Abstract

Road Pothole Detection System Based on Stereo Vision

Abstract

by

YAQI LI

Abstract

In this thesis, we propose a stereo vision system which detects potholes during driving. The objective is to benefit drivers to react to potholes in advance. This system contains two USB cameras taking photo simultaneously. We use parameters obtained from camera calibration with checkerboard to calculate the disparity map. 2-dimensional image points can be projected to 3-dimensional world points using the disparity map. With all the 3-dimensional points, we use the bi-square weighted robust least-squares approximation for road surface fitting. All points below the road surface model can be detected as pothole region. In case there are more than one pothole on the road, we use the connected component labelling algorithm to label pothole points into different potholes according to the 0 or 1 connection between pixels in binary images. The size and depth of each pothole can be obtained as well. The experiments we conducted show robust detection of potholes in different road and light conditions.

1 Introduction

1.1 Introduction

Potholes are bowl-shaped openings on the road that can be up to 10 inches in depth and are caused by the wear-and-tear and weathering of the road[1]. They emerge when the top layer of the road, the asphalt, has worn away by lorry traffic and exposing the concrete base. Once a pothole is formed, its depth can grow to several inches, with rain water accelerating the process, making one of the top causes of car accidents. Potholes are not only main cause of car accidents, but also can be fatal to motorcycles. Potholes on roads are especially dangerous for drivers when cruising in high speed. At high speed, the driver can hardly see potholes on road surface. Moreover, if the car passes potholes at high speed, the impact may rupture car tires. Even though drivers may see the pothole before they pass it, it is usually too late for drivers to react to the pothole. Any sharp turn or suddenly brake , may cause car rollover or rear-end.

Motivated from above reasons, we decided to investigate a system to detect potholes on roads while driving and the proposed system will produce the 3-dimensional information of potholes and determine the distance from pothole to car for informing the driver in advance.

Currently, the main methods for detecting potholes still rely on public reporting through hotlines or websites, for example, the potholes reporting website in Ohio[2]. However, this reporting usually lacks accurate information of the dimensional and location of potholes. Moreover, this information is usually out of date as well.

A method to detect potholes on road has been reported in a real-time 3D scanning system for pavement distortion inspection[3] which uses high-speed 3D transverse scanning techniques. However, the high-speed 3D transverse scanning equipment is too expensive. Rajeshwari Madli et al. have proposed a cost-effective solution[4] to identify the potholes on roads, and also to measure the depth and height of each pothole using ultrasonic sensors. All the pothole information is stored in database (cloud). Then alerts are provided in the form of a flash messages with an audio beep through android application. To detect the depth of pothole correctly, the ultrasonic sensor should be fixed under the car, which means the car should pass the pothole first.

2D vision-based solutions can detect potholes as well[5]. Regions corresponding to potholes are represented in a matrix of square tiles and the estimated shape of the pothole is determined. However, the 2D vision-based solution can work only under uniform lighting conditions and cannot obtain the exact depth of potholes.

To remove the limitations of the above approaches, we propose a detection method based on computer stereo vision, which provides 3-dimensional measurements. Therefore, the geometric features of potholes can be determined easily based on computer vision techniques. The proposed method requires two cameras to take photos simultaneously. Compared with the expensive high-speed 3D transverse scanning equipment, USB cameras are affordable and flexible. Stereo camera parameters, including intrinsic parameters and extrinsic parameters, are obtained with a checkerboard using Zhang's camera calibration method[6]. Before convert the image coordinates to the world coordinates, some preparation work needs to be done.

Image pairs of road surface should be undistorted and rectified. In undistorted images lens distortion has been removed. Rectification refers to projecting image pairs onto a common image plane, respectively. The rectified and undistorted image pairs are used to calculate the disparity map using the stereo camera parameters obtained before with the semi-global matching algorithm[7] provided in OpenCV (Open Source Computer Vision Library). This algorithm uses a pixelwise, Mutual Information-based matching cost for compensating radiometric differences of the stereo image pairs. The disparity map illustrates the corresponding pixels'difference in a pair of stereo images. Thus, with the disparity map, image points can be transferred to world points. The road surface is fitted using the bi-square weighted robust least-squares algorithm with all the points in world coordinate of the road surface image . Subsequently, all the points below the road surface correspond to the pothole region. In case there are more than one pothole in the region of interest, pothole points are labelled into different potholes according to their connections using the connected component labelling algorithm.

In the reminder of this thesis, chapter 2 briefly discusses the background of this work. Chapter 3 introduces the technical approach and principles that we applied in this pothole detection system. Chapter 4 illustrates the experimental setup of the proposed system. The results of pothole detection system are provided in chapter 5. Finally, chapter 6 concludes the proposed method and discusses the future work which can be done to improve the pothole detection system.

2 Background

2.1 Background

In this thesis, we use computer stereo vision based methods to detect potholes on road surfaces. Stereo vision is an attempt to imitate the eyes of human beings. For the single camera, as shown in Figure 2.1, two different real points P and Q project to a same point in the image plane when they are located in the same line with the optical center. However, when it comes to stereo vision, as illustrated in Figure 2.2, we are able to obtain the depth by means of triangulation, if we can find the corresponding pixels in the stereo image pairs. As shown in Figure 2.3, the B is the distance between the 2 cameras' optical

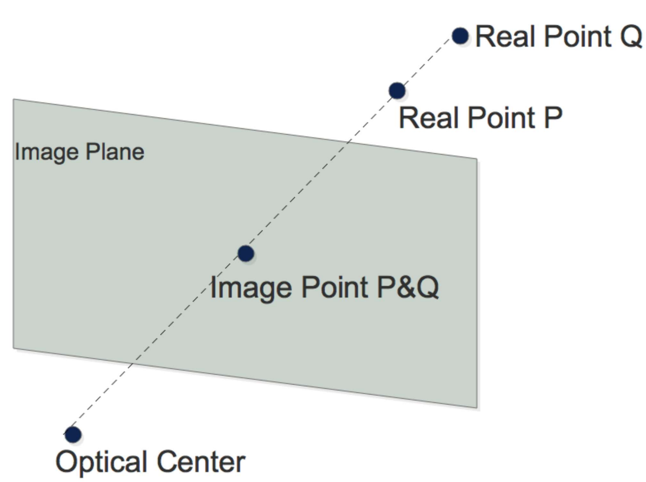

Figure 2.1. Single Camera

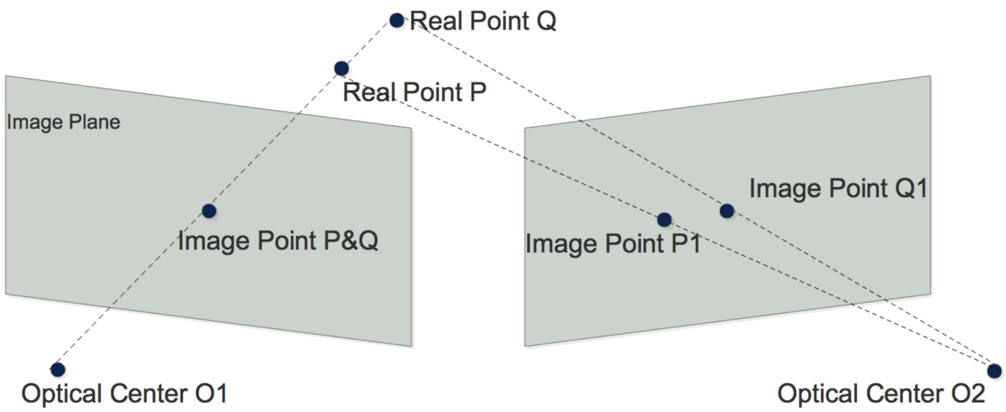

Figure 2.2. Stereo Camera

centers. f is the focal length obtained from stereo camera calibration. XLR and X-OL-OR are equivalent triangles. We can write their equivalent equation as following:

$$\frac{B}{z} = \frac{B + XR - XL}{z - f}. \tag{2.1}$$

Therefore, we can calculate using following equation:

$$Z = \frac{B * f}{XL - XR} = \frac{B * f}{d}. \tag{2.2}$$

Z is the depth of point X, and it is inversely proportional to the disparity. So once we find the corresponding points in the stereo image pairs, we can calculate the disparity and the depth of a real point on roads correctly.

The main difficulty is how to find the best corresponding points in the stereo image pairs. For better matching results, rectification needs to be done first. Rectification includes removing the lens distortion and turning the stereo image pairs in standard form.

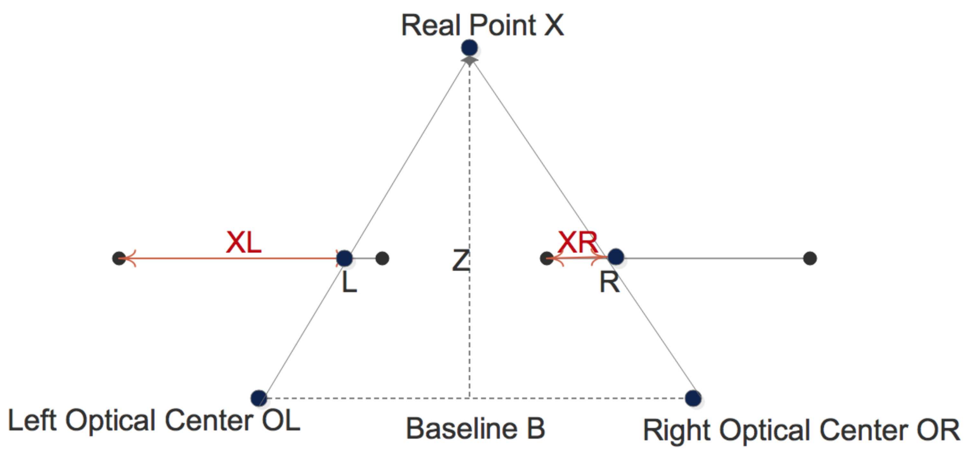

Figure 2.3. Depth Calculation

In this thesis, we use the semi-global Matching algorithm proposed by Heiko[7] which is provided in OpenCV[8]. OpenCV[8] is the open source software library for computer vision. It is a library of programming functions mainly aimed at real-time computer vision and supported by the python. Given the disparity map and stereo camera parameters, the corresponding coordinates in 3 dimensional coordinate system can be calculated. Given all 3D points in an image, a road surface can be fitted using the bi-squares weighted robust least-squares algorithm[9]. Then all the outliers can be labelled as road potholes. However, we need to distinguish different road potholes in one region of interest. We use the connected component labelling algorithm[10] to label different road potholes into different numbers.

3 Approach to Pothole Detection System

In this chapter, we introduce the proposed road pothole detection system. The proposed system consists of 2 modules: Off-line processing and on-line processing. The off-line flowchart of proposed system is illustrated in Figure 3.1.Stereo camera parameters, including intrinsic parameters and extrinsic parameters, are obtained using a checkerboard based on Zhang's camera calibration method[6]. The on-line flowchart of proposed system is shown in Figure 3.2. Before transferring the image coordinates to the world coordinates, some preparation work needs to be done. The image pairs taking by 2 cameras of road surface should be undistorted and rectified, which transforms images to compensate for lens distortion and project image pairs on to a common image plane respectively. The rectified and undistorted image pairs are used to calculate the disparity map with the stereo camera parameters obtained earlier using the semi-global matching algorithm[7] provided, in OpenCV[8].

With the disparity map, image points can be transferred to world points. The road surface is fitted using the bi-square weighted robust least-squares algorithm[9] with all the 3-dimensional points of road surface. Subsequently, all the points below the road surface correspond to the pothole region. In case there are more than one potholes in

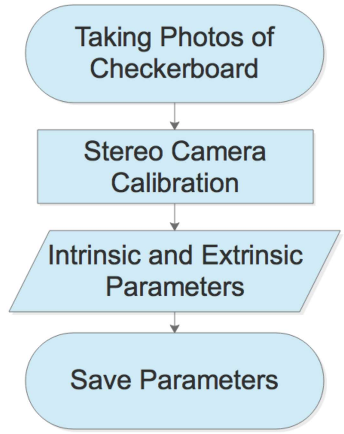

Figure 3.1. Off-line Flowchart of the Pothole Detection System

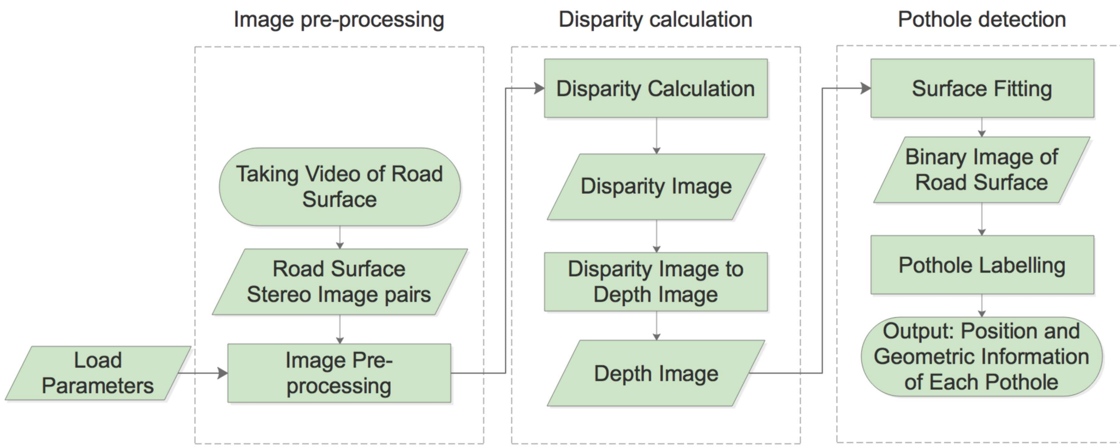

Figure 3.2. On-line Flowchart of the Pothole Detection System

the region of interest, pothole points are labelled into different potholes according to their connections using the connected component labelling algorithm[10].

In the reminder of this chapter, section 1 introduces the stereo camera calibration. Section 2 introduces the stereo processing. The method to convert the disparity map to depth image is introduced in section 3. Section 4 introduces the bi-squares weighted robust least-squares algorithm[9] which is used to fit the road surface. Section 5 introduce the connected component labelling algorithm[10] which is used in the pothole labelling process.

3.1 Stereo Camera Calibration

Camera parameters are necessary for disparity calculation. We can obtain camera parameters, both intrinsic and extrinsic parameters, by stereo camera calibration. To calibrate stereo cameras, a flexible camera calibration approach proposed by Zhang[6] is used in this system. Compared with other classic camera calibration methods which use expensive equipment, the method proposed is economical and flexible. Only two USB cameras are used to observe a checkerboard in different orientations. The proposed system uses an 8×6 checkerboard with 24.5 mm squares as shown in Figure 3.3 . Either two cameras or the checkerboard can be freely moved. Theoretically, a minimum of 2 orientations are needed for camera calibration. However, 20 orientations are used in this system for better quality[11]. Figures 3.4 and 3.5 illustrate the relative movement between the checkerboard and the stereo camera pair regarding different frame of reference. All checkerboards'orientations regarding the fixed stereo camera pair are shown in Figure 3.4. Relatively, all the stereo camera's orientations regarding fixed checkerboards are shown in Figure 3.5.

Both intrinsic and extrinsic parameters of the stereo camera can be obtained from the camera calibration process. The intrinsic parameters, named camera matrix as well, are independent. Therefore, once the intrinsic parameters are determined, they can be

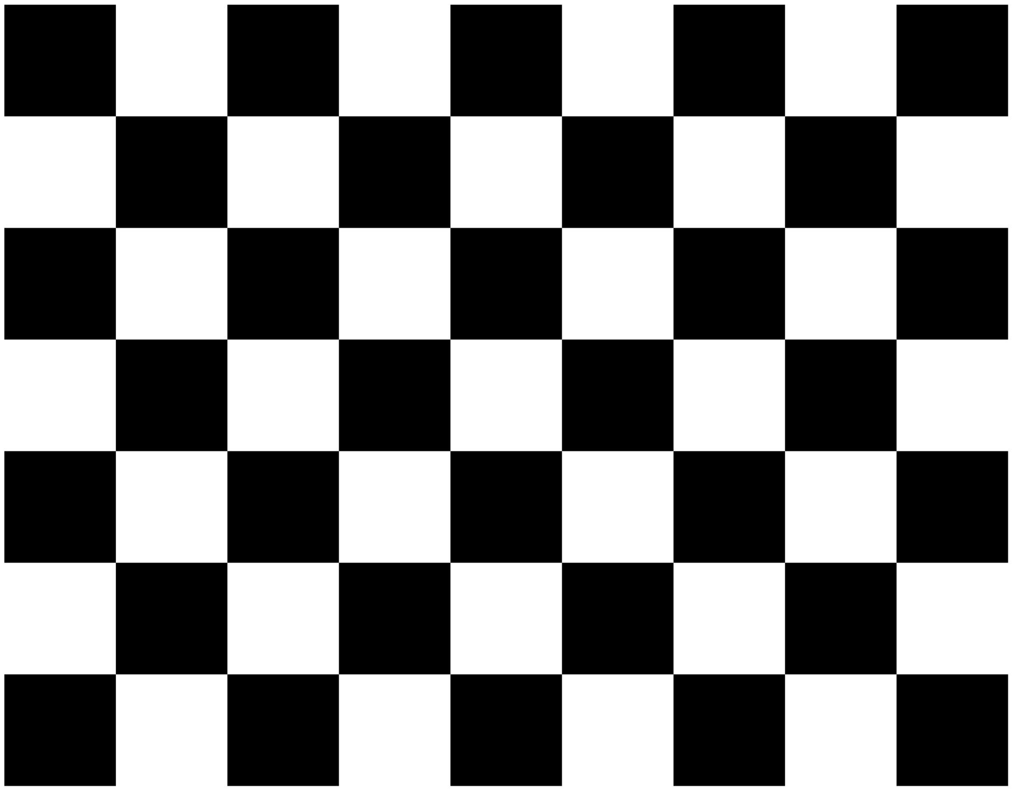

Figure 3.3. 8 x 6 Checkerboards

used as long as the focal length of the camera remain unchanged. The output 3×3 float-point camera matrix is shown as follows:

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.1)$$

f_x, f_y are the focal lengths expressed in pixel units. (c_x, c_y) is the principal point that is usually the center of the image. The joint rotation-translation matrix is called extrinsic matrix. Extrinsic parameters can translate a point (X, Y, Z) to the coordinate system with

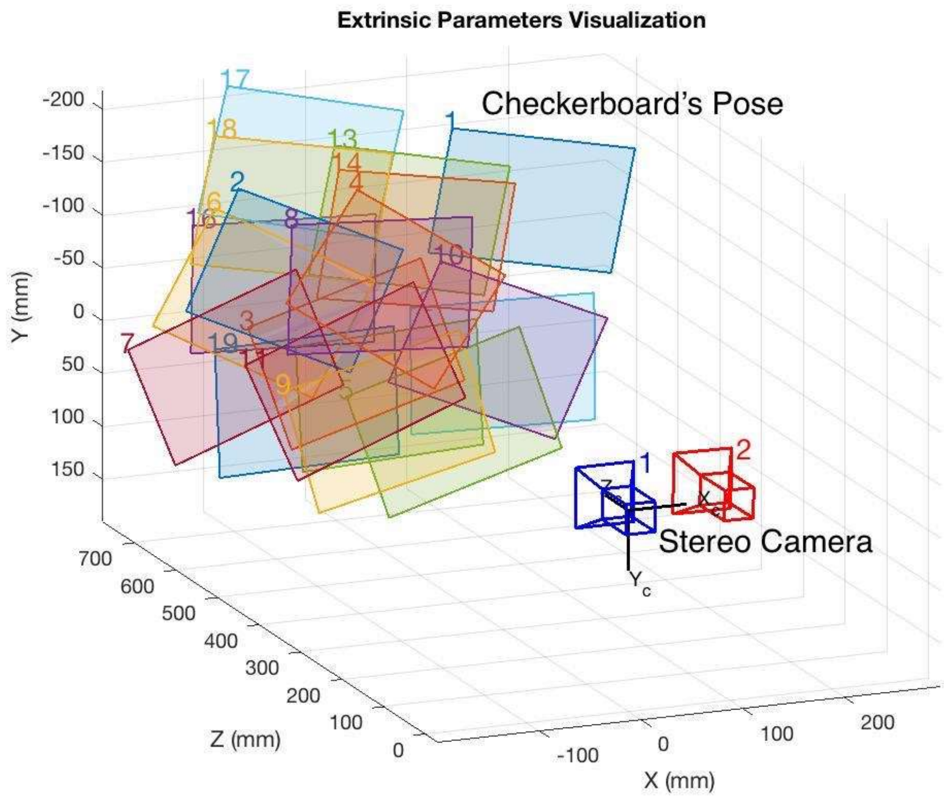

Figure 3.4. Checkerboards'Orientations Regarding the Stereo Camera

fixed camera. The translation is illustrated as following:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t. \tag{3.2}$$

R is the rotation matrix. t is the translation vector. The joint rotation-translation matrix Rt is the extrinsic matrix. With both intrinsic and extrinsic parameters, some preprocessing work including removal of lens distortions and rectification of the stereo image pairs should be done. At this time, light correction can be added to remove the influences cased by lighting condition as well. Therefore, the stereo pairs can be turned

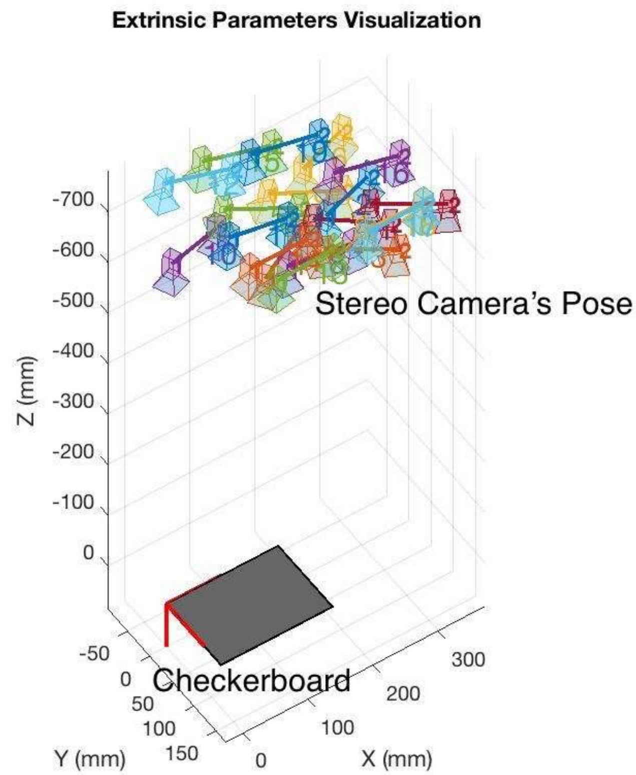

Figure 3.5. Stereo Camera's Orientations Regarding Checkerboards

into standard form which is shown in Figure 3.6. In standard formed image pairs corresponding points locate at same horizontal line. They are better for disparity calculation, because corresponding pixels remain in the same horizontal line would low the matching cost a lot when we calculate the disparity map which will be discussed in detail in the following section.

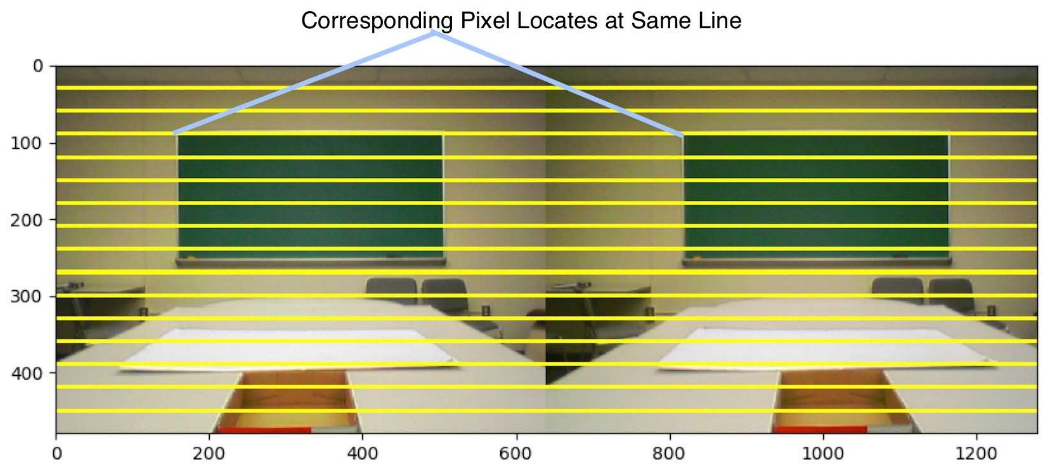

Figure 3.6. Rectified Image pairs in Standard Form

3.2 Stereo Processing

The stereo processing in this system uses the Semiglobal Matching Algorithm proposed by Heiko[7]. This algorithm is employed in four steps: matching cost computation, cost aggregation, disparity computation/optimization, and disparity refinement.

3.2.1 Matching Cost Calculation

The matching cost calculation is based on Mutual Information, which is insensitive to illumination changes[12]. The Mutual Information of the stereo image pairs, MI_{I_1,I_2}, is a measure of the mutual dependence between the two images. It is computed from the entropy H of the stereo image pairs:

$$MI_{I_1,I_2} = H_{I_1} + H_{I_2} - H_{I_1,I_2}. \tag{3.3}$$

In order to obtained the Mutual Information (MI), we need calculate the single entropy and the joint entropy. The entropy of single image is calculated from the probability distributions P of intensities of the image:

$$H_I = -\int_0^1 P_I(i) \log P_I(i) di. \tag{3.4}$$

For the well-rectified images, the joint entropy H_{I_1,I_2} is low. It is calculated from the probability distributions P of intensities of the stereo image pairs.

$$H_{I_1,I_2} = -\int_0^1 \int_0^1 P_{I_1,I_2}(i_1,i_2) \log P_{I_1,I_2}(i_1,i_2) di_1 di_2. \tag{3.5}$$

The joint entropy is transformed into a sum over pixels using Taylor expansion by Kim[13]:

$$H_{I_1,I_2} = \sum_P h_{I_1,I_2}(I_{1P}, I_{2P}). \tag{3.6}$$

The term h_{I_1,I_2} is computed from the joint probability distribution P_{I_1,I_2} of corresponding intensities:

$$h_{I_1,I_2}(i,k) = -\frac{1}{n} \log(P_{I_1,I_2}(i,k) \otimes g(i,k)) \otimes g(i,k). \tag{3.7}$$

Where, n is the number of corresponding pixels, and $\otimes g(i,k)$ indicates a 2D Gaussian convolution. The probability distribution of corresponding intensities is defined with the operator $T[]$. $T[] = 1$ when its argument is true, otherwise $T[] = 0$:

$$P_{I_1,I_2}(i,k) = \frac{1}{n} \sum_P T[(i,k) = (I_{1P}, I_{2P})]. \tag{3.8}$$

Similarly, the single image entropy is calculated as follow:

$$H_I = \sum_P h_I(I_p), \tag{3.9}$$

$$h_I(i) = -\frac{1}{n} \log(P_I(i) \otimes g(i)) \otimes g(i). \tag{3.10}$$

Where P is the intensity distribution, n is the number of corresponding pixels, and $\otimes g(i)$ indicates Gaussian convolution. In summary, we can calculate the matching cost basing on Mutual Information using above equations.

3.2.2 Cost Aggregation

The Semiglobal Matching Algorithm[7] defines the energy $E(D)$ that depends on the disparity image D also known as disparity map:

$$E(D) = \sum_{P}(C(p, D_P) + \sum_{q \in N_P} P_1 T[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1]). \tag{3.11}$$

$|D_p - D_q|$ is the difference of disparities at pixel P and pixel Q respectively. The first term is the sum of all pixel matching costs. The second term adds a constant penalty P_1 for all pixels q in the neighborhood of p. The third term adds a larger constant penalty P_2 for all larger disparity changes. This process is cost aggregation. The cost is aggregated into a cost volume by going into 8 directions through all pixels in the image[14]. The cost in the direction r of the pixel p at disparity d is calculated as:

$$\begin{aligned} L'_r(p,d) = C(p,d) + \min(L'_r(p-r,d), \\ L'_r(p-r,d-1) + P_1, \\ L'_r(p-r,d+1) + P_1, \\ \min_i L'_r(p-r,i) + P_2). \end{aligned} \tag{3.12}$$

The value of $L'_r(p,d)$ always increase along the path. To avoid overflow, the minimum path cost is subtracted from equation 3.12. Equation 3.12 can be modified as follow:

$$\begin{aligned} L_r(p,d) = C(p,d) + \min(L_r(p-r,d), \\ L_r(p-r,d-1) + P_1, \\ L_r(p-r,d+1) + P_1, \\ \min_i L_r(p-r,i) + P_2) - \min_k L_r(p-r,k). \end{aligned} \tag{3.13}$$

Therefore, the final cost of pixel p at disparity d is calculated as the sum of all costs from all directions as follow:

$$S(p,d) = \sum_{r} L_r(p,d). \tag{3.14}$$

3.2.3 Disparity Calculation

Given the cost we calculated in equation 3.14, all the pixels in an image that corresponds to the minimum cost contribute to the disparity image. For the stereo image pair, each disparity of D_{rq} (the disparity at pixel q in right image) is compared to its corresponding disparity of D_{lp} (the disparity at pixel p in left image). The disparity at D_{lp} is set to be invalid (D_{inv}) if both differ:

$$D_p = \begin{cases} D_{lp} & \text{if } |D_{lp} - D_{rq}| \leq 1 \\ D_{inv} & \text{otherwise} \end{cases} \tag{3.15}$$

3.2.4 Disparity Refinement

The disparity map we obtained above might still contain kinds of errors. We need to recover the invalid values. The disparity map might contain outliers caused by noise, reflections and so forth. There are several approaches aimed at improving the raw disparity map including intensity consistent disparity selection and discontinuity preserving interpolation[7]. The advantages of these methods are that they are independent of the used stereo matching method. Besides these approaches, median filtering is useful to remove the remaining irregularities and additionally soothe the disparity map.

3.3 Disparity Image Reprojection

Having got the disparity map, we can use triangulation method to reproject the disparity image to 3D space. For the stereo cameras, given the disparity image and the camera parameters like camera's focal length, we can calculate the 3 dimensional coordinates in real world.

As shown in Figure 3.7, the optical center of camera L is the origin. f is the focal length of the camera. Triangle OAE and triangle OMP are similar triangles, therefore we can write:

$$\frac{z}{f} = \frac{x}{XL}. \tag{3.16}$$

For the right camera, triangle OCD and triangle ONP are similar triangles:

$$\frac{z}{f} = \frac{x - B}{XR}. \tag{3.17}$$

Similarly, along the Y- axis, we can obtain:

$$\frac{z}{f} = \frac{y}{YL} = \frac{y}{YR}. \tag{3.18}$$

The 3 dimensional points as a function of the disparity can be derived from above equations and equation2.2:

$$x = \frac{B * XL}{d}, \tag{3.19}$$

$$y = \frac{B * YL}{d}. \tag{3.20}$$

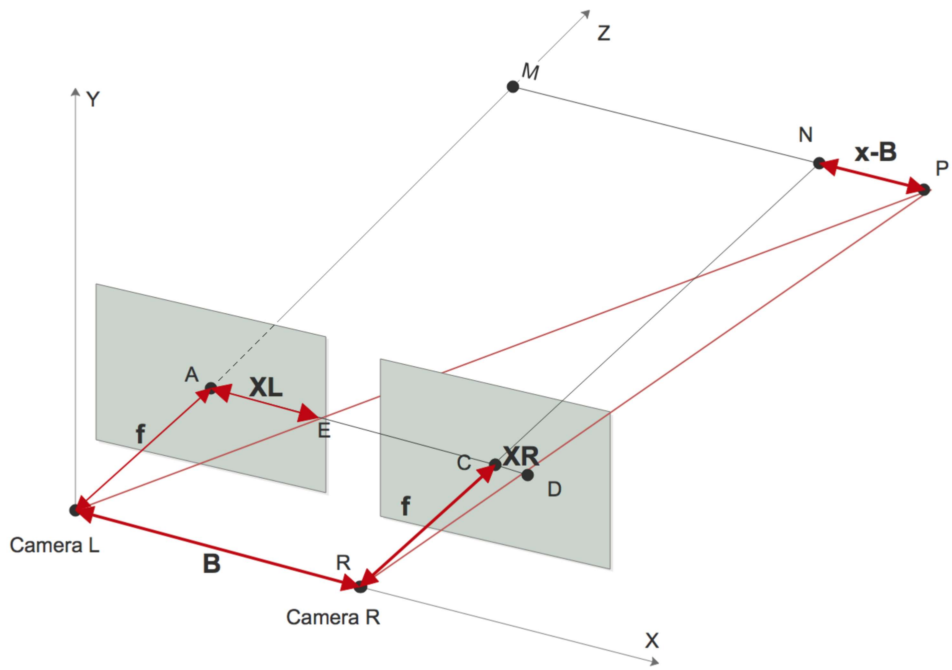

Figure 3.7. Triangulation Method to Reproject Disparity Image

3.4 Road Surface Fitting

With all the 3 dimentional world coordinates points, we can fit the road surface using bi-square weighted robust least-squares method[15 16]. All the points are usually regard as equal quality when fit to a road surface using the least square method[17]. Those pothole points that below the road surface or those noise points might influence the accuracy of the fitted road surface. The bi-square weighted robust least-squares method[15] used in this road pothole detection system minimum the outliers'influences during the fitting processes by add an additional scale factor (the weight).

At first, we usually model the road surface into a quadratic surface as shown in equation 3.21. At each point, we calculate the coefficients $a_1, \ldots, a_6$ to minimize the difference between the modeled road surface and the actual road surface. The procedure of finding the best coefficients are illustrated as follow:

(1) Fit the road surface model with bi-square weighted robust least-squares:

$$y = a_1 + a_2 x + a_3 z + a_4 x^2 + a_5 xz + a_6 z^2 \tag{3.21}$$

(2) Minimize the residuals $r_i = (y_i - \hat{y})^2$ by differentiating the sum with respect to the coefficients. The resulting equation in matrix form is illustrated as follow:

$$\begin{bmatrix} n & S_x & S_z & S_{x^2} & S_{xz} & S_{y^2} \\ S_x & S_{x^2} & S_{xz} & S_{x^3} & S_{x^2z} & S_{xz^2} \\ S_z & S_{xz} & S_{z^2} & S_{x^2z} & S_{xz^2} & S_{z^3} \\ S_{x^2} & S_{x^3} & S_{x^2z} & S_{x^4} & S_{x^3z^2} & S_{x^3z} \\ S_{xz} & S_{x^2z} & S_{xz^2} & S_{x^3z} & S_{x^2z^2} & S_{xz^3} \\ S_{z^2} & S_{xz^2} & S_{z^3} & S_{x^2z^2} & S_{xz^3} & S_{z^4} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} S_y \\ S_{xy} \\ S_{zy} \\ S_{x^2y} \\ S_{xzy} \\ S_{z^2y} \end{bmatrix}, \tag{3.22}$$

where, $S_{xz} = \sum_{i=1}^{n} \omega_i x_i z_i$

(3) Compute the adjusted residuals as following:

$$r_{adj} = \frac{r_i}{\sqrt{1 - h_i}}, \tag{3.23}$$

h_i are leverages that adjust the residuals by reducing the weight of high-leverage data. Then the standardized adjusted residuals are calculated as following:

$$u = \frac{r_{adj}}{Ks}, \tag{3.24}$$

K is the tuning constant equal to 4.685. s is the robust variance given by the median absolute deviation of the residual (MAD)/0.6745.

(4) Compute the bi-square weights as the function of u_i for the i_{th}points as following:

$$w_i = \begin{cases} (1 - (u_i)^2)^2 & |u_i| < 1 \\ 0 & |u_i| \geq 1 \end{cases} \tag{3.25}$$

(5) If the fit converges, then we are done. Otherwise, calculate the next iteration of the fitting processes from the first step until it converges.

3.5 Road Pothole Labelling

We cannot simply regard one black region as a pothole. Because, the detected pothole region is not connected. Therefore, we need to label the detected pothole region using the connected component labelling algorithm[10]. Or in case there are more than one road pothole in the region of interest, we numbering these road potholes with the connected component labelling algorithm[10]. Taking Figure 3.8 as an example, we start off at the top left corner. The first pixels (1,1) is not background (black in this example) and there is not any label around this pixel, so we create a new label named 1(mark as red) for this pixel. Then we move to pixel (1, 2), it is not background and there is a label to its left, so we copy the label of its left pixel. Then next pixel (1, 3) is background, so we skip it. Next comes the pixel (1, 4), the situation of this pixel is same as pixel (1, 1), so we create a new label name 2 (marked as blue). Repeat these steps for each pixel until pixel (3, 4). There are 2 labels around this pixel. We label this pixel as 1 (the smaller number). Then mark that label 2 (the larger number) is a child of label 1 as shown in Figure 3.9. Follow the above rule, we can label the whole image illustrated as Figure 3.10. The flowchart of the first pass of judgement is illustrated in Figure 3.11.

Then we start the second pass of labelling the image. We start at the top left corner as well and check whether its label is a child of any other label. If it is a child label of a smaller label, replace the child label with the smaller label as shown in Figure 3.12. Repeat these judgements for each pixel in the image. Finally, the labelling result is shown in Figure 3.13. The flowchart of the second pass of judgement is illustrated in Figure

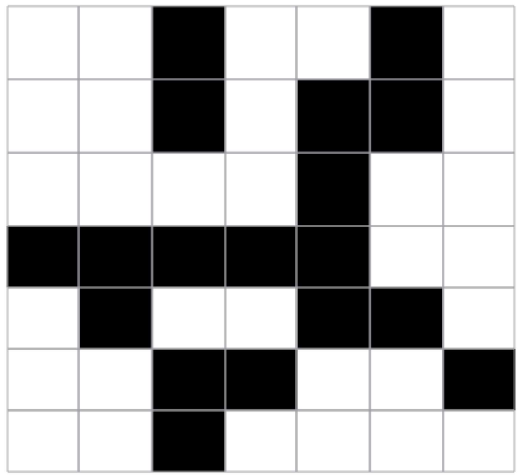

Figure 3.8. Example Binary Image

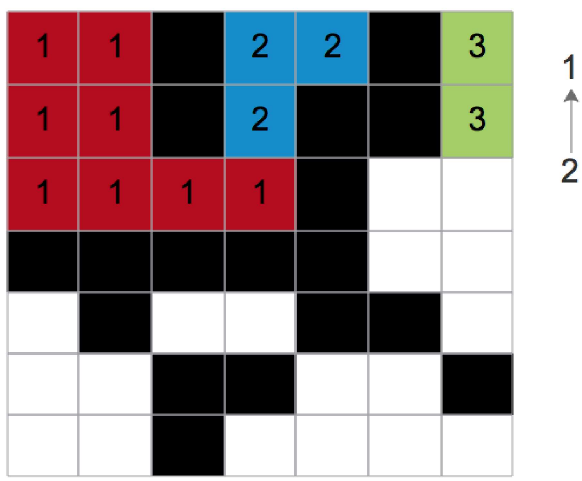

Figure 3.9. 2 Labels Around 1 Pixels

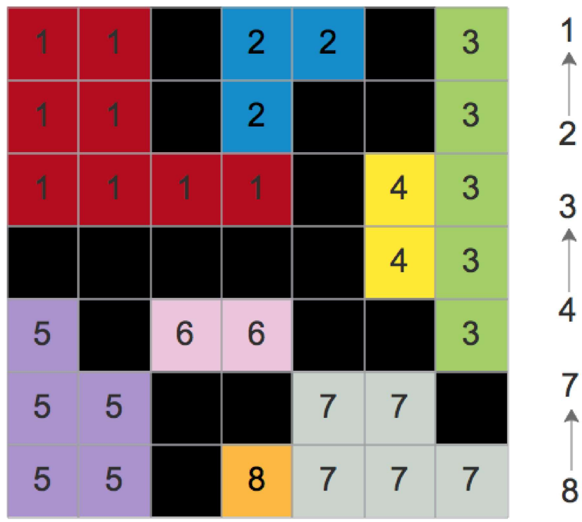

Figure 3.10. Labels After the First Pass

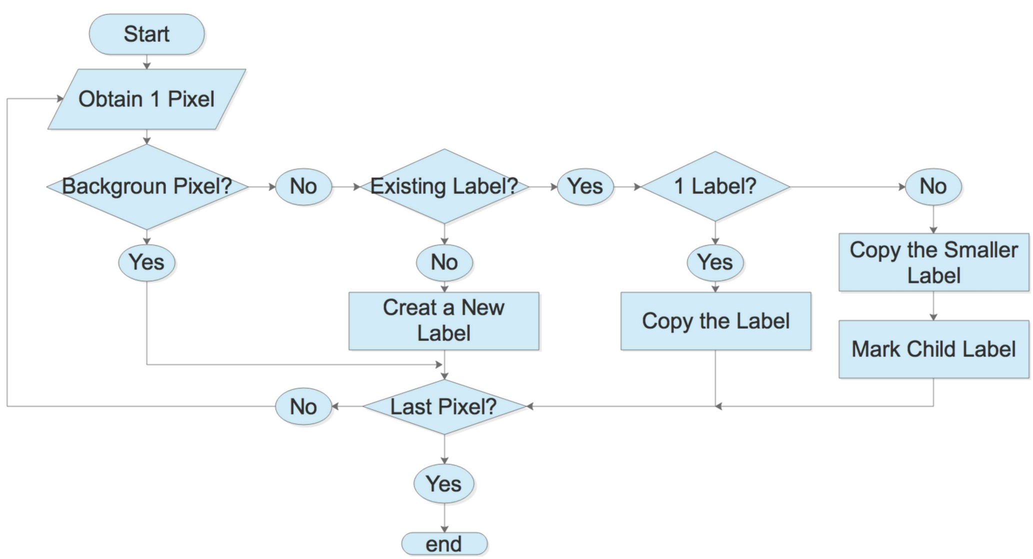

Figure 3.11. The First Pass Flowchart

3.14.

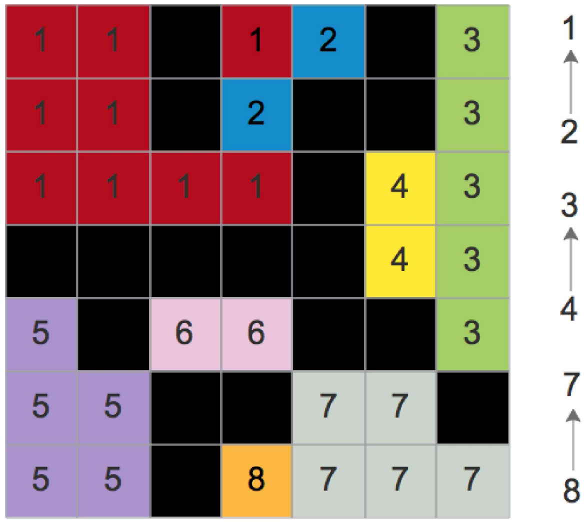

Figure 3.12. The Second Pass

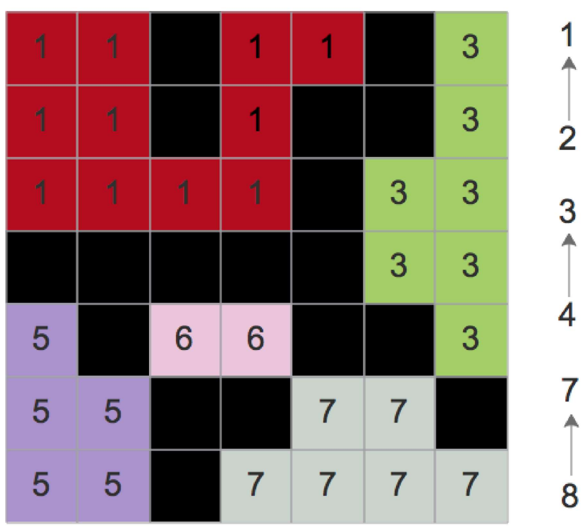

Figure 3.13. The Labelling Result

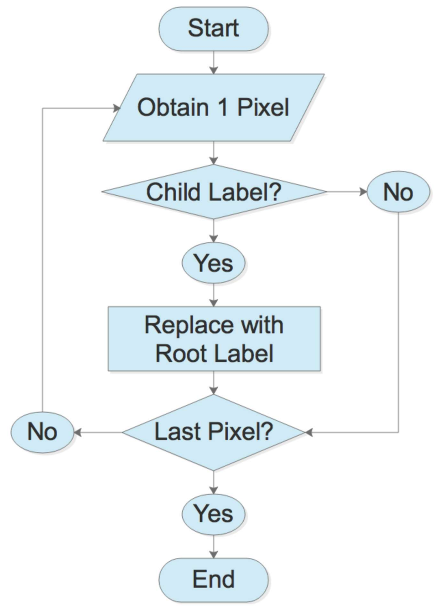

Figure 3.14. The Second Pass Flowchart

With the labelled potholes, we can easily calculate the distance from pothole to the car or the other geometric information of each pothole in the region of interests using the image in world coordinates.