Keypoint-based Binocular Distance Measurement for Pedestrian Detection System on Vehicle

by

Mingchang Zhao

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.A.Sc degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

The Pedestrian Detection System (PDS) has become a significant area of research designed to protect pedestrians. Despite the huge number of research work, the most current PDSs are designed to detect pedestrians without knowing their distances from cars. In fact, a priori knowledge of the distance between a car and pedestrian allows this system to make the appropriate decision in order to avoid collisions. Typical methods of distance measurement require additional equipment (e.g., Radars) which, unfortunately, cannot identify objects. Moreover, traditional stereo-vision methods have poor precision in long-range conditions. In this thesis, we use the keypoint-based feature extraction method to generate the parallax in a binocular vision system in order to measure a detectable object; this is used instead of a disparity map. Our method enhances the tolerance to instability of a moving vehicle; and, it also enables binocular measurement systems to be equipped with a zoom lens and to have greater distance between cameras. In addition, we designed a crossover re-detection and tracking method in order to reinforce the robustness of the system (one camera helps the other reduce detection errors). Our system is able to measure the distance between cars and pedestrians; and, it can also be used efficiently to measure the distance between cars and other objects such as Traffic signs or animals. Through a real word experiment, the system shows a 7.5% margin of error in outdoor and long-range conditions.

Acknowledgements

I would like to thank my mentor Professor Azzedine Boukerche for giving me an excellent research opportunity in PARADISE Research Laboratory. I would like to offer my sincere gratitude for his encouragement and knowledge and for the financial support he has provided me with throughout my master, in addition to the experience he has showed with me. Without him this thesis would not have been written or completed.

I wish to express my gratitude to my group leader, Dr. Abdelhamid Mammeri; he has been my guide during my research and a friend in life. He is so selfless and has been willing to solve problems for me during the research.

I would like to thank all PARADISE researchers, especially Mobile Vision Group members. They have provided me with so much inspiration, support and friendship during these two years.

Most of all, I would like to express my great thanks to my parents for supporting me throughout my studies at the University of Ottawa.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Advanced Driver Assistance Systems (ADAS) refer to the set of smart components (e.g., pedestrian protection system, lane detection and tracking) used by vehicles to detect danger and avoid it. An important task which should be performed by ADAS is to measure the distance between a vehicle and a detected target, e.g., a pedestrian. For this purpose, pattern vision-based techniques and Radar distance measurement systems are used in the automotive industry. The two parts of detection and distance measurement are usually independent; this is due to constraints in micro computer performance. However, with the improvement of processor speed, the smart car can be given the ability to recognize a specific target and to determine the distance at the same time all through the image processing technique.

For human beings, the judgement of distance and depth, is performed by combining images on our two retinas (see Figure 1.1); this works in such a way that we are not aware of this ability, because we are not psychologists or visual physiologists. For example, when watching 3-D movies. one must wear special glasses, based on stereopsis. Driving a car or bicycle, skating, skiing, or just performing as everyday task like grabbing your cup from a table for even a few minutes with one eye closed; leads to uncoordinated movements compared with two open eyes. That is our innate dimensional sense.

Figure 1.1: Stereopair of the Cloisters. The right photograph was taken by a camera about 3 inches to the right of the camera that the photograph on the left was taken.[1]

Actually, a human being can judge the distance using other techniques [1]. If the approximate size of an object, such as a person or a car, is known, we can tell the distance by the principle that objects appear to be smaller when they are further away from us. However, we might become confused by a smaller model of the object that is expected. We also might be confused about the position of two objects having dissimilar sizes at different distances, which appeared in the same size in a particular point of view, as shown in Figure 1.2a. If a object is partly blocked by another object, we considers the object on top closer. But, the Penrose stairs (Figure 1.2b) show the risk in this kind of judgement. Parallel lines in an image extend towards one vanishing point as the distance increases, for example road lanes are an example of perspective. Shadow is another indicator of depth. Because a light source always comes from above, we judge that a bump on a wall that juts out is brighter on top. However, this judgement is not

reliable - for example, while looking at a picture of moon surface (Figure 1.2c), we are sometimes under the misapprehension of thinking that the craters are hummocks.

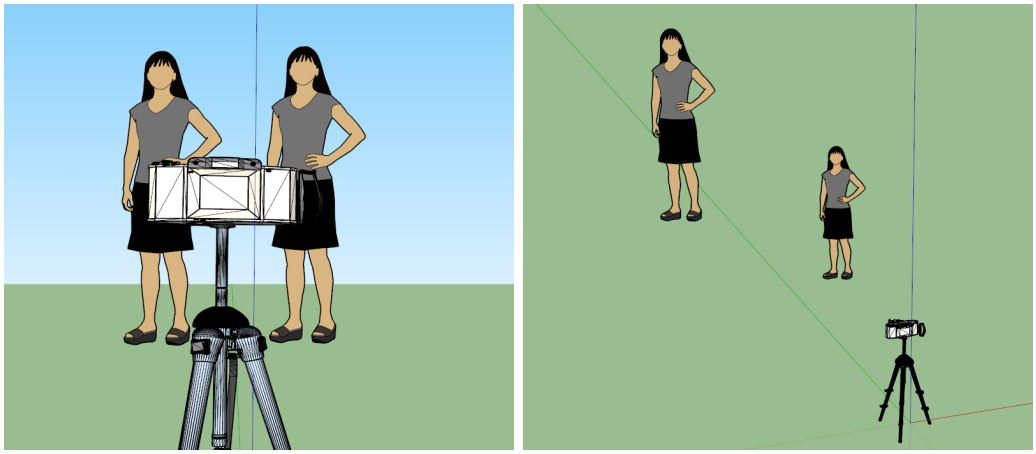
For a smart vehicle, driving safety is not only related to whether there is a pedestrian near the road but also how far away he is, not only whether there is another vehicle in front of my car but also whether it is approaching or leaving my car. So it is necessary for a vehicle to have a sense of stereopsis, and one of the most reliable methods is binocular vision. If we regard each car as a node on a big sensor network [4] [5] [6], by sharing these information gathered by all the vehicles via the VANET [7] [8] [9], a map of the nearby situation can be easily formed [10] [11].

1.1 Motivation

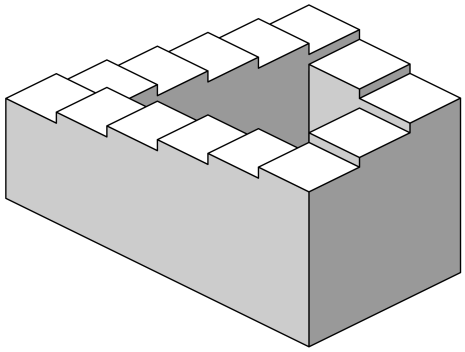
Our main object is to measure the distance from the cameras to a given object. Radar distance measurement system was used in this area for many years, but it cannot identify objects. Moreover, functions such as zooming in and out are not supported by Radars. On the other hand, 3D information is not easily extracted from cameras. Microsoft released Kinect in 2009 [12] as a motion control system which brought 3D detection into gaming system. Google has just started the Project Tango [13] by building a stereo vision cellphone which can build a depth information map of any indoor scene.

However, the distance between cameras decides the resolving power of the 3D world, which is why a human being can only distinguish the 3D information near us, but not the positional relationships of two faraway objects. Most stereo cameras have to be posifioned close to each other in order to control cost and maintain stability, which limits the ability to measure faraway objects.

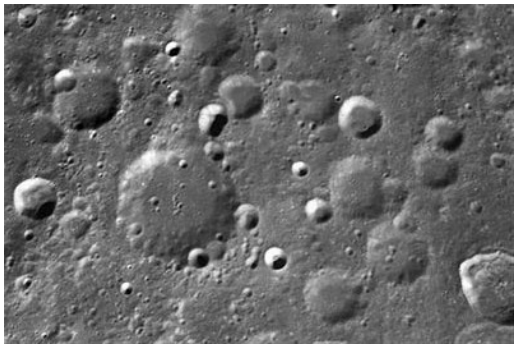
This system aims to provide an effective distance measuring system using computer vision technology, which is suitable for both long-range and wide-angle detection in outdoor scenes, and even in a vibrating, moving vehicle.



(a) From the view of the camera in the left camera, the two ladies appear to stand next to each other. However, the truth is there is one giant woman and a human lady.



(b) Penrose stair, a visual paradox about the illusion of depth.



(c) Moon surface, which makes your brain think that the craters are hummocks, because the light source is coming from the bottom.

Figure 1.2: Optical illusions, these figures show that we cannot trust any depth estimation other than parallax.

1.2 Outline of the Thesis

This thesis is structured as follows.

In chapter one, a literature review will be given about the status quo of the distance measurement system including radar, LIDAR, and image processing techniques.

In chapter two, I will discuss the components of the system and briefly introduce the technique they use. The main process of the system will be mentioned.

In chapter three, the pedestrian detection method will be discussed. You will read about the introduction of the HOG feature descriptor and SVM classifier. The improvement of the detector based on binocular version is stated in this chapter.

In chapter four, different kinds of keypoint detectors will be talked about. The pinhole camera geometry and parallax principle, a calibration method based on Zhang's work, will be mentioned in this part. At the end of this chapter, I will discuss how to decide the camera position and the suitable angle of view while of various driving speeds.

In chapter five, I will show you some experiments to judge the accuracy and stability in the real world testing, results and its analysis.

Chapter six contains the conclusion and discusses future works.

Chapter 2

Related Work

Currently, on a global scale, road traffic accidents cause huge losses to national economies, to peoples' lives and to their property every year. According to the Center for Disease Control and Prevention, the cost incurred by automobile crash injuries was over 99 billion dollars [14]. Highway accidents are caused mainly by factors such as rapid increase in highway mileage, rising traffic rates, and inattentive drivers. Such accidents include vehicle collisions and cars hitting pedestrians or animals. According to WHO, the global burden of death caused by road injury was in 2010 at 1,328,536 [15]; of this amount, 31% of these deaths are car occupants and 22% are pedestrians [16]. Among the vaious causes, accurate judgement of distance by the driver is one of the most direct factors that leads to accidents.

Studies have shown that if the driver is alerted half a second before a potential collision, this warning can reduce rear-end accidents by 30%, road-related accidents by 50%, and head-on crashes by 60% [17]. Therefore, the study of a collision preventable sensing technology has become a hot topic in research institutes, universities and automotive industries in many countries, with the goal of reducing traffic accidents. Among this research, the study of ADAS with an anti-collision alarm function has attracted a great deal of attention.

2.1 Distance Measurement: Methods and Devices

Nowadays, major distance measurement systems include the following: ultrasonic ranging, millimeter-wave Radar, LIDAR, computer vision and infra-red ranging. A comparison between these methods is shown in Table 2.1.

Table 2.1: Comparison of distance measurement on vehicle.

Methods	All day	Range	Units	Obstructions	Cost	Recognizing Objects	Applications
URD	Yes	Very short	Meters	Doppler effect	Low	No	[18]
Radar	Yes	Long	Meters	Electromagnetic interference and Multipath effect	High	Via motion tracking	[19], [20] and [21]
LIDAR	Yes	Long	Meters	Between devices and atmospheric interference	High	Via motion tracking	[22], [23] and Google Self-Driving Car
Stereo Vision	Can be	Short	Pixels	Backlight condition, fog, snow, and rain	Low	Via appearance	[24] and [25]

2.1.1 Ultrasonic ranging

Ultrasonic ranging normally uses a mechanical wave with a 20 kHz or higher frequency. The ultrasonic ranging device (URD) is a combination of a transmitter, a receiver and a single processing device [26].

As it is a kind of sonic wave, the ultrasonic wave also has physical properties like other sound waves. A transmitter in the URD repeatedly transmits a pulses sequence and gives another short pulse to the processor at the same time. When the receiver in the URD catches the reflection of the ultrasonic wave, it sends another pulse to the

processor. A bistable circuit will then transform the two pulse into a square wave, and the width of the square wave is the time gap between the two pulses. When measuring the width of this square wave, we can obtain the distance of the object, (see Figure 2.1).

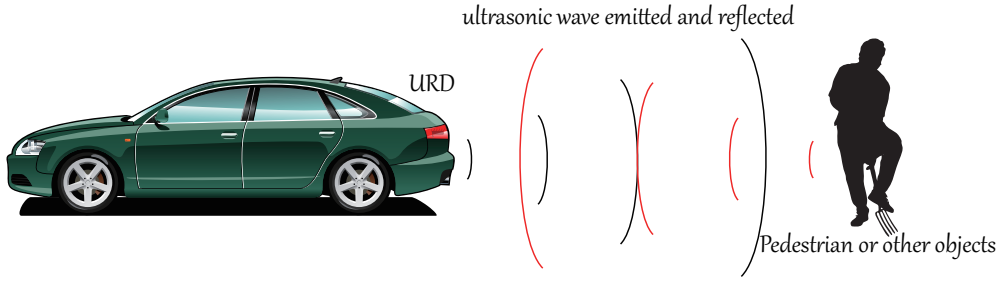


Figure 2.1: Operational principle of URD. The URD can emit an ultrasonic wave then receive the echo.

The ultrasonic wave has the following advantages:

- Good penetration through rain, snow and fog.
- Simple in principle, low in cost.

Its weaknesses are obvious however;

- The speed of sound waves is relatively slow such that it cannot be used in high-speed vehicle condition because of the Doppler effect.
- The directional characteristics of ultrasonic wave are worse than those of the laser beam; also, the angle of divergence is bigger.

Due to the above characteristics of the URD, it is now only used in the area of reverse radar and automatic parking systems [18]. It can detect objects behind a vehicle or send out alerts if a person bursts into the parking area. This system is widely used and has provided great help to drivers.

2.1.2 Millimeter wave radar

Radar Ranging is used to obtain a reflection of electromagnetic waves in order to measure the location of an obstacle, as Figure 2.2 shows. A millimeter wave is an electromagnetic wave with a wavelength of under 1 cm, and with a frequency between 30 GHz to 300 GHz [27]. For an on-vehicle application, there are two bands predominantly used: 24 GHz and 76-81 GHz. In these two bands, 24 GHz is limited by the transmission power, so 76-81 GHz is more frequently used current applications. Since different countries have different regulations, 76-77 GHz is the most widely used band. Examples of these systems are the 76 GHz millimeter wave radar by Fujitsu Ten [19] and 77 GHz millimeter wave radars from Bosch, Conti and Denso [20]. There are also some 3D scan radars, such as a 3D-Scan millimeter-Wave radar for automotive applications by Fujitsu Ten [21].

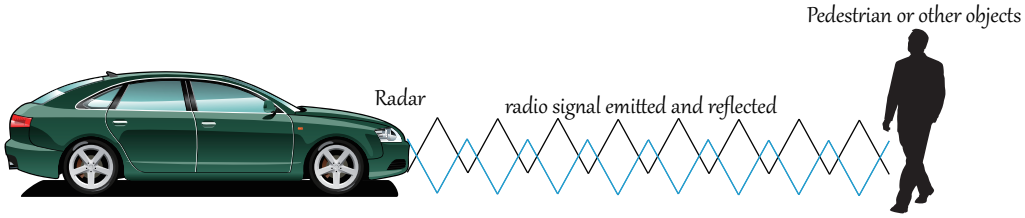


Figure 2.2: Operational principle of radar. The radar emits a millimeter wave and catches the echo.

Millimeter wave radar ranging mainly includes continuous wave radar ranging and pulse radar ranging.

Pulse radar is simple in principle but complex in practice. The voltage-controlled oscillator of a radar instantaneously changes from low frequency high frequency to shoot a high power pulse signal; thus, pulse radar is high in price and complex in structure. Besides, since the echo signal usually uses a 60 to 100 dB attenuation compared to the transmitter signal, the echo signal should be strictly isolated with the transmitter signal. This should be done before the echo signal is amplified. If this is not the case, the

amplifier of the echo signal will become saturated, because the transmitter signal has been disturbed. This will therefore lead to an increased complexity in the hardware structure and a higher construction cost. For these reasons, pulse radar is rarely used in on-vehicle distance measurement.

Continuous wave radar follows the principle of Frequency Modulated Continuous Wave (FM-CW) Ranging. This radar ranging approach structure is relatively simple. The basic principle is that the radar antenna transmits a continuous FM signal (usually a continuous triangle wave), receives the echo signals with a delay through the radar antenna, and than uses the frequency mixing processing function with the transmitter and the echo signal. The radar uses a microprocessor to calculate distance through the mixed signal.

Millimeter radar has a high frequency. It can minimize the angle of the beam amplitude of the electro-magnetic wave. It thereby reduces the interference and malfunction caused by the unwanted reflections.

On-vehicle millimeter radar is superior due to its detection stability, which cannot be disturbed by the shape, color and texture of the surface of the target. Interference is low during rain, snow and fog.

However, the millimeter radar detection system requires prevention against electromagnetic interference from other communication facilities and this makes the cost of the millimeter radar detection system higher than other devices [28]. The electromagnetic interference between radar equipments is also a weak point of radar.

2.1.3 LIDAR

A LIDAR (Laser Imaging, Detection and Ranging [29]) system emanates UV light, catches the reflection and uses the time of delay to measure distance, (see Figure 2.3). Nowadays, the types of LIDAR used by intelligent vehicle systems are divided into non-imaging laser radar and imaging laser radar [30].

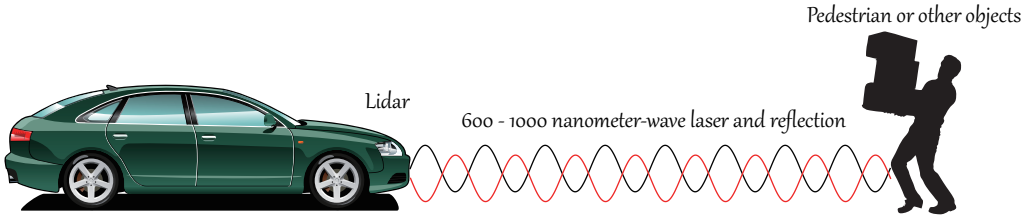


Figure 2.3: Operational principle of LIDAR. The LIDAR emits light and receives the reflection.

The non-imaging laser radar measures the distance according to the time of propagation of the laser beam. A laser pulse emitted by a high-power narrow pulse laser transmitter goes through the transmitting objective lens and focuses on a formed beam. A scan mirror then reflects the laser outside the device. The echo signal from an object in front goes through the receiving objective lens and returns an optical fiber to a photodiode. The distance to the object is measured, according to the difference in time domain between the enabling pulse of the laser diode and the receiving pulse of the photodiode.

The imaging laser radar can be divided into scanning imaging laser radar and the non-scanning imaging laser radar. Scanning LIDAR combines the laser radar with an optical scanner. It does so by using the scanner to control the emission direction of the laser. Scanning LIDAR scans the field of view point by point, gathering 3D information in the space. Non-scanning LIDAR uses a laser beam splitter to split the emitted laser into multiple beams aimed in various directions. Since non-scanning LIDAR has fewer detecting points, the speed of 3D recovery is higher.

The transmitter of LIDAR periodically transmits a series of high-frequency narrow pulses into space. If there is a target on the path of the electromagnetic wave propagation, the receiver will detect echoes reflected from the target. As the echo signals come and go between the LIDAR and the target, there will be a round trip time delay behind the transmitted pulse(see Figure 2.4).

Since the electromagnetic wave propagates at the speed of light in the air, the distance

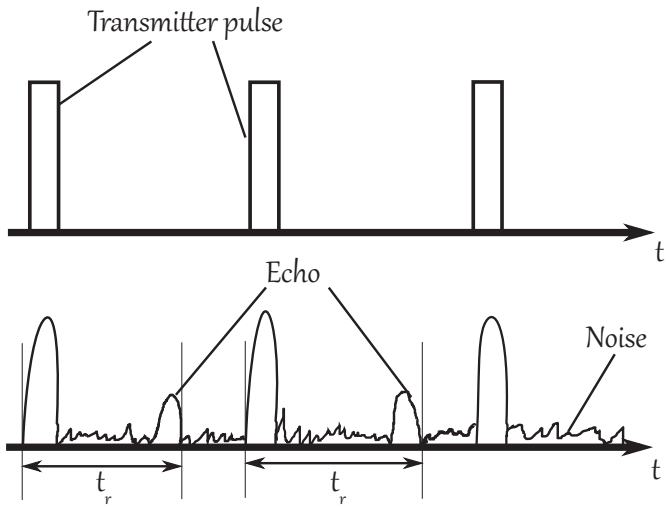


Figure 2.4: Laser ranging. LIDAR uses the time delay between the transmitter pulse and echo to measure distance.

of the laser is equal to the light speed times the amount of delay:

$$2D = c \cdot t_r$$

Where D is the distance between the object and LIDAR, t_r is the time lag between the echo and the transmitter pulse, and $c \approx 3.0 \times 10^8$m/s. This may be understood as the following equation:

$$D = \frac{c \cdot t_r}{2}$$

The accuracy and resolution of LIDAR is related to the transmission signal bandwidth. The narrower the pulse is, the better the performance becomes.

Since LIDAR is a radar system adopting a laser to function as the carrier waveform, the wavelength is much shorter than that of a microwave and a millimeter wave. So, it has the following benefit features::

- Works around the clock, without limitations on lighting conditions.
- Small divergence angle of laser beam, energy concentration, and better resolution and sensitivity.

- Large Doppler shift which can detect the target at low to high speeds.
- No interference from radio waves.
- Invulnerable to multipath effect.

There are also some disadvantages of LIDAR:

- Laser is largely impacted by atmospheric and meteorological conditions.
- Disturbance between devices with a near light modulation frequency.

With its own unique advantages, LIDAR has been widely used in remote sensing of the atmosphere, oceans, and land and for other goals, such as collision avoidance systems on vehicles.

2.1.4 Computer stereo vision

Computer stereo vision is a kind of bionics system based on the computational theory of human stereo vision; it was proposed by Marr and Poggio in the middle of the 1960s [31].The vision-based system has become a complete system starting from image acquisition to the reconstruction of the scene. Binocular vision is becoming an increasingly important branch of the computer vision area. CCD camera has the following good characteristics: small size, light weight, low power consumption, low noise, a high dynamic range and accurate measurement of light. Moreover, its optoelectronic signal output from line scan is conducive to subsequent signal processing; it thus has been widely applied in the automotive industry. The most striking point of computer vision is that an ADAS can only provide a response to a particular type of target (see Figure 2.5).

The basic principle of stereo vision is to observe the same scene from multiple viewpoints (usually two); this enables digital images in the three-dimensional scene. To be obtained, by using epipolar geometric principles, the three-dimensional shape and position of the surrounding scene is rebuilt.

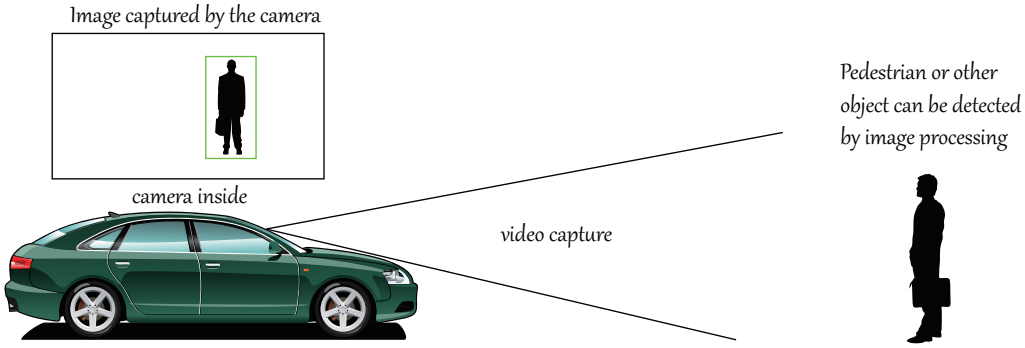


Figure 2.5: Principle of stereo vision. Cameras capture video frames and process with a GPU on board.

Stereo vision methods can be divided in two categories: passive and active.

For active stereo vision, the 3D information of a scene is reconstructed from a stereo camera with an auxiliary structured light source. In other words, active stereo vision approaches are similar to 3D scanning.

In [32], a projector and a single camera form a system, in which the camera obtains pixel-accurate correspondence from the structured light. Their system does not require the calibration information of the light sources, and maps out the disparity between all pairs of cameras and projectors. In [33], band-pass white noise patterns are introduced to reduce specular reflection. In [34], Kerstein has presented an approach for acquiring 3D shapes with weak textures on the surface using multiple camera pairs and an auxiliary laser pattern projector.

For passive stereo vision approaches, only an unstructured light source is used; thus, the system has to use only camera pairs to recover the 3D information.

3D scene information recovery is the main objective of basic stereoscopic research. To achieve this goal, a complete stereo vision system is typically comprised of six modules: image acquisition, camera calibration, feature extraction, stereo matching, 3D reconstruction, and post-processing. These modules are explained as follows:

1. Image Acquisition

Digital images are the resource of the stereo information. The most common method of acquiring digital images is to use two cameras at two different observation points; also some systems use several cameras.

2. Camera Calibration

   Binocular stereo vision system camera calibration entails mapping the coordinate position of an object in a 3D scene in the camera image plane, and its world-space coordinates. It is a basic and crucial step in stereoscopic three-dimensional model reconstruction.

   Currently, nearly all research on camera calibration is based on the work of Zhang [35]. This author proposed a flexible technique to easily calibrate a camera, which only requires images of a planar chessboard with a few different orientations observed by the camera.

3. Feature Extraction

   The purpose of feature extraction is to obtain image features that re required to be matched. The properties of the image feature and image matching method selection are closely linked. Currently, there is no established theory of universal access to image features; this has resulted in the diversification of feature matching methods in the stereoscopic vision research area .

   Generally, larger scale image features contain more information within a smaller number of features, and so, matching efficiency in this context is high. However, the process of feature extraction and description is harder and positioning accuracy is poorer. On the other hand, for smaller-scale image features, the description and expression of these features is relatively simple and there exists a greater positioning accuracy; however, due to a larger number of features and less contained information, more stringent constraints and matching strategies are needed during matching. This is required to minimize ambiguity and to improve matching

efficiency.

4. Image Matching

   Image matching, the most important and difficult problem in stereo vision, has been the focus of this study. When a spatial three-dimensional scene is converted into a two-dimensional image through a perspective projection, the images of the same scene will occur in varying degrees of distortion because the cameras are positioned at different viewpoints. Moreover, the impact of scene lighting conditions, geometry and surface properties (of the measured object), noise and distortion, and camera parameters, in addition to many other factors, are embodied in one single grayscale image. Clearly, an exact match of images which contain so many negative factors is not easy, especially in the case of an object with little surface texture.

5. 3D Reconstruction

   After completing the stereo camera calibration and image matching work, three-dimensional information for the measured object surface point will be recovered. The main factors affecting measurement accuracy are: camera calibration error, the quantization effects of the CCD imaging device, feature extraction and matching accuracy.

6. Post-processing

   For the purpose of distance measurement, post-processing is as simple as generating a disparity map using 3D information. However, in order to restore the complete information of the visible surface, interpolation and texture mapping is needed.

   During the interpolation process, the most important issue is to ensure that the protection of the visible surface information is continuous; therefore, interpolation must meet the principle of surface compatibility. Texture mapping refers to the function of mapping the texture onto the surface of the three-dimensional recon-

structed geometric model, making the visual effects of the geometric model appear real and natural in order to achieve higher visual requirements.

For depth information gathering, or for disparity map generation, Hoff and Ahuja proposed a method to integrate the processes of feature matching, surface interpolation, and contour detection; this is to ensure the smoothness of detected surfaces [36].

Lhuillier and Quan proposed a quasi-dense matching algorithm which is based on the match propagation principle [37]. Their method produces a quasi-dense disparity map that starts with a set of sparse seeds matched and propagated to neighbouring pixels. This algorithm has a robustness for initial spare match outliers due to the best-first tactics.

Banz et al. presented an algorithm on FPGA using non-parametric rank transform with semi-global matching (SGM) to estimate the disparity map [38]. SGM is one of the top performing matching methods in stereo vision; and SGM has proven robust under difficult imaging conditions. Their algorithm meets real-time requirements with a 64 pixel disparity range.

Kim and Park proposed an algorithm which is effective at deciding the correct correspondence and at increasing the accuracy of stereo matching [39]. In this algorithm, the length and color information of link features was used in stereo images; also, the average time computation is about 18.7 ms every frame, which is suitable for a real-time application. In [40], the authors introduced a workflow for organ surfaces, reconstructed from stereo endoscopic images.

This paper [41] describes a human detection algorithm using depth information given by Kinect. This detection algorithm does not use HOG nor does it extract points of interest features for detection; but, it uses a 3-D head surface model, showing good accuracy for indoor and shoot distance scenarios.

In [42], they present a solution using stereo vision to detect pedestrians and to measure their corresponding distances. However, this algorithm does not show accurate precision

in the presented paper.

CV distance measurement is low in cost and flexible with angle and width dimensions. However, its weak point is obvious: this method can only work with a visible target. In a situation like snow, fog, or a raindrop before the camera, the system will show a significant error. Also, near the entrance and exit of the tunnel, the tunnel in front of the vehicle may also cause a false positive result because of the overlay. When the sun is near the horizon ahead, the camera sometimes does not recognize the vehicle right in front of it.

The speed of processing is also a bottleneck of CV distance measurement. However, with the improvement of on-board CPU and GPU, this will not be a problem any more.

2.1.5 Infrared ranging

Infrared Ranging is based on the same principle as pulse radar or LIDAR which measures the distance according to the round trip delay time of the light between the target and the transmitter.

There is no proliferation during infrared light spread, and the refractive index is small when crossing other substances. Thus infrared ranging is widely used in short distance detection. Also, since any object can emit infrared at any time, according to the intensity and wavelength of the signal, the type of object can be detected at the same time.

Because the human eye cannot detect infrared, it is easily concealed; thus, this distance measurement device is widely used in military vehicles.

In my opinion, a combination of radar and computer vision will be the best solution for on vehicle distance measurement.

2.2 Distance Measurement System on Vehicle

The research on vehicle distance measuring devices was started in the 1960s, although the first concept on the automatic vehicle control system was given in 1940 by Williams Allison R. This research was first carried out in some western countries such as Germany and the United States, as well as Japan; it can be divided into two periods of time.

The first period is from the 1960s to the late 1970s. This stage is characterized by low levels of microwave theory and device integration, high hardware costs and a lack of criteria; thus, the devices developed then cannot be run efficiently. During that period, ultrasonic devices [43] and, separately, transmitters with receivers on different vehicles [44] were tested.

The study concerning the on-vehicle millimeter-wave radar was began in the 1970s [45]. A typical representative is the ASR100 millimeter-wave radar, developed by German Company ADC; which uses pulse ranging. Automotive collision avoidance systems developed by Benz, Nissan, and Ford were mostly made to use this radar.

The electronically scanning millimeter-wave radar developed by Toyota, Denso and Mitsubishi used a frequency modulated continuous wave ranging mode with a compact structure and good anti-jamming capability [45]. They are also the first to use advanced technology vehicle phased array radar in the world. Compared with mechanical scanning radar, the antenna of phased array radar does not need to rotate. It is more flexible with beam scanning and it has an excellent performance for target recognition.

In 1961, after the world's first ruby laser appeared, laser radar began to be used for measuring distance. After a series of engineering studies and trials, the army was finally equipped with laser radars.

The second research period covers the time from the 1980s until now. With the rapid development of microwave technology theory and device integration technology, it is possible to make the collision avoidance radar at a low cost and a high performance. Honda patented an automotive radar monitor system in 1982; it describe a front monitor

radar detecting the relative velocity between the object ahead and itself [46]. A consensus was also formed on the performance requirements for the collision avoidance radar system. After the beginning of the 1990s, Germany took the leading position in this field.

Researchers began to develop LIDAR in the 1990s. In 1993, LIDAR was first registered with a patent [47]. Currently, there is a trend suggesting that LIDAR will replace YAG laser radar in long-range radar applications. Over the recent years, a kind of portable, eye-safe and cheap LIDAR ranger, used for families was developed. It can be used not only as a telescope but as a ranger as well.

For example, in 1996, an American company named Bushnell launched a small cheap LIDAR ranger, Yardage400, with a 400-yards ranging capacity. A ranger with an 800-yards ranging capacity was produced in 1998. Also in 1998, Tasco produced a 800-yards LIDAR ranger with a camera function. In 2000, a Canadian company named NEWCON produced a laser ranger by the name of LRM1000 with a ranging capacity of 1000 yards. Now, the ranger provided by Bushnell has a measuring range of 15-1300 meters, and the deviation can be controlled in the range of 50 cm. It is reported by the USA that a high frequency laser radar, using the principle of pulsed laser ranging, has achieved a 10 KHz frequency measurement; and, it can reach a range capacity of 2000 m and a range precision of 50 mm. Also, it can clearly form a distance phase of a building or some other thing. Research shows that LD laser ranging technology has a trend towards precision and miniaturization, while remaining eye-safe.

In 1992, Science and Engineering Services Inc. (SESI) developed micro pulse lidar (MPL). Based on this, LIDARs have been developed such as the differential absorption laser radar which is used in atmosphere environment monitoring, and the Doppler laser radar [48], which is used for range and speed measurements in car applications. This marks the fact that the laser radar has entered a practical and commercial stage. Some related laser crystal materials, key technologies and system technologies have made great strides, and show promise in military and national economic areas.

In 1994, Benz proposed a method for distance control between moving motor vehicles [49]. In this method an automatic speed control and distance detection system was mounted on the rear of the vehicle.

In 1996, Micro Pulse LIDAR achieved commercialization [50]. It overcame several previous shortcomings of the traditional laser radar: low reliability, destabilization, large bulk, complex structure, eye hazard and high cost. MPL can obtain a relatively long-distance effect and a high resolution at low power. The low energy short-pulse laser is safe for our eyes. A high repetition rate can help us get more information. MPL also solves the problem of miniaturization and reliability. All these features have promoted the practical and commercial development of laser radars.

Benz designed and released a speed-distance control system in 1997 [51]. This kind of radar system uses the FMCW system. It works with a frequency of 76GHz, 3mW transmission power, and has the effective range of 150 meters. This system can also follow 30 targets simultaneously, while it can only be used with targets whose speed is slower or equal to the vehicle itself. The echo signal from the target is transferred to the vehicle control system through the processor, in this case, the control system could control the distance between each vehicle via auto-brake or a change in speed.

Toyota introduced the Adaptive Cruise Control (ACC) system in May 1998. This system uses a laser beam or a radar to measure the distance and relative speed between two cars. If there is a car inserted into the lane in front, and the distance is smaller than the set minimum distance, the vehicle will auto-brake at the rate of $3.5rmm/s^2$. This action will last until the realistic distance meet the requirements. If the front vehicle accelerates or leaves the lane, the system will allow for acceleration until the speed reaches the appropriate setting.

In September 1999, Jaguar began to apply ACC system to the XKR series sedan and convertible in Germany and the UK.

In September 2000, Mercedes-Benz and Lexus began to use ACC system. Lexus

installed the ACC system in its top vehicle LS430, and Mercedes-Benz installed the system for its C series and S series vehicles in Europe. Now, a Distronic (DTR) Radar Control System is installed in many Benz vehicles. This is an updated smart ACC system composed of range radar, a DTR monitoring computer, an indicator light etc. The radar sensor can measure the distance between the front object and the headstock. If this distance is within the range of risk, the indicator light will turn on. If the vehicle is too close to the object, a warning horn will ring to alert the driver. In the meantime, the DTR computer will connect with an engine computer, a transmission computer, an ABS, and an ESP brake system. These computers will connect with each other through the CAN-BAS net to limit engine output rotation speed, to modify the brake force shift gearbox and to control the speed of the vehicle.

After this development, GMC, Ford, OPEL, SAAB and VOLVO have gradually installed the ACC system. The anti-collision system has entered into its period of maturity; in the meantime, the competition has became fierce.

In 2007, the famous Swedish brand VOLVO introduced a LIDAR device produced by Laseroptronix to develop the anti-collision system for its vehicles. The frequency of the radar is 200 Hz, and its maximum measuring range is 800 meters. However, this system can only avoid 50% of accidents because of the disadvantages of the fixed azimuth angle inherent in LIDAR.

In 2008, the Velodyne LIDAR company in California produced HDL-64E LIDAR [22]. This system has 64 lasers that provide a 360 degree field of view. It has a wide range of applications, such as intelligent vehicles, automatic driving, unmanned control, intelligent control, 3D mapping, Unmanned Aerial Vehicle (UAV), city mapping, surveying and mapping, robot, security monitoring, city modeling, ect. However, the systems use an array of 64 detectors, results in a complexity of the control circuit, low reliability and high cost; thus, its popularization is limited.

In the same year, Fuji Heavy Industries released Subaru Legacy LSI. This is installed

with a driving assistant system EyeSight, comprised of a camera unit and an imaging processor [24]. EyeSight system uses a stereo camera instead of a millimeter wave radar or a laser radar for the distance measuring between vehicles and pedestrians to prevent collisions and to achieve 0-100km/h cruise control. When the vehicle is likely to collide with an obstacles ahead, this system will use an alert horn and an alert light to warn drivers; it will then brake automatically.

In 2010, German company IBEO published a four-beam laser scanning radar; named ALASCA XT four-beam LIDAR [23]. This LIDAR has been installed in Volkswagens, BMWs and other vehicles, and has undergone development in the areas of anti-collision, autopilot and other research. Through the public patent analysis of the product, it is understood to be essentially a linear scanning radar.

Recently, Bosch has launched a new stereo camera [25]. This camera will be used for driving assistance systems in the future. In addition, it will enhance the detection capability for objects ahead. Bosch also developed another new feature based on this capability, which enabled the vehicle to perform auto-making evasive actions when it detected the object ahead.

Gerhard Steiger, President of the Bosch Chassis Systems, said: "*stereo technology opens up new potential for video-based safety systems.*[52]" Because of binocular cameras, Bosch's new device can calculate the distance between cars and other objects according to signals received in the moment via video. In addition, new features will be combined with existing auxiliary functions, such as the ACC and the automatic emergency braking system. Because the stereo camera can also be integrated in the automatic emergency braking system, if an accident has become unavoidable, the speed will be reduced to a minimum. Coupled with the airbags and seat belt prevention system, the overall impact of the accident can be greatl reduced. According to media reports, this stereo camera is the smallest on-vehicle camera so far.

In the current luxury car market, most cars are installed with a vehicle distance

measuring device which can be used to track objects in the vicinity. Up to now, NVIDIA processors can be found in a rapidly growing number of more than 4.5 million cars [53]. This supports the GPU accelerated computer vision for ADAS of vehicles. GPU acceleration can boost the speed of both detection [54] and feature extraction [55], [56]; it can also boost the speed of distance measurement via computer vision by a large degree[57].

Chapter 3

Components of the System

3.1 Problem Statement

Our on-board system aims to detect particular targets and to measure the distance between the vehicle and the target. When switching the detection system, the measurement component can be adapted to evaluate the distance to any detectable target. For example, when targeting a car, the system can be used as an adaptable cruise control system that follows can track a specific vehicle. When targeting a pedestrian (or animal), the system can be used as a pedestrian (or animal) collision avoidance system.

Since our architecture is an on-board system, the critical range of detection should be based on the velocity of the vehicle. For this purpose, some companies have tried to use 3 pairs of radars with different angles to cover all the areas including long distance and short-wide areas. However, for an image processing system, the use of several cameras with different focal lengths is not the best choice, as it means that the limited computing resource will be dispensed in small pairs. There is no need to detect long ranges when the vehicle is moving slowly; conversely, there is also no need to detect wide areas when it is moving quickly.

In our opinion, there are some rules that an on-board object detection system must

follow:

- If the car will hit an object when following its current trajectory, the object must be detected in time for the car to stop. If the object is a pedestrian, this means he is in danger; if the object is a large animal, this means the driver and the passengers are in danger.

- If an object in danger is out of the range of detection (too near or too far), it should have already been detected if the object is too near, or it will be detected later if the object is too far away.

- If a detected object cannot be affected by the car, to protect the safety of the driver and passengers, the system will provide a warning but will not take control.

For the above rules, the object detection system must be able to measure the distance to the object and must be able to predict a collision through continuous image acquisition.

The object detection system will be installed on a moving vehicle; we consider that the focal length, which determines the angle of view of the camera, must change according to the velocity of the vehicle. When the velocity of the vehicle is slow, the angle must be wider; when the car speeds up, the angle will be narrower, in order to get a longer range of view.

As a result, the system should be equipped with a zoom lens stereo camera. This would introduce some new concerns to the traditional stereo vision measurement, which uses camera calibration to correct distortion and to obtain depth information. One concern is that when the focal length changes its linear information, the distortion matrix will not follow a specific rule; the lack of distortion will make object detection much more difficult. The other concern is that even though the distortion matrix can be given as a database in the program, accurate synchronicity between the two camera will be very difficult to achieve. Therefore, the main idea of this system is to avoid the usage of undistorted information and the depth information.

3.2 System Architecture

In general, this binocular-based on-board object detection and distance measurement system requires the three following stages: detector, matcher, and calculator. The role of the detector stage is to locate the object in the image. After that, the matcher stage establishes a correspondence between the object (e.g., pedestrian) in the left and right images; and finally, the calculator stage provides the results of the distance measurement. Each following stage will strongly rely on the robustness of the previous stage, so a method must be implemented that aims to enhance stability.

Since it is easier to conduct the experiment with pedestrians than with uncontrolled animals, this system will be experimented with a pedestrian detection system.

3.2.1 Detector

For pedestrian detection, the most common method of deriving the feature descriptor is HOG (histograms of oriented gradients), and the corresponding classifier is SVM (support vector machine) [58]. There are also some studies discussing the use of HOG with Adaboost; although this method is faster than HOG+SVM, the detection rate is not dependable for real world design [59]. Since GPU acceleration shows a great improvement in computer vision performance [60], CUDA based openCV is used to build the HOG + SVM detector.

For large animal detection, some research was performed with a dual-stage detection system [61]. LBP adopting the AdaBoost algorithm was used as the first stage, which supplied the second stage with a set of Region of Interests (ROI) containing all the true and false positive results. The second stage was based on an adapted version of HOG + SVM classifiers to reject the false positives. Some research presented HOG + SVM detection based on ROI determined by inferred detection using a thermographic camera [62].

When using computer vision in vehicle detection, some research presents a multi-orientation detector using HOG features of SVM classification [63]. On the other hand, some research uses Haar-like features and AdaBoost classification with an active learning method [64].

Some other methods use Radar or Lidar and spatial segmentation, or motion, to recognize specific objects. These methods are not considered in this paper, since this system is focused on image processing methods.

For a situation in which the detection component will sometimes give a false negative result, tracking method is used applied. This kind of mistake cannot be easily eliminated by improving the detection method itself. It can only be eliminated by adding a tracking method.

Two widely used tracking methods are the particle filter [65] and the Kalman filter [66]. The Kalman filter is used in this paper because it has the ability to predict the movement of a pedestrian when the detector misses the target. The other reason for the use of the Kalman filter is that there will always be a new target appearing in the field of view. When the old target vanishes, the detector has to scan every frame, and the particle filter becomes redundant for the detection system.

Since there are two cameras in this system, the crossover control of the tracker must be taken into account.

3.2.2 Matcher

Keypoint matching information is used in this system to calculate parallax. The disparity map does not calculates this as it would for the common stereo camera, because this detection system is set on a car; it must tolerate the imprecise stereo calibration information caused by shaking cameras.

For keypoint feature descriptors, SIFT [2], [67], SURF [3] and ORB [56] are tested for their feasibility in this task. This is the same as for the detector, CUDA based GPU

computing, which will be tested for acceleration instead of CPU computing.

3.2.3 Calculator

For the distance measurement, parallax is used to calculate the distance of the two images of the system. Before the calculation, a calibration is completed according to Zhang's work [35].

There are two stages of this system that will give results with outliers. One is keypoint descriptor, and the other is the calculation results.

For the first stage, the outliers are non-linear; RANCAS [68] and LMedS [69] are commonly used in this area, especially RANCAS, which is designed for computer vision areas. The results of calculation are produced one by one in the real-time system, so the estimation method should implicate a tracker rather then other estimation methods. Another Kalman filter tracker is applied to estimate the measurement results for outlier elimination.

3.3 Process of the Program

The architecture of our system, used for detection and distance measurement, is explained as follow (see Figure 3.1). We highlight that the Kalman filter tracker is not included in Figure 3.1.

1. Read HOG+SVM detector information and initialize keypoint detector.
2. Initialize class Object; this class consists of the left and right location of the object in the image, and the Kalman filter tracker.
3. Capture left and right images.
4. Detect object with HOG+SVM at both images.

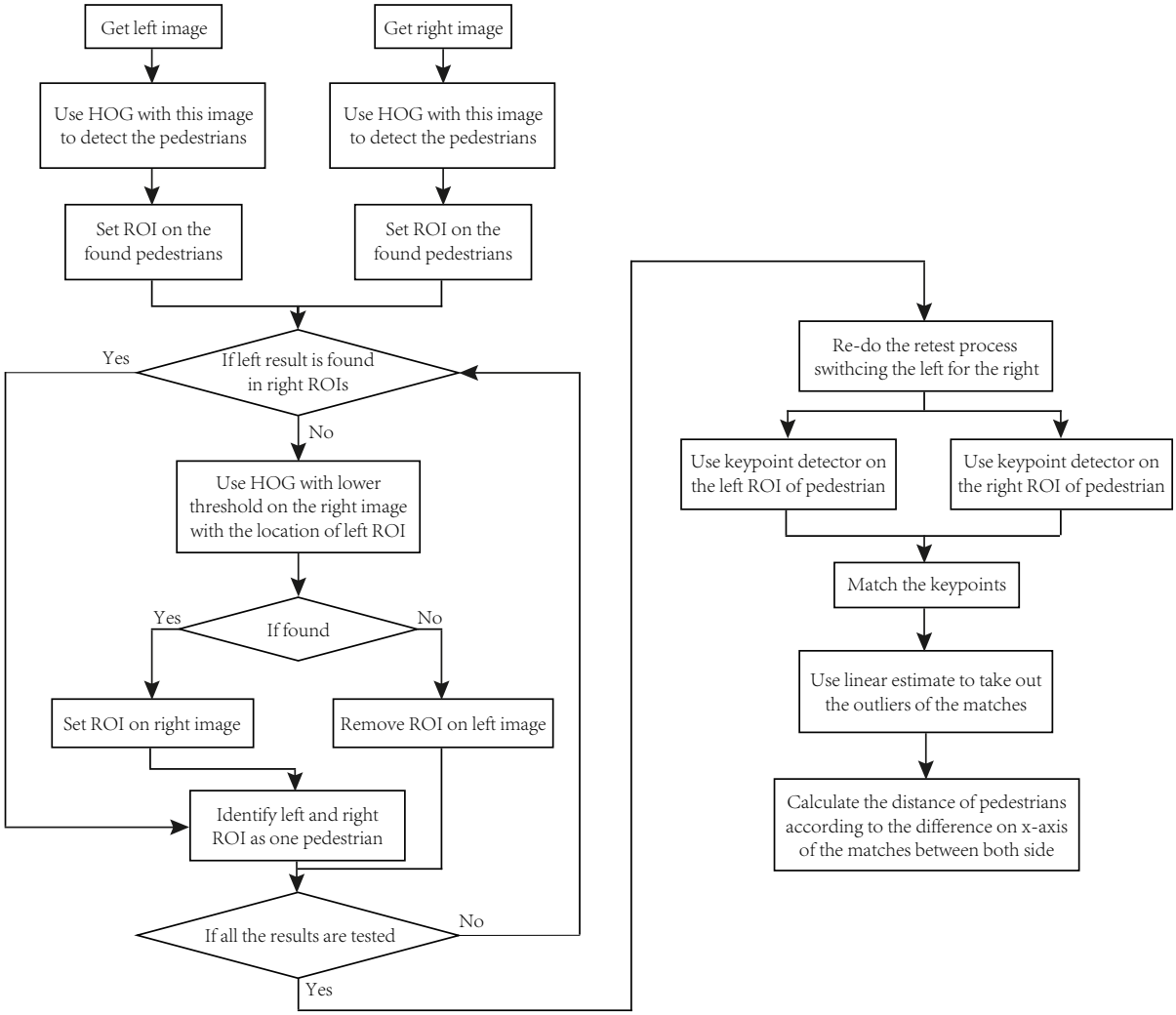


Figure 3.1: Detection and distance measurement process

5. Check all the detection results to find the correspondence between the results, and mark all the related results as one object in two images.

6. Re-detect all the results which have not found a pair, while judging whether they can be marked as an object.

7. Check the existing Object Vector and the results found in this frame for a new detection result. Verify which one has been updated and which one has not been found in this frame (if one object is missing in 8 continuous frames then it is erased from the Object Vector).

8. Use the Kalman filter for all the elements in the Object Vector to predict the movement of the object.

9. For all the objects not found in this frame but that exist in the Object Vector, use the prediction result to update the Object Vector.

10. For all the elements in the Object Vector, use the keypoint detector in the left and right images.

11. Match all the keypoints and use linear estimation on the matches.

12. Calculate the distance using parallax method.

13. Use the Kalman filter to track the distance change of the object and to estimate the result.

14. Repeat from step 3 to step 13 for the next frame.

3.4 Platform for System Design and Experiment

To implement our system, we use two identical cameras, each of which is equipped with a variable focal length lens connected to a central unit (laptop). The two cameras are

mounted at a specific distance from one another on a two-headed tripod head plate (see Figure 3.2). The distance between the cameras on the tripod head can be set from 40 mm up to 220 mm.

Figure 3.2: Cameras on board for video capture.

3.4.1 Hardware

The Point Gray Flea®3 camera FL3-U3-13S2C-CS has the following features:

- Sony IMX035 CMOS, 1/3", 3.63 μm
- Rolling Shutter
- 1328×1048 at 120 FPS

The cameras are equipped with 5.8 mm - 58 mm zoom lenses. In our future work, the lenses will zoom in and out according to the velocity of the vehicle; however, this is not necessary for this situation. With the CCD size of 1/3", the maximum horizontal angle of the camera view is 44.96°, and the minimum view is 4.74°. In fact, the frame size has

a significant impact on the total speed of the system. For real-time speed, the image is resized to 640×480 after it has been captured by the cameras.

The laptop has the following specifications:

- Inter® Core™ i7-3610QM CPU @ 2.30GHz
- nVIDIA GeForce GTX 675M 2048MB 384 cores
- 12GB RAM 1600MHz

Since GPU acceleration is used for the detector and keypoint descriptor, the performance of GPU is greater than that of the CPU.

3.4.2 Interface

This system is designed for moving vehicles. Thus, the camera should have a fast frame sampling rate and the two cameras should be highly synchronized. For better real-time performance, the data transmission from cameras to laptop should not result in the bottleneck of this system. Therefore, USB 3.0 is used in this project instead of IEEE 1394 or USB 2.0, which are more widely used interfaces. As time has elapsed over a period of project development, USB 3.0 has become more widely used as a PC standard layout.

3.4.3 Software

OpenCV is an open source computer version library written in C and C++, which can be developed on C, C++, Java, Python, Matlab and other languages in Windows, Linux and Mac [70]. OpenCV was designed for computational efficiency and focuses primarily on real-time applications. It is written in optimized C/C++ and can take advantage of multi-core processing not only on CPU but also on GPU. All methods used in this paper are written with OpenCV 2.4.6 on the platform of Visual Studio 2010.

Chapter 4

Binocular-based Object Detection and Tracking

In this chapter, the detection and tracking system is designed to detect pedestrians, and will be applied to the distance measurement to be presented in the following chapter.

4.1 Detection using Single Camera

Pedestrian and animal detection is a key consideration in the field of driving assistant systems. The most common method used the well-known HOG-SVM detector.

4.1.1 Histogram of oriented gradients

The histogram of oriented gradients (HOG) is proposed by Dalal and Triggs who focused on the problem of pedestrian detection; their algorithm had an excellent performance compared with other feature sets [71]. The advantage of the HOG descriptor is that it can describe contour and edge features outstanding in objects other than human beings, such as bicycles, and different types of cars.

The main concept behind HOG is that the appearance and shape of the detecting

target can be described by the light intensity gradient or the edge direction distribution.

HOG gradient computation

Before the computation of the gradient values, gamma/colour normalization is applied to improve the performance [71]. After that, the gradient direction and gradient magnitude of each pixel is computed. The horizontal and vertical gradients obtained by convolution acts as a gradient operators. According to the test by the author, the best mask is a 1-D derivative $[-1, 0, 1]$ which is better than the uncentred $[-1, 1]$ or cubic-corrected $[-1, 8, 0, -8, 1]$. This means that the gradient of pixel (x, y) is:

$$G_x(x, y) = H(x + 1, y) - H(x - 1, y) \tag{4.1}$$

$$G_y(x, y) = H(x, y + 1) - H(x, y - 1) \tag{4.2}$$

where, $G_x(x, y)$ and $G_y(x, y)$ are the horizontal gradient and vertical gradient of point (x, y) respectively, $H(x, y)$ is the pixel gray value. Then, the gradient magnitude $G(x, y)$ and gradient direction $\alpha(x, y)$ can be obtained as:

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \tag{4.3}$$

$$\alpha(x, y) = \tan^{-1}[\frac{G_y(x, y)}{G_x(x, y)}] \tag{4.4}$$

Cell histograms and HOG feature vector generation

The next step after gradient computation is spatial/orientation binning. Since according to the author, the gradient is a vector, the histogram channels are spread over $0° \sim 180°$ if the gradient is "unsigned", or $0° \sim 360°$ if the gradient is "signed". Normally, increasing the orientation bin would improve the performance. However, some tests show that the optimal bins number is 9. The image window is divided into several larger spatial regions called "blocks", and each block contains several small spatial regions

called "cells". The pixel's gradient magnitude is contributed as the vote weight. We then obtain a 9-dimensional feature vector for each cell.

Before the binning, the gradient intensity needs to be normalized to make the feature vector space robust to local illumination changes:

$$V = \frac{v}{\sqrt{\| v \|_2^2 + e^2}} \tag{4.5}$$

where v is the non-normalized vector containing all histograms and e is a small constant. Note that the maximum value of v is limited at 0.2 to improve the performance [67].

After the HOG feature vectors are extracted, a classification algorithm would be applied to detect the target. The SVM algorithm was specifically introduced by Dalal.

4.1.2 Support vector machines

Support vector machines (SVM), also called support vector networks, is a famous and powerful pattern recognition classification algorithm improved by Vapnik [72] in 1995. SVM is a standard application in image classification and recognition, as well as data analysis, regression analysis, pattern recognition, etc. The most common application of SVM is pedestrian detection, proposed by Navneet Dalal and Bill Triggs [71]; they adopted locally normalized HOG descriptors and applied SVM as a baseline classifier throughout the study to build an excellent pedestrian detector.

Algorithm explanation

SVM algorithm is explained by a general optimal separating hyperplane problem. In the linear separable case, the batabase of training examples can be divided by a hyperplane:

$$w \cdot x + b = 0 \tag{4.6}$$

where, w is a n dimensional vector, b is the offset.

The optimal hyperplane meets the requirement that the margin of the hyperplane is maximized for each type of data. The margin indicates the distance from the nearest

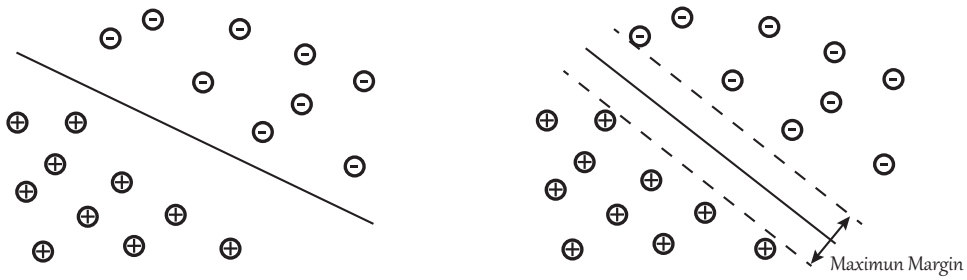


Figure 4.1: An example of optimal segmentation in a 2D plane. Both lines separate the data in the plane perfectly. However, the image on the right has the largest margin and it has the optimal segmentation. The optimal separating hyperplane follows the same rule.

vector to the hyperplane, (see Figure 4.1). In other words, to find the optimal hyperplane is to solve this optimization problem:

$$\min \Phi(w) = \frac{1}{2}||w||^2 \tag{4.7}$$

satisfying the constraints: $y_i(w \cdot x_i + b) \geq 1 \qquad \forall i.$

Solving Equation 4.7, we get the final classification function:

$$f(x) = \sum_{1=n}^{n} y_i \alpha_i (x \cdot x_i) + b' \tag{4.8}$$

where, $\alpha = (\alpha_1 \cdots \alpha_n)$ is Lagrange multiplier, b' is the offset of the optimal hyperplane. The sign of f is used to determine the classification of x.

In many other cases, the training data is not separable in a linear manner. In this condition, it is not feasible to find the optimal separating hyperplane. The general method is to map the dataset X into a high dimensional feature space H, and use the function of the original space to achieve the inner product. Thus, using an appropriate inner product function on the optimal hyperplane, we can achieve this linear non-separable classification.

The objective function at this time is:

$$\max w(\alpha) = \sum_{i=1}^{n} -\frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{4.9}$$

where $K(x, y)$ is the inner product kernel function.

Correspondingly, the final classification function is:

$$f(x) = \sum_{1=n}^{n} y_i \alpha_i K(x, x_i) + b \tag{4.10}$$

4.1.3 HOG-SVM classifier

Object detection based on SVM algorithm in image processing is a non-separable model. Figure 4.2 indicates the overview of HOG features extraction and target detection process adopting the classifier of SVM.

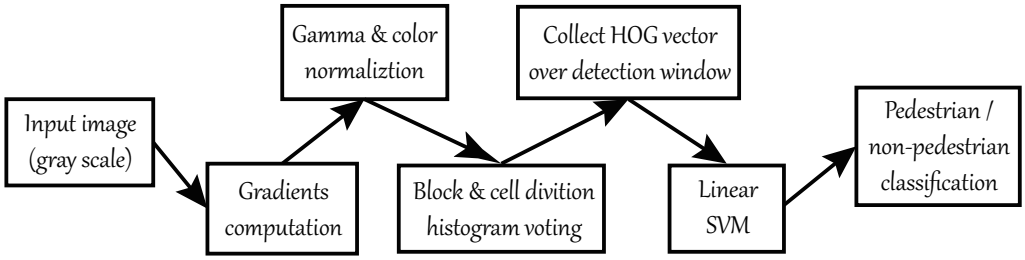


Figure 4.2: Process of HOG features extraction and object detection adopting SVM classifier. Redrawn from [73]

.

A type of cell and block-based HOG feature is extracted for describing each image sample; this descriptor is a high dimensional vector ($3,780$ dimensions in 64×128 pedestrian window). In other words, the SVM training algorithm is aimed at finding the classification vector w and the classifier threshold b with the minimum value in Equation 4.10.

In fact, HOG features do not have rotation-invariance nor scale-invariance. In practice, the detecting image will scale to adapt to different target sizes.

4.2 Crossover Detection Between Two Cameras

Since we are using the result of HOG+SVM detection, which yields a miss rate of approximately 0.1 at a 10^{-4} false positive rate [73], the most notable problem is consistency of results. In other words, the system should successfully detect one pedestrian in both of the images or miss the detection at the same target to ensure the distance measurement part can run. Thus, a re-detection process based on a binocular system needs to be added.

We know that a problem arises when one camera successfully detects the pedestrian but the other one does not. We also acknowledge that when the threshold increases, the detection rate will decrease. Therefore, the process is to follow this rule:

- For every pedestrian found in the left image, we create a Region Of Interest (ROI) on the right image based on the size and position of the pedestrian in the left image. The rule is that we enlarge the size of ROI from the center, then we shift the ROI according to the height of the target obtained from the left image. (Line 7-20)

- When the ROI of the right-image on the right is created, we search exhaustively within the image to determine whether there is a presumed pedestrian obtained in this ROI. (Line 8-19).

- If there is only one pedestrian inside the aforementioned ROI, we consider him to be the same pedestrian detected in the image on the left. This pedestrian is then removed from the searching candidates. If more than one candidate meets the conditions above (Line 3-14), we pick the best match according to the size and coordinates of the presumed pedestrians. (Line 9-12)

- If no pedestrian appears within the ROI of the right-hand image, a new HOG-SVM will go through this ROI with a lower threshold. When the threshold gets lowers, the false positive rate increases and the miss rate decreases. If there is a

pedestrian, a lower threshold will locate this person more easily. However, it is worth nothing that if the result on the left is a false (positive mostly caused by the detector mistaking something from a specific angle of view), even a lower-threshold SVM cannot detect that from another angle. (Line 14)

  - If the detection yields a positive result, the new results on the right and left will be marked as one pedestrian. If more than one result is found, we use the information of the ROI itself. (Line 15-17)
  - If the detection yields a negative result, we will consider this result of the left image as a false positive and discard it from the result set.

- After the search for all presumed pedestrians within the left-hand image, the detection process will start again for all the results in the right-hand image which are in need of finding their pairs from the left-hand image. However, at this moment, a lower-threshold SVM will be applied to the ROI immediately because all the left results have been previously searched before, thus there cannot be any available searching candidates remaining in the left image. (Line 21-28)

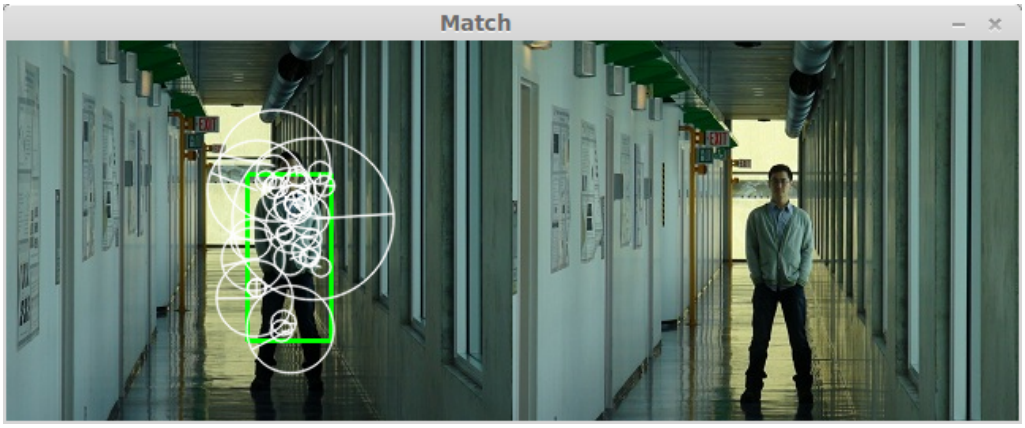
After the second detection process shown in Algorithm 1, all the independent results from isolated detection become consistent, as shown in Figure 4.3.

Subsequently, all the independent results from the first detection become correspondence, and this supplement solves the issue of discrepancy in detection results caused by the uncertainty of HOG+SVM, as seen in Figure 4.3.

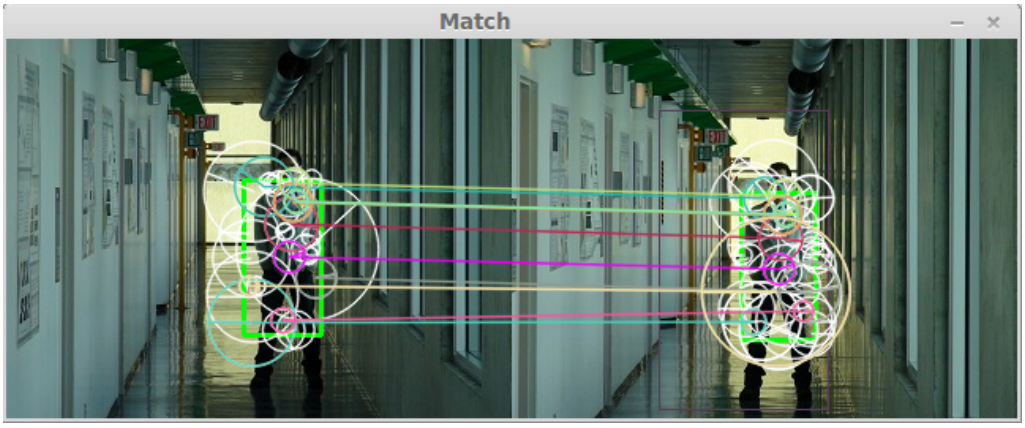
After this method is completed, the robustness of the system is increased. The testing result shows that the rate of false positives are decreased with the overall positive rate. Within the tracking and distance measuring system, fewer false positives mean higher speeds and better real-time performance. Although the false negatives are increased a minimal amount, in the continued frames and with the tracking method, this shortcoming influences the outcome less than the benefit does.

```
1  Input: Left and right isolated detection results
2  Output: Paired results of pedestrian detection
3  L_i: the ith of left results;
4  R_j: the jth of right results;
5  L.size(): the total number of presumed pedestrians of left;
6  R.size(): the total number of presumed pedestrians of right
7  for i = 1 to L.size() do
8      for j = 1 to R.size() do
9          if L_i and R_j is the same pedestrian then
10             Mark L_i and R_j as one pedestrian in different images.
11             Next ith loop.
12         end
13         if j = R.size() then
14             Create a new ROI on the right image based on the size and position of L_i, re-detect
               the ROI using 0.5 lower threshold of SVM than the isolated detection.
15             if new result found then
16                 Mark the new result and L_i as the same pedestrian in different images.
17             end
18         end
19     end
20 end
21 for j = 1 to R.size() do
22     if R_j not marked with an L_i then
23         Create a new ROI on the left image based on the size and position of R_j, repeat
           detection of the ROI using 0.5 lower threshold of SVM than the isolated detection.
24         if new result found then
25             Mark the new result and R_j as the same pedestrian in different image.
26         end
27     end
28 end
```

Algorithm 1: Algorithm of crossover re-detection.



(a) The detection result on isolated cameras



(b) Result after crossover re-detection

Figure 4.3: The process of system with re-detection supplements the issue of low accuracy of HOG+SVM, so that the keypoint can find its pair in the other image.

4.3 Object Tracking

In this system, since the false positive rate is designed to be as low as possible, the negative impact of this is the increased miss rate. For the purpose of retaining the positive results, a tracker must be designed to give a result of the location of pedestrian in the image even though the detector has missed the target in one frame. In addition, the distance measurement results must transition smoothly, so the tracker will follow one pedestrian and distinguish him from the others in order to continuously monitor the distance measurement results.

Kalman filter is therefore used to continuously record, track and forecast the position of the pedestrian. When a pedestrian is detected, a Kalman filter is created for tracking his position and size both in the left and right image. If he is not detected in any frame, the prediction of his position will be used as the pedestrian position in this frame. Since the pedestrian has a high likelihood of vanishing from the view of the camera, the rule is if one pedestrian is missing in 8 continuous frames, the tracker will be released, having assumed that the pedestrian disappeared from sight or that was a false positive for saving memory.

4.3.1 Kalman filter

The Kalman filter algorithm consists of a two-step process: the prediction step and the update step. In the prediction step, at time 1, the current state variables are estimated by the Kalman filter, along with the uncertainties of the current state variables. Once the outcome of the measurement at time 2 is observed, the update step begins to work. These estimates at time 1 are updated using a weighted average. Higher estimates of certainty are given more weight. Then the prediction step will again give the prediction again for the state variable at time 3, followed by the update step. When the two-step process begins to iterate, the Kalman filter is tracking and predicting function is initiated.

Because of the algorithm's inherent recursive ability, it can run in real-time only by using the current time measurements as input, as well as the previously calculated state and its uncertainty matrix. The algorithms of Kalman filter is described as follows.

The true state in the model of Kalman filter is assumed that at time k is evolved from the state at $(k-1)$ according to

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_{k-1} + \mathbf{w}_k \tag{4.11}$$

where

- $\mathbf{F}_k$ is the state transition model which is applied to the previous state $\mathbf{x}_{k1}$;
- $\mathbf{B}_k$ is the control-input model which is applied to the control vector $\mathbf{u}_k$;
- $\mathbf{w}_k$ is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance Q_k.

To Predict the state estimate at time $k+1$ from a given state at time k, we will use the following:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}_{k+1} \hat{\mathbf{x}}_{k|k} + \mathbf{B}_{k+1} \mathbf{u}_k \tag{4.12}$$

where

- $\hat{\mathbf{x}}_{n|m}$ is the estimate of $\mathbf{x}$ at time n given observations up to, and including at time m.

4.3.2 Kalman filter tracker for binocular crossover control

In the case of pedestrian tracking in two images, the Kalman filter tracker should track the position and size in both the left and right images. Therefore, the measurement vector x of the tracker contains 8 elements, which are: x_l, y_l, $width_l$, $height_l$, x_r, y_r, $width_r$, $height_r$. Then the state vector x contains 16 elements, which includes the measurement

vector and its derivative. The state vector is:

$$\mathbf{x}_k = \Big[x_l, y_l, w_l, h_l, x_r, y_r, w_r, h_r, x'_l, y'_l, w'_l, h'_l, x'_r, y'_r, w'_r, h'_r,\Big]'$$

Since the abscissa of one pedestrian in the two sets of images should be associated, the variation of the location in the two sets of images should be a same value between frames. The transition matrix is set to be this:

$$\mathbf{F}_{initial} = \left(\begin{array}{c:cccccccc} & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hdashline 0 & \multicolumn{8}{c}{I} \end{array}\right)$$

In the matrix above, the element 1 out of the principal diagonal is reflecting the relationship between x'_l, x'_r and x_l, x_r. As the iteration goes, the abscissa of the pedestrian in the right image will remain consistent with the change of the abscissa in the left image.

We see that the results of tracking show a high synchronization between the left and right images having a decreased error rate compared with tracking the left and right position independently.

Chapter 5

Binocular Distance Measurement

In the field of computer vision, recovering 3D information from images is a long-standing problem. Many methods have been tested to solve the bionic problem. For a human being, we can easily estimate the distance of an object by sense of sight, because we have depth perception. To give the smart vehicle a sense of stereopsis, we need to use two cameras and link the images captured by them mimicking a human being's eyes. Unlike our eyes, it is impractical for the cameras on a vehicle to have the ability to move both eyes toward each other the converge at an object, which is how we receive the depth information. This is because the detector must search the whole image before giving the position of the object on image, which is different in process that with humans. However, we can still make use of binocular parallax, according to the principle of parallax Figure 5.1 [70], which has been used by the astronomers to measure distances of celestial objects for a long period of time.

There are two ways to generate the parallax of a object on stereo images: one way is to use depth information, and the other way is to calculate the abscissa of matched keypoints in different images. This requires a high synchronization rate and stable relative positional relationship of cameras, to generate a disparity map. A keypoint-based method can avoid these necessities, so we pick keypoint-based method for this system.

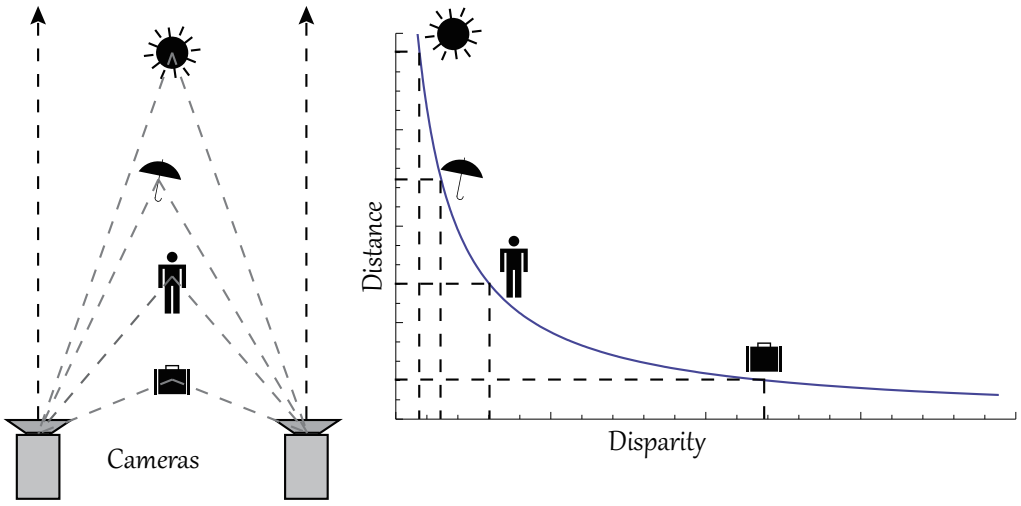


Figure 5.1: Parallax of two cameras

5.1 Keypoints and Match

There are many methods to extract keypoints from an image. The most common way is by detecting corner points. Many methods are used to extract corner points, for example: morphological filters, Harris and KLT, which can only get the corner points and need an extra method to extract the feature. There are also keypoint detectors with feature descriptors such as FAST, SIFT, SURF and ORB. A brief description of each method is given in the following sections.

5.1.1 SIFT: Scale Invariant Feature Transform

SIFT (Scale Invariant Feature Transform) was published by Lowe in 1999 [2] and enhanced by himself in 2004 [67]. It is an approach for detecting and extracting local feature descriptors that are reasonably invariant to changes in illumination, image noise, rotation, scaling, and small changes in viewpoint.

Interest points for SIFT features of the input image $I(x, y)$ correspond to a local extrema of difference-of-Gaussian filters $G(x, y, \sigma)$ at different scales. The scale space is

defined as $L(x, y, \sigma)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y, \sigma) \tag{5.1}$$

where $*$ is the convolution operation, and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{5.2}$$

and Lowe said we could choose $\sigma = 1.6$ in practice.

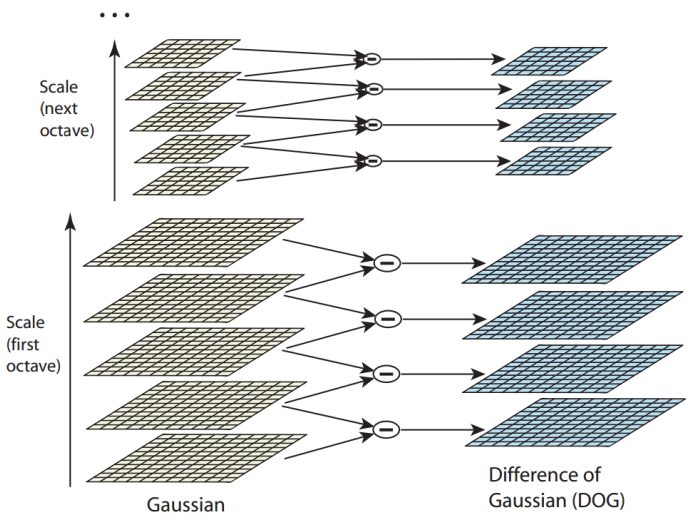


Figure 5.2: The difference-of-gaussian images. [2]

The result of convolving an image with a Difference-of-Gaussian (DoG) filter is computed from the difference of two nearby scales which is separated by a constant multiplicative factor k:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{5.3}$$

Figure 5.2 shows the set of scale space images produced by repeatedly convolving the initial image with Gaussians. The convolved images are grouped by octave, and the value of k is selected so that we obtain a fixed number of blurred image per octave. This also ensures that we obtain the same number of DoG images per octave.

The keypoints are identified as local maximum or minimum value points of the DoG images across scales. In order to detect these positions in $D(x, y, \sigma)$, SIFT compares each sample point to 26 neighbours in the space of the Gaussian pyramid. The 26 neighbours includes the 8 neighbours of the point in the same scale of image, and 18 neighbours above and below. One keypoint is determined only if it is larger or smaller than any other point in the neighbourhood, shown in Figure 5.3.

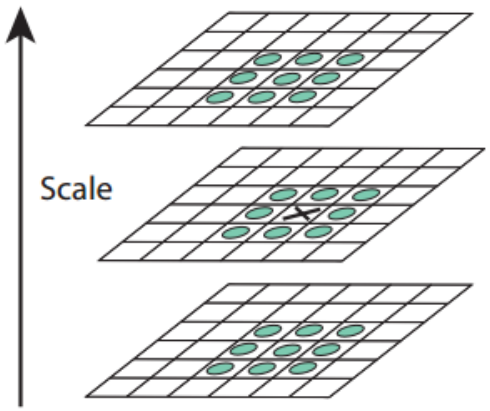


Figure 5.3: Each pixel (marked with X) in the DoG image is compared to its 26 neighbours in a 3×3 region including the same scale and two neighbouring scales. [2]

After comparing each pixel to its neighbours, the keypoint candidates are found. The next step is to perform the details including location, scale, and ratio of principal curvatures. First, unstable extrema with low contrast will be rejected by the function value at the extreme point. Second, as the DoG function will have a strong response along edges, candidates along edges are eliminated.

Then, every keypoint will be assigned an orientation, by computing a gradient orientation histogram in the neighbourhood of keypoints; then, peaks in the histogram will correspond to dominant orientations.

Once a keypoint orientation has been selected, the feature descriptor is computed as a set of orientation histograms in a 4×4 pixels neighbourhood. (see Figure 5.4) The

histograms contain 8 bins each, and this leads to a SIFT feature vector with 4×4×8 = 128 elements.

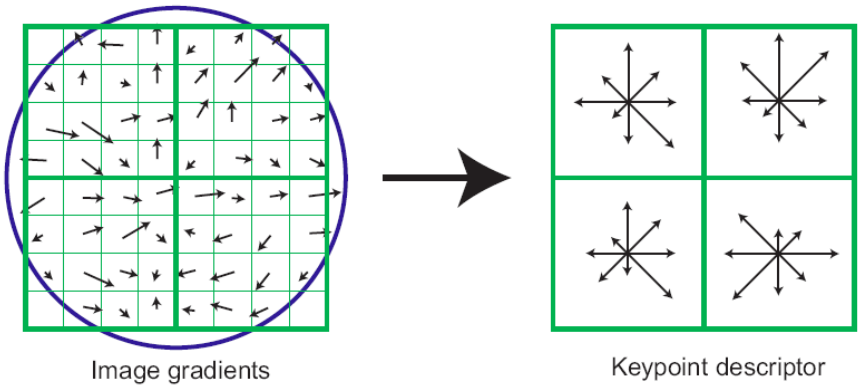


Figure 5.4: The keypoint descriptor. [2]

Using SIFT with the resulting ROI of the detection, we can obtain the keypoints of the pedestrian we found. These keypoints are accurate enough to calculate the parallax after the matching with the image of the other side of view.

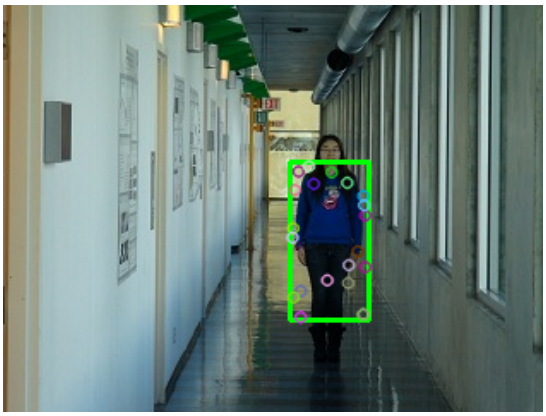


Figure 5.5: Keypoints on pedestrian that was found.

5.1.2 SURF: Speeded up robust features

SURF (Speeded Up Robust Features) by Herbert Bay is a SIFT-like feature descriptor but faster [3]. The SURF detector is based on the determinant of the Hessian matrix H.

$$H(f(x,y)) = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} \tag{5.4}$$

The determinant, which is the discriminant of the matrix, is calculated by [3]:

$$det(H) = \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - (\frac{\partial^2 f}{\partial x \partial y})^2 \tag{5.5}$$

The Hessian matrix $H(x,\sigma)$ in x at scale σ of a given point $x = (x,y)$ in an image I, is defined as follows:

$$H(x,\sigma) = \begin{pmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{pmatrix} \tag{5.6}$$

where $L_{xx} = \frac{\partial^2}{\partial x^2} g(\sigma)$, and similarly for $L_{xy}(x,\sigma)$ and $L_{yy}(x,\sigma)$.

At the step of pyramid establishing, SURF applies kernels of increasing size to the original image. That method allows multiple layers of the scale-space pyramid to be processed and avoids the need to sub-sample the image to increase performance and speed. In other words, SIFT changes the size of the image, but SURF scales the size of the filter.

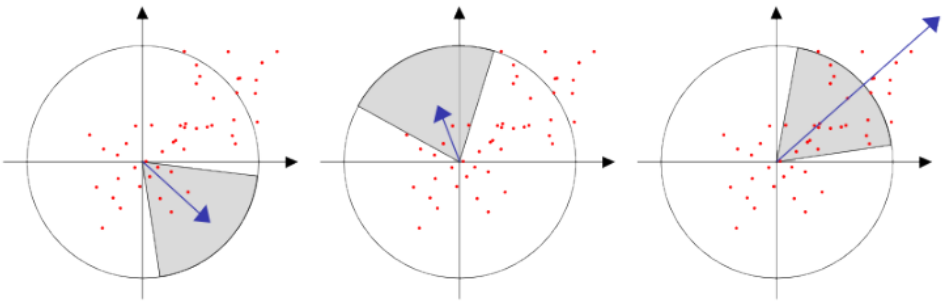


Figure 5.6: Orientation assignment. [3]

At the step of orientation assignment of SURF, Haar wavelet responses of size 4σ are used on set pixels within a radius of 6σ of the detected point, where σ refers to the scale at the point to be detected. Then weight a Gaussian centered at the interest point of the responses, and rotate a circle segment covering an angle of 60 degrees around the origin. The longest vector of the response yields the orientation of the keypoint.

5.1.3 ORB: Oriented FAST and rotated BRIEF

ORB (Oriented FAST and Rotated BRIEF) is a very fast binary descriptor based on BRIEF demonstrated by Ethan Rublee [56] in 2011, which is much faster than SURF and SIFT.

Although the SIFT keypoint detector and descriptor have proven useful, it is also too complicated to compute. In this case, most SIFT need GPU devices to speed up.

ORB is a more computationally-efficient replacement of SIFT that can improve image-matching capability without GPU acceleration. Every feature builds on the well-known FAST keypoint detector and BRIEF descriptor. This technology is very popular for its low cost and high performance.

ORB is the combination of oFAST keypoints and rBRIEF features. It has outstanding performance on data concentration, and a low rate of dropping incorrect points.

oFAST

FAST features are well known for their computational properties. It is better to add an efficiently computed orientation component.

FAST points are like the circular rings around the center. Usually, FAST-9 (circular radius of 9) performs well.

Because FAST cannot detect the corner data, a Harris corner method is needed to ensure the FAST produces it. In addition, FAST lacks the ability to produce multi-scale features, so FAST features (filtered by Harris) should be applied at every level if they

are used to measure a scale pyramid of image.

Intensity centroid is used as an effective and simple measure of corner orientation. If the corner's intensity is away from its center, the vector is used to show the orientation. The moments of a patch is [56]:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{5.7}$$

and the moments can find the centroid:

$$C = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}) \tag{5.8}$$

The vector from the corner's center O to the centroid,$\overrightarrow{OC}$. The orientation of the patch is:

$$\theta = \text{atan2}(m_{01}, m_{10}) \tag{5.9}$$

To determine that the rotation is inherent, we must make sure that the moments calculated by x and y is calculated with the circular region of radius r. When $|C|$ is close to zero, FAST corners break down.

rBRIEF

The BRIEF descriptor patches an image through a set of binary intensity tests. A binary test τ is defined by [56]:

$$\tau(p;x,y) := \begin{cases} 1 : p(x) < p(y) \\ 0 : p(x) \geq p(y) \end{cases} \tag{5.10}$$

where p is a smoothed image patch and $p(x)$ is the intensity of p. The vector of n binary tests is:

$$f_n(p) = \sum_{1 \leq i \geq n} 2^{i-1} \tau(p;x_i,y_i) \tag{5.11}$$

In order to inherit the BRIEF in-plane rotation, more degrees of in-plane rotation need to be removed. In this case, the BRIEF descriptor is better to compute for various

rotations and perspective warps of each patch. But this method is very expensive. A more effective way is to steer BRIEF relative to the orientation of keypoints. The feature of n binary tests at location (x_i, y_i):

$$S = \begin{pmatrix} x_1, \cdots, x_n \\ y_1 \cdots, y_n \end{pmatrix} \tag{5.12}$$

A steered version S_θ of S is originated from patch orientation θ and the related rotation matrix R_θ:

$$S_\theta = R_\theta S \tag{5.13}$$

And then the steered BRIEF operator is:

$$g_n(p, \theta) := f_n(p) | (x_i, y_i) \in S_\theta \tag{5.14}$$

As long as the keypoint orientation θ is across views, S_θ will always able to calculate its descriptor.

One advantage of BRIEF is that each feature has a large variance and a mean near 0.5. High variance makes a feature more discriminating and makes the tests uncorrelated. A good set of binary tests is needed to select and reduce the correlation among the binary tests. The method is as follows:

First, set up a large number of keypoints. Second, calculate all possible binary tests. The number of the tests is $\begin{pmatrix} N \\ 2 \end{pmatrix}$. Third, run each test against all patches and then form the vector T. If $T's$ correlation is greater than a threshold, discard it; otherwise add it.

Scalable Matching of Binary Features

ORB has a better performance than SIFT and SURF in nearest-neighbor matching over large databases of images. ORB uses the recovery of variance, which is more efficient during NN search.

Locality Sensitive Hashing (LSH) points are in several hash tables and buckets. In terms of binary features, the descriptors of the buckets in different tables have the same signature. rBRIEF descriptors help speed up LSH.

Since the binary features of rBRIEF are not correlated, the hash function performs well at classifying data.

After using rBRIEF, LSH speeds up dealing with keypoints in the hash maps. Meanwhile, rBRIEF LSH also enhances the accuracy.

5.2 Error Reduction

There are two steps to reduce errors: one takes place before detection, the other occurs after the detection. The ROI pretreatments are beneficial to both accuracy and saving the time.

Another problem is that there must be outliers in the keypoints' match results. A linear estimation method is therefore added to reduce error. But unlike the other situation, these outliers are not normally distributed, so that we cannot use a linear smoother such as least squares method to smooth the data. However, there are two linear estimation methods: RANSAC and LMedS.

5.2.1 Human shape mask

If we use the pedestrian detection results as the ROI directly, there must be several points that do not belong to the pedestrian; even though most background points will not find their pair after matching, they waste the time of the keypoint detection step. A simple way to solve this problem is to cut the region of the pedestrian detection result into a human-like shape as in Figure 5.7.

This shape is given by narrowing the top one fifth of the image by 50%, and narrowing the bottom quarter by 30%. From figure 5.8, we can see that a large amount of space

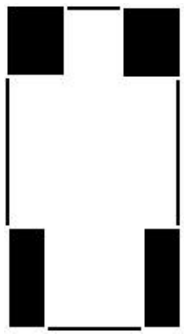


Figure 5.7: The keypoint detector will only detect the white area.

which not on the pedestrian can be eliminated, as it is not used to generate the keypoints.

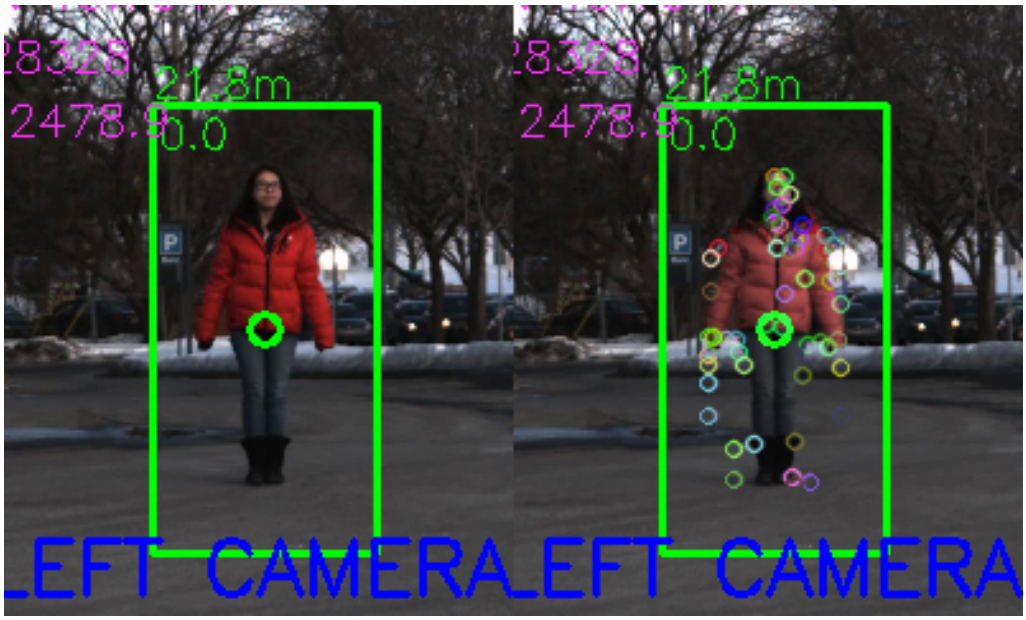


Figure 5.8: The detection result (right) and keypoints detection result (left)

There is one problem involved which is that when the pedestrian is walking, the keypoints on his feet will not be computed. However, if the ground under the pedestrian has keypoints and they can match, these keypoints should be in the same range as the pedestrian.

5.2.2 Random sample consensus

The RANdom SAmple Consensus (RANSAC) algorithm is a general parameter estimation approach proposed by Fischler and Bolles in 1981 [68], which is designed to deal with input data that has a large proportion of outliers in it. Its purpose is to make such data smoother, and can fit the model well. So it is ideally applied in automated image analysis where interpretation is based on the data provided by error-prone feature detectors.

Unlike other smoothing techniques, RANSAC aims to use the smallest data set possible to recover the landscape. The official definition in [68] is given as follows:

> *Given a model that requires a minimum of n data points to instantiate its free parameters, and a set of data points P such that the number of points in P is greater than n, randomly select a subset S1 of n data points from P and instantiate the model. Use the instantiated model M1 to determine the subset S1 of points in P that are with some error tolerance of M1. The set S1 is called the consensus set of S1.*
>
> *If S1 is greater than some threshold t, which is a function of the estimate of the number of gross errors in P, use S1 to compute a new model M1.*
>
> *If S1 is less than t, randomly select a new subset S2 and repeat that above process. If, after some predetermined number of trials, no consensus set with t or more members has been found, or terminate in failure.*

They are three components in this method: error tolerance for establishing model compatibility, the maximum number of attempts to find a consensus set, and the threshold t which is used to find whether the point is properly located in the model.

Error Tolerance for Establishing Model Compatibility

Error tolerance is useful for to determining the bonds of errors. It helps compute the model, and measure the implied errors.

Error tolerance may vary in different standards of a model. While the error tolerances are relatively small, one error tolerance is always sufficient.

The Maximum Number of Attempts to Find a Consensus Set

The number is based on the ideal number of trials k. W is the probability of the point in the error tolerance of the model. The equation is [68]:

$$E(k) = b + 2 \times (1-b) \times b + 3 \times (1-b)^2 \times b \cdots + i \times (1-b)^{i-1} \times b + \cdots \tag{5.15}$$

$$E(k) = b \times (1 + 2 \times a + 3 \times a^2 \cdots + i \times a^{i-1} + \cdots) \tag{5.16}$$

where E is the expected value of k, $b = w^n$, and $a = (1-b)$. An identity for the sum of a geometric series is:

$$a/(1-a) = a + a^2 + a^3 + \cdots + a^i + \cdots \tag{5.17}$$

Thus:

$$E(k) = 1/b = w^{-n}$$

The standard deviation of k, $SD(k)$, is:

$$SD(k) = sqrt[E(k^2) - E(k)^2]$$

$$E(k^2) = \Sigma(b \times i^2 \times a^{i-1}) = \Sigma(b \times i \times (i-1) \times a^{i-1}) + \Sigma(b \times i \times a^{i-1})$$

then using the geometric series identity:

$$2a/(1-a)^3 = \Sigma(i \times (i-1) \times a^{i-1})$$

$$E(k^2) = (2-b)/(b^2)$$

so,

$$SD(k) = [sqrt(1 - w^n)] \times (1/w^n) \tag{5.18}$$

After substituting several numbers in w, it is surprising that $E(k)$ is always approximately equal to $SD(k)$, which means the number of k needs to be tried several times. On the other hand, if we calculate k with z, which is the probability that at least one of our selected points is an error, we get the function that:

$$(1 - b)^k = (1 - Z) \tag{5.19}$$

so that,

$$k = log(1 - z)/log(1 - b) \tag{5.20}$$

Threshold t

The threshold t must meet the two requirements: from the correct model, a great enough number of consistent points should be found to smooth the final procedure. In this case, t must be large enough to satisfy these two conditions.

We assume that when y, which stands for the probability of a given data is within the tolerance of wrong model, is smaller than w, the probability of a given data is within the tolerance of correct model. Thus the wrong model won't be established.

5.2.3 Least median of square

Least Median of Square regression [69] is a kind of improved algorithm based on the method of least squares (LS).

$y_i = x_{i1}/theta_1 + \cdots + x_{ip}\theta_p + e_i$ is a classical linear model where e_i is considered as deviation. The regression aims to figure out θ by the data$(x_{i1}, \cdots, x_{ip}, y_i)$. The most popular method is [69]:

$$minimize_{\hat{\theta}} \sum_{i=1}^{n} r_i^2 \tag{5.21}$$

where $r_i = y_i - x_{i1}\hat{\theta}_1 - \cdots - x_{ip}\hat{\theta_P}$. It is called the method of least squares (LS).

Although it is simple, its accuracy is doubted by many researchers. In this case, Hampel (1971) proposed the notion of the breaking point ϵ^*. ϵ^* is the smallest percentage of outliers which lead to large error derivation. According to least squares, $\epsilon^* = 0$.

The first step of the regression estimator was from Edgeworth (1887). His *least absolute values* or L_i criterion is:

$$minimize \sum_{i=1}^{n} |r_i| \tag{5.22}$$

Although it ignored outlying y_i, it cannot deal with the wrong value of x_i which may make a large influence on the model.

The next step was the *M estimator* (Huber 1973), using $\rho(r_i)$ instead of r_i^2, where ρ is a symmetric function whose minimum happens at zero. The scale parameter is estimated by:

$$\sum_{i=1}^{n} \psi(r_i/\hat{\sigma})x_i = 0 \tag{5.23}$$

$$\sum_{i=1}^{n} \chi(r_i/\hat{\sigma})x_i = 0 \tag{5.24}$$

where ψ is the derivative of ρ and χ is a symmetric function. Substituting in minimax asymptotic variance arguments, we can obtain the function

$$\phi(u) = \min(k, \max(u, -k))$$

where k is the constant around 1.5, while $\epsilon^* = 0$ for the leverage points.
Because of the leverage points, Mallows (1975) introduced generalized *M estimators (GM estimators)* to bound the effect of outlying x_i

$$\sum_{i=1}^{n} w(x_i)\psi(r_i/\hat{\sigma})x_i = 0 \tag{5.25}$$

where Schweppe suggested using

$$\sum_{i=1}^{n} w(x_i)\psi(r_i/(w(x_i)\hat{\sigma}))x_i = 0 \tag{5.26}$$

influence functions. It proved that *GM estimators* had a breakdown point at most $1/(\rho+1)$, where ρ is the dimension of x_i.

There is a problem about the robust regression's range of application noted by Siegel (1982). He started that the repeated median with a 50% breakdown point is the best for larger amounts of contamination. The repeated median is defined as [69]:

$$\hat{\theta_j} = \underset{i_1}{\text{med}}(\cdots(\underset{i_{p-1}}{\text{med}}(\underset{i_p}{\text{med}}\theta_j(i_1,\cdots,i_p)))\cdots) \tag{5.27}$$

Then, the least median of squares (LMedS) is raised

$$\underset{\hat{\theta}}{minimize}\ \underset{i}{\text{med}}\ r_i^2 \tag{5.28}$$

This equation satisfies $\epsilon^* = 50\%$.

Properties of LMedS

The n observations $(x_i, y_i) = (x_{i1}, \cdots, x_{ip}, y_i)$ is a $p+1$ dimensional row vector. $\theta = (\theta_1, \cdots, \theta_p)^t$ is a p dimensional column vector. The linear model is $y_i = x_i\theta + e_i$, where the distribution of e_i is $N(0, \sigma)$. It is obvious that no zero is allowed in x vector because they are useless on θ. In addition, it is assumed that no vertical plane has more than $n/2$ observations. The vertical plane is a subspace of p-dimension containing $(0, \cdots, 0)$ and $o, \cdots, 0, 1)$.

When p is determined, it can ensure that a unique non-vertical plane contains the observations. In this case, equation 5.28 always has a solution, while LMedS has its breakdown properties. We assumed a regression estimator T and n data points(x_i, y_i). The n data points is the sample X. If $\beta(m; T, X) = ||T(X') - T(X)||$, where X' is corrupted sample and m is the number of arbitrary values instead of the original data points. Then the breakdown point of T is:

$$\epsilon^*(T, X) = \min\{m/n;\ \beta(m; T, X)\text{is infinite}\} \tag{5.29}$$

The equation 5.29 means that for least squares, $\epsilon^*(T, X) = 1/n$. One bad observation can result in a breakdown point. On the other hand, the least median of squares is in

the opposite condition. If $p > 1$, the number of breakdown points is $([n/2] - p + 2)/n$. When n trends to unlimited, the number of breakdown points is very small.

5.2.4 Result after error reduced

After that, we can obtain the relatively more accurate matches as shown in Figure 5.10. The difference of the keypoints abscissa values is also much more accurate than the original HOG detection result and the average of the original keypoints shown in figure 5.9.

Figure 5.9: Keypoints and their matches

5.3 Distance Calculation Using Binocular Geometry

Since we have obtained the parallax information, we can calculate the distance of a target to the camera through geometric methods.

5.3.1 Camera model

The pinhole model is the simplest camera model. Under ideal conditions, only a single ray of light from each point in the space can enter the camera through the pinhole.

Figure 5.10: Final Matches

Therefore, each point on the image corresponds to only one particular point in the real world.

In Figure 5.11, the image plane inside the camera is behind the pinhole. As we know, light travels in straight lines, so that the the object in the real world is centrosymmetric with its image in the camera. The camera aperture is located at the origin O of a 3D orthogonal coordinate system. The image plane is parallel with plane X_1OY_1. We assume that the center point C of the image plane is also the origin of the Cartesian coordinate system X_2CY_2 inside the camera. However ,if the origin of a digital image is on the top-left corner, the related coordinate is unchanged for translational displacements. The distance f between the two origins is referred as the focal length of the pinhole camera. A ray of light emanates from point P onto the image plane through the pinhole, formatting the projection which denoted p'. We denoted $P = (x, y, z)^{\mathrm{T}}$ for the in world coordinate system, and $P' = (x', y')^{\mathrm{T}}$ in the image coordinate system. At this time, one of the world coordinate z is lost, and we will talk about how to recover this information with two cameras later.

If we see Figure 5.11 from Y_1 axis, we get Figure 5.12. In Figure 5.12, we see two similar triangles. Using the similar triangulation formula, we get $x' = -\frac{fx}{z}$. Similarly,

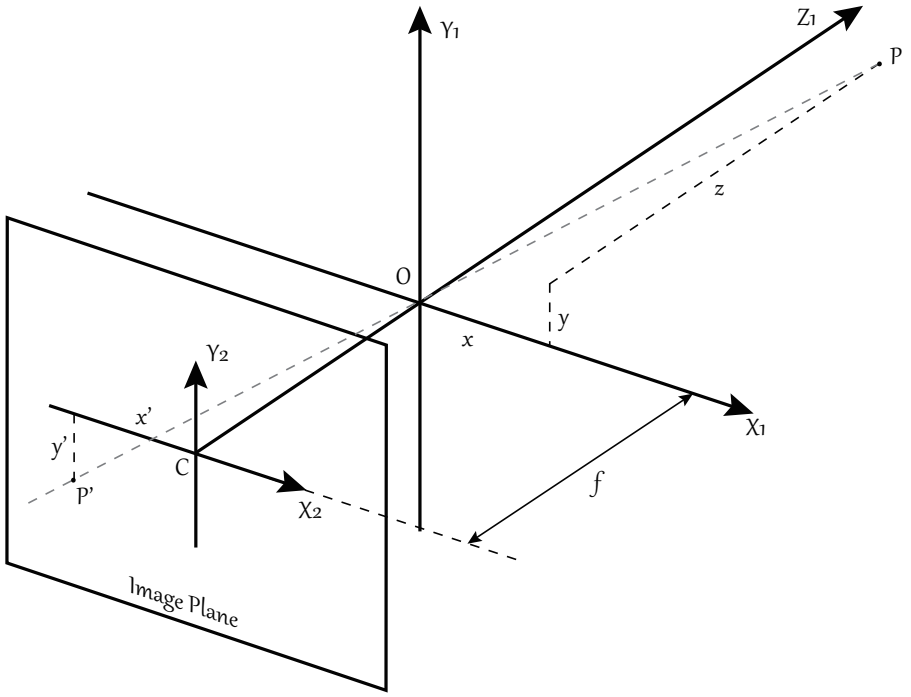


Figure 5.11: Pinhole camera geometry

looking in the direction of X_1 axis follows that $y' = -\frac{fy}{z}$. Overall,

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = -\frac{f}{z} \begin{pmatrix} x \\ y \end{pmatrix} \tag{5.30}$$

If we rotated the image plane 180°, which equates to assuming the image plane in front of the pinhole as seen in Figure 5.13, the resulting mapping from real world to the image coordinates is given by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \frac{f}{x} \begin{pmatrix} x \\ y \end{pmatrix} \tag{5.31}$$

In matrix form, we have:

$$\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \tag{5.32}$$

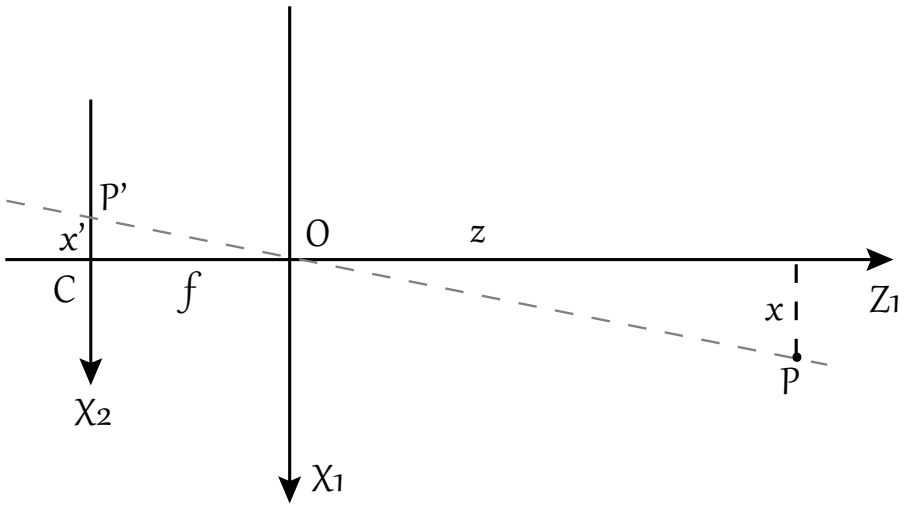


Figure 5.12: Pinhole camera geometry as seen along Y_1 axis

Equation 5.32 is obtained by the ideal pinhole camera model. Since the unit of an image is a pixel, we use two new parameters f_x and f_y which actually represent the parameters along x' and y' instead of the focal length f. Additionally, the optical axis of the camera cannot meet the image plane perfectly in the center. Usually, offsets c_x and c_y are incorporated to represent the origin of the coordinate system of the image plane away from the optical axis. To eliminate these errors, the equation is rewritten as follows:

$$P' = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} P \tag{5.33}$$

in which, P' represents the coordinate of a point on the image in the pixel unit; P represents the coordinate of a point in the real world.

f_x, f_y, c_x, c_y are called intrinsic parameters which are constant and unchanged with the view of the camera, and the matrix is called intrinsic matrix. They are once measured, applicable in future scenarios. Since these parameters are in the pixel unit, if the image is downsampled or upsampled, these parameters must be scaled with same proportion.

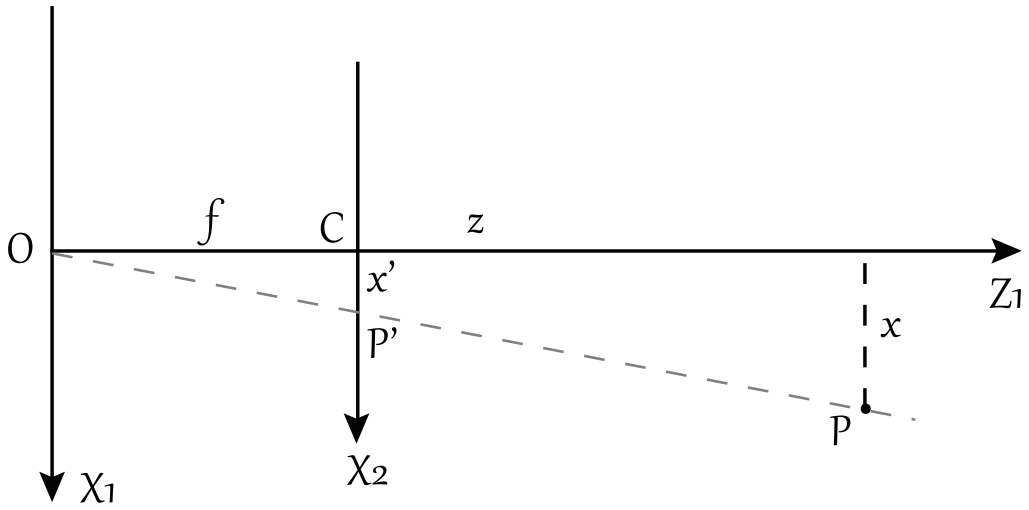


Figure 5.13: The rotated pinhole camera model.

Since the camera is fixed on a vehicle, there should be a rotation-translation matrix R given to fit the image coordinate system to the vehicle coordinate system. The Equation 5.33 can be rewritten as follows:

$$P' = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} P \tag{5.34}$$

The rotation-translation matrix is also known as the external parameters matrix, which describes the rigid motion of the camera. The introduction of this matrix contributes to the assembly error when the camera is set on the vehicle.

5.3.2 Calibration

To know the intrinsic parameters of a camera and the distance between the two cameras, we need to do a calibration. Although there are many way to calibrate a camera, there are two in particular that apply: OpenCV and Matlab Calibration ToolBox [74] based on [35]. Because these two methods have the same order of parameters, we can get the parameters by Matlab Calibration ToolBox, and use the parameters directly in OpenCV.

Calibration data is collected by taking several photos of a chessboard shown in Figure 5.14 with both cameras at the same time. As inputting the size of the blocks in the real world and marking the corner points, the Matlab Calibration ToolBox can output the parameters inside the camera such as the focal length and the distortion matrix.

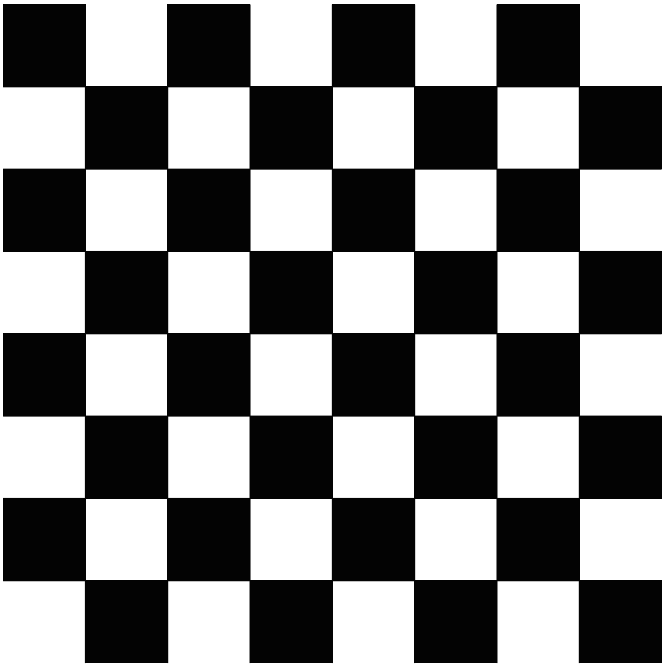


Figure 5.14: Chessboard for calibration

5.3.3 Distance calculation

From the keypoints and their matches, it is easy to find the average disparity, which is a sub-pixel value given by the abscissa of each match. Then, all the parameters to calculate the distance are obtain (see Figure 5.15). Where f is the focal length, Z is the distance that we want to calculate, x is the abscissa of the object in the image, T is the distance between the two cameras. We denote the disparity between the two views as $d = x_l - x_r$. In practice, d is given by the average of the abscissa difference of all pairs of matched keypoints we obtained previously. Since the distance of the object Z

is inversely proportional to the disparity, it follows that the relationship between these two terms should be:

$$\frac{T-(x_l-x_r)}{Z-f}=\frac{T}{Z} \tag{5.35}$$

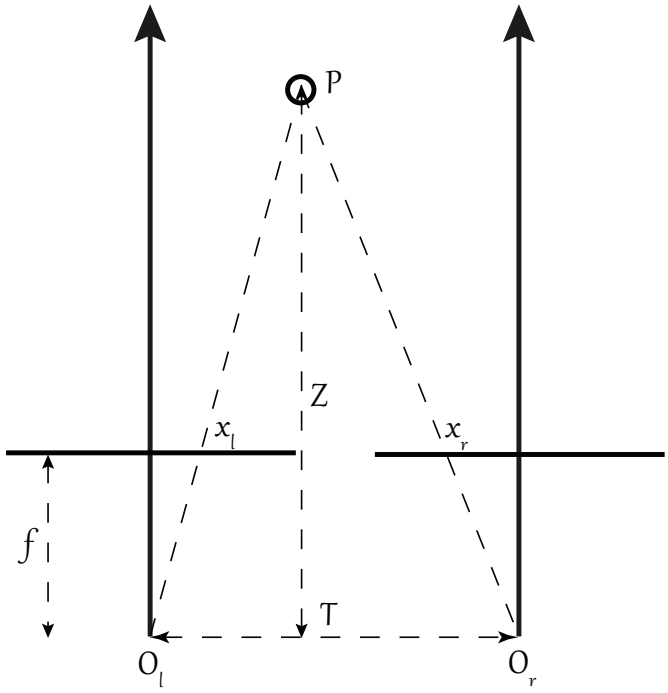


Figure 5.15: The algorithm of distance calculation.

Where, the unit of T is millimeter which is the same as Z, the dimensions of d and f are pixel. The Equation 5.35 can be simplified as follows:

$$Z=\frac{Tf}{d} \tag{5.36}$$

The discussion above is based on the situation that the optical axes of the two cameras are parallel. But in the real world, this kind of condition is too ideal. In a real situation, the two optical axes always have an angle in between.

The old method is to compute the rotation matrix by doing stereo calibration, then rotate one image to match the other one according to the rotation matrix. However, this

step requires a highly accurate calibration result and camera strictly fixed relatively to the position, which is what our on-board system is trying to avoid.

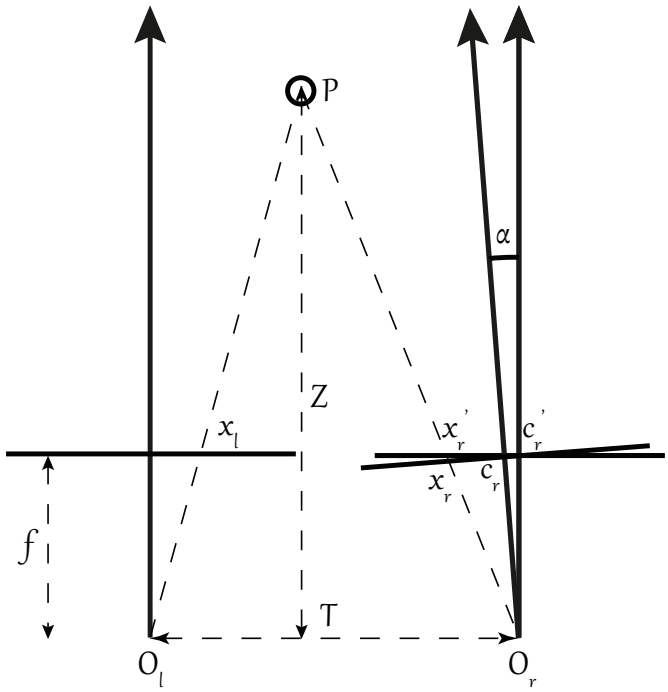


Figure 5.16: The sketch of distance calculation algorithm when one camera is rotated.

From Equation 5.35, x_l and x_r are the key issues in solving the distance. In Figure 5.16, f is the focal length, Z the distance that we want to calculate, c the centre of the image plant, x the abscissa of the object in the image, T the distance between the two cameras, and x_r the abscissa of the object in the right image, but the Equation 5.35 requires parameter x'_r.

Figure 5.17 shows a portion of Figure 5.16. Since AC_rO_r and $BC'_rO'_r$ are congruent triangles, $AC_r = BC'_r$, also since $\lim_{\theta\to 0} BX'_r = C_rX_r$, the relationship between x'_r and x_r is: $x'_r \approx x_r - AC_r$, where $AC_r = f\sin\theta$. The value of x'_r is finally given by the equation followed:

$$x'_r \approx x_r - f\sin\theta \tag{5.37}$$

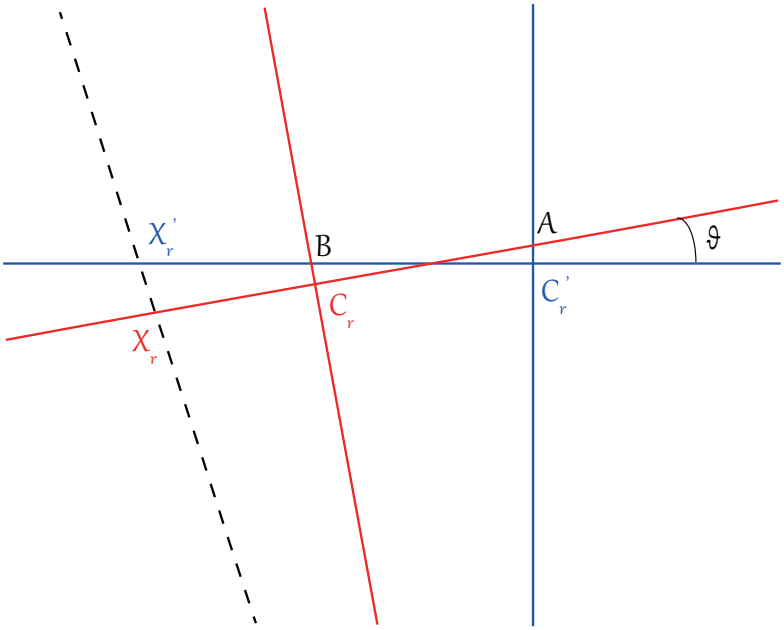


Figure 5.17: The close look of distance calculation algorithm when one camera is rotated.

Combine Equation 5.37 with Equation 5.35, the distance between the pedestrian and the car can be written as:

$$Z = \frac{Tf}{x_l - x_r + f\sin\theta} \tag{5.38}$$

5.3.4 Estimation of distance measurement results

Another Kalman filter tracker is used to estimate the distance calculation result. This time, the result of the Kalman filter is used to estimate the incorrect sudden change.

However, adding a tracker may cause a delay of the measured results, so the times of the iterations will change with the velocity of the vehicle. This is because the estimated value will approach to the measured value as the number of iterations increases. When the vehicle drives 10km/h faster, the times of iteration will add 1. The initial number of iterations is 2, for the reason that the pedestrian can walk fast.

5.4 Parameter Selection

As all the data is at sub-pixel quantity, the accuracy of the system depends on the focal length, the distance between cameras, and the resolution of the image captured.

5.4.1 Focal length

As previously discussed, the focal length of the lens will change with the velocity of the vehicle in order to detect the right range. This range will be decided by the braking distance and the resolution of the camera.

The minimal range of detection should be longer than the braking distance. The braking distance is given by [75]:

$$s = vt + \frac{M}{2gC_{ae}} \ln(1 + \frac{C_{ae}v^2}{\eta\mu M + f_r M \cos\theta \pm M \sin\theta}) \tag{5.39}$$

Where, g is coefficient of gravity (value for 9.8), M the load of the vehicle, v vehicle initial speed, η braking efficiency, μ the coefficient of road adhesion, f_r coefficient of rolling resistance and θ slope angle of road (+ when uphill, and − when downhill).

If we do not consider air resistance, we can simplify Equation 5.39 as follows:

$$s = vt + \frac{v^2}{2g(\eta\mu + f_r \cos\theta \pm \sin\theta)} \tag{5.40}$$

According to the Equation 5.40, when the vehicle is driving at 40 km/h on a flat wet asphalt road whose coefficient of friction is 0.7 [75] and f_r equal to 0.015, η is 0.9 thanks to the use of an anti-lock braking system. When the trigger delays 0.5 s, the braking distance will be 15.3 m, and the vehicle will stop after 2.26 s. Assuming a pedestrian walks at a normal speed of 1.5 m/s, during this period of time, he walks 3.38 m towards the driving direction of the car. If the general width of cars is 2.0 m, the minimal angle of view should be 32.2 degrees (see Figure 5.18). The equation for choosing the angle of view is:

$$\theta = 2\tan^{-1} \frac{[t + (s - tv_{car}) \div v_{car} \times 2] \times v_{pedestrian} + 0.5w}{s} \tag{5.41}$$

where w is the width of the car.

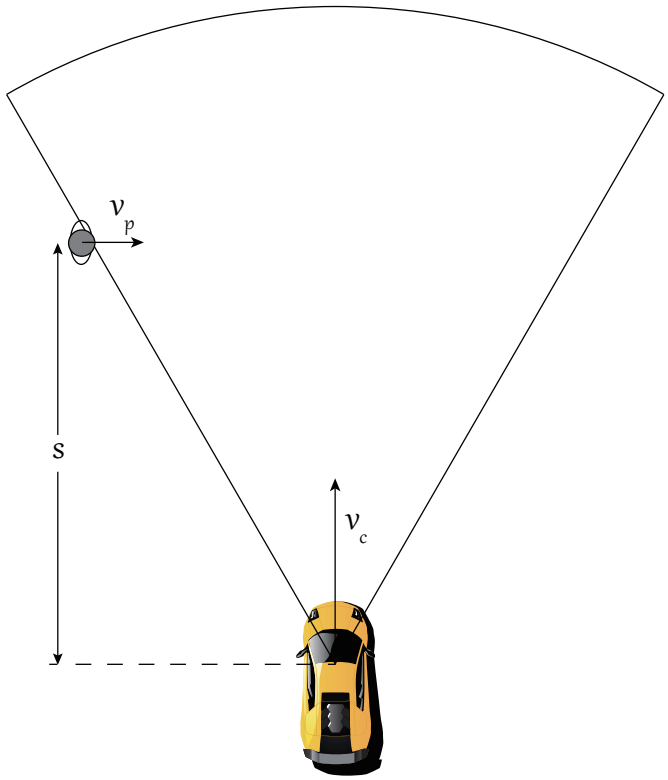


Figure 5.18: The Minimal Detecting Angle

When the car drives at the same condition but 100km/h, the angle of view should be at least 12.7 degrees, according to Equation 5.41.

The degrees calculated above give the minimal angle of view, but in practice the angle should be a little larger. Since the width of the area that the pedestrian walks is 6.76m from the example above, the total height of the area should be 5.07m for a 4:3 resolution. This height is enough for the pedestrian to be detected.

The field of view (FOV) of a camera is given by this equation:

$$FOV = 2\tan(\frac{d}{2f}) \tag{5.42}$$

where FOV is the angle of view, d is the dimension of the sensor and f is the focal length. From Equation 5.42, the focal length can be easily calculated by a given angle

of view which is calculated from the velocity of the car.

5.4.2 Image resolution

For real-time needs, the resolution should be no more then 800×600 pixels. The bigger image resolution, the more keypoints will be detected on the background, which has a bad effect on the accuracy of the matching step. The ideal resolution is 640×480 pixels, which is selected by trial.

For an image resolution larger than that, the keypoints of a pedestrian will be too much to verify an influence on speed; and, with a smaller resolution, the detection range will be too short because the size of the pedestrian is smaller in the image.

5.4.3 Distance between cameras

As stated before, the resolution is chosen as 640×480 pixels, for a pedestrian 50 meters away, using a focal length of 20 mm. This data is the detection distance and the corresponding suitable detection angle at speed of 80 km/h. For the 1/3" sensor we have selected (the width of the sensor is 4.8 mm), the focal length in pixels is 1000.

To meet the measurement accuracy requirement, the change of the value in 0.1 m should result in at least 1 pixel in the image. That means when $Z = 50$, $Z' = -10$ according to the Equation:

$$\begin{cases} Z = \dfrac{Tf}{d} \\ Z' = -\dfrac{Tf}{d^2} \end{cases} \tag{5.43}$$

We can get:

$$\begin{cases} 50 = \dfrac{Tf}{d} \\ -10 = -\dfrac{Tf}{d^2} \end{cases} \tag{5.44}$$

and, $d = 0.25$m as the solution of equation. Therefore, the minimum distance between cameras should be 250 mm in this case, which is much larger than the stereo cameras

can provide. In practice, the distance should be much larger to meet the accuracy requirement. We propose that the distance between the two cameras is preferably 600 mm, which can keep a balance between the accuracy and the dead angle caused by the parallax.

Chapter 6

Experiment and Result Analysis

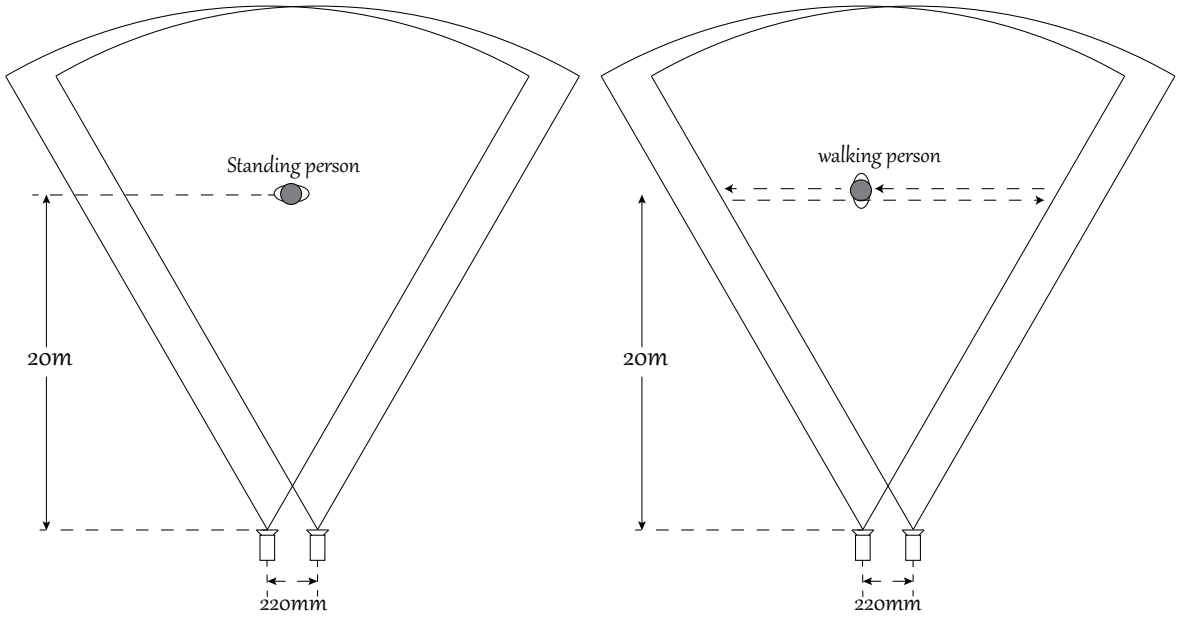
In this section, the overall feasibility of the on-board pedestrian detection and distance measurement system is tested; meanwhile, some methods that play the same role are compared for both accuracy and speed.

Accuracy, frame rate and on-vehicle performance are judged in this chapter.

6.1 Accuracy Test

According to ISO 5725-1:1994 *Accuracy (trueness and precision) of measurement methods and results*, the accuracy of a measurement is described by its level of trueness and precision. An experiment was conducted for the situation in which a pedestrian is at a distance of 20 m away from the cameras, as shown in Figure 6.1. The results of the measurement are calculated according to Equation 5.38. The calibration result shows that the focal lengths of the two cameras are 3,674 and 3,747; and, the angle between them is 0.26°. In the program, the focal length is set to 3,710 which is the geometric mean of 3,674 and 3,747; and, the angle between the optical axes is set to 0.26°. The distance between the two cameras is 220 mm which is the maximum value of the tripod that we bought.

The distance of the target and the distance between the cameras is linearly related (see Equation 5.38). We can thus say that the result of this experiment, limited by the performance of the tripod, is approximately equal to an experiment where the target is standing at 50 m away. We can moreover say that the distance between the cameras is 600 mm which is the optimal value.



(a) standing target

(b) moving target

Figure 6.1: Diagram of the experiment.

The experiment for both methods (with and without tracking) has been run 8 times for standing pedestrians vs. walking pedestrians, using SURF vs. ORB as the keypoint extractor, and RANSAC vs. LMedS as the method of estimation. For standing targets, 496 results were gathered for each measurement using the same video. For walking targets, 400 results were gathered for each combination of methods by also using a shared video clip. The accuracy of the calibration was not considered.

6.1.1 Performance of Trueness

The level of trueness refers to the closeness of the measurement results to the actual value. Table 6.1 shows a comparison between SURF against ORB, and RANSAC against LMedS. In this table, we discover that RANSAC has a better performance in the estimation of keypoint match, and that ORB does better with walking targets but a little worse with standing targets. However, the difference between the mean of the results and the true distance may also be caused by the calibration and the measurement of the true value; the level of trueness of the system is about the same for the two keypoint detectors.

Table 6.1: The comparison of the mean in the 4 measurement results while true value is 20 m.

		The mean of results (m)	
Keypoint	Estimate	Standing target	Walking target
SURF	RANSAC	19.8901	20.6669
SURF	LMedS	19.9427	21.1907
ORB	RANSAC	20.1851	20.6895
ORB	LMedS	20.1317	20.9672

Figure 6.2 represents the probability density functions (PDF) of the 4 combinations of the keypoint detector (ORB vs SURF) and the linear estimation method (LMedS vs RANSAC) compared with a normal PDF, which have a mean value $\mu = 20$ and variance $\sigma = 1.5$. The figure shows that the four measurements are pretty close to the normal PDF. This means that the error in this system follows the normal distribution. The figure also shows a slightly better performance of ORB than SURF at the actual value. This is the case, although the highest value of ORB results is higher than that of the SURF results, which can be easily eliminated by adding a tracking method on the results

from continuous frames.

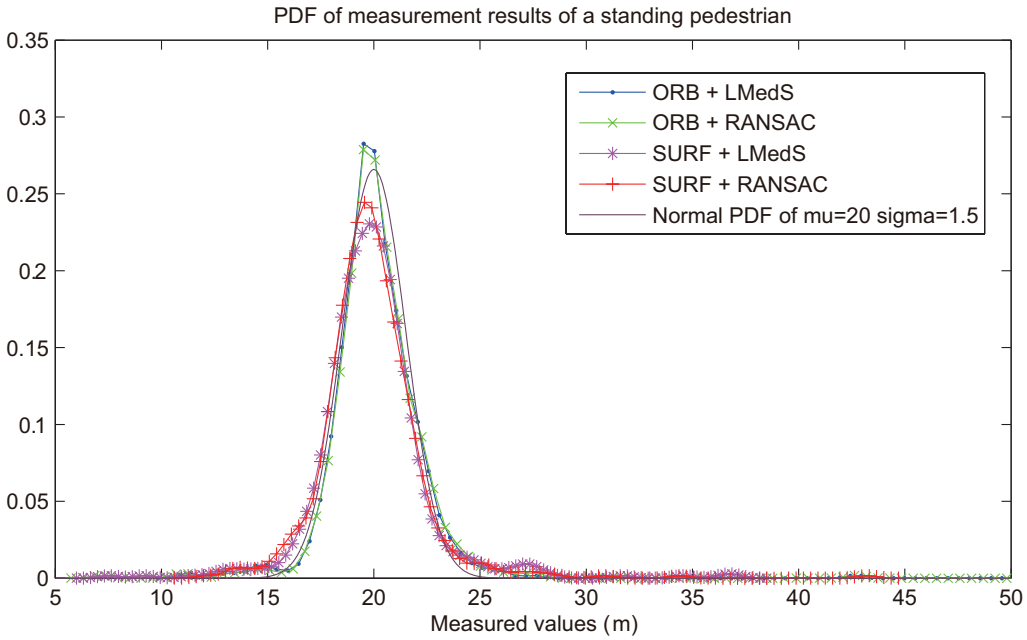


Figure 6.2: PDF of the 4 combinations measuring standing target compared with a normal PDF when $\mu = 20$ and $\sigma = 1.5$.

From this, we can say that the ORB and RANSAC demonstrate greater trueness than SURF and LMedS, even though the difference is not significant.

6.1.2 Performance of precision

In a measurement system, precision refers to the closeness of agreement within individual results or the degree of dispersion.

The Figure 6.3 shows the box plots of the four combinations of the method (SURF, ORB, RANSAC, LMedS), measuring the standing target 20 m away. The figure on the right shows the scaled view of the box plots around a 20 m point. The interquartile range (IQR) of ORB is more narrow and the median is closer to the actual point than the IQR of SURF. That means ORB outperforms SURF not only in trueness but also in precision. This figure also shows that RANSAC performs better than LMedS.

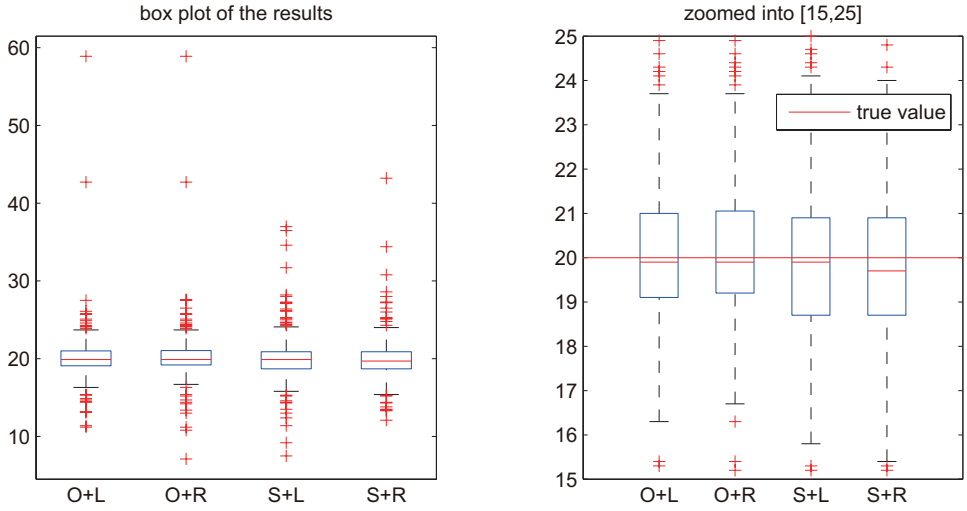


Figure 6.3: Boxplot of the 4 combinations of measurement with standing target at 20 m. (O:ORB, S:SURF, L:LMedS, R:RANSAC)

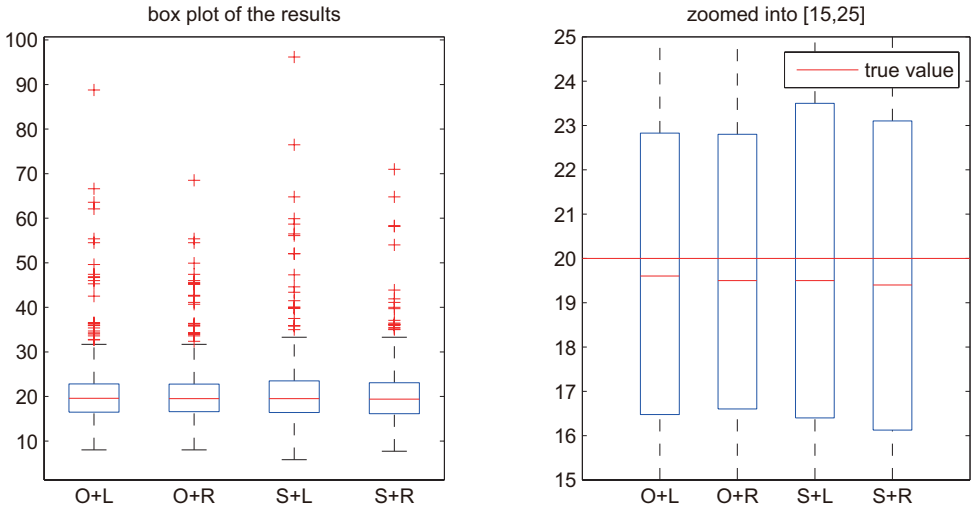


Figure 6.4: Boxplot of the 4 combinations of measurement with moving target at 20.0m. (O:ORB, S:SURF, L:LMedS, R:RANSAC)

Although there are two fatal outliers in the results obtained using ORB which are shown in the left plot, they can be easily eliminated while the measurement results are processed under another tracker. This can be designed in a very simple way that can prohibit hopping.

The box plots in Figure 6.4 shows the results tested with a moving target. The microcosmic scaled figure shows that when measuring a moving pedestrian, ORB does superior to SURF. The IQR of ORB+RANSAC is the best among these four methods; the median is, however, a little weaker than ORB+LMedS. The large scaled figure, on the other hand, shows a significantly better performance in the centralization of RANSAC than LMedS.

Figure 6.5 shows the box plot of the measurement results of a target standing at 15 m. In this figure, the performances of ORB and SURF are nearly the same; however, LMedS shows a slightly better performance than RANSAC. This is because at 15 m, the pedestrian in the image is larger than at 20m and is attributed richer details. So there are more keypoints picked out for the pedestrian than for the background; this also means that the outliers of keypoints are less. In this case, LMedS shows better stability than RANSAC. However this kind of difference is too slight and the range in difference of IQR is no more than 0.1, which is an error margin of 7.5%. The accuracy of camera calibration cannot be tested and the error is not quantified. Additionally, the camera is not fixed and is thus not absolutely stable. Therefore, there must be some tiny deformation between the test and the calibration; this is similar to the situation of an unstable on-board environment. We can say that results obtained are pretty close to the real system.

Tracking with Kalman operator

Since there are many unpredictable outliers in the results, a Kalman tracker has been added to estimate the results. The Boxplots 6.6 and 6.7 show the same video clips in

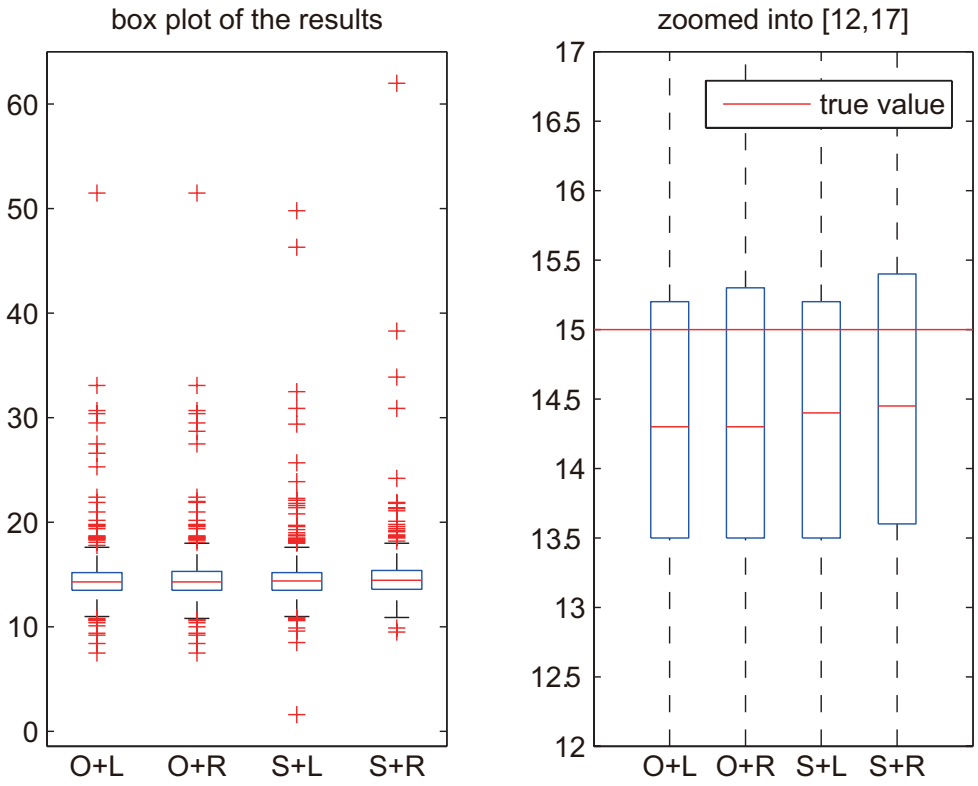


Figure 6.5: Boxplot of the 4 combinations of measurement with standing target at 15 m. (O:ORB, S:SURF, L:LMedS, R:RANSAC)

which a scenario with a pedestrian standing at 20 m was tested with the same methods, but using a Kalman tracker after the calculation.

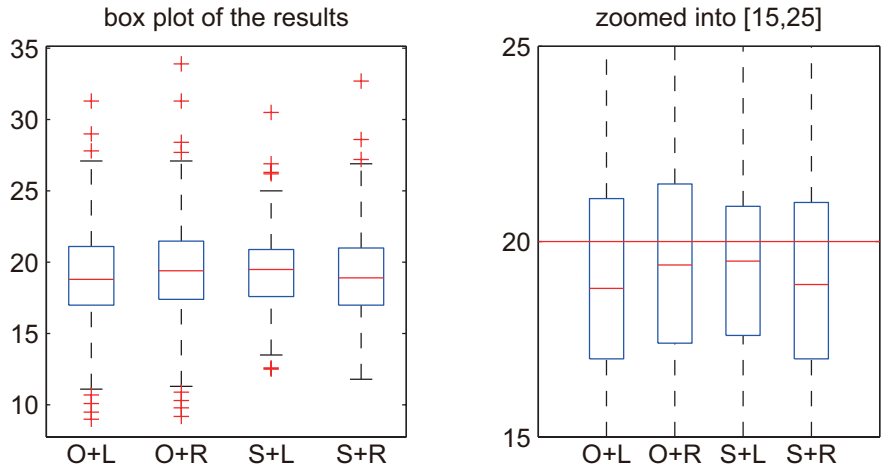


Figure 6.6: Boxplot of the 4 combinations of measurements with standing target at 20 m. (O:ORB, S:SURF, L:LMedS, R:RANSAC)

The boxplots show that the outliers have been significantly eliminated, and that the data is more stable. However, they also show a phenomenon in which the confidence interval is actually wider than the results noted before. This is because the tracker magnifying the influence of the off-center points and the outliers are brought back to the acceptable range.

From the PDF of the results using the Kalman tracker, in Figure 6.8, we can see that the probability is much higher and closer to the true value than before. However, the different order in the data sequence leads to different results when applied with a tracker. So, the PDF performs unlike the normal PDF; also, the improvement is significant and effective.

To summarize, ORB+RANSAC is somewhat superior to the other three combinations in terms of performance accuracy.

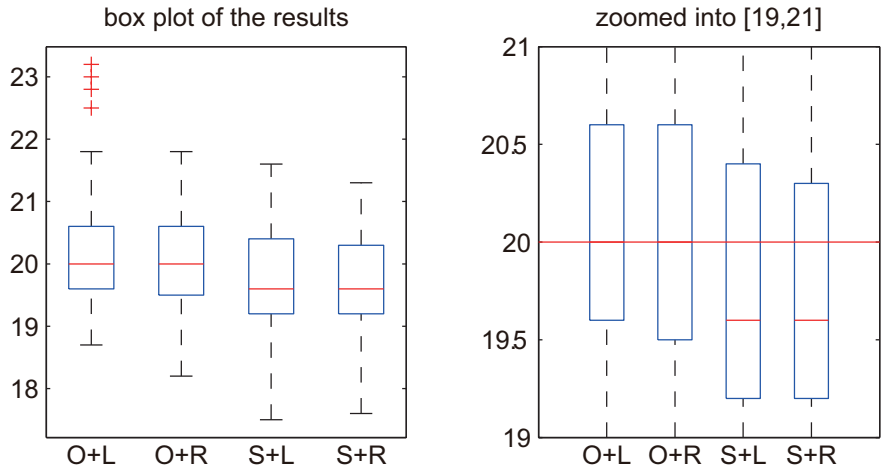


Figure 6.7: Boxplot of the 4 combinations of measurements with moving target at 20 m. (O:ORB, S:SURF, L:LMedS, R:RANSAC)

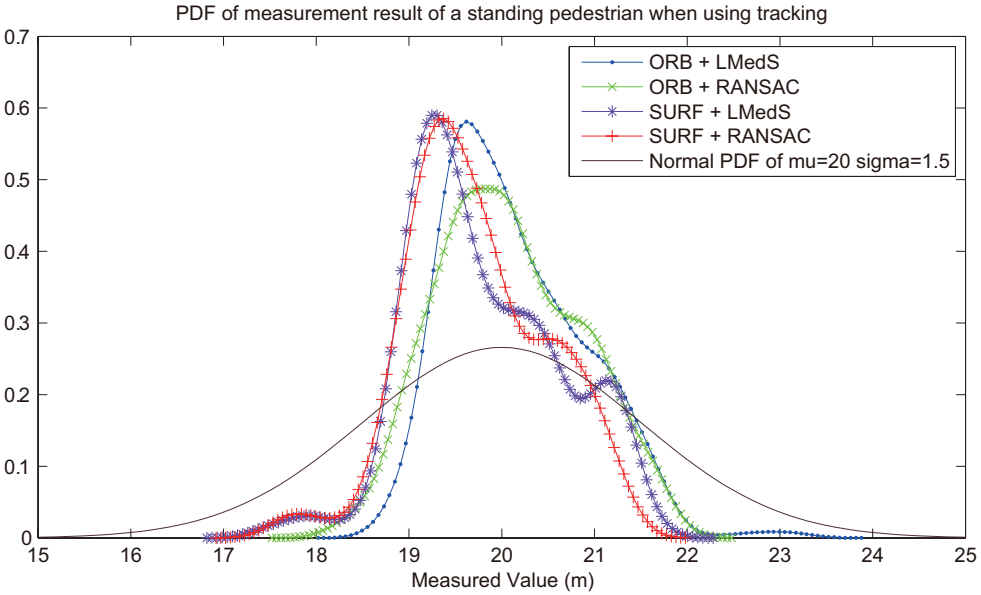


Figure 6.8: PDF of the 4 measurement combinations for standing targets with Kalman tracker.

6.2 On-vehicle Experiment

Some on-vehicle experiments are tested under different scenarios: highway and urban.

6.2.1 Testbed

An on-vehicle experiment aims to test the performance of the system when the vehicle is being driven unsuitably. This experiment is done on a straight road in the city of Ottawa (see Figure 6.9). In this experiment, the car is equipped with pedestrian detection and the distance measurement system is driven at 60 km/h. The pedestrian is standing at the roadside for safety.

It is impossible to test the trueness because it is impossible to get a comparable result for other reliable methods of measurement which give a result at just the time that this system does.

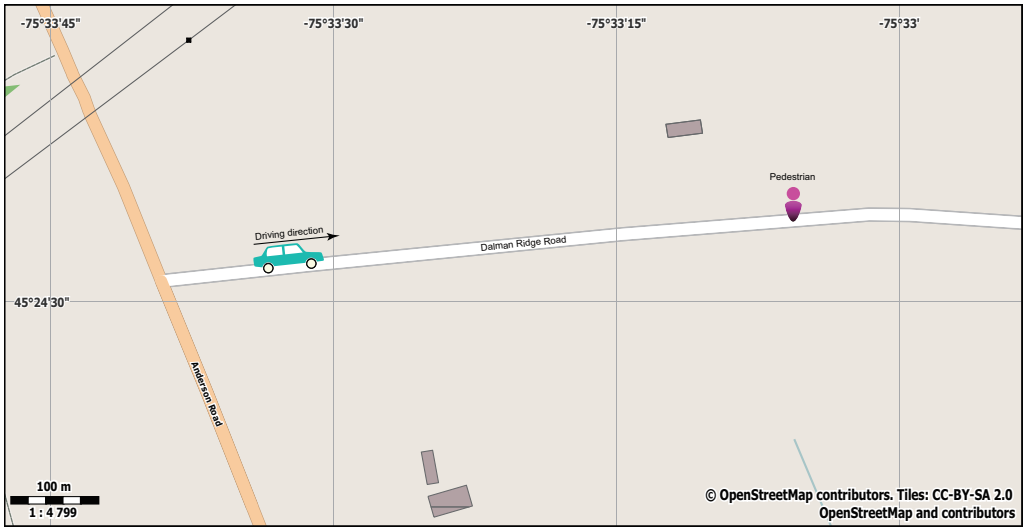


Figure 6.9: The schematic diagram of an on-vehicle expriment.

6.2.2 Result

However, from Figure 6.10, we witness a pretty good level of stability even when the car is driving at 60 km/h, which means that the iteration time of the Kalman tracker is 8, when the focal length is 5,900. The figure shows that the system started tracking and measuring the distance from the pedestrian at 47 m and then the pedestrian vanished from the detection range at around 23 m. This does not mean that the system can only detect pedestrians 23 m away. The reason the pedestrian vanished into the side of the image is that he walked on the roadside as a safety measure.

Figure 6.11 shows snapshots of a video clip every four frame. We can see that the detection window keeps tracking the target in a very stable manner; and, when the pedestrian vanished from the left side of the camera view, the detection window continued tracking until the 8th frame.

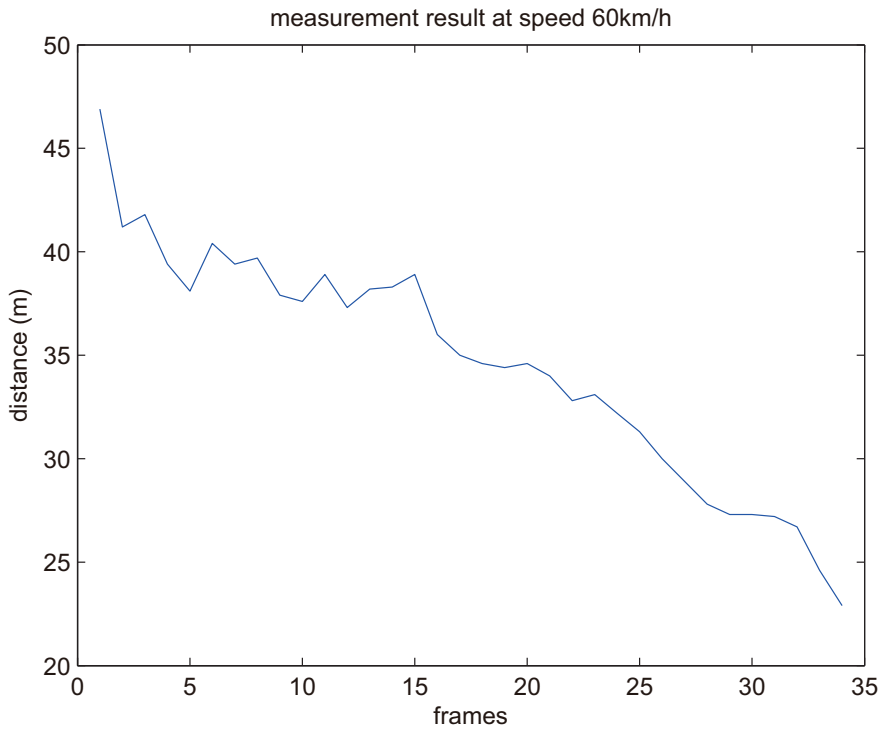


Figure 6.10: Result of distance measurement while driving at 60km/h.

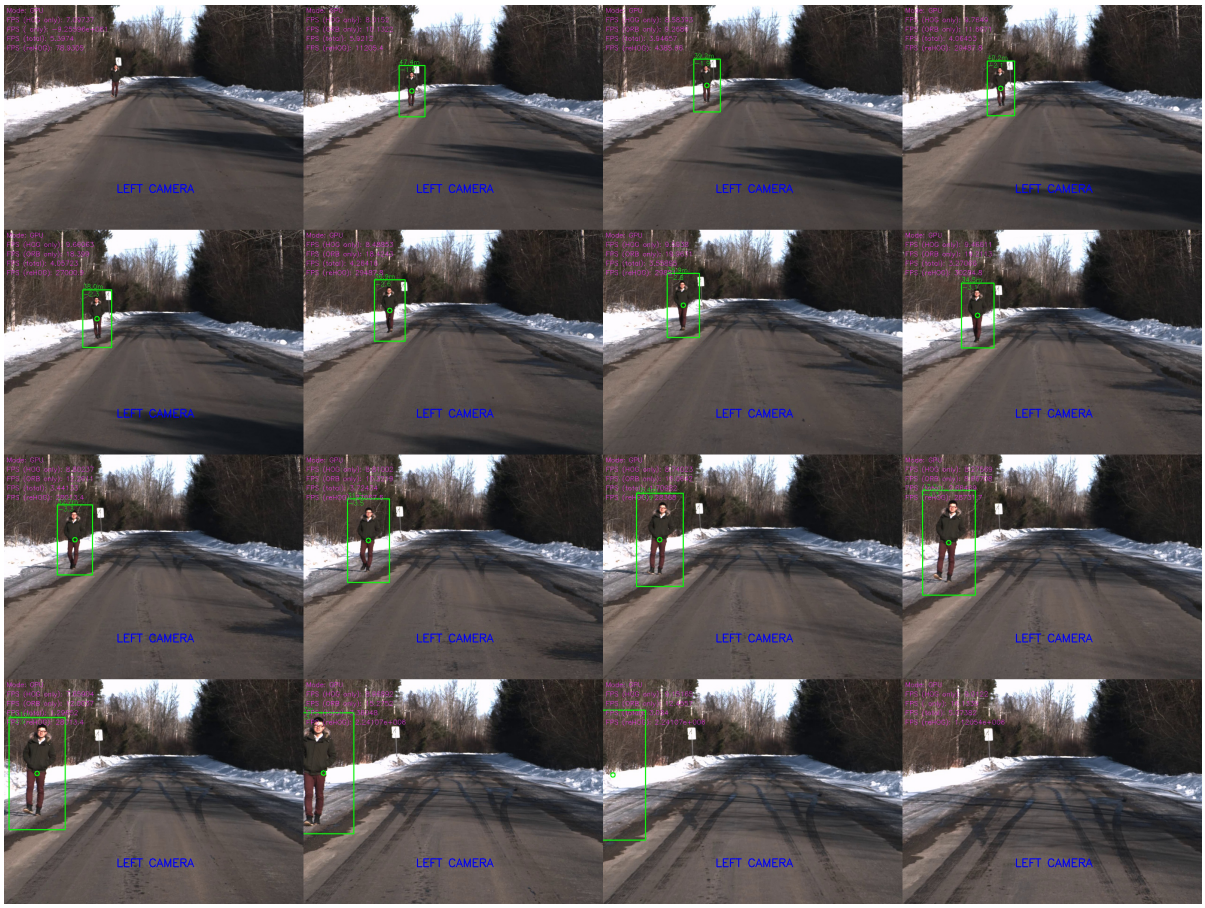


Figure 6.11: The image from the left camera while driving at 60km/h. In this test, the car is driving toward the pedestrian at high speed.

Another test shows that in a urban driving scenario, the camera is set at the focal length of 2,600. The results show good stability when the car is running at low speed. In Figure 6.12, we can find the pedestrian is detected soon after he walked into the FOV of the camera.

From the measurement results shown in Figure 6.13, we see that there are no erroneous points in the data. The error in the start is due to the initialization of the Kalman filter.

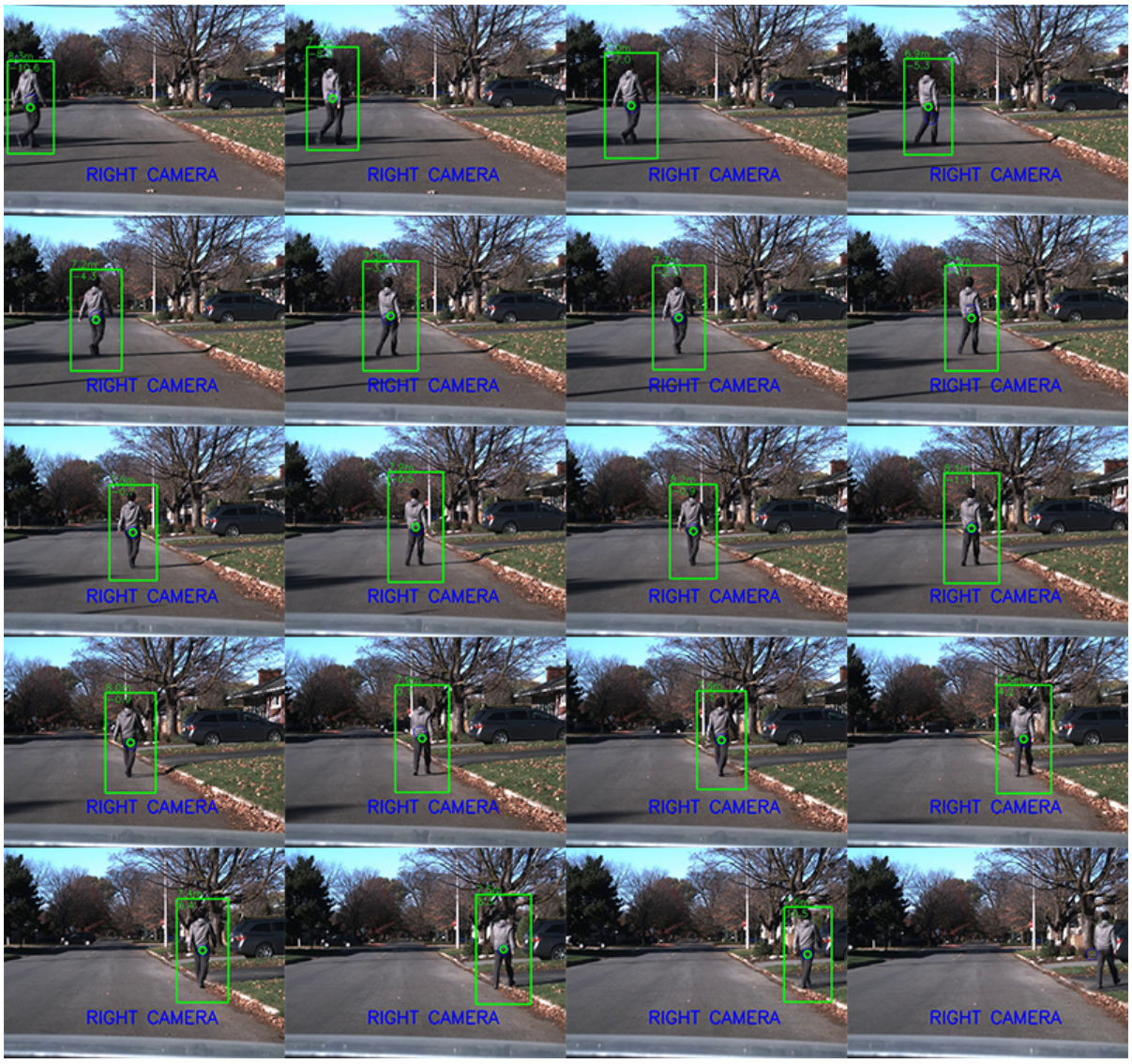


Figure 6.12: The image from the right camera while driving in urban scenario.

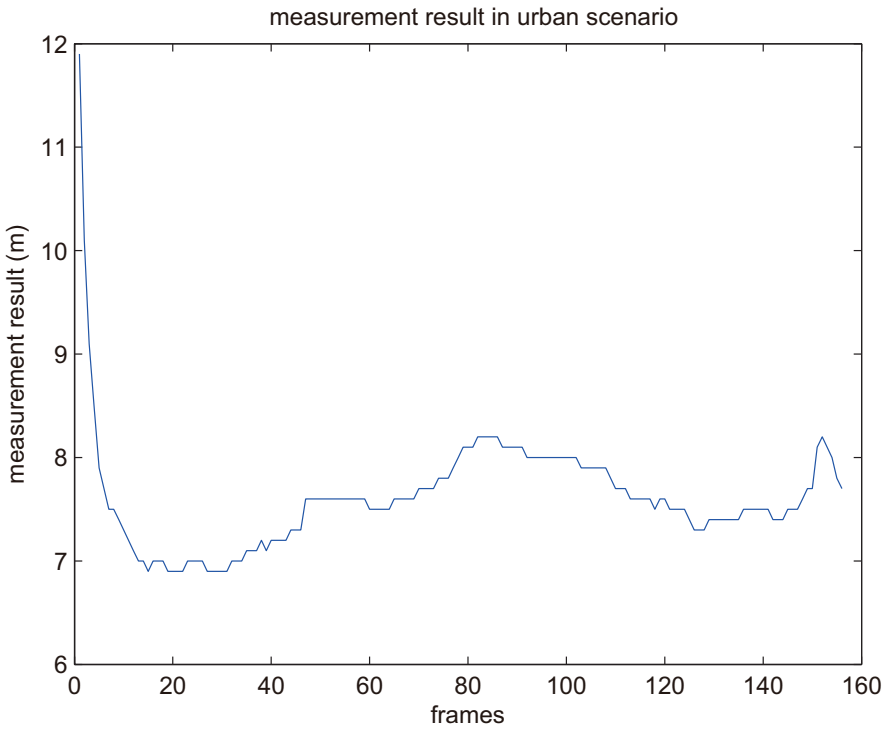


Figure 6.13: Result of distance measurement in a urban scenario. In this test, the driver is trying to maintain the distance between the car and the pedestrian.

6.3 Frame Rate

The speed test shows the four main method combinations comparing CPU SIFT and CPU SURF.

The speed test is run in single-pedestrian scenario and the image size is 640×480 on a laptop which is mentioned in Section 2.4. The speed of ORB and SURF is the most important information to be compared. The results in Table 6.2 show that ORB is 2 to 3 frames faster than SURF at every second. If the number of pedestrians increases, the speed difference will become more significant.

There is no GPU SIFT in OpenCV, but when comparing the speed of CPU SURF with the one of GPU, the extent to which GPU accelerates is clearly shown. So we can speculate that GPU SIFT should also be the slowest of all these methods.

Table 6.2: The comparison of average fps (using debug) of measurements.

Keypoint	Estimate	matching fps	matching time(ms)	total fps
SURF	RANSAC	20.9899	47.64	6.9692
SURF	LMedS	21.0904	47.14	6.9980
ORB	RANSAC	23.4989	42.56	6.9851
ORB	LMedS	23.0357	43.41	6.9125
CPU SURF	RANSAC	3.4734	287.90	2.6555
CPU SIFT	RANSAC	0.5654	1768.66	0.5449

The total frame per second (fps) is the overall speed of the whole program. This was also effected by the speed of HOG+SVM. The sample size may influence the results so that there is scarcely a difference between candidates. Compared to the fps of keypoint detector and linear estimation, HOG+SVM consumes the most time. Even though GPU acceleration is used in the program, the overall fps equals approximately 7. This means that at the speed of 50 km/h, the distance measurement result will lag 2 metres behind

the true distance. However, the performance of speed here is subject to the performance of the computer and the graphics card.

According to the paper of Mammeri [61], 57.1 ms would be cost in the double-layer animal detection system which uses LBP with AdaBoost as the first layer and HOG+SVM as the second layer. This could lead us to infer that the time consumed when measuring the part applied to animal detection will not be long either. The system is basically in line with real-time requirements.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The experimental results clearly show that the keypoint-based binocular distance measurement system works well in an on-vehicle environment. Using ORB as the keypoint extractor and RANSAC as the estimation method is the best combination so far, with an error margin of 7.5% in long range detection. The frame rate can meet the requirement of real time processing with stronger GPU.

Introducing the concept of crossover control into a detection system can boost detection accuracy and stabilize tracking. Our method enhanced the tolerance of instability of a moving vehicle and also gives the binocular measurement system the ability to be equipped with zoom lenses and allowing greater spacing.

Contribution

The contribution of our thesis can be summarized as follows:

1. Design a keypoint-based binocular photogrammetry system using a zoom lens camera.

2. Design a two-image crossover detection result synchronization process.

3. Reform a crossover control Kalman Filter for binocular system.

4. Simplify the camera rotation model.

7.2 Future Work

In the future, this system can be improved following two method.

The first method consists of changing the detecting system from a pedestrian detection to a car tracking system using tracking learning detection [76]; it also requires building a front car tracking adaptive cruise driving system which can distinguish which car we are following.

The second method involves replacing the existing camera with a camera mounted with motor a zoom lens and pan-tilt (see in Figure 7.1). In keeping with out observation that the focal length will change related to the velocity of the vehicle.

The camera will pan inward according to the speed as well. The inward panning action will reduce the dead angle caused by the distance between the cameras seen at Figure 7.2. In the figure, the light-colored area is the dead angle of the system in one camera view but not in the other one. The panning will control the overlapping surfaces only on the distance that concerns us.

Also, while turning a corner or changing direction on highway, the cameras will spin codirectionally 7.3. With this kind of rotation, the camera will keep the field of view on the road.

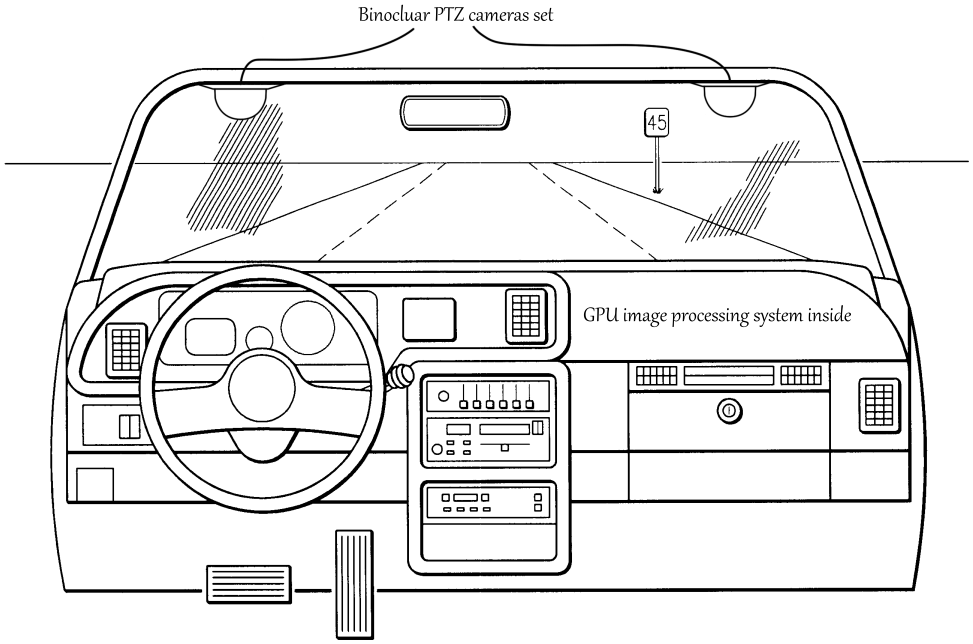
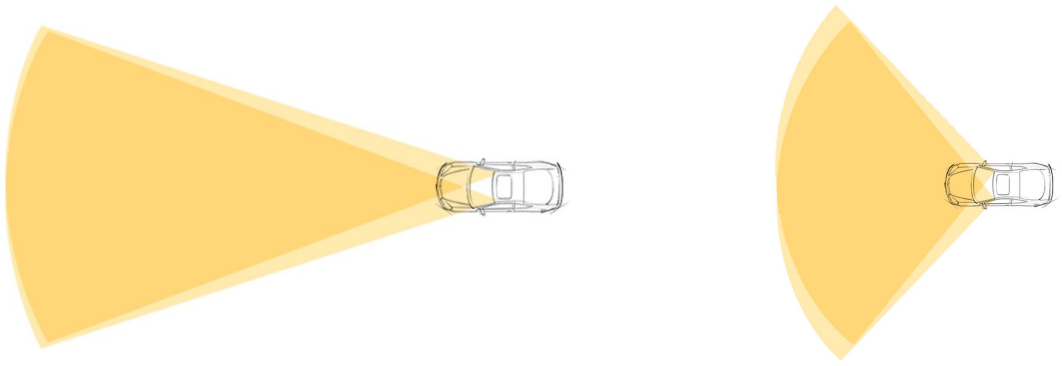


Figure 7.1: Binocular PTZ camera set on vehicle.



(a) telescope

(b) wide angle

Figure 7.2: The dead angle of the two cameras can be reduced by inward panning.

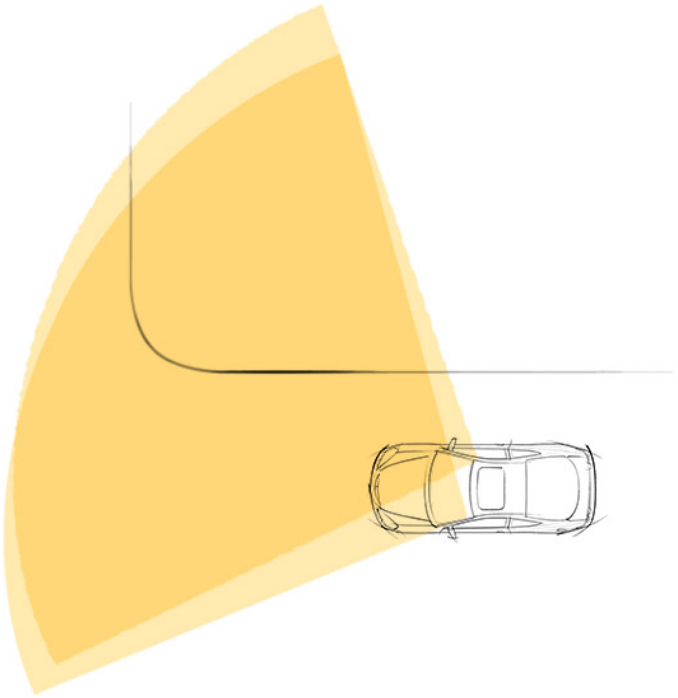


Figure 7.3: Codirectionally panning around conner.

Bibliography

[1] David H Hubel. *Eye, brain, and vision.* Scientific American Library/Scientific American Books. ISSN 1040-3213, 1995.

[2] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

[4] Azzedine Boukerche and Xin Fei. A coverage-preserving scheme for wireless sensor network with irregular sensing range. *Ad hoc networks*, 5(8):1303–1316, 2007.

[5] Azzedine Boukerche and Xu Li. An agent-based trust and reputation management scheme for wireless sensor networks. In *Global Telecommunications Conference. GLOBECOM'05.*, volume 3, pages 5–pp. IEEE, 2005.

[6] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina B Araujo. Hpeq a hierarchical periodic, event-driven and query-based wireless sensor network protocol. In *Local Computer Networks. 30th Anniversary. The IEEE Conference on*, pages 560–567, 2005.

[7] Yonglin Ren and Azzedine Boukerche. Modeling and managing the trust for wireless and mobile ad hoc networks. In *Communications. ICC'08. IEEE International Conference on*, pages 2129–2133. IEEE, 2008.

[8] Azzedine Boukerche and Yonglin Ren. A secure mobile healthcare system using trust-based multicast scheme. *Selected Areas in Communications, IEEE Journal on*, 27(4):387–399, 2009.

[9] Azzedine Boukerche and Luciano Bononi. Simulation and modeling of wireless, mobile, and ad hoc networks. *Mobile ad hoc networking*, pages 373–409, 2004.

[10] HAB de Oliveira, Eduardo Freire Nakamura, Antonio Alfredo Ferreira Loureiro, and Azzedine Boukerche. Directed position estimation: A recursive localization approach for wireless sensor networks. In *Computer Communications and Networks. Proceedings. 14th International Conference on*, pages 557–562. IEEE, 2005.

[11] Azzedine Boukerche, Sajal K Das, and Alessandro Fabbri. Swimnet: a scalable parallel simulation testbed for wireless and mobile networks. *Wireless Networks*, 7(5):467–486, 2001.

[12] microsoft. Kinect for windows. URL: `http://www.microsoft.com/en-us/kinectforwindows/`.

[13] Johnny Lee and the ATAP-Project Tango Team. Project tango, 2014. URL: `http://www.google.com/atap/projecttango/`.

[14] Cost of auto crashes and statistics, 2013. URL: `http://www.rmiia.org/auto/traffic_safety/Cost_of_crashes.asp`.

[15] *Transport for health: the global burden of disease from motorized road transport*. The World Bank, 2014.

[16] World Health Organization et al. Pedestrian safety: a road safety manual for decision-makers and practitioners. 2013.

[17] Automobile driving assisting system, February 16 2011. CN Patent 201,745,529. URL: `http://www.google.com/patents/CN201745529U?cl=en`.

[18] Brett Miller and Daniel Pitton. Vehicle blind spot detector, September 15 1987. US Patent 4,694,295. URL: `http://www.google.com/patents/US4694295`.

[19] Shinichi Yamano, Hirofumi Higashida, Masayoshi Shono, Sadanori Matsui, Tomohiko Tamaki, Hidekazu Yagi, and Hisateru Asanuma. 76ghz millimeter wave automobile radar using single chip mmic. *Fujitsu Ten Tech. J*, (23):12–19, 2004.

[20] Jürgen Hasch, Eray Topak, Raik Schnabel, Thomas Zwick, Robert Weigel, and Christian Waldschmidt. Millimeter-wave technology for automotive radar sensors in the 77 ghz frequency band. *Microwave Theory and Techniques, IEEE Transactions on*, 60(3):845–860, 2012.

[21] Kazuo SHIRAKAWA, Shuhei KOBASHI, Yasuhiro KURONO, Masayoshi SHONO, and Osamu ISAJI. 3d-scan millimeter-wave radar for automotive application. *Fujitsu Ten Tech. J*, 38:3–7, 2013.

[22] Datasheet of hd lidar hdl-64e, 2010. URL: `http://velodynelidar.com/lidar/products/brochure/HDL-64E%20S2%20datasheet_2010_lowres.pdf`.

[23] Christian Boehlau and Johann Hipp. Optoelectric sensing device with common deflection device, March 18 2008. US Patent 7,345,271. URL: `https://www.google.com/patents/US7345271`.

[24] Subaru eyesight, Accessed on 2014. URL: `http://www.subaru.com/engineering/safety.html`.

[25] Bosch stereo video camera enhances comfort and safety, Accessed on 2014. URL: `http://www.bosch-presse.de/presseforum/details.htm?txtID=5951&locale=en`.

[26] James L Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *Robotics and Automation, International Conference on*, pages 674–680. IEEE, 1989.

[27] Martin Schneider. Automotive radar–status and trends. In *German microwave conference*, pages 144–147, 2005.

[28] Brian D. Cordill, Sarah A. Seguin, and Lawrence Cohen. Electromagnetic interference to radar receivers due to in-band ofdm communications systems. In *Electromagnetic Compatibility (EMC), 2013 IEEE International Symposium on*, pages 72–75.

[29] Paul E Bauhahn, Bernard S Fritz, and Brian C Krafthefer. Systems and methods for safe laser imaging, detection and ranging (lidar) operation, November 5 2009. US Patent App. 12/112,517. URL: `https://www.google.com/patents/US20090273770`.

[30] Fernando Garcia, Pietro Cerri, Alberto Broggi, Jose Maria Armingol, and Arturo de la Escalera. Vehicle detection based on laser radar. In *Computer Aided Systems Theory-EUROCAST*, pages 391–397. Springer, 2009.

[31] David Marr and Tomaso Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 204(1156):301–328, 1979.

[32] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, volume 1, pages I–195. IEEE, 2003.

[33] Vincent Couture, Nicolas Martin, and Sebastien Roy. Unstructured light scanning to overcome interreflections. In *Computer Vision (ICCV), IEEE International Conference on*, pages 1895–1902. IEEE, 2011.

[34] Thomas Kerstein, Martin Laurowski, Philipp Klein, Michael Weyrich, Hubert Roth, and Jürgen Wahrburg. Optical 3d-surface reconstruction of weak textured objects based on an approach of disparity stereo inspection. In *Proceedings of International Conference on Pattern Recognition and Computer Vision (ICPRCV)*, number 78, pages 581–586, 2011.

[35] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673. IEEE, 1999.

[36] William Hoff and Narendra Ahuja. Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(2):121–136, 1989.

[37] Maxime Lhuillier and Long Quan. Match propagation for image-based modeling and rendering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(8):1140–1146, 2002.

[38] Christian Banz, Sebastian Hesselbarth, Holger Flatt, Holger Blume, and Peter Pirsch. Real-time stereo vision system using semi-global matching disparity estimation: architecture and fpga-implementation. In *Embedded Computer Systems (SAMOS), 2010 International Conference on*, pages 93–101. IEEE, 2010.

[39] Chang-Il Kim and Soon-Yong Park. Fast stereo matching of feature links. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), International Conference on*, pages 268–274. IEEE, 2011.

[40] S Röhl, S Bodenstedt, S Suwelack, H Kenngott, BP Mueller-Stich, R Dillmann, and S Speidel. Real-time surface reconstruction from stereo endoscopic images for intraoperative registration. In *SPIE Medical Imaging*, pages 796414–796414. International Society for Optics and Photonics, 2011.

[41] Lu Xia, Chia-Chih Chen, and JK Aggarwal. Human detection using depth information by kinect. In *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Computer Society Conference on*, pages 15–22. IEEE, 2011.

[42] Brojeshwar Bhowmick, Sambit Bhadra, and Arijit Sinharay. Stereo vision based pedestrians detection and distance measurement for automotive application. In *Intelligent Systems, Modelling and Simulation (ISMS), Second International Conference on*, pages 25–29. IEEE, 2011.

[43] Walker Philip Hulme. Ultrasonic object detection systems, July 23 1968. US Patent 3,394,342. URL: `https://www.google.com/patents/US3394342`.

[44] Helmut Saufferer. Distance warning device for vehicles, July 1 1975. US Patent 3,892,483. URL: `https://www.google.com/patents/US3892483`.

[45] J Wenger. Automotive mm-wave radar: Status and trends in system design and technology. In *Automotive Radar and Navigation Techniques (Ref. No. 1998/230), IEE Colloquium on*, pages 1–1. IET, 1998.

[46] Katsutoshi Tagami, Takaya Senzaki, Nobuhiko Suzuki, and Eiji Murao. Automotive radar monitor system, September 14 1982. US Patent 4,349,823. URL: `https://www.google.com/patents/US4349823`.

[47] Andrew A Frank and Masahiko Nakamura. Laser radar for a vehicle lateral guidance system, April 13 1993. US Patent 5,202,742.

[48] Xuesong Mao, Daisuke Inoue, Satoru Kato, and Manabu Kagami. Amplitude-modulated laser radar for range and speed measurement in car applications. *Intelligent Transportation Systems, IEEE Transactions on*, 13(1):408–413, 2012.

[49] Gerhard Nocker. Method for controlling the distance between moving motor vehicles, December 20 1994. US Patent 5,375,060. URL: `https://www.google.com/patents/US5375060`.

[50] James D Spinhirne. Micro pulse lidar. *Geoscience and Remote Sensing, IEEE Transactions on*, 31(1):48–55, 1993.

[51] Josef Wenger. Automotive radar-status and perspectives. In *Compound Semiconductor Integrated Circuit Symposium (CSIC)*, pages 4–pp. IEEE, 2005.

[52] Two eyes for binocular vision: Bosch stereo video camera enhances comfort and safety, 2012. URL: `http://www.bosch-presse.de/presseforum/pressdownload/text/PI7950.pdf?id=5951,2`.

[53] nVidia. Nvidia automotive driving innovation, 2014. URL: `http://www.nvidia.com/object/tegra-automotive.html`.

[54] Victor Prisacariu and Ian Reid. fasthog-a real-time gpu implementation of hog. *Department of Engineering Science, Oxford University, Tech. Rep*, 2310(09), 2009.

[55] Sudipta N Sinha, Jan-Michael Frahm, Marc Pollefeys, and Yakup Genc. Gpu-based video feature tracking and matching. In *EDGE, Workshop on Edge Computing Using New Commodity Architectures*, volume 278, page 4321, 2006.

[56] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), International Conference on*, pages 2564–2571. IEEE, 2011.

[57] Christian Banz, Holger Blume, and Peter Pirsch. Real-time semi-global matching disparity estimation on the gpu. In *Computer Vision Workshops (ICCV Workshops), International Conference on*, pages 514–521. IEEE, 2011.

[58] David Geronimo, Antonio M Lopez, Angel Domingo Sappa, and Thorsten Graf. Survey of pedestrian detection for advanced driver assistance systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(7):1239–1258, 2010.

[59] Hui-Xing Jia and Yu-Jin Zhang. Fast human detection by boosting histograms of oriented gradients. In *Image and Graphics (ICIG). Fourth International Conference on*, pages 683–688. IEEE, 2007.

[60] John Owens, UC Davis, David Luebke, and NVIDIA. Intro to parallel programming, an open, online course from udacity. URL: `http://www.nvidia.com/object/cuda_home_new.html`.

[61] Abdelhamid Mammeri, Depu Zhou, Azzedine Boukerche, and Mohammed Almulla. An efcient animal detection system for smart cars using cascaded classiers. IEEE International Conference on Communications, 2014.

[62] Debao Zhou. Thermal image-based deer detection to reduce accidents due to deer-vehicle collisions. 2013.

[63] Quan Yuan, Ashwin Thangali, Vitaly Ablavsky, and Stan Sclaroff. Learning a family of detectors via multiplicative kernels. In *Topics in Medical Image Processing and Computational Vision*, pages 1–32. Springer, 2013.

[64] Sayanan Sivaraman and Mohan M Trivedi. A general active-learning framework for on-road vehicle recognition and tracking. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):267–276, 2010.

[65] Hossein Tehrani Niknejad, Akihiro Takeuchi, Seiichi Mita, and David McAllester. On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation. *Intelligent Transportation Systems, IEEE Transactions on*, 13(2):748–758, 2012.

[66] Greg Welch and Gary Bishop. An introduction to the kalman filter, 1995.

[67] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[68] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[69] Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.

[70] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[71] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[72] Vladimir Vapnik. *The nature of statistical learning theory*. springer, 2000.

[73] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Computer Vision–ECCV*, pages 428–441. Springer, 2006.

[74] Jean-Yves Bouguet. Camera calibration toolbox for matlab. 2004.

[75] Jo Yung Wong. *Theory of ground vehicles*. Wiley. com, 2001.

[76] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1409–1422, 2012.