The Report committee for Janice Shuay-ann Pan certifies that this is the Approved version of the following report:

Automotive Top-view Image Generation using Orthogonally Diverging Fisheye Cameras

APPROVED BY

SUPERVISING COMMITTEE:

Alan C. Bovik, Supervisor

Joydeep Ghosh

Automotive Top-view Image Generation using Orthogonally Diverging Fisheye Cameras

by

Janice Shuay-ann Pan, B.S.E.E.

REPORT

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2016

Dedicated to my family: my mom and dad, who have always encouraged me and supported my education, and my sisters, Cassie and Emily, who inspire me to no end.

Acknowledgments

I would first like to thank my incredibly supportive advisor, Professor Al Bovik, without whom this work would not be possible. At the onset of this project, the direction and problems we wanted to explore were unknown, and his mentorship, ideas, and feedback were invaluable in helping me approach a problem that has yet to be solved.

I would also like to thank Texas Instruments for their support and Vikram Appia, specifically, for his insight, guidance, and assistance from the very beginning of this project. While the project was proposed by TI, Vikram always gave me the flexibility and control to drive my own research.

I must also acknowledge my fellow LIVE members for their constant encouragement, as well as Professor Joydeep Ghosh, who has been a mentor to me outside of this project and is serving as the second reader to this report.

Automotive Top-view Image Generation using Orthogonally Diverging Fisheye Cameras

Janice Shuay-ann Pan, M.S.E.
The University of Texas at Austin, 2016

Supervisor: Alan C. Bovik

Advanced Driver Assistance Systems in vehicles can be a great assistance to drivers by providing them a quick and easy way to visualize their entire 360^o surroundings. We introduce a new camera set-up for a surround-view imaging system that may be part of an ADAS. This set-up involves four wide-angle fisheye cameras with orthogonally diverging camera axes, which allows for capturing the entire 360^o around a vehicle in four images, captured from the lateral, front, and rear views.

Simple perspective transforms can be used to convert these images into a synthesized top-view image, which displays the scene as viewed from above the vehicle. These transforms, however, are typically derived using a basic calibration procedure that is only capable of correctly mapping ground-plane points in captured images to their corresponding locations in the top-view image, and subsequently, all off-the-ground points look distorted. We present a new method for calibrating a top-view image, in which objects and off-the-ground points are accurately represented.

We also present a method for using specifically designed disparity search bands to segment the scene in the overlapping field-of-view (FOV) regions between adjacent cameras, each pair of which is effectively a stereo imaging system. Such wide-baseline stereo systems with orthogonally diverging camera axes make stereo matching difficult, and traditional correspondence algorithms cannot reliably generate the dense disparity maps that might be computed in a parallel stereo set-up involving cameras that follow a rectilinear model. We segment the scene into the ground plane, objects of interest, and the background, and show that our new virtual camera calibration parameters can be applied to represent objects in the scene in a more realistic manner.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

A National Motor Vehicle Crash Causation Survey (NMVCCS) conducted between 2005 and 2007 found that 94% of crashes are caused by human error, and of these driver-related crashes, the largest error category is related to the driver's inattention, distractions, and inadequate surveillance [1]. Advanced Driver Assistance Systems that can provide drivers with more awareness of their surroundings and a wider field of view (FOV) are thus highly desirable.

This work focuses on improving the Texas Instruments ADAS, which involves four fisheye cameras placed on all four sides of a vehicle and generating a birdseye view image to display to the driver. No previous work has used the same set-up, so we will begin by discussing some background for this project, including hardware configurations and problems imposed by the set-up.

1.1 Fisheye

Fisheye lenses are frequently used in automotive imaging applications and surround-view system monitoring [2, 3, 4, 5], because they can capture wider FOVs, and fewer cameras are required to capture the entire 360^o outward

view. Fisheye images, however, suffer from inherent radial distortion and generally require distortion correction before further processing.

Fisheye images captured from four cameras placed with orthogonal principal axes on the front, sides, and rear of a vehicle can capture the entire surrounding 360^o view and can be stitched together to create an image as if taken from a virtual camera placed overhead. This image is referred to as the top-view, surround-view, or birdseye-view image, and is an integral part of an ADAS. It is easy for a driver to understand and quickly interpret, and it provides invaluable information about objects in a vehicle's immediate surroundings, to which a driver may be blind.

The ultimate goal is to generate a top-view image that is realistic, clear, and useful. Any objects in the scene should be realistically represented, and relative distances to the car should be clear, so drivers can make quick decisions about navigating a scene. The best way to obtain such a view is not inherently clear, so we approach this problem by considering the underlying stereo correspondence problems intrinsic to our set-up. We begin by describing top-view image generation to motivate this work, and then we narrow the focus to stereo correspondence. No previous work has explored stereo correspondence using cameras that have orthogonally diverging camera axes, a set-up which does require wide-angle lenses to ensure that there exists overlapping FOV regions between image pairs. Thus, one contribution of this work is introducing the problem and providing a method for dealing with the image data.

1.2 Top-view image generation

For top-view image generation, calibration of the system is required to understand how pixels in the captured images map to the desired view. A simple method for calibrating this four-camera fisheye system to generate a birdseye-view image is to use regular checkerboard patterns placed in the overlapping FOV regions between adjacent cameras, as shown in Figure 1.1. Then the radial distortion from each fisheye image can be corrected, and a perspective transform can be computed using knowledge of keypoints such as the corners of the checkerboards. From this procedure, a look-up table (LUT) can be derived, which specifies for each pixel in the top-view image, which pixel in the four captured fisheye images that pixel maps to. This look-up table can then be applied to any set of four fisheye images captured with the system. That is, for any set of images taken with the same four-camera set-up, the look-up table can be used to efficiently generate the corresponding top-view image.

However, the problem with this calibration method is that the derived perspective transforms used to compute the LUT are only able to correctly map captured ground-plane points to their corresponding points in the top-view image, and any points that do not lie in the ground plane in 3D space will not be correctly mapped and represented in the synthesized virtual camera image. Additionally, depending on how the views are stitched together, top-view images generated using this look-up table method can suffer from the blind spot problem, which is when objects in the overlapping FOV partially or

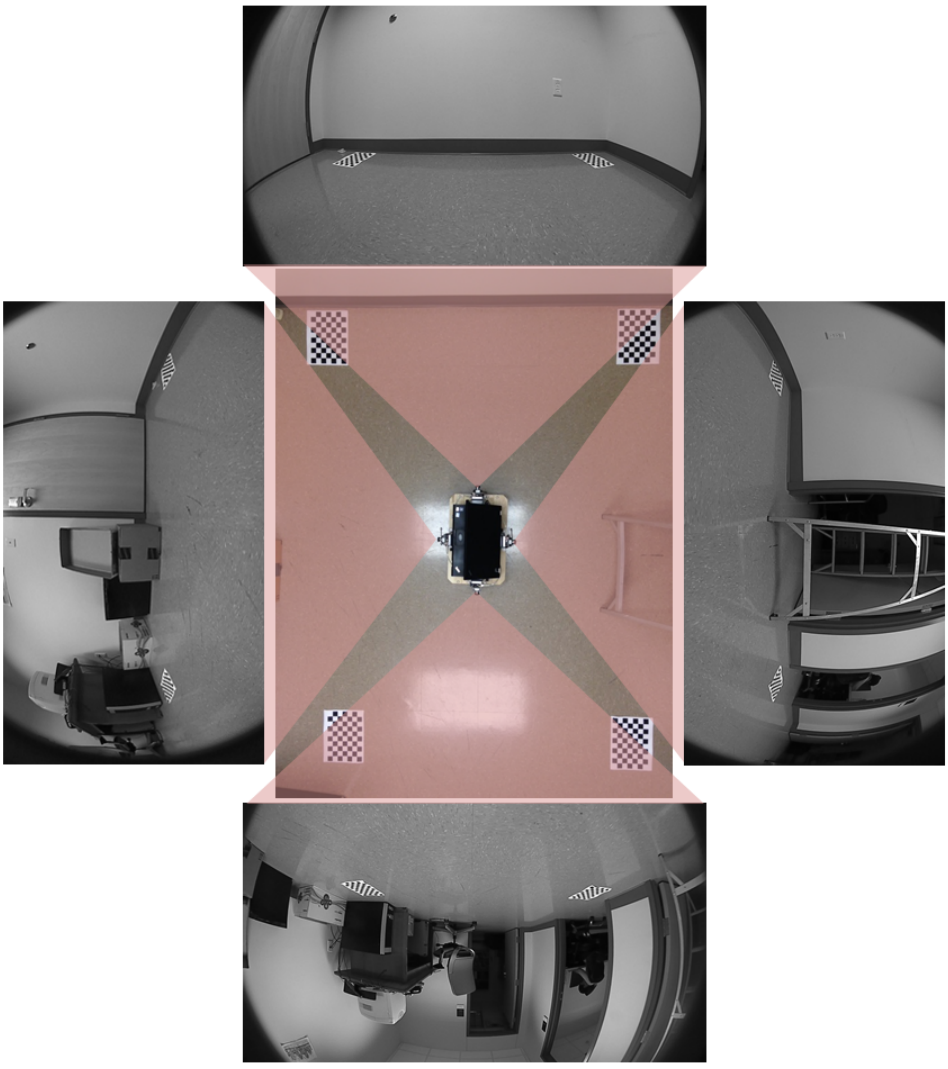


Figure 1.1: Surround-view image capture for calibration
Four fisheye images captured from front, side, and rear cameras and top-view image, with four calibration checkerboards placed in overlapping FOV regions between each pair of adjacent orthogonal cameras.

completely disappear. Figure 1.2(a) shows an example in which the front and right captured images are stitched together along a diagonal seam that bisects the overlapping FOV region. Because of the stitching method and the use

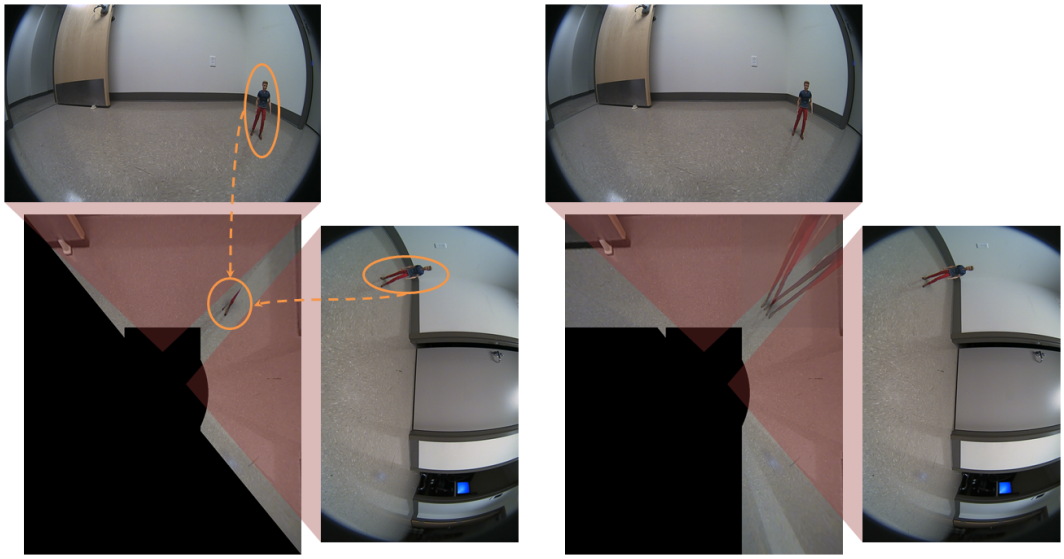
of the mapping LUT that was generated using only ground-plane projection geometry, the figure in the overlapping FOV almost completely disappears; only the feet are visible.

Figure 1.2(b) shows another method for combining the perspective-transformed images: alpha-blending, which renders the overlapping FOV region from both cameras with transparency, so the resulting image shows content from both images. In this method, however, still only ground-plane perspective transforms were used to transform the images, so any points not lying in the ground plane (e.g., the doll in Figure 1.2) become distorted in the perspective-transformed images, and alpha-blending the images produces a result that, like Figure 1.2(a), is also not realistic.

Because all points not coplanar with the ground are being incorrectly represented, we now ask how we can realistically represent objects in the top-view image, and we narrow our focus to the overlapping region between a pair of stereo cameras in our set-up. That is, we consider only the region of overlap between two adjacent fisheye cameras with orthogonal camera axes and ask how we can compute the correct mappings of off-the-ground points in this region.

1.3 Orthogonally-divergent wide-baseline fisheye stereo

In the region of overlapping FOV's between adjacent camera pairs, stereo vision can be exploited to extract information about objects that might be present in the region. Having an accurate and dense stereo map of the



(a) (b)

Figure 1.2: Examples of top-view images generated using LUT
Two examples of top-view images generated using the look-up table derived from ground-plane calibration data: (a) Blind spot problem with image stitching along a seam; (b) Double object representation problem with alpha-blending.

scene can provide valuable depth information that would be useful in scene reconstruction, which might then help improve scene and object rendering in the top-view image.

No previous work has explored stereo using a set-up with orthogonally diverging fisheye cameras. The wide FOV of the fisheye lenses allows for the camera axes to diverge with an approximately 90^o angle between them while still maintaining significant content overlap between the captured images. Typical stereo camera configurations involve rectilinear cameras, and thus, the cameras must be placed close to one another with parallel or converging (i.e., toed-in) camera axes in order to capture an overlapping FOV

between the stereo image pair. Additionally, cameras with orthogonally diverging axes that also have a greater distance between camera centers might also capture very different views of any objects that are visible to both cameras. The *baseline* refers to the line segment connecting the camera principal points, and again, the wide-angle nature of fisheye allows for a wider baseline compared to what is permissible in stereo with rectilinear cameras with parallel axes. Such different corresponding views requires using modified approaches to perform stereo matching and compute disparity estimates. We will present a method to compute these estimates and use them to segment the scene captured by the stereo cameras.

Chapter 2

Virtual Top-View Calibration[1]

As mentioned previously, the problem with typical calibration methods of placing calibration patterns in the ground plane and computing perspective transforms using corresponding points is that the derived perspective transforms work to correctly map only ground-plane points to their locations in the top-view image.

To address the problem of perceptually distorted object representation in top-view images, we have developed a new calibration procedure that computes parameters for finding the correct mapping of any point that has an associated accurate disparity estimate and that lies in the overlapping FOV region between cameras, regardless of whether it lies in the ground plane. Our method involves three main components to compute the necessary parameters: ground-plane calibration, stereo calibration, and height calibration. Ground-plane calibration is used to generate the look-up table to correctly

[1]The work presented in this chapter was recently published in [6]:
J. Pan, V. Appia, A.C. Bovik, "Virtual top-view camera calibration for accurate object representation," *Image Analysis and Interpretation (SSIAI), 2016 IEEE Southwest Symposium on*, March 2016.
J. Pan developed the presented method and is the primary author of the cited paper. She conducted this research under the supervision and assistance of A. C. Bovik and with support from V. Appia and Texas Instruments Incorporated.

map ground-plane points in the fisheye images to their corresponding points in the top-view image, and stereo calibration involves computing intrinsic and extrinsic parameters for any pair of adjacent fisheye cameras, but the images captured for ground-plane and stereo calibration can give much more information for understanding image structure and point mappings, which we use to compute additional calibration parameters that allow for more accurate representations of objects in the synthesized top-view image.

In this chapter, we first present a summary of the images and data required for each component of calibration. Then we describe the details of each calibration step and how to compute the relevant parameters. Finally, we summarize the derived calibration parameters that allow for computing correct mappings of points in overlapping FOV regions between adjacent cameras.

2.1 Data collection

The system set-up has four fisheye lenses (Sunex DSL219) and four camera sensors (OV10635) mounted with approximately orthogonally-oriented camera axes. A preliminary round of calibration is performed to generate a foundation for the rest of calibration. This process involves fisheye distortion removal and computation of perspective transforms using the calibration checkerboard patterns shown in Figure 1.1, followed by stereo calibration using the MATLAB Stereo Camera Calibrator application [7]. These two initial calibration steps generate the look-up table for the fisheye-to-top-view mapping of ground-plane points and the camera intrinsic and extrinsic parameters

for rectification of stereo image pairs.

We introduce additional calibration steps involving ground-plane, stereo, and height data that allow for the generation of images in which objects are represented with higher fidelity with respect to the position of the overhead virtual camera. Table 2.1 summarizes the kind of data required for each step.

Though we list only one set of images as *required* to compute calibration parameters in both the ground-plane calibration and height calibration steps, capturing more images is recommended to improve accuracy, and in our tests and implementations, we collect more than one set and compute averages of parameters.

Further, we assume that preliminary calibration of the system has been performed. In other words, we assume that we have the look-up table, which maps points captured in the four fisheye images to a virtual camera image, as well as stereo camera intrinsic and extrinsic parameters for image rectification.

2.1.1 Ground-plane Images and Point Selection

Four exemplar fisheye images and the top view image required for ground-plane calibration are shown in Figure 1.1. We will continue with only considering the overlapping FOV between the front and right cameras, because all calibration steps performed for one stereo pair can be applied to the other three adjacent camera pairs. Thus, while complete ground-plane mapping calibration requires all four images, only the front and right images are actually required to calibrate the overlapping region between the front and

Table 2.1: Summary of images and points required for virtual camera calibration

Calibration step	**Calibration surface**	**Images**	**Number**	**Selected points**
Ground-plane	On ground plane in overlapping FOVs	4 fisheye images (one from each camera)*	1 set	Checkerboard corners; keypoints
		Top-view image	1 for each set of fisheye images*	Checkerboard corners
Stereo	Orthogonal to ground plane in overlapping FOV	Stereo image pairs	10 pairs for MATLAB calibration app	Rectangular surface corners
Height	Orthogonal to ground plane in overlapping FOV	Stereo image pairs**	1 pair	Rectangular surface corners
		Top-view image	1 for each pair of stereo images**	Rectangular surface corners; camera locations
		Look-up-table-generated top-view image	1 for each pair of stereo images*	Ground-adjacent checkerboard corners

right cameras.

Rectified stereo images taken of the checkerboards also contain more

information descriptive of the relationship between the disparity and the y-coordinate of any ground-plane point in the disparity map. Specifically, knowing a set of corresponding ground-plane points between a pair of stereo images allows us to derive the relationship between y-coordinate and disparity, which is linear. If the black squares in the checkerboard pattern are selected as keypoints, as marked in Figure 2.1, the horizontal disparity at each keypoint p_i is

$$\text{disp}_i = x_i^l - x_i^r,$$

where x_i^l and x_i^r are the x-coordinates of keypoint p_i in the left and right rectified images respectively.

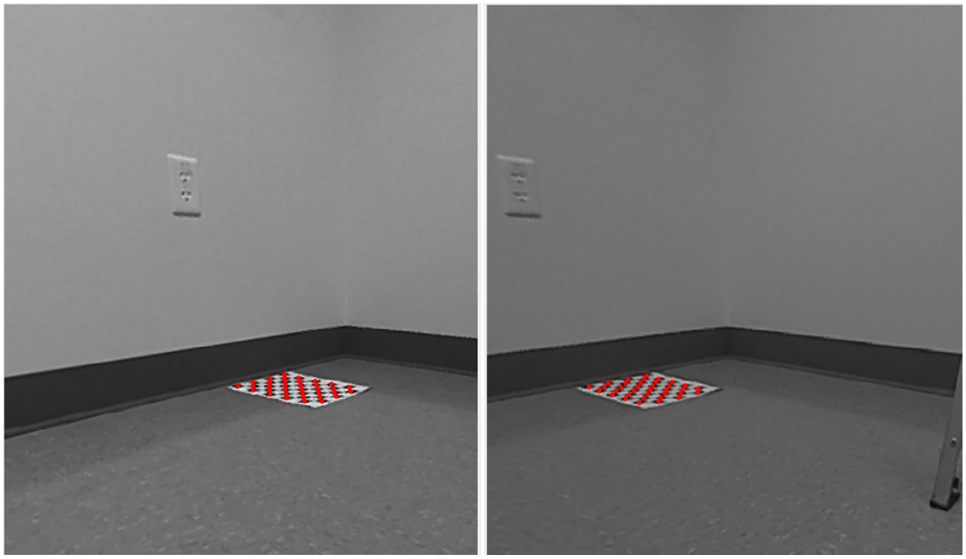


Figure 2.1: Stereo images for deriving y-disparity relationship
Front and right rectified stereo images of the checkerboard pattern placed in the overlapping FOV. The black squares in the checkerboard are marked and can be used to derive the y-disparity relationship.

We assume vertical disparity is approximately zero, so we will use *dis-*

parity to simply mean *horizontal disparity*. Thus, we assume the left and right y-coordinates of each keypoint are equal, i.e., $y_i^l = y_i^r = y_i$.

Using the disparities $\mathbf{disp} = (\text{disp}_1, ..., \text{disp}_N)$ and y-coordinates $\mathbf{y} = (y_1, ..., y_N)$, for N keypoints, the linear relationship can be derived. That is, the slope m and y-intercept b of the line that best fits the data can be computed:

$$\mathbf{y} = m \times \mathbf{disp} + b. \tag{2.1}$$

So for any point p_k in the disparity image, regardless of whether it is a ground-plane point, knowing the disparity disp_k is sufficient for computing the y-coordinate of its corresponding ground-plane point, which can also be thought of as the image of the ground-plane projection of point p_k.

2.1.2 Stereo Images and Point Selection

It is assumed that stereo camera calibration was performed to obtain camera intrinsic and extrinsic parameters for rectifying stereo image pairs. The calibration process uses the MATLAB Stereo Camera Calibrator application [7] and requires capturing multiple pairs of images in which a checkerboard pattern is positioned and oriented differently in each pair.

In addition to helping to make it possible to compute estimates of camera parameters, stereo image pairs also provide information for mapping non-ground-plane points in the disparity map to the top-view image. In the rectified images, the user should select the corners of the rectangular surface in all stereo pairs to accurately derive calibration parameters.

While the y-disparity relationship (2.1) derived using ground-plane calibration images allows for finding the y-coordinate of a ground-plane point, to determine the x-coordinate of the ground-plane projection of any point in the disparity map, one must make use of the geometry in rectified pairs of stereo images of the checkerboard used in stereo calibration.

Assuming the checkerboard surface is orthogonal to the ground plane, we compute a vanishing point for the image of all lines that run orthogonal to the ground plane. All vertical lines in world coordinates, when seen in the rectified stereo pair or the disparity map, will intersect at a common vanishing point. Figure 2.2 shows the rectified window of images taken with the front camera. The dotted lines are vertical edges of the rectangular calibration plane when it is orthogonal to the ground plane, so all the imaged edges are parallel to one another and should intersect at some common vanishing point.

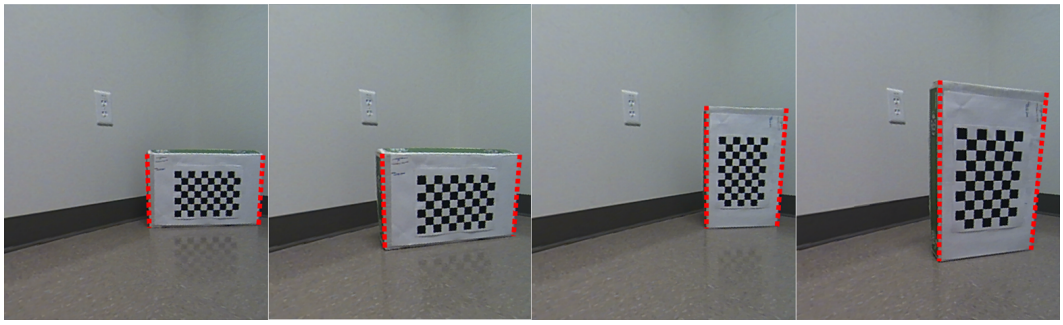


Figure 2.2: Four examples of the rectified window from the front camera All dotted lines are vertical in 3D space and are, therefore, parallel in 3D space.

After computing the vanishing point, which is the intersection of any two vertical edges, we know that for any point in the disparity map, the x-coordinate of its ground plane projection lies on the line segment connecting

it to the vanishing point. Linking this x-coordinate with the y-coordinate computed using the linear y-disparity relationship derived previously (2.1), we can obtain the orthogonal (as considered in 3D space) ground-plane projection of any point in the disparity map for which we have a disparity estimate.

2.1.3 Height Images and Point Selection

To compute the final calibration parameters, we require knowledge of the camera locations in the captured top-view image and all four corners of the rectangular calibration surface in the rectified stereo images and in the top-view image. In the mapped, i.e., synthesized, top-view image, we also need the ground-adjacent corners of the rectangular calibration surface. Since they can be assumed to lie in the ground plane, they can be considered correctly mapped using the look-up table. Knowing corresponding ground-plane points in all the images tells us the FOV in the top-view image that is represented in the rectified images.

The relationship between the captured and mapped top-view images determines the FOV of the captured image to which it should be cropped. If the desired virtual camera image has dimensions 1080×880, the correct stretch and shift factors can be computed by aligning any two pairs of corresponding ground points in the captured and synthesized top-view images. These stretch and shift parameters can then be applied to any captured top-view image to obtain the 1080×880 virtual surround-view image.

The next step is to compute the virtual camera principal point, or the

center of the top-view image, which is crucial for computing projections in the virtual camera view. This point is where the principal axis of the virtual camera intersects the image plane. Lines that are orthogonal to the ground plane in 3D space, when captured in top-view and extended beyond their endpoints should intersect at the image center. As shown in Figure 2.3, from the user-selected corners of the rectangular calibration plane in the top-view, the image center can be computed by finding the point of intersection between the lines that run through the left and right edges of the rectangle.

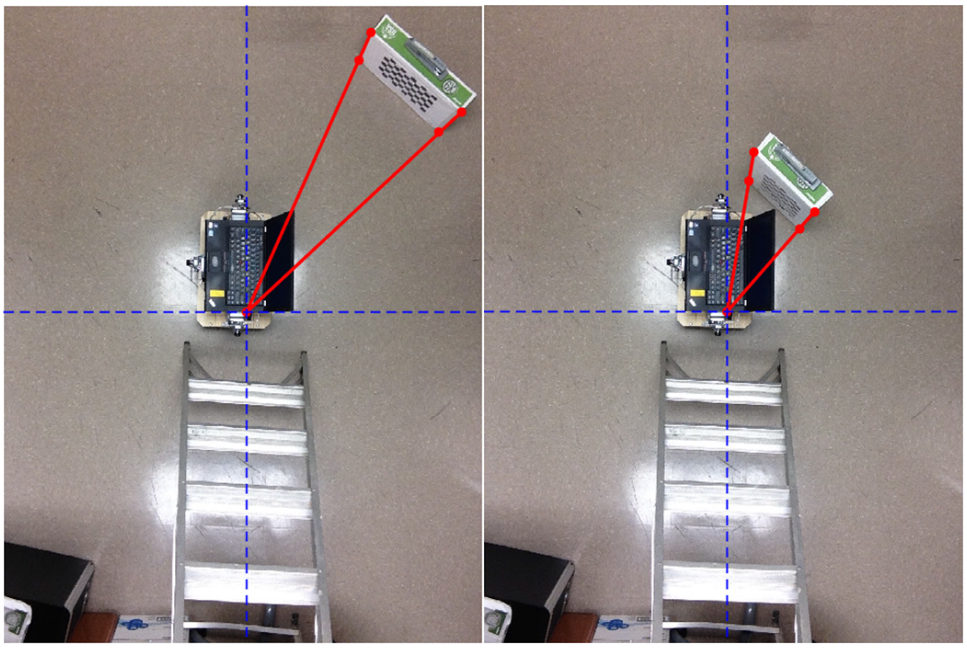


Figure 2.3: Examples of top-view images showing height projection
The same calibration plane (oriented differently) is placed in different locations/depths from the image center. Extending the line segments that represent the side edges of any orthogonally-placed rectangular plane and finding their intersection can help compute the virtual camera's principal point.

We next derive the depth-disparity relationship:

$$\text{depth} = \frac{f \times B}{\text{disp}}, \tag{2.2}$$

where depth is measured as the orthogonal displacement from the baseline between a stereo camera pair. Thus, knowing the camera locations also allows for computing the baseline parameters, and the depth of any ground-plane point in the top-view can also be computed. Disparity estimates in the disparity map have an inversely proportional relationship to depths. Figure 2.4 shows the extended baseline between the front and right cameras, while the solid line segments represent the depth of the bottom two corners of the rectangular surface.

To derive the depth-disparity relationship, we use the known locations in the rectified stereo pair of the ground-adjacent corner points. Using the known disparity and depth at each point, the constant factor fB can be computed in (2.2), so for any point in the disparity image which has a disparity estimate, the depth of its ground-plane projection can be computed.

The final calibration parameter is the ratio that relates depth to the ratio between imaged height h_J and projected height h_T. Figure 2.2 shows imaged heights, assuming the bottom two corners are the ground-plane projections of the top two corners. The projected height of a point is the magnitude of the line segment in the top-view between the mapping of that point and the mapping of its ground plane projection (Figure 2.4).

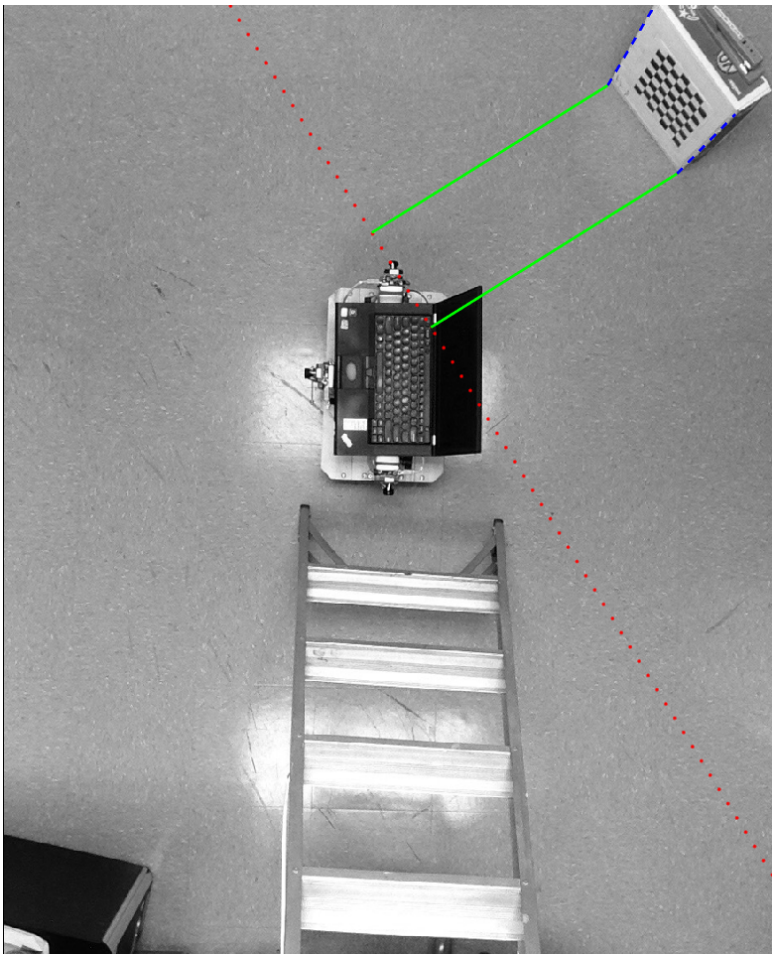


Figure 2.4: Top-view image showing baseline and depths
Top-view image showing the extended baseline between the cameras (dotted line) and the lines (solid) along which depth to the rectangle's bottom two corners are measured. The dashed line segments represent the magnitude of the projected height of the top two corner points.

Using the images and selected keypoints (Table 2.1), we can compute constant C in the relationship:

$$\text{depth}_k = C \times \frac{h_{k,T}}{h_{k,J}}, \tag{2.3}$$

where the ratio of between imaged height and projected height is constant given depth.

2.2 Summary of parameters and procedure

The three calibration modules described in Section 2.1 should supply sufficient parameters to compute the correct mapping of any point p_k in the disparity map for which an accurate disparity estimate is available. We can roughly classify the parameters into three groups: resizing parameters, projection parameters, and stereo parameters. See Table 2.2 for a summary of the parameter classification.

Table 2.2: Summary of all the calibration parameters

Resizing Parameters	**Stereo Parameters**	**Height Parameters**
- Stretch factor - Shift factor - Shift factor	- Stereo camera baseline coefficients - Stereo camera center (baseline midpoint) - FOV boundaries in top--view image - Vertical vanishing point	- Virtual camera center - y-disparity coefficients - fB - C

Resizing parameters allow for transforming captured top-view image points to their correct locations in the 1080×880 output view. Stereo parameters are associated with rectified image pairs, and projection parameters mainly pertain to the output top-view image, but both help map non-ground-plane points of known disparities to their projections in the virtual camera

output image.

The method for computing the correct mapping of any point $p_{k,J}$ in the disparity map using the derived parameters is:

1. In the disparity map:
   (a) Compute the ground-plane projection $p_{0,J}$ using the y-disparity relationship (2.1) and the vertical vanishing point.

2. In the output top-view image:
   (a) Compute the mapping $p_{0,T}$ of $p_{0,J}$ using the depth-disparity relationship (2.2), knowledge of baseline between the two cameras, and knowledge of the overlapping FOV in the top-view image.
   (b) Find the line l_k through the virtual camera principal point p_o and $p_{0,T}$.
   (c) Compute the projected height $h_{k,T}$ of $p_{k,J}$ using the relationship between depth and height ratio (2.3). The displacement $h_{k,T}$ from $p_{0,T}$ along l_k gives the desired mapping of of $p_{k,J}$ in the virtual camera view.

Chapter 3

Scene Segmentation

In a driver assistance application, stereo vision can greatly enhance obstacle detection and avoidance capabilities to further increase safety when obstacles may not be located in a driver's line of sight. Being able to understand and compute stereo correspondences can help generate disparity maps, which can give invaluable information about scene geometry and contents, because depth can be computed directly from disparity estimates. A lot of work has been done in developing stereo correspondence algorithms [8, 9, 10, 11, 12, 13, 14], and some research has even focused on wide-baseline applications [15, 16, 17, 18], however most wide-baseline stereo methods rely on feature-based or region-based matching with a goal of computing homographies, epipolar geometries, or simply finding sparse matches to assist in particular applications like object detection and recognition. With our application, however, we want a more generally-applicable method for computing disparities and understanding scene structure. Thus, we focus on the broader problem of scene segmentation.

Specifically, we propose a method for segmenting the scene in the overlapping FOV region between adjacent camera pairs into the ground plane,

objects or obstacles of interest, and the background. Such a segmentation can provide a simple way for a driver to quickly visualize the geometry of their surroundings. An accurate and dense disparity map alone would be enough for scene segmentation. However, stereo matching with our orthogonally diverging fisheye camera set-up has been a perpetually difficult problem, even with commonly used stereo matching methods like Semi-Global Matching (SGM) [12], because the cameras can capture widely disparate views of objects in the scene, which makes computing point correspondences more difficult. Also due to our specific set-up is the issue that the disparity range over which an algorithm like SGM must search is very wide. For example, in our system, with a baseline distance on the order of centimeters, the disparities of points can range from zero (points at infinity, e.g., points on the horizon) to more than 200.

Most dense pixel-based stereo matching algorithms (such as SGM) require a disparity range over which to search for matches. With a wide search range, however, the confidence in computing accurate stereo matches decreases. Therefore, we will discuss an approach based on decomposing the disparity search space into bands to focus on extracting more accurate disparity information for different parts of the scene. In this project, we use SGM as our stereo matching algorithm, but our proposed method involves applying SGM in a multi-level approach to get accurate disparity estimates and develop a better understanding of the scene structure.

3.1 Disparity Band Disparity Maps

Because disparity is directly associated with depth, specific disparity ranges can reveal parts of the scene at certain depths of interest. For an example, see Figure 3.1, which shows the the original captured pair (Figure 3.1(a)), the processed images after fisheye distortion has been removed (Figure 3.1(b), and the windows of the overlapping FOV between the stereo images after the undistorted images go through stereo rectification (Figure 3.1(c)). In this example, the largest disparities correspond to the ground plane points closest to the cameras, because depth and disparity have an inverse relationship. Points with the largest disparities are also frequently the most disparate in representation, i.e., their capture angles are the most different, so between the stereo image pair, the captured scene around these points can look very different, which makes computing correspondences very difficult.

Because of the widely differing views of certain points and regions in the scene, accurate, dense, and complete disparity maps are difficult to obtain. The ground plane, in particular, presents a huge problem for computing disparities. The ground plane usually requires the largest disparity search range and includes the largest disparity values, because most of the closest, i.e., shallowest in depth, points in any captured scene will belong to the ground plane. Objects of interest in the scene will also belong to their own disparity search ranges; as will background pixels, which will have disparities near zero.

Additionally, the reliability in estimating disparities varies for different parts of the scene as well. For instance, computing a disparity map for a dis-

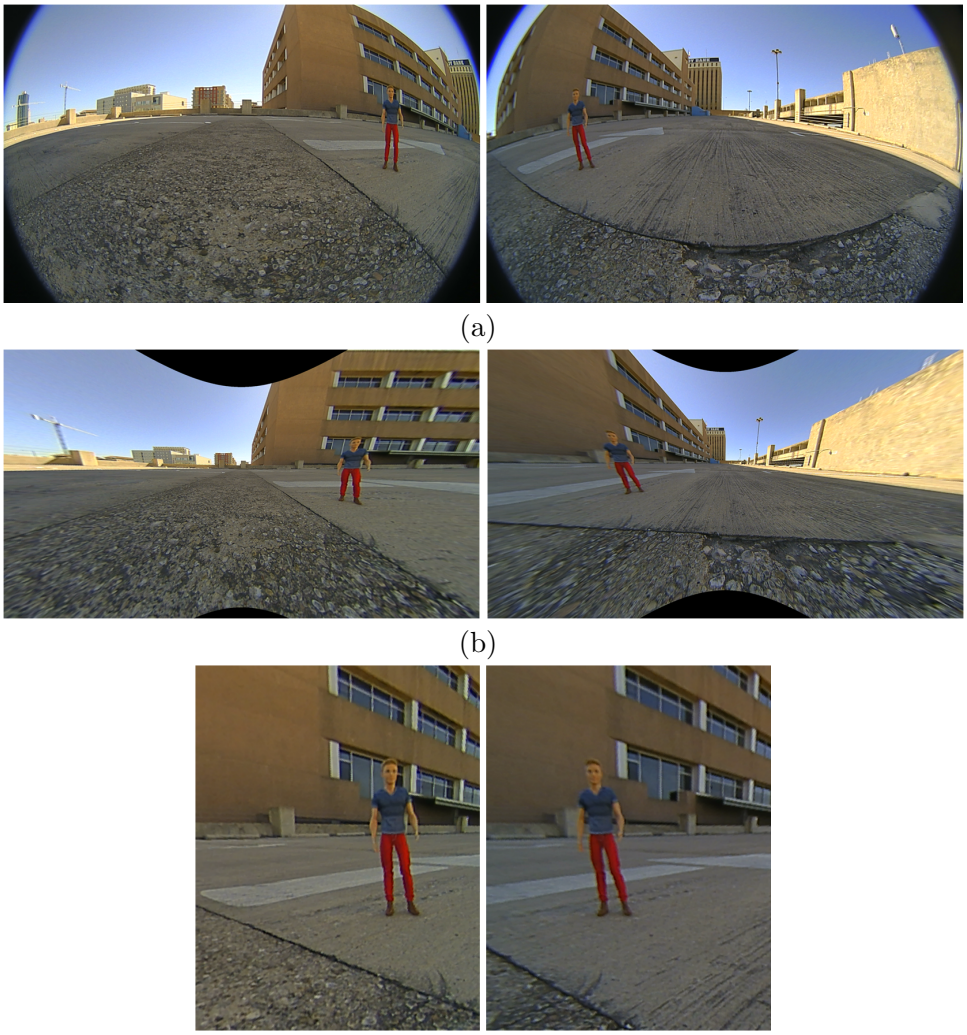


Figure 3.1: Processing pipeline of captured fisheye images
(a) Captured images, (b) After fisheye removal, (c) Rectified overlapping FOV

tinct object with plenty of discernible features can be reliable across different parameter values (e.g., different uniqueness thresholds, different window sizes, etc...). However, being able to obtain dense ground plane disparity estimates would be more sensitive to parameter tuning, because the ground plane spans

a very wide range of disparities and contains points with the largest disparities. Due to the orthogonality of our stereo set-up, the disparity search ranges for stereo correspondence algorithms need to be much wider, and the disparities need to be much larger than what would be required with typical parallel stereo applications.

To address the problem of obtaining accurate disparity estimates for different objects or regions in a scene, we begin by creating different disparity bands with ranges defined for computing only a subset of the entire disparity map at a time. Specifically, we want to avoid generating a single disparity map for which the entire span of the disparity search range (0 to 200+) is used in one iteration of SGM, because for different parts of the scene, there is no need to search over certain disparities. The sizes of the bands we need to use are, in general, proportional to the disparities and inversely proportional to 3D point depth. In other words, for closer points with larger disparities, such as ground-plane points for which point correspondences may be more ambiguous, a larger disparity search range is necessary, and we know that we do not need to search over smaller disparities to obtain estimates for almost all the ground-plane points. Additionally, one goal with this method is to extract the ground plane by estimating its parameters, and to do so, using a tuned range of disparity values is sufficient.

While we exponentially decrease the size of the disparity search range as the disparity values decrease, we also want to check for consistency in the disparity estimates, and to do so, we first use overlapping bands and a method

that involves linearly decreasing the disparity search values and exponentially decreasing the range of the search depending on the disparity magnitude and region of interest. Specifically, we begin by dividing our disparity range into three categories: near, medial, and distant. For our set-up, we consider disparities in [132, 256] as *near*, [64, 192] as *medial*, and [0, 100] as *distant*, where the bounds between the groups are soft and overlap quite a bit. These bounds were heuristically selected for our particular set-up. For stereo set-ups of different scales and dimensions, different disparity ranges can be used.

We also search over different numbers of disparities when computing stereo correspondence in each range. For near disparities, the search interval size is $num\_disp = 64$, which is quite large; for medial disparities, $num\_disp = 32$; and for distant disparities, $num\_disp = 16$, which is the smallest allowed search size. We use this exponentially decreasing pattern, because to obtain a single disparity map with enough sample points from which we can estimate the ground plane, which may contain points spanning a wide range of disparities, we need to use a wider search range. Distant points, on the other hand, should all have disparities near or approaching 0, so the smallest search range of 16 allows us to obtain disparity maps which contain fairly accurate and dense estimates of the entire *distant* space.

Specifically, if we let N denote the set of all near intervals, M denote the set of all medial intervals, and D denote the set of all distant intervals, or

$$N = \{[i, i+64] \mid i \in \{132 : 4 : 192\}\}, \tag{3.1}$$

$$M = \{[j, j+32] \mid j \in \{64 : 4 : 160\}\}, \tag{3.2}$$

$$D = \{[k, k+16] \mid k \in \{0 : 4 : 84\}\}, \tag{3.3}$$

then we compute disparity maps for the entire set of intervals $N \cup M \cup D$. We use step sizes of 4 between successive intervals, because while such a small step size generates redundant disparity estimates, the individual maps generated for overlapping disparity ranges can help provide additional accuracy checks for estimating single surfaces or objects. Figure 3.2 shows the disparity maps estimated for this set of intervals using the stereo images shown in Figure 3.1.

It is important to note that we simply use *medial* to describe a set of disparity bands of relatively average size (compared to the largest and smallest band sizes we design) that include relatively average disparity values (compared to the near and distant bands used for our particular set-up). These medial bands work for visualizing objects in our examples, but objects can be located in the near disparity range as well, so we will redefine disparity search ranges for the specific task of object detection in Section 3.3.

3.2 Ground Plane Estimation

To estimate ground plane parameters, we use the random sample consensus (RANSAC) [19] algorithm, which has already been shown to be effective in fitting ground planes in stereo vision applications [20, 21, 22, 23, 24, 25].

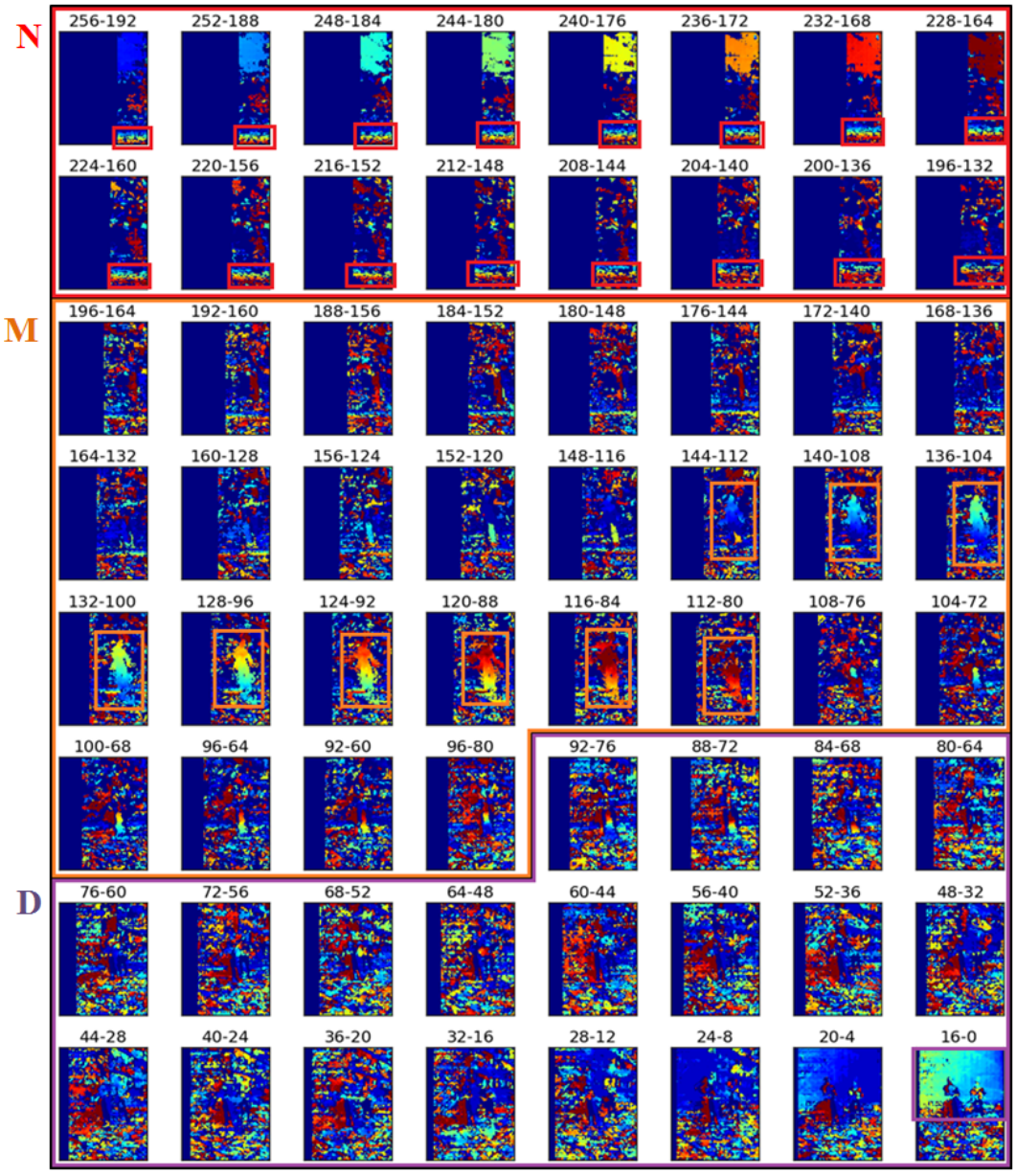


Figure 3.2: Disparity maps computed with the disparity bands in the near, medial, and distant ranges

In this section, we will discuss the theory behind RANSAC and similar approaches others have taken for ground plane estimation, and then we will describe in detail our ground plane estimation method.

3.2.1 RANSAC

The idea behind RANSAC [1] is to randomly sample from a set of experimental data, use the samples to instantiate a model (i.e., instantiate its parameters), and then use that model to determine how much of the experimental data falls within some error tolerance of being considered inliers to that model (i.e., how much of the experimental data supports the model hypothesis). The set of data that supports the model is called the consensus set, and if there are too many errors (determined using a threshold), the consensus set is used to derive a new model hypothesis. If the model is sufficiently supported, then a new subset of experimental data points is randomly selected, and the process of instantiating a model and evaluating its support is repeated. After a predetermined number of trials, if a large-enough consensus set has not been found, the model with the largest consensus set can be used to determine the model of best fit. (Terminating in failure is also an option.)

Proven approaches usually make use of the disparity map, which assigns an estimated disparity value to each pixel in the reference image. A straightforward approach is to subsample the disparity map and apply RANSAC, which just involves hypothesizing a plane and finding inliers [20, 21]. Another method, if the ground plane homography is known, is to randomly select a

point, find a sufficient number of neighbors that are not collinear to compute a homography H, and try to fit the estimated homography to the known one. If it fails to fit, re-select a random point and start over, but otherwise, the estimated homography can be used to find more inliers and re-compute H to fit more and more inliers until it converges.

RANSAC has also been used for spatio-temporal estimation of the ground plane using spatio-temporal range image data from a Time-of-Flight (TOF) camera [23]. Their imaging set-up involves equipment we do not have, but it is still helpful to see how they applied RANSAC. Their TOF range imaging devices are robust against shadow, brightness, and poor visibility, which unfortunately, our SGM method is not. They model the ground plane as a 3D feature with two spatial directions and one temporal direction. To compute a ground plane hypothesis, they require the random selection of four data points, and then they compute three independent vectors lying in the spatio-temporal ground plane feature. Then, the 4D cross product of the vectors is computed to obtain a model for the ground plane, which is then evaluated in the standard approach of the RANSAC algorithm. Finally, the best model is used to identify inlier and outlier points of the range image data, and the final ground plane is computed using maximum likelihood (ML) estimation on inlier data.

3.2.2 RANSAC for ground-plane estimation

Even if we do not have a complete disparity map, as long as we have reliable disparity estimates for ground-plane points, we can use RANSAC to fit

a planar model to those points. In doing so, we make a key assumption about the search space, which is that the ground plane begins at the bottom of the rectified frames, extends deep enough in the scene and, therefore, high enough (in the vertical direction) in the disparity map. We make this assumption and make sure the ground plane is sufficiently represented in our test images so that we can show that our method of ground-plane extraction can work on real data.

For ground plane data, we use the disparity maps generated using the near disparity intervals in set N (3.1) and run RANSAC on data points (x, y) with $x \in [x_{i1}, x_{i2}]$, $y \in [y_{i1}, y_{i2}]$, and $i \in \{132 : 4 : 192\}$ as defined in (3.1). We will use I_n to denote this set of points (x, y) that are used from disparity map n, for $n \in \{1, ..., |N|\}$.

The parameter x_{i2} is the width of the disparity map, which is constant in our case, and $x_{i1} = i + 64$, which is the upper bound on the disparity search range, because in computing stereo correspondence between two rectified windows of the same dimensions, the smallest horizontal position for which a disparity can be estimated is the largest disparity in the search range. The parameters y_{i1} and y_{i2} can be computed using an estimated y-disparity relationship (2.1), which can be derived using empirically calculated disparities or using the process as described in Section 2.1.1. These vertical limits describe the rows in each map that may contain the ground plane points that fall in the disparity range used to compute that map.

We use the RANSAC implementation from [26] and modified it for our

application. Running RANSAC requires the selection of three important parameters: the sample size, a limit on the number of iterations before simply selecting the best model found so far, and a target number of inliers for each model to explain. Typically, the sample size is chosen to be the minimum number of samples that allows you to fit a model. In our case, since we are trying to fit a plane, we choose the sample size to be three, so the algorithm will begin by selecting three random data points among the set I_n and computing the plane which describes those points. Then the rest of the points are classified as being inliers to the plane or outliers to the proposed model, and if a sufficient number of points are inliers, meaning that the estimated plane describes enough of the points (i.e., the target number of inliers), then that model is taken to be the estimated ground plane from the set of data points from that particular disparity map, which was computed using a specific disparity search range. If the target number of inliers is never reached before the iteration limit is reached, then the model describing the most inliers is selected. We estimate a ground plane using this method for each disparity band in N and record the inliers across all bands.

Figure 3.3 shows the estimated ground plane and the inlier points that describe the estimated plane in each disparity band, and Figure 3.4 shows the estimated ground plane using all the inliers across the sixteen bands shown in Figure 3.3.

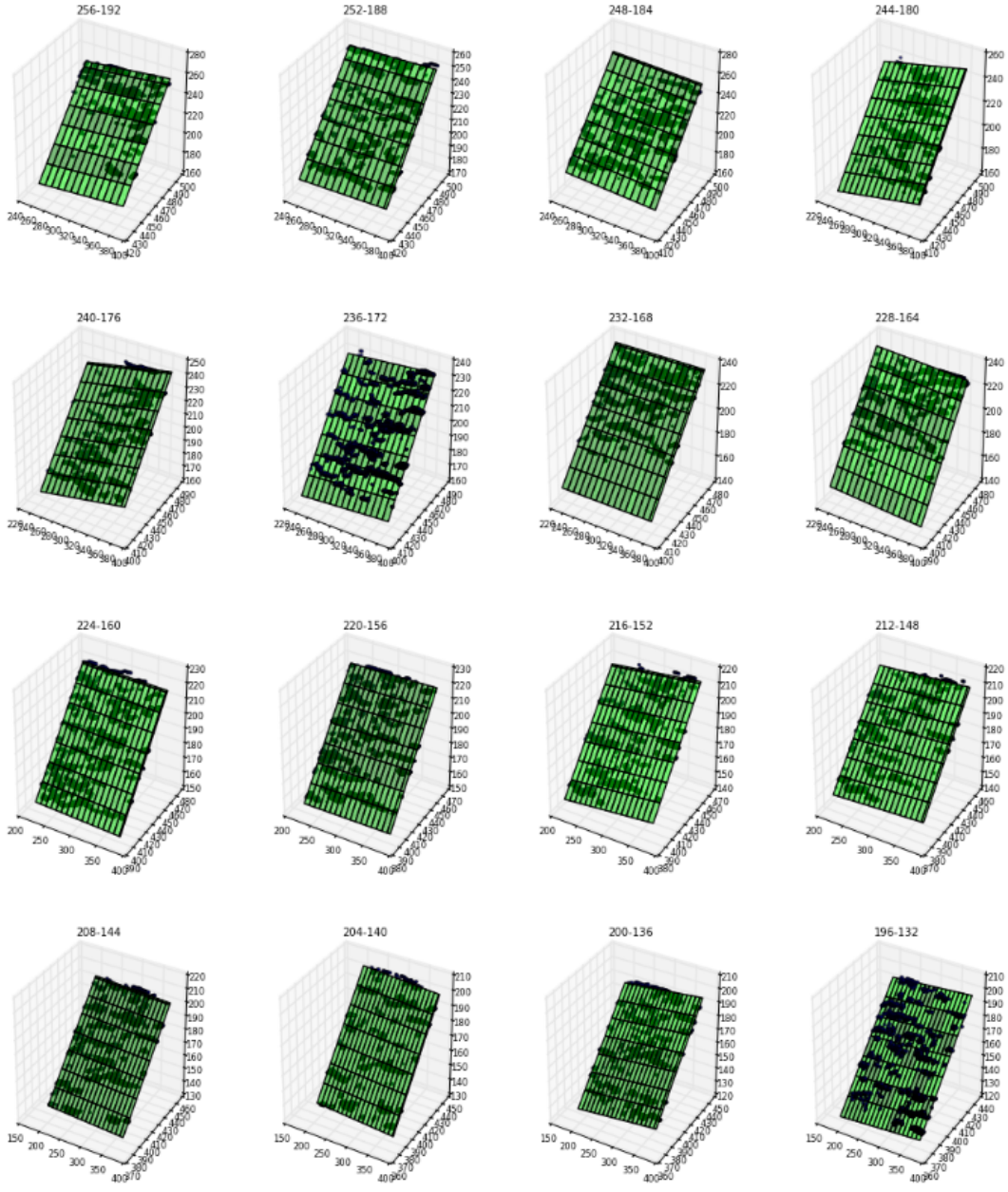


Figure 3.3: RANSAC fits for ground plane points in the disparity maps generated with disparity ranges in set N

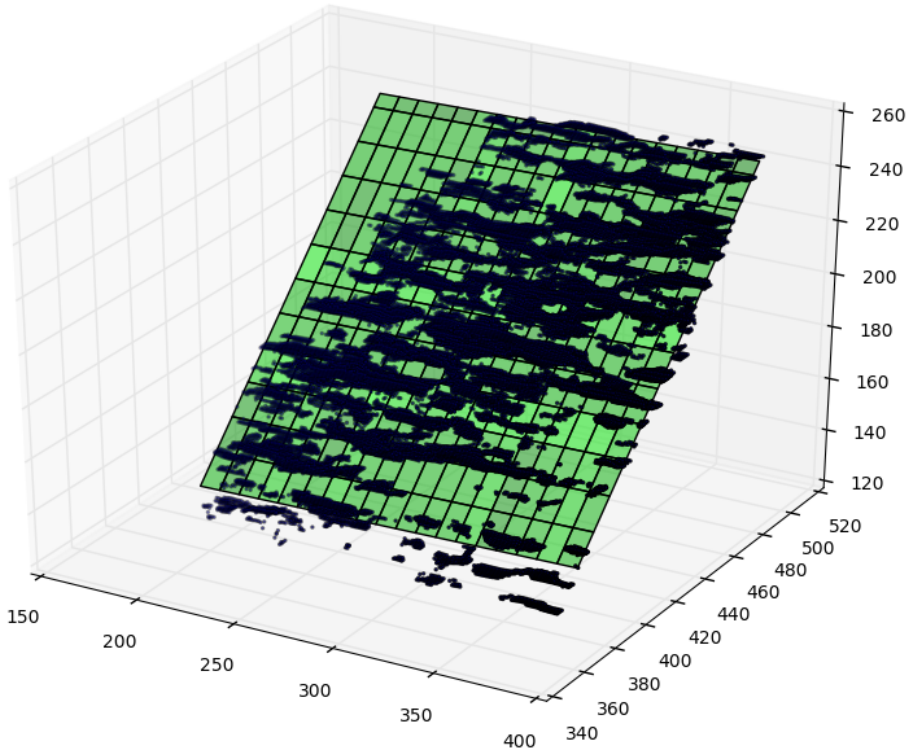


Figure 3.4: RANSAC-estimated ground plane using inliers across all disparity bands in N

Using all the inliers in an additional run of RANSAC is one way to obtain a single ground plane estimate, which is what we ultimately want. Any of the estimates shown in Figures 3.3 and 3.4 can be used as the ground plane model, but to select only one, we compute how well points in other disparity maps fit in each estimated model. Specifically, if we want to see how well a model m fits the data in map n, for $m, n \in \{1, ..., |N|\}$ and $n \neq m$, we compute the percentage of points in I_n that can be considered inliers to the

model estimated for map m. If we let p_{mn} denote the percentage of inliers in I_n to model m, the overall performance p_m of a model m can be averaged over its performance in specific disparity bands:

$$p_m = \frac{1}{|N| - 1} \sum_{n=1, n \neq m}^{|N|} p_{mn},$$

so continuing, we simply select the model m with the largest p_m.

3.3 Object Extraction

Having an estimated ground plane is very helpful moving forward, because it can assist in detecting objects and obstacles in the scene in order to effectively display them for a driver, which may be the most important feature in a driver assistance application. To detect objects or obstacles, we also use an array of disparity bands. In Figure 3.2, we can see that the object of interest becomes easily detectable in some of the maps in set M. We, as humans, can easily detect the shape of the object of interest, because the human visual system is so powerful; however, creating a robust method for automatic object detection is not so easy. In this section, we present a method for automatically computing a convex hull around detected objects in a disparity range of interest.

3.3.1 Disparity Bands for Object Extraction

Though we defined the set of medial disparity bands M in (3.2) and can see the object of interest using these ranges in Figure 3.2, these intervals do not include large disparities, i.e., shallow depths. Objects can certainly be close to the cameras and have large disparities that may only be correctly estimated using the bands in N (3.1), so for the task of detecting relevant objects in the overlapping view between the cameras, we define a new set of disparity band search ranges that span the entire disparity range in the scene with a finer granularity.

We use disparity bands of a constant size and perform stereo matching from the largest disparity (i.e., shallowest depth) all the way down to zero disparity (i.e., distant depth). Specifically, we use overlapping bands of size 16 that span the entire range of disparities in our scene $[0, 256]$:

$$B = \{[i, i + 16] \mid i \in \{-16 : 4 : 256\}\}. \tag{3.4}$$

Figure 3.5 shows disparity maps computed with these overlapping bands. Clearly, there are a lot of unreliable estimates as well as noise, and to filter out some of them, we propose the use of a consistency check, in which we search over all bands for consistent disparity estimates, which indicate high-confidence estimates. After running such a check at each pixel in the disparity map, we obtain a *consistency map* containing values only at pixels with the highest-confidence disparity estimates among the many erroneously-estimated stereo correspondences.

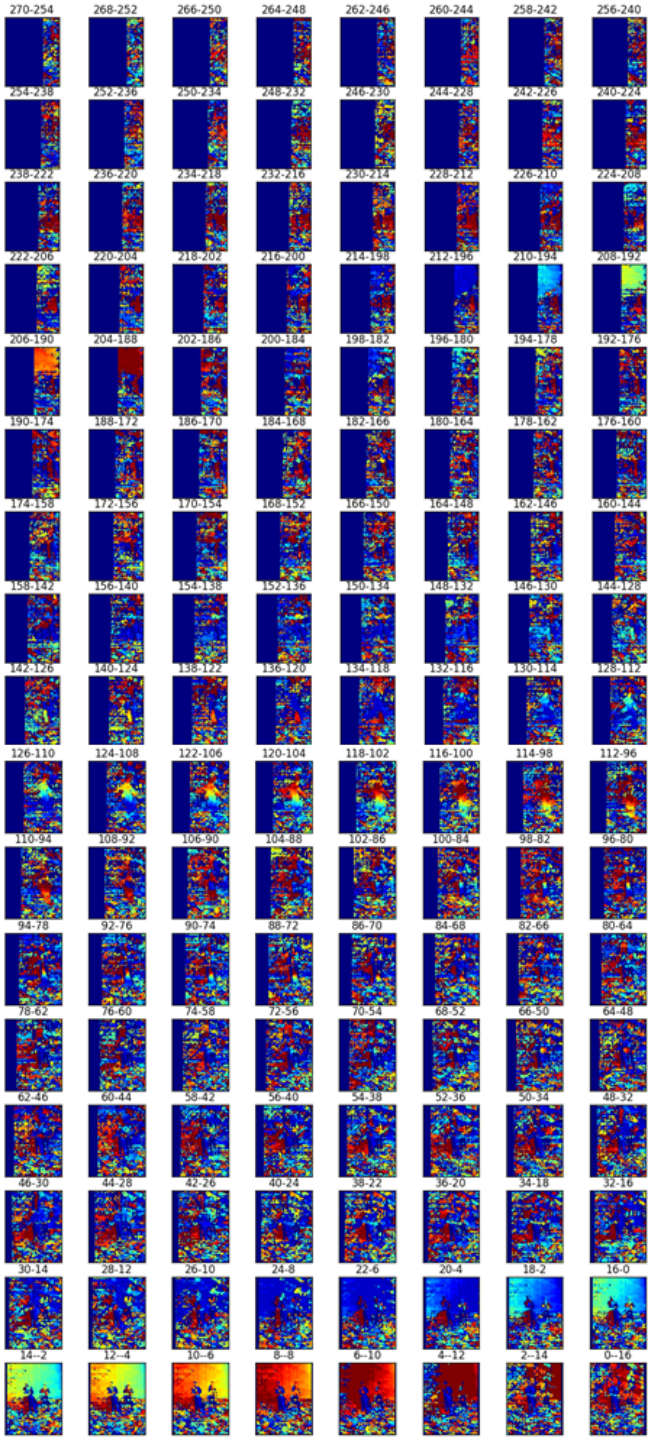


Figure 3.5: Disparity maps generated using narrow equally-sized disparity band search ranges that linearly decrease with overlap (3.4)

3.3.2 Consistency maps

If the stereo maps are stored in a 3D array S of size $h \times w \times n$, where h and w are the height and width of a single map, and n is the number of maps, then for computing a consistency map that checks for an integer consistency c, we search over all maps in the array $S[i, j, :] \ \forall \ i \ \epsilon \ [0, h-1], j \ \epsilon \ [0, w-1]$ and search for a repeated sequence of c high-confidence disparity estimates. In our application, each disparity value (from 0 to 256) is actually searched for (in SGM) in eight different maps. Thus, we may have a point being consistently estimated with a constant disparity value across eight of the maps shown in Figure 3.5. Figure 3.6 shows examples of consistency maps checking for consistencies of $c \ \epsilon \{4, 5, 6, 7\}$. The bottom row shows the results after removing points that are inliers to the estimated ground plane model, points with very small disparities (i.e., distant points), and very small objects in the consistency maps, because small irregularly-shaped objects are assumed to not be objects of significance. In the raw consistency maps (Figure 3.6, top row) that the background is consistently detected, but for extracting the object, we filter out small disparities, so we can focus on relevant points.

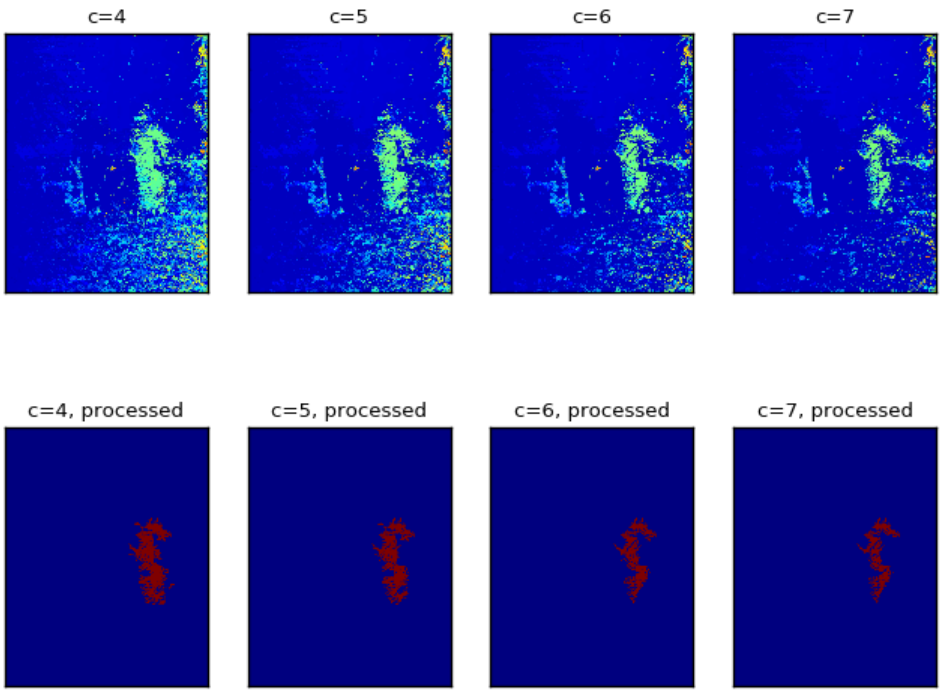


Figure 3.6: Consistency map examples
Generated consistency maps requiring consistently computed disparities across 4, 5, 6, and 7 disparity maps. Top row: raw consistency maps; Bottom row: after filtering out ground-plane inliers, small disparities (distant points), and small objects.

While a disparity map derived using SGM is itself is already a form of a confidence map for disparity estimates, the nature of our set-up (orthogonally diverging fisheye cameras) requires additional steps to filter out inaccurate and noisy estimates. Our goal is to obtain a convex hull around any objects of interest in the scene, so after computing a consistency map and filtering out the background, ground-plane, and small objects, we compute the histogram to analyze the remaining disparities. The histogram allows for binning the

dominant disparities and drawing hulls around the corresponding connected components that are described by these disparity values. (Viewing a histogram of the consistent distortion estimates can also be an easy way to understand object depths in the scene.)

As discussed in Chapter 2, objects cannot be realistically mapped if only ground-plane calibration data is used for computing mapping locations. To effectively obtain more perceptually realistic mappings of off-the-ground points, our proposed method that uses the virtual camera calibration parameters requires accurate disparity estimates. Our process of computing and processing consistency maps indicates regions of the scene for which we have high-confidence disparity estimates, on which virtual camera calibration parameters can be used.

Figure 3.7 shows examples of the convex hulls computed for the detected objects of interest in the scene. In Section 3.5, we will show how these regions can be processed with our virtual camera calibration parameters (Chapter 2) to be more accurately represented in a top-view image.

Figure 3.7: Convex hull examples
Two examples of pairs of rectified overlapping-FOV windows. In each, the convex hull is drawn in the reference image around the detected object of interest.

3.4 Background extraction

The third and final part of the image we wish to segment is the background, and while it may not appear in the top-view image, we still present a simple method for background extraction. Any points with small disparity estimates will be considered part of the background, and again, we use the consistency maps for locating high-confidence background pixels. Our proposed method for background extraction is to take the consistency map, remove ground-plane inliers and disparities greater than a threshold d_{obj}, and take the largest connected component as the background. Small disparities can generally be robustly estimated simply by using SGM and the appropriate disparity search range (as seen in the last row in Figure 3.5). However, checking for consistency across disparity maps computed with SGM helps to further eliminate outliers and generate a high-confidence estimate of the background pixels.

Figure 3.8 shows stereo examples (first two columns) from which the background region (third column) has been estimated using the proposed method. The last column shows the result after the image texture has been painted onto the estimated background region for visualization purposes. For these examples, the disparity threshold for objects of interest is $d_{obj} = 100$, and $c = 5$ for consistency computations, which was selected because it helps retain high-confidence estimates without being too stringent.

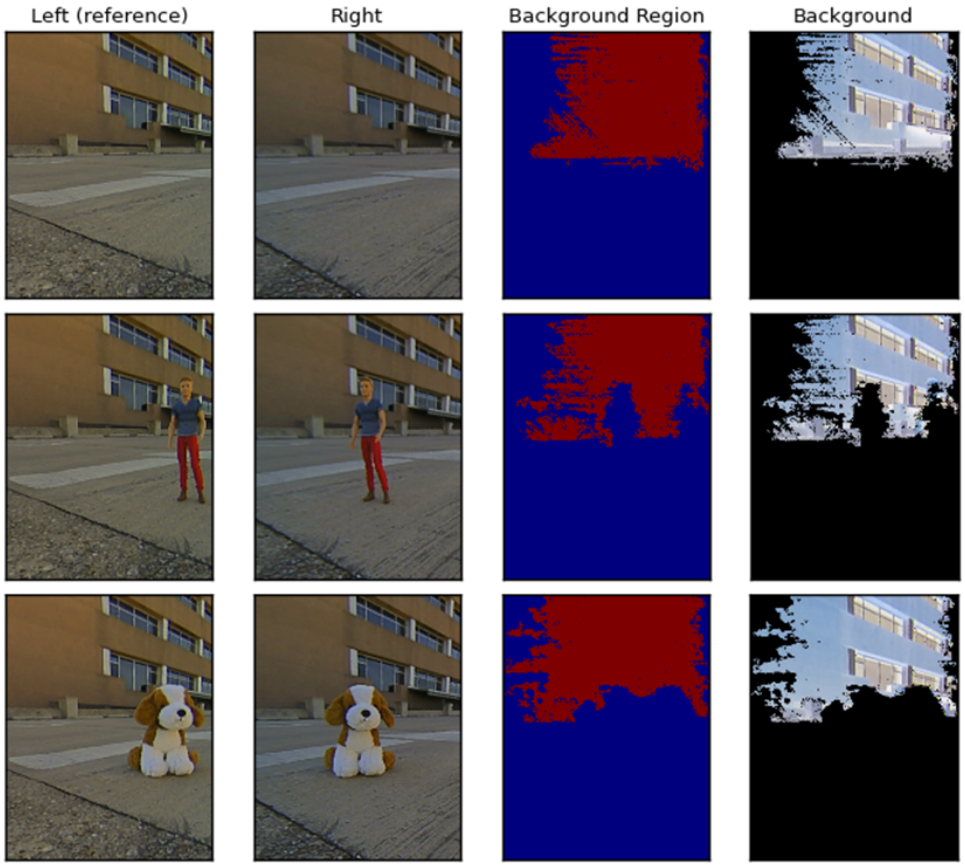


Figure 3.8: Background-extraction examples

Methods for segmenting the scene into its ground plane, objects of interest, and the background were presented here; however, moving forward, we will focus on the detected and extracted objects for applying virtual camera calibration parameters in order to project the objects into the top-view image.

3.5 Object mapping under calibrated virtual camera

The calibration method presented in Chapter 2 relies on accurate and dense disparity estimates to be able to compute mappings for off-the-ground points. Not only are such disparity maps very difficult to generate, applying pixel-wise computations to surfaces that can be approximated as planar is unnecessary and inefficient. The problem with top-view images generated using only ground-plane calibrated look-up tables is that objects in the top-view look unnaturally stretched out. Even just reconstructing an object's image, however rough, to have the correct projected height, would be an improvement on the current output.

To combine ideas presented in this chapter with the proposed method from Chapter 2, we use the the virtual camera calibration parameters to compute new mappings of extracted objects of interest. However, we will treat the area enclosed by an object's computed convex hull as a planar surface with a constant disparity, because our disparity maps do not have the smoothness or completeness for us to compute the mappings in a pixel-wise manner. Even with approximating the disparity of an object to be constant, we are still able to demonstrate that the new virtual camera calibration parameters can be used to project the object into the top-view image to be more realistically represented.

Figure 3.9 shows the two examples from Figure 3.7 after the object within the computed (and shown) convex hull is re-mapped to its realistic virtual camera location in the top-view image. The top-view images generated

using the ground-plane-calibrated look-up table are used as the base for the new top-view image. Only the projection of the detected object described by the hull is computed using the virtual camera calibration parameters, and this newly-computed projection is overlayed on the base image.

To apply the virtual camera calibration parameters, each detected object is assumed to have a constant disparity. At this point, we cannot rely on individual pixel disparities, which do not exist for all points within the hull. Recall that we locate the object and compute the hull using only the highest-confidence disparity estimates, with the assumption that an object will have enough discernible features so that SGM will detect a sufficient number of high-confidence, accurate disparity estimates to approximate a hull for the object. Holes are sure to remain, which is why a convex hull is used to group together all these estimates. For each example, we found that the median disparity value among the high-confidence estimates within the hull was a robust approximation for the disparity of the entire object. The results show that even when assuming a constant disparity across the area within the object's hull, the newly calibrated virtual view is still able to capture an object's 3D location while representing its height more realistically.

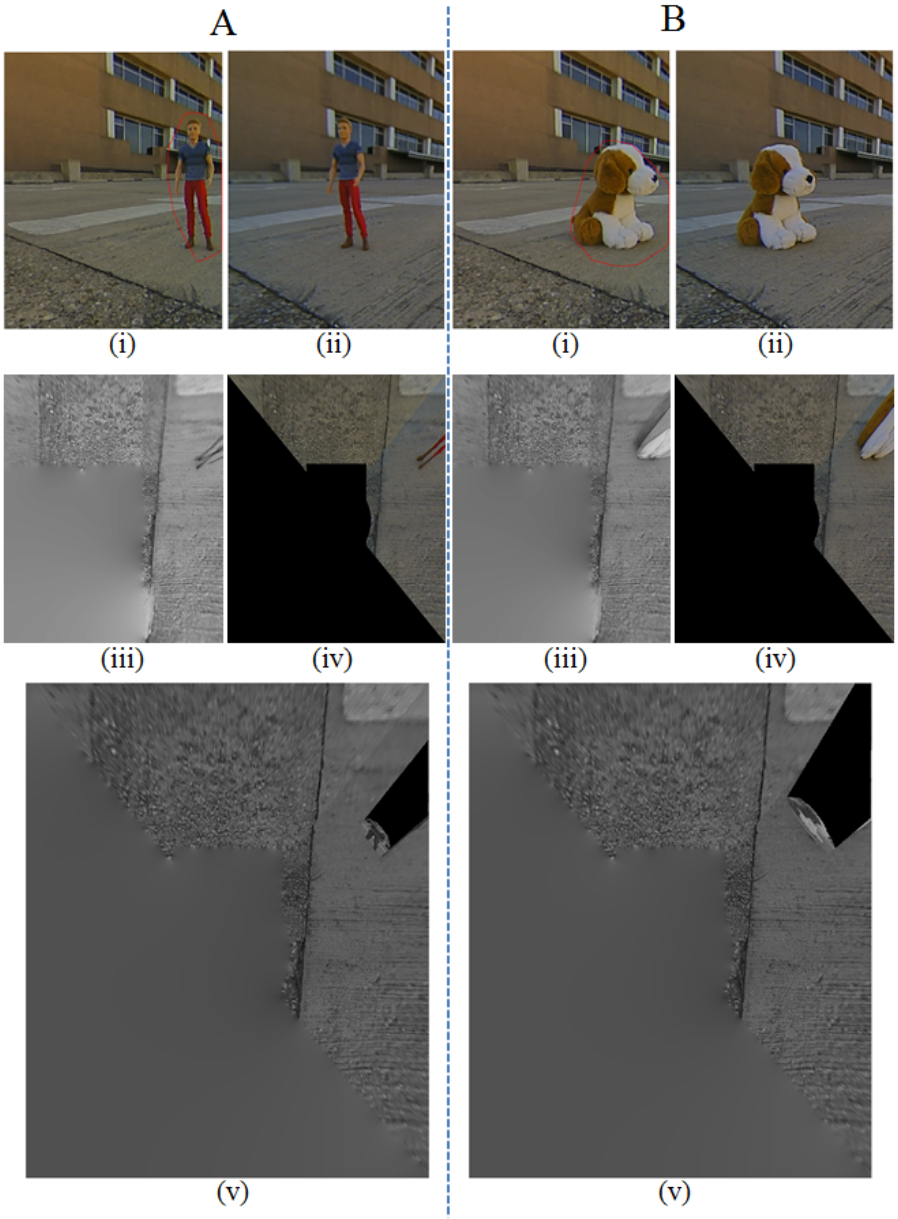


Figure 3.9: Projection of objects in the virtual camera view

The two stereo examples from Figure 3.7. For each example (A and B): (i) Left (reference) image window with the hull drawn, (ii) Right rectified window, (iii) Output top-view image generated using only ground-plane calibration and alpha-blending the overlapping region, (iv) Output image generated using only ground-plane calibration and image partitioning/stitching along a seam in the overlapping region, (v) output top-view image after re-mapping the object using virtual camera calibration parameters and blacking out pixels behind the object.

Chapter 4

Conclusion

In Advanced Driver Assistance Systems, displaying a birdseye view of the vehicle's immediate surroundings can be very beneficial to the driver by providing them increased awareness of any obstacles that may be in their blind spots. Generating these views in which objects are realistically represented is not straight-forward. Simple calibration methods involving patterns placed on the ground around the vehicle are insufficient for obtaining knowledge about the scene beyond ground-plane mappings. Thus, any objects, or any points not coplanar with the ground, will be distorted in the top-view mapping and look unrealistic to the driver.

Our imaging system has four fisheye cameras placed with orthogonally diverging camera axes and has a wide baseline between each pair of adjacent cameras. While this set-up allows us to easily capture the entire 360^o surroundings, the cameras have inherent spatial distortion, and the stereo correspondence problem posed by the geometry is very difficult to solve. While much work has been done in improving stereo matching methods, few have dealt with wide-baseline applications; even fewer have used fisheye lenses with diverging camera axes; and no work has been done with stereo vision with

orthogonally diverging axes. Additionally, most work in wide-baseline applications involves region-based or feature-based matching to handle the potentially widely-disparate views of objects. We, however, want dense matches and therefore used SGM, a popular stereo correspondence for parallel rectilinear stereo systems, and applied it strategically.

In this report, we presented a new method for obtaining necessary parameters for determining more realistic mappings for objects in the top-view image. We also presented a simple method for segmenting the scene in the overlapping FOV between a stereo camera pair into the ground plane, objects of interest, and the background by strategically defining disparity bands over which to apply SGM. For robust estimation of the parameters of the ground plane, we found RANSAC to work well. To detect the object and background, we introduced a consistency check for finding high-confidence estimates over an array of disparity maps, tuning the SGM algorithm for each one to search over a different range of disparities. We demonstrated that computing a hull around high-confidence object disparities was reliable in effectively segmenting the object from the scene. Further, we showed that the areas defined by these hulls can undergo our proposed virtual camera mapping to give output top-view images in which the proportions of the object are more realistically represented.

A limitation of the method that uses the virtual camera calibration parameters is that it depends on dense and accurate disparity estimates. Using SGM to compute disparity maps for images captured with our set-up is

very difficult, but we were able to obtain enough high-confidence estimates to approximate reliable disparities of object surfaces. Future work may involve improving object extraction for drawing better hulls, computing denser and more accurate disparity estimates over the span of all disparities in the scene, or developing the virtual camera calibration to rely less on the accuracy or density of disparity maps.

Bibliography

[1] "Traffic safety facts: Crash stats," *U.S. Department of Transportation, Nat'l Highway Traffic Safety Admin.*, Feb 2015.

[2] Y.-C. Liu, K.-Y. Lin, and Y.-S. Chen, *Robot Vision: Second International Workshop, RobVis 2008, Auckland, New Zealand, February 18-20, 2008. Proceedings*, ch. Bird's-Eye View Vision System for Vehicle Surrounding Monitoring, pp. 207–218. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[3] Y. Ishii, K. Asari, H. Hongo, and H. Kano, "A practical calibration method for top view image generation," in *Consumer Electronics, 2008. ICCE 2008. Digest of Technical Papers. International Conference on*, pp. 1–2, Jan 2008.

[4] Z. Xiong, J. Ying, and R. Zhang, "Research of bird's-eye panoramic view for vehicle parking," in *Multimedia Technology (ICMT), 2011 International Conference on*, pp. 456–459, July 2011.

[5] S. M. Santhanam, V. Balisavira, S. H. Roh, and V. K. Pandey, "Lens distortion correction and geometrical alignment for around view monitoring system," in *Consumer Electronics (ISCE 2014), The 18th IEEE International Symposium on*, pp. 1–2, June 2014.

[6] J. Pan, V. Appia, and A. Bovik, "Virtual top-view camera calibration for accurate object representation," in *Image Analysis and Interpretation (SSIAI), 2016 IEEE Southwest Symposium on*, March 2016.

[7] MATLAB, *version 8.6.0 (R2015a)*. Natick, Massachusetts: The MathWorks Inc., 2015.

[8] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 807–814, IEEE, 2005.

[9] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 7, pp. 787–800, 2003.

[10] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3, pp. 15–18, IEEE, 2006.

[11] G. Medioni and R. Nevatia, "Segment-based stereo matching," *Computer Vision, Graphics, and Image Processing*, vol. 31, no. 1, pp. 2–18, 1985.

[12] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 328–341, Feb 2008.

[13] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 3, pp. 492–504, 2009.

[14] C. L. Zitnick and T. Kanade, "A cooperative algorithm for stereo matching and occlusion detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 7, pp. 675–684, 2000.

[15] D. Tell and S. Carlsson, "Wide baseline point matching using affine invariants computed from intensity profiles," in *Computer Vision-ECCV 2000*, pp. 814–828, Springer, 2000.

[16] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[17] P. Pritchett and A. Zisserman, "Wide baseline stereo matching," in *Computer Vision, 1998. Sixth International Conference on*, pp. 754–760, IEEE, 1998.

[18] T. Tuytelaars and L. J. Van Gool, "Wide baseline stereo matching based on local, affinely invariant regions.," in *BMVC*, vol. 1, p. 4, 2000.

[19] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated car-

tography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[20] K. Konolige, M. Agrawal, M. R. Blas, R. C. Bolles, B. Gerkey, J. Solà, and A. Sundaresan, "Mapping, navigation, and learning for off-road traversal," *Journal of Field Robotics*, vol. 26, no. 1, pp. 88–113, 2009.

[21] K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, and B. Gerkey, "Outdoor mapping and navigation using stereo vision," in *Experimental Robotics*, pp. 179–190, Springer, 2008.

[22] J. Zhou and B. Li, "Robust ground plane detection with normalized homography in monocular sequences from a robot platform," in *Image Processing, 2006 IEEE International Conference on*, pp. 3017–3020, IEEE, 2006.

[23] F. Mufti, R. Mahony, and J. Heinzmann, "Spatio-temporal ransac for robust estimation of ground plane in video range images for automotive applications," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pp. 1142–1148, IEEE, 2008.

[24] G. Baguley, "Stereo tracking of objects with respect to a ground plane," 2009.

[25] J. Zhao, J. Katupitiya, and J. Ward, "Global correlation based ground plane estimation using v-disparity image," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 529–534, IEEE, 2007.

[26] F. Dai, "py-ransac." https://github.com/falcondai/py-ransac, 2013.