國立臺灣大學電機資訊學院資訊網路與多媒體研究所

碩士論文

Graduate Institute of Networking and Multimedia

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

基於深度可調三維模型以產生車輛環周影像之研究

On Generating Vehicle Surrounding Images Based on Depth-Adaptive 3D Model

葉彥廷

Yen-Ting Yeh

指導教授：洪一平博士

Advisor: Yi-Ping Hung, Ph.D.

中華民國 103 年 7 月

July, 2014

國立臺灣大學碩士學位論文
口試委員會審定書

基於深度可調三維模型以產生車輛環周影像之研究
On Generating Vehicle Surrounding Images Based on Depth-adaptive 3D Model

本論文係葉彥廷君（學號 R01944005）在國立臺灣大學資訊網路與多媒體研究所完成之碩士學位論文，於民國一百零三年七月卅一日承下列考試委員審查通過及口試及格，特此證明

口試委員：

洪一平 （簽名）

（指導教授）

黃彥男　　　陳祝嵩

徐宏民

所　長：逢愛君

致謝

首先要感謝洪老師在這幾年的指導，他充分支持我的研究且給予我極大的自由發揮空間，能關鍵地找到我研究的盲點並提示我解決的方向。感謝在交大任教的陳永昇教授，在這個研究的一開始給我很多相關知識，並在研究的過程中，抽空和我討論。感謝交大的俊康，和我一同做這個研究題目，有了你錄製的影片與校正前處理，我才能完成這篇論文。

感謝這幾年來帶我的大源學長，和學長的討論中，讓我學到做研究的基礎，並激發我許多不同的想法。因為和學長的合作，讓我有機會參加人機互動領域中很好的國際研討會，參與了整個研討會，看到許多特別且重要的研究成果，使我對研究的熱誠更加深厚。感謝實驗室的大學長韋哥，每次和韋哥的討論，都可以讓我感受到非常認真的研究態度與學到不同的實驗技巧。

再來要謝謝實驗室這屆的夥伴們，偉傑、可昀、儒涵、育淳、雪翔、博翔、煬昇和魏驍，這兩年的幫助與支持，讓我可以度過許多深夜博理的時光。感謝博理五樓的倉愷，在較為煩悶的碩二時間，和我一起在大台北中吃吃喝喝，紓解壓力。

感謝身在英國的彥廷學長與在美國的彥伶學姊，和你們愉快的合作與有趣的多媒體系統，使我學到如何與他人合作、溝通的技巧，並提供我研究遇到難題時，能先躲到較為輕鬆的研究題目中使頭腦清醒，再面對自己的問題。

最後要感謝我的家人，給予我許多體諒與支持，讓我能夠專心地面對繁雜的研究生生涯。

中文摘要

車輛駕駛人常因為車體本身的視覺盲點而造成交通意外，現有的車輛輔助系統常利用警示燈號或聲響來提供駕駛車輛環周狀況。由於鏡頭的售價漸趨便宜，目前車廠也將攝影機架設於車輛周圍，並將攝取到的影像或是車輛上方的鳥瞰影像顯示於車內，作為安全駕駛輔助視訊系統。相對於固定的鳥瞰視角，我們提出了一使用第三人稱視角的車輛環周監視系統—“天使之眼”。我們接合車輛四周的魚眼攝影機所擷取的影像，將環周影像投影在一混合式3D模型中，並提供不同視角的影像，使得駕駛者可自由轉換視角。

但因系統未知車輛周圍前景物的深度，在視覺呈現上，會有前景物的影像扭曲或是鬼影的現象產生。所以我們將深度攝影機加裝於汽車上，並透過擷取到的障礙物深度資訊，設計了一個會隨深度資訊不同而調整的3D投影模型，以解決影像接合處的鬼影問題，並減低車輛周圍物體的影像扭曲。

關鍵字：車輛環周監視系統；視像內插；影像縫合；混合式投影模型；車輛安全空間計算；深度可調三維模型；

Abstract

Driving assistance systems help drivers to avoid car accidents by providing warning signals or visual cues of surrounding situations. Instead of the fixed bird's-eye view monitoring proposed in many previous works, we developed a real-time vehicle surrounding monitoring system, ”Angel Eye”, that can assist drivers to perceive the vehicle surrounding situations more easily. In our system, four fisheye cameras are mounted around a vehicle. To integrate these four fisheye camera views, we firstly use fisheye camera calibration method to dewarp the captured images into perspective projection ones. Then, we calculated the camera intrinsic parameters and homography transform matrix to get the camera extrinsic parameters. To stitch these dewarpped images, we projected undistorted images into a 3D hybrid projection model and finally the images of the selected viewpoint are rendered.

However, the unknown position of foreground obstacles would cause some visual noises, like image distortion of objects or ghost effect. So we add depth camera into previous system to obtain the depth information of foreground obstacles. The proposed 3D model can be adjusted based on the distance between vehicle and foreground obstacles. The depth-adaptive model can facilitate the rendering of vehicle surroundings in a more realistic and correct way.

Keywords: vehicle surrounding monitoring; view interpolation; image stitching; hybrid projection model; free space calculation; depth-adaptive 3D model;

Contents

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

Blind spot of the vehicle is one of the major reasons causing car accidents. Helping drivers perceive the situation around the vehicle while driving is the main goal of intelligent driving assistance systems. Toward this goal, many kinds of driving assistance systems have been developed. For example, parking sensors measure the distances of the obstacles from the vehicles and warn the drivers by beepers. Automatic braking system can stop or slow down the vehicle to prevent collisions. Rear-view camera captures the view behind the vehicle and can help drivers to drive backward. Because the cost of cameras keeps dropping, it becomes practical to mount multiple cameras around the car to provide drivers the surrounding views for better perception of situations around the vehicles.

Figure 1.1 shows the nowadays systems developed by some vehicle manufacturers. Nissan constructed an assistant system, Around View Monitor [1], to let drivers see the surrounding view of the vehicle by using four wide-angle cameras. This system can provide drivers both the aerial view and the original four cameras' images. The Eagle View [2] from Luxgen and Multi-View Camera System [3] from Honda use similar methods to provide drivers better concept of surrounding view. And Fujitsu released a system, 360° warp-around video imaging system [4], which project the acquired images onto a 3D curved plane and provide drivers a third person view.

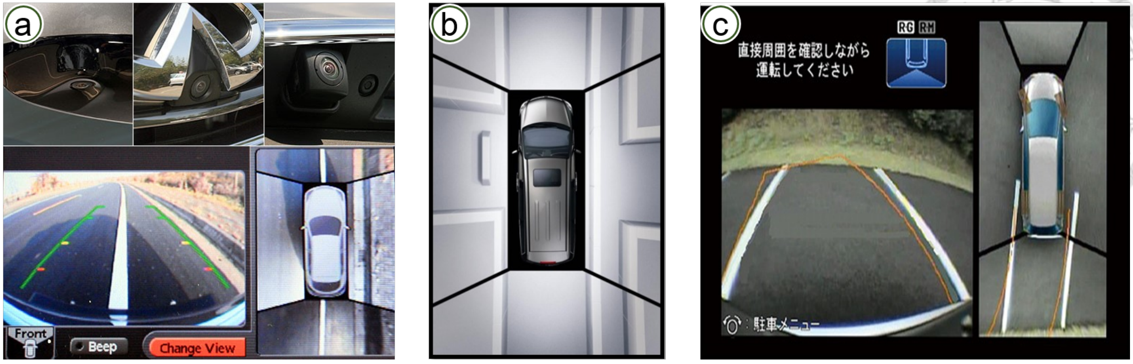

Figure 1.1: Nowadays systems developed by different vehicle manufacturers. (a) Around View Monitor, (b) Eagle View, and (c) Multi-View Camera System.

So we propose a monitoring system for driving assistance, ”Angel Eye”. We place four fisheye cameras around the vehicle and project these acquired images onto a static hybrid model. We can render the result images according to a selected viewpoint. The system can provide a third person viewpoint with many orientations. And these viewpoints can give drivers larger visual range then the fixed bird’s-eye viewpoint. However, because of the unknown distance between the vehicle and the foreground obstacles, there exists image distortion and ghost effect of the objects. So we add one depth camera in the system to get the scene information. After we calculate the free space around the vehicle from the depth image, we design a depth-adaptive 3D model according to the free space. The depth-adaptive model can facilitate the rendering of vehicle surroundings in a more realistic and correct way. Figure 1.2 shows the difference between static model and depth-adaptive model and their corresponding result.

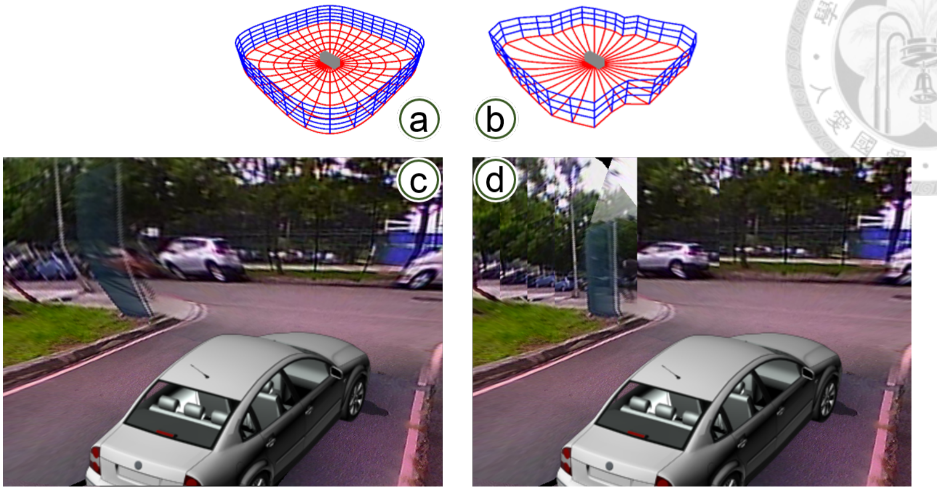

Figure 1.2: Static model and depth-adaptive model with respective result.(a)The static hybrid model.(b)The depth-adaptive model.(c)(d)The corresponding results

1.2 Organization of the Thesis

The preprocessing stage of the system, including the environmental setup and camera calibration, is explained in chapter 3. Chapter 4 illustrates the vehicle surrounding monitoring system with static model and model comparison. Chapter 5 reports the details of the algorithms we use to find the depth-adaptive 3D model.

Chapter 2

Related Work

In this chapter, we list the previous works related to our system: vehicle surrounding monitoring system and model construction in 3D scene.

2.1 Vehicle Surrounding Monitoring System

To help driver perceive the situation near the vehicle, Ehlgen and Pajdla [5] proposed a method by mounting four omnidirectional cameras around a truck. They found several ways to split the overlap area and provided a bird's-eye view image. In Liu's method [6], they mounted six fisheye cameras around the vehicle. They selected the optimal seam in the overlap area and used dynamic image warping to stitch all six images together to provide a bird's-eye view for the surrounding environments of the vehicle. The Around View Monitor of Nissan [1], Eagle View of Luxgen [2] and Multi-View Camera System from Honda [3] all mounted wide-angle cameras onto the vehicle. These systems provide drivers both the aerial view and the original images acquired by cameras. Delphi Automotive proposed a parking system, 360° Surround View System with Parking Guidance [7], by using the same idea. Moreover, they applied better blending algorithm to make smooth visual results. Fujitsu released a system, 360° warp-around video imaging system [4], which project the acquired images onto a 3D curved plane and provide drivers a third person view.

2.2 Model Construction in 3D Scene

To construct the depth-adaptive model, we need to find the position of foreground objects. Instead of using traditional image recognition method, we simply use the stereo image set or disparity map to estimate the position of obstacles. S. Kubota, T. Nakano, and Y. Okamoto proposed an obstacle detection system by using two stereo images [8]. They assumed that the road plane function is estimated, so we can affine transform the right image according to the road plane. Then comparing the affined image with left one to find the free space. The free space of one vehicle means the ground region near the car so there are no obstacles inside the space. A. Wedel used B-spline to model the non-planar ground surfaces, and stereo disparities to track the free space [9]. H. Badino used dense disparity image to calculate the free space [10]. He transformed the disparity map into occupancy grids and applied dynamic programming to segment the free space and obstacles. Then he proposed a representation of the 3D world, Stixel World [11]. After finding the free space, he also applied dynamic programming to find the optimal segmentation between foreground obstacles and backgrounds. He used the stixel to represent the obstacles and stixel's color to show their depth. D. Pfeiffer then used stixels to represent the 3D world and traffic scenes [12] [13] [14]. R. Benenson proposed a method to estimate stixels with stereo images [15]. And using the stixels to reduce the number of searching windows while detecting foreground objects.

Chapter 3

Preprocessing

In this section, we briefly described the environmental setup and image preprocessing to get the undistorted images and camera parameters of fisheye camera. Figure 3.1 shows the flow of preprocessing stage.

3.1 Fisheye Camera Calibration and Intrinsic Parameters

We use four fisheye cameras to acquire the images around the vehicle. The FOV of camera lens is 183°. By using such wide-angel camera, we can have larger overlapped area between neighboring cameras. However, because the FOV is so wide that we can't apply any existed fisheye models to dewarp the distorted images. So we design an experiment, as shown in Figure 3.2, to calculate the correspondence between the displacement, the

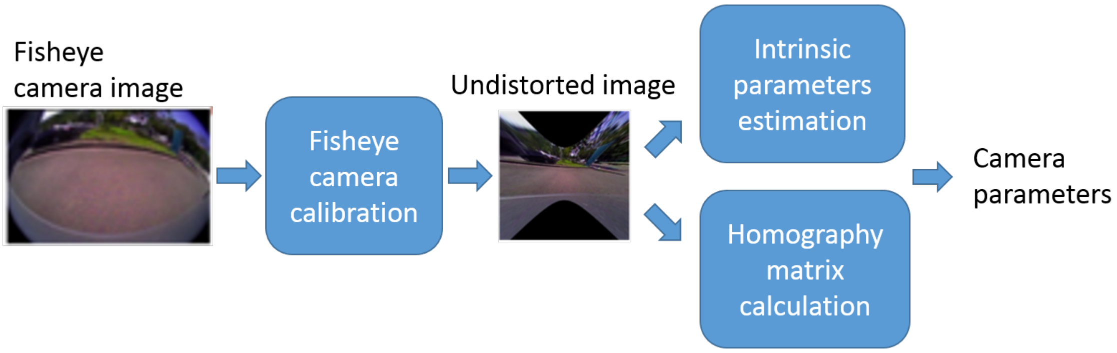
Fisheye camera image
Fisheye camera calibration
Undistorted image
Intrinsic parameters estimation
Homography matrix calculation
Camera parameters

Figure 3.1: The flow of preprocessing.

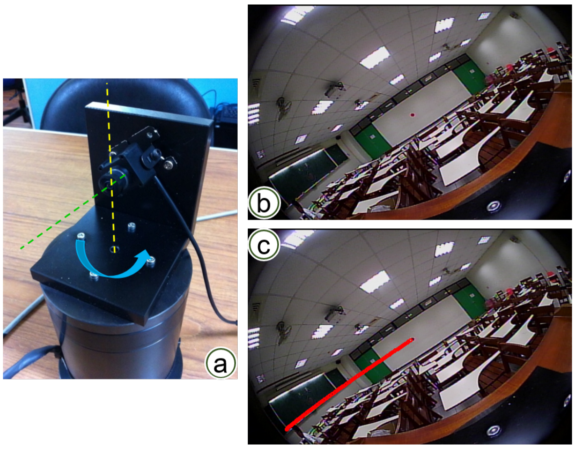

Figure 3.2: The experiment of fisheye camera calibration. (a)The rotating device with stepper motor. (b)The feature point on the wall 5 meters away. (c)The sequential locus of feature points.

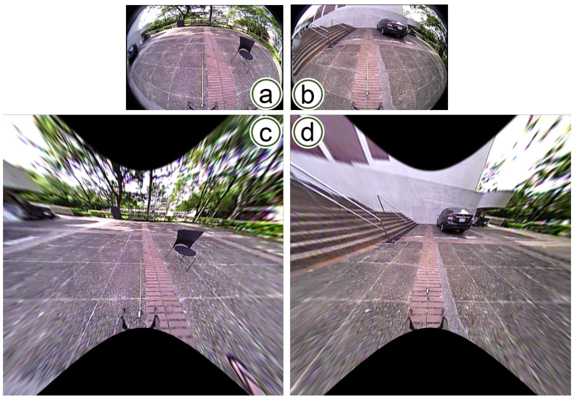

Figure 3.3: The results of fisheye camera calibration. (a)(c)The images captured by fisheye camera. (b)(d)The corresponding dewarpped results.

distance from a pixel to center of image, and its incident angle. Then we can use this relationship to dewarp a distorted image into a perspective one. Figure 3.3 shows the results of fisheye camera calibration. After we get the undistorted images, we assume these images are taken by a virtual perspective camera and apply Zhang's calibration method [16] to find the intrinsic parameters of virtual camera.

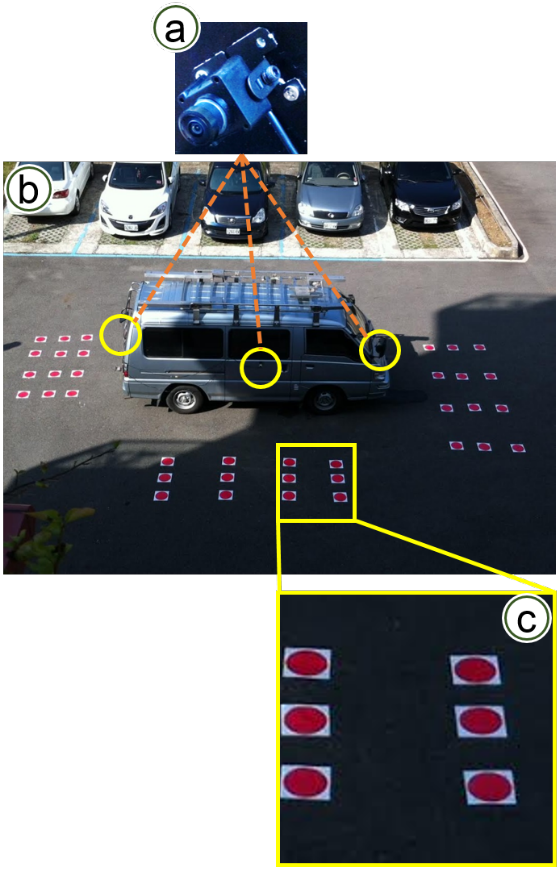

Figure 3.4: Fisheye camera positions and homography experimental setup.(a)Fisheye cameras. (b)Four cameras are mounted and 48 circle patterns are placed. (c) The feature points.

3.2 Homography and Extrinsic Parameters

To project the undistortd image onto a 3D model, we need to know the extrinsic parameters of the fisheye cameras. As shown in Figure 3.4, we mounted cameras around the vehicle, park it at a flat ground and place 48 feature points around the vehicle. Each camera can capture 12 feature points, so we have 12 correspondences between image coordinate and real-world coordinate. By using four of them and applying least squares algorithm, we can get the homography matrix of each camera. Then we combine the homography matrix and the intrinsic parameters to obtain the extrinsic parameters by following equations.

$$M_{ext} = \begin{bmatrix} R_1 & R_2 & R_3 & T \end{bmatrix} \tag{3.1}$$

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = s \cdot M_{int} \cdot \begin{bmatrix} R_1 & R_2 & T \end{bmatrix} \tag{3.2}$$

$$R_1 = \frac{1}{s} \cdot M_{int}^{-1} \cdot h_1 \tag{3.3}$$

$$R_2 = \frac{1}{s} \cdot M_{int}^{-1} \cdot h_2 \tag{3.4}$$

$$T = \frac{1}{s} \cdot M_{int}^{-1} \cdot h_3 \tag{3.5}$$

and

$$R_3 = R_1 \times R_2 \tag{3.6}$$

where h_1, h_2, h_3 are the column of homography matrix H, s is a scaling factor, M_{int} is the intrinsic matrix and M_{ext} is the extrinsic matrix which can be composed to rotation matrix R_1, R_2, R_3 and translation matrix T.

Chapter 4

Vehicle Surrounding Monitoring Using Static 3D Model

To propose a system that can let driver choose different third person angle, and synthesize the result image from the four images acquired by fisheye cameras. The direct way is to perform view interpolation proposed by S. Chen [17]. However, in the real-time situation of vehicle surrounding monitoring, the passed-by objects are moving and the system should have less computational time. So we design a 3D hybrid model and project four undistorted images onto it. Then we use intrinsic and extrinsic parameters of the selected viewpoint to synthesize results. The flow of this section is shown in figure 4.1 and the concept of view interpolation is shown in figure 4.2.

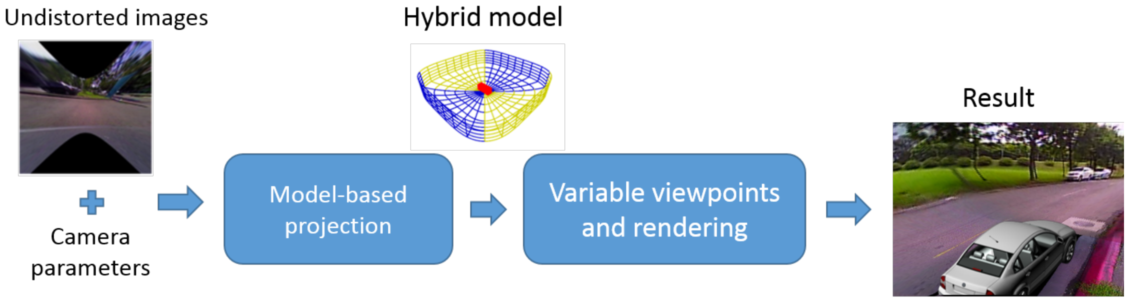

Figure 4.1: The flow of vehicle surrounding monitoring using static 3D model.

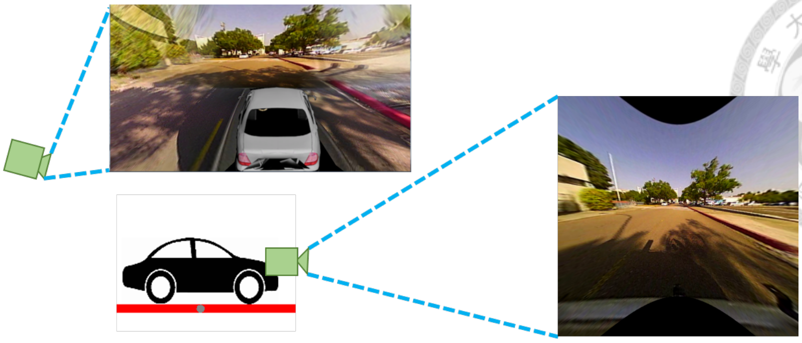

Figure 4.2: View interpolation.

4.1 View Interpolation Using Static 3D Model

As other vehicle surrounding systems mentioned at chapter 1, We can stitch four dewarpped images into ground plane to see the objects around vehicle. However, there exists two major problems. First, the image would be with some distortions. Second, image textures, which are above the vanishing point, would be projected to an infinite position. It makes us unable to see the real scene above the vanishing line from the viewpoint. To solve these problems, we calculate the camera extrinsic parameters from homography matrix and camera intrinsic parameters. And project the acquired images to 3D model. We use back-projection to find the texture of 3D model. The projection equation is shown in equation 4.1.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{w} \cdot M_{int} \cdot M_{ext} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.1}$$

where $\begin{bmatrix} x & y & 1 \end{bmatrix}^T$ is the coordinate in dewarpped image. w is the homogeneous factor. M_{int}, M_{ext} is the intrinsic and extrinsic parameters of fisheye camera. And $\begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ is the coordinate in 3D model.

We use four parameters to define the viewpoint, shown as figure 4.3. Pan is from $-\pi$ to π, which describes the main direction drivers interested. $Tilt$ is set as $\frac{\pi}{3}$, which is the angle between virtual camera and normal vector of ground. The elevation angle, γ, is set

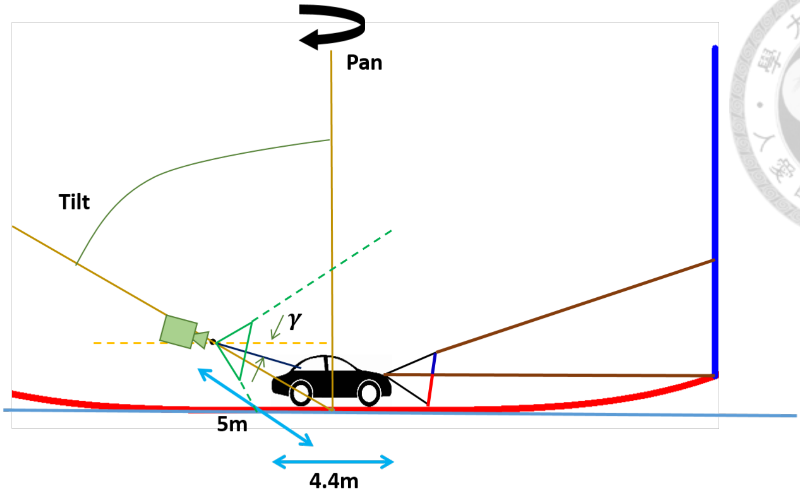

Figure 4.3: Variable viewpoint.

as $\frac{\pi}{12}$. The distance from virtual camera to centroid of curved surface is 5 meters.

4.2 Model Comparison

We compared result images by projecting acquired images into four different 3D models. The equation of these models are:

Model 1: Ground plane and cylinder surface.

$$\begin{cases} z = 0, & \text{while } \ x^2 + y^2 < d^2 \\ z \geq 0, & \text{while } \ x^2 + y^2 = d^2 \end{cases} \tag{4.2}$$

Model 2: Second degree paraboloid.

$$z = \frac{(x^2 + y^2)}{a^2} \tag{4.3}$$

Model 3: Fourth degree paraboloid.

$$z = \frac{(x^4 + y^4)}{a^4} \tag{4.4}$$

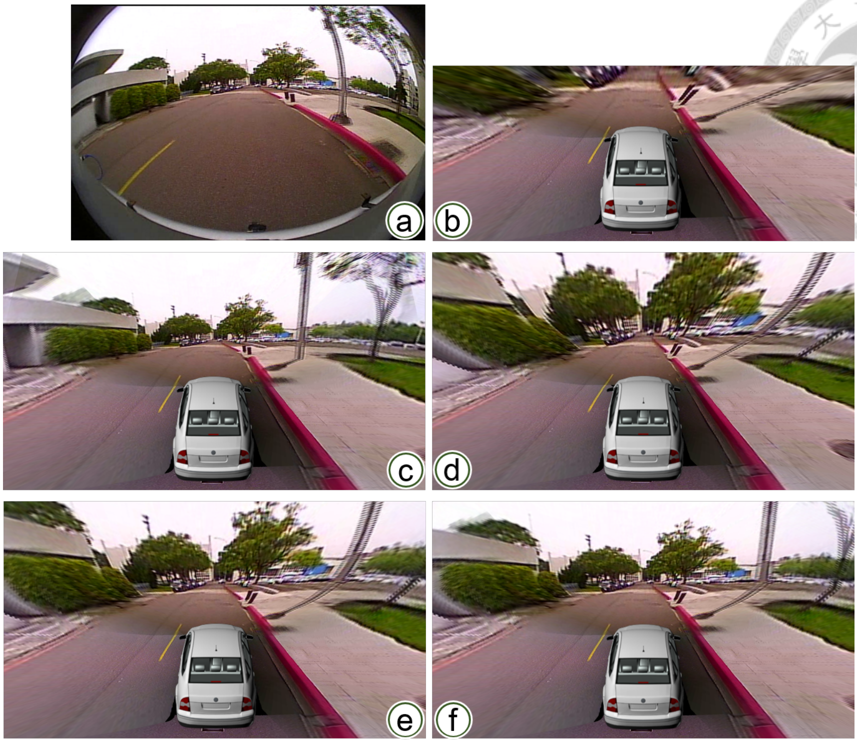

Figure 4.4: The results of different 3D model. (a)Acquired image from fisheye camera. (b)After homography. (c)Use ground plane and cylinder model. (d)Use second degree paraboloid. (e)Use fourth degree paraboloid. (f)Use hybrid model.

Model 4: Hybrid model.

$$\begin{cases} z = \frac{(x^4+y^4)}{a^4}, & \text{while } \; x^4 + y^4 < c^4 \\ z \geq \frac{(x^4+y^4)}{a^4}, & \text{while } \; x^4 + y^4 = c^4 \end{cases} \tag{4.5}$$

where a, c and d are the coefficients of the paraboloid, columnar and cylinder. Figure 4.4 shows an example of comparison results respectively. After image undistorted, we can project the images into ground plane by using homography matrix, as shown figure 4.4(b). But there are some distortions with the objects in height, such as tree or building. The other problem is that the content above the vanishing point won't show on the ground plane. So we project the acquired images into a 3D surface to reduce the distortions. In the model 1(figure 4.4(c)), due to the ground plane and cylinder is vertical, objects would bend in an unusual way. In model 2 and 3(figure 4.4(d)(e)), although scene is smooth, there is expansion effect between surroundings.

There are two reasons why we choose the hybrid model as our 3D model. First, the

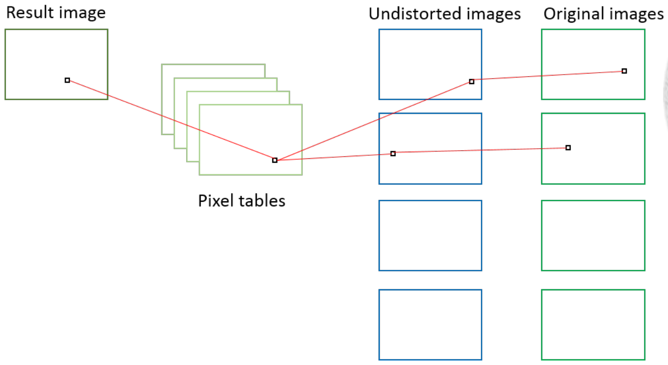

Figure 4.5: Structure of lookup table.

columnar surface makes the view more realistic in four directions. Second, the smoothness between fourth degree paraboloid and columnar is better. As figure 4.4(f), better result with less distortion is generated.

4.3 Lookup Table

To decrease the computation time, we use a lookup table approach instead of warping and stitching the whole images for each frame. According to the look-up table, we can get the pixel value of output image from the original fisheye camera images by at most seven add operations and six multiplication operations.

Each pixel of the table includes three data values while the pixel's information can be retrieved from single camera frame. One is the camera id, and the other two is the coordinate (x, y) in that image. For those pixels in the overlapping area, there will be seven data values in the lookup table. They are two coordinates of two different images and further one linear blending weighting factor. The structure of lookup table is shown in figure 4.5.

4.4 Experimental Setup

In our experiments, four fisheye cameras and one experiment car, Delica of Mitsubishi, were used. We tested our system in a campus and recorded about 300 seconds. The computation time of each frame with Intel Xeon CPU, E3-1230 @ 3.3G Hz, and 16G

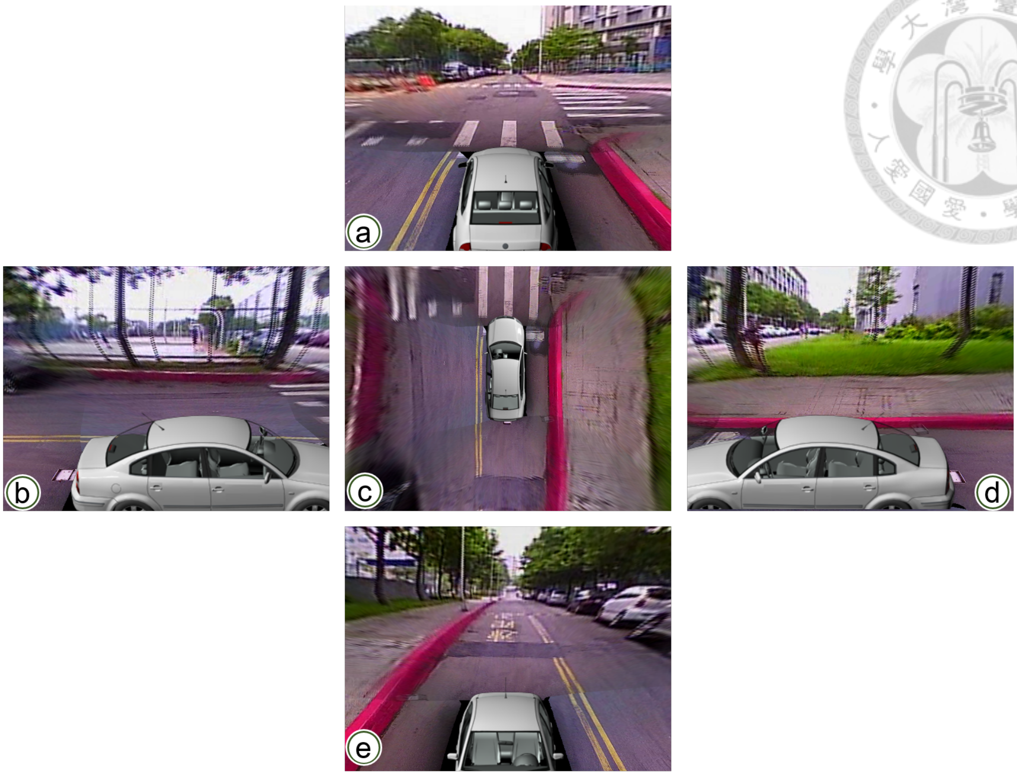

Figure 4.6: The results of different viewpoints. (a)Front view. (b)Left view. (c)Top view. (d)Right view. (e)Rear view.

RAM is 0.7 seconds. The compiled environment is Visual Studio 2010, Release mode with Full Optimization.

4.5 Results and Discussion

Figure 4.6 shows the result in different viewpoint. Drivers can check every side of her vehicle carefully. And more sequential results is shown in figure 4.7. With this system, it's much safer to drive through a small lane, change lane in the highway, turn right or left in the intersection and backward in a narrow area.

However, we can find two problems in those result images, image distortion and ghost effect, shown in figure 4.8. This is because we use one static 3D model for all vehicle surrounding scenes. No matter the passing by walker or the tree in 15 meters away, we both project them onto the same 3D model. So those objects would have some distortions as figure 4.8(b). And inside the camera overlap area, the objects come from two different camera image simultaneously. Due to the unknown depth of objects, the same person

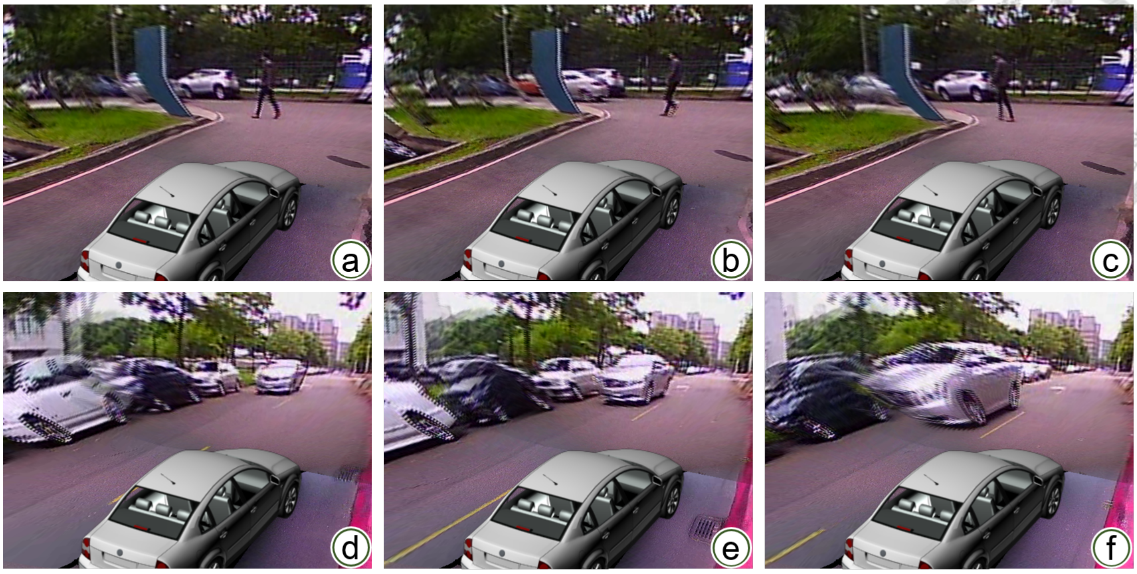

Figure 4.7: The sequential results using static hybrid model. (a)(b)(c) One example. (d)(e)(f) Another example.

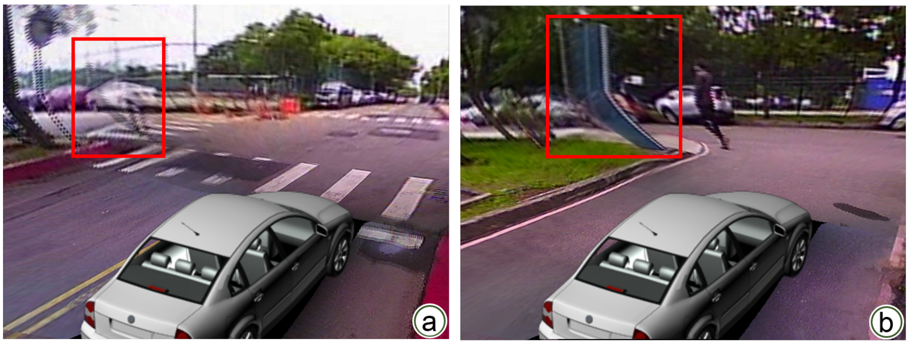

Figure 4.8: Ghost effect and image distortion. (a)Ghost effect. (b)Image distortion.

in different camera would be projected to slightly different position and cause the ghost effect, 4.8(a). So we need a 3D model which can adjusted it's shape according to the surroundings' position to solve these problems..

Chapter 5

Vehicle Surrounding Monitoring Using Depth-Adaptive 3D Model

To solve the image distortion and ghost effect caused by unknown foreground obstacles' position. Figure 5.1 shows an example results of different columnar parameter. When the columnar depth is matched with object depth, the ghost effect and image distortion could be minimized. Therefore, we add depth camera into the previous system. So we can get the depth information of objects around the vehicle. And we apply the same camera calibration procedure to find the intrinsic and extrinsic parameters of depth camera. According to these parameters and depth image, we calculate the free space of vehicle. Then we can construct a depth-adaptive 3D model based on the free space and project the undistorted images onto that model to synthesize better result image. The flow is shown in figure 5.2.

5.1 Depth Image Processing

We use a stereo camera to obtain depth image(figure 5.3). The advantage of stereo camera is that it can be used outdoor. We use H. Badino's method [10] to calculate the free space, which means the ground space without any objects around the vehicle, shown as figure 5.4. First, we transform the depth image into an occupancy grid(figure 5.4(a)(b)). The size of occupancy grid is 640 columns, same as the width of depth image, and 256

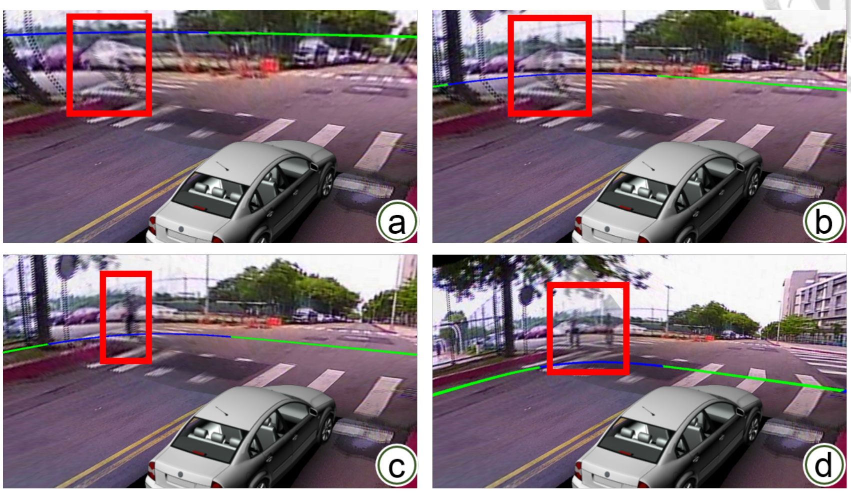

Figure 5.1: The example of ghost effect using different columnar parameter. (a)$c = 15m$, (b)$c = 10m$, (c)$c = 8m$ and (d)$c = 5m$.

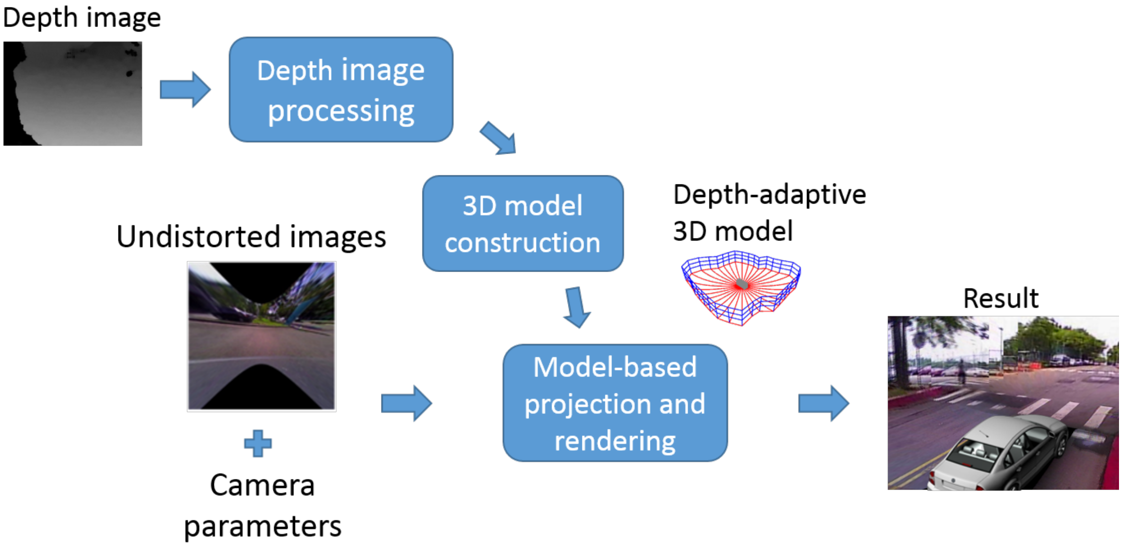

Figure 5.2: The flow of vehicle surrounding monitoring using depth-adaptive 3D model.

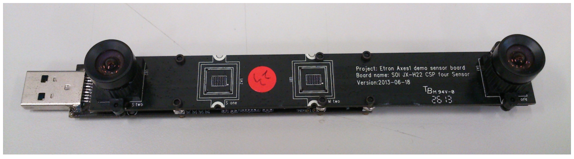

Figure 5.3: Stereo camera.

rows, the number of depth levels of depth image. So if the value of (x, y) in occupancy grid is k, it means in the column x of depth image, there are k pixels with their depth values are y. Second, we apply dynamic programming to find a optimal path in the occupancy grid(figure 5.4(b)(c)). The equation 5.1 shows the cost while performing the dynamic programming to the occupancy grid.

$$c_{i,j,k,l} = E_d(i,j) + E_s(i,j,k,l) \tag{5.1}$$

is the cost of the edge connecting V_{ij} and V_{kl} where;

$$E_d(i,j) = \frac{1}{D(i,j)} \tag{5.2}$$

is the data cost. If $D(i,j) = 0$, assign a large value to $E_d(i,j)$. And;

$$E_s(i,j,k,l) = \begin{cases} C_s \cdot d(j,l) & \text{if } d(j,l) \leqslant Ts \\ C_s \cdot Ts & \text{if } d(j,l) > Ts \end{cases} \tag{5.3}$$

is the smoothness cost where $d(j,l)$ is the distance between row j and row l, and Ts is the threshold to preserve depth discontinuities.

This path means the obstacle's depth in each column of depth image. Finally, after we transform the segmentation result by using intrinsic and extrinsic parameters of stereo camera, we can get the free space inside the depth image(figure 5.4(c)(d)).

However, because the depth data in the depth image would be missing when it can't find the match between right and left images of stereo camera, there exists some errors in the result of free space calculation. The missing value usually caused in the ground plane area because of lacking feature points. Besides, the height of obstacles usually higher than

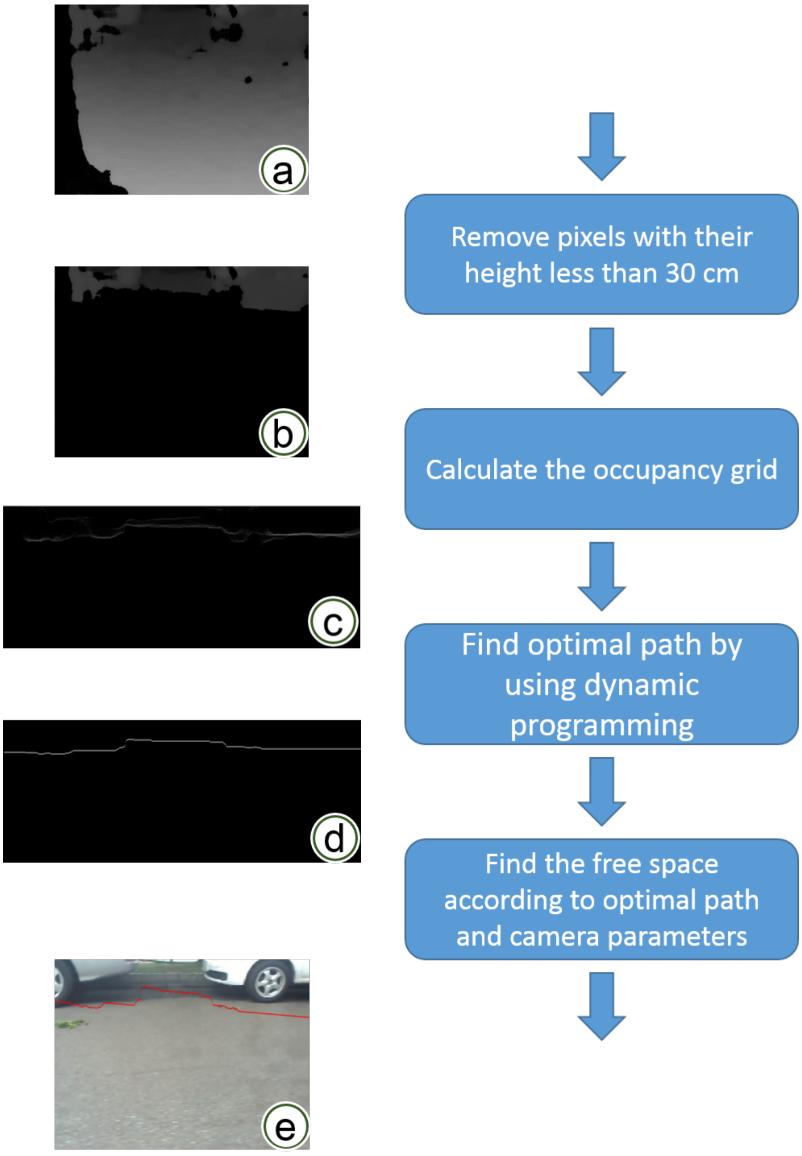
a
Remove pixels with their height less than 30 cm
b
Calculate the occupancy grid
c
Find optimal path by using dynamic programming
d
Find the free space according to optimal path and camera parameters
e

Figure 5.4: The flow of free space calculation. (a)Depth image. (b)Depth image excluding ground pixels. (c)Occupancy grid. (d)Obtain optimal path by using dynamic programming. (e)Calculate free space with optimal path, intrinsic and extrinsic parameters of stereo camera.

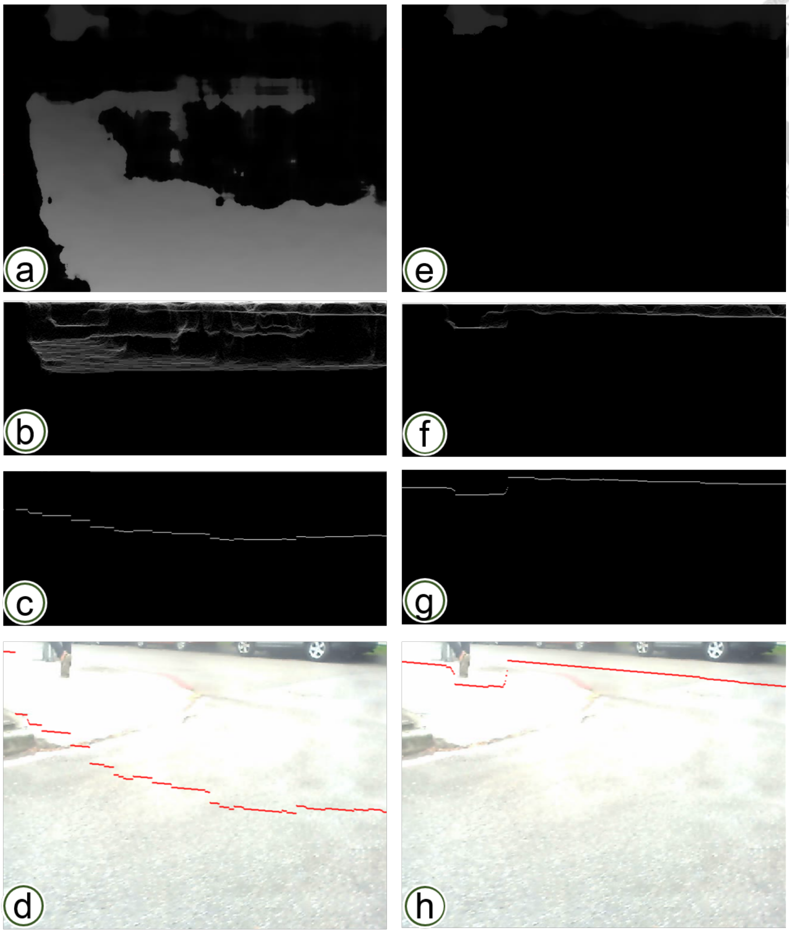

Figure 5.5: Results of free space calculation. (a)(b)(c)(d)The result is worse by using the original depth image with lost data. (e)(f)(g)(h)The result is much better by using the depth image excluding ground pixels.

30 centimetres. So we set 30 centimetres as height threshold to eliminate the ground plane data while transforming the depth image into occupancy grid. By doing this, we can get better results. Figure 5.5 shows the comparison before and after our modification.

5.2 Depth-Adaptive Model Construction

Based on the free space, intrinsic and extrinsic parameters of stereo camera, we can get the position information of foreground objects. Figure 5.6(a) shows the 3D model we construct based on the corresponding free space. The top-right corner is the heading direction of the vehicle. A closer look of depth-adaptive region is shown in Figure 5.6(b). We combine ground plane and columnar while rendering. And adjust the columnar parameter c to fit the 3D model we construct. Figure 5.6(c) shows the top view of the rendering result

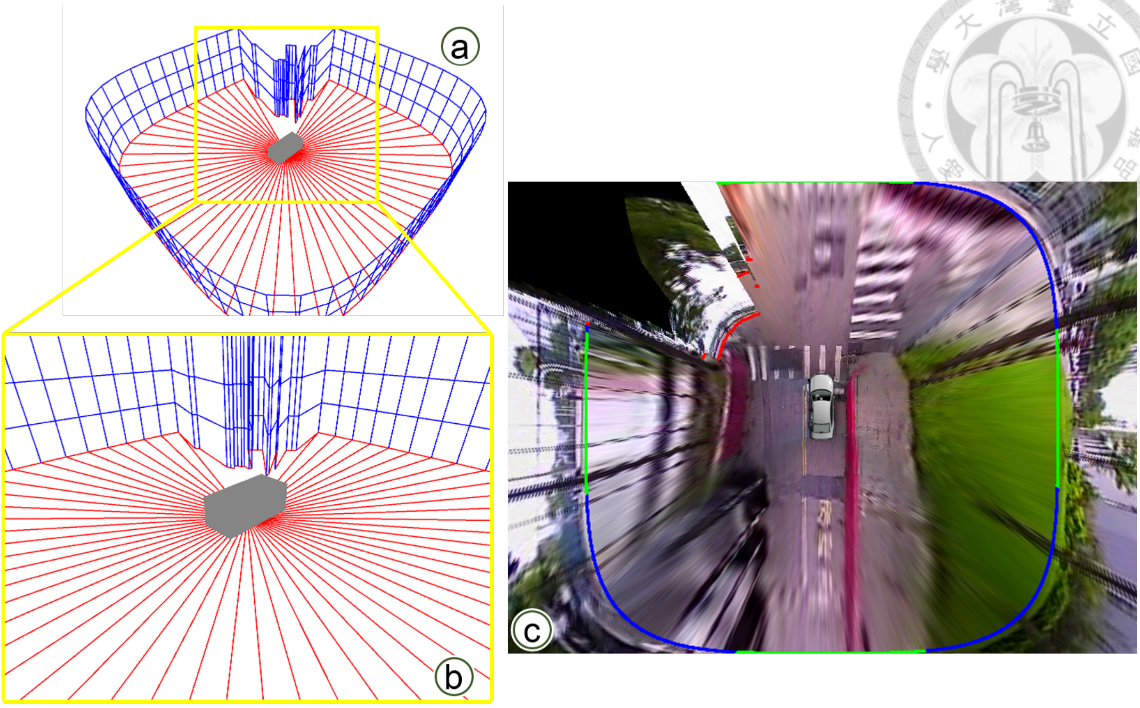

Figure 5.6: Depth-adaptive model construction. (a)The depth-adaptive model based on free space calculation. (b)A closer look of the depth-adaptive region of the model. (c)The corresponding top view rendering result.

corresponding to the depth-adaptive 3D model(Figure 5.6(a)).

5.3 Rendering Results and Discussion

We add one stereo camera(figure 5.3), Etron Technology, into our experimental setup which is described in previous chapter. While rendering, we use ten different depth of columnar to fit the 3D model due to the computation limit. The sequential results are shown in figure 5.7. In these images, image distortion and ghost effect of foreground objects are improved because we project them onto the model which representing their depth. Figure 5.8 and figure 5.9 show the same scene but using static hybrid model and depth-adaptive model. The ghost effect of the pedestrian in figure 5.8(c) is eliminated in figure 5.8(f). And the image distortions of the trees in figure 5.9(a)(b)(c) are also improved in figure 5.9(d)(e)(f).

But there is other issue we should consider, the discontinuities of scene due to the discrete depth model(figure 5.10). Because of the computing memory limitation, we only

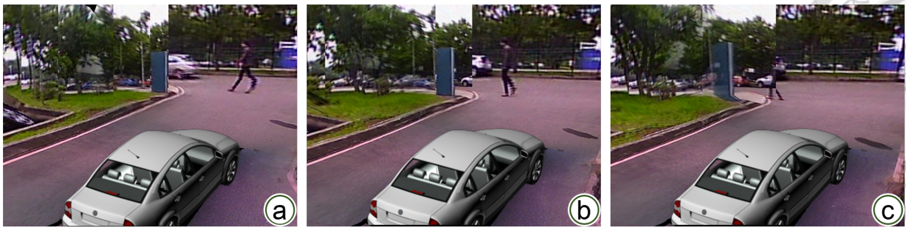

Figure 5.7: The sequential results using depth-adaptive 3D model.

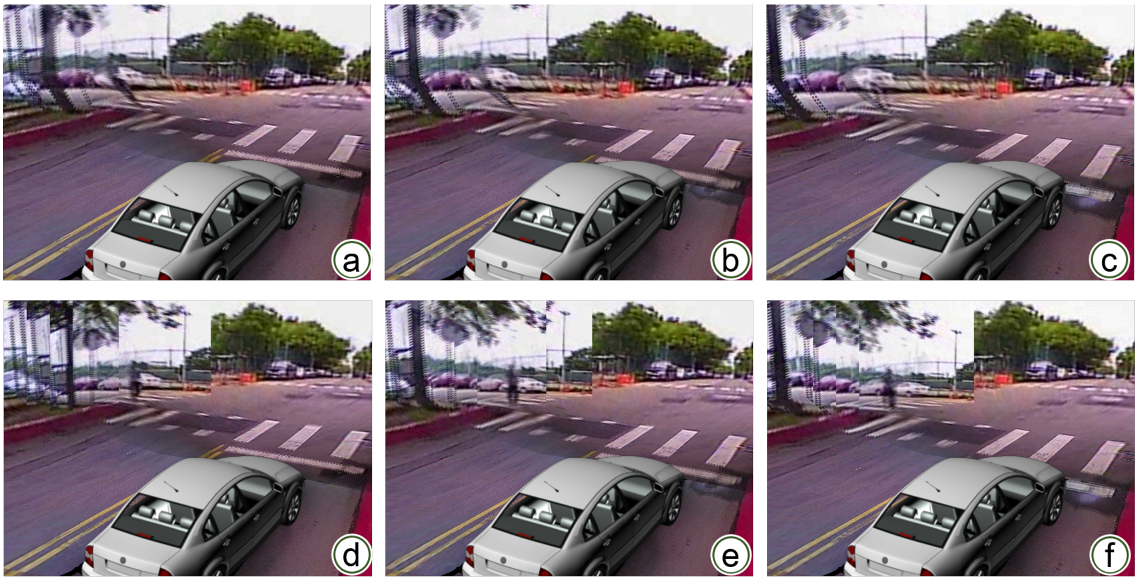

Figure 5.8: (a)(b)(c)Sequential results using static hybrid model. (d)(e)(f)Corresponding results using depth-adaptive model.

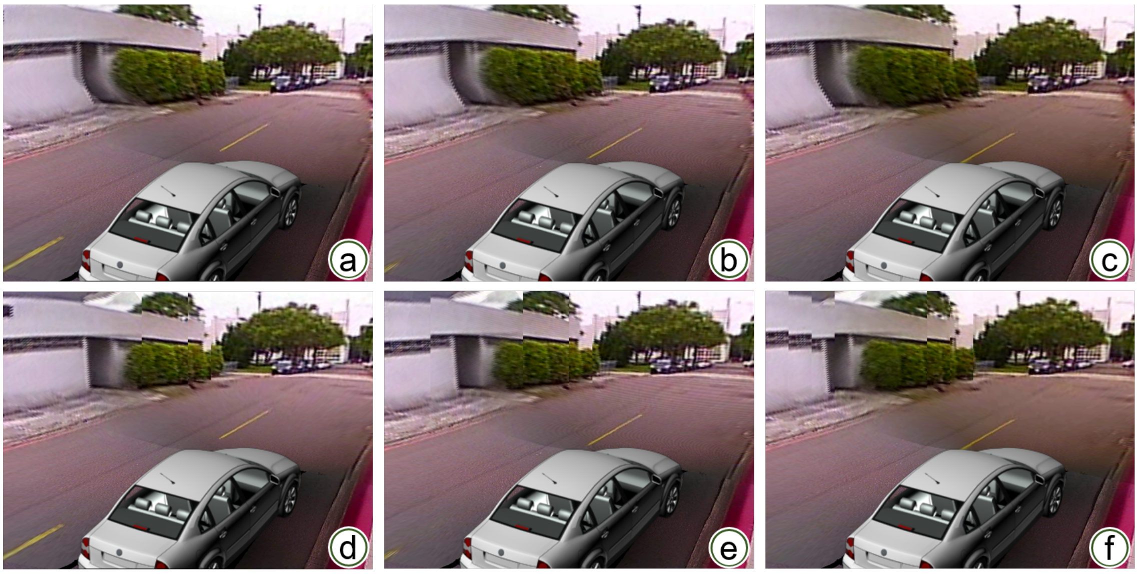

Figure 5.9: (a)(b)(c)Sequential results using static hybrid model. (d)(e)(f)Corresponding results using depth-adaptive model.

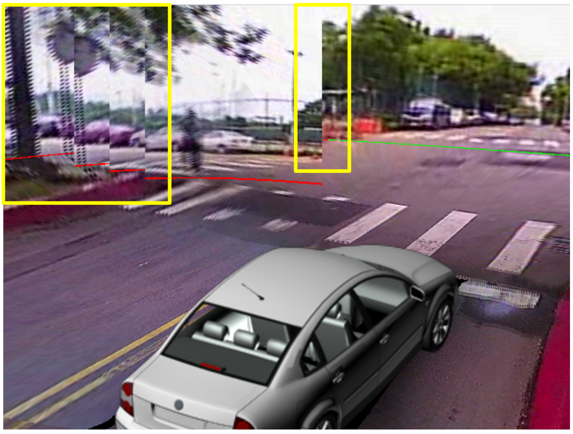

Figure 5.10: The discontinuous problem cause by discrete depth-adaptive model.

can build ten to fifteen lookup table to represent different depth model and use them to fit the real scene. We use ten different depth to render results shown in this thesis. The results sometimes are noisy when an object are cut into pieces and projected to different depth of model. To solve this problem, we need a smoother 3D projection model. And because the limitation caused by using lookup table as implement method, we may use 3D rendering library such as OPENGL to render the real 3D scene. So we can build a depth-adaptive 3D model with spatial and temporal smoothness and render the corresponding 3D texture.

Chapter 6

Conclusion and Future Work

In this work, we propose a vehicle surrounding monitoring system and provide driver variable third person viewpoint. Not like those nowadays systems developed by some manufacturers, drivers can both perceive ground situation around the vehicle and the the surroundings in a more realistic and intuitive way. The system can help the drivers better comprehend the driving situation and avoid most of car accidents.

First, we propose a hybrid model to make the rendered view with less distortion. Furthermore, we design a depth-adaptive model to solve the image distortion and ghost effect in previous system.

But the rendered images using depth-adaptive model have discontinuities because of the different depth. The results would become noisy when a continuous object is cut into pieces. In the future, object detection technologies can help system project one continuous object onto same depth model, and decrease the noise of discontinuities. And we can build a depth-adaptive 3D model with spatial and temporal smoothness to solve the problem. Moreover, driver warning applications can be added in our system. More convenient, and user friendly driving assistance system will be provided.

References

[1] NISSAN | Around View Monitor TECHNOLOGICAL DEVELOPMENT ACTIVITIES.

[2] LUXGEN | Eagle View+ 360° surround imaging system.

[3] Honda Worldwide | ”Honda Develops New Multi-View Camera System to Provide View of Surrounding Areas to Support Comfortable and Safe Driving”.

[4] Fujitsu United States | 360° Wrap-Around Video Imaging Technology.

[5] Tobias Ehlgen and Tomas Pajdla. Monitoring surrounding areas of truck-trailer combinations. In *Proceedings of the 5th International Conference on Computer Vision Systems*, number Icvs, 2007.

[6] Yu-Chih Liu, Kai-Ying Lin, and Yong-Sheng Chen. Bird’ s-Eye View Vision System for Vehicle Surrounding Monitoring. In *Proceedings of 2nd International Workshop, RobVis 2008*, volume 4931/2008 of *Lecture Notes in Computer Science*, pages 207–218. Springer, 2008.

[7] Mengmeng Yu and Guanglin Ma. 360° Surround View System with Parking Guidance. *SAE International Journal of Commercial Vehicles*, pages 7(1):19–24, April 2014.

[8] Susumu Kubota, Tsuyoshi Nakano, and Yasukazu Okamoto. A Global Optimization Algorithm for Real-Time On-Board Stereo Obstacle Detection Systems. In *2007 IEEE Intelligent Vehicles Symposium*, pages 7–12. IEEE, June 2007.

[9] Andreas Wedel, Uwe Franke, Hernan Badino, and Daniel Cremers. B-spline modeling of road surfaces for freespace estimation. In *2008 IEEE Intelligent Vehicles Symposium*, pages 828–833. IEEE, June 2008.

[10] Hernán Badino, Uwe Franke, and Rudolf Mester. Free space computation using stochastic occupancy grids and dynamic programming. In *Proc. International Conference on Computer Vision, Workshop on Dynamical Vision, Rio de Janeiro, Brazil*, page 20, 2007.

[11] Hernán Badino, Uwe Franke, and David Pfeiffer. The stixel world-a compact medium level representation of the 3d-world. In *Proceedings of the 31st DAGM Symposium on Pattern Recognition,*, pages 51—-60, 2009.

[12] David Pfeiffer and Uwe Franke. Efficient representation of traffic scenes by means of dynamic stixels. In *2010 IEEE Intelligent Vehicles Symposium*, pages 217–224. IEEE, June 2010.

[13] David Pfeiffer and Uwe Franke. Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data. In *Procedings of the British Machine Vision Conference 2011*, pages 1—-12. British Machine Vision Association, 2011.

[14] David Pfeiffer, Friedrich Erbs, and Uwe Franke. Pixels , Stixels , and Objects. In Andrea Fusiello, Vittorio Murino, and Rita Cucchiara, editors, *Proceedings of the 12th European Conference on Computer Vision(ECCV'12), Workshops and Demonstrations*, volume 7585 of *Lecture Notes in Computer Science*, pages 1–10, Berlin, Heidelberg, October 2012. Springer Berlin Heidelberg.

[15] Rodrigo Benenson, Radu Timofte, and Luc Van Gool. Stixels estimation without depth map computation. In *Proc. International Conference on Computer Vision, Computer Vision Workshops*, pages 2010—-2017, 2011.

[16] Zhengyou Zhang. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(November 2000):1330–1334, 2009.

[17] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. *Proceedings of the 20th annual conference on Computer graphics and interactive techniques - SIGGRAPH '93*, pages 279–288, 1993.