國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

使用單目相機對已知高度物體深度預測

Depth Estimation of Objects with Known Heights using a Monocular Camera

黃冠仁

Kuan-Jen Huang

指導教授：周俊廷 博士

Advisor: Chun-Ting Chou, Ph.D.

中華民國 112 年 1 月

January 2023

doi:10.6342/NTU202300097

誌謝

非常感謝在一路上幫助過我的人。謝謝周俊廷老師兩年來的指導，總是教導我們如何用清楚的邏輯與例子來讓聽眾理解我們想傳達的事物，雖然不一定能完全達到老師的要求，但在未來也會不斷提醒自己這些老師教導的原則，也感謝逄愛君老師、魏宏宇老師、莊永裕老師撥空擔任我的口試委員。

Roy 學長：謝謝您能以跟一般學生較不同的角度跟我們分享經歷，祝您身體健康。

記綱學長：謝謝學長在實習的時候給予我很多研究上的建議，課業上也給很多幫助。

季勳學姊：雖然學姊總是很逗，但事實上很聰明，很佩服你的理解力和做事效率，謝謝前實驗室的老大。

感謝同學們都是很好相處的人，有困難時大家都會互相幫忙，能跟各位在同個實驗室真是非常幸運！

里昂：很佩服你常常做事過程很驚險但結果卻意外地好，我想這就是聰明吧，謝謝你是實驗室少數懂迷因的人，祝你實況之路順利。

正鈞：謝謝你常常能在研究上給我好的建議，也跟我討論很多報告上的問題，尤其要感謝你的大學人脈還有專業知識的幫助。

彥筑：感謝修課時都會借我筆記，做報告時還會給予很好的建議，去行政大樓時也會借我腳踏車，還有你的理解和表達能力真的很厲害。

丞傑、皓宇、霽原：雖然學弟們剛入學就慘遭疫情無情遠距，相處的時間很少，感謝你們進實驗室幫忙分擔職務，特別感謝丞傑常常能展現數學能力幫我理解看不懂的論文與研究上給我建議與幫忙。

莫庭安：感謝這半年來的相處，給予我心靈上的支持與陪伴，能在我空閒時和我探索各種景點，幸好有認識你。

感謝來到陌生台北這兩年的新舊朋友、利用網路常常聯繫的朋友們，沒有你們的陪伴，這短暫的兩年多可能會變得很平淡乏味，我也無法好好認識這個城市，謝謝你們豐富了我兩年的生活，特別感謝盧皓宇常給予我英文上的建議與幫助我排解日常的壓力。最後要感謝我的家人提供我很棒的環境讓我能無後顧之憂地專心唸書，我也會繼續努力，希望未來可以成為家人的依靠，不辜負你們的期待。

doi:10.6342/NTU202300097

中文摘要

距離預測技術在未來智慧城市中扮演重要的角色，透過預測物體的距離，可以發展出各式各樣的應用，包含自動駕駛系統、車輛定位和街景圖資的更新等，在這些應用中，距離預測都是不可或缺的技術。在現有的距離預測技術中，有一類方法採用感測器如無線電雷達(Radar)或是光學雷達(Lidar)，藉由對周圍物體發出光或電磁波來預測距離，這類方法雖然精準且快但是成本較為昂貴。另種方法則是使用影像等低成本的方式，再經由強大的演算法來預測出影像中物體的距離。

在本篇論文中，我們嘗試透過單一鏡頭相機(Monocular Camera)來估計物體距離。此類問題可被細分為兩個種類包含地面上物體和相機的距離，以及非地面物體和相機的距離，像是預測紅綠燈的距離。大多數現有的研究專注在預測地面上物體的距離且這些研究的結果都很準確與穩定。而第二類的問題通常需要額外的資訊才能被解決，因此我們提出使用目標物體的真實高度與相機成像模型(Camera Imaging Model)進而推導出物體在影像中的座標與此物體對應之地面點座標的關係。透過這個關係，我們可以推出任何已知高度的物體之地面座標，進而利用現有處理第一類問題的演算法來預測此地面物體與相機的距離。

透過本文所提出的演算法，不但可以用低成本且即時的方式來預測出單一影像中已知高度的物體之距離，此方法更具有資料獨立性，也就是效能不受使用的資料不同而影響。實驗結果顯示我們的方法在三個不同環境中的整體平均絕對百分比誤差(Mean Absolute Percentage Error)為 8%，在估計非地面物體之距離比基於學習(Learning-based)的模型好 30%。但是我們的方法無法使用在預測低高度物體上，且需要一個額外的物件偵測模型來獲得物體在影像中的位置。

關鍵字：單一影像距離預測，相機幾何，視角轉換

doi:10.6342/NTU202300097

ABSTRACT

Distance estimation technology is an essential component of future smart cities. It can be used to enable a variety of applications such as autonomous driving systems, and map updating. There are various methods for distance estimation, including using active sensors like radar and lidar, which emit waves or light to measure the distance to surrounding objects. These methods are generally accurate and fast, but they can also be expensive. Alternatively, low-cost techniques such as using images can be combined with powerful algorithms to estimate the distance of objects in the image.

In this thesis, we try to estimate the distance of objects through a monocular camera. This problem can be divided into two categories. One is estimation of distance between the objects on the ground and the camera. The other is the estimation of distance between the objects above the ground and the camera, such as traffic lights. Most of the existing researches focus on the former and the estimation is quite accurate and stable. Solving the latter problem usually requires additional information such as the object size. Our method utilizes the height of the target object and camera imaging model to derive the relationship between the coordinate of an object in the image plane and its projection point on the ground. By this relationship, we can calculate the coordinates of the object's projection point on the ground. Then, the existing algorithm that is used in the first type of problems is adopted to estimate the distance of the projection point.

By using our proposed method, the distance of an object with a known height in the monocular image can be predicted in a low-cost and real-time way. Moreover, our method is data-independent. The experimental results show that the overall mean absolute percentage error of our proposed method in three different environments is 8%, which outperforms the learning-based model by 30%. However, our method cannot be used to

doi:10.6342/NTU202300097

estimate the distance of objects with a low height and needs an additional object detection model to obtain the position of an object in the image.

Keywords: monocular distance estimation, camera geometry, perspective transform

doi:10.6342/NTU202300097

CONTENTS

doi:10.6342/NTU202300097

doi:10.6342/NTU202300097

LIST OF FIGURES

doi:10.6342/NTU202300097

doi:10.6342/NTU202300097

doi:10.6342/NTU202300097

LIST OF TABLES

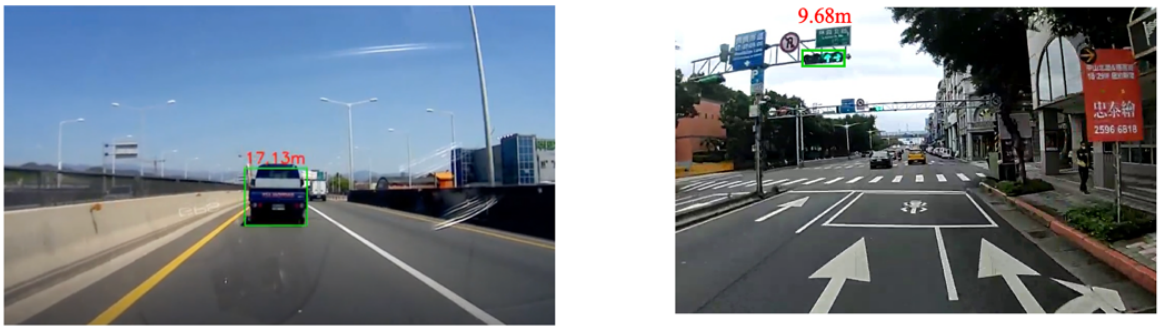
doi:10.6342/NTU202300097

CHAPTER 1

INTRODUCTION

1.1 Motivation

With the rapid development of technology, semi-autonomous driving vehicles has entered our lives. A common intelligent vehicle system, such as advanced driver assistance systems (ADAS) [1], consists of the modules of path planning, lane departure warning, and collision avoidance. Distance estimation that determines the distance between an object and a vehicle is a crucial part of all these modules. A few applications are shown below. In **Figure 1(a)**, we can use the distance between the front vehicle and a vehicle to avoid collisions or give some driving advice to the driver. In **Figure 1(b)**, the distance between the traffic light and a vehicle can be used in the navigation system. It notifies the drivers that they should turn right or turn left at a distance of 9.68m. In **Figure 2**, we can use the estimated distance between the object and camera to know the corresponding position of this object in a map. In this way, the information such as the status of the traffic light can be updated in a map. Map with real-time information is an important component in the autonomous vehicle, because it provides precise object location and real-time information about the surrounding environment.



(a) (b)

Figure 1: Applications of distance estimation

doi:10.6342/NTU202300097

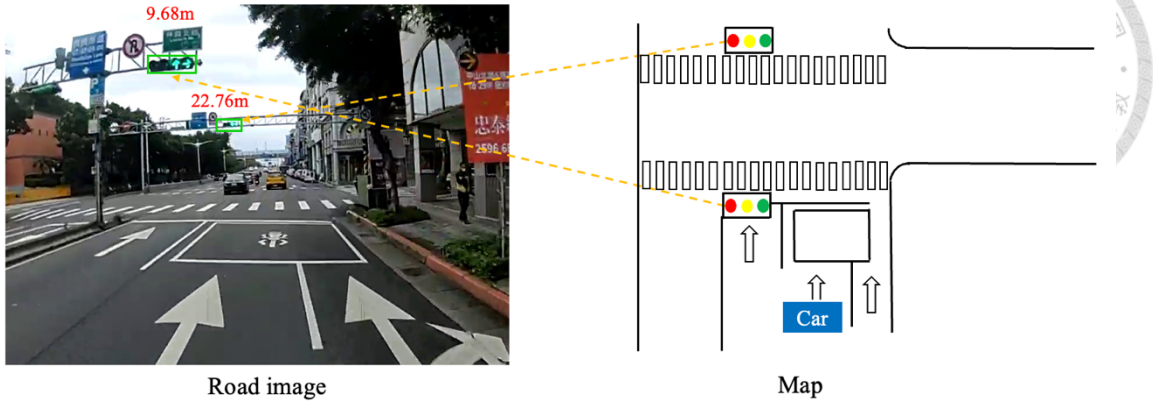


Figure 2: Schematic diagram of map matching

Image-based distance estimation can be divided into monocular and binocular distance estimation, as shown in **Figure 3**. Monocular distance estimation relies on one image captured by a single camera. On the other hand, binocular distance estimation requires two images, each captured by a different camera and relies on the disparity of visual angles of the two cameras. It is just like the parallax of the human eye. The comparison of these two types of distance estimation is shown in **Table 1**.

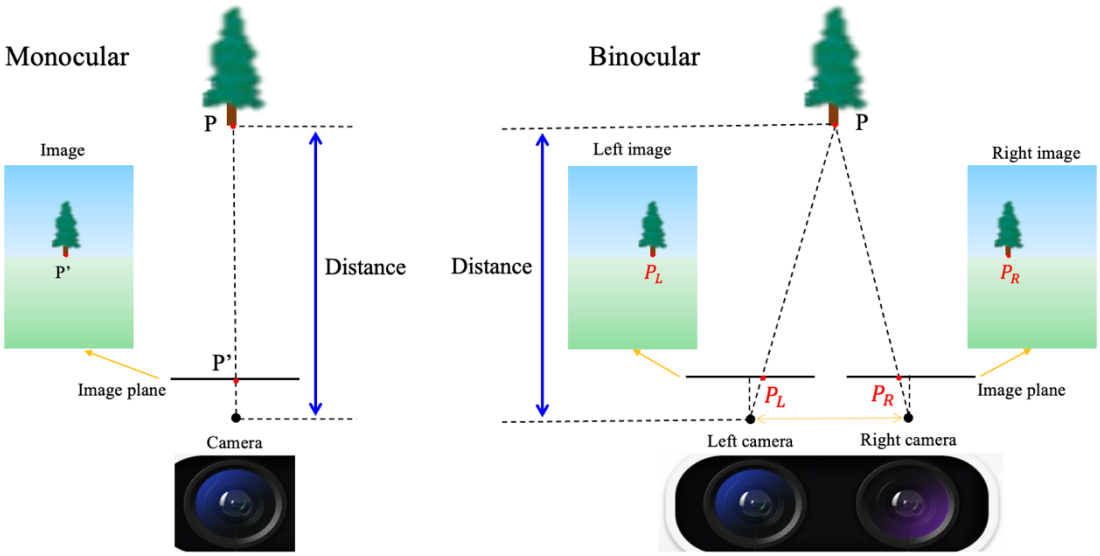


Figure 3: Monocular vs binocular distance estimation

doi:10.6342/NTU202300097

	Pros	Cons
Monocular distance estimation	• Low-cost • Light computation	• Low accuracy • Require prior object information
Binocular distance estimation	• High accuracy	• High computation • Complex feature matching

Table 1: Compare two types of distance estimation

The main advantage of monocular distance estimation is low-cost and light computation. However, it is generally less accurate compared to binocular distance estimation. Monocular distance estimation usually requires some prior object information. On the contrary, the binocular distance estimation has high accuracy, but it requires more computations. Usually, it needs to use GPU for processing complex feature matching. There have been a lot of research and discussions on binocular distance estimation in the field of autonomous vehicles, and they have achieved good results [2][3]. However, monocular distance estimation has not yet been fully developed.

In this thesis, we will focus on discussing monocular distance estimation. We divide the distance estimation into two categories, which are the distance from the camera to the objects on the ground and the distance from the camera to the objects above the ground. This is illustrated in **Figure 4**. As shown in figure, the xy-plane is the ground plane and the z-coordinate of a point represents the height of that point above the ground. There is a tree with a height of h in the real world, and the top of the tree is marked as point P2. Point P1 is the P2's projection point on the ground. Point C is the camera's projection point on the ground. The distance d from Point P1 to Point C can be decomposed into dx along the x-axis and dy along the y-axis. In this thesis, we define the distance between the object above the ground (P2) and the camera as dy, and we will focus on estimating dy.

doi:10.6342/NTU202300097

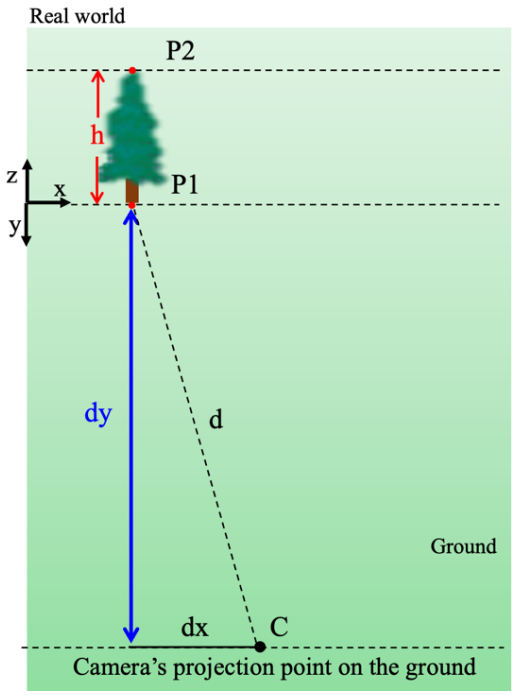


Figure 4: The distance we focus on in this thesis

To estimate the distance between an object above the ground and a camera, it is necessary to determine the object's projection point on the ground. However, determining the object's projection point on the ground can be challenging. As an example, the diagram in **Figure 5** illustrates a side view of **Figure 4** along the x-axis. P2 is an object above the ground in the real world and its projection on the image plane is located at P2'. P1 is the P2's projection point on the ground in the real world, and its projection on the image plane is at P1'. Without additional prior information about object P2, it is uncertain to determine which position in the real world corresponds to the projection of P2' in the image. Each point on the yellow line in **Figure 5** is the possible projected position of P2' The projection point on the ground for each point is different from P1. Therefore, by providing information such as the object's height, we can determine which position in the real world corresponds to the projection of P2' in the image.

doi:10.6342/NTU202300097

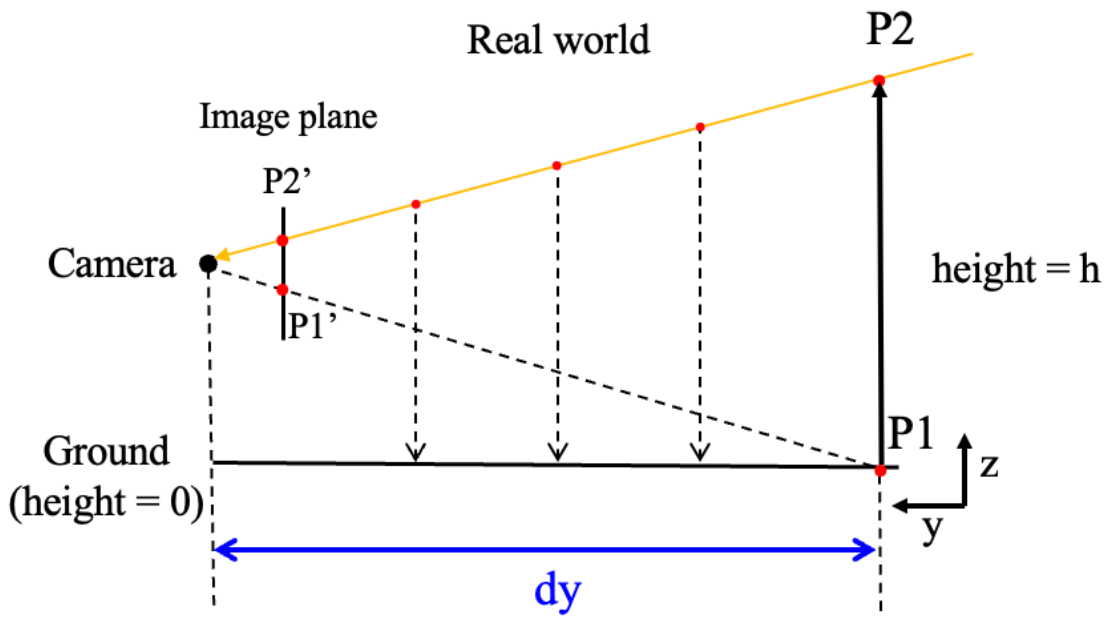


Figure 5: Difficulty of finding the obejct's projection point on the ground

1.2 Related Work

Monocular distance estimation utilizes a single camera to capture one image and then estimates the distance. Currently, the methods of monocular distance estimation are divided into sensor-based method, learning-based method, and geometry-based method. In the traditional distance estimation, the pinhole imaging model of the camera is mainly used.

1.2.1 Sensor-based Method

The use of active sensors such as Radio Detection and Ranging (Radars) and Light Detection and Ranging (Lidars) is a common choice for measuring the distance between the surrounding objects and the sensors. Radars send out electromagnetic waves which may be partly reflected back to the Radars by objects, as shown in **Figure 6**. They can

doi:10.6342/NTU202300097

sense the range up to 150 meters but they usually have low resolution [4]. Lidars emit light pulse into the surrounding environment. The light pulses are reflected by surrounding objects and return to the sensors. The sensors can use the time it took for each pulse to return to the sensors to compute the distance each pulse traveled, as shown in **Figure 7** [5]. Lidars have the higher spatial resolution but they are too expensive [6]. Real-time operation is the major advantage of using these active sensors. Although the sensor technology is improved and the mass production is expected to lower the cost of lidars [6], they still have limitations of performance in bad weather conditions such as snow, fog and rain.

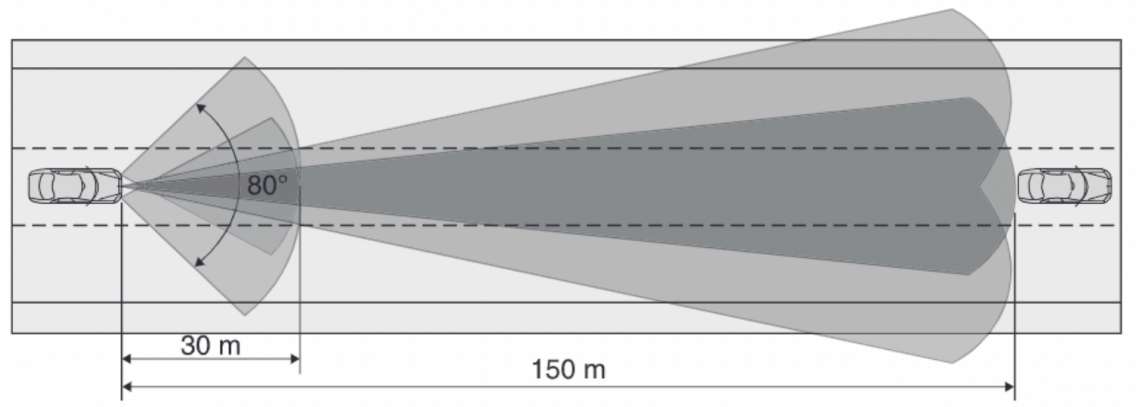


Figure 6: Combination of long-range radar and short-range radar [4]

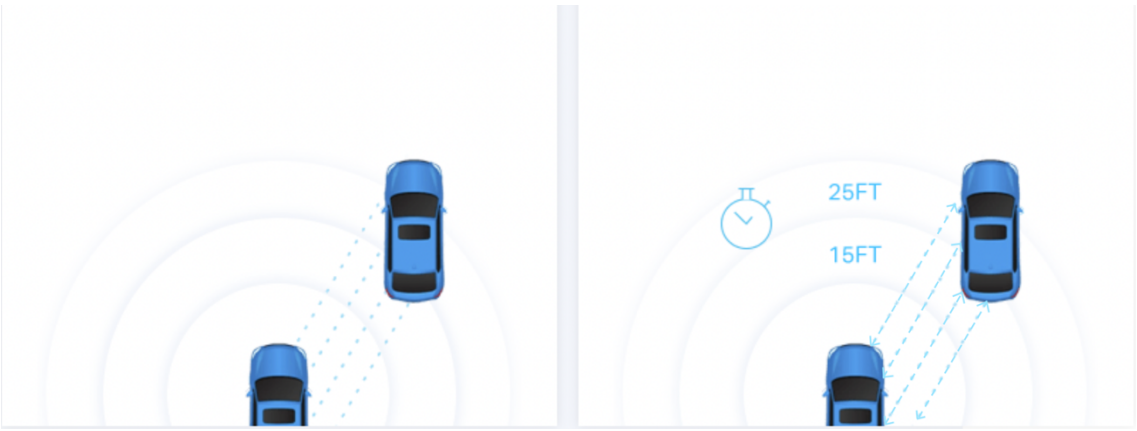


Figure 7: Lidar sensor computes the distance by the reflected pulse [5]

doi:10.6342/NTU202300097

1.2.2 Learning-based Method

This type of method is also called the data-driven approach. Despite limited success, data-driven approaches for depth estimation do not acquire significant attention until the rise of deep learning. Deep learning-based depth estimation makes a huge improvement in the field of monocular depth prediction. This is because deep learning can perform better in image processing than other machine learning methods. Therefore, a significant amount of research has emerged over a short period of time. These methods can be roughly divided into two categories which are supervised learning and self-supervised learning. Supervised learning takes a single image and depth data which are measured by ranging sensors, such as Radars or Lidars, as ground truth for training. In [7], the authors use the pixel-wise understanding to predict depth. They divide a single RGB image into four separate object groups that cover the entire scene. Then, they individually estimate the depth of each group. Four groups of objects are subsequently passed as input to a depth generator network. Then, they fuse the separated depth prediction parts to obtain the final result. BTS [8] proposed a novel network architecture of adding local planar guidance layers in the decoding phase. In other normal encoder-decoder network architectures, the loss function usually works in the final stage of decoding to optimize the depth prediction result. However, in [8], the use of local planar guidance layers can impose constraints on the process of the decoding phase. Thus, the depth prediction results will be improved. The experiments show that their method outperforms the state-of-the-art works, especially at the margin of the objects.

Self-supervised learning utilizes a vast amount of unlabeled data without the need for ground truth depth information. Instead, it uses the inherent structure or characteristics of the data to automatically label the data by existing methods. With the labeled data, self-supervised learning can train these data like supervised learning. [9] and [10] propose

doi:10.6342/NTU202300097

self-supervised learning methods that transform the problem from depth prediction to image reconstruction. Using rectified stereo image pairs, they generate the disparity map. Their networks attempt to reconstruct one view from the other through the estimated disparities and define the reconstruction loss as the primary training objective. The learning process only requires rectified stereo image pairs, without the need for ground truth depth data associated with the corresponding RGB images. This significantly reduces the effort required to gather a dataset for new types of environments. However, the accuracy does not outperform that of the current top supervised learning methods.

1.2.3 Geometry-based Method

In the monocular geometry-based approach, most existing researches focus on estimating the distance between the front vehicle and the camera. However, the distance between the objects above the ground and the camera, such as traffic lights, is rarely concerned. Compared with Sensor-based method, geometry-based method estimates the distance without expensive instruments, so it is a low-cost method. However, it demands intelligent post-processing. In [11] and [12], the authors first compute the y-coordinate of the vanishing point in the image plane by the intrinsic and extrinsic parameter matrix of the camera. Then, the position of the target vehicle in the image is detected by vehicle detection models. Finally, they use a simple pinhole imaging model for estimating the distance between the target vehicle and the camera, as shown in **Figure 8** [11]. It consists of a camera with a focal length and an image plane. H is the height of the camera that the author measures in advance. The target vehicle is at a distance Z from the camera. Z is the distance we want to know. P_{tg} is the target vehicle's projection point on the ground in the real world and its projection on the image plane is at P_{tg}'. The difference of y-coordinate between P_{tg}' and the vanishing point in the image is denoted as Δy. The

doi:10.6342/NTU202300097

distance between the camera and the target vehicle can be computed by similar triangles. That is, the ratio of Δy to focal length will be equal to the ratio of H to Z. For example, if the focal length is 300 pixels, Δy is 30 pixels and the camera height is 1m, then the distance of target vehicle Z computed by similar triangles is 10m. The method in [13] estimates the distance from the front vehicle to the camera in a way similar to [11] and [12]. The difference is that [13] takes the pose information of the vehicle into consideration by equipping an inertial measurement unit. Therefore, [13] can use the collected pose angles to reduce the error caused by the bump during driving.

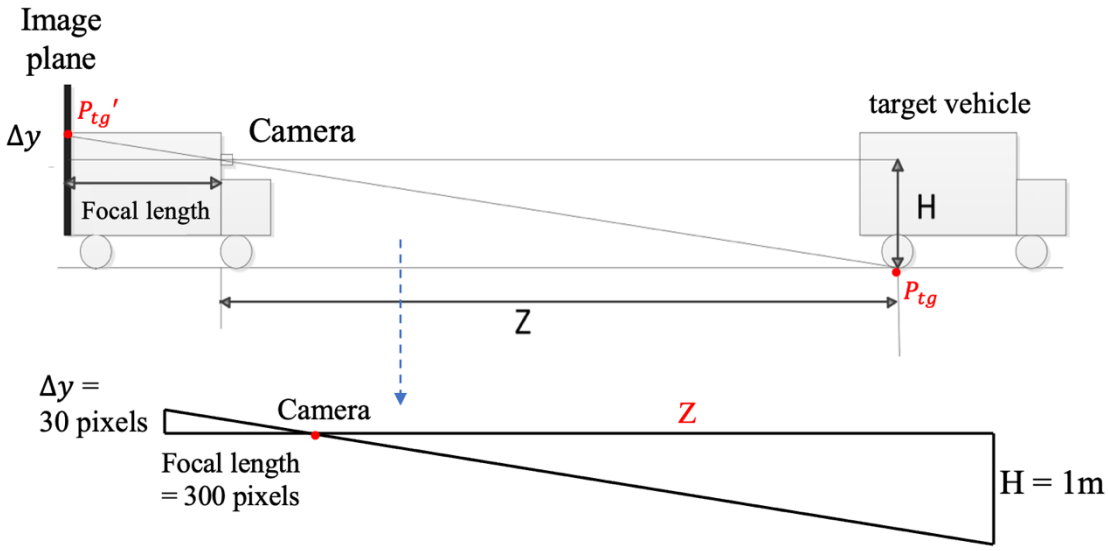


Figure 8: The pinhole imaging model

These papers estimate the distance between the front vehicle and the camera. Besides, there are also papers that propose the method of estimating the distance from the object above the ground to the onboard camera. In [14], the authors try to estimate the distance between the traffic light and the camera. The main concept behind this method is the representation of the real world as a stack of multiple layers of images. Each layer of images consists of pixels with different scales and at varying distances. The camera captures these layers of images and projects them onto the image plane, as shown in

doi:10.6342/NTU202300097

Figure 9(b). Additionally, the authors use the real width of the traffic light and the width of the traffic light in the image to get the resolution at the layer where the traffic light locates. The camera's field of view (FOV) is employed to determine the horizontal angular extent of the image captured, quantified in degrees, as shown in **Figure 9(a)**.

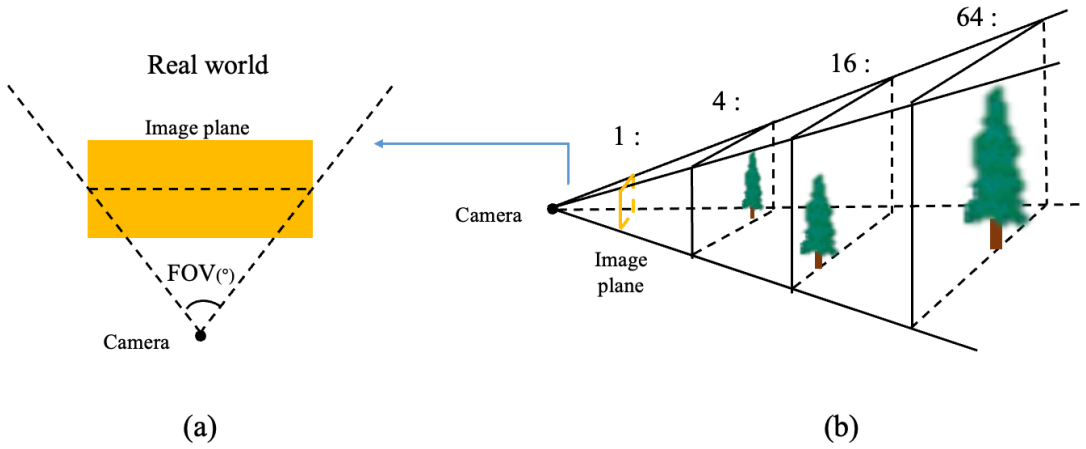


Figure 9: The real world is comprised of many layers of images

An example of this method is shown in **Figure 10**. The distance between the traffic light and the camera can be calculated through a tangent. First, we have to know the pixel resolution of the layer where the traffic light locates. The real width of the traffic light is 0.3 meters and the width of the traffic light in the image is 20 pixels. Thus, the pixel resolution is obtained as $\frac{0.3}{20}$ = 0.015 (meter/pixel) which represents that a pixel is 0.015 meters in the real world. Second, by multiplying the width of the half image and the pixel resolution, we can obtain the width of the half image in the real world. That is $\frac{1280}{2}$ * 0.015 = 9.6 meters, and it is the width of the red horizontal dotted line in **Figure 10**. Finally, the distance from the traffic light to the camera can be determined by applying the tangent function with an angle of 40° on the triangle shown in the bottom right corner of **Figure 10**. The distance d is $\frac{9.6}{\tan(40°)}$ = 11.428 meters.

doi:10.6342/NTU202300097

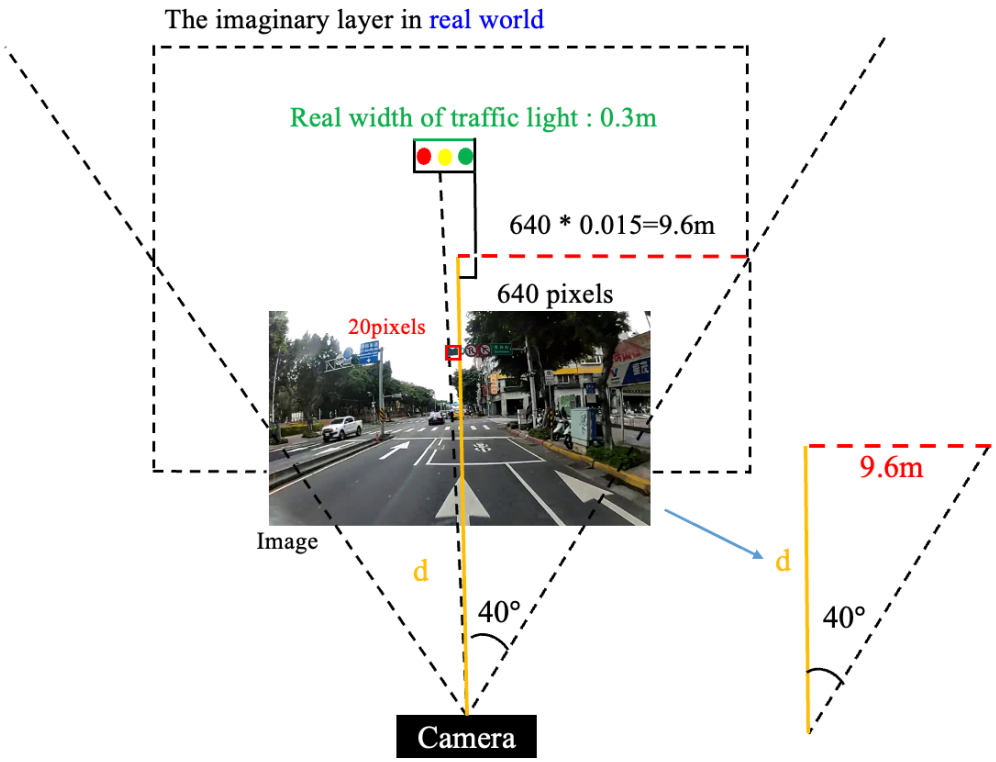


Figure 10: An example of the traffic light distance estimation

1.2.4 Summary of the Related Work

Although the sensor-based methods have higher accurate depth estimation and require minimal post-processing, the implementation cost is too high. On the other hand, the learning-based methods and geometry-based methods provide low-cost solutions. Learning-based methods require a large dataset to train the model. They almost have poor performance in the environment not contained in the training dataset. Geometry-based approaches are data-independent and usually have less computation. They need some prior information, such as focal length of the camera or some known distance reference. The method proposed in this thesis is included in this category. A comparison of these types of methods is shown in **Table 2**.

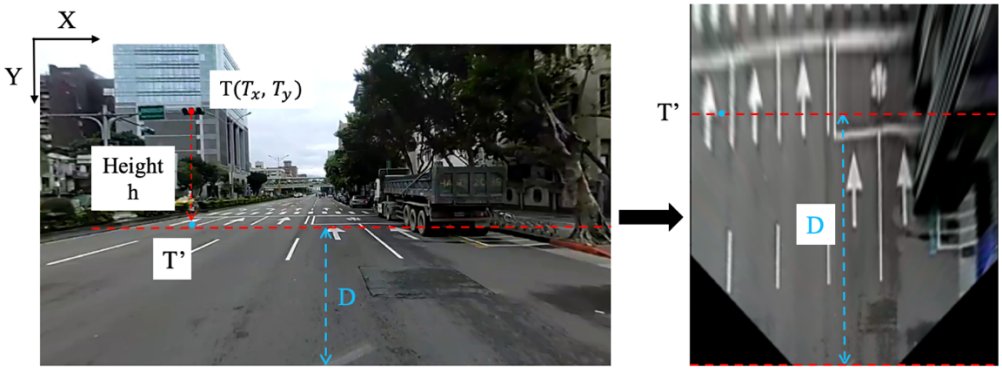
doi:10.6342/NTU202300097

	Cost	Computation	Property
Sensor-based	High	Low	High accuracy, very expensive
Learning-based	Low	High	Accurate, require large dataset
Geometry-based	Low	Low	require prior object information

Table 2: Comparison of three types of methods

1.3 Problem Statement

We have an image frame captured by a monocular camera and get the coordinate of the target object with known height in this image. The coordinate of the target object in the image frame is T(T_x, T_y) with a known height h. Then, we output the longitudinal distance D of this target object. The schematic diagram is shown in **Figure 11(a)**. The longitudinal distance D of target object is defined below. First, we find the ground point T' of the target object in the image. Second, we transform the image into a bird-eye image and find the corresponding position of the ground point T', as shown in **Figure 11(b)**. Finally, the y-coordinate difference between T' and the bottom edge of the bird-eye image is the longitudinal distance D.



(a) Image frame (b) Bird-eye image

Figure 11: The definition of longitudinal distance

doi:10.6342/NTU202300097

1.4 Contributions

This thesis proposes a low-cost, real-time solution using a monocular camera. The proposed method is different from other existing works that just focus on the distance between the objects on the ground and camera or the distance between the front vehicle and the camera. Our proposed method mainly solves the distance between the objects above the ground and the camera. The performance of estimating the distance of objects with a known height in three different environments is stable. Moreover, the overall mean absolute percentage error (MAPE) is 8%. Besides, the performance of our proposed method outperforms that of the learning-based model by 30% in three different environments. From the experimental results, our proposed method is data-independent. In other words, the performance is not influenced by the different data. Furthermore, even though the exact height of the target object is unknown, our method can still be used in the road scene of Taiwan with acceptable performance.

1.5 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we describe the system settings of this thesis. In Chapter 3, we introduce our proposed algorithm. In Chapter 4, we demonstrate the performance evaluation and analyze the experimental results. Finally, in Chapter 5, we conclude this thesis and discuss our future work.

doi:10.6342/NTU202300097

CHAPTER 2

SYSTEM SETTINGS AND ASSUMPTIONS

In this chapter, we introduce the system settings and give a brief overview of the proposed method. The input data are described in Section 2.1. An overview of the proposed solution is illustrated in Section 2.2.

2.1 Input Data of Driving video

In our system, driving videos are captured by a monocular onboard camera with a resolution of 1280*720 at a frame rate of 24 fps. The frames of a video are denoted as V_f (f = 1, 2, …, F), where F is the total number of frames in a video. An example of frames in the urban environment is shown in **Figure 12**.

Figure 12: A frame captured in the urban area

doi:10.6342/NTU202300097

2.2 Key Ideas of the Proposed Method

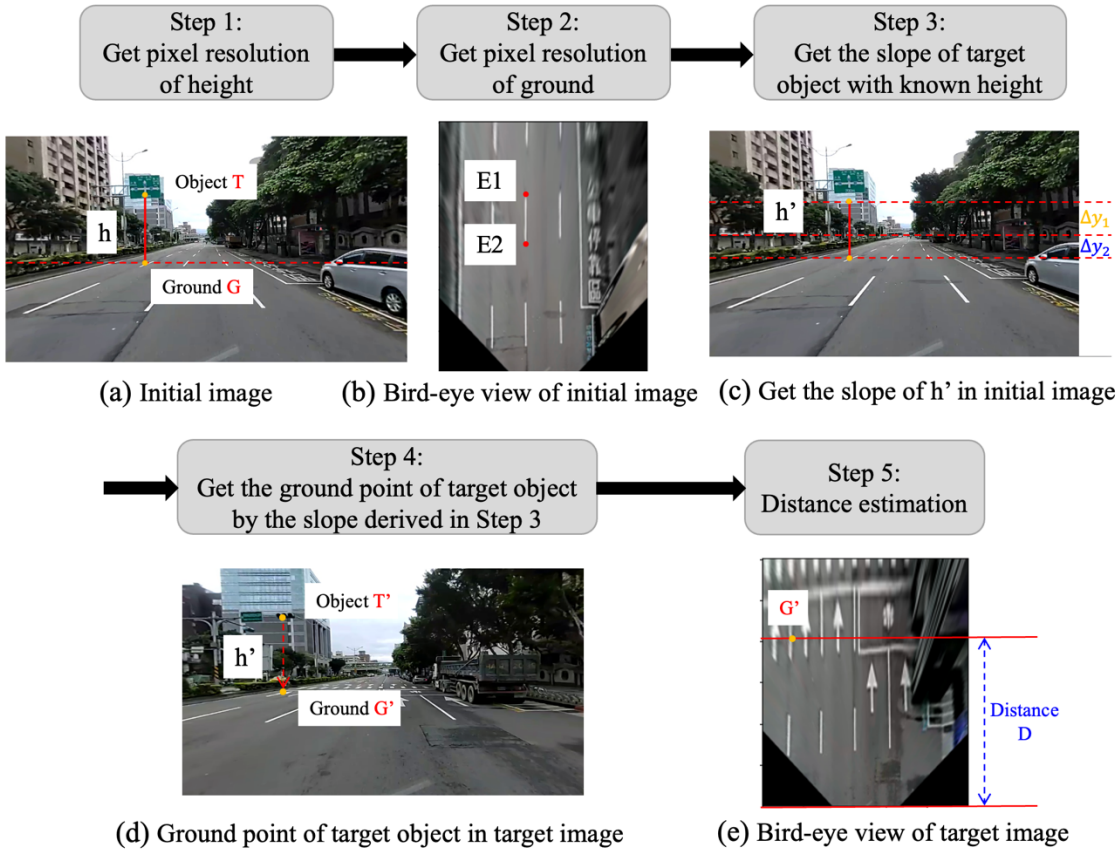


Figure 13: Key ideas of the proposed method

In this section, we briefly explain the key ideas of our proposed method. The key idea of our proposed method is made up of five steps, as shown in **Figure 13**. In Step 1, an initial image which contains an object with known height is used. An example is shown in **Figure 13(a)**. The known height of this object is denoted as h. Besides, we label the coordinate of the object T and the coordinate of its ground point G. Then, we obtain the pixel resolution of height by the ratio of h and the y-coordinate difference between T and G.

In Step 2, we transform the initial image into bird-eye image. According to the traffic rules in Taiwan [15], the white dashed lines on the ground is formulated as four meters.

doi:10.6342/NTU202300097

Thus, we find the endpoints E1 and E2 of a white dashed line in the bird-eye image, as shown in **Figure 13(b)**. The pixel resolution of ground is obtained by the ratio of four meters and the y-coordinate difference between E1 and E2.

In Step 3, we define the relationship between the coordinate of an object with known height and the coordinate of its ground point in the image as the "slope". If we want to estimate the ground point of a target object with height h', the slope of this target object will be derived first and the slope is only related to height. We derive the slope of h' in the initial image by finding Δy_1 and Δy_2, as shown in **Figure 13(c)**. Δy_1 and Δy_2 are two y-coordinate differences in the image and they will be introduced in Section 3.3.

In Step 4, when a deep-learning model is used to detect the target object in the target image during driving on the road, the coordinate of target object can be obtained. Therefore, we use the slope of h' derived in Step 3 to estimate the ground point of target object, as shown in **Figure 13(d)**. The coordinate of target object is T' with height of h' and its ground point is estimated at G'.

In Step 5, we transform the target image into bird-eye image. The corresponding position of ground point G' in the target image is found in bird-eye image, as shown in **Figure 13(e)**. Finally, we use the y-coordinate difference between G' and the bottom edge of the bird-eye image, and the pixel resolution of ground obtained in Step 2 to compute the distance D.

doi:10.6342/NTU202300097

CHAPTER 3

PROPOSED ALGORITHM

In this chapter, we first present the pipeline of the proposed method. Then we introduce the procedure of data preprocessing. Finally, the details of the proposed method are introduced from Section 3.3 to Section 3.5.

3.1 Pipeline of the Proposed Solutions

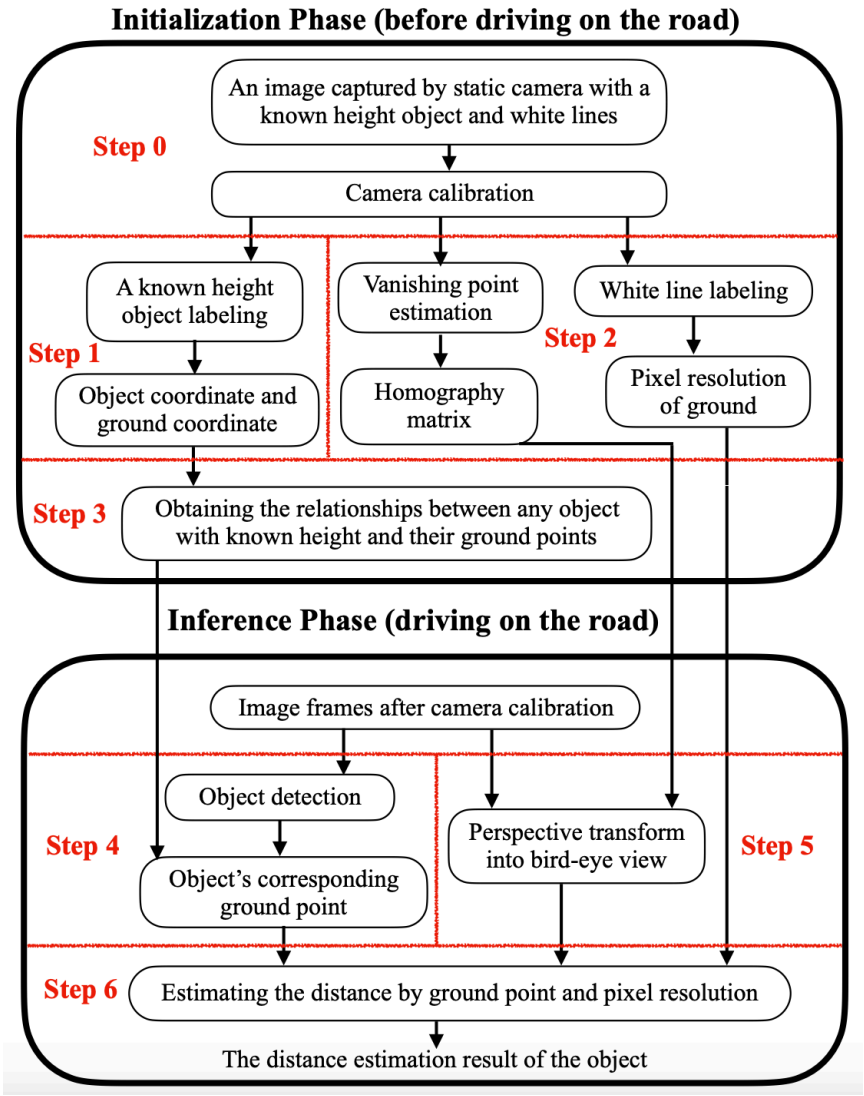


Figure 14: Pipeline of the proposed solutions

doi:10.6342/NTU202300097

The overall procedure of the proposed solutions is shown in **Figure 14**. We divide the system into two phases, which are the initialization phase and the inference phase. The initialization phase is before driving on the road; the inference phase is during driving on the road. First of all, in Step 0, we select an image captured by a static monocular camera. That is to say, the vehicle is not moving. Then, we calibrate this image so that the fisheye distortion disappears. Furthermore, the necessary conditions for this image are that this image contains an object with a known height and a white dashed line with a known length for reference. In Step 1, we label the coordinate of the object with a known height and the coordinate of its ground point in the image. In Step 2, the position of the vanishing point is estimated via parallel white dashed lines on the ground. With the vanishing point, we can define the region of interest in the image. Then, we compute a homography matrix by this region of interest. The homography matrix can be used to transform the image into a bird-eye image. Moreover, we label the endpoints of these white dashed lines to calculate the pixel resolution of ground. In Step 3, we obtain the relationship between the coordinate of any object with a known height and the coordinate of their ground points. This relationship can be used to estimate the ground point of any object with a known height in the inference phase.

After the initialization process mentioned above, the vehicle is driving on the road during the inference phase. In Step 4, we use the image frames captured by camera and these images have finished the camera calibration. As long as an object with a known height is detected in any image frames, we can use the relationship derived in the initialization phase to find the corresponding ground point of this object. In Step 5, we use the homography matrix obtained in the initialization phase to generate the bird-eye image of this frame. Eventually, in Step 6, we use the coordinate of ground point in the bird-eye image and the pixel resolution calculated in the initialization phase. The

doi:10.6342/NTU202300097

longitudinal distance of this object will be estimated. The amount of computation in the inference phase is composed of simple mathematic. Therefore, the distance estimation is able to accomplish in real-time.

3.2 Step 1: Data Preprocessing - Camera Calibration

The method to calibrate the distorted images was proposed by Zhengyou Zhang [16]. It is a famous technique to correct the fish-eye distortion in computer vision. Thus, we are able to accomplish camera calibration by utilizing existing libraries. First of all, we introduce the process that a camera captures the environment information into an image. The coordinates of the process consist of a world coordinate system, a camera coordinate system, and an image coordinate system. Furthermore, the procedure of transformation is usually divided into two parts: extrinsic and intrinsic. The extrinsic matrix of a camera depends on its location and orientation. The intrinsic matrix of a camera depends on how it captures the images. Parameters such as focal length, aperture, resolution, etc, determine the intrinsic matrix of a camera model. A schematic diagram of the camera model is shown in **Figure 15**.

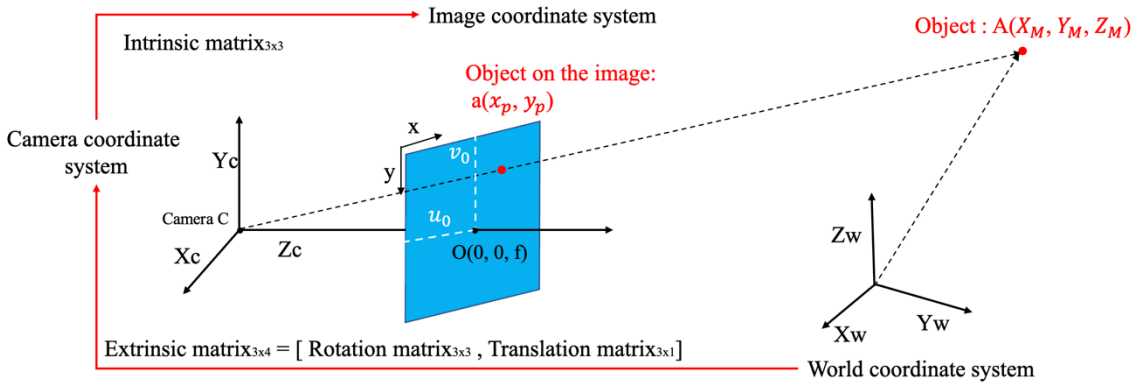


Figure 15: A schematic diagram of the intrinsic and extrinsic matrix

doi:10.6342/NTU202300097

We can see from the figure, there is an object A with coordinate (X_M, Y_M, Z_M) in the world coordinate system. The coordinate of object A is transformed into a camera coordinate system by an extrinsic matrix, which is composed of a three-dimensional rotation matrix and a translation matrix. Thus, object A is in the camera coordinate system that regard camera C as the origin. Then, object A can be transformed into an image coordinate system that regards the point O(0, 0, f) as the origin by an intrinsic matrix. Therefore, we can represent the transformation procedure as an equation:

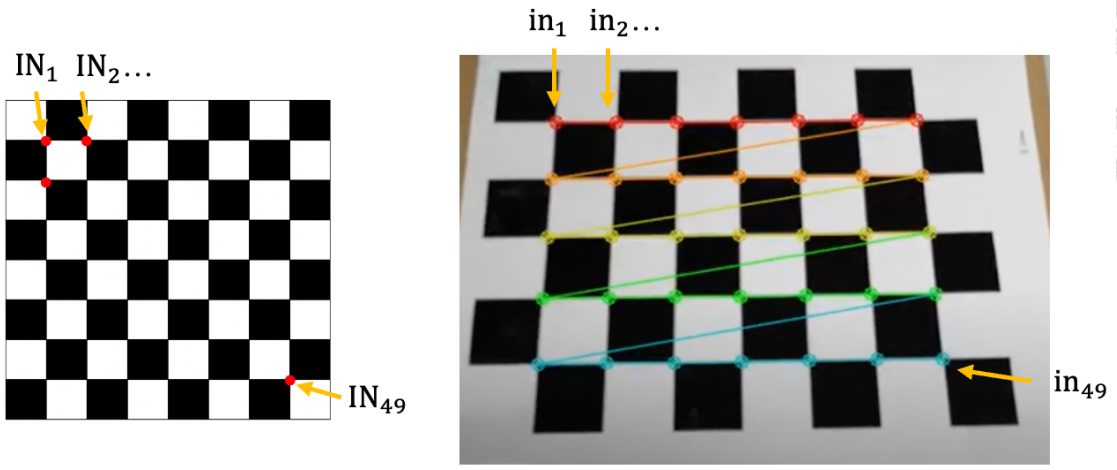
Image coordinate = Intrinsic matrix * Extrinsic matrix * world coordinate

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} r_x & \gamma & u_0 \\ 0 & r_y & v_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_4 & r_7 & t_1 \\ r_2 & r_5 & r_8 & t_2 \\ r_3 & r_6 & r_9 & t_3 \end{bmatrix} \begin{bmatrix} X_M \\ Y_M \\ Z_M \\ 1 \end{bmatrix} \quad (3.1)$$

In Equation (3.1), the notation x_p and y_p are the coordinate of object A in the image coordinate system. The notation r_x and r_y denote how many pixels per meter are in the x and y directions. The notation u_o and v_0 denote the x-offset and y-offset from the point O to the upper left corner of the image. The notation γ is the distortion coefficient. The notation r_1, r_2,…, r_9 and t_1, t_2, t_3 are the elements of the rotation matrix and translation matrix that is the extrinsic matrix. The notations X_M, Y_M, and Z_M are the coordinate of object A in the world coordinate.

After introducing the camera model, we introduce how to implement the camera calibration method proposed in [16]. Firstly, we prepare a chessboard with equal sizes of black and white squares, as shown in **Figure 16(a)**. The advantage of using the chessboard is that it is easy to produce and it can be conveniently marked as a coordinate on the intersections of the squares, as shown in **Figure 16(b)**.

doi:10.6342/NTU202300097



(a) World coordinate (b) Image coordinate

Figure 16: A chessboard with black and white squares

As we can see, there are 49 intersections in the chessboard. We give them the world coordinates which begin with the upper left intersection. They are denoted as $IN_1(0, 0, 0)$, $IN_2(1, 0, 0)$, …, $IN_{49}(6, 6, 0)$, as shown in **Figure 16(a)**. Moreover, we detect the position of these intersections in the image coordinate by existing libraries. They are denoted as in_1, in_2, …, in_{49}, as shown in **Figure 16(b)**. Now we have 49 pairs of points between the world coordinates and the image coordinates. We are ready to acquire the calibration matrix which is the intrinsic matrix mentioned above. However, we only take the intrinsic matrix into consideration because the extrinsic matrix which is composed of locations and orientations of the camera is difficult to be obtained in future applications. Therefore, the equation without considering the extrinsic matrix can be rewritten as below:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} r_x & \gamma & u_0 \\ 0 & r_y & v_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_M \\ Y_M \\ 1 \end{bmatrix} \tag{3.2}$$

where the x_p and y_p are replaced by the coordinate of in_1, in_2, …, in_{49} and the X_M and Y_M are replaced by the X-coordinate and Y-coordinate of IN_1, IN_2, …, IN_{49}, respectively.

doi:10.6342/NTU202300097

However, our purpose is to solve five unknowns of the intrinsic matrix, as r_x, γ, u_0, r_y, and v_o in Equation (3.2). Theoretically, we can only use five pairs of points to acquire the result, but the detection of intersections and the transformation of the coordinate may lead to some errors. For instance, a float number will round into an integer number in the image coordinate. Thus, the more pairs of points between the world coordinate and the image coordinate, the higher accuracy of the intrinsic matrix will be obtained. Once possessing the intrinsic matrix, we are able to calibrate the image captured by the camera of such specification. Consequently, the final step is to correct the fish-eye distortion. We can utilize the existing OpenCV library (Open Source Computer Vision) with function name "undistort" [18] to accomplish the camera calibration [16]. Two examples of camera calibration are shown in **Figure 17**.

(a) Original image with distortion (b) undistorted image of (a)

(c) Original image with distortion (d) undistorted image of (c)

Figure 17: Two examples of camera calibration

doi:10.6342/NTU202300097

The original images with distortion are shown in **Figure 17(a)** and **Figure 17(c)**. The results of camera calibration are displayed in **Figure 17(b)** and **Figure 17(d)**. As we can observe from the results, the distortion effect of the crosswalk and the footbridge is evident on the sides of the figures. After calibration, the distortion effect is removed.

3.3 Step 2: Estimate the Ground Point of Any Object with a Known Height

In this section, we will introduce how to estimate the ground point of any object with a known height in the image step by step. In Section 3.3.1, we present the projection of the coordinate from the real world to image plane. In Section 3.3.2, the difference of y-coordinate in the image is introduced. In Section 3.3.3, we derive the relationship between the object position and the difference of y-coordinate. Finally in Section 3.3.4, we describe how to estimate the ground point of any object with a known height in the image.

3.3.1 Step 2.1: Coordinate from Real World to Image Plane

In this subsection, we describe the projection of the coordinate in the real world to the coordinate on the image plane by a camera imaging model. The camera imaging model used here has a little different from the model mentioned in **Figure 15**. We directly assign the position of the camera as the origin of the whole coordinate system. In other words, there is no extrinsic matrix in this camera model. The camera imaging model describes the mathematical relationship between the coordinate of a point in the three-dimensional real world and the projection of this point on the image plane. It is an ideal camera that the camera aperture is seen as a point and no lenses are used to gather the light. Thus, this camera does not include the fish-eye distortion caused by the lenses and

doi:10.6342/NTU202300097

the finite size of the apertures. The camera imaging model can often be used as a rational explanation of how a camera depicts a three-dimensional scene. For instance, a schematic diagram of the camera imaging model is shown in **Figure 18**. The coordinate of the origin is at the position of the camera. The image plane is located at Z = f, where f is the focal length of the camera. It is supposed that there is a point A(X_0, Y_0, Z_0) on the road plane and its projection on the image plane is point a(x, y, f). Then, we can obtain the coordinate of point a through similar triangles as shown in **Figure 19**. From the figure, the ratio of y to Y_0 is equal to the ratio f to Z_0; the ratio of x to X_0 is equal to the ratio f to Z_0. Therefore, we can represent the coordinate of point a as (x, y, f), where x = $\frac{X_0}{Z_0}$f, y = $\frac{Y_0}{Z_0}$f. Afterwards, we will frequently use this camera imaging model to introduce our proposed method.

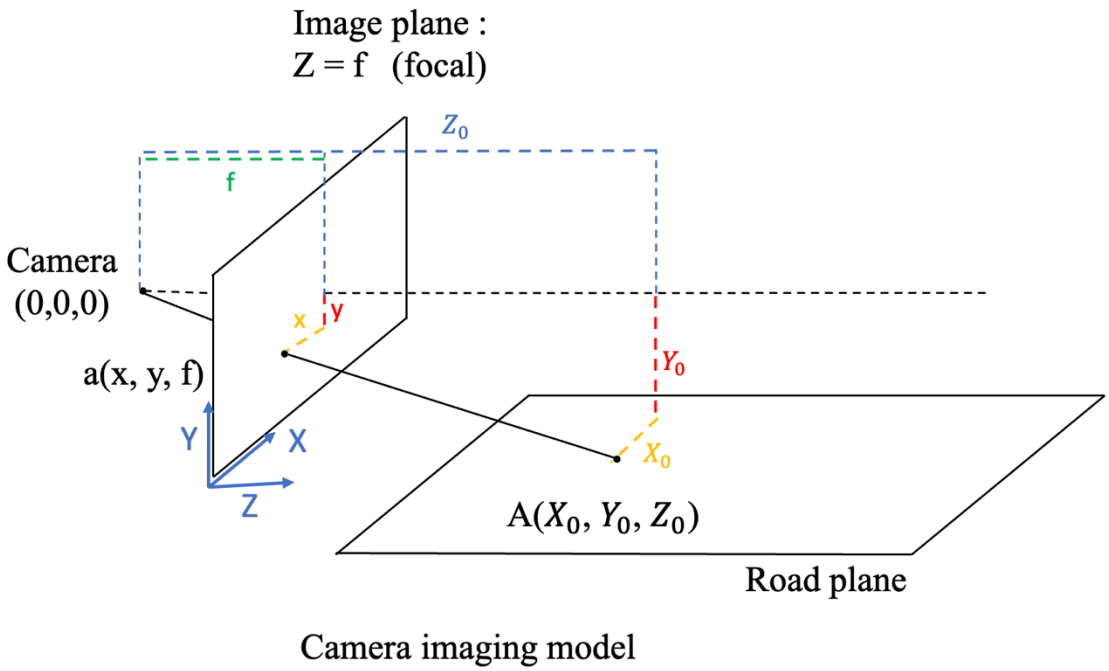


Figure 18: The schematic diagram of the camera imaging model

doi:10.6342/NTU202300097

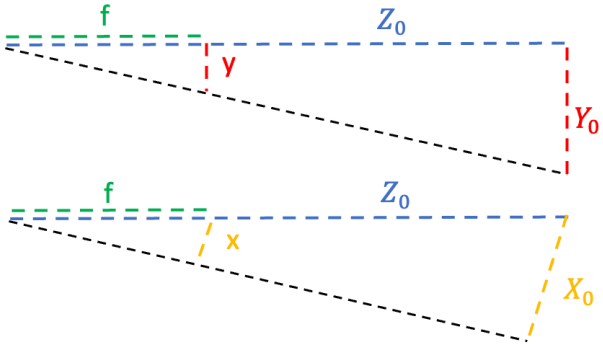


Figure 19: The similar triangles of the camera imaging model

As we can see from the camera imaging model, if a line is extended along the Z direction from the camera to infinity, this line will intersect the image plane at the point V(0, 0, f). The schematic diagram is shown in **Figure 20**. This concept will be used later in the introduction.

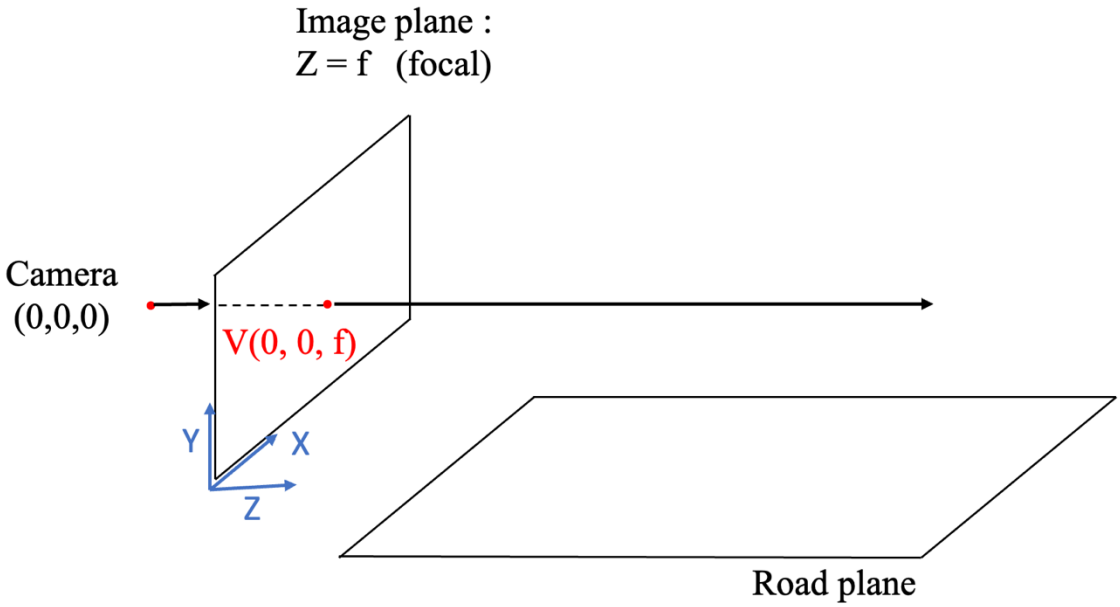


Figure 20: Extending a line from the camera to infinity

3.3.2 Step 2.2: The Difference of Y-coordinate in the image

In this subsection, we will define the difference of y-coordinate in the image and present how to calculate it. We directly use an example as shown in **Figure 21** to explain

doi:10.6342/NTU202300097

what is the difference of y-coordinate in the image. In this example, we assume that there is an object B(1, 3, 6) with its ground point BG(1, -1, 6) in the real world. The difference of Y-coordinate between object B and ground point BG is denoted as ΔY_B as the blue vertical line in **Figure 21**. The value of ΔY_B in this example is 4 in the world coordinates. Besides, we use the projection of the coordinate in the real world to the coordinate on the image plane as introduced in Section 3.3.1. Therefore, the projected point of B and BG in the image are point b and point bg, respectively. The projection of ΔY_B on the image plane is also obtained and is denoted as Δy_b as the red vertical line in **Figure 21**. Since we only consider the difference of y-coordinate, only the y-coordinate of the object and the ground point is calculated. Thus, the y-coordinate of point b in the image is $\frac{3}{6}$f; the y-coordinate of ground point bg in the image is $\frac{-1}{6}$f. Then, we can obtain Δy_b which is the difference of y-coordinate in the image. By directly subtracting the y-coordinate of b from the y-coordinate of bg, Δy_b is calculated as $\frac{4}{6}$f in the image coordinates. This process of obtaining Δy_b will be used in the next subsection.

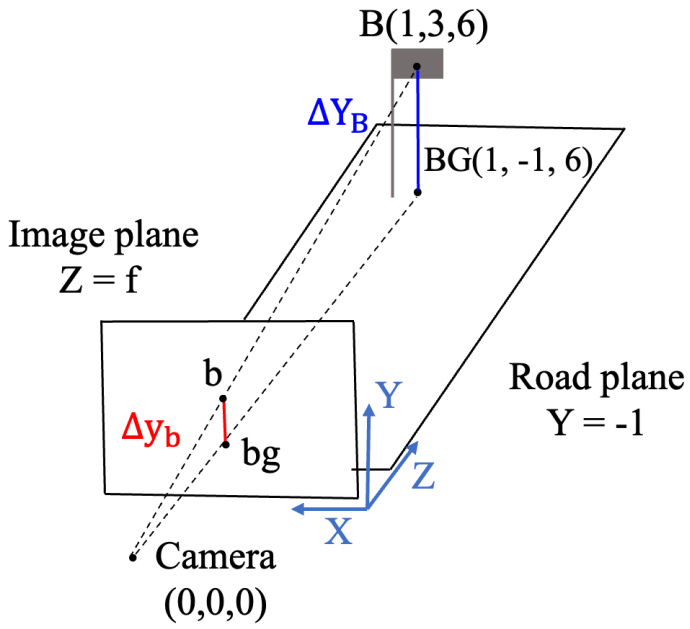


Figure 21: An exmaple of computing the difference of y-coordinate

doi:10.6342/NTU202300097

3.3.3 Step 2.3: Relationship between Object Position and Difference of Y-coordinate

In this subsection, we want to know the relationship between the object position and the difference of y-coordinate in the image. First, we utilize an example to illustrate the usage and the property of this relationship. Second, we deduce the general form of this relationship. We use coordinates to represent this example as shown in **Figure 22**. From this figure, the camera is the origin of the coordinate system and the image plane is located at Z = f, where f is the focal length of the camera. Two objects B(1, 3, 6) and E(-2, 3, 5) with the same height are given and the road plane is Y = -1. The ground points of these two objects are obtained as BG(1, -1, 6) and EG(-2, -1, 5) by projecting these objects to the ground. Then, we demonstrate the schematic diagram of these two objects in the real world as shown in **Figure 23**. They are two traffic lights with the same height as different distances. We suppose that these two objects are moving in the direction $-\vec{v}(0, 0, -1)$ towards the camera. It is just like a camera moving towards the fixed objects.

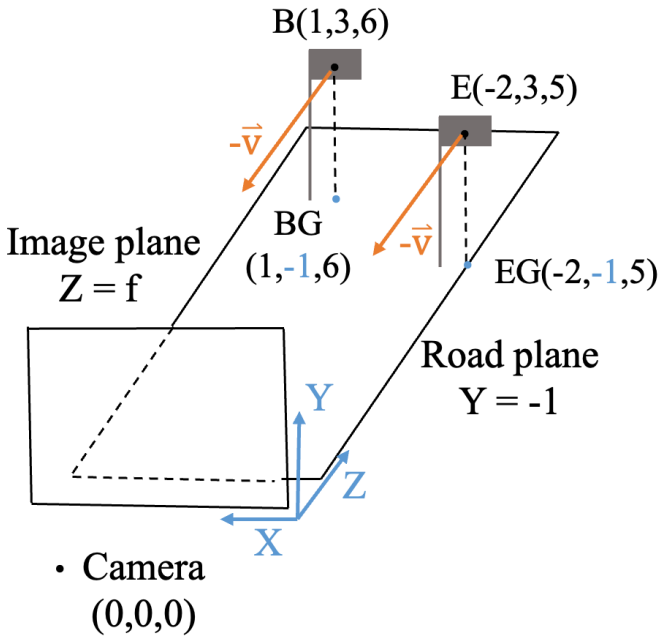


Figure 22: An example of two objects with the same height

doi:10.6342/NTU202300097

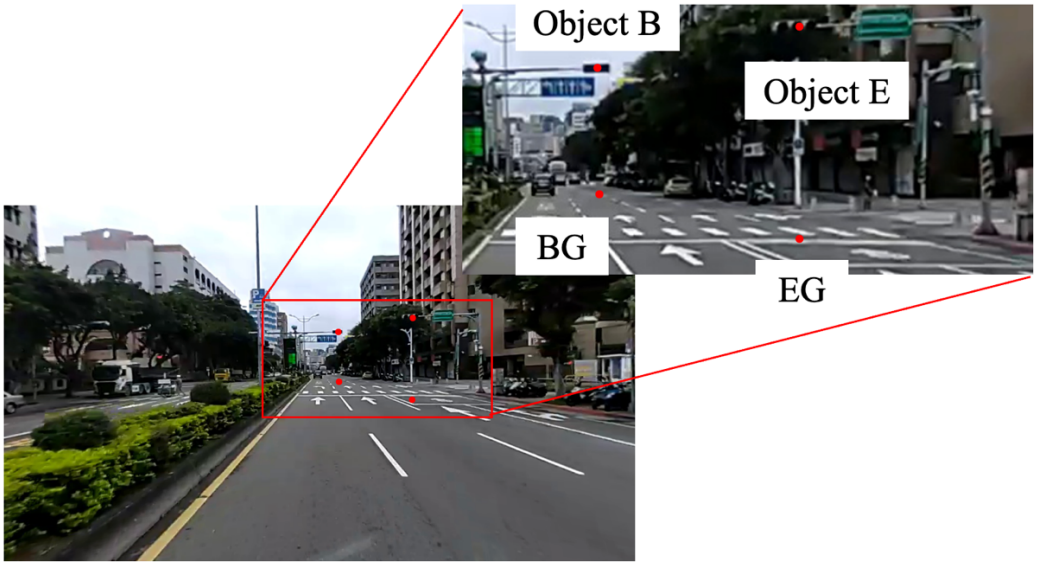


Figure 23: A schematic diagram of object B and E in the real world

Then, we display the geometric coordinate of object B at four different times to simulate the the camera moving forwards. The coordinates of object B in the real world from t = 0 to t = 3 are denoted as B_0 to B_3, and its ground point BG from t = 0 to t = 3 are denoted as BG_0 to BG_3, respectively. We demonstrate the scene of these four times by the coordinates as shown in **Figure 24**. As we can see from the figure, this object B moves a unit vector in the -z direction every time. Besides, the difference of y-coordinate between B and BG is 4 in the world coordinate. We demonstrate the schematic diagrams of four different times in the actual world in **Figure 25** for easier understanding. These images which contain object B and its ground point BG are continuously captured by a camera moving forwards. Then, we calculate the coordinate of these points in the image by the projection of the coordinate in the real world to the coordinate on the image plane as introduced in Section 3.3.1. Moreover, we want to find the relationship between the position of object B and the difference of y-coordinate in the image.

doi:10.6342/NTU202300097

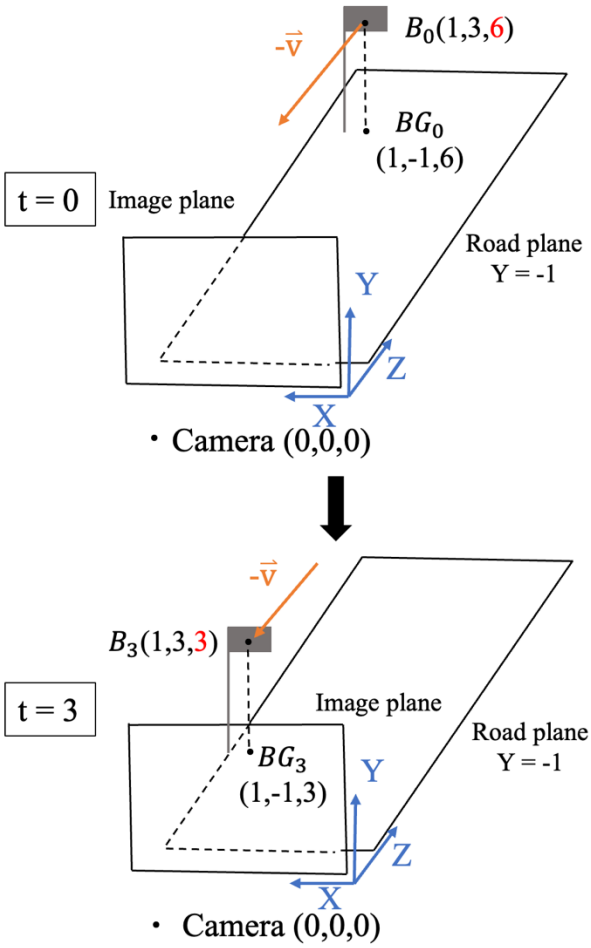


Figure 24: The coordinates of object B from t = 0 to t = 3

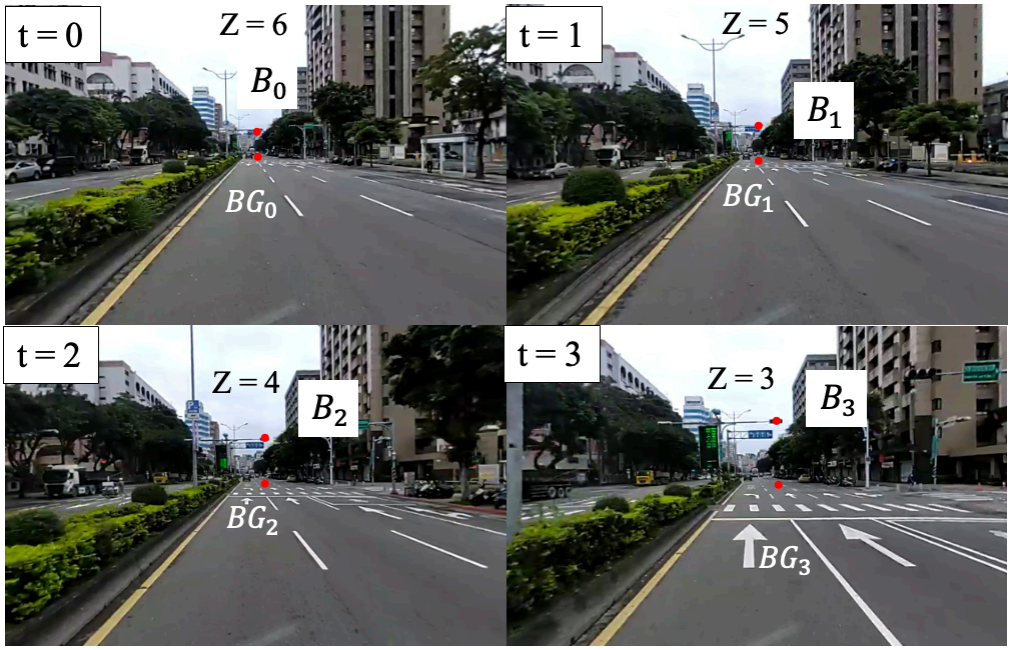


Figure 25: The schematic diagram of object B from t = 0 to t = 3 in the real world

doi:10.6342/NTU202300097

Therefore, the coordinates of object B and ground point BG at different times in the image are denoted as b_t and bg_t, where t is from 0 to 3. The y-coordinates of object B and ground point BG at four times in the image are calculated and the results are shown in **Table 3**. The difference of y-coordinate in the image is denoted as Δy_b and is obtained by directly subtracting the y-coordinate of b_t and the y-coordinate of bg_t. Then, we plot the value of y-coordinate of b_t and the value of Δy_b into a figure with customized xy-coordinate as shown in **Figure 26**. The purpose of ploting a new figure is to observe the relationship more clearly. As we can see, the x-coordinate of this figure is the y-coordinate of the b_t; the y-coordinate of this figure is Δy_b. Each point in this figure represents the position of the object and the difference of y-coordinate at a time. Finally, we obtain a stright line passing through the origin with slope value of $\frac{4}{3}$. Next, we implement the same procedure above to analyze the object E.

	B_0	B_1	B_2	B_3
y-coordinate of b_t	$\frac{3}{6}f$	$\frac{3}{5}f$	$\frac{3}{4}f$	$\frac{3}{3}f$
	BG_0	BG_1	BG_2	BG_3
y-coordinate of bg_t	$\frac{-1}{6}f$	$\frac{-1}{5}f$	$\frac{-1}{4}f$	$\frac{-1}{3}f$
The difference of y-coordinate (Δy_b)	$\frac{4}{6}f$	$\frac{4}{5}f$	$\frac{4}{4}f$	$\frac{4}{3}f$

Table 3: The coordinates of B and BG from t = 0 to t = 3 in the image

doi:10.6342/NTU202300097

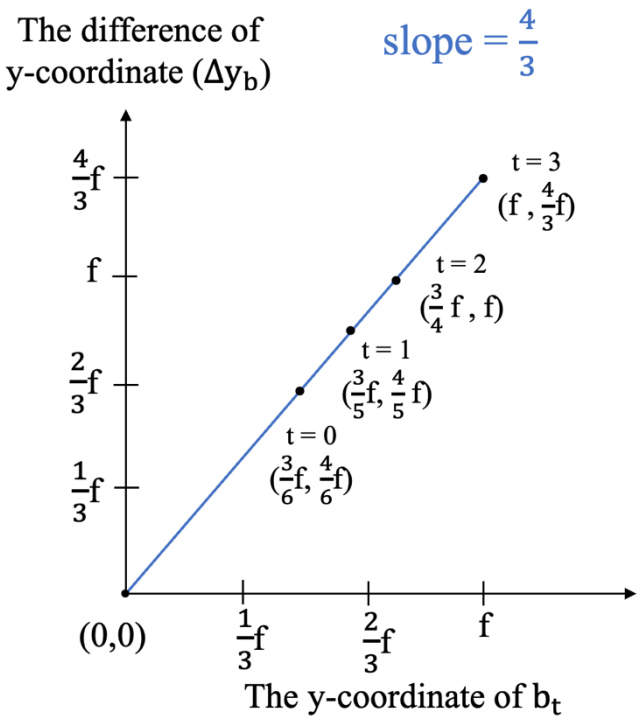


Figure 26: The relationship between the y-coordinate of b_t and Δy_b

The same as the previous work conducted on object B, we demonstrate the coordinate of object E at four different times. The only difference between these two objects is the starting position of the object. We denote the coordinates of object E from t = 0 to t = 3 as E_0 to E_3, and its ground point EG from t = 0 to t = 3 as EG_0 to EG_3, respectively. Therefore, we present the scene at four times by the coordinates as shown in **Figure 27**. The difference of y-coordinate between E and EG is 4 in the world coordinate. Moreover, the schematic diagram of four different times in the actual world are also shown in **Figure 28** for easier understanding. These images are the same as **Figure 25** but the traffic light marked on the images are different. Afterwards, we calculate the coordinate of these points in the image and conduct the same analysis as previous.

doi:10.6342/NTU202300097

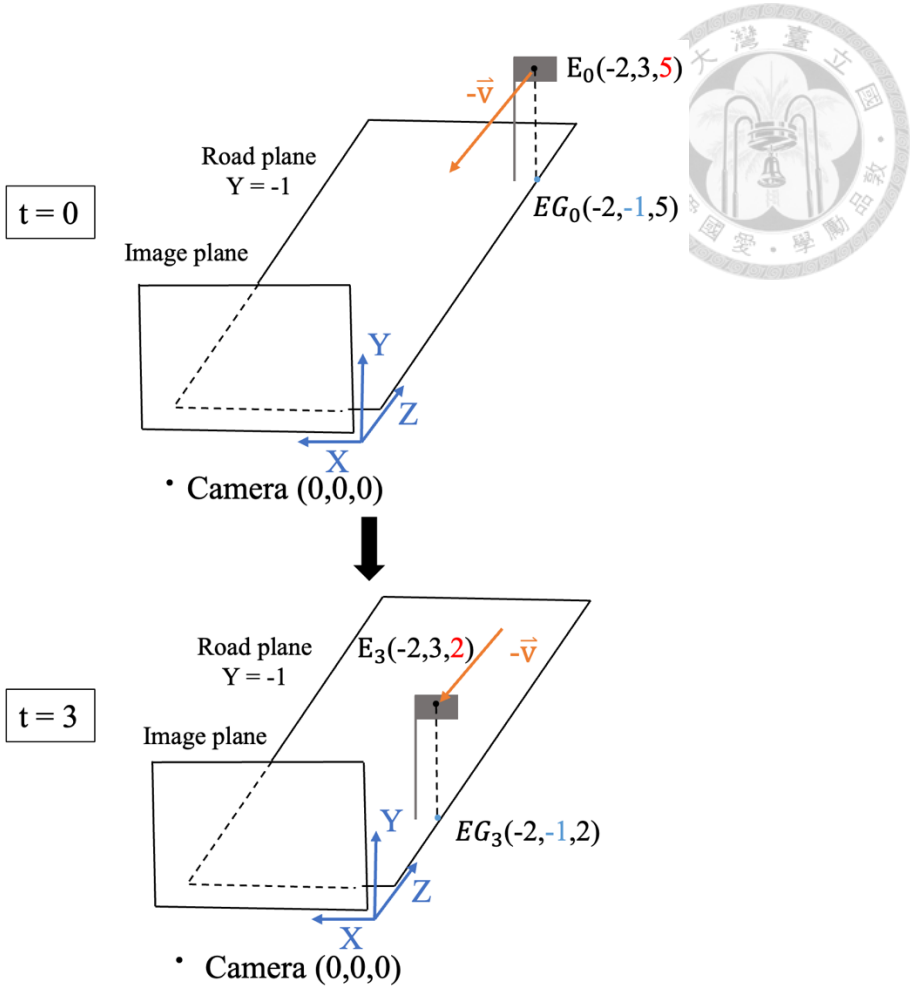


Figure 27: The coordinates of object E from t = 0 to t = 3

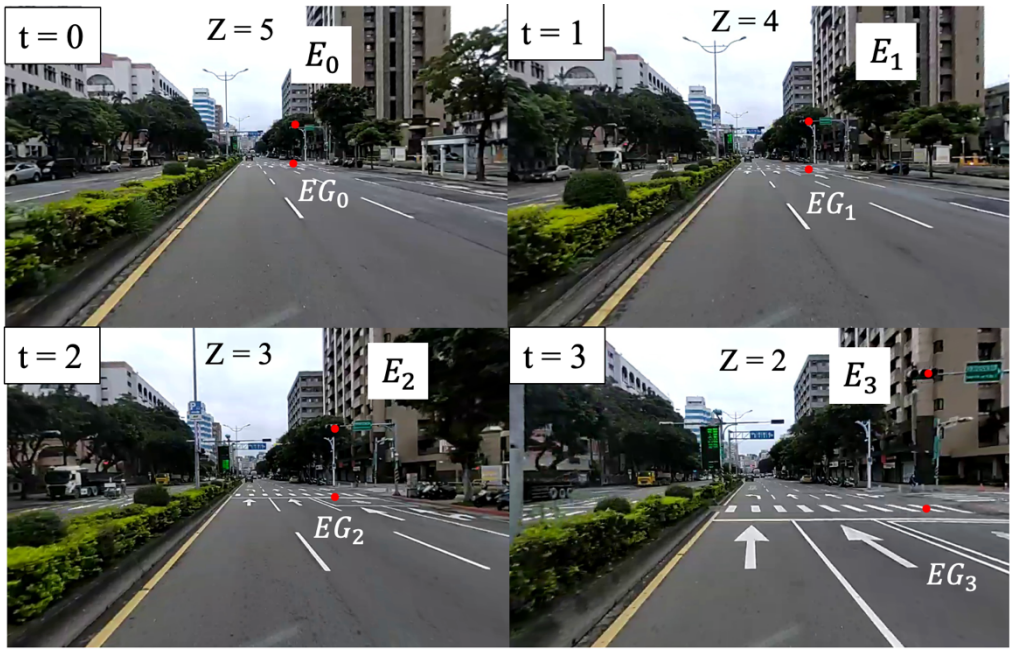


Figure 28: The schematic diagram of object E from t = 0 to t = 3 in the real world

doi:10.6342/NTU202300097

We denote the coordinate of object E and ground point EG at different times in the image as e_t and eg_t, where t is from 0 to 3. The y-coordinates of object E and ground point EG at four times in the image are shown in **Table 4**. The difference of y-coordinate in the image is denoted as Δy_e. Then, we take the same fields of the table as the previous analysis of object B to plot a new figure, as shown in **Figure 29**. As we can see, the x-coordinate of this figure is the y-coordinate of the e_t; the y-coordinate of this figure is Δy_e. The same as **Figure 26**, we get a line passing through the origin with slope of $\frac{4}{3}$.

	E_0	E_1	E_2	E_3
y-coordinate of e_t	$\frac{3}{5}f$	$\frac{3}{4}f$	$\frac{3}{3}f$	$\frac{3}{2}f$
	EG_0	EG_1	EG_2	EG_3
y-coordinate of eg_t	$\frac{-1}{5}f$	$\frac{-1}{4}f$	$\frac{-1}{3}f$	$\frac{-1}{2}f$
The difference of y-coordinate (Δy_e)	$\frac{4}{5}f$	$\frac{4}{4}f$	$\frac{4}{3}f$	$\frac{4}{2}f$

Table 4: The coordinates of E and EG from t = 0 to t = 3 in the image

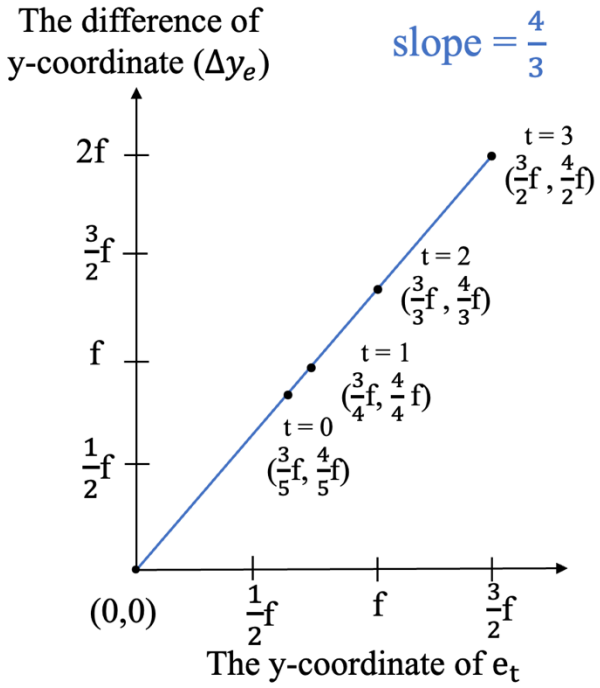


Figure 29: The relationship between the y-coordinate of e_t and Δy_e

doi:10.6342/NTU202300097

Finally, we have a few conclusions from the example described above. We can see from the **Figure 26** and **Figure 29**, objects with the same height at any position will obtain the same slope value, which is independent of the object position. Therefore, the slope value is only related to the height of the object. Moreover, this slope value represents the relationship between the y-coordinate of an object and the difference of y-coordinate in the image. Therefore, if we have the coordinate of an object with a known height, we can derive its ground point in the image by this slope value. Next, we want to obtain the general form of the slope for the application in the future.

General form of the slope

We will use the geometric coordinate to derive the general form of the slope. The coordinate assumption is shown in **Figure 30**. It is supposed that the camera is the origin of the coordinate system and the image plane is at Z = f. The road plane is at the coordinate of Y = -HC, where HC is the height of the camera. Besides, there is an object P at the coordinate of (X_P, HY, Z_P) and its ground point on the road plane is PG(X_P, -HC, Z_P). This object is moving in the direction $-\vec{v}(0, 0, -1)$ towards the camera. The purpose to utilize the vector $\vec{v}$ is to simulate the movement of the camera on the road. The height of this object is denoted as HO as the blue vertical line in the figure. The projection of HO on the image plane is denoted as Δy_p as the red vertical line in the figure. Then, we denote the y-coordinates of object P and PG in the image as p and pg, respectively. The coordinate of p and pg at any time t ($t \in \mathbb{R}$) as well as the difference of y-coordinate between p and pg in the image are listed in **Table 5**. Since we want to derive the general form of the slope, we compute that Δy_p divided by the y-coordinate of p. The calculation result is as follows:

doi:10.6342/NTU202300097

$$\text{Slope} = \frac{HY+HC}{HY} \tag{3.3}$$

where HY is the difference of y-coordinate between the camera and the object P in real world, HC is the height of the camera.

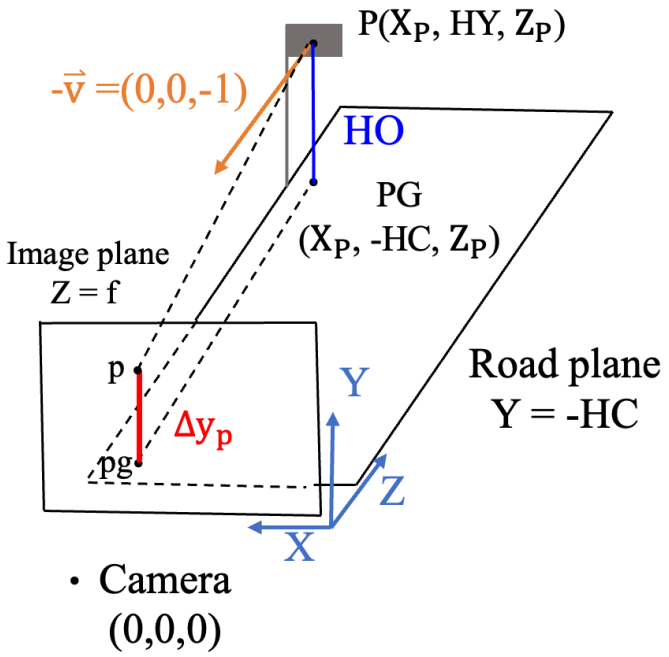


Figure 30: The coordinate assumption for obtaining general form of the slope

	P at any time t ($t \in \mathbb{R}$)
The y-coordinate of p	$\frac{HY}{Z_P+t}f$
	PG at any time t ($t \in \mathbb{R}$)
The y-coordinate of pg	$\frac{-HC}{Z_P+t}f$
The difference of y-coordinate (Δy_p)	$\frac{HY+HC}{Z_P+t}f$

Table 5: The coordinates of P and PG at any time in the image

doi:10.6342/NTU202300097

We call Equation (3.3) the slope equation in the following description. From the equation, HC is the height of camera as the blue vertical line in **Figure 31**; HY is the difference of y-coordinate between the camera and the object P in 3D real world as the red vertical line in **Figure 31**. Besides, the height of the object is composed of HY and HC. Thus, we rewrite the Equation (3.3) as below:

$$\text{Slope} = \frac{\text{HO}}{\text{HO}-\text{HC}} \tag{3.4}$$

where HO represents the height of the object.

It can be seen from the Equation (3.4), when the height of camera (HC) is constant, the slope is only related to the height of the object. This is consistent with the conclusion of the previous example. However, even though we can know the height of some objects on the road according to the traffic rules in Taiwan, the height information of camera at which the image has been taken cannot be obtained. It is impossible for us to reproduce the situation that this image was taken. Thus, we find the projection of HY and HC on the image plane to obtain the ratio of them. The schematic diagram is shown in **Figure 32**.

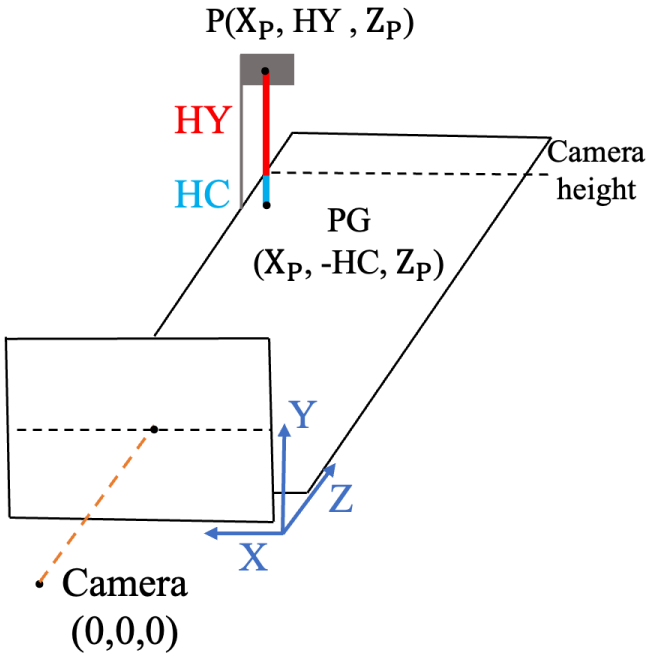


Figure 31: The schematic diagram of HY and HC

doi:10.6342/NTU202300097

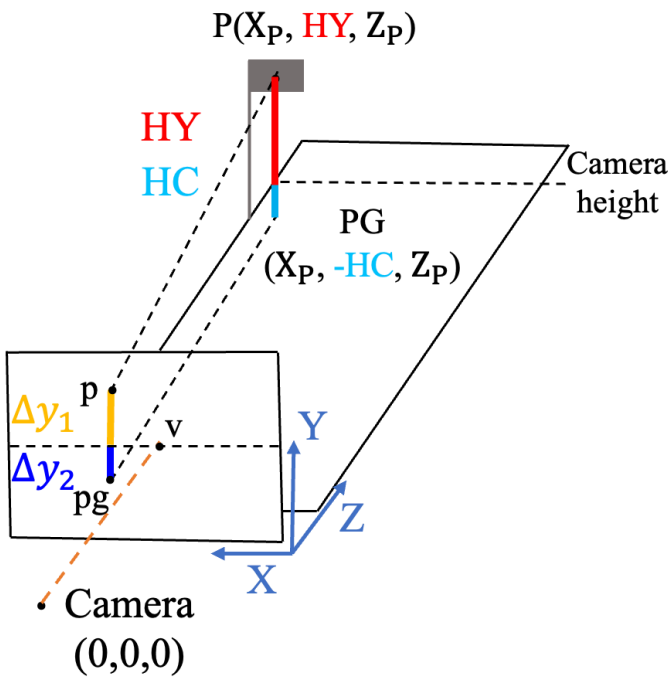


Figure 32: The projections of HY and HC on the image

Previously introduced in Section 3.3.1, we know that a line extended along the Z direction from the camera to infinity will intersect the image plane at the point v(0, 0). That is to say, the height of the point v(0, 0) in the image is identical to the height of the camera in the world coordinates. As we can see from the **Figure 32**, the difference of y-coordinate between p and v is denoted as Δy_1; the difference of y-coordinate between v and pg is denoted as Δy_2. By the camera imaging model described in Section 3.3.1, the coordinate of object P in the image is at p($\frac{X_P}{Z_P}$f, $\frac{HY}{Z_P}$f); the coordinate of ground point PG in the image is at pg($\frac{X_P}{Z_P}$f, $\frac{-HC}{Z_P}$f). Thus, we can calculate the value of Δy_1 and Δy_2 as $\frac{HY}{Z_P}$f and $\frac{HC}{Z_P}$f, respectively. It is observed that the ratio of Δy_1 and Δy_2 in the image is equal to the ratio of HY and HC. An example of Δy_1 and Δy_2 in the real world is demonstrated in **Figure 33**. In the figure, we label the position of the green board at point p and the position of the ground point of this green board at pg. The yellow vertical line

doi:10.6342/NTU202300097

which represents Δy_1 is the difference of y-coordinate between object p and the origin v; the blue vertical line which represents Δy_2 is the difference of y-coordinate between the origin v and the ground point pg. Finally, we rewrite Equation (3.3), namely the slope equation, as below:

$$\text{Slope} = \frac{\Delta y_1 + \Delta y_2}{\Delta y_1} \tag{3.5}$$

With Equation (3.5), we can obtain the slope value directly from an 2D image by finding the value of Δy_1 and Δy_2, instead of calculating the coordinate relationship between consecutive images.

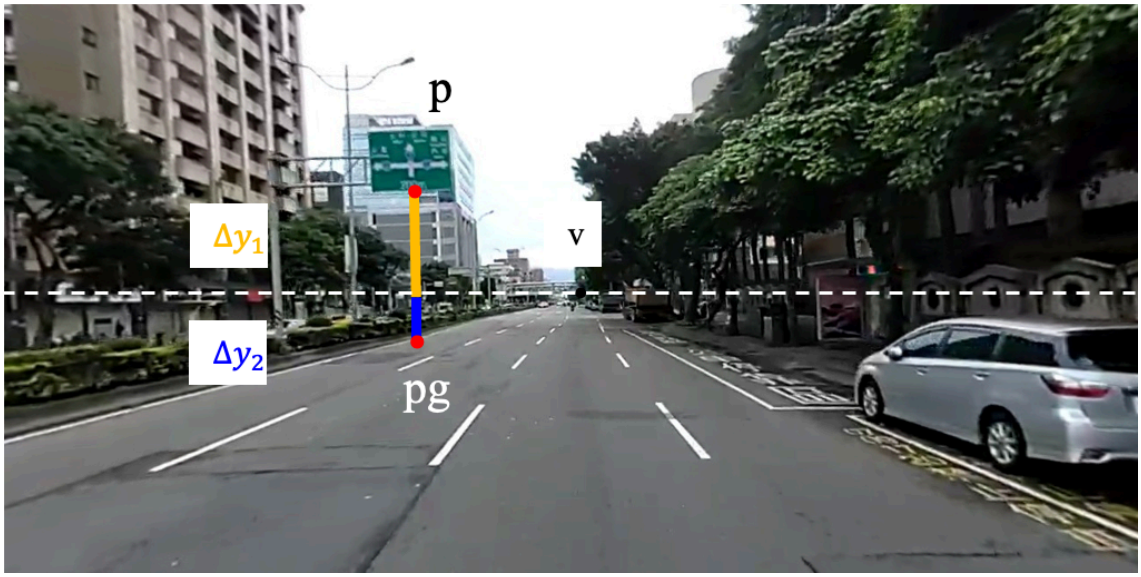


Figure 33: An example of Δy_1 and Δy_2 in the real world

3.3.4 Step 2.4: Estimate the Ground Point of Any Object with a Known Height

In the previous subsection, we derive the relationship between the position of an object with a known height and its difference of y-coordinate in the image. In this section, we will introduce how to apply this relationship to any object with a known height in the

doi:10.6342/NTU202300097

target image by an initial image. This procedure can be accomplished in the initialization phase by labeling the position of a known height object in the initial image. Using the actual data to introduce this procedure is more easily to understand than using the unknowns. Therefore, we continue the example as shown in **Figure 33**. Given an initial image contains a green board at the coordinate of p(220, 200) with the height h of 6m, as shown in **Figure 34**. The origin of this image is at v(0, 0). The coordinate of its ground point is at g(220, -100). In this figure, the edge of the pole connected to the green board and the common boundary between the refuge island and the road planar are used to find the y-coordinate of the ground point. Then, we obtain this ground point g by downwards stretching the object. This step of labeling the ground point is subjective. The initial image is the known information we utilize to estimate the ground point of any object with a known height.

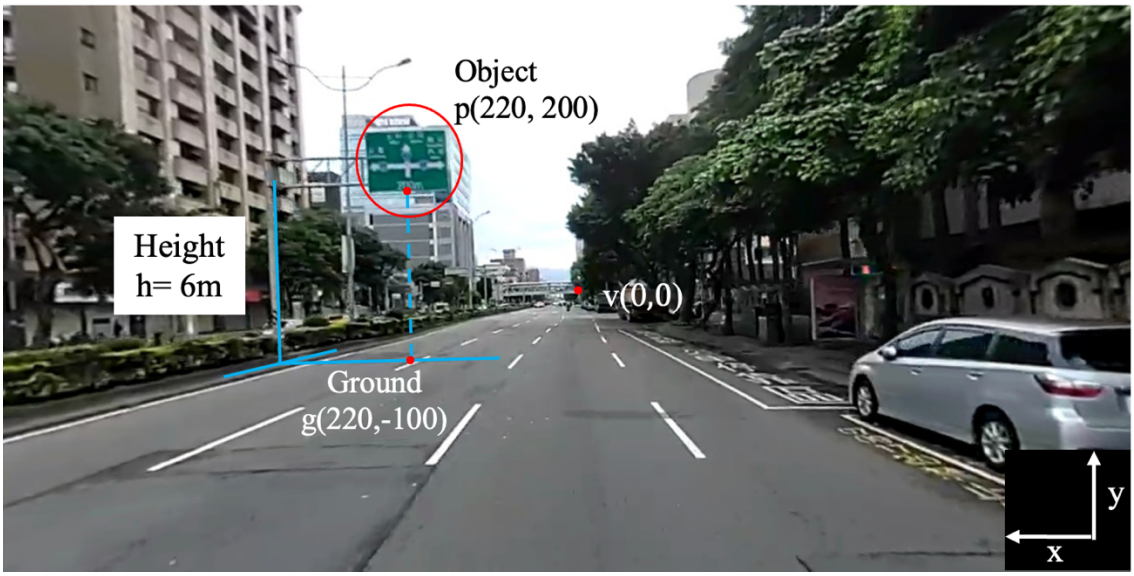


Figure 34: The example of an initial image

It is assumed that in the future application, there is another object with the height h' of 5m detected by the model in the target image. The coordinate of this object is located

doi:10.6342/NTU202300097

at p'(420, 250) in the target image, as shown in **Figure 42**. Since the existing methods cannot directly detect the ground point of this object, we want to use the slope value introduced in the previous subsection to find the coordinate of the ground point g'. We can downwards shift the difference of y-coordinate in the image from the position of the object p' to the ground point g'. Besides, the difference of y-coordinate in the image is calculated by slope value of 5m height. However, we cannot directly calculate the slope value of 5m height from the target image, since the position of ground point g' is unknown so that Δy_2 cannot be obtained. Therefore, we will compute the slope value of 5m height from the known information in the initial image.

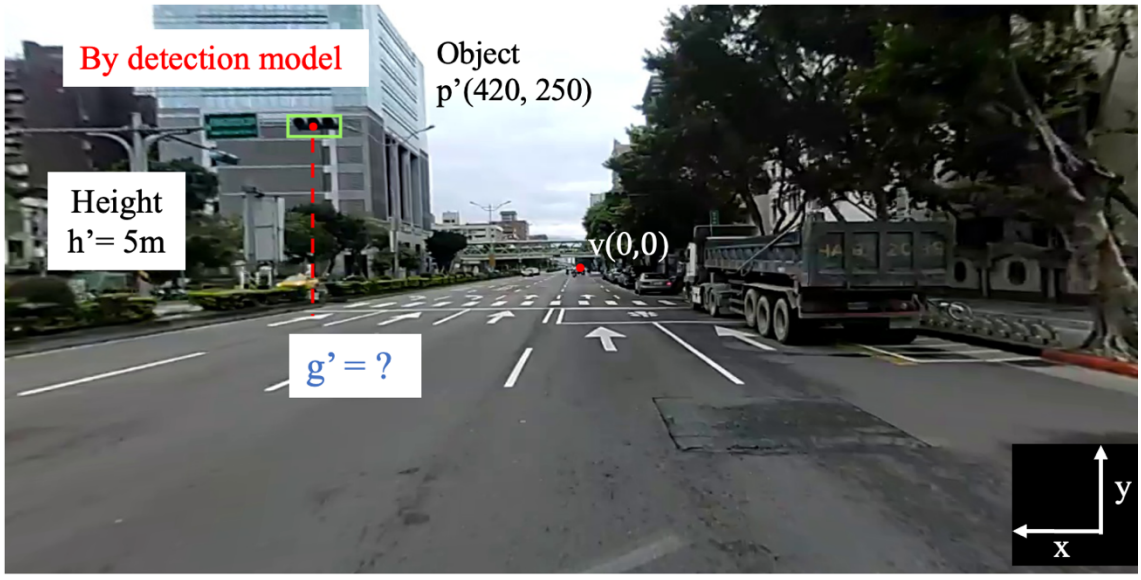


Figure 35: The example of an target image

First, we have to calculate the pixel resolution of height by the ratio of the real height of this object and the number of pixels this object in the image. The schematic diagram of calculating the pixel resolution of height is shown in **Figure 36**. As we can see from the figure, the number of pixels at the height of object p in the image is obtained as the

doi:10.6342/NTU202300097

difference of y-coordinate between p and g, which is 300 pixels. Then, the pixel resolution of height is calculated as $\frac{6}{300}$ = 0.02 meters per pixel. That is, the actual height of a pixel in the image is 0.02m. Review the previous conclusion of the example described in Section 3.3.3, the slope value of objects with the same height calculated at any position in the image is identical. Therefore, we can use the pixel resolution of height to imagine the position of a 5m height object on the ground point g in the initial image.

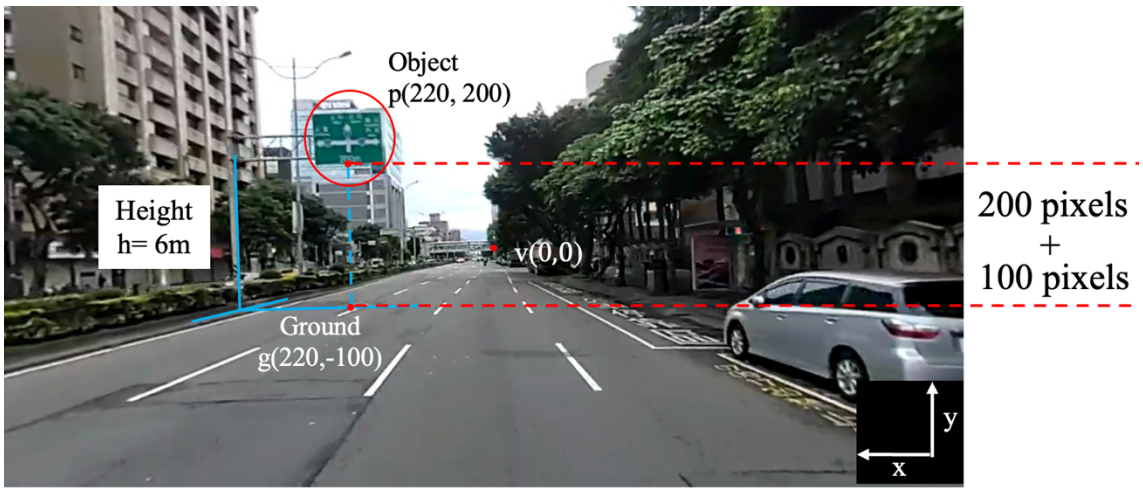


Figure 36: Calculating the pixel resolution of height in the initial image

The height of 5 meters in pixels is calculated as $\frac{5}{0.02}$ = 250 pixels. Therefore, we can imagine that a 5m height object is placed on the ground point g. The imaginary position of a 5m height object is located at the coordinate of (220, -100+250) and is denoted as ip, as shown in **Figure 37**. We can see from the figure, the position of ip is lower than the position of green board with 6m height. As long as the coordinate of the object and its ground point is obtained, the slope value of 5m height object will be calculated by Equation (3.5). In this figure, Δy_1 and Δy_2 are 150 and 100, respectively. Thus, the slope value of 5m height object is calculated as $\frac{150+100}{150}$ = 1.667.

doi:10.6342/NTU202300097

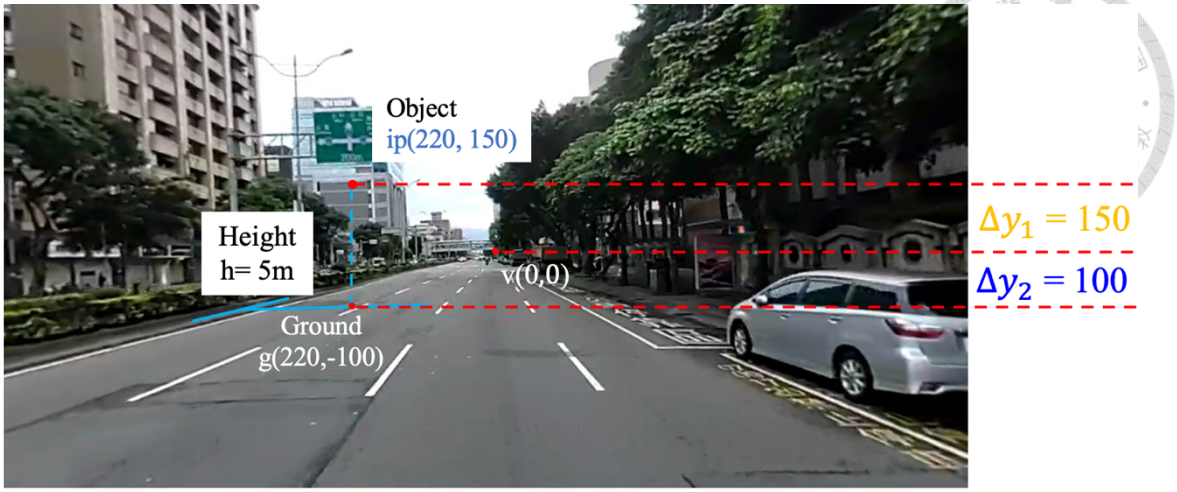


Figure 37: The imaginary position of a 5m height object in the initial image

Finally, we can estimate the ground point g' in the target image by the coordinate of p'(420, 250) and the slope value of 5m height object. The difference of y-coordinate between p' and its ground point is computed as 250 * 1.667 = 416. The ground point p' is shifting down the position of object p' by 416 pixels. As a result, we obtain the coordinate g'(420, 250 - 416) = g'(420, -166), as shown in **Figure 38**. Through this procedure, the ground point of any object with a known height can be estimated by an initial image in the inference phase.

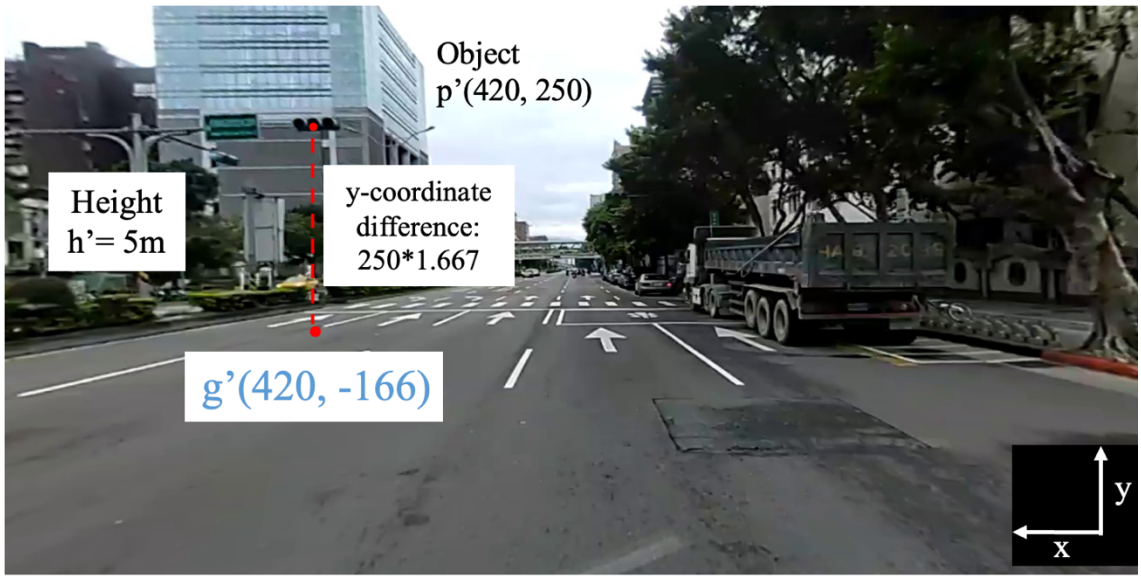
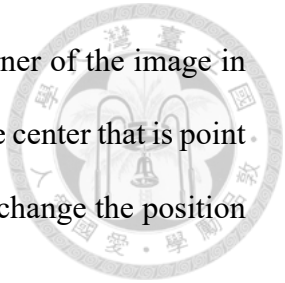


Figure 38: Estimate the groud point of the object in target image

doi:10.6342/NTU202300097



However, the origin of the image is usually at the upper left corner of the image in common usage, but in our introduction, the origin of the image is at the center that is point v of the image. Therefore, we can simply impose some xy-offsets to change the position of the origin to satisfy the common usage of the image.

3.4 Step 3: Perspective Transform

In Section 3.3, we introduce how to estimate the ground point of any object with a known height in the image. Next, we will describe how to estimate the distance of a ground point by transforming the perspective to a bird-eye view in this section. We also introduce the method that utilizes a known length reference in Section 3.4.3.

3.4.1 Step 3.1: Define the Region of the Bird-eye Image

We process a road image collected by the onboard camera. The region of interest in the image must be selected to transform into a bird-eye image. First, we estimate the coordinate of the vanishing point in the image and the procedure will be introduced later. In our solutions, we fix the y-coordinate position of the vanishing point in the initialization phase, since normally the height of the onboard camera does not change. Although the position of the vanishing point may change during driving on the flat road, the amount of change will be small in the y-coordinate. Generally, most of the changes in the position of the vanishing point are in the x-coordinate when driving on a flat road.

Vanishing point estimation

In the initialization phase, the image that we select contains several parallel lines, such as, the white dashed lines on the road. Since we cannot determine the position of

doi:10.6342/NTU202300097

parallel lines directly from the image, a region of interest mask is defined to filter out the nonparallel lines. Therefore, we are able to obtain the position of the vanishing point. An example of the procedure to select parallel lines is shown in **Figure 39**. First, we implement the canny edge detection algorithm on the original image. The result of the edge detection is shown in **Figure 39(b)**. Moreover, we produce the region of interest (ROI) mask by manually assigning four points to surround a quadrilateral, as shown in **Figure 39(c)**. All of the detected edges contained in the ROI mask are parallel in the real world. Eventually, we filter the edge detection result using the ROI mask as shown in **Figure 39(d)**. Then, we utilize the Hough lines transform to detect the lines by the filtered edge detection result.

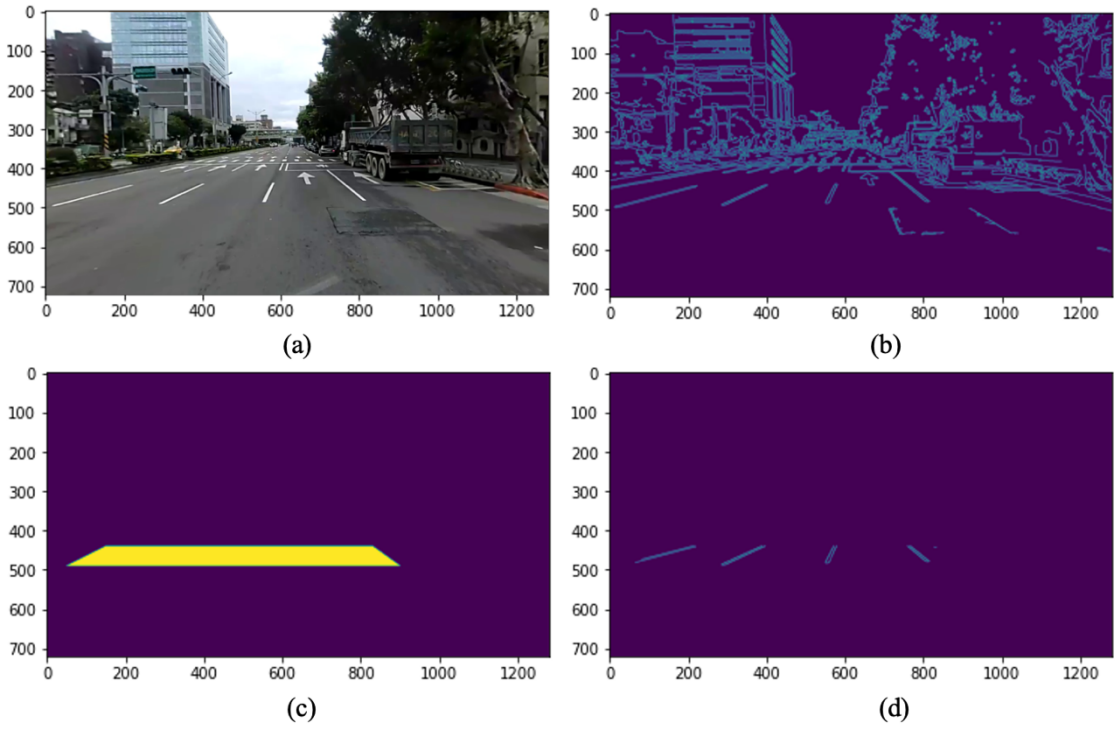


Figure 39: An example of filtering parallel lines

(a) The original image (b) The result of edge detection

(c) Region of interest mask (d) The selected parallel edges

doi:10.6342/NTU202300097

It is observed that we just select four white dashed lines in the **Figure 39(d)**. Unfortunately, when extending these four lines, they do not exactly intersect at one point most of the time. Moreover, the Hough lines transform may usually detect more than four lines in this example even though the parameters of the algorithm are fine-tuned. Therefore, we employ an optimization procedure proposed in [17] to determine one unique intersecting point among more than two parallel lines. Then, we describe the optimization procedure. Each line found by the Hough lines transform can be represented by a point p_i on each line. The unit normal of each line found by the Hough lines transform is denoted as n_i. The coordinate of the vanishing point is denoted as vp. Then, the cost function to be minimized is defined as below:

$$I = \frac{1}{2} * \sum (n_i^T * (vp - p_i))^2 \tag{3.6}$$

where i represents the number of lines found by the Hough lines transform and I is the cost that we want to minimize.

To find the minimum value of the cost function, the partial derivative of Equation (3.6) with respect to vp is calculated. After conducting some derivations, the following equation is obtained:

$$\frac{\partial I}{\partial vp} = 0$$

$$\Longrightarrow (\sum n_i * n_i^T) * vp = (\sum n_i * n_i^T * p_i)$$

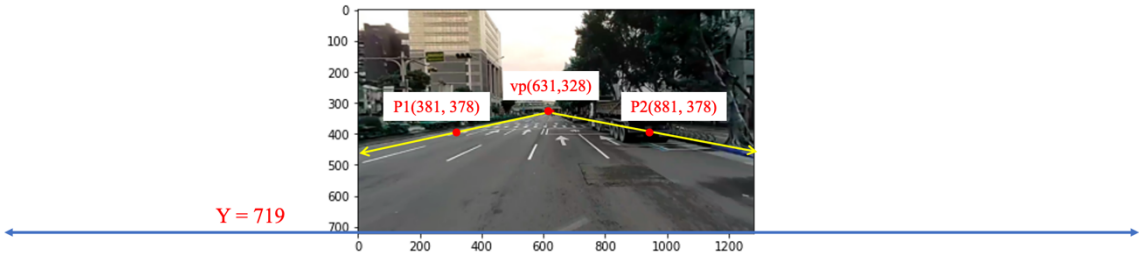
$$\Longrightarrow vp = (\sum n_i * n_i^T)^{-1} * (\sum n_i * n_i^T * p_i) \tag{3.7}$$

We can use the Equation (3.7) to find the coordinate of the vanishing point by the lines found by the Hough lines transform. Once the coordinate of the vanishing point is known, the region that will transform into a bird-eye image can be acquired. Next, we are going to define four pairs of points which are four source points of the image and four destination points of its bird-eye image. The coordinates of the first two points in the

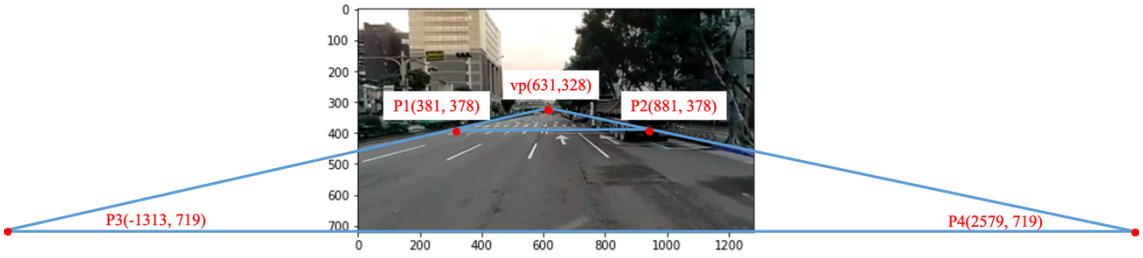
doi:10.6342/NTU202300097

image are calculated by manually adding xy-offset to the vanishing point. For instance, the coordinate of the vanishing point obtained in the previous procedure is at vp(631,328). Then, we assign that the x-offset is ±250 and the y-offset is +50. The coordinate of the first two points P1 and P2 are selected at (381, 378) and (881, 378), respectively.

Moreover, we utilize the vp, P1, and P2 to obtain the other two points which are denoted as P3 and P4. The method of finding the coordinates of P3 and P4 is extending the two lines starting from the coordinate of vp. These two extending lines separately pass through P1 and P2, as the yellow arrow lines shown in **Figure 40(a)**. The extended bottom edge of the image is denoted as a blue line with y-coordinate value of 719, as shown in **Figure 40(a)**. These two extending lines will intersect the extended bottom edge of the image at P3 and P4. Therefore, the coordinates of P3 and P4 are obtained as (-1313, 719) and (2597, 719), respectively. The result of four points in the image is shown in **Figure 40(b)**. P1, P2, P3, and P4 are used as source points to surround the ground area in the image.



(a) Extending two lines passing through P1 and P2



(b) The result of four source points in the image

Figure 40: An example of finding four source points

doi:10.6342/NTU202300097

3.4.2 Step 3.2: Generate the Bird-eye Image

After obtaining four source points in the image, we define the size of the bird-eye image and the corresponding four destination points. Furthermore, we are able to utilize these four pairs of points and the original image to acquire the bird-eye image. The procedure is divided into two parts which are getting the homography matrix and warping the image into bird-eye image. These two parts will be introduced in Section 3.4.2.1 and Section 3.4.2.2, respectively.

3.4.2.1 Get Homography Matrix

First, we define the size of the bird-eye image as 500*600 which is referred to [15]. The reason why the author defines this size is probably that it can exhibit the bird-eye image sufficiently. Moreover, the size of the bird-eye image has little amount of influence on the distance estimation result. It is only related to the pixel resolution of the ground, which will be introduced in later section. With the size of the bird-eye image, we can generate four destination points which are the coordinate of four corners of the bird-eye image. The four pairs of source points and destination points are shown in **Figure 41**.

Source	Destination
P1(381, 378)	P1’(0, 0)
P2(881, 378)	P2’(500, 0)
P3(-1313, 719)	P3’(0, 600)
P4(2597, 719)	P4’(500, 600)

Figure 41: Four pairs of source and destination points

doi:10.6342/NTU202300097

Next, we utilize these four pairs of points to calculate the homography matrix which is an important concept in computer vision. It has various applications, such as image rectification and perspective transform. Besides, it can correlate arbitrary two images of the same planar surface in real world. Previously, the procedure of the camera imaging model is derived in Equation (3.1). Therefore, we denote the homography matrix by H, and the correlation of arbitrary two images of the same planar in space can be written as follows:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = H * \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{3.8}$$

$$\text{where H is } \begin{bmatrix} r_x & \gamma & u_0 \\ 0 & r_y & v_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_4 & t_1 \\ r_2 & r_5 & t_2 \\ r_3 & r_6 & t_3 \end{bmatrix}$$

The difference between Equation (3.1) and Equation (3.8) is that the transformation does not contain three-dimensional points. It is noticed that the homography matrix consists of the intrinsic matrix and the extrinsic matrix. Then, we represent the homography matrix as a 3 x 3 matrix, and rewrite Equation (3.8) as follows:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_4 & h_7 \\ h_2 & h_5 & h_8 \\ h_3 & h_6 & h_9 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{3.9}$$

where the x_1, y_1, x_2, and y_2 are the coordinate of points in two images of the same planar surface in real world. h_1, h_2, …, h_9 are the elements of homography matrix and solving these unknows is our calculation purpose. After expanding the matrix in Equation (3.9), we obtain three equations as below:

$$x_2 = h_1x_1 + h_4y_1 + h_7 \tag{3.10}$$

$$y_2 = h_2x_1 + h_5y_1 + h_8 \tag{3.11}$$

$$1 = h_3x_1 + h_6y_1 + h_9 \tag{3.12}$$

doi:10.6342/NTU202300097

We put Equation (3.12) into Equation (3.10) and Equation (3.11). Then, we obtain the equation as follows:

$$x_2 = \frac{h_1 x_1 + h_4 y_1 + h_7}{h_3 x_1 + h_6 y_1 + h_9} \quad (3.13)$$

$$y_2 = \frac{h_2 x_1 + h_5 y_1 + h_8}{h_3 x_1 + h_6 y_1 + h_9} \quad (3.14)$$

Since it is a homogeneous coordinate, the homography matrix can be scaled in any non-zero values. That is to say, we can divide all elements by h_9 and the value of h_9 in Equation (3.13) and Equation (3.14) as 1. Therefore, we obtain the following two equations:

$$x_2 = \frac{h'_1 x_1 + h'_4 y_1 + h'_7}{h'_3 x_1 + h'_6 y_1 + 1} \quad (3.15)$$

$$y_2 = \frac{h'_2 x_1 + h'_5 y_1 + h'_8}{h'_3 x_1 + h'_6 y_1 + 1} \quad (3.16)$$

where h'_1 to h'_8 are the notations which represent h_1/h_9 to h_8/h_9 in the Equation (3.13) and Equation (3.14), respectively.

Eventually, with four pairs of points previously obtained in **Figure 41**, Equation (3.15) and Equation (3.16) can be solved. We replace the unknown x_1 and y_1 with the source points, the unknown x_2 and y_2 with destination points, respectively. Therefore, the homography matrix will be obtained as below:

$$H = \begin{bmatrix} h'_1 & h'_4 & h'_7 \\ h'_2 & h'_5 & h'_8 \\ h'_3 & h'_6 & 1 \end{bmatrix} \quad (3.17)$$

3.4.2.2 Warping the Image into a Bird-eye Image

After acquiring the homography matrix between the source image and the destination image, we are able to warp the image into a bird-eye image. The concept of

doi:10.6342/NTU202300097

warping an image is very intuitive. We can use the homography matrix to transform a coordinate of the original image into a coordinate of the bird-eye image. For example, the homography matrix computed by the four pairs of points in **Figure 41** is $\begin{bmatrix} -0.1524 & -0.762 & 348 \\ 0 & -2.098 & 793 \\ 0 & -0.00305 & 1 \end{bmatrix}$. Therefore, we can match a coordinate (556, 485) of the original image to $(x_2,\ y_2)$ of the bird-eye image by Equation (3.15) and Equation (3.16) derived above. The following calculation shows the transformation.

$$x_2 = \frac{-0.1524*556 - 0.762*485 + 347.9}{0*556 - 0.00305*485 + 1} \approx 222$$

$$y_2 = \frac{0*556 - 2.098*485 + 792.9}{0*556 - 0.00305*485 + 1} \approx 468$$

The relationship between the coordinate of the original image and the coordinate of the bird-eye image is obtained. Then, we can directly assign the RGB color space values to the corresponding position. An example of color transfer is shown in **Figure 42**.

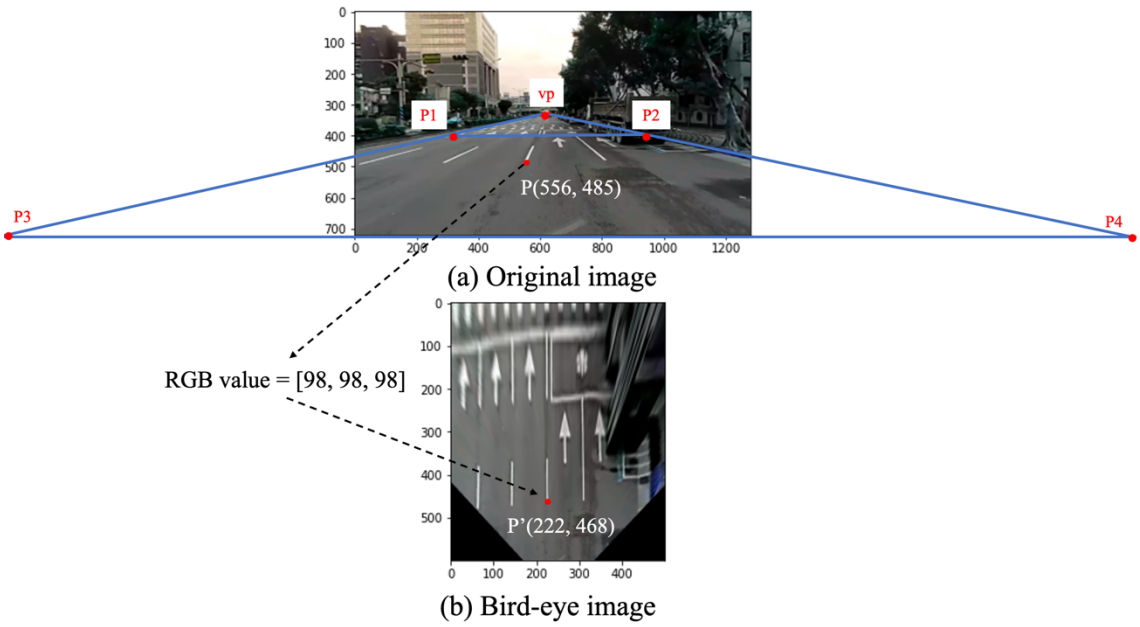


Figure 42: An example of color transfer

doi:10.6342/NTU202300097

In **Figure 42(a)**, the original image contains a previously selected region which is surrounded by P1, P2, P3 and P4. We take a point P(556, 485) in original image for example. The RGB color value of point P is [98,98,98] and it is transferred to the corresponding point P'(222, 468) in the bird-eye image. Besides, the RGB color value of other points in the selected region of original image are also transferred to the corresponding point in the bird-eye image in the same way. The result of bird-eye image is shown in **Figure 42(b)**.

3.4.3 Step 3.3: Pixel Resolution of Ground

Before we estimate the distance on the ground by a bird-eye image, the pixel resolution of ground, which means how many pixels there are in a meter, must be known. In this subsection, we will introduce an algorithm that we develop to obtain the scale of the pixels in the bird-eye image. This step is implemented in the initialization phase and only requires a one-time computation. The procedure of acquiring the pixel resolution is divided into two steps. In first step, we extract the two endpoints of a white dashed line in the original image by the endpoint extraction algorithm that will be introduced later. In second step, we calculate the corresponding coordinate of these endpoints in the bird-eye image by homography matrix. Through this, we are able to know how many pixels of the white dashed lines are in the bird-eye image. The reason why we utilize the white dashed lines is that the traffic rules in Taiwan formulate these lines as four meters [15]. Therefore, we can take these lines as length reference.

Endpoint extraction algorithm

The purpose of this algorithm is to find two coordinates to represent a white dashed line. When we manually label the endpoints of the white line, their positions are not

doi:10.6342/NTU202300097

usually identical every time. Therefore, we develop a procedure to obtain the endpoints of white dashed lines as well as it can facilitate the work of labeling a line for verification in the future. The algorithm can be divided into seven steps and the pipeline of this algorithm is demonstrated in **Figure 43**.

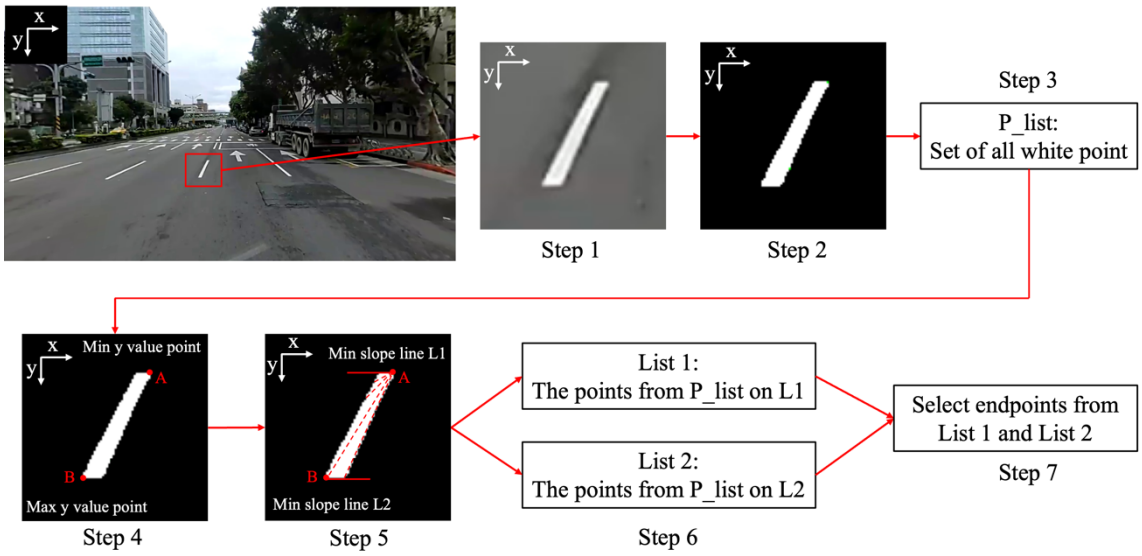


Figure 43: A pipeline of endpoints extraction algorithm

In Step 1, it is impossible to directly extract the position of lines from the entire image. Thus, we manually select the region that only contain a white dashed line in the image. In Step 2, we find the area of the white dashed line by binarizing the region selected in Step 1. The color of the pixels in the binary image is black or white. After observing plenty of white dashed lines in road images, we found that their shape may have various shapes in the image. However, their shape is a rectangle in the real world. Due to the relative position of the camera and the white dashed lines, the shape of white dashed lines is not always a rectangle in the image.

Moreover, we can roughly divide the shape of white dashed lines into four types by our observation. The schematic diagram is shown in **Figure 44**. In Type 1, both ends of the lines are parallel and their shape may be skewed. In Type 2, they are in the shape of

doi:10.6342/NTU202300097

parallelogram and trapezoid with left or right skewed. In our observation, the first two types are about 79% of the majority in the road image. In Type 3, one of the ends is horizontal to the bottom of image and the other end is like any ends in Type 2. Type 3 is about 13% in our observation. The other shapes of the white dashed line are categorized in type 4, such as a white line with twisted ends that may be caused by fading color. Type 4 is about 8% in our observation.

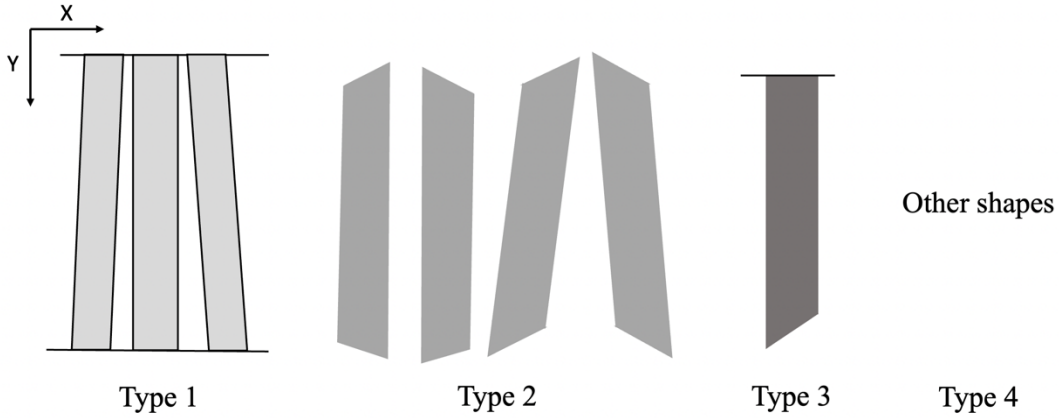


Figure 44: Schematic diagram of the white dashed line shape

Therefore, in order to find the endpoints of these various shapes of lines, we come up with the following steps. In Step 3, we collect all the coordinates of white points into a list which is denoted as P_list. In Step 4, the points in P_list with minimum y-value and maximum y-value are selected as point A and point B, respectively. In Step 5, we compute the slope value of point A and the other points in P_list. Then, the minimum absolute slope value is selected. Moreover, we define a line which starts from point A as L1. The slope of L1 is the minimum absolute slope value. Identically, we conduct the same process to point B and define a line which is denoted as L2. The reason why we select the minimum absolute slope value is that we will label the endpoints of a line on the upper edge and the under edge in the common situation. Besides, the characteristic of upper and under edge observed in **Figure 44** is the most horizontal line. Therefore, we are able to

doi:10.6342/NTU202300097

fix a point on the top and at the bottom to find a line with a minimum absolute slope value. In Step 6, we collect the white points on L1 into List 1 and the white points on L2 into List 2 separately. Finally, in Step 7, the two endpoints of the white dashed line are selected. We calculate the midpoint of the point with maximum x-value and the point with minimum x-value in List 1 and List 2, separately. The calculated midpoints are regarded as the endpoints of the white dashed line. Eventually, two coordinates to represent the selected white dashed line are obtained through this procedure.

Through the endpoint extraction algorithm, we can obtain the pixel resolution of ground by an image during the initialization phase. An example is shown in **Figure 45**. In **Figure 45(a)**, the original image with the endpoints of white dashed lines, namely E1 and E2, are obtained by the endpoint extraction algorithm. Then, we transform this image into a bird-eye image with corresponding endpoints, namely E1' and E2', as shown in **Figure 45(b)**. Therefore, we are able to obtain the pixel resolution of ground by computing the ratio of the y-coordinate difference of the white dashed line in the bird-eye image to the real length of the white dashed line. As a result, the pixel resolution of ground is computed as $\frac{468-353}{4} = 28.75$, which means there are 28.75 pixels per meter.

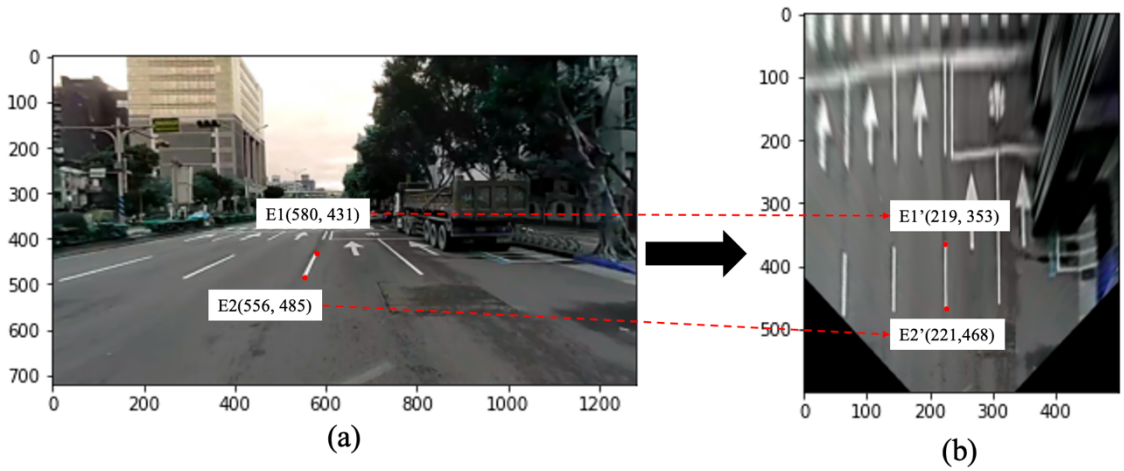


Figure 45: An example of obtaining pixel resolution

doi:10.6342/NTU202300097

3.5 Step 4: Distance Estimation

In Section 3.3, we introduce how to estimate the ground point of any object with a known height in the image. In Section 3.4, the procedure of transforming from the original road image to a bird-eye image with a pixel resolution of ground is described. In this section, we will present the last step of our proposed method. That is to estimate the distance of a ground point by the bird-eye image with a pixel resolution of ground. We take the image in **Figure 38** for example. The ground point g' of the object is obtained by the procedure described in Section 3.3.4. However, the origin of the image is usually at the upper left corner of the image in common usage, but the origin of this image is at the center that is point v of the image. Therefore, we change the position of the origin to the upper left corner of the image by simply imposing xy-offsets to the point v at first. The schematic diagram of changing the origin is shown in **Figure 47**. Then, we have a new coordinate of ground point g' that regards the upper left corner of the image as the new origin of the image.

Figure 46: The schematic diagram of changing the origin

doi:10.6342/NTU202300097

After obtaining the coordinate of the ground point, we transform the image into a bird-eye image by the procedure elaborated in Section 3.4. Besides, the corresponding ground point g' in the bird-eye image is calculated as (82, 248) by Equation (3.15) and Equation (3.16). Eventually, the distance between this ground point G' in the bird-eye image and the bottom edge of the bird-eye image will be estimated. We use the difference of the y-coordinate between G' in the bird-eye image and the bottom edge of the bird-eye image, which is 352, as shown in **Figure 47**. The pixel resolution of ground obtained in Section 3.4.3, which is 28.75, is also used. Therefore, the distance between G' and the bottom edge of the bird-eye image is estimated as $\frac{352}{28.75}$ = 12.243 meters.

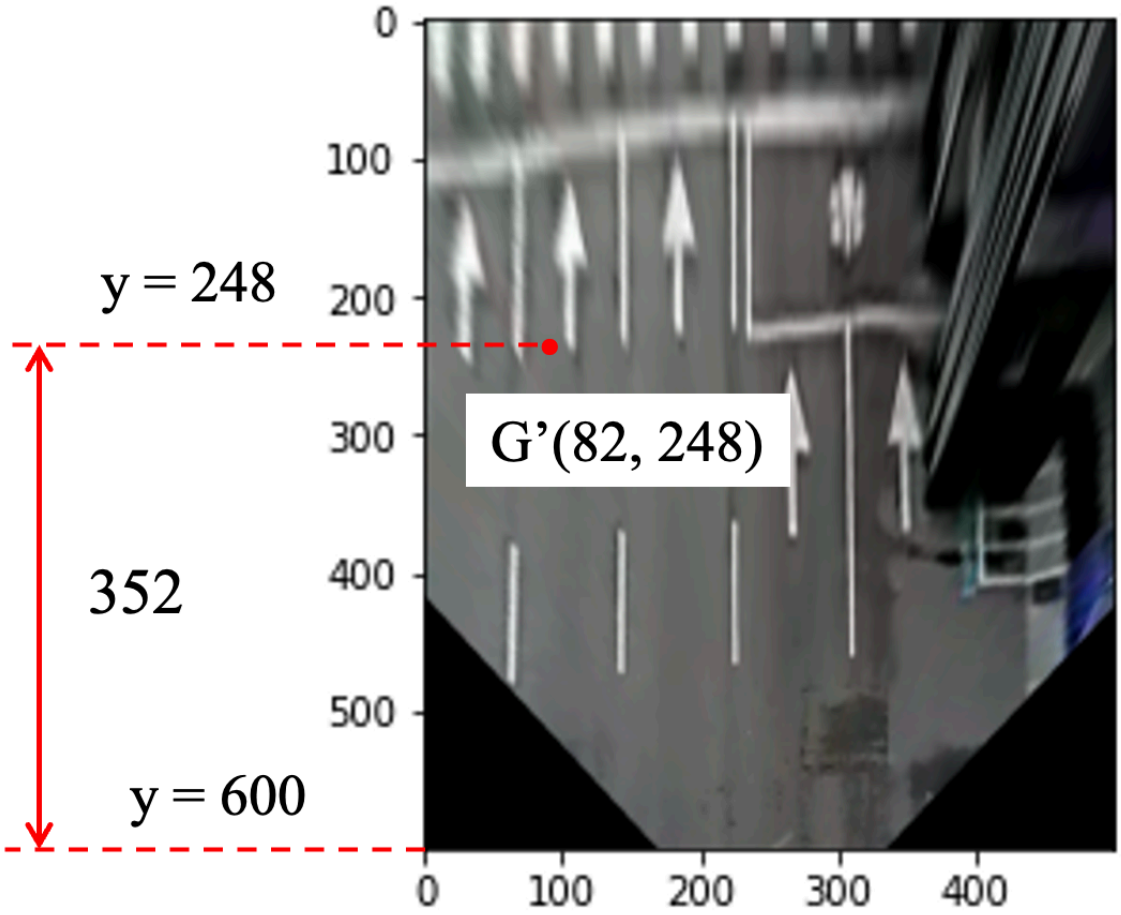


Figure 47: Y-coordinate between G' and the bottom edge of the bird-eye image

doi:10.6342/NTU202300097

CHAPTER 4

PERFORMANCE EVALUATION

In this chapter, we discuss the experimental results of our proposed method. In Section 4.1, the ground truth dataset and the performance metric are introduced. In Section 4.2, we exhibit the mean absolute percentage error (MAPE) of the distance from the objects on the ground to the camera in two environments. Besides, we show the MAPE of the distance from the objects with known height to the camera in two environments. Furthermore, we compare the results with a learning-based model [7]. In Section 4.3, we analyze the influence of objects without a precise height on the estimated distance.

4.1 Ground Truth Datasets & Settings

In our proposed method, we have to find the ground point of an object in the image according to its real height as the introduction in Section 3.3. Then, we estimate the distance between this ground point and the camera by perspective transform as the introduction in Section 3.4. First, we want to know the accuracy of estimating the distance from the ground points to the camera, since our method is based on well-developed algorithms. Therefore, we prepare a ground truth dataset to evaluate the performance. Also, another ground truth dataset is used to evaluate the performance of estimating the distance from the objects with known height to the camera.

4.1.1 Description of Ground Truth Dataset

Without precise sensors, such as Lidars, it is difficult to collect the distance information of the surrounding objects in a convenient way. Besides, there are no fixed

doi:10.6342/NTU202300097

objects that can be seen as a distance reference on the road. Fortunately, the traffic rules in Taiwan formulate the white dashed lines on the road as four meters, as the red bounding box shown in **Figure 48(a)**. We can regard these lines as the ground truth of the distance to evaluate the performance of estimating the distance between two endpoints of these lines by perspective transform. If we can estimate the difference of four meters between two endpoints of these lines precisely, it means the distance estimation between two points on the ground is accurate. Besides, we also collect the ground truth of distance from the objects on the ground to the camera by ourselves in the indoor environment for further analysis. However, the distance from the objects with known height to the camera cannot be obtained in the driving videos. Therefore, we also collect the ground truth dataset by ourselves to evaluate the performance of estimating the distance from the objects with known height to the camera. In the later introduction, we denote these two types of data as Dataset_1 and Dataset_2. Dataset_1 is used for evaluating the distance from the objects on the ground to the camera; Dataset_2 is used for evaluating the distance from the objects with a known height to the camera. Dataset_1 consists of image frames divided from driving videos and the indoor images collected by ourselves; Dataset_2 consists of indoor and outdoor images collected by ourselves.

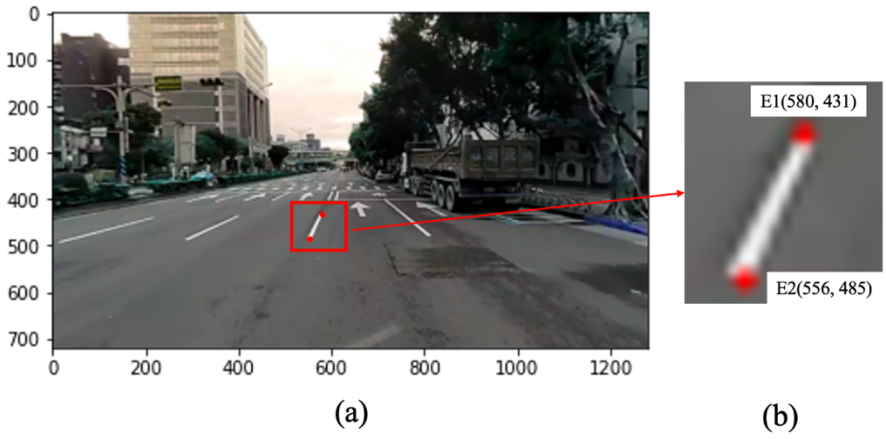


Figure 48: An example of 4m white dashed line

doi:10.6342/NTU202300097

In Dataset_1, we use the testing driving videos that are provided by the startup company OmniEyes to evaluate the performance of distance estimation on the ground. OmniEyes cooperates with Taiwan Pelican Express CO., Ltd. by equipping onboard cameras on the trucks to obtain street views from all over Taiwan. The locations of our testing videos are primarily in Taipei city. They are all collected during the daytime so that the white dashed lines can be distinguished easily. At first, we down sample the driving videos from 24fps to 5fps. Then, we correct the fish-eye distortion of these image frames by the procedure described in Section 3.2.1. We utilize the endpoints extraction algorithm that is described in Section 3.4.3 to label both endpoints of the white dashed lines from these image frames. In our evaluation, a total of 96 lines are labeled and each of them contains two endpoints. To clearly display the position of these lines, we use the midpoint of the two endpoints to represent a white dashed line on a blank image. The size of this blank image is equal to that of the image frames, as shown in **Figure 49**.

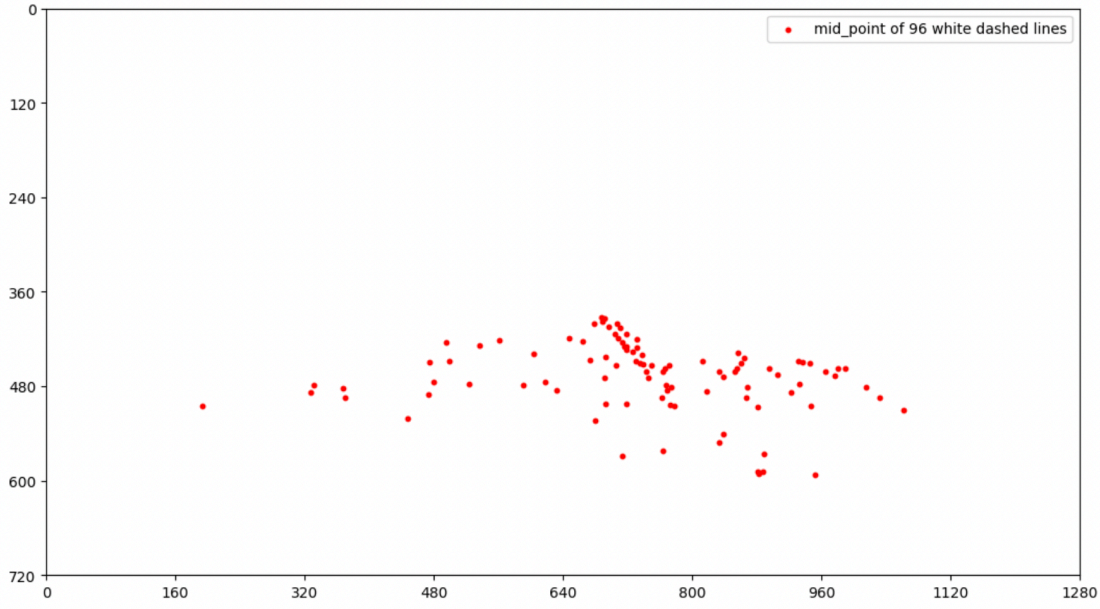


Figure 49: The position of 96 white dashed lines in the image

doi:10.6342/NTU202300097

In **Figure 49**, there are 96 red points. The reason why the y-coordinate of them ranges from 380 to 600 is that the y-coordinate of the road plane ranges from 330 to 720. Therefore, there are no red points on the upper half of the image. Moreover, the white dashed lines appearing at the bottom of the image frames are usually incomplete to be regarded as four meters. Thus, there are also no red points at the y-coordinate from 600 to 720. Furthermore, we divide the corresponding bird-eye image of the image frames into nine equal size regions. It can demonstrate the position of these 4m lines as well as the distance relation of them, as shown in **Figure 50**. R1 to R9 represent the nine equal size regions of the bird-eye image. Thus, the 96 white dashed lines will be classified into nine regions by their positions in the bird-eye image.

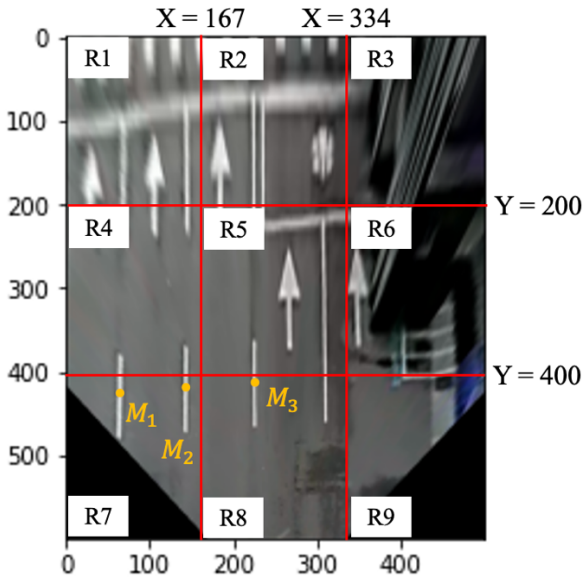


Figure 50: Nine equal size regions of the bird-eye image

Take **Figure 50** for example. M_1, M_2 and M_3 are the midpoints of three white dashed lines. Then, we will classify M_1, M_2 into R7 and M_3 into R8. We count the number of lines in each region and the statistical results are shown in **Table 6**. R1, R2, R3 belong to long distance from the camera; R4, R5, R6 belong to medium distance from

doi:10.6342/NTU202300097

the camera; R7, R8, R9 belong to close distance from the camera. As we can see from the table, most of the lines are concentrated at close distance and few lines at long distance. It is because of the low resolution of image frames to distinguish the two endpoints of the white dashed lines at long distance. Afterwards, we will evaluate the performance of estimating the distance between the two endpoints of these white dashed lines in each region by perspective transform and a learning-based model [8].

Region	Number of lines	
R1	0	
R2	3	
R3	0	Long distance
R4	0	
R5	23	
R6	2	Medium distance
R7	5	
R8	47	
R9	16	Close distance
Total	96	

Table 6: Statistics for the positions of 96 lines

In addition to verifying that the distance estimation between any two points is accurate, we also need to know that the distance estimation between a point and the camera is accurate. However, it is difficult to obtain the distance information between the camera and arbitrary points in the driving videos. Therefore, we collect the ground truth data of the distance on the ground in an indoor environment for further analysis.

doi:10.6342/NTU202300097

Simultaneously, we can also verify whether the performance of perspective transform is independent of the data by testing in different environments. We use the laser distance measuring instrument that is Leica DISTO D510 to collect the ground truth dataset. This instrument has a measurement accuracy of ±1mm. An example of an indoor image in Dataset_1 is shown in **Figure 51**.

Figure 51: An example of indoor image in Dataset_1

As we can see, we put a lot of objects on the ground and measure the distance from each object to the camera. A total of five images with an interval of 0.5m are taken. Moreover, these images contains many objects with distances measured in advance. The reason that we take the image with an interval is to simulate the video captured when the car was moving forward. It is similar to capturing image frames from videos. The total number of objects that we collect in the ground truth dataset is 217. We display the positions of these 217 objects in the image, as shown in **Figure 52**.

doi:10.6342/NTU202300097

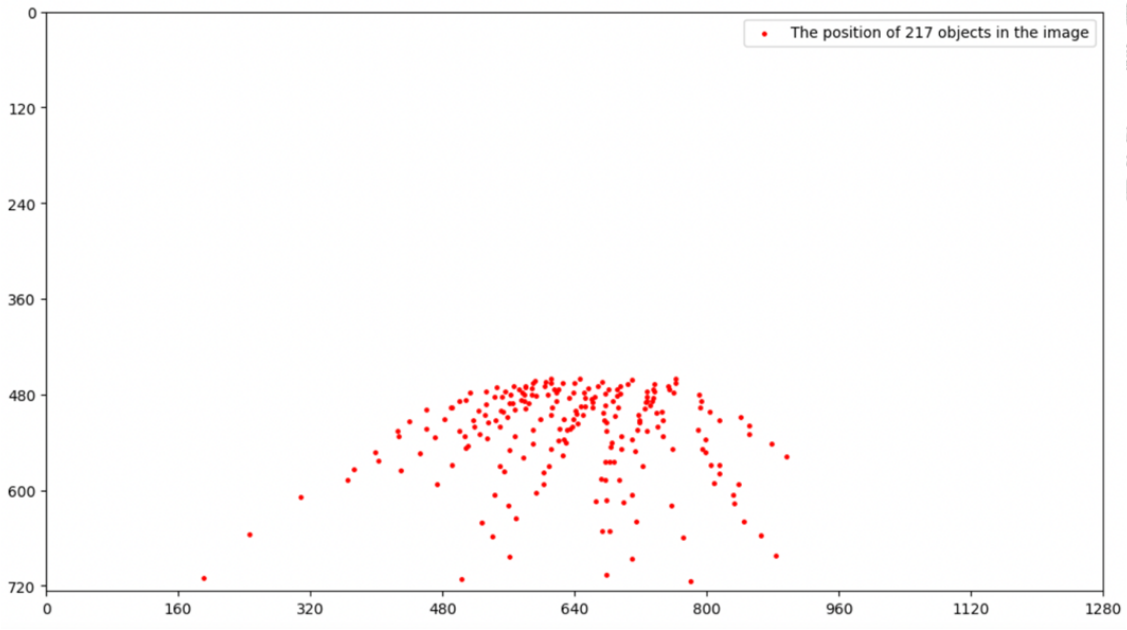


Figure 52: The position of 217 objects in the image

As we can see from the figure, the y-coordinate of these objects ranges from 460 to 714. The range of the distance from these 217 objects to the camera is from 2.5m to 10m. The distance distribution of these 217 objects is shown in **Figure 53**. Similarly, we implement the same process to classify these 217 objects into nine regions according to their location in the bird-eye image. The classification result is displayed in **Table 7**. The distribution of these ground truth data in the indoor environment is more uniform than the distribution of that in **Table 6**.

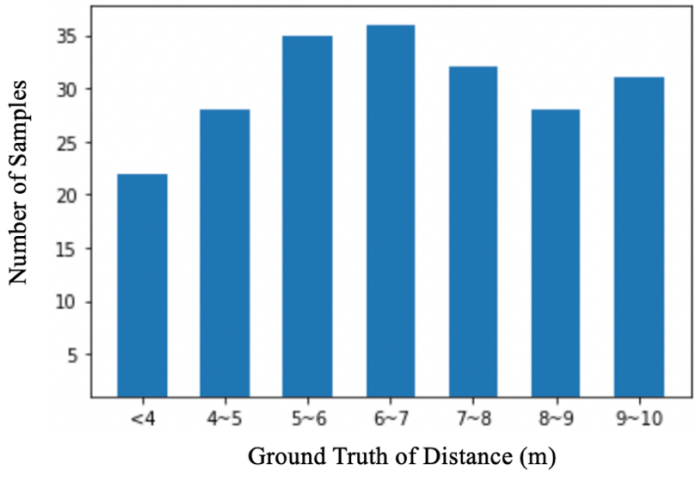


Figure 53: Distance distribution of 217 indoor objects

doi:10.6342/NTU202300097

Region	Number of lines	
R1	10	
R2	12	
R3	7	Long distance
R4	33	
R5	38	
R6	14	Medium distance
R7	24	
R8	67	
R9	12	Close distance
Total	217	

Table 7: Statistics for the positions of 217 objects indoors

In Dataset_2, we collect the ground truth dataset that contains the distance from the objects with known height to the camera by ourselves. Besides, the height of each object in this dataset is measured to evaluate the performance of our proposed method. The measuring instrument is identical to that used in Dataset_1. In order to prove our proposed method is data-independent, we use three different environments for evaluating. That is to say, the performance is not affected by different types of data. We collect the ground truth dataset in three different environments, which are indoor environment, outdoor daytime environment, and outdoor nighttime environment. The examples of three environments are shown in **Figure 54(a)**, **Figure 54(b)**, and **Figure 54(c)**. In these environments, using the fixed objects that are already in that position is more convenient than placing objects in the scene by ourselves. We measure the height of these fixed objects and the distance from the camera to them. For example, the points p_0, p_1, ..., p_8 in **Figure 54(a)** are railings, ceiling and walls. Moreover, we also utilize the street lights

doi:10.6342/NTU202300097

and roadside boards as our target objects, like points p_0, p_1, ..., p_4 in **Figure 54(b)** and **Figure 54(c)**.

(a) Indoor environment

(b) Outdoor daytime environment

(c) Outdoor nighttime environment

Figure 54: Three different environments in Dataset_2

doi:10.6342/NTU202300097

The total number of objects with measured height and the distance from the camera in three environments is 360. These objects are composed of 62 indoor objects, 149 outdoor objects during daytime, and 149 outdoor objects at night. However, since they are the objects with a known height above the ground, the distribution of the positions of these objects is concentrated in the upper half of the image, as shown in **Figure 55**. Furthermore, we also implement the same procedure to classify these objects into nine regions according to the position of their ground points in the bird-eye image. The statistic results are shown in **Table 8**.

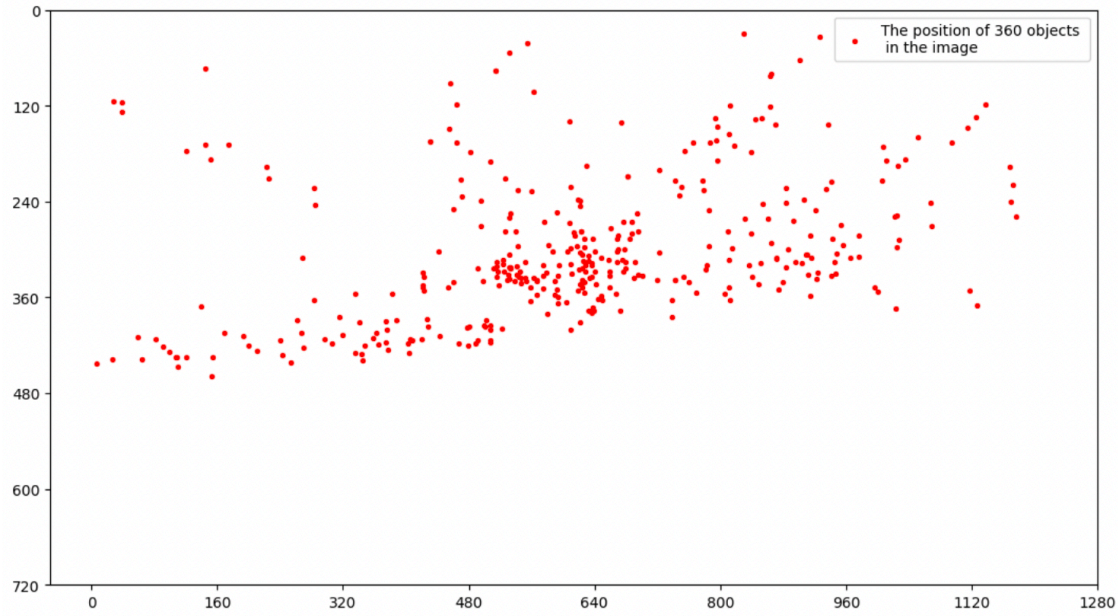


Figure 55: The position of objects in the image in Dataset_2

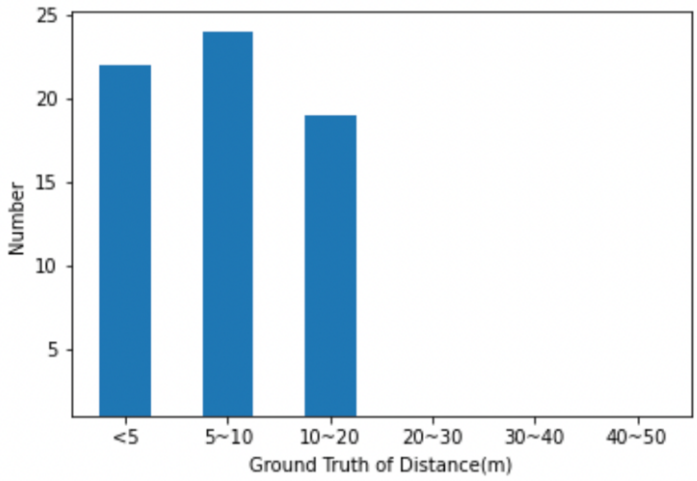
doi:10.6342/NTU202300097

Region	Number of lines	
R1	29	
R2	56	
R3	38	Long distance
R4	96	
R5	8	
R6	64	Medium distance
R7	38	
R8	10	
R9	21	Close distance
Total	360	

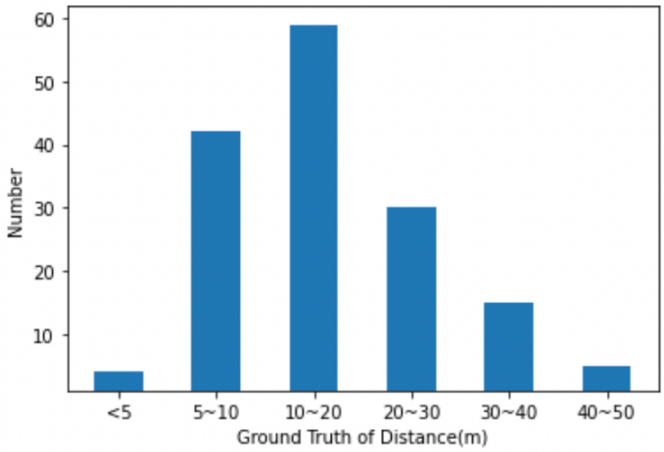
Table 8: Statistics for the positions of objects in Dataset_2

Furthermore, the distance distributions of the objects in three types of environments in Dataset_2 are demonstrated in **Figure 56(a)** and **Figure 56(b)**. Due to the resolution of the camera lens, objects beyond 50 meters cannot be clearly recognized in the image. Thus, the objects in Dataset_2 are farthest 48 meters from the camera. We also display the height distributions of the objects in three types of environments in Dataset_2, as shown in **Figure 57(a)** and **Figure 57(b)**. The height of these objects ranges from 0.81m to 5.26m and it is difficult to accurately measure higher objects by ourselves. The data of the outdoor daytime and outdoor nighttime environments are collected in the same place. The objects in the scene are also the same. Therefore, the distance distribution and the height distribution of the objects in these two types of environments are the same.

doi:10.6342/NTU202300097

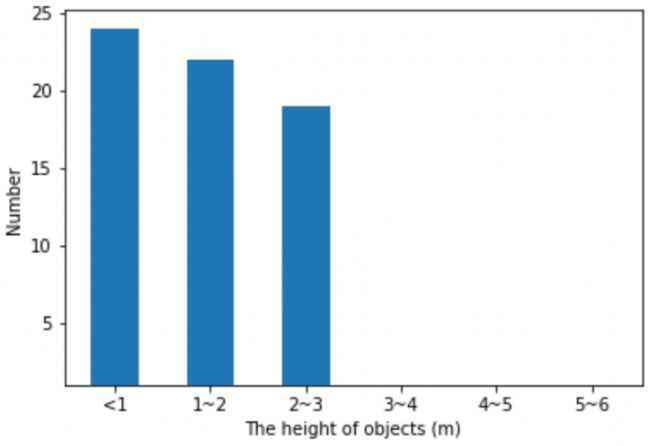


(a) Distance distribution of objects in indoor environment



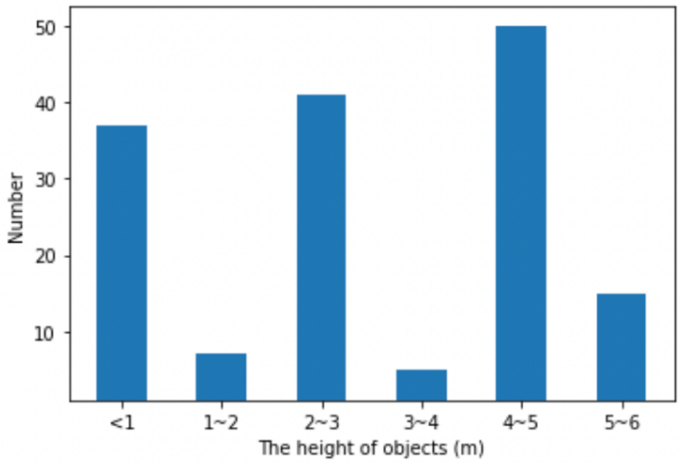
(b) Distance distribution of objects in outdoor daytime and nighttime environment

Figure 56: The distance distribution of objects in Dataset_2



(a) Height distribution of objects in indoor environment

doi:10.6342/NTU202300097



(b) Height distribution of objects in outdoor daytime and nighttime environment

Figure 57: The height distribution of objects in Dataset_2

4.1.2 Performance Metrics

Since there are various objects with different distances in our dataset, we cannot just consider the number of errors. We care about the percentage of errors in different ground truth of distance. Therefore, the mean absolute percentage error (MAPE) is used to evaluate the performance of our proposed method. The real distance between the target object i and the camera is denoted as GT_i, where i = 0, 1, 2, …, I. The estimated distance between the target object i and the camera is denoted as D_i, where i = 0, 1, 2, …, I. The notation I represents the total number of objects. Thus, we formulate the equation for computing the MAPE of the objects as follows:

$$\frac{1}{I}\sum_{i=0}^{I}\left|\frac{GT_i - D_i}{GT_i}\right| \tag{4.1}$$

An example for illustrating the performance metric is shown in **Figure 58**. The schematic diagram of the side view is also demonstrated in **Figure 59** for easier understanding.

doi:10.6342/NTU202300097

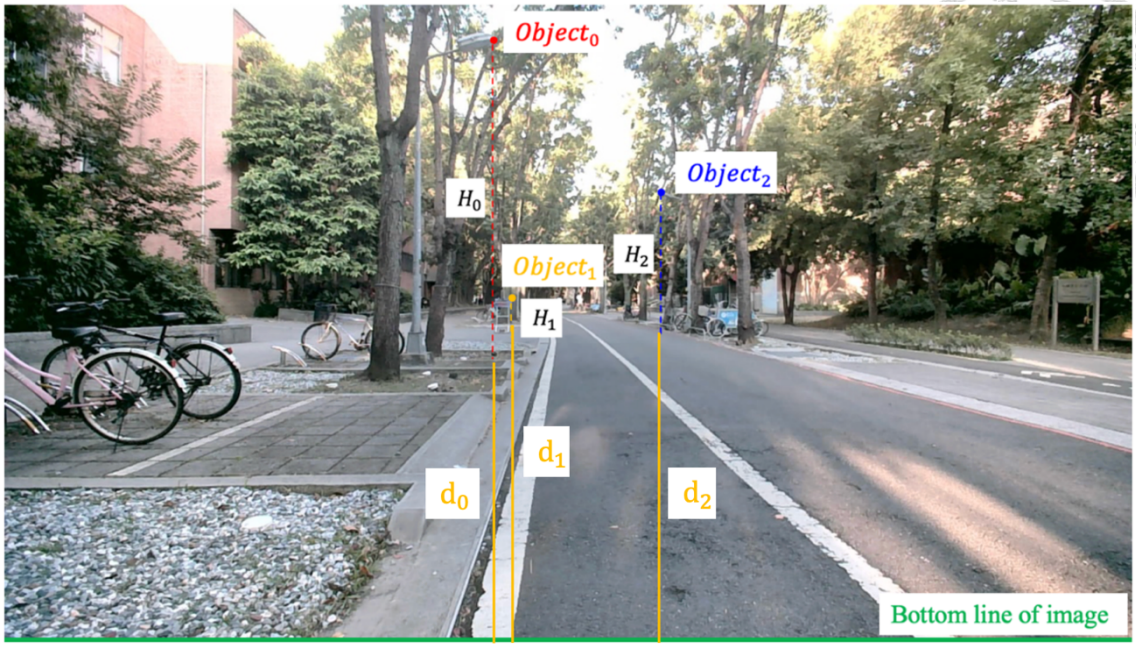


Figure 58: An example for illustrating the performance metric

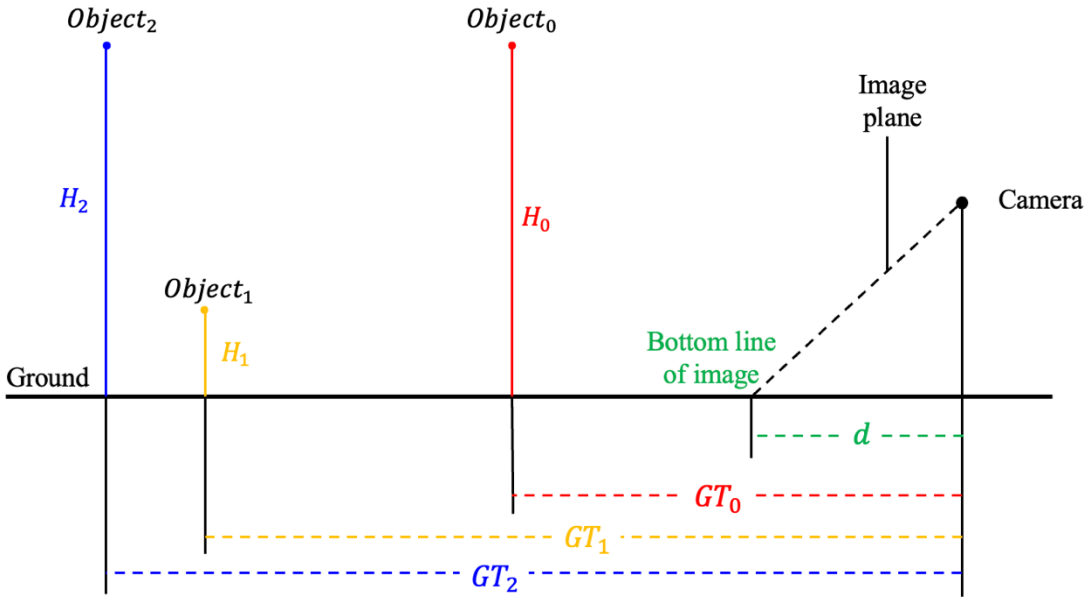


Figure 59: The side view of the example of performance metric

There are three target objects in **Figure 58**, which are $Object_0$, $Object_1$, and $Object_2$. Their heights are denoted as H_0, H_1, and H_2. We define the real distances between the camera and the objects as GT_0, GT_1 and GT_2, respectively. They are the

doi:10.6342/NTU202300097

longitudinal distances between the ground point of the camera and the ground point of the objects instead of the straight-line distances between the camera and the objects. Furthermore, in our proposed method, we can only estimate the distance between the target objects and the bottom line of the image, as d_0, d_1 and d_2 shown in **Figure 58**. Thus, we measure the real distance between the bottom line of the image and the camera. It is denoted as d, as shown in **Figure 59**. This distance d depends on the focal length and the height of the camera. When we evaluate the performance of our proposed method, the distance D_i in the Equation (4.1) is defined as follows:

$$D_i = d_i + \mathrm{d} \tag{4.2}$$

where d_i is the distance estimation result of the object i by our proposed method and d is the real distance between the bottom line of the image and the camera, which is a fixed distance measured in advance.

Suppose GT_0, GT_1, and GT_2 are measured as 8m, 15m, and 25m, respectively. The distance estimation results of them by our proposed method are 6m, 12m, and 26m, respectively, and the distance d is measured as 1.5m in advance. Then, we can compute the MAPE of this example by Equation (4.1) as follows:

$$\left(\left|\frac{8-(6+1.5)}{8}\right| + \left|\frac{15-(12+1.5)}{15}\right| + \left|\frac{25-(26+1.5)}{25}\right| \right) / 3 = 8\%$$

4.2 Experimental Results

4.2.1 Performance of Estimating the Distance of Objects in Dataset_1

In this section, we evaluate the performance of distance estimation on the ground using the perspective transform that we implement in our proposed method. Then, we compare the distance estimation results with a learning-based method [8]. Since our proposed method is based on perspective transform which is an existing method for

doi:10.6342/NTU202300097

estimating the ground distance, the performance of perspective transform is very important.

4.2.1.1 Predict the Length of White Dashed Lines on the Road

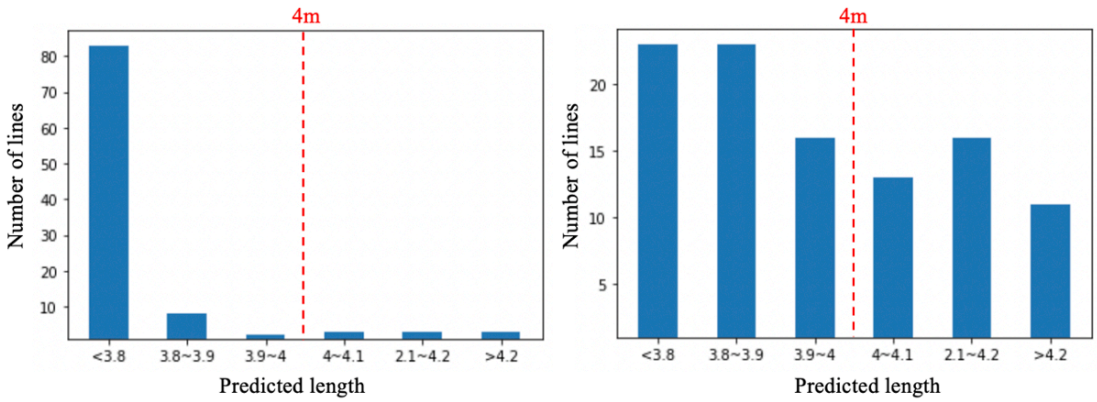
At first, we use both endpoints of the white dashed lines in Dataset_1 to evaluate the performance of estimating the distances between two points on the ground. The distance estimated by our method is perpendicular to the bottom line of the image. The white dashed lines are also perpendicular to the bottom line of the image when transforming to a bird-eye image, as shown in **Figure 47**. Thus, we can calculate the length of them by directly subtracting the estimated distance of two endpoints. We also use the same procedure to predict the length of these lines by a depth model [8] which is a learning-based method. As previously mentioned, Equation (4.1) is used to compute the mean absolute percentage error (MAPE). Moreover, we separately calculate the MAPE of the nine regions as well as the overall MAPE. The comparison results of perspective transform and learning-based depth model are shown in **Table 9**.

Region	Number of lines	MAPE of Perspective Transform	MAPE of Depth model [8]	
R1	0	0	0	
R2	3	2.031%	16.292%	
R3	0	0	0	Long distance
R4	0	0	0	
R5	23	5.665%	13.744%	
R6	2	2.686%	7.356%	Medium distance
R7	5	6.715%	8.823%	
R8	47	5.309%	11.368%	
R9	16	4.587%	12.724%	Close distance
Total	96	5.19%	12.1%	

Table 9: Results of predicting the length of white dashed lines on the road

doi:10.6342/NTU202300097

As we can see from the results, it is observed that the performance of predicting the length of white dashed lines by perspective transform is stable in nine regions. That is, the MAPE in each region does not change significantly. Besides, the performance of perspective transform is better than that of depth model. This may be due to the different street-view in Taipei from the training datasets of the depth model, so the performance is poor. The performance of learning-based methods is usually related to whether the input data is similar to the training dataset. It is also the main disadvantage of the depth model. On the contrary, perspective transform is a geometry-based method that uses the imaging principle of the camera as basis. Thus, the performance is not affected by the various scenarios. Besides, we demonstrate the distribution of the predicted length by both methods, as shown in **Figure 60**. We can see that most of the predicted length by the depth model are shorter than the actual length.



(a) Depth model [8] (b) Perspective transform

Figure 60: Distribution of the predicted length of white dashed lines

4.2.1.2 Predict the Distance of Objects on the Indoor Ground

In the previous experiments, we evaluate the performance of predicting the length of white dashed lines on the road by directly subtracting the estimated distances of two

doi:10.6342/NTU202300097

endpoints. However, these experiments can only demonstrate that the relative distance between two points is accurate. The accuracy is defined as the difference between the estimated length of a white dashed line and four meters. This is not what we expected. Therefore, we conduct another experiment to verify that the distance from a ground point to the camera is accurate. We use the ground truth dataset collected by ourselves in the indoor environment as described in **Figure 51**. The performance of the distance estimation results by depth model and perspective transform are listed in **Table 10**.

Region	Number of objects	MAPE of Perspective Transform	MAPE of Depth model [8]	
R1	10	4.169%	34.241%	
R2	12	3.687%	31.787%	
R3	7	4.296%	37.659%	Long distance
R4	33	3.352%	28.855%	
R5	38	3.265%	26.331%	
R6	14	3.227%	28.967%	Medium distance
R7	24	2.342%	30.701%	
R8	67	3.103%	23.873%	
R9	12	2.652%	26.224%	Close distance
Total	217	3.188%	27.635%	

Table 10: Results of predicting the distance of objects on indoor ground

As we can see, the results of both methods at long distance have larger errors than those at close distance. The performance of the depth model is very poor and different from the result in **Table 9**. It greatly demonstrates the disadvantage of the learning-based methods. That is, the learning-based methods usually have poor performance in an

doi:10.6342/NTU202300097

environment which is different from the training dataset. It is difficult for the depth model to distinguish the objects in indoor environment. However, perspective transform is a geometry-based method, so its performance of distance estimation is similar to that of the outdoor experiment in **Table 9**. Moreover, these experimental results also prove perspective transform is more data-independent than learning-based methods. Therefore, perspective transform is suitable for being the basis of our proposed method.

4.2.2 Performance of Estimating the Distance of Objects in Dataset_2

After evaluating the performance of estimating the distance from the objects on the ground to the camera, we know that perspective transform is suitable for being the basis of our proposed method. Then, we will focus on estimating the distance from the objects with a known height to the camera by our proposed method in this section. Dataset_2 that we collect by ourselves is used to evaluate the performance. The environments in our dataset can be divided into three types, which are indoor environment, outdoor daytime environment, and outdoor nighttime environment, as previously shown in **Figure 54**. We implement the same procedure as previous to classify the objects in the three environments into nine regions for separate analysis. The overall analysis of the entire dataset is conducted as well. We also use Equation (4.1) to calculate the mean absolute percentage error (MAPE) of our proposed method and compare the results with a learning-based method. The experimental results of estimating the distance from the objects to the camera in three environments are demonstrated in **Table 11**, **Table 12**, and **Table 13**, respectively. Moreover, the overall comparison results between two methods are shown in **Table 14**.

doi:10.6342/NTU202300097

Region	Number of objects	MAPE of Proposed method	MAPE of Depth model [8]	
Type1 -R1	7	6.878%	20.632%	
Type1 -R2	4	5.691%	21.564%	
Type1 -R3	2	10.314%	23.166%	Long distance
Type1 -R4	20	6.32%	29.85%	
Type1 -R5	2	12.188%	36.295%	
Type1 -R6	10	8.15%	37.012%	Medium distance
Type1 -R7	8	7.244%	31.06%	
Type1 -R8	2	5.926%	34.417%	
Type1 -R9	7	9.572%	42.259%	Close distance
Type1-Total	62	7.429%	31.127%	

Table 11: Results of indoor environment in Dataset_2

Region	Number of objects	MAPE of Proposed method	MAPE of Depth model [8]	
Type2 -R1	11	9.127%	37.926%	
Type2 -R2	26	10.576%	41.638%	
Type2 -R3	18	8.261%	40.157%	Long distance
Type2 -R4	38	8.864%	32.531%	
Type2 -R5	3	7.896%	36.154%	
Type2 -R6	27	7.753%	30.875%	Medium distance
Type2 -R7	15	8.429%	34.728%	
Type2 -R8	4	7.388%	30.291%	
Type2 -R9	7	7.475%	33.628%	Close distance
Type2-Total	149	8.739%	35.425%	

Table 12: Results of outdoor daytime environment in Dataset_2

doi:10.6342/NTU202300097

Region	Number of objects	MAPE of Proposed method	MAPE of Depth model [8]	
Type3 -R1	11	11.441%	51.738%	
Type3 -R2	26	9.657%	47.634%	
Type3 -R3	18	9.826%	53.457%	Long distance
Type3 -R4	38	9.783%	49.825%	
Type3 -R5	3	8.166%	56.231%	
Type3 -R6	27	9.585%	39.783%	Medium distance
Type3 -R7	15	8.634%	38.531%	
Type3 -R8	4	9.733%	31.392%	
Type3 -R9	7	7.292%	45.736%	Close distance
Type3-Total	149	9.586%	46.508%	

Table 13: Results of outdoor nighttime environment in Dataset_2

Region	Number of objects	MAPE of Proposed method	MAPE of Depth model [8]
Type1-Total	62	7.429%	31.127%
Type2-Total	149	8.739%	35.425%
Type3-Total	149	9.586%	46.508%
Total	360	8.864%	39.272%

Table 14: Overall comparison results of Dataset_2

From the results, it is observed that the performance in three various environments by our proposed method is more stable than that of the depth model. That is to say, our proposed method has similar performance in different environments. It also strengthens

doi:10.6342/NTU202300097

the data-independent property of the geometry-based method. Besides, the overall performance of our proposed method outperforms that of the depth model by 30.408%. Compared to the previous experiment results in **Table 10**, the MAPE of estimating the distance of objects in Dataset_2 is slightly higher than that of estimating the distance of objects in Dataset_1 by perspective transform. It is probably because of the errors caused by the procedure of estimating the ground point of any object with a known height as introduced in Section 3.3. On the other hand, the depth model has very poor performance in three types of environments. It is most likely because the dataset we used and the training dataset the depth model used are so different. Therefore, the model cannot recognize the scenario we used. Especially in **Table 13**, the depth model has the worst performance among these environments. This may be caused by the very different environment at night. From the distance estimation results, the reliability of the depth information is very low. It is also one of the main disadvantages of learning-based methods. Furthermore, the depth model does not take the height of objects into consideration, so it is difficult for the depth model to estimate the distance from an object without height information to the camera.

Additionally, we classify the objects in Dataset_2 into six classes according to the ground truth of distance rather than the position of them in the bird-eye image. The ground truth of distances is classified into six classes: 0m~5m, 5m~10m, 10m~20m, 20m~30m, 30m~40m and 40m~50m. We conduct the same procedure to analyze the performance of our proposed method and the depth model according to these six classes. The experimental results are shown in **Table 15**. As we can see from the table, the MAPE is not greatly affected by the difference in the distance even though the distances from the objects to the camera are up to 50 meters. It also demonstrates the advantage of the geometry-based methods. That is, the estimation results of proposed method are more

doi:10.6342/NTU202300097

stable than those of depth model due to the entire process which is comprised of the imaging principle of camera. In contrast, the MAPE of the depth model at different distances flutuates more significantly than that of proposed method.

Ground Truth of Distance	Number of objects	MAPE of Proposed method	MAPE of Depth model [8]
0m~5m	27	6.92%	34.738%
5m~10m	105	8.674%	31.652%
10m~20m	134	8.951%	42.534%
20m~30m	58	10.306%	47.125%
30m~40m	28	7.539%	38.359%
40m~50m	8	10.635%	46.215%
Total	360	8.864%	39.272%

Table 15: Performance of two methods at different distances

4.3 Estimating the Distance of Objects without a Precise Height

In our method, we utilize the real height of the target object and the coordinate of it to estimate the distance from this object to the camera. However, we want to know how much the influences are when estimating the distance of an object without a precise height. Therefore, three target objects with known heights in Dataset_2 are selected to simulate our proposed method, as shown in **Figure 61**. It is one of the continuous images which are captured at different intervals on the same road. In this figure, $Object_0$ is a 2.81m height object. $Object_1$ and $Object_2$ are 5.25m height objects. The reason for choosing these objects as our simulation targets is because of the similar height to the signs on the

doi:10.6342/NTU202300097

road. For instance, the height of traffic signs is mostly between 4.6m and 5.6m and the height of pedestrian signs is mostly between 2.1m and 3m. We use these images to label the actual coordinate of these three target objects and their corresponding real distance. By labeling these objects, we can easily evaluate the errors between the theoretically calculated distance values and the real distance values in our experiment.

Figure 61: Three target objects we selected in Dataset_2

At the beginning, we use the proposed method to calculate the distance from the object with a height of 5.25m to the camera. Since objects that are too far from the camera will not be obtained, we consider the object appears at the y-coordinate 0 to 300. The simulation result of the 5.25m height object is shown in **Figure 62**. As we can see, the blue curve is the results of distance estimated by our proposed method at different y-coordinates in the image. The P_1, P_2, ..., P_6 in the figure are the real distances of the 5.25m height object we label from six image frames in Dataset_2. We can observe that the errors between the simulation results and the real distances range from -10.886% to

doi:10.6342/NTU202300097

12.265%. Since it is difficult to know the real distance of each coordinate, we only label six points to represent the errors between the theoretical values and the actual values.

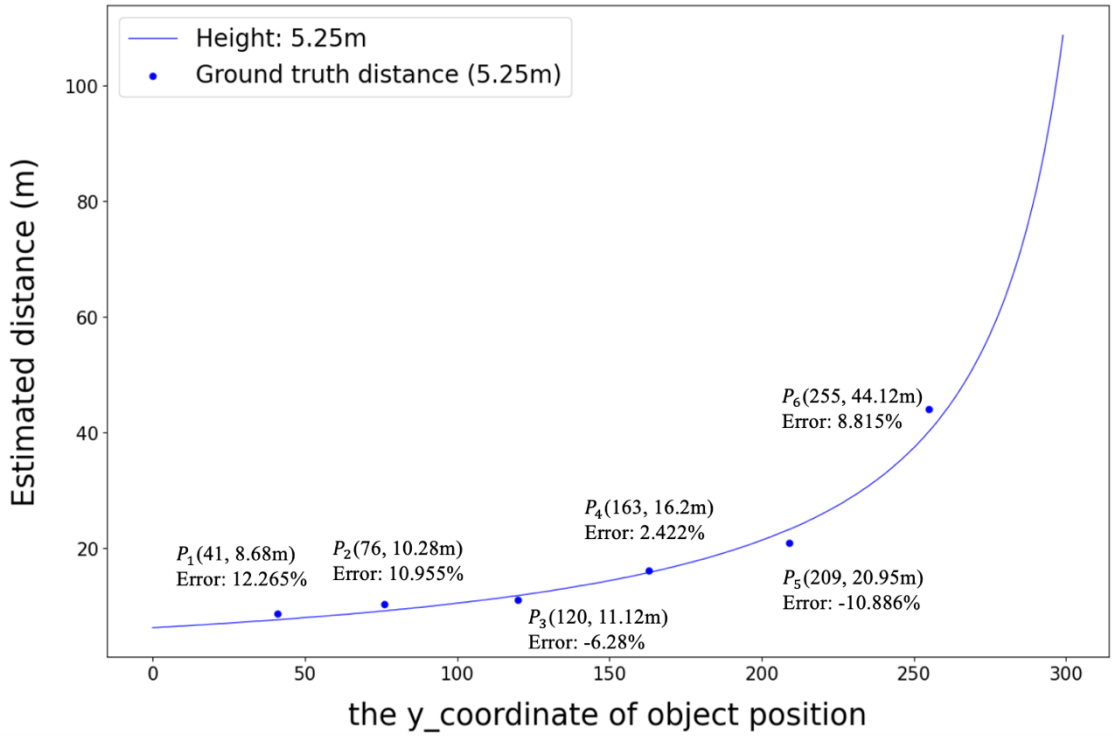


Figure 62: Distance estimation of 5.25m height object

We want to know how much the height information affects the distance estimation results. Thus, we analyze the influence of the $\pm 10\%$ height errors on the distance estimation of the 5.25m height object. Therefore, the error between the distance of the 5.25m height object and the distance of 5.25m+10% height object is denoted as E1; the error between the distance of 5.25m height object and the distance of 5.25m-10% height object is denoted as E2. Furthermore, we define the error as the ratio of distance difference with height error to the distance calculated without the height error. For instance, if the distance calculated by a 5.25m height object is 10m and the distance calculated by a 5.25m+10% height object is 12m, the error can be estimated as $\left|\frac{(12-10)}{10}\right| = 20\%$. Then, we plot E1 and E2 at different y-coordinates on a graph to observe the errors, as shown in **Figure 63**.

doi:10.6342/NTU202300097

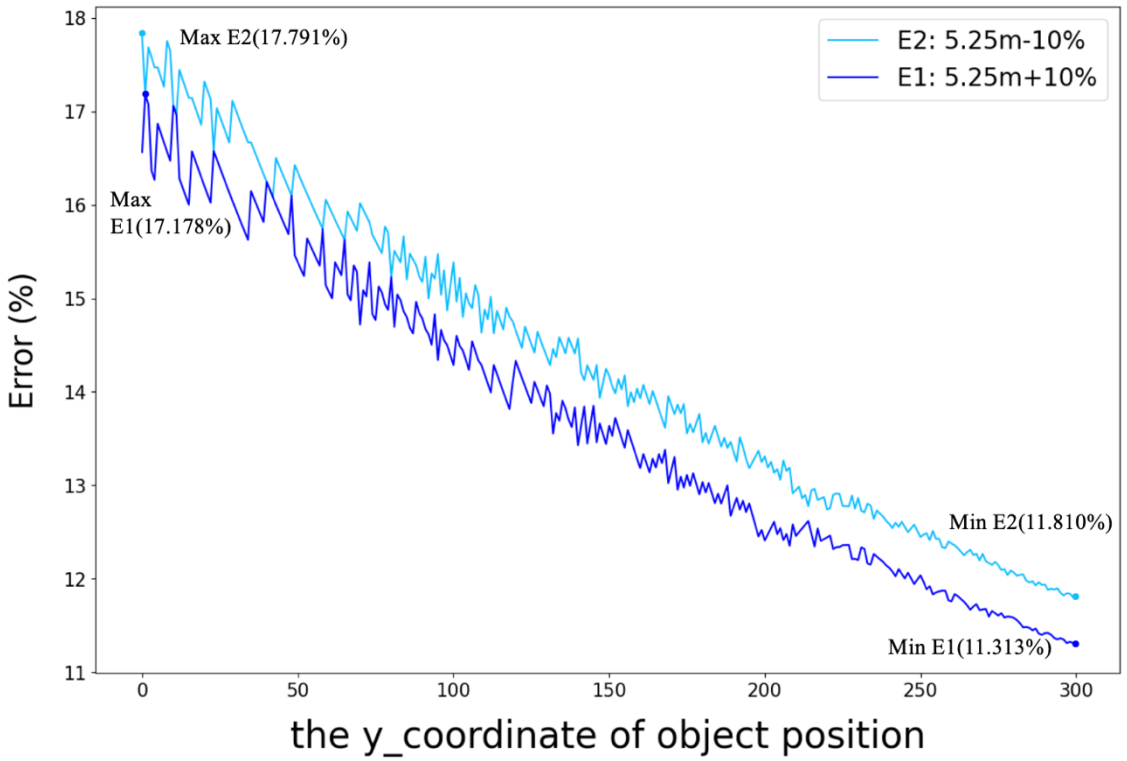


Figure 63: E1 and E2 at different y-coordinate

As we can see from the results, the maximum value of E1 reaches 17.178% and E2 reaches 17.791% when the y-coordinate is small. On the other hand, the minimum value of E1 is 11.313% and E2 is 11.810% when the y-coordinate is large. The reason why the error curve is jagged is that the difference of y-coordinate calculated in Section 3.3.2 by each object with different heights at the same position is different. Moreover, a float number will be automatically rounded into an integer in the coordinate system. Even if the increments of the y-coordinate of objects with different heights are the same, the increments of their calculated ground points will not be the same. Therefore, the increments of the distance between different ground points to the camera are not estimated identically. For exmaple, the distance of a 5.25m+10% height object will increase every three y-coordinates, such as 6m, 6m, 6m, 7m, 7m, 7m in six consecutive y-coordinates; the distance of a 5.25m height object will increase every two y-coordinates, such as 5m, 5m, 6m, 6m, 7m, 7m in six consecutive y-coordinates. The errors in six consecutive y-

doi:10.6342/NTU202300097

coordinates will be 20%, 20%, 0%, 16.7%, 0%, 0%. Therefore, the error curve is not as smooth as the distance curve in **Figure 62**.

In addition, we can observe that the smaller the y-coordinate is, the greater the error of the height affecting the distance will be. This is because the appearance of the object will become larger in the image when it is closer to the camera. Thus, the height difference will have a greater impact on the distance estimation at the close distance. On the contrary, objects with different heights at the long distance will look similar in the image. Afterwards, we use the pixel resolution of 5.25m height object to deduce the 2.81m height object and estimate the distance of 2.81m height object by our proposed method. The simulation results are shown in **Figure 64**.

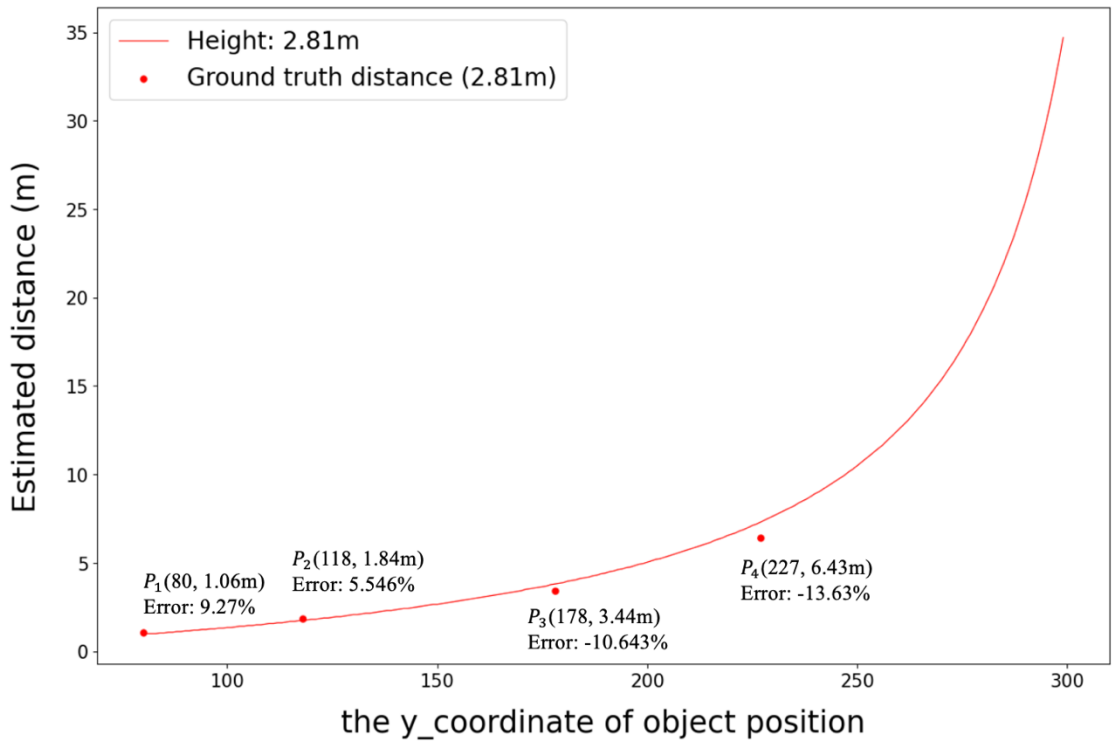


Figure 64: Distance estimation of 2.81m height objects

When the y-coordinate of a object is too small, the calculated distance of this object will be close to 0. Besides, the 2.81m height object is rarely captured in the image at the close distance when driving on the road. Therefore, we consider the y-coordinates from

doi:10.6342/NTU202300097

80 to 300, where 80 is the y-coordinate of an object appearing in the labelled dataset. The red curve is the distances estimated by our proposed method. In order to know how many the errors are from the actual situation, we also mark four points with known distance from Dataset_2 as in the previous part. In the **Figure 64**, P_1, P_2, P_3 and P_4 are the real distances of the 2.81m height object we labeled from four image frames. We can observe that the errors between the simulation results and the real distances range from -13.63% to 9.27%. The same as the previous part, we analyze the influence of the height error on the distance estimations of a 2.81m height object. Thus, the errors between the distances of the 2.81m height object and the distances of the 2.81m+5% height object are denoted as E3; the errors between the distances of the 2.81m height object and the distances of the 2.81m-5% height object are denoted as E4. Finally, we plot E3 and E4 on a graph in the same way as plotting E1 and E2, as shown in **Figure 65**.

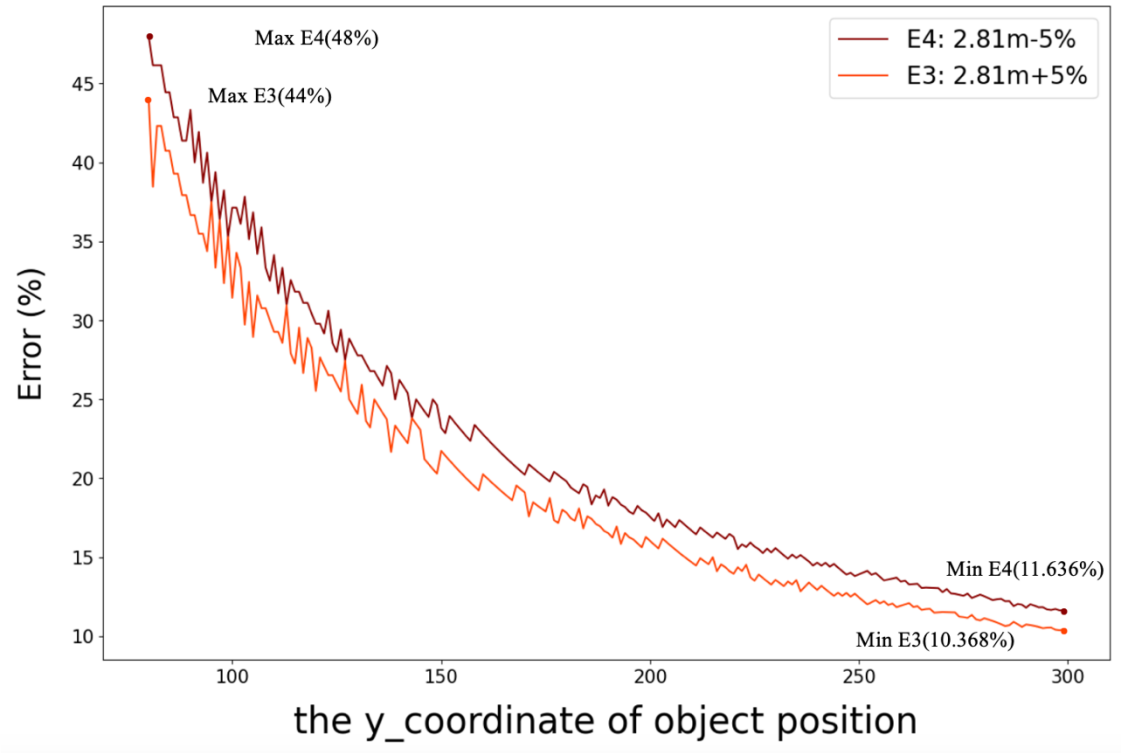


Figure 65: E3 and E4 at different y-coordinate

doi:10.6342/NTU202300097

From the results, the maximum value of E3 reaches 44% and E4 reaches 48% when the y-coordinate is small. On the other hand, the minimum value of E3 is 10.368% and E4 is 11.636% when the y-coordinate is large. This is also the reason for choosing a ±5% error of height of a 2.81m height object instead of ±10%. A ±5% error of height is enough to make the performance of distance estimation for a 2.81m height object very poor. Objects with such height have a relatively small tolerance for error of height comparing with the 5.25m height objects. Same as the 5.25m height objects, the smaller the y-coordinate is, the greater the error of the height affecting the distance estimation will be.

Thus, we can conclude that the performance of our proposed method is easily affected by the height of objects, especially with lower heights objects. Even though there is such a disadvantage, we can still put our proposed method into practice in the actual road environment in Taiwan. There are many objects on the streets that can be used in our method. For example, the traffic lights and some traffic signs for assisting driving are required to be set at heights of 4.6m to 5.6m. Most of the street lights on the road are higher than 8m and the height of the street lights on the highway is fixed at 12m. Besides, some signs are installed on the gantry. The height of gantry is at least 4.9m on the highway, and is at least 4.6m on the general road. According to the experiments, the performance of our method for the objects of these heights is acceptable. We can also use our method in autonomous driving. For example, our method helps drivers determine whether to slow down or accelerate to pass the traffic light. In the navigation system, the distance estimation can also be used to remind the driver to turn a few meters away. Moreover, we can use the distance of objects for map matching as well as we can update the information on the map, such as the status of traffic light. Furthermore, we can also use our method with other information, such as, map and direction, to achieve vehicle localization.

doi:10.6342/NTU202300097

CHAPTER 5

CONCLUSIONS

In this thesis, we propose a low-cost and real-time distance estimation using a monocular camera. According to the experimental results, using perspective transform as the basis of our proposed method for estimating the distance on the ground is more suitable than using the depth model. Moreover, the performance of estimating the distance of objects with known actual heights in three different environments by our proposed method is better and more stable than that of the depth model. The mean absolute percentage error of our method is 8% which outperforms that of the depth model by 30%. These results show that our proposed method can be used to estimate the distance of objects with known heights and is data-independent.

However, our method contains an inherent disadvantage in that the performance is sensitive to the height of the objects, especially with lower heights objects. Simulation results show that for a 5.25m height object, a 10% height error may cause a maximum 17% distance estimation error. On the other hand, for a 2.81m height object, a 5% height error may cause a maximum distance estimation error 48%. Therefore, this disadvantage will be more obvious on objects of lower height, namely, less tolerance for height errors on objects of lower height. Even though there is such a disadvantage, we can still put our proposed method into practice in the actual road in Taiwan or use in autonomous driving. The traffic rules in Taiwan formulate the height of traffic signs on the road within a certain range. From the experimental results, the performance of using our method on those objects with higher than 4.6m is acceptable.

doi:10.6342/NTU202300097

REFERENCES

[1] Thalen, J.P. (2006). *ADAS for the Car of the Future* (Bachelor's thesis, University of Twente).

[2] Howard, I. P., & Rogers, B. J. (1995). *Binocular vision and stereopsis*. Oxford University Press, USA.

[3] Koller, D., Luong, Q. T., & Malik, J. (1994, October). Using binocular stereopsis for vision-based vehicle control. In *Proceedings of the Intelligent Vehicles' 94 Symposium* (pp. 237-242). IEEE.

[4] Uttamchandani, D. (Ed.). (2013). *Handbook of MEMS for wireless and mobile applications*. Elsevier.

[5] "What is lidar?" https://velodynelidar.com/what-is-lidar/

[6] Khader, M., & Cherian, S. (2020). An introduction to automotive lidar. *Texas Instruments*.

[7] Atapour-Abarghouei, A., & Breckon, T. P. (2019, September). Monocular segment-wise depth: Monocular depth estimation based on a semantic segmentation prior. In *2019 IEEE International Conference on Image Processing (ICIP)* (pp. 4295-4299). IEEE.

[8] Lee, J. H., Han, M. K., Ko, D. W., & Suh, I. H. (2019). From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326v6*.

[9] Garg, R., Bg, V. K., Carneiro, G., & Reid, I. (2016, October). Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision* (pp. 740-756). Springer, Cham.

doi:10.6342/NTU202300097

[10] Godard, C., Mac Aodha, O., & Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 270-279).

[11] Stein, G. P., Mano, O., & Shashua, A. (2003, June). Vision-based ACC with a single camera: bounds on range and range rate accuracy. In *IEEE IV2003 intelligent vehicles symposium. Proceedings (Cat. No. 03TH8683)* (pp. 120-125). IEEE.

[12] Gat, I., Benady, M., & Shashua, A. (2005). A monocular vision advance warning system for the automotive aftermarket. *SAE transactions*, 403-410.

[13] Qi, S. H., Li, J., Sun, Z. P., Zhang, J. T., & Sun, Y. (2019, February). Distance estimation of monocular based on vehicle pose information. In *Journal of Physics: Conference Series* (Vol. 1168, No. 3, p. 032040). IOP Publishing.

[14] Bao, C., Chen, C., Kui, H., & Wang, X. (2019, June). Safe driving at traffic lights: An image recognition based approach. In *2019 20th IEEE International Conference on Mobile Data Management (MDM)* (pp. 112-117). IEEE.

[15] "道路交通標誌標線號誌設置規則，第158條" https://law.moj.gov.tw/LawClass/LawSingle.aspx?pcode=K0040014&flno=158

[16] Zhang, Z. (1999, September). Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the seventh IEEE international conference on computer vision* (Vol. 1, pp. 666-673). IEEE.

[17] Ajsmilutin. (2017). CarND-Advanced-Lane-Lines. Github. https://github.com/ajsmilutin/CarND-Advanced-Lane-Lines

[18] *Camera Calibration and 3D Reconstruction.* https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html#ga69f2545a8b62a6b0fc2ee060dc30559d

doi:10.6342/NTU202300097