 2$).

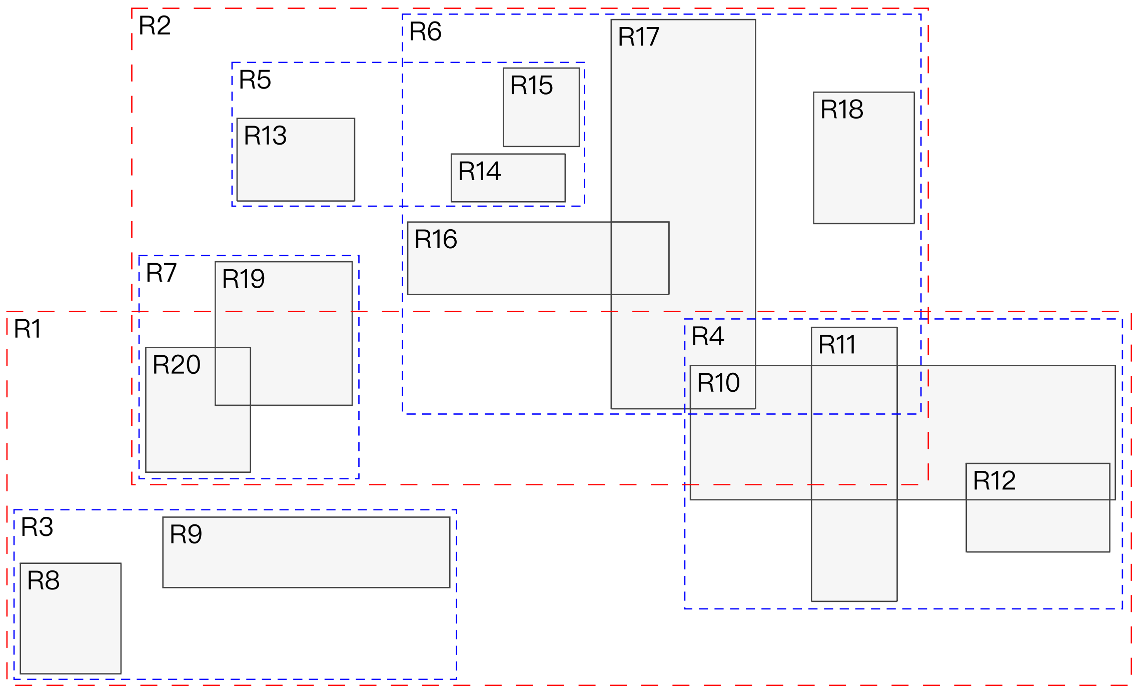


Figure 2.3: Data (gray rectangles) organized in a R-tree with $M = 3$, $m = 2$

Since the main idea of R-tree is to group nearby objects and represent them with their minimum bounding rectangle (MBR) in the next higher level of the tree. The key idea

doi:10.6342/NTU201800542

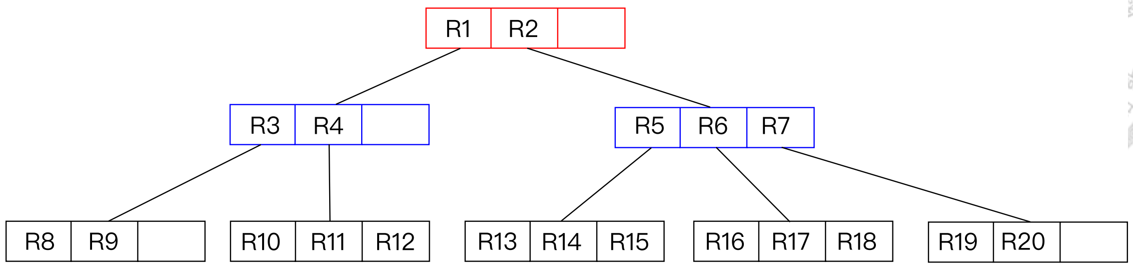


Figure 2.4: R-tree (with $M = 3$, $m = 2$) for the data rectangles of Figure 2.3

of searching algorithm in R-tree is to use the bounding boxes to decide whether or not to search inside a subtree. In this way, most of the nodes in the tree are never read during a search. Like B-tree, this makes R-tree suitable for large data sets and databases, where nodes can be paged to memory when needed, and the whole tree cannot be kept in main memory. Assuming M is the maximum number of entries that can fit in a leaf or intermediate node, N is the number of leaf nodes. The time complexity of searching algorithm is $O(\log_M N)$. Due to the searching algorithm, the performance of R-tree depends on the quality of the algorithm that clusters the data rectangles on a node. If the cluster algorithm results in several overlapping between MBRs, the performance will degrade because of the increasing of the searching subtrees.

In this thesis, we assume that the data rectangles are static (do not require dynamic insertions or updates). The low-x packed R-tree [14] is a step towards to construct an R-tree with 100% space utilization which will have as good response time as possible at the same time. However, this method will result in degradation of performance for region queries. Hilbert R-tree [15] is proposed by I. Kamel in order to cluster the region data in a better way than the low-x packed R-tree. Instead of sorting the data on the x or y coordinate, Hilbert R-trees use the Hilbert curve to impose a linear ordering on the data rectangles. The basic Hilbert curve on 1x1, 2x2, 4x4 grid are shown in Figure 2.5. The number in each grid presents Hilbert value. For example, on the 4x4 grid (denote by H_2, the (0,0) is at lower left corner), the point (0,0) on the H_2 curve has a Hilbert value of 0, while the point (2,1) has a Hilbert value of 13. Also, the Hilbert value of a rectangle needs to be defined. Following the experiments in [13], a good choice is that the Hilbert

doi:10.6342/NTU201800542

value of a rectangle is defined as the Hilbert value of its center.

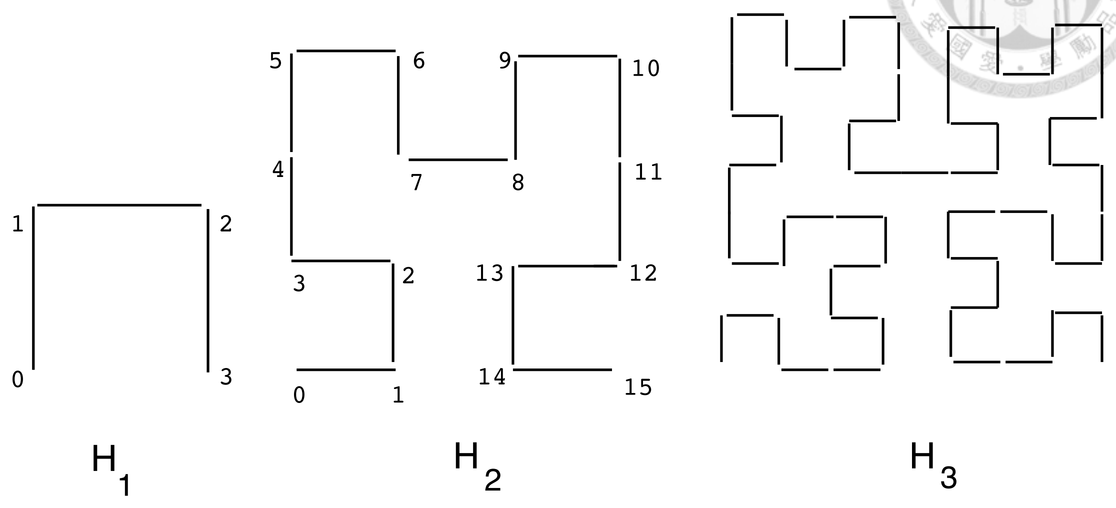


Figure 2.5: Hilbert curves of order 1, 2 and 3 [15]

The Hilbert R-tree claims that the overlapping between MBRs will decrease by using the ascending Hilbert value to pack the rectangles during the construction of R-tree. Since the performance of Hilbert R-tree is better and the construction cost is low (only change the packing rules from the original R-tree), we will use Hilbert R-tree in this thesis.

2.4 Related Work

Various approaches are taken to build ADAS platforms nowadays, with focus being reliability, high performance, low cost and low power consumption. These platforms usually contain a few processing units with different purposes on the same system on chip (SoC). There are several research focus on heterogeneous sensing fusion. That is, they use camera and radar or other sensors together [16] [17] [18] [19]. In [16], they propose a vehicle recognize algorithm base on radar and vision sensors with the application to automatic emergency braking. Since the radar is sensitive, there are a lot of false detection caused by radar. To improve this, they propose a vehicle recognition method which is based on shape and motion attribute. The motion attribute is designed to determine whether the object is either stationary or dynamic and the shape attribute aims to identity

doi:10.6342/NTU201800542

whether the objective is a vehicle or not by sensor fusion. In [17], they use mobile smart phone as a computing platform because the mobile smart phones today are equipped with numerous sensors that can help to aid in safety enhancements for drivers on the road. In [18], they use informations that are provided by in-vehicle Lidar and monocular vision to present a detect, track and classify entities in semi-structured outdoor scenarios. In [19], they use radar and camera to recognize whether the detected object is either vehicle or non-vehicle with the application of AEBS. Most of the researches use different type of sensors simultaneously on AEBS. Different from them, we use only camera-based system to construct AEBS system.

doi:10.6342/NTU201800542

Chapter 3

System Architecture and Problem Definition

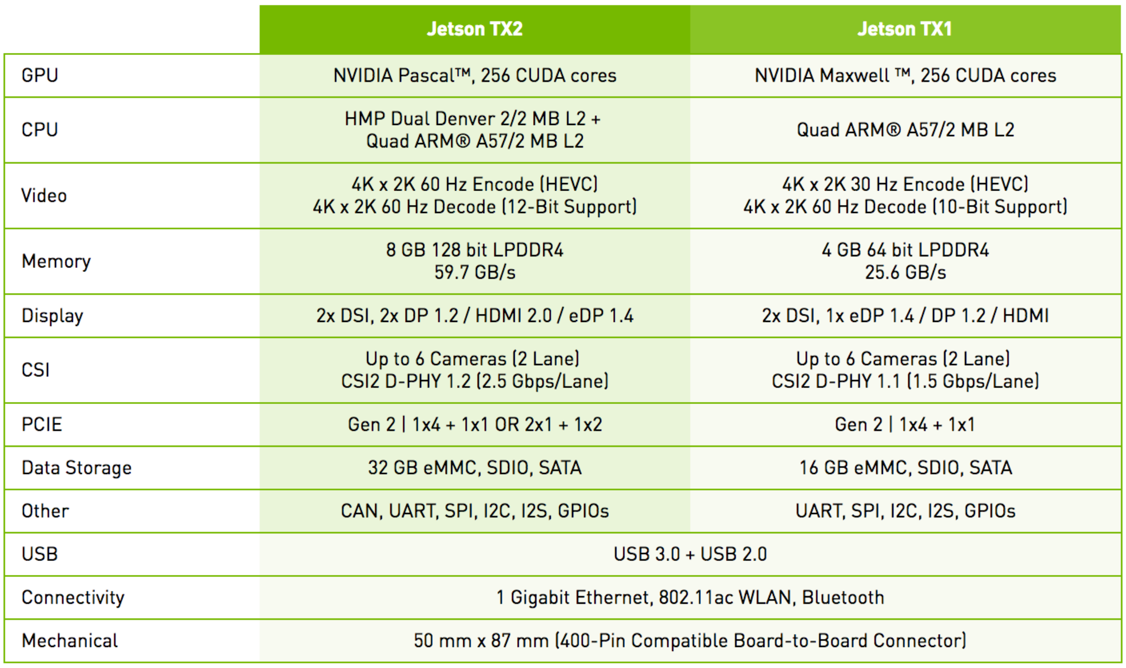
3.1 System Architecture

Safety-critical embedded systems are undergoing an evolution towards greater autonomy. In this thesis, we use the recently released NVIDIA Jetson TX2 as our computing platform. Since we use a GPU-based deep learning (YOLO) as our vehicle detection algorithm and several different FoV cameras simultaneously, the computing platform we used must have GPU to run YOLO system and suppors for multiple cameras module. Also, this computing platform should be portable because the ADAS is running on moving vehicles. Moreover, this computing platform should have low power consumption since the energy on vehicle is limited. Thus, NVIDIA Jetson TX2 is one of the most suitable computing platform for us due to these limitation.

NVIDIA Jetson is the world's leading AI computing platform for GPU-accelerated parallel processing in mobile embedded systems and is called for "autonomous everything" [20]. NVIDIA Jetson TX2 is part of the Jetson family of embedded computers. It shares a common GPU architecture with the higher-end NVIDIA Drive PX2, which is currently available only to automotive companies and suppliers. It is one of the most outstanding GPU-enabled platforms marketed today for autonomous systems. It has two

doi:10.6342/NTU201800542

important attributes for embedded use cases. First, it provides significant computing capacity. Second, it meets pratical limits on monetary cost as well as size, weight, and power consumption. It doubles the performance of its predecessor. And it can run at more than twice the power efficiency, while drawing less than 7.5 watts of power [21]. Figure 3.1 shows the capability of NVIDIA Jetson TX1 and TX2.



	Jetson TX2	Jetson TX1
GPU	NVIDIA Pascal™, 256 CUDA cores	NVIDIA Maxwell ™, 256 CUDA cores
CPU	HMP Dual Denver 2/2 MB L2 + Quad ARM® A57/2 MB L2	Quad ARM® A57/2 MB L2
Video	4K x 2K 60 Hz Encode (HEVC) 4K x 2K 60 Hz Decode (12-Bit Support)	4K x 2K 30 Hz Encode (HEVC) 4K x 2K 60 Hz Decode (10-Bit Support)
Memory	8 GB 128 bit LPDDR4 59.7 GB/s	4 GB 64 bit LPDDR4 25.6 GB/s
Display	2x DSI, 2x DP 1.2 / HDMI 2.0 / eDP 1.4	2x DSI, 1x eDP 1.4 / DP 1.2 / HDMI
CSI	Up to 6 Cameras (2 Lane) CSI2 D-PHY 1.2 (2.5 Gbps/Lane)	Up to 6 Cameras (2 Lane) CSI2 D-PHY 1.1 (1.5 Gbps/Lane)
PCIE	Gen 2 | 1x4 + 1x1 OR 2x1 + 1x2	Gen 2 | 1x4 + 1x1
Data Storage	32 GB eMMC, SDIO, SATA	16 GB eMMC, SDIO, SATA
Other	CAN, UART, SPI, I2C, I2S, GPIOs	UART, SPI, I2C, I2S, GPIOs
USB	USB 3.0 + USB 2.0	
Connectivity	1 Gigabit Ethernet, 802.11ac WLAN, Bluetooth	
Mechanical	50 mm x 87 mm (400-Pin Compatible Board-to-Board Connector)	

Figure 3.1: The ability of NVIDIA Jetson TX1 and TX2[1]

We set up three different FoV cameras at the same view direction on this platform. Figure 3.2 shows the proposed heterogeneous camera-based system on vehicle with a collection of sensors to enable sensor fusion and actions. For simplicity, we use normal camera to represent the camera which focal length between wide-angle camera and telephoto camera. In the rest of this thesis, the term "wide-angle camera", "normal camera" and the term "telephoto camera" will be used frequently. The horizontal angle of wide-angle camera is 150°, which covers all two lanes next to the vehicle. The horizontal angle of normal camera is 52°, which covers part of two lanes next to the vehicle and the lane that the moving vehicle is traveling at. The horizontal angle of telephoto camera is 28°, which only cover the lane that the moving vehicle is traveling at.

[1]Source: http://www.nvidia.com/object/embedded-systems-dev-kits-modules.html

doi:10.6342/NTU201800542

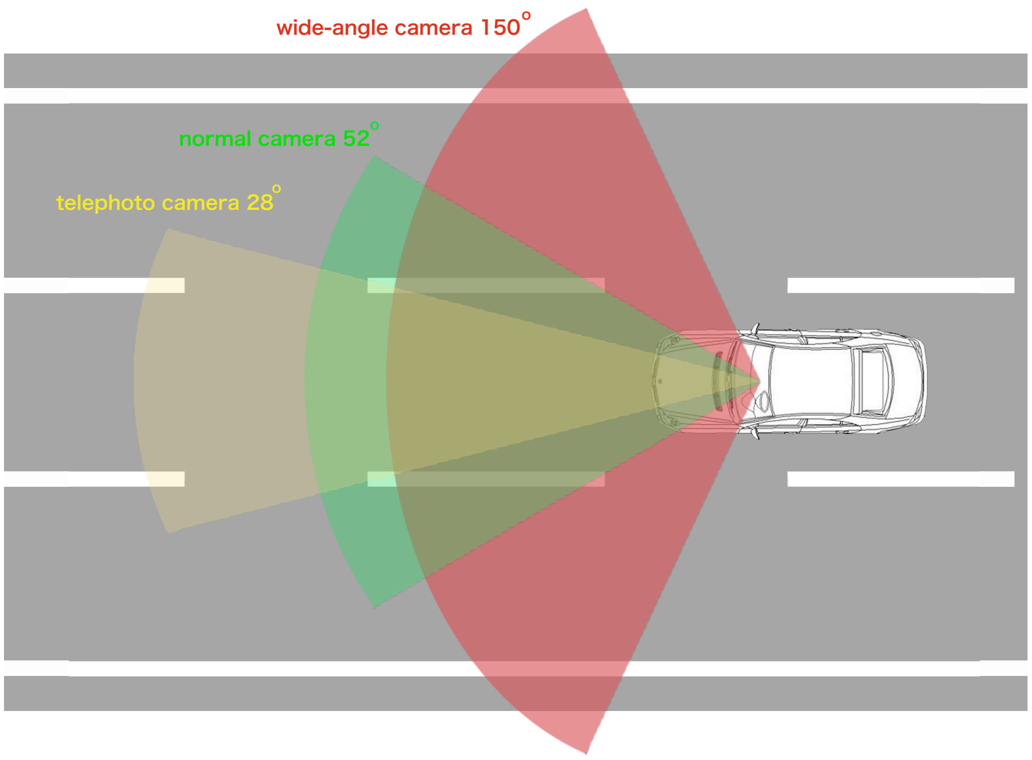


Figure 3.2: The architecture of the proposed heterogeneous camera-based system

In this thesis, we assume that the real-time clocks of all the cameras are synthesized, and all the camera frames are merged into a single frame in advance. Figure 4.2 shows the single frame composed by three different FoV camera. The frame at lower left corner comes from wide-angle camera, the frame at upper half comes from normal camera, and the frame at lower right corner comes from telephoto camera. The red rectangls are the region of interest (ROI) since AEBS only concern the target vehicles in front of the subject vehicle.

3.2 Problem Definition

The target problem is to increase the recall of vehicle detection algorithm (YOLO system) by using sensor fusion for heterogeneous camera-based system and optimize the sensor fusion method since the computing resource on embedded system is limited. According to the regulation of UNECE R131 [7], AEBS should start the emergency braking phase before a TTC equal to or less than 3.0 seconds. As shown in Figure 2.1, when the

doi:10.6342/NTU201800542

distance between the target vehicle and the subject vehicle equal to or less than 66.67m, the final stage should start the emergency braking phase. Thus, the proposed heterogeneous camera-based system should have the ability to detect the target vehicle when the distance between the target vehicle and the subject vehicle equal to or less than 66.67m. Another problem is to decrease the false positive rate since there are some false detections that caused by the sensor fusion method.

doi:10.6342/NTU201800542

Chapter 4

Design and Implementation

In this chapter, we design the heterogeneous camera-based system and implement sensor fusion method and false positive removal method on it. Figure 4.1 shows the data flow of the heterogeneous camera-based system. First, we discuss the impact of input image size on YOLO system in Section 4.1. Second, we need to transform all the detected vehicle rectangles from different FoV cameras into the same coordinate system before we design the sensor fusion method to fuse all the detected vehicle rectangles from different FoV cameras. We use linear transformation to transform the coordinate system in Section 4.2. After that, we use existed sensor fusion method into the system in Section 4.3. However, the effect of existed sensor fusion method in the night scenario is not significant enough. Thus, we proposed an advanced sensor fusion method in Section 4.4. Unfortunately, we may increase the false positive rate during the sensor fusion. The higher false positive rate may cause the system do more illogical operations during car driving. Thus, we proposed a false positive removal method to decrease the false positive rate in Section 4.5. Finally, since the computing resource on embedded system is limited, we need to reduce the search space of the proposed sensor fusion method. The discussion of reducing search space is in Section 4.6.

doi:10.6342/NTU201800542

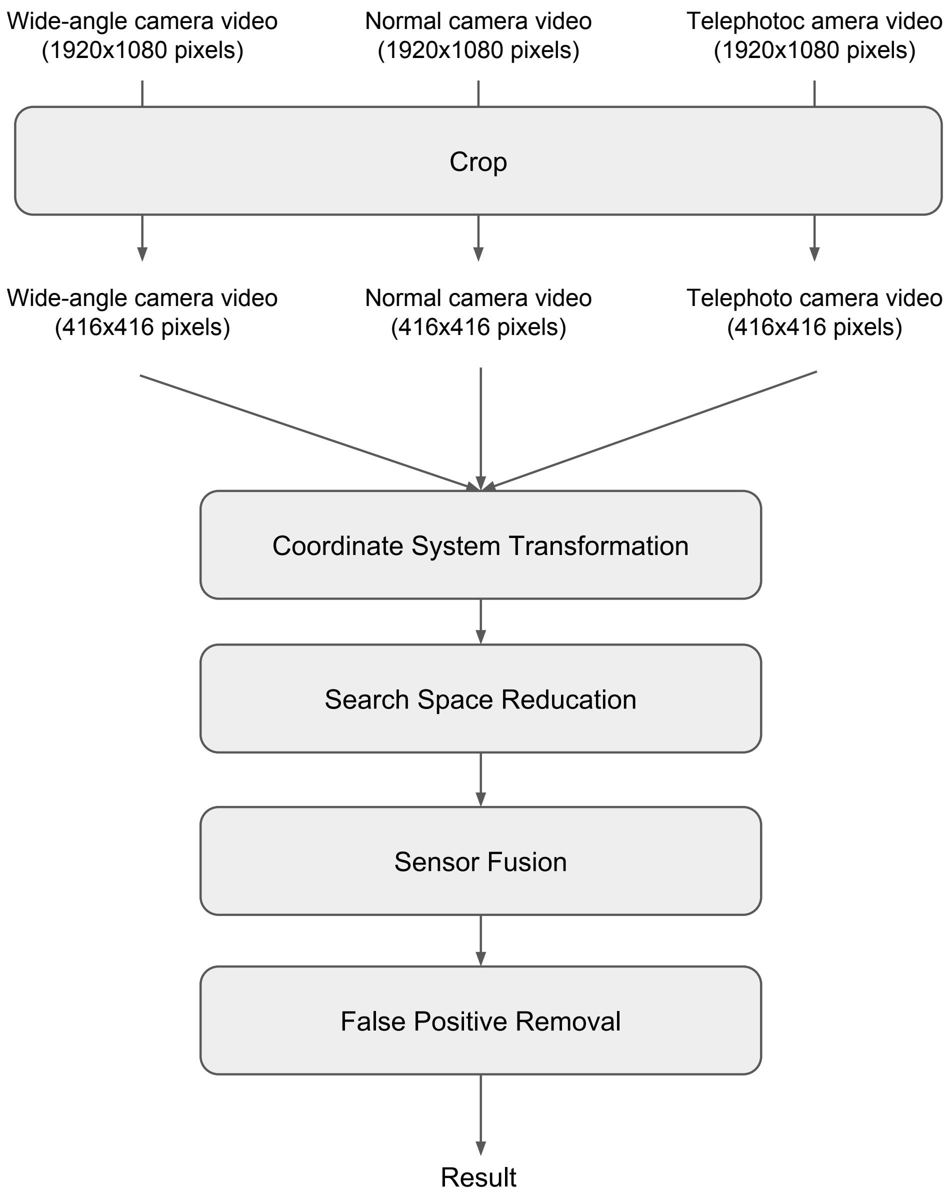


Figure 4.1: Data flow of the heterogeneous camera-based system.

doi:10.6342/NTU201800542

4.1 The Impact of Input Image Sizes

As we mentioned at section 2.2 , YOLO will resize the input image to 416 x 416 pixels before it feeds the input image into the convolution network. If the system uses a single frame which frame size is 1920 x 1080 pixels as an input image, YOLO will resize it into 416 x 416 pixels in advance. Consequently, the *recall* of using frame that frame size is 1920 x 1080 pixels as input image is worse than using frame that frame size is 416 x 416 pixels as input image. Table 4.1 shows the result of it. To solve this problem, the system crops the frame into several images which size are 416 x 416 pixels before the system invokes the YOLO system. As shown in Figure 4.2, the system crops a 416 x 416 pixels image (denote to the red rectangles) from each camera. These red rectangles are the region of interest (ROI) of wide-angle camera, normal camera, and telephoto camera since AEBS only concerns the target vehicles in front of the subject vehicle.

Size of input image (pixel)	Recall
1920 x 1080	45%
416 x 416	94.5%

Table 4.1: Recall of object detection using YOLO

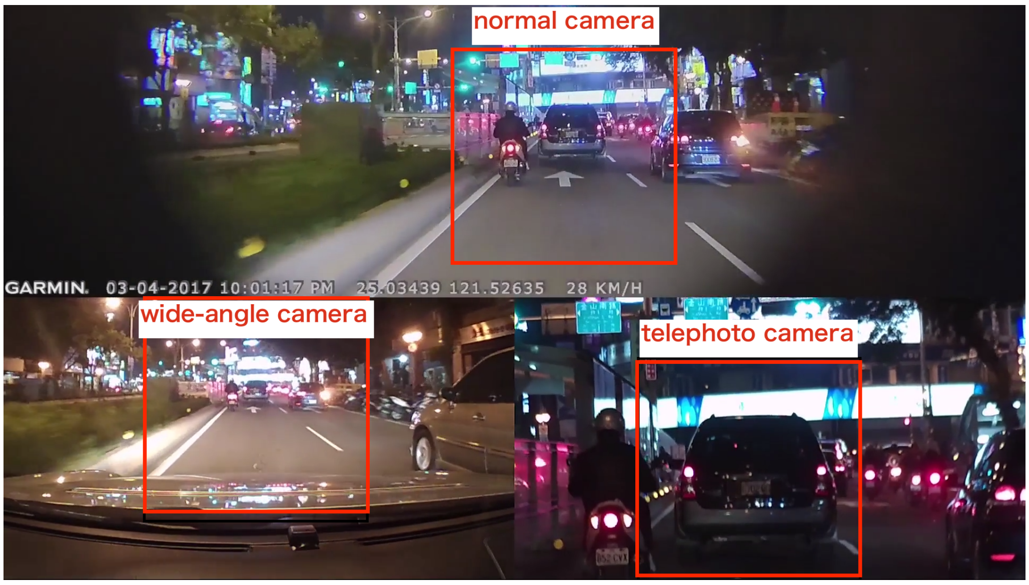


Figure 4.2: The frame that composed by three different FoV camera and the ROI of each camera.

doi:10.6342/NTU201800542

4.2 Coordinate System Transformation

In order to fuse all the vehicle rectangles that detected by different FoV cameras, we demand to transform these rectangles into the same coordinate system, which is "global coordinate system". In this chapter, we use N_1 to represent for wide-angle camera, N_2 to represent for normal camera, and N_3 to represent for telephoto camera. We use the linear function to transform the position of the detection rectangles into global coordinate system, and it can be shown as follows:

$$\begin{aligned} {x_i}' &= \frac{x_i}{mag_i} + X_i^{\mathit{offset}} \\ {y_i}' &= \frac{y_i}{mag_i} + Y_i^{\mathit{offset}} \end{aligned} \tag{4.1}$$

where x_i and y_i represent the position in the coordinate system of camera N_i; ${x_i}'$ and ${y_i}'$ represent the position in the global coordinate system; mag_i represents the magnification of camera N_i (That is, $mag_1 = 1$, $mag_2 = 2$, and $mag_3 = 4$); X_i^{offset} represents the translation offset of ${x_i}'$ in global coordinate system, and Y_i^{offset} represents the translation offset of ${y_i}'$ in global coordinate system where

$$\begin{aligned} X_i^{\mathit{offset}} &= W * (1 - \frac{1}{mag_i}) \\ Y_i^{\mathit{offset}} &= H * (1 - \frac{1}{mag_i}) \end{aligned} \tag{4.2}$$

W represents the width of croped input image of YOLO, and H represents the height of croped input image of YOLO.

4.3 Existed Sensor Fusion Method

As we mentioned at Section 2.2, YOLO filters the resulting detections by the model's confidence after it runs a single convolutional network. In YOLO, the confidence threshold of vehicle is 0.2. Therefore, if the confidence of the detection is lower than 0.2, this detection will be filtered out by YOLO. However, there are some detections that are true positive but its confidence is lower than the threshold. These detections will be filtered out

doi:10.6342/NTU201800542

by YOLO although they are true positive. If we integrate the low confidence detections from different FoV cameras, we can keep the true positives that are filtered out by YOLO system. Thus, we use an existed sensor fusion method, De Morgan's law, to fuse the low confidence detections from different FoV cameras into a new detection. Once the confidence of the new detection is higher than the threshold, YOLO will not filter out this true positive detection. And the $recall$ will be increased because of the increasing number of the true positive detections. The De Morgan's law for three sensors A, B, C can be written formally as

$$\overline{A \cup B \cup C} = \overline{A} \cap \overline{B} \cap \overline{C} \tag{4.3}$$

and the probability of the event detected by three sensors is defined as

$$P(A \cup B \cup C) = 1 - (1 - A) * (1 - B) * (1 - C) \tag{4.4}$$

where $P(A \cup B \cup C)$ denotes to the probability of $A \cup B \cup C$.

First of all, we use the method at Section 4.2 to transform all the vehicle rectangles that detected by different FoV cameras into global coordinate system. Second, we test if the detections from different FoV cameras denote to the same vehicle or not. Assume that the detection R_1 is detected by camera N_1, the detection R_2 is detected by camera N_2, and the detection R_3 is detected by camera N_3. In order to determine that the detections denote to the same vehicle, the overlap testing is defined as follows:

$$Overlap(R_1, R_2, R_3) = \begin{cases} true & , \frac{area\ of\ intersection}{area\ of\ union} > \text{Overlap threshold} \\ false & , \quad otherwise \end{cases} \tag{4.5}$$

and the value of Overlap threshold is 0.5 in this thesis. Once the detection R_1, R_2, R_3 from differents FoV cameras pass the overlap testing, we ensure that the detections R_1, R_2, R_3 denote to the same vehicle. At last, we use De Morgan's law to fuse these detections R_1, R_2, R_3 into a new detection, which is R_4. The position of R_4 is determined by the intersection rectangle of R_1, R_2, R_3, and we can obtain the confidence of R_4 by using De Morgan's law:

doi:10.6342/NTU201800542

$$C_4 = P(C_1 \cup C_2 \cup C_3) = 1 - (1 - C_1) * (1 - C_2) * (1 - C_3) \tag{4.6}$$

where C_1 represents the confidence of R_1, C_2 represents the confidence of R_2, C_3 represents the confidence of R_3, and C_4 represents the confidence of R_4.

4.4 Proposed Sensor Fusion Method

In this section, we propose a sensor fusion method that is based on De Morgan's law, which is "Weighted De Morgan's law". The significant difference between Weighted De Morgan's law and De Morgan's law is that we add a weighted function on it. The $recall$ of different FoV cameras are different when the distance between the target vehicle and the subject vehicle is the same. For instance, as shown in Figure 4.3, the recall of telephoto camera is higher than wide-angle camera when the distance between the target vehicle and the subject vehicle is 60 meters at night. At this distance, the confidence of detected vehicle rectangles from telephoto camera are more reliable than the rectangles that detected by wide-angle camera. On the other hand, the recall of wide-angle camera is higher than telephoto camera when the distance between the target vehicle and the subject vehicle is 10 meters at night. Because the capability of different FoV cameras under different distances are variaty, the weighted function in our proposed sensor fusion method will concern the capability of each camera.

The distance between the subject vehicle and the target vehicle is a significant information in the Weighted De Morgan's law. Since our AEBS architecture is camera-based system, we apply a camera-based method to observe the distance between the subject vehicle and the target vehicle rather than using radar or lidar sensors to measure the distance. The distance measurement method is to utilize the length of vehicle width in camera to measure the distance. The length of the target vehicle width in camera depends on the distance between the target vehicle and the subject vehicle. For instance, Figure 4.4 shows that the target vehicle is place at different distances between the subject vehicle. In this thesis, we assume the minimum vehicle width is 1.5 meters. Figure 4.5

doi:10.6342/NTU201800542

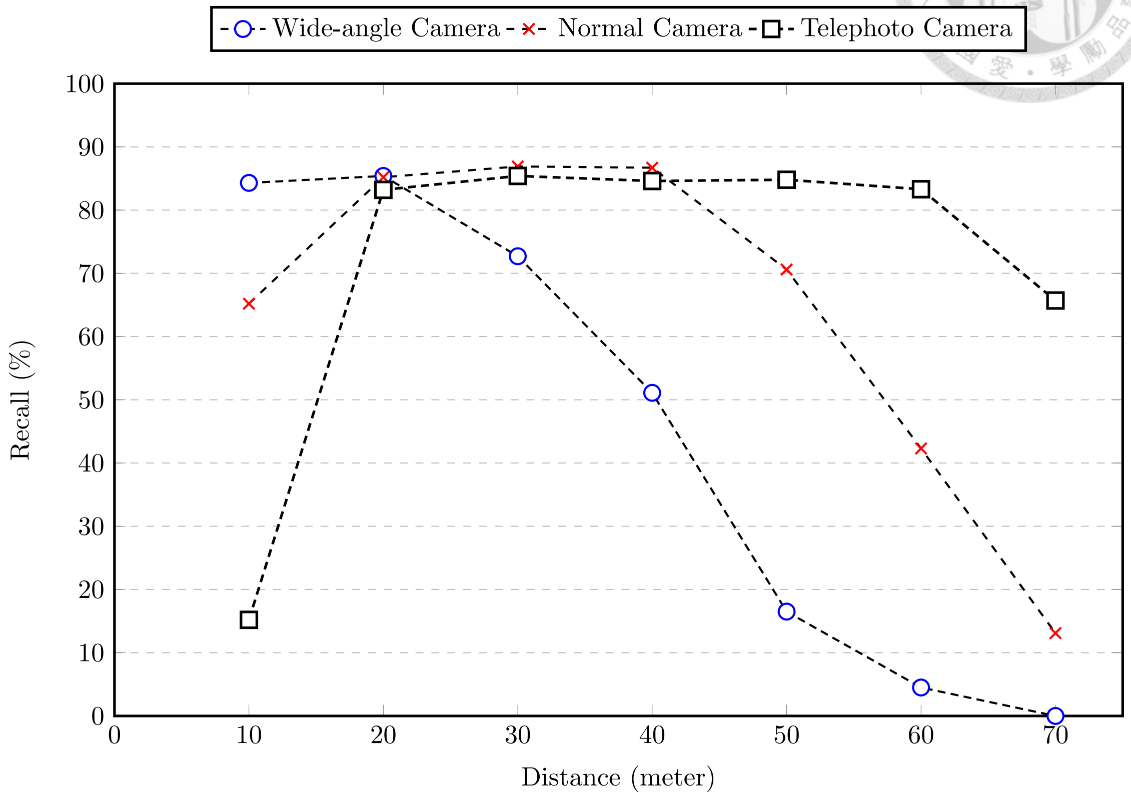
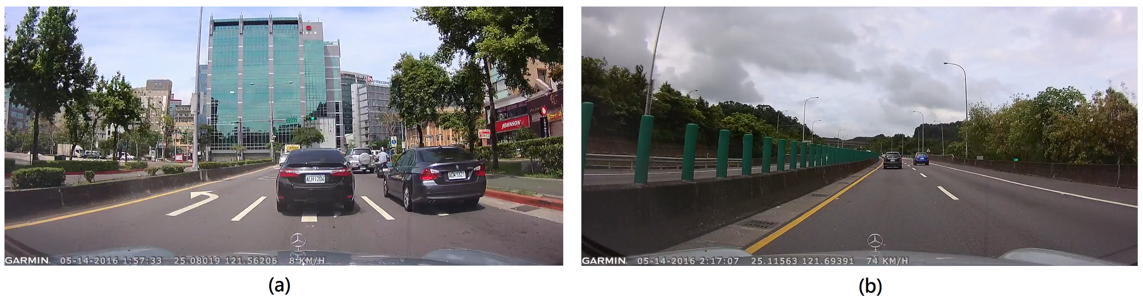


Figure 4.3: The recall of different FoV camera when the target vehicle is placed at different distance at night.



GARMIN. 05-14-2016 1:57:33 25.08019 121.56206 8 KM/H
(a)
GARMIN. 05-14-2016 2:17:07 25.11563 121.69391 74 KM/H
(b)

Figure 4.4: The target vehicle is placed at different distances. (a) The distance is 10 meters. (b) The distance is 40 meters.

doi:10.6342/NTU201800542

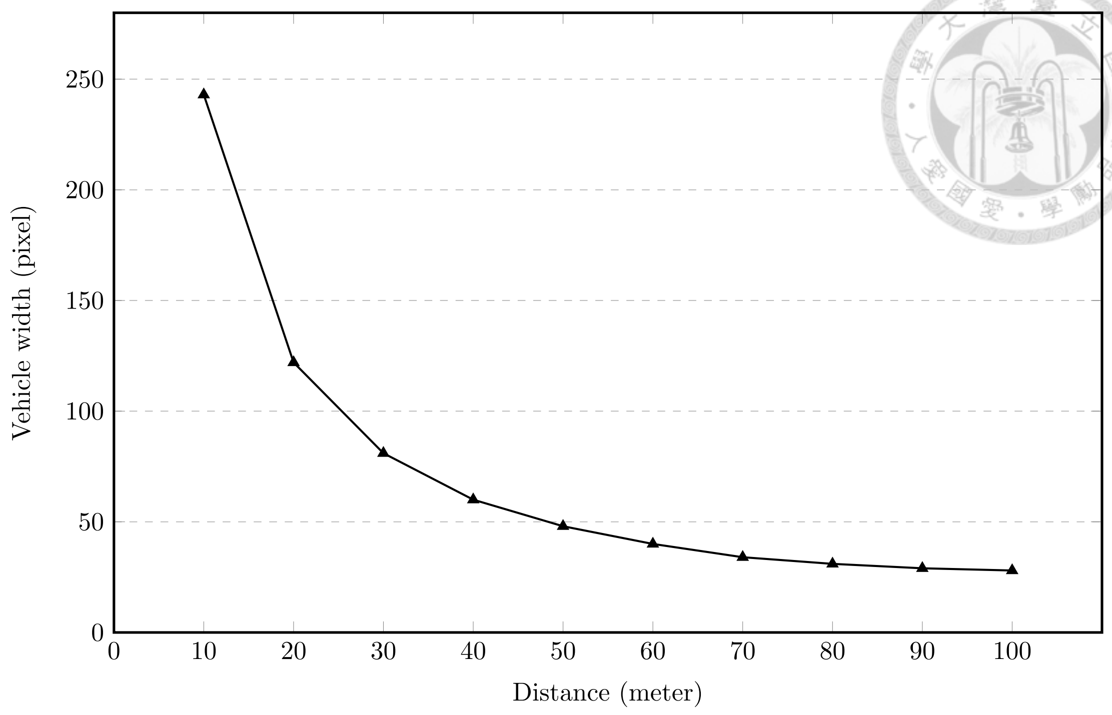


Figure 4.5: The length of the target vehicle width at different distances. The input frame size is 1920x1080 pixels and we assume the target vehicle width is 1.5 meters.

shows the experiment of the length of the target vehicle width at different distances between the subject vehicle. The $weight$ in the Weighted De Morgan's law is a function to the $recall$ of each camera under the certain distance. For instance, when the distance between the target vehicle and the subject vehicle is 60 meters, the $recall$ of wide-angle camera is 4.5, the $recall$ of normal camera is 42.3 and the $recall$ of telephoto camera is 83.3. Thus, the $weight$ of wide-angle camera is $4.5/(4.5 + 42.3 + 83.3) * 3$, the $weight$ of normal camera is $42.3/(4.5 + 42.3 + 83.3) * 3$ and the $weight$ of telephoto camera is $83.3/(4.5 + 42.3 + 83.3) * 3$.

The Weighted De Morgan's law for three sensors A, B, C can be written formally as

$$\overline{\alpha A \cup \beta B \cup \gamma C} = \overline{\alpha A} \cap \overline{\beta B} \cap \overline{\gamma C} \tag{4.7}$$

and the probability of the event detected by three sensors is defined as

$$P(\alpha A \cup \beta B \cup \gamma C) = 1 - (1 - A)^{\alpha} * (1 - B)^{\beta} * (1 - C)^{\gamma} \tag{4.8}$$

doi:10.6342/NTU201800542

where $P(\alpha A \cup \beta B \cup \gamma C)$ denotes to the probability of $\alpha A \cup \beta B \cup \gamma C$. α, β, and γ represents the $weight$ of each camera.

The fusion step is the same as De Morgan's law. Once the detections R_1, R_2, R_3 from different FoV cameras pass the overlap testing, we use Weighted De Morgan's law to fuse these detections R_1, R_2, R_3 into a new detection, which is $R_4{}'$. The position of $R_4{}'$ is determined by the intersection rectangle of R_1, R_2, R_3, and we can obtain the confidence of $R_4{}'$ by using Weighted De Morgan's law:

$$C_4{}' = P(\alpha C_1 \cup \beta C_2 \cup \gamma C_3) = 1 - (1 - C_1)^{\alpha} * (1 - C_2)^{\beta} * (1 - C_3)^{\gamma} \tag{4.9}$$

where $C_4{}'$ represents the confidence of $R_4{}'$, α represents the $weight$ of the confidence C_1 , β represents the $weight$ of the confidence C_2, and γ represents the $weight$ of the confidence C_3. The value of α, β, γ are defined as follows:

$$\begin{aligned} \alpha &= \frac{r_{D,N_1}}{\sum_{k=1}^{T} r_{D,N_k}} * T \\ \beta &= \frac{r_{D,N_2}}{\sum_{k=1}^{T} r_{D,N_k}} * T \\ \gamma &= \frac{r_{D,N_3}}{\sum_{k=1}^{T} r_{D,N_k}} * T \end{aligned} \tag{4.10}$$

where D represents the distance between the target vehicle and the subject vehicle, T represents the total number of cameras, r_{D,N_k} represents the $recall$ of camera N_k when the distance between the target vehicle and the subject vehicle is D.

4.5 False Positive Removal

Since the target vehicle will not change its position significantly between consequently frames, we can use the information from previous frames to remove the false positive noise. We use a heuristic algorithm to solve the problem of increasing false positive rate. The main idea of this algorithm is using the information from previous frames to remove

doi:10.6342/NTU201800542

the false positive in current frame. Here we use a buffer to store the information from previous frames. Once the detection R appears (no matter this detection R is produced by YOLO or Weighted De Morgan's law) in current frame, we will check the buffer. If there are no others detections that confidence greater than 0.2 in previous frames appear at the same position, the detection R is considered as a noise and we will filter out the detection R as a noise in current frame. Otherwise, the detection R will be kept in current frame. At last, we add all the detections into our buffer. If the a new detection R' appears at the same position as R in the next frame, the new detection R' will be considered as a true positive since the detection R appeared at the same position before. Also, the buffer will update the information of the new frame and discard the information that past for a long time. The number of the stored frames depends on the buffer size.

Algorithm 1 False Positive Removal

```
 1: R.lx/R.ly : the left up corner x/y of detection R.
 2: R.rx/R.ry : the right down corner x/y of detection R.
 3: R.confidence : the confidence of detection R.
 4: Buff[x][y][Buff_SIZE] : the array buffer to store the information of previous frames.
 5: for each detection R in current frame do
 6:     int pixels = 0;
 7:     for y = R.ly; y < R.ry; y++ do
 8:         for x = R.lx; x < R.rx; x++ do
 9:             int buff_count = 0;
10:             for j = 0; j < Buff_SIZE; j++ do
11:                 if Buff[y][x][j] == true then
12:                     buff_count++;
13:                 end if
14:                 if buff_count > FrequencyThreshold then
15:                     pixels++;
16:                 end if
17:                 if R.confidence > 0.2 then
18:                     Buff[y][x][frame_id%Buff_SIZE] = true;
19:                 end if
20:             end for
21:         end for
22:     end for
23:     float IntersectArea = pixels / [(R.rx-R.lx) * (R.ry-R.ly)];
24:     if IntersectArea < AreaThreshold then
25:         ignore the detection R
26:     end if
27: end for
```

doi:10.6342/NTU201800542

4.6 Search Space Reducation

Before we use Weighted De Morgan's law to enhance the confidence of the detections from different FoV cameras, we need to find out that which detections are denoted to the same vehicles in all detections. This problem can be transform to maximal clique problem. We can transform the detections in global coordinate system into a simple undirected graph. In the undirected graph, each node represents an entity (such as detection rectangle) and each edge represents that these two nodes pass the overlap testing. Notice that if the two nodes (denoted to the detection rectangles) are detected by the same camera, they will not be connected by an edge even though they pass the overlap testing. Figure 4.6 (a) shows an example of global coordinate system that the red rectangles represent the detections of camera N_1, the blue rectangles represent the detections of camera N_2, and the black rectangles represent the detections of camera N_3. In this example, only (R_1, R_3), (R_1, R_5), (R_3, R_5), (R_1, R_3, R_5), (R_2, R_4) pass the overlap testing. That is, we will fuse (R_1, R_3, R_5) and (R_2, R_4) by using Weighted De Morgan's law. As shown in Figure 4.6 (b), we transform the detection rectangles in global coordinate system into an undirected graph. In undirected graph, (R_1, R_3, R_5) and (R_2, R_4) are the maximal clique. The result of maximal clique in the graph is the same as the overlap testing of all detection rectangles in global coordinate system. Thus, we proof that the problem of finding detections that are denoted to the same vehicles in all detections can be transform to maximal clique problem.

However, the maximal clique problem is NP-complete, and it can not be solved in the polynomial time. Since we want to construct a safety critical embedded real-time systems, the shorter response time of the system is better. Thus, instead of constructing all the detection rectangles in a large undirected graph, we construct several small undirected graph to reduce the cost of time. In each iteration, we will focus on a detection rectangles and construct an undirected graph for it. We only do the Weighted De Morgan's law for the maximal clique who covers the node that we focus on in this iteration. Assume that we focus on R_1 in this iteration. R_1 only intersect to $R_2 \sim R_6$ in global coordinate system as shown in Figure 4.7. If we use all the detection rectangles to construct the

doi:10.6342/NTU201800542

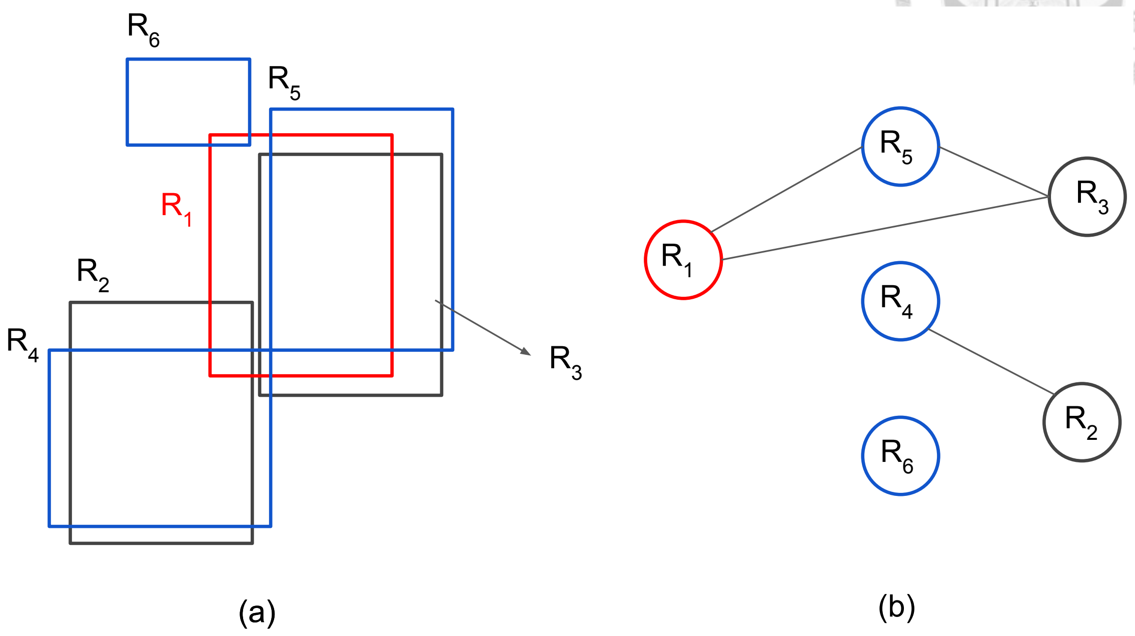


Figure 4.6: An example of transforming the detection rectangles in global coordinate system into an undirected graph.

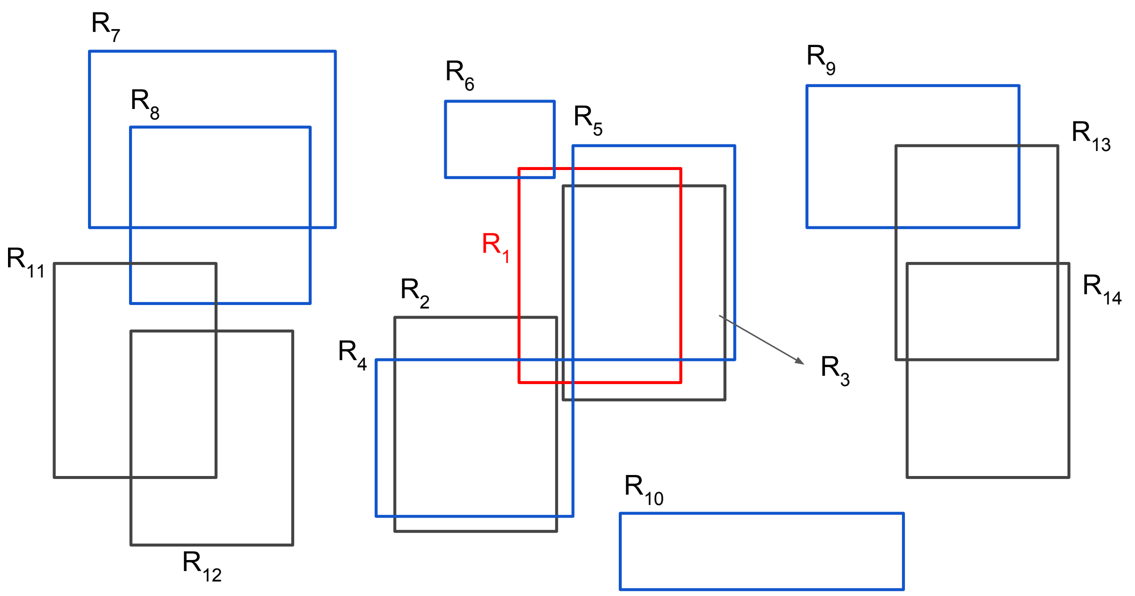


Figure 4.7: An example of global coordinate system.

doi:10.6342/NTU201800542

undirected graph, there will be lots of nodes that don't have the connection between R_1 and itself. These nodes is invaluable when we are focus on R_1. Thus, if we can use a search algorithm to find out the detections that pass the overlap testing with R_1 before we transform the detections from global coordinate system into an undirected graph, we can reduce the number of invaluable nodes in the graph.

We can use the regional of the vehicle detections to reduce the search space although we cannot solve the NP-complete problem in the polynomial time. Since the detected vehicle rectangles denote to represent the position of the target vehicles, these detected vehicle rectangles will appear closely in the global coordinate system. As shown in Figure 4.8, the red rectangles are detected by wide-angle camera, the yellow rectangles are detected by normal camera, and the green rectangle is detected by telephoto camera. We can observe that the overlap between vehicles is seldom. The reason of the situaction is that light is straightforward in nature. Assume that there is another vehicle T' in the front of the target vehicle T in Figure 4.8. The YOLO system cannot detect T' because of the property of light. There will not have a detection that covers another detection under the camera-based system. Thus, the overlap situation between all the detections is limited. That is why we can use a search algorithm to filter out those who cannot pass the overlap testing detections to reduce the search space to speed up the sensor fusion method.

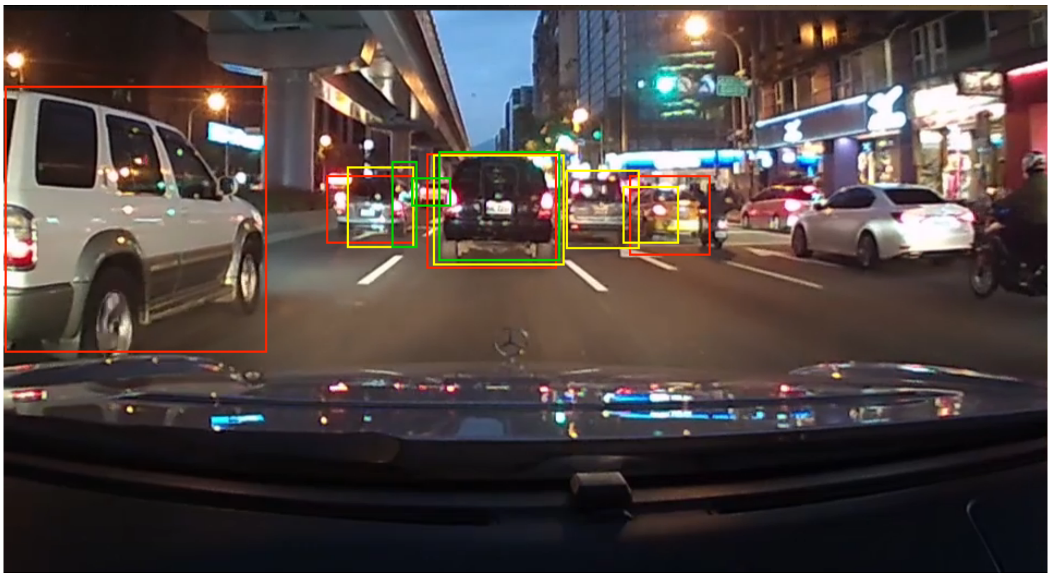


Figure 4.8: An example of overlap situation.

doi:10.6342/NTU201800542

R-tree is a known solution for solving the multi-dimensional information searching problem. It use hierarchical MBRs to obtain the better performance. For each frame, we construct a R-tree to store the detections in the global coordinate system. We can obtain the detection rectangles that the focused detection R intersect to by going through the R-tree once. Figure 4.9 shows an example of using R-tree to speed up the searching problem. Figure 4.9 (a) is the result of transforming all the detections from each FoV camera into global coordinate system. The purple rectangles represent the detected vehicle rectangles. Figure 4.9 (b) and (c) shows the progress of constructing R-tree for the detected vehicle rectangles. We assume a maximum branching number $M = 3$ and minimum branching number $m = 2$. The blue and green rectangles represent the MBRs of R-tree. Figure 4.9 (d) shows the constructed R-tree of this frame. Figure 4.9 (e) $\sim$ (g) shows the progress of finding the detections that intersect to the focused detection by using R-tree (assuming we focus on the detection R_4). We can observe that the nubmer of detections in the global coordinater system in Figure 4.9 (g) is less than Figure 4.9 (a).

The Grid method is also a known solution for searching algorithm. The Grid method divide the global coordinate system into several grids G. If a detection intersects to a grid G_i, then the detection will be added to the list of grid G_i. Different from R-tree, the detection in Grid method may appear twice, or even three times, four times in the lists of G_i. The usage of grids G in Grid method is similar to the MBRs in R-tree. We can obtain the detection rectangles that the focused detection R intersect to by using these grids G. First, we will check whether the focused detection intersect to grid G_i or not. We only need to check the detections in the list of G_i if the grid G_i intersect to the focused detection. Figure 4.10 shows an example of using Grid method to speed up the searching problem. Figure 4.10 (a) is the result of transforming all the detections from each FoV camera into global coordinate system. The purple rectangles represent the detected vehicle rectangles. Figure 4.10 (b) shows that the global coordinate system is divided into several grids (we assume the global coordinate system is divided into 9 grids in this example). Each detection will be added into the grid list G_i if the detection intersect to the grid G_i. Figure 4.10 (c) shows all the content in grid lists from $G_1 \sim G_9$.

doi:10.6342/NTU201800542

Figure 4.10 (d) and (e) shows the progress of finding the detections that intersect to the focused detection by using Grid method(assuming we focus on the detection R_4).

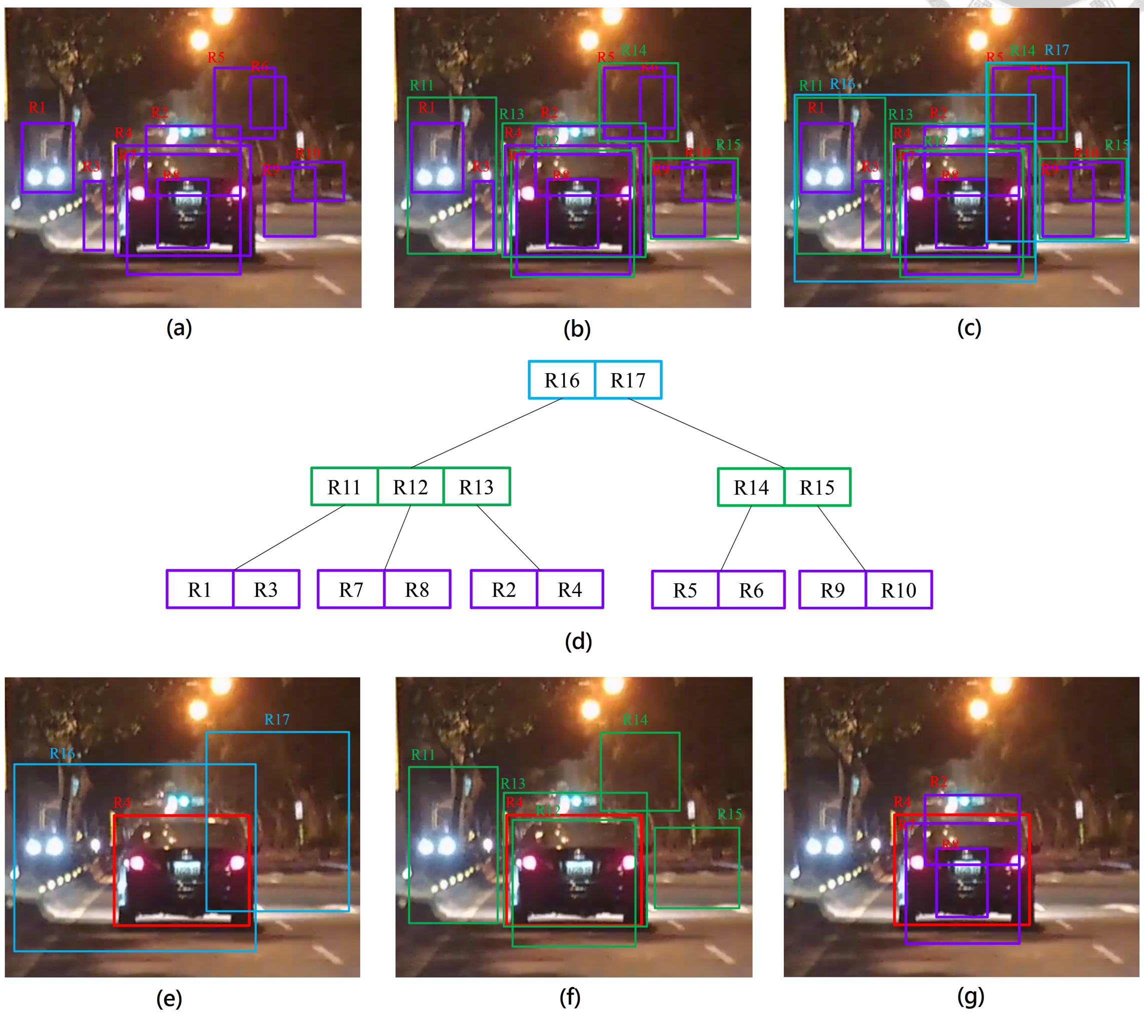


Figure 4.9: An example of using R-tree to speed up the searching problem.

doi:10.6342/NTU201800542

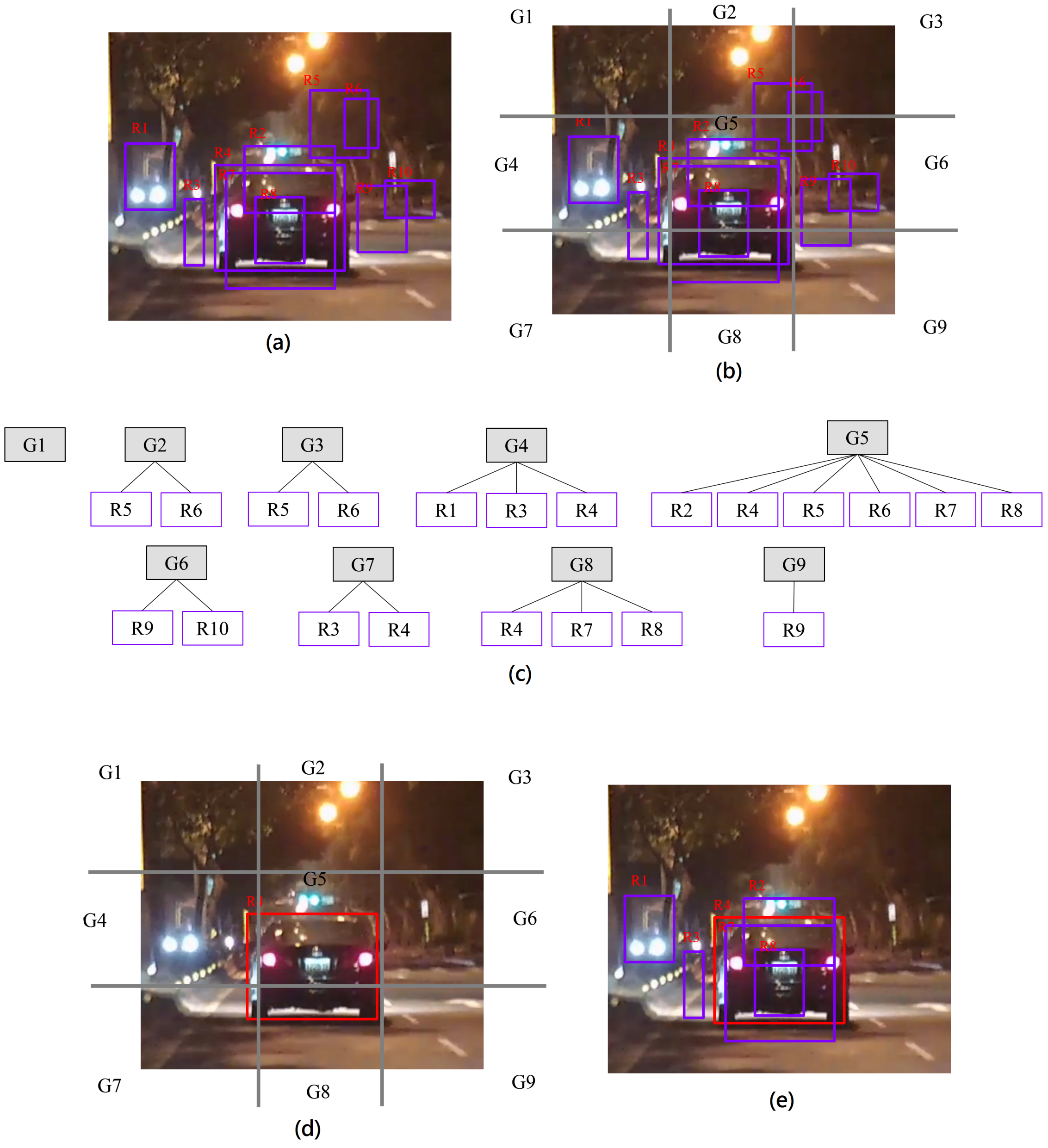


Figure 4.10: An example of using Grid method to speed up the searching problem.

doi:10.6342/NTU201800542

Chapter 5

Performance Evaluation

In this chapter, we evaluate the experiment result of our design, including the $recall$ of the sensor fusion method, the $precision$ of false positive removal method and the performance measurement on NVIDIA TX2. Since the AEBS only concerns the target vehicle, the $recall$ and $precision$ below represents the $recall$ and the $precision$ of the target vehicle. Table 5.1 shows the definition of $recall$ and $precision.$ For the notation TP_i and FP_i, we use intersection over union (IOU) to determine whether two rectangles is the same. If the IOU of the detected rectangle and the ground truth rectangle is higher than 0.5, we will count the detected rectangle into TP_i. Otherwise, we will count it into FP_i.

Notation	Definition
G_i	Ground truth rectangles of the target vehicles of frame i
D_i	Detected rectangles of frame i
TP_i	$G_i \cap D_i$
FP_i	$D_i - G_i \cap D_i$
FN_i	$G_i - G_i \cap D_i$
$recall$	$\frac{\sum size(TP_i)}{\sum size(TP_i)+\sum size(FN_i)}$
$precision$	$\frac{\sum size(TP_i)}{\sum size(TP_i)+\sum size(FP_i)}$

Table 5.1: Notation table

doi:10.6342/NTU201800542

5.1 Evaluation of Sensor Fusion Method

In this section, we evaluate the $recall$ of the sensor fusion method under different scenarios. Figure 5.1 shows the $recall$ of different FoV cameras, De Morgan's law and Weighted De Morgan's law in the night scenario and Figure 5.2 shows the $recall$ of different FoV cameras, De Morgan's law and Weighted De Morgan's law in the sunny scenario. We can observe that the De Morgan's law and Weighted De Morgan's law have a significant effect on $recall$ in the night scenario. In the sunny scenario, we can observe that the $recall$ of telephoto camera is higher than 90% until the distance between the target vehicle and the subject vehicle is greater than 140 meters. Thus, the effect of De Morgan's law and Weighted De Morgan's law in the sunny scenario is smaller than the night scenario. Luminous intensity is the reason of this phenomenon. A camera's shutter determines when the camera sensor will be open or closed to incoming light from the camera lens. The shutter speed specifically refers to how long this light is permitted to enter the camera. "Shutter speed" and "exposure time" refers to the same concept, where a faster shutter speed means a shorter exposure time. In general, the cameras need more exposure time in the night scenario since the luminous intensity in the night scenario is lower than the sunny scenario. Besides, the focal length of telephoto camera is higher than other cameras. Therefore, the motion of a camera has a great effect on it during exposure. If the camera moves quickly during exposure, the result image will turn into a blurred image. Thus, the $recall$ of telephoto camera in the night scenario is lower than the $recall$ in the sunny scenario. Because the motion of a camera has a little effect on wide-angle camera and normal camera, the image will not be blurred so much as telephoto camera. The non-blurred image from wide-angle camera and normal camera reinforcement the blurred image from telephoto camera during the sensor fusion method. This is the reason that why De Morgan's law and Weighted De Morgan's law have a signigicant effect in the night scenario. In summary, the difference of exposure time between night and sunny scenario results in the different effect on the $recall$ of De Morgan's law and Weighted De Morgan's law.

doi:10.6342/NTU201800542

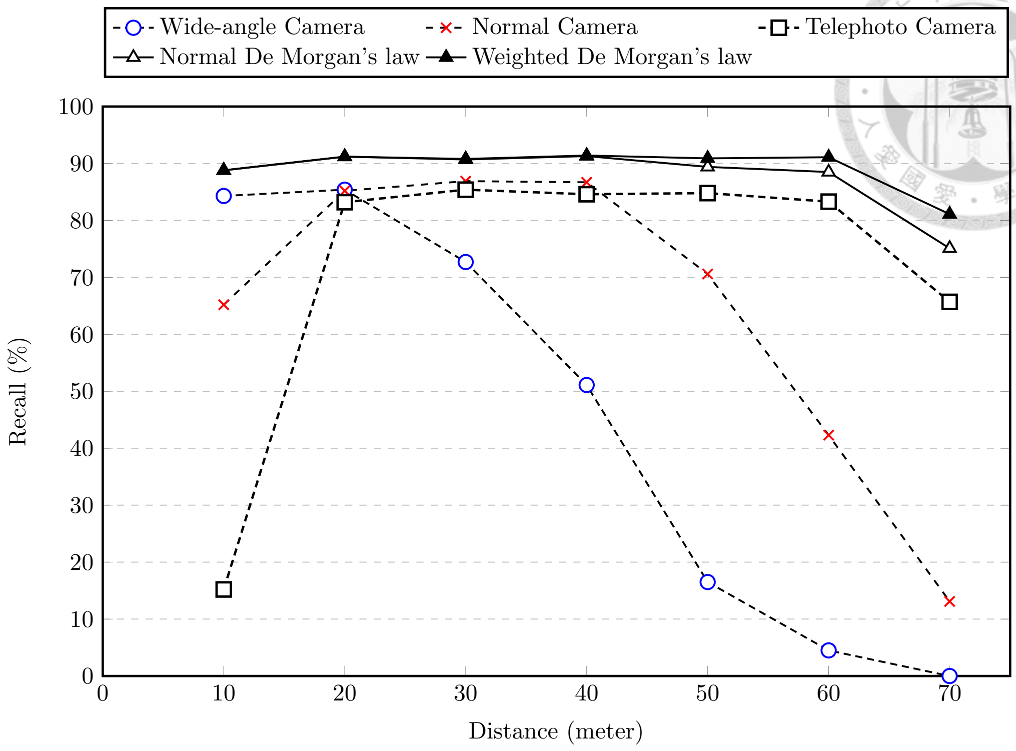


Figure 5.1: The recall of different FoV cameras, De Morgan's law, Weighted De Morgan's law when the target vehicle is placed at different distance at night.

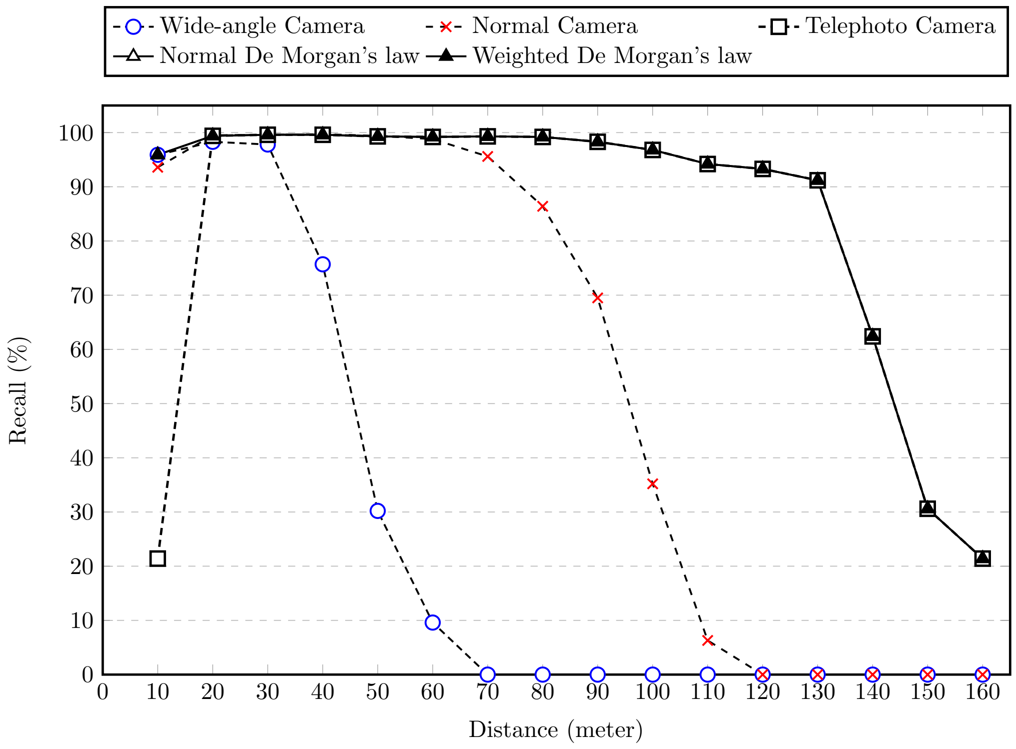


Figure 5.2: The recall of different FoV cameras, De Morgan's law, Weighted De Morgan's law when the target vehicle is placed at different distance at sunny.

doi:10.6342/NTU201800542

5.2 Evaluation of False Positive Removal

Figure 5.3 shows the *precision* of Weighted De Morgan's law and Weighted De Morgan's law with false positive removal method. We can observe that the *precision* of the Weighted De Morgan's law can be increased up to 8% by using the false positive removal method. However, the *precision* increases 2% when the distance between the target vehicle and the subject vehicle is 10 meters. The reason is described below. When the distance between the target vehicle and the subject vehicle is 10 meters, the target vehicle will be very large in telephoto camera that only the part of vehicle is in the ROI of telephoto camera. Since the telephoto camera only see the part of vehicle, it is hard to determine whether it is a vehicle or not. Therefore, the number of true positive and false positive of telephoto camera will be decreased. Because the number of false positive is decreased, the false positive removal method have smaller effect on it.

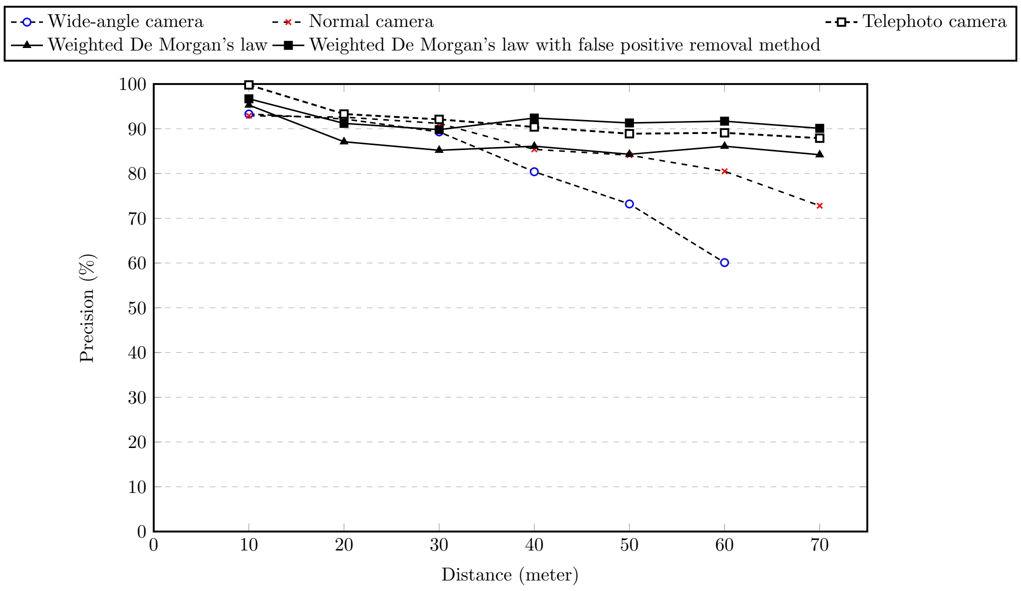


Figure 5.3: The precision of different FoV cameras, Weighted De Morgan's law and False positive removal method

doi:10.6342/NTU201800542

5.3 Performance Measurement on NVIDIA TX2

In this section, we test our heterogeneous camera-based system, and evaluate the system by time consumption. We test the system under night scenario and sunny scenario. The maximum number of rectangles that generated by YOLO system is around 100 rectangles. Thus, the maximum number of all the detection rectangles in our system is around 350 rectangles since we use three different focal length cameras in our camera-based system. Figure 5.4 shows the performance measurement on NVIDIA TX2. The time cost of using R-tree to reduce the search space is minimum. The reason that why R-tree is faster than Grid method is described below. Since the detections only appear once in R-tree, we can get all the detections that the focused detection R intersects to by a single query. However, the detections may appear twice, three times or four times in Grid method. Assume that the detections that the focused detection R intersects to are stored in the list L. To avoid a repetitive adding the same detection into L, the Grid method needs to check L at each iteration. That why the reason that the Grid method is slower than R-tree.

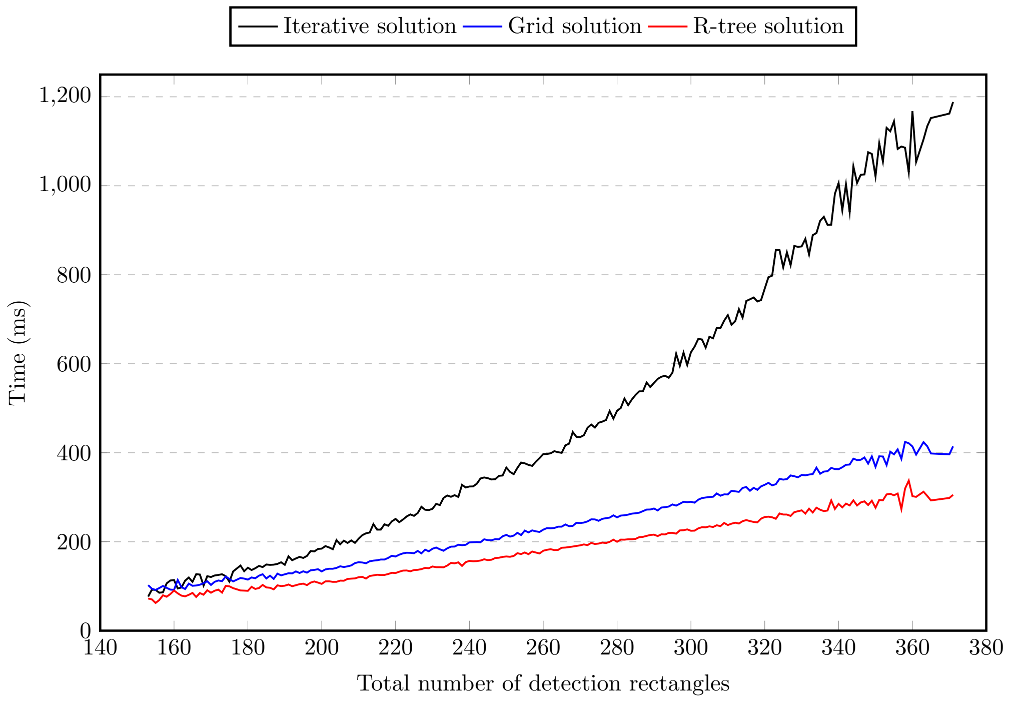


Figure 5.4: Performance measurement on NVIDIA TX2

doi:10.6342/NTU201800542

Chapter 6

Conclusion

In this thesis, we design and implement a heterogeneous camera-based system on AEBS to enhance the recall of YOLO system by using sensor fusion method to combine the strengths of the different FoV cameras. Also, we use a heuristic false positive removal method to decrease the false positive rate that caused by the sensor fusion method. We optimize the sensor fusion method because of the the limitation of computing resource on embedded system. As a result, the recall of YOLO can be increased up to 10% through our heterogeneous camera-based system.

doi:10.6342/NTU201800542

Bibliography

[1] "The national highway traffic safety administration research in 2015," https://www.nhtsa.gov/equipment/safety-technologies.

[2] P. P. D. W. R. Fernandes, C. Premebida and U. Nunes, "Road Detection Using High Resolution LIDAR," *Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–6, 2014.

[3] A. A. A. P. R. Cristiano Premebida, Luis Garrote and U. Nunes, "High-resolution LIDAR-based Depth Mapping using Bilateral Filter," *International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2469–2474, 2016.

[4] B.-H. L. A. L. Sang-Mook Lee, Jeong Joon Im and A. Kurdila, "A real-time grid map generation and object classification for ground-based 3D LIDAR data using image analysis techniques," *International Conference on Image Processing*, pp. 2253–2256, 2010.

[5] E. Guizzo, "How Google's self-driving car works," *IEEE Spectr. Online*, vol. 18, 2011.

[6] "Google just made a big move to bring down the cost of self-driving cars," http://www.businessinsider.com/googles-waymo-reduces-lidar-cost-90-in-effort-to-scale-self-driving-cars-2017-1.

[7] "Unece r131 regulation," https://www.unece.org/trans/main/wp29/wp29regs121-140.html.

[8] M. B.-A. J. P. C. C. L. Azevedo, J. L. Cardoso and M. Marques, "Automatic vehicle trajectory extraction by aerial remote sensing." *Proc. Soc. Behavioral Sci*, vol. 111, pp. 849–858, 2013.

[9] A. C. Shastry and R. A. Schowengerdt, "Airborne video registration and traffic-flow parameter estimation," *IEEE Trans. Intell. Transp. Syst*, vol. 6, no. 4, pp. 391–405, 2005.

[10] R. G. J. Redmon, S. Divvala and A. Farhadi, "You only look once: Unified, real-time object detection," *in Computer Vision and Pattern Recognition*, 2015.

doi:10.6342/NTU201800542

[11] J. Redmon and A. Farhadi., "Yolo9000: Better, faster, stronger." *in Computer Vision and Pattern Recognition*, 2016.

[12] A. Guttman, "R-Tree: A Dynamic Index Structure for Spatial Searching," *Proc. ACM SIGMODE*, pp. 47–57, 1984.

[13] I. Kamel and C. Faloutsos, "On packing r-trees." *In Proc. 2nd International Conference on Information and Knowledge Management(CIKM-93)*, pp. 490–499, 1993.

[14] N. Roussopoulos and D. Leifker, "Direct spatial search on pictorial databases using packed r-trees." *Proc. ACM SIGMOD*, pp. 17–31, 1985.

[15] I. Kamel and C. Faloutsos, "Hilbert Rtree: An improved R-tree using fractals." *In Proceedings of the Twentieth International Conference on Very Large Data Bases*, pp. 500–509, 1994.

[16] H. tae Kiml and B. Songl, "Vehicle Recognition Based on Radar and Vision Sensor Fusion for Automatic Emergency Braking." *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pp. 1342–1346, 2013.

[17] R. D. M. B. M. Fazeen, B. Gozick and M. C. Gonzalez, "Safe Driving Using Mobile Phones," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1462–1468, 2012.

[18] U. N. P. P. C. Premebida, G. Monteiro, "A Lidar and Vision-based Approach for Pedestrian and Vehicle Detection and Tracking." *Intelligent Transportation Systems Conference 2007*, pp. 1044–1049, 2007.

[19] H. tae Kiml and B. Songl, "Vehicle Recognition Based on Radar and Vision Sensor Fusion for Automatic Emergency Braking." *13th International Conference on Control, Automation and Systems (ICCAS 2013)*, p. 1342–1346, 2013.

[20] "Nvidia jetson. the embedded platform for autonomous everything." http://www.nvidia.com/object/embedded-systems-dev-kits-modules.html.

[21] "Surrounded by ai devices that do everything from flying to farming, nvidia launches jetson tx2." https://blogs.nvidia.com/blog/2017/03/07/surrounded-by-ai-devices-that-do-everything-from-flying-to-farming-nvidia-launches-jetson-tx2/.

doi:10.6342/NTU201800542