

3D Reconstruction by a Moving Monocular Multi-View Camera System

by

Qiqin Dai

in Partial Fulfillment of the Requirements for the Degree

of

Bachelor of Engineering

at the Department of Control Science and Engineering

Zhejiang University

June 1, 2012

Thesis Supervisor:

Prof. Masatoshi Okutomi

This thesis was done when the author was with
Okutomi & Tanaka Laboratory
at the Department of Mechanical and Control Engineering
Tokyo Institute of Technology

Abstract

Stereo matching and 3D reconstruction has aroused much attention in the computer vision community, due to its potential application in real object modeling and human computer interaction. In this thesis, we systemically accomplished the stereo vision with the monocular multi-view camera system. A complete calibration procedure is proposed and obtains accurate precession. In the linear part of calibration, we show how the process in mean rotation outperforms the baseline method in robustness. By nonlinear refinement based on Bundle-Adjustment, the reprojection error is minimized by the iteration techniques. A Matlab toolbox, implementing the whole calibration method, is presented to make the calibration process convenient and automatic. With an accurate calibration result, we realized the multi-baseline stereo in disparity space, which is rapid in speed and free from distortion. Sufficient experiments on both synthetic and real object experiment demonstrate the validation of calibration. The depth map generated in disparity space verified the efficiency of this approach. Finally, with the ICP algorithm, the 3D reconstruction is accomplished by aligning those depth maps captured from a moving camera.

Key words

monocular multi-view camera system; calibration; stereo matching; disparity space, ICP algorithm

Acknowledgements

First of all, I have to give my sincere appreciation to my supervisor Prof. Masatoshi Okutomi for his enlightening guidance and his kind offering of my research. I also conceive my gratitude to Dr. Akihiko Torii for his meticulous advice and valuable time in discussing with me. My special gratitude should be directed to the team of Young Scientist Exchange Program (YSEP) for providing me an opportunity to finish my diploma thesis in Tokyo Institute of Technology.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

_____ May 20, 2012

Qiqin Dai

Contents

1. Introduction	1
1.1 Related Research	2
1.1.1 Camera Calibration	2
1.1.2 Stereo Vision.....	3
1.2 Overview of Our Work.....	4
2. Preliminaries and Notations	6
2.1 A Parametric Camera Model for Omnidirectional Camera	6
2.2 System Configuration	9
2.3 Geometry of Catadioptric Stereo.....	13
2.3.1 Position and Orientation of the Mirrored Cameras	13
2.3.2 Restrict 6 DOF to 3 DOF	15
3. Calibration of the Monocular Multi-view Camera System.....	17
3.1 The Formulation of Calibration.....	17
3.2 The Solution of Calibration Model	20
3.2.1 Solving for camera extrinsic parameters.....	20
3.2.2 Solving for camera intrinsic parameters	24
3.3 Iterative Center detection	26
3.4 Non-linear Refinement by Bundle-Adjustment	27
4. Rectification	29
4.1 Perspective Reprojection.....	29
4.2 Epipolar Curve	30

4.3 Comparison of Perspective Reprojection with Epipolar Curve	33
5. Multi-Baseline Stereo.....	36
5.1 General Approach	36
5.1.1 The SSD function	36
5.1.2 Elimination of Ambiguity by SSSD function	38
5.1.3 Implementation in our camera system with epipolar curve	42
5.2 Fast Implementation in Disparity Space	44
5.2.1 The Formation of Disparity Space	45
5.2.2 Multi-Baseline Stereo in Disparity Space	46
5.2.3 Merits of the Disparity Space Approach	50
6. Experimental Results	53
6.1 Calibration Results	53
6.1.1 Simulation Experiment.....	53
6.1.2 Real Experiment.....	56
6.2 Rectification Results	62
6.2.1Perspective Reprojection Result	62
6.2.2 Epipolar Curve Result.....	63
6.3 Multi-Baseline Stereo Results	64
6.3.1 General Approach	64
6.3.2 Fast Implementation Results	67
6.4 3D Reconstruction Results	69
7. Conclusion and Future Work	76
8. Reference	78

Chapter 1

Introduction

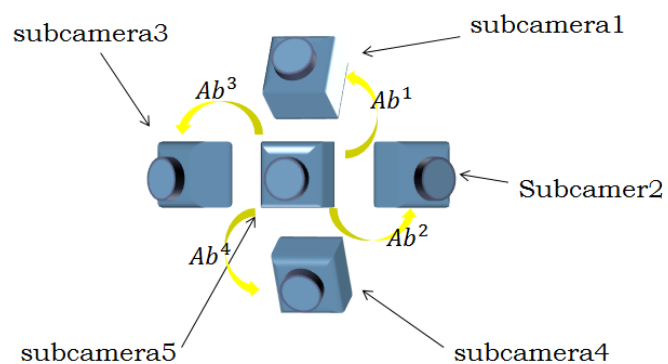
With the rapid development in field of computer vision, measuring the real world with stereo vision is a hot trend with tremendous research. To this end, Jiang, etc., proposed a prototype for monocular multi-view camera system in 2009 [1], which is composed of one fisheye camera with four mirrors around it. (Fig. 1.1 (a)) This camera system is equivalent to five subcameras, with the same intrinsic parameters, at different positions (Fig. 1.1 (b)). A typical image obtained by this camera system is shown in Fig. 1.2. The mirror effect should be taken care of that the left and right images are horizontal symmetrical to the center image, and the up and bottom images are vertical symmetrical to the center image.

This camera system holds such advantages as:

- Identical Intrinsic Parameters with five subcameras;
- Synchronous in data acquisition;
- Possible to get stereo vision with one shot;



(a)



(b)

Fig. 1.1 The Monocular Multi-view camera system

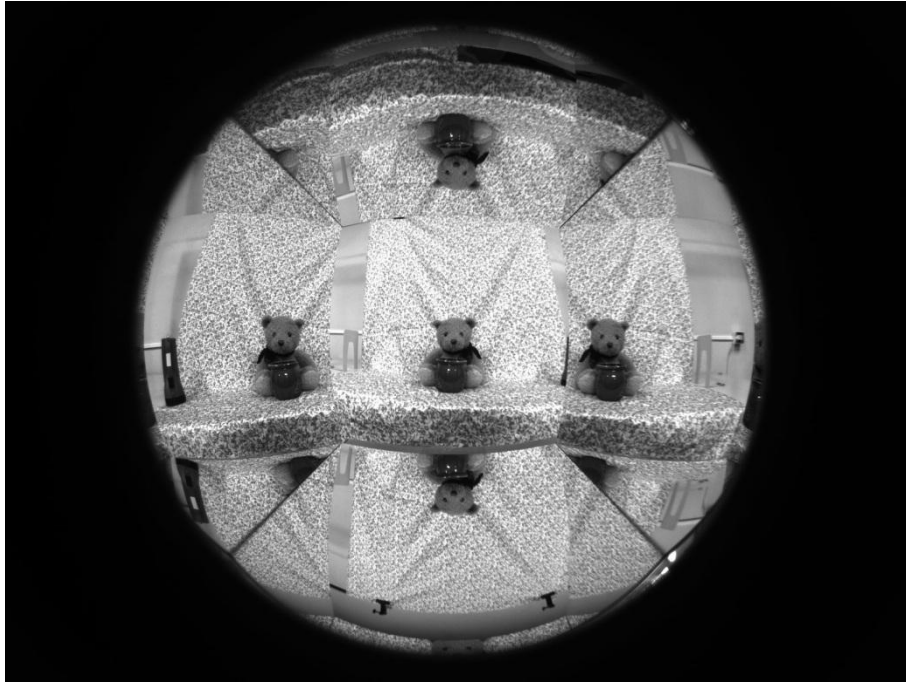


Fig. 1.2 A typical image taken by the camera system

The greatest merit of this camera system in 3D reconstruction is the depth map could be obtained with only one shot, while it is impossible to do the structure from motion (SfM) with one shot. To this end, we choose the approach of computing the depth map from each shot with a moving camera system, then aligning the 3D depth points. We especially focus on depth map generation, and contributions are made in both calibration and stereo matching.

1.1 Related Research

1.1.1 Camera Calibration

When it comes to camera calibration, there are tremendous researches on this topic. In the field of omnidirectional camera calibration, D. Scaramuzza's camera model and calibration methods ([2],[3]) outperform others and have merits of no prior knowledge of the camera is needed and auto center detection. So we intend to use this novel technique in our camera system.

To this monocular multi-view camera system, Jiang, etc. proposed a calibration method [1] based on catadioptric geometry with single shot. However, our experience shows it is not sufficient. Even taking average of the results by many shots sometimes suffer from the trouble of misleading.

There are some researches on calibration of multiple camera system: T. Svoboda [10] proposed a self-calibration method by a moving light spot to calibrate the multiple camera system mounted inside a room. K. Danilidis [12] then extended this method by adding radical distortion into consideration. However, their multiple camera system is set in different places inside a room to monitor the activities inside the room, which is not appropriate to calibrate our camera system.

1.1.2 Stereo Vision

According to the D. Scharstein's taxonomy [8], numerous methods exist in stereo matching. Usually these methods are based on one rectified image pairs with one baseline. Since our camera system is designed for multi-baseline stereo with four baselines, M. Okutomi's multiple baseline stereo [4] would be a proper choice. In detail, the SSSD function are introduced and proved to be capable of eliminating mismatching caused by ambiguity in the input images.

The top performer in Middlebury benchmark [8] by X. Mei, etc. shows how to implement stereo matching in disparity space, which is parallel in natural and could

achieve real time performance in GPU. Their approach enlightens us in multi-baseline stereo could also be implement in disparity space.

1.2 Overview of Our Work

Our work's contribution lies in two aspects: calibration and stereo matching.

In camera calibration, we systematically accomplished the calibration of the monocular multi-view camera system. Our proposed method is composed of linear initialization part and nonlinear refinement based on Bundle-Adjustment.

In detail, the key part in linear initialization is the process of 'mean rotation' procedure, which estimates two sets of rotation: rotation from chessboard to center camera, rotation from center camera to subcamera, simultaneously, which makes the proposed algorithm robust and free from misleading. The computational cost of this linear calibration is so low that D. Scaramuzza's strategy of iterative center detection could be succeeded. So our calibration method also does not need the visible circle bound to detect the image center.

The nonlinear refinement based on Bundle Adjustment is intended to maximize the likelihood in by minimizing the reprojection error. By experience we found optimizing both intrinsic and extrinsic parameters at the same time always fails into local minimum, so we split the optimization problem into two steps: refine the extrinsic parameters followed by optimize the intrinsic parameters, and solve this problem by iteration with Levenberg-Marquardt algorithm [15].

The proposed calibration method is implemented by a Matlab Toolbox and verified by both synthetic data and real object experiment, showing that our proposed linear approach is free from misleading and outperforms those baseline methods.

In stereo matching, our contribution lies in the implementation of multi-baseline stereo in disparity space.

The disparity space approach is nature in parallel and compact in data structure. Our implementation in Matlab shows it is at least 100 times faster than the general approach. GPU computation by CUDA would get an even higher computation speed, which is essential in case of the moving of camera.

What's more, the disparity space method is free from both lens distortion and perspective distortion, making the matching with a large local window feasible.

With the computed depth maps obtained by a moving camera system, the well-known ICP algorithm [20] is utilized to align those 3D points from depth map, and then the 3D reconstruction by the moving camera system is accomplished.

Chapter 2

Preliminaries and Notations

2.1 A Parametric Camera Model for Omnidirectional Camera

D. Scaramuzza's model [2], [3] of omnidirectional camera is shown in Fig. 2.1. Two distinct references are identified: the camera image plane(u'', v'') and the sensor plane(u', v'). The camera image plane is the same as the camera CCD, where the points are expressed in pixel coordinates. The sensor plane is a hypothetical plane orthogonal to the mirror axis, with the origin located at the plane-axis intersection.

Let X be a scene point. Then assume $u'' = [u'', v'']^T$ be the projection point of X into the sensor plane (Fig. 2.1 (b)), and $u' = [u', v']^T$ be its image in the camera plane (Fig. 2.1 (c)). As observed by B. Micusik [16], these two coordinate system are related by an affine transformation, which incorporates the digitizing process and small axes misalignments; thus $u'' = Au' + t$.

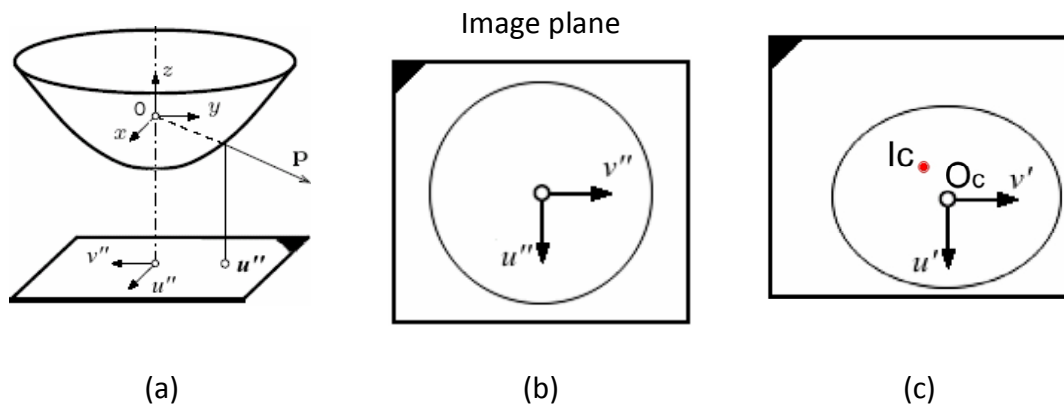


Fig. 2.1

At this point X , the imaging function g is introduced to captures the relationship between a point u'' in the sensor plane and the vector P emanating from the viewpoint O to the scene point X :

$$\lambda \cdot p = \lambda \cdot g(u'') = g(u' + t) = PX \quad (2.1)$$

where $P \in \mathfrak{R}^{3 \times 4}$ is the perspective projection matrix. Let us assume for g the following expression:

$$g(u'', v'') = (u'', v'', f(u'', v''))^T \quad (2.2)$$

A polynomial form is used to describe $f(u'', v'')$

$$f(u'', v'') = a_0 + a_1 \rho'' + a_2 \rho''^2 + \dots + a_N \rho''^N \quad (2.3)$$

where

$$\rho'' = \sqrt{u''^2 + v''^2}$$

What's more, according to research [17], [18] and [19], f should satisfy:

$$\left. \frac{df}{d\rho} \right|_{\rho=0} = 0 \quad (2.4)$$

So a_1 should be 0, and thus (2.3) can be rewritten as:

$$f(u'', v'') = a_0 + a_2 \rho''^2 + \dots + a_N \rho''^N \quad (2.5)$$

And the whole parametric camera model could be simplified as

$$\lambda \cdot \begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} = P \cdot X \quad (2.6)$$

The novelty of D. Scaramuzza's model is that it occupies a common polynomial equation to describe the distortion of the lens, thus free the user from some specific model which needs some prior knowledge about lens (fisheye or catadioptric, different kinds of mirrors in catadioptric case, etc.). So this camera model is truly general and easy to implement with D. Scaramuzza's toolbox [3].

After the calibration of the camera, the projection from 3D points onto the image is often used, here we simply demonstrate how it is implemented.

Assume there is a 3D point $M = [X, Y, Z]^T$, and its projection in the image is $m = [u, v]^T$

So such equation holds

$$\lambda \begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.7)$$

Where $\rho = \sqrt{u^2 + v^2}$

From equation (2.7), it could be derived that

$$\frac{f(\rho)}{\rho} = \frac{Z}{\sqrt{X^2 + Y^2}} \equiv c \quad (2.8)$$

here c is a constant determined by M

So

$$f(\rho) = c \cdot \rho \quad (2.9)$$

Thus ρ could be uniquely solved by the following polynomial equation

$$a_0 + (a_1 - c)\rho + a_2\rho^2 + a_3\rho^3 + a_4\rho^4 = 0 \quad (2.10)$$

With the solved ρ , the projection point in image $m = [u, v]^T$ is available:

$$m = [u, v]^T = \left[\rho \cdot \frac{X}{\sqrt{X^2+Y^2}}, \rho \cdot \frac{Y}{\sqrt{X^2+Y^2}} \right]^T \quad (2.11)$$

Thus the problem of projection from a 3D point onto the image is solved.

2.2 System Configuration

As shown in Fig. 1.1 (a), four well-polished planar mirrors are placed around the fisheye lens. The Field of View (FOV) and optical axes of those mirrored cameras is solely determined by the placement angle and the size of the mirrors.

So raises the problem of how to design such camera system: the optimum position and size of those mirrors. To the purpose of 3D reconstruction, common FOV of all subcameras is our main concern, since with the increased number of matching in different sides of view, the multi-baseline stereo would show its advantage in elimination of mismatching.

The system configuration in [1] should be extended since some of the fisheye lens does not have a view angle of 180° in arbitrary sectional side, which may lead such former design invalid with such fisheye lens (Fig. 2.2).

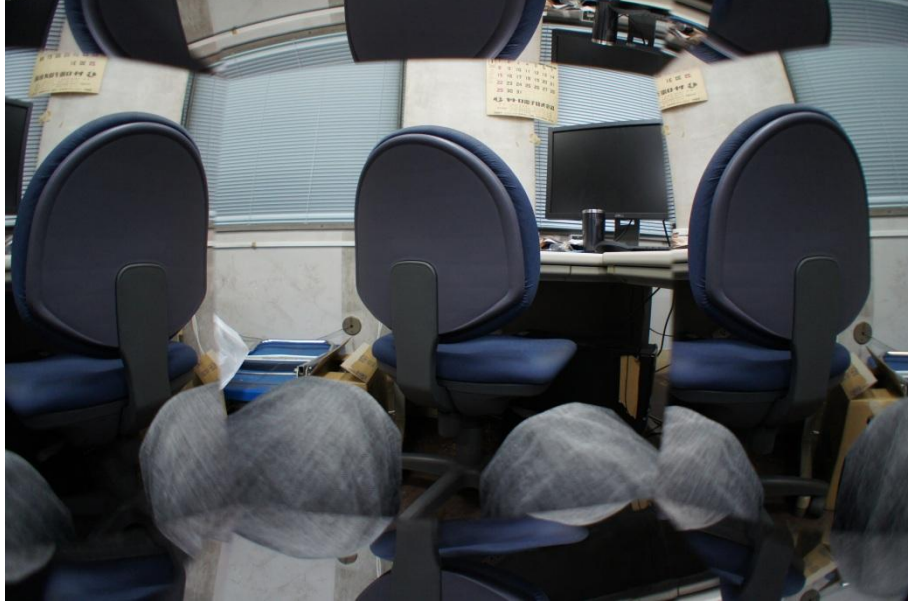


Fig. 2.2 Image captured by the proposed system with Sony NEX-5D with Lens: 16mm/F2.8+ Fisheye converter. This camera's maximum view-angle is only about 160° in horizontal and 90° in vertical, which leads to an invalid system with such small common FOV.

The extended model is shown in Fig. 2.3. Here λ means the maximum view-angle of the lens in the slide. With the limitation of mirrors, α, β are the view-angles of the center real camera O and mirrored subcamera O' respectively. The mirror size is m , and n, b are the distance from mirrors to Y axis and Z axis. γ and θ mean the mirror angle and the angle between the axes of the two cameras.

The relationship between the above parameters is shown as follows:

$$\alpha = \pi - 2 \cdot \tan^{-1} \frac{m \cdot \sin \gamma + b \cdot \cot \frac{\lambda}{2}}{m \cdot \cos \gamma + b} = \pi - 2 \cdot \tan^{-1} \frac{\hat{m} + \cot \frac{\lambda}{2}}{\hat{m} \cdot \cos \gamma + 1} \quad (2.12)$$

$$\beta = \frac{\lambda - \alpha}{2} \quad (2.13)$$

$$\theta = \pi - 2 \cdot \gamma \quad (2.14)$$

where $\hat{m} = \frac{m}{b}$, represents a normalized mirror size.

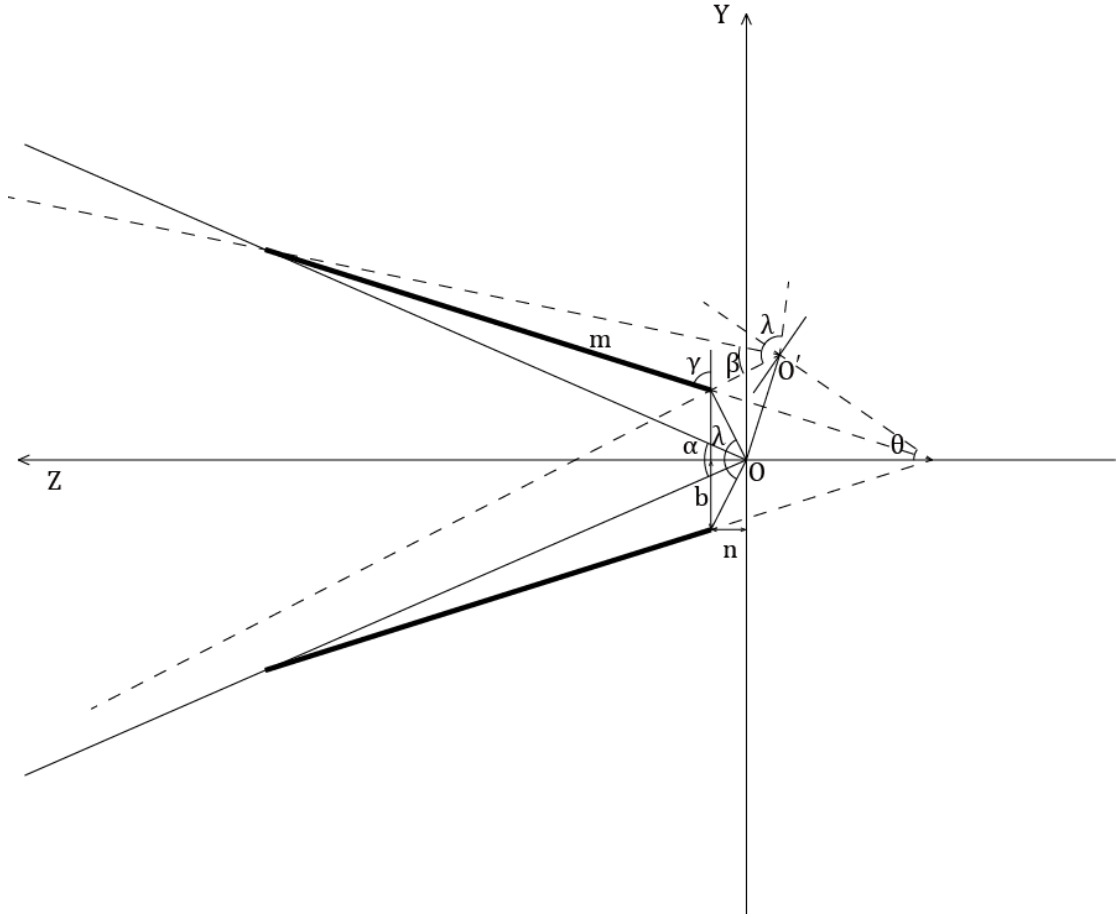


Fig. 2.3 The extended model for mirror placement and mirrored camera

Compare to distance from the camera system to object, the baseline is relatively short, so it is reasonable to put do such approximation in computing the common view angle Ω (Fig. 2.4):

$$\Omega = \min(\alpha_2, \beta_2) - \max(\alpha_1, \beta_1) \quad (2.15)$$

where

$$\alpha_1 = \tan^{-1} \frac{\hat{m} + \cot \frac{\lambda}{2}}{\hat{m} \cdot \cos \gamma + 1}$$

$$\alpha_2 = \pi - \alpha_1$$

$$\beta_1 = 2\gamma - \frac{1}{2} \cdot \lambda + \frac{\pi}{2} - \alpha_1$$

$$\beta_2 = 2\gamma$$

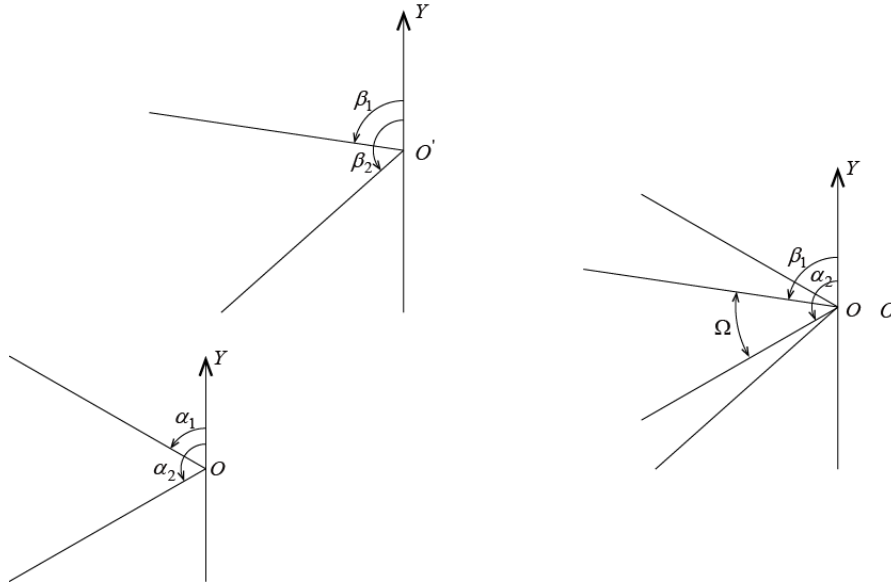


Fig. 2.4 Common view angle for a far distant object

Fig. 2.5 portrays the relationship between common view angle with mirror angle with different maximum view-angle and different normalized mirror size:

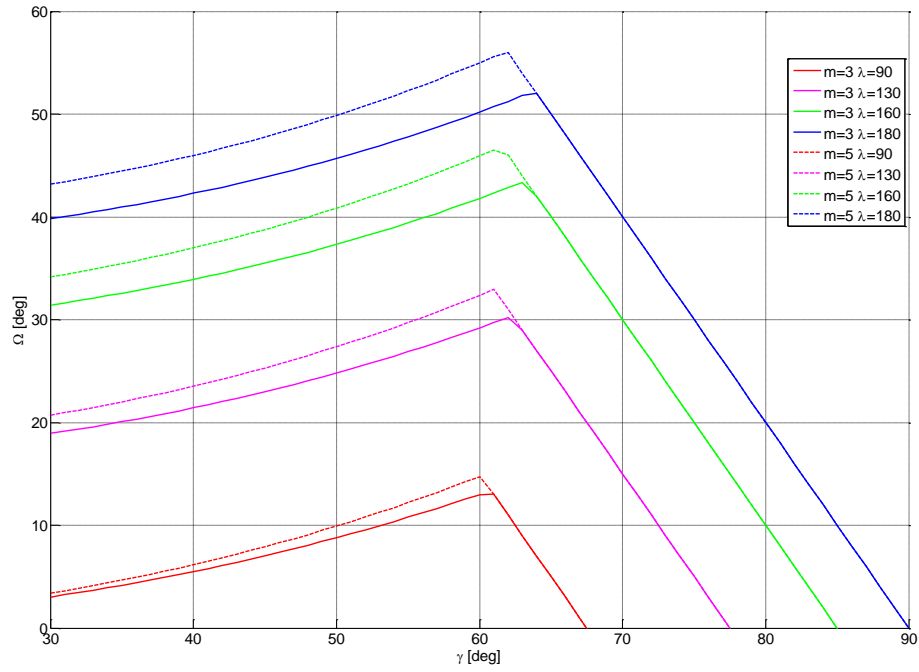


Fig. 2.5 Common angle Ω

From Fig. 2.5, such conclusions about the design of the system could be obtained:

- With the decrease in maximum view-angle (λ), the optimum common view angle (Ω) decreases;
- The optimum mirror angle (γ) is not affected by the decrease in maximum view-angle (λ) and is always around 60 degrees;
- With a fixed maximum view-angle (λ), a larger \hat{m} is preferred, which mean we should make the mirror bigger (increase m) or move the mirror closer to the camera in parallel (decrease b);

2.3 Geometry of Catadioptric Stereo

2.3.1 Position and Orientation of the Mirrored Cameras

As stated in [1], images by one real camera and four mirrored cameras could be taken in one shot, and the extrinsic parameter between cameras could be determined in 2 steps:

Step 1: Determine the position of Mirrors

As depicted in Fig. 2.6, with the geometry relationship between a real and mirrored target, the mirror position and orientation could be determined:

$$\mathbf{n}_i^m = \frac{\mathbf{n}_o^t - \mathbf{n}_i^t}{|\mathbf{n}_o^t - \mathbf{n}_i^t|} \quad (2.16)$$

$$d_i^m = \frac{(\mathbf{n}_o^t - \mathbf{n}_i^t) \cdot \mathbf{n}_i^t}{|\mathbf{n}_o^t - \mathbf{n}_i^t| (1 - \mathbf{n}_o^t \cdot \mathbf{n}_i^t)} \cdot (d_o^t - d_i^t) \quad (2.17)$$

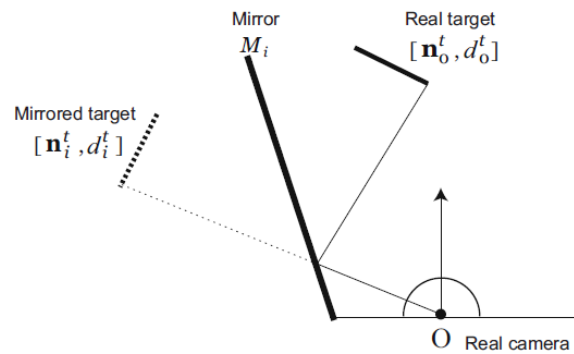


Fig. 2.6 Real and mirrored targets

Step 2: Determine the position of Mirrored Cameras

As shown in Fig. 2.7, the real camera and mirrored camera are symmetry about the mirror, which position and orientation are already known, so the optical center O'_i and the rotation matrix R_i of the i^{th} mirrored camera are available:

$$O'_i = -2d_i^m \mathbf{n}_i^m \quad (2.18)$$

$$R_i = I - 2\mathbf{n}_i^m \mathbf{n}_i^{m^T} \quad (2.19)$$

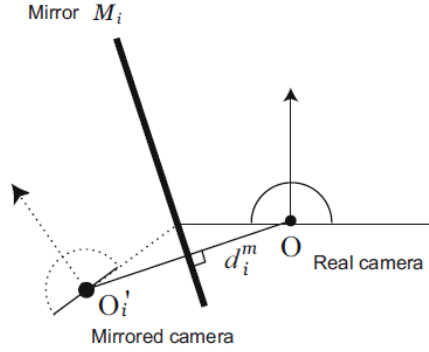


Fig. 2.7 Real and mirrored cameras

Look deep into (2.18) and (2.19) we may find the transform from the real camera to one mirrored camera has only 3 degree of freedom (DOF) while a general transform should have 6 DOF. The decrease in DOF is caused by the limitation of ‘mirrored motion’, which cuts 1 DOF in rotation and 2 DOF in translation. The 3 DOF could also be explained by the fact that the mirror plane in 3D space has 3 DOF:

$$a \cdot x + b \cdot y + c \cdot z = d \quad (2.20)$$

with the constraint

$$a^2 + b^2 + c^2 + d^2 = 1 \quad (2.21)$$

2.3.2 Restrict 6 DOF to 3 DOF

Our proposed calibration starts from a general case so there raises the problem of how to restrict the general 6 DOF to the mirror 3 DOF.

Our proposed strategy is to calculate the normal vector of the rotation first then use this normal vector to cut 1 DOF in rotation and 2 DOF in translation.

Let us assume a general rotation matrix R rotates the vector $\mathbf{n}_0 = [0 \ 0 \ -1]^T$:

$$\mathbf{n}_r = R \cdot \mathbf{n}_0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} = -[r_{13} \ r_{23} \ r_{33}]^T = -R(:,3) \quad (2.22)$$

where $R(:, i), i = 1, 2, 3$ indicates the i^{th} column of the rotation matrix R .

Thus the normal vector of mirrors is:

$$\mathbf{n} = \frac{\mathbf{n}_0 - \mathbf{n}_r}{\|\mathbf{n}_0 - \mathbf{n}_r\|} \quad (2.23)$$

Then use equation (2.19) to get the mirror rotation matrix.

The normal vector of the mirror $\mathbf{n} = [n_1 \ n_2 \ n_3]^T$ could be used to restrict the direction of a general translation $\mathbf{b}_0 = [b_1 \ b_2 \ b_3]^T$ by solving k in the following linear system:

$$\begin{cases} k \cdot n_1 = b_1 \\ k \cdot n_2 = b_2 \\ k \cdot n_3 = b_3 \end{cases} \quad (2.24)$$

Finally the mirror translation \mathbf{b} equals:

$$\mathbf{b} = k \cdot \mathbf{n} \quad (2.25)$$

Chapter 3

Calibration of the Monocular Multi-view Camera System

By calibration our monocular multi-view camera system we mean the estimation of the intrinsic parameters of the fisheye camera and the extrinsic parameters which describe the transform between subcameras and chessboard. This calibration is extended from D. Scaramuzza's calibration method for single omnidirectional camera [2], [3] and inherits the advantage of auto center detection. In detail, our calibration composes of a linear estimation followed by a nonlinear refinement based on bundle adjustment. Finally, we demonstrate the possibility to extend our calibration method to general multiple omnidirectional cameras system.

3.1 The Formulation of Calibration

Davide Scaramuzza's camera model could be summed up in Fig. 2.1 (see detail in Section 2.1) and equation:

$$\lambda_{ij} \cdot \begin{bmatrix} u_{ij} \\ v_{ij} \\ f(\rho_{ij}) \end{bmatrix} = P_{ij} \quad (3.1)$$

Where $f(\rho_{ij}) = a_0 + \dots + a_N \rho_{ij}^N$

In detail, a polynomial equation $f(\rho_{ij})$ is utilized to describe the distortion characteristic of the omnidirectional camera.

When it comes to calibration of single Omnidirectional Camera, equation (1) could be extended to

$$\lambda_{ij} \cdot \begin{bmatrix} u_{ij} \\ v_{ij} \\ f(\rho_{ij}) \end{bmatrix} = P_{ij} = [r_1^i \quad r_2^i \quad r_3^i \quad t^i] \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 0 \\ 1 \end{bmatrix} = [r_1^i \quad r_2^i \quad t^i] \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 1 \end{bmatrix} \quad (3.2)$$

The detailed calibration method is shown in his paper [2], [3].

Our Monocular Multi-view Camera system (Fig. 1.1 (a)) is composed of one fisheye camera and four mirrors around it, which it equivalent to five fisheye cameras at different positions (Fig. 3.1).

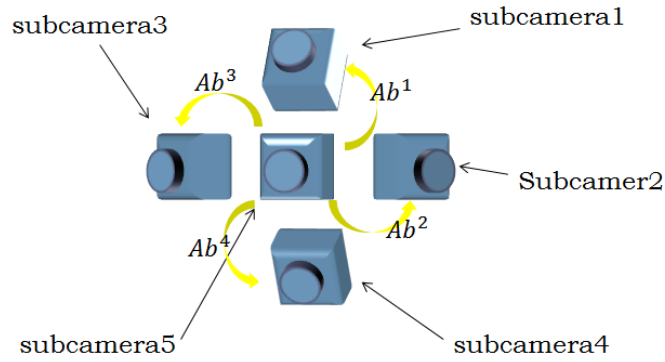


Fig. 3.1 The equivalent camera system

As shown in Fig. 3.1, the homogeneous transform matrix Ab^m ($m = 1,2,3,4$) could be used to describe the relative space transformation between the center real subcamera and the other virtual subcameras.

Fig. 1.2 shows a typical image taken by this camera system, which could be divided into five regions and the image in each region is viewed in different perspectives.

Thus equation (3.2) could be transformed in to the following structure:

$$\lambda_{ij} \begin{bmatrix} u_{ij} \\ v_{ij} \\ f(\rho_{ij}) \\ 1/\lambda_{ij} \end{bmatrix} = \left(\begin{bmatrix} a_{11}^m & a_{12}^m & a_{13}^m & b_1^m \\ a_{21}^m & a_{22}^m & a_{23}^m & b_2^m \\ a_{31}^m & a_{32}^m & a_{33}^m & b_3^m \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11}^n & r_{12}^n & r_{13}^n & t_1^n \\ r_{21}^n & r_{22}^n & r_{23}^n & t_2^n \\ r_{31}^n & r_{32}^n & r_{33}^n & t_3^n \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 0 \\ 1 \end{bmatrix} \right) = 0 \quad (3.3)$$

Cross product could be applied to eliminate the unknown scale factor λ_{ij} ,

$$\begin{bmatrix} u_{ij} \\ v_{ij} \\ f(\rho_{ij}) \\ 1/\lambda_{ij} \end{bmatrix} \wedge \left(\begin{bmatrix} a_{11}^m & a_{12}^m & a_{13}^m & b_1^m \\ a_{21}^m & a_{22}^m & a_{23}^m & b_2^m \\ a_{31}^m & a_{32}^m & a_{33}^m & b_3^m \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11}^n & r_{12}^n & r_{13}^n & t_1^n \\ r_{21}^n & r_{22}^n & r_{23}^n & t_2^n \\ r_{31}^n & r_{32}^n & r_{33}^n & t_3^n \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 0 \\ 1 \end{bmatrix} \right) = 0 \quad (3.4)$$

Then equation (3.4) could then be expanded and simplified as follows:

$$\begin{bmatrix} u_{ij} \\ v_{ij} \\ f(\rho_{ij}) \end{bmatrix} \wedge \left(\begin{bmatrix} a_{11}^m \cdot r_{11}^n + a_{12}^m \cdot r_{21}^n + a_{13}^m \cdot r_{31}^n & a_{11}^m \cdot r_{12}^n + a_{12}^m \cdot r_{22}^n + a_{13}^m \cdot r_{32}^n & b_1^m + a_{11}^m \cdot t_1^n + a_{12}^m \cdot t_2^n + a_{13}^m \cdot t_3^n \\ a_{21}^m \cdot r_{11}^n + a_{22}^m \cdot r_{21}^n + a_{23}^m \cdot r_{31}^n & a_{21}^m \cdot r_{12}^n + a_{22}^m \cdot r_{22}^n + a_{23}^m \cdot r_{32}^n & b_2^m + a_{21}^m \cdot t_1^n + a_{22}^m \cdot t_2^n + a_{23}^m \cdot t_3^n \\ a_{31}^m \cdot r_{11}^n + a_{32}^m \cdot r_{21}^n + a_{33}^m \cdot r_{31}^n & a_{31}^m \cdot r_{12}^n + a_{32}^m \cdot r_{22}^n + a_{33}^m \cdot r_{32}^n & b_3^m + a_{31}^m \cdot t_1^n + a_{32}^m \cdot t_2^n + a_{33}^m \cdot t_3^n \end{bmatrix} \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 1 \end{bmatrix} \right) = 0 \quad (3.5)$$

Now, let us focus on a particular observation of the calibration pattern. From (3.5), we have each point $(X_{ij}, Y_{ij}, 0)$ on the pattern contributes three homogeneous equations:

$$\begin{cases} v_{ij} \cdot [b_3^m + a_{31}^m \cdot t_1^n + a_{32}^m \cdot t_2^n + a_{33}^m \cdot t_3^n] + v_{ij} \cdot X_{ij} \cdot [a_{31}^m \cdot r_{11}^n + a_{32}^m \cdot r_{21}^n + a_{33}^m \cdot r_{31}^n] + v_{ij} \cdot Y_{ij} \cdot [a_{31}^m \cdot r_{12}^n + a_{32}^m \cdot r_{22}^n + a_{33}^m \cdot r_{32}^n] \\ - f(\rho_{ij}) \cdot [b_2^m + a_{21}^m \cdot t_1^n + a_{22}^m \cdot t_2^n + a_{23}^m \cdot t_3^n] - f(\rho_{ij}) \cdot X_{ij} \cdot [a_{21}^m \cdot r_{11}^n + a_{22}^m \cdot r_{21}^n + a_{23}^m \cdot r_{31}^n] - f(\rho_{ij}) \cdot Y_{ij} \cdot [a_{21}^m \cdot r_{12}^n + a_{22}^m \cdot r_{22}^n + a_{23}^m \cdot r_{32}^n] = 0 \\ \\ u_{ij} \cdot [b_3^m + a_{31}^m \cdot t_1^n + a_{32}^m \cdot t_2^n + a_{33}^m \cdot t_3^n] + u_{ij} \cdot X_{ij} \cdot [a_{31}^m \cdot r_{11}^n + a_{32}^m \cdot r_{21}^n + a_{33}^m \cdot r_{31}^n] + u_{ij} \cdot Y_{ij} \cdot [a_{31}^m \cdot r_{12}^n + a_{32}^m \cdot r_{22}^n + a_{33}^m \cdot r_{32}^n] \\ - f(\rho_{ij}) \cdot [b_1^m + a_{11}^m \cdot t_1^n + a_{12}^m \cdot t_2^n + a_{13}^m \cdot t_3^n] - f(\rho_{ij}) \cdot X_{ij} \cdot [a_{11}^m \cdot r_{11}^n + a_{12}^m \cdot r_{21}^n + a_{13}^m \cdot r_{31}^n] - f(\rho_{ij}) \cdot Y_{ij} \cdot [a_{11}^m \cdot r_{12}^n + a_{12}^m \cdot r_{22}^n + a_{13}^m \cdot r_{32}^n] = 0 \\ \\ u_{ij} \cdot [b_2^m + a_{21}^m \cdot t_1^n + a_{22}^m \cdot t_2^n + a_{23}^m \cdot t_3^n] + u_{ij} \cdot X_{ij} \cdot [a_{21}^m \cdot r_{11}^n + a_{22}^m \cdot r_{21}^n + a_{23}^m \cdot r_{31}^n] + u_{ij} \cdot Y_{ij} \cdot [a_{21}^m \cdot r_{12}^n + a_{22}^m \cdot r_{22}^n + a_{23}^m \cdot r_{32}^n] \\ - v_{ij} \cdot [b_1^m + a_{11}^m \cdot t_1^n + a_{12}^m \cdot t_2^n + a_{13}^m \cdot t_3^n] - v_{ij} \cdot X_{ij} \cdot [a_{11}^m \cdot r_{11}^n + a_{12}^m \cdot r_{21}^n + a_{13}^m \cdot r_{31}^n] - v_{ij} \cdot Y_{ij} \cdot [a_{11}^m \cdot r_{12}^n + a_{12}^m \cdot r_{22}^n + a_{13}^m \cdot r_{32}^n] = 0 \end{cases} \quad (3.6)$$

In the above three equations, u_{ij}, v_{ij}, X_{ij} and Y_{ij} are all already known. The purpose of the calibration is to solve those Ab^m (describe the space transformation between subcameras), Rt^n (describe the position of chessboard in each shot) and (a_0, a_2, a_3, a_4) (intrinsic parameters which describe the distortion of fisheye camera).

When analyze deeply in these three homogeneous equations, we may find that though the parameters to solve are nonlinear, they are symmetric in combination thus provide us some strategy to solve these parameters step by step.

According to Jiang's paper [1] (see detail in Section 2.3), the transformation matrix from base camera to subcamera should have only 3 Degree of Freedom (DOF); however the above transformation serves as a general transformation, which holds 6 DOF. For one thing, we hope our model as general as possible, for another, put the constraint of 3 DOF at the beginning would make the equation highly nonlinear, which is impossible to solve. A strategy is provided to take the constraints into consideration in section 2.3.2.

3.2 The Solution of Calibration Model

3.2.1 Solving for camera extrinsic parameters

Observe that the first two equations of (3.6) are highly nonlinear since $f(\rho_{ij})$ is multiplied with other unknown factor there. Thus we should begin with the third equation of (3.6).

First let us put those nonlinear combinations together.

$$\begin{cases} b_2^m + a_{21}^m \cdot t_1^n + a_{22}^m \cdot t_2^n + a_{23}^m \cdot t_3^n = E_1^{mn} \\ a_{21}^m \cdot r_{11}^n + a_{22}^m \cdot r_{21}^n + a_{23}^m \cdot r_{31}^n = E_2^{mn} \\ a_{21}^m \cdot r_{12}^n + a_{22}^m \cdot r_{22}^n + a_{23}^m \cdot r_{32}^n = E_3^{mn} \\ b_1^m + a_{11}^m \cdot t_1^n + a_{12}^m \cdot t_2^n + a_{13}^m \cdot t_3^n = E_4^{mn} \\ a_{11}^m \cdot r_{11}^n + a_{12}^m \cdot r_{21}^n + a_{13}^m \cdot r_{31}^n = E_5^{mn} \\ a_{11}^m \cdot r_{12}^n + a_{12}^m \cdot r_{22}^n + a_{13}^m \cdot r_{32}^n = E_6^{mn} \end{cases} \quad (3.7)$$

Then the third equation of (3.6) would become linear

$$u_{ij} \cdot E_1^{mn} + u_{ij} \cdot X_{ij} \cdot E_2^{mn} + u_{ij} \cdot Y_{ij} \cdot E_3^{mn} - v_{ij} \cdot E_4^{mn} - v_{ij} \cdot X_{ij} \cdot E_5^{mn} - v_{ij} \cdot Y_{ij} \cdot E_6^{mn} = 0 \quad (3.8)$$

By stacking all the unknown entries of (3.8) into a vector we rewrite the equation (3.8) for all points of the calibration pattern as a system of linear equations

$$M \cdot H = 0 \quad (3.9)$$

where

$$M = \begin{bmatrix} u_{i1} & u_{i1} \cdot X_{i1} & u_{i1} \cdot Y_{i1} & -v_{i1} & -v_{i1} \cdot X_{i1} & -v_{i1} \cdot Y_{i1} \\ u_{i2} & u_{i2} \cdot X_{i2} & u_{i2} \cdot Y_{i2} & -v_{i2} & -v_{i2} \cdot X_{i2} & -v_{i2} \cdot Y_{i2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{ij} & u_{ij} \cdot X_{ij} & u_{ij} \cdot Y_{ij} & -v_{ij} & -v_{ij} \cdot X_{ij} & -v_{ij} \cdot Y_{ij} \end{bmatrix}$$

$$H = [E_1^{mn} \quad E_2^{mn} \quad E_3^{mn} \quad E_4^{mn} \quad E_5^{mn} \quad E_6^{mn}]^T$$

A linear estimate of H can be obtained by minimizing the least-squares criterion $\min \|M \cdot H\|^2$, subject to $\|H\|^2 = 1$, which could be accomplished by using the SVD. The solution of (3.9) is known up to a scale factor k^{mn} .

Notice in (3.7), $E_2^{mn}, E_3^{mn}, E_5^{mn}$ and E_6^{mn} could be expressed in the following way

$$\begin{bmatrix} a_{11}^m & a_{12}^m & a_{13}^m \\ a_{21}^m & a_{22}^m & a_{23}^m \\ a_{31}^m & a_{32}^m & a_{33}^m \end{bmatrix} \cdot \begin{bmatrix} r_{11}^n & r_{12}^n & r_{13}^n \\ r_{21}^n & r_{22}^n & r_{23}^n \\ r_{31}^n & r_{32}^n & r_{33}^n \end{bmatrix} = \begin{bmatrix} E_5^{mn} & E_6^{mn} & * \\ E_2^{mn} & E_3^{mn} & * \\ * & * & * \end{bmatrix} \quad (3.10)$$

Since the product of two rotation matrixes is still a rotation matrix, which is orthonormal. Because of the orthonormality, the unknown scale factor k^{mn} can be computed uniquely, thus H is available now.

Let us still focus on (3.10) and rewrite it as $A^m \cdot R^n = E^{mn}$. As the described in the notations, when $\text{mod}(m, 5) = 0$, $A^m = I$ since it is directly shot by the base-camera. So $R^n = E^{5k,n}, k \in N$, which gives us an initial estimation of R^n .

In different shots, with the change of position of chessboard, R^n may change while A^m should be constant since the space translations between subcameras are constant.

Here a 2 Steps algorithm is proposed to solve A^m and R^n :

Step 1: Mean rotation of A^m

In each shot, A^m could be calculated with the estimated value of R^n

$$A_n^m = E^{mn} \cdot R^{nT} \quad (3.11)$$

Here we write A^m as A_n^m because due to errors, A^m calculated by each shot is different. A natural idea to get a good estimation of rotation matrix A^m among all rotation matrixes A_n^m is to calculate the 'mean rotation' of all A_n^m

$$A^m = \wp(A_1^m, A_2^m \dots A_N^m) \quad (3.12)$$

\wp means the mean rotation in Riemannian space, and M. Moakher in [9] derives the formula to calculate the mean rotation matrix

$$A^m = A_1^m \left(A_1^{m^T} A_2^m \left(A_2^{m^T} A_3^m \left(\dots A_{N-1}^m \left(A_{N-1}^{m^T} A_N^m \right)^{\frac{1}{2}} \right)^{\frac{2}{3}} \dots \right)^{\frac{N-2}{N-1}} \right)^{\frac{N-1}{N}} \quad (3.13)$$

Step 2: Mean rotation of R^n

Now we have the overall estimation of A^m from all shots, which makes it possible to update the initial estimation of R^n

$$R^n = A^{m^T} \cdot E^{m,n} \quad (3.14)$$

In this term, we could get M matrixes with each shot. Just like as done above, let us denote them as R_m^n and estimate R^n by mean rotation matrix of those R_m^n

$$R^n = \phi(R_1^n, R_2^n \dots R_M^n) \quad (3.15)$$

Since we have a better estimation of R^n now, we could repeat Step 1 to get better estimation of A^m then repeat Step 2 to achieve better estimation of R^n . Our experience is usually 5 such iteration is enough to converge.

Up to now, all the parameters related to rotation are solved. Let's move on to (3.7.1) and (3.7.2) to solve translation related parameters.

$$\begin{cases} b_2^m + a_{21}^m \cdot t_1^n + a_{22}^m \cdot t_2^n + a_{23}^m \cdot t_3^n = E_1^{mn} \\ b_1^m + a_{11}^m \cdot t_1^n + a_{12}^m \cdot t_2^n + a_{13}^m \cdot t_3^n = E_4^{mn} \end{cases} \quad (3.16)$$

With the estimation of rotation, (3.16) is a linear system. However, if we expand such equation, we would get such linear system

$$\begin{array}{rcl}
a_{11}^1 t_1^1 + a_{12}^1 t_2^1 + a_{13}^1 t_3^1 & + b_1^1 & = E_4^{11} \\
a_{21}^1 t_1^1 + a_{22}^1 t_2^1 + a_{23}^1 t_3^1 & + b_2^1 & = E_1^{11} \\
a_{11}^2 t_1^1 + a_{12}^2 t_2^1 + a_{13}^2 t_3^1 & + b_1^2 & = E_4^{21} \\
a_{21}^2 t_1^1 + a_{22}^2 t_2^1 + a_{23}^2 t_3^1 & + b_2^2 & = E_1^{21} \\
a_{11}^3 t_1^1 + a_{12}^3 t_2^1 + a_{13}^3 t_3^1 & + b_1^3 & = E_4^{31} \\
a_{21}^3 t_1^1 + a_{22}^3 t_2^1 + a_{23}^3 t_3^1 & + b_2^3 & = E_1^{31} \\
a_{11}^4 t_1^1 + a_{12}^4 t_2^1 + a_{13}^4 t_3^1 & + b_1^4 & = E_4^{41} \\
a_{21}^4 t_1^1 + a_{22}^4 t_2^1 + a_{23}^4 t_3^1 & + b_2^4 & = E_1^{41} \\
t_1^1 & & = E_4^{51} \\
& & = E_1^{51} \\
& & = E_4^{12} \\
& & = E_1^{12} \\
& & = E_4^{22} \\
& & = E_1^{22} \\
& & = E_4^{32} \\
& & = E_1^{32} \\
& & = E_4^{42} \\
& & = E_1^{42} \\
& & = E_4^{52} \\
& & = E_1^{52} \\
& & \dots
\end{array}
\quad (3.17)$$

It is fairly possible that the group of (b_1^m, b_2^m) changes correspond to the change of t_3^n while the total evaluation of the system does not change, which means it is now impossible to solve (b_1^m, b_2^m) and t_3^n (The matrix is not full-rank). However, we could combine them together then solve these combinations

Let

$$\begin{cases} b_1^m + a_{13}^m \cdot t_3^n = C_1^{mn} \\ b_2^m + a_{23}^m \cdot t_3^n = C_2^{mn} \end{cases} \quad (3.18)$$

(3.16) could be rewritten as

$$\begin{cases} C_1^{mn} + a_{11}^m \cdot t_1^n + a_{12}^m \cdot t_2^n = E_4^{mn} \\ C_2^{mn} + a_{21}^m \cdot t_1^n + a_{22}^m \cdot t_2^n = E_1^{mn} \end{cases} \quad (3.19)$$

Then the least-squares solution of $C_1^{mn}, C_2^{mn}, t_1^n$ and t_2^n could be obtained by using pseudo inverse.

3.2.2 Solving for camera intrinsic parameters

Till now, we are working on the third equation of (3.6), having got the solution of rotation parameters together with t_1^n, t_2^n , and partly got the solution of b_1^m, b_1^m and t_3^n .

Now let's move on to the first two equations of (3.6) to solve those intrinsic parameters and the rest of extrinsic parameters.

By substitute (3.7) into these two equations, system (3.6.1) and (3.6.2) could be simplified as

$$\begin{cases} v_i \cdot C_3^{mn} + F_1^i \cdot a_0 + \rho_i^2 \cdot F_1^i \cdot a_2 + \rho_i^3 \cdot F_1^i \cdot a_3 + \rho_i^4 \cdot F_1^i \cdot a_4 = F_2^i \\ u_i \cdot C_3^{mn} + F_3^i \cdot a_0 + \rho_i^2 \cdot F_3^i \cdot a_2 + \rho_i^3 \cdot F_3^i \cdot a_3 + \rho_i^4 \cdot F_3^i \cdot a_4 = F_4^i \end{cases} \quad (3.20)$$

where

$$\begin{cases} F_1^{ij} = -[E_1^{mn} + X_{ij} \cdot E_2^{mn} + Y_{ij} \cdot E_3^{mn}] \\ F_2^{ij} = -v_{ij} \cdot [a_{31}^m \cdot t_1^n + a_{32}^m \cdot t_2^n] - v_{ij} \cdot X_{ij} \cdot [a_{31}^m \cdot t_{11}^n + a_{32}^m \cdot t_{21}^n + a_{33}^m \cdot t_{31}^n] - v_{ij} \cdot Y_{ij} \cdot [a_{31}^m \cdot t_{12}^n + a_{32}^m \cdot t_{22}^n + a_{33}^m \cdot t_{32}^n] \\ F_3^{ij} = -[E_4^{mn} + X_{ij} \cdot E_5^{mn} + Y_{ij} \cdot E_6^{mn}] \\ F_4^{ij} = -u_{ij} \cdot [a_{31}^m \cdot t_1^n + a_{32}^m \cdot t_2^n] - u_{ij} \cdot X_{ij} \cdot [a_{31}^m \cdot t_{11}^n + a_{32}^m \cdot t_{21}^n + a_{33}^m \cdot t_{31}^n] - u_{ij} \cdot Y_{ij} \cdot [a_{31}^m \cdot t_{12}^n + a_{32}^m \cdot t_{22}^n + a_{33}^m \cdot t_{32}^n] \end{cases}$$

$$C_3^{mn} = b_3^m + a_{33}^m \cdot t_3^n$$

As done before, the linear system (3.20) could be solved by using pseudoinverse.

Finally, we solve the following equations to obtain all the parameters.

$$\begin{cases} b_1^m + a_{13}^m \cdot t_3^n = C_1^i \\ b_2^m + a_{23}^m \cdot t_3^n = C_2^i \\ b_3^m + a_{33}^m \cdot t_3^n = C_3^i \end{cases} \quad (3.21)$$

Now all the parameters are obtained. Although the above solving method is complicate, the calculations there are all linear, which means that the computation is really fast.

3.3 Iterative Center detection

To desire the capability of identifying the center of the omnidirectional image O_c , We similarly employ D. Scaramuzza's way to detect the camera center.

To this end, observe that our calibration procedure correctly estimates the intrinsic parametric model only if O_c is taken as origin of the image coordinates. If not so, we observe that the reprojection error would be quite large. Motivated by this observation, we assume that the Sum of Squared Reprojection Errors (SSRE) always has a global minimum at the correct center location.

This assumption leads us to an iterative search of the center O_c :

1. At each step of this iterative search, a particular image region is uniformly sampled in a certain number points.
2. For each of these points, calibration is performed by using that point as a potential center location, and SSRE is calculated.
3. The point giving the minimum SSRE is assumed as a potential center.
4. The search proceeds by refining the sampling in the region around that point, and steps 1,2 and 3 are repeated until the stop conditions (small difference between two potential center locations or total number of iteration) is satisfied.

As mentioned above, since the computation for each calibration is really fast, the iterative search does not take long to stop and would provide reasonable precision.

3.4 Non-linear Refinement by Bundle-Adjustment

The linear solution given in previous subsections 3.1, 3.2, 3.3 is obtained through minimizing an algebraic distance, which is not physically meaningful. To this end, we chose to refine it through maximum likelihood inference.

If we have taken N shots with M subcameras and J corner points on chess board, let us assume that those corner points are corrupted by independent and identically distributed noise. Then the maximum likelihood estimate could be derived by minimizing the following cost function:

$$CostFunction = \sum_{m=1}^M \sum_{n=1}^N \sum_{j=1}^J \left\| \begin{bmatrix} u_{mnj} & v_{mnj} \end{bmatrix}^T - m \left(\begin{bmatrix} A^m & b^m \end{bmatrix} \begin{bmatrix} R^n & t^n \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_j & Y_j & 0 & 1 \end{bmatrix}^T \right) \right\|^2 \quad (3.22)$$

where $m \left(\begin{bmatrix} A^m & b^m \end{bmatrix} \begin{bmatrix} R^n & t^n \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_j & Y_j & Z_j & 1 \end{bmatrix}^T \right)$ is the projection of the point $\begin{bmatrix} X_j & Y_j & Z_j & 1 \end{bmatrix}^T$ of the n^{th} shot in m^{th} subcamera (see detail in Section 2.1). Observe that now we could incorporate the affine matrix AF (see in [2], [3]) and the center of the omnidirectional image O_c into this cost function.

Both extrinsic parameters (A^m, b^m, R^n, t^n) and intrinsic parameters $((a_0, a_2, a_3, a_4), AF, O_c)$ are optimized by minimizing the cost function (3.22), which is actually minimizing the reprojection error. As mentioned in [2], we could split the minimization into two steps:

1. Refines the extrinsic parameters;
2. Uses the extrinsic parameters refined in step (1) to refine the intrinsic ones;

To minimize (3.22) separately, we use the Levenberg-Marquadt algorithm [15], which is available in Matlab's Optimization Toolbox implemented by the function

lsqnonlin. Our linear calibration results would provide the initial guess of the algorithm. We choose the unitary matrix as the guess of AF , while for O_c we used the position estimated through the iterative procedure explained in Section 3.3.

Because of the difference between our camera system and single omnidirectional camera (more parameters to estimate), we repeat step (1) and step (2) in sequence to get a better calibration results, usually 10 iterations are enough to converge.

Chapter 4

Rectification

In this section, we present two ways to rectify the image: perspective reprojection and epipolar curve. The purpose of rectification is to compensate the lens distortion and making the stereo matching easier: restricted into one dimension. In perspective reprojection, we reproject the images obtained by five subcameras to five parallel planes, making the system equivalent to 5 perspective camera with the same focal length. In epipolar curve, we derive the epipolar constraint in our camera system by directly project a 3D vector from center camera into subcameras around it. All these two approaches are supported by the calibration results. Finally, we compare these two approaches and choose the epipolar curve for further research.

4.1 Perspective Reprojection

In the perspective reprojection approach, five perspective projection images are created for further research on stereo matching.

As shown in the slide of the camera system in Fig.4.1, with the calibration results of intrinsic and extrinsic parameters of the camera system, we project 5 sub-images into 5 different parallel planes.

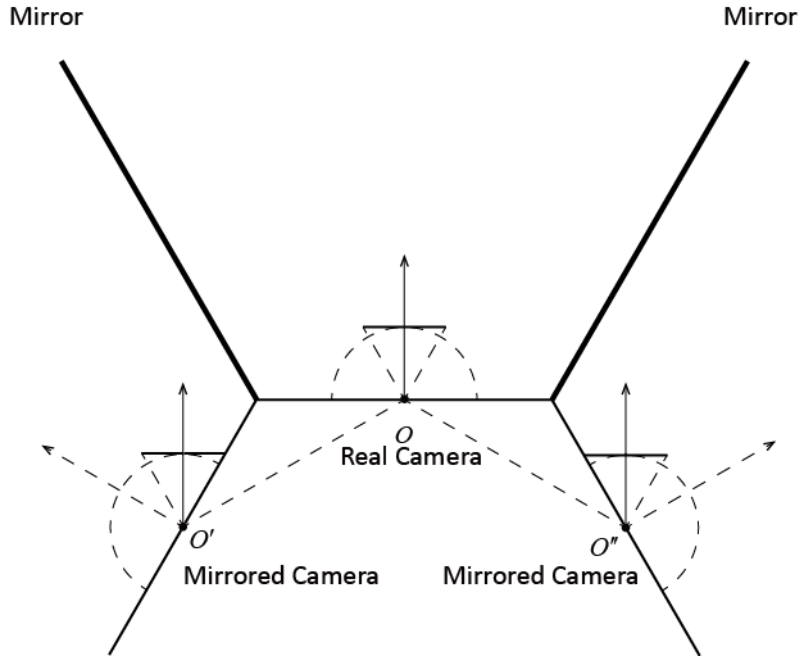


Fig. 4.1 Projection planes for the reprojection in slide of the camera system

In detail, the reprojection is done actually in an inverse process as described above: for each pixel in reprojection image, we compute its position in 3D space then do the 'world to camera' projection (see detail in 2.1) to get its position in pixel in original images, finally a bilinear interpolation is used to get the intensity of the pixel with sub-pixel precision.

After rectification of perspective reprojection, the camera system is equivalent to parallel stereo with an anteroposterior offset composed of 5 perspective camera with known position.

4.2 Epipolar Curve

Another way of rectification is choosing epipolar curve. As shown in Fig. 4.2, consider one pixel (u, v) in image of center camera O , according to the intrinsic parameters

of the camera, a vector P in 3D space could be obtained. All the possible stereo matching point to (u, v) in image of subcamera O' should lie in the curve ℓ : the projection of vector P into subcamera O' .

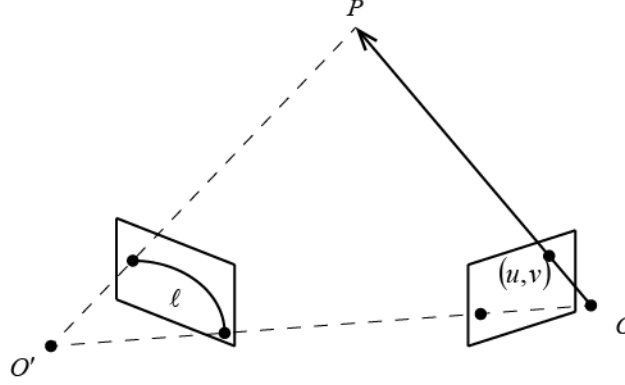


Fig. 4.2 Formation of Epipolar curve

However, due to the limitation of camera model, the analytical expression of the curve ℓ with respect to point (u, v) is unavailable. So we turn to numerical methods: using discrete points to represent the curve ℓ according to point (u, v) . In detail, some discrete 3D points are sampled on P and projected (see detail in 2.1) to the image of subcamera O' to generate curve ℓ .

Here rises another problem: how to choose the discrete points in vector P to make there projection points in curve ℓ as even as possible?

According to parallel stereo, there is a relation between disparity (d) and inverse depth ($1/Z$):

$$d = BF \frac{1}{Z} \quad (4.1)$$

Here B means the length of baseline and F is the focal length.

Although our camera system is not parallel stereo camera system, (4.1) still guides us that we should choose the search of depth with arithmetic sequence in inverse depth:

$$\frac{1}{z_k} - \frac{1}{z_{k+1}} = c \quad (4.2)$$

Let Z_{min} and Z_{max} be the minimum and maximum of the sequence of depth respectively and n be the resolution of depth, i.e., the number of series of depth, thus

$$z_k = \frac{1}{\frac{1}{Z_{max}} + \frac{k-1}{n-1} \cdot \left(\frac{1}{Z_{min}} - \frac{1}{Z_{max}} \right)} \quad k = 1, 2, \dots, n \quad (4.3)$$

Equation (4.3) formulate the discrete point in depth search sequence, which is depend on three factors: Z_{min} , Z_{max} , n .

n could be simply decided by the resolution of image, for example with the resolution of 1600X1200 of the whole image, after divided into 5 sub images, each sub image has the resolution about 400X300, then it is reasonable to choose $n = 300$ since those projected points is approximately equally spaced.

As shown in Fig. 4.3, with the increase length of P , the projection point of P in subcamera O' changes. However, the projection point would finally stop changing at the Vanish Point when P reaches infinity. So the nearest integer pixel around vanish point is used to estimate maximum depth.

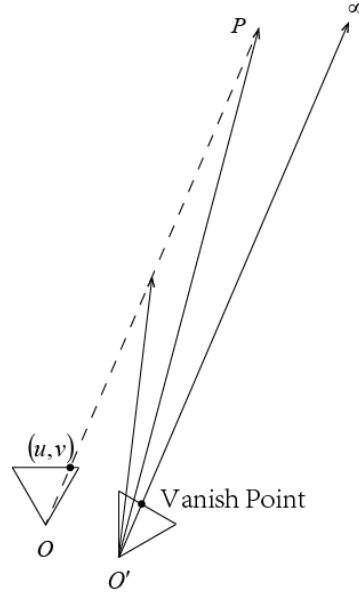


Fig. 4.3 The Vanish Point to estimate the maximum depth

After calculation, the maximum depth of our camera system is about 3m, which guides us to choose the proper value of Z_{max} . As to Z_{min} , 10 cm is usually used.

So far are the details in epipolar curve, the image is unchanged while the curves are used as the epipolar constraint.

4.3 Comparison of Perspective Reprojection with Epipolar Curve

Now two methods are available now, and we want to choose a better one for our future research, so Table 4.1 is made to compare the advantages and disadvantages of these two approaches:

	Perspective Reprojection	Epipolar Curve
Advantages	Distortion are moved;	Much lighter computation; Flexible;
Disadvantages	Heavy computation; Not parallel stereo so still need epipolar constraints;	Distortion still exists;

Table 4.1 Comparison between Perspective Reprojection and Epipolar Curve

As to distortion, perspective reprojection does not have such problem since the whole images are rectified, while epipolar curve still face the problem of distortion, which might be a problem in stereo matching since the content in a matching window is distorted. However, our novel approach of stereo matching in disparity space is free from lens distortion in nature (presented in Section 5.2.3).

When it comes to the cost of computation, the epipolar curve approach has huge advantage not only because relative lighter computation but also the key fact that all those curves need to be computed only once. In practice, the strategy of saving time by space could be utilized: storing those curves in memory and free from the computation of projection of curves in each time. While the perspective reprojection does not hold such merits: the whole image are rectified in each time and the computation cost is heavy.

Last but not least, after perspective reprojection, the rectified images still does not hold the direct constraint:

$$f_1(x, y) = f_2(x + d(x, y), y) \quad (4.4)$$

The fact Equation (4.4) does not hold implies further rectification is need with heavier computation cost.

Taking all these merits and demerits into consideration, we would prefer epipolar curve as the rectification method and utilize it in stereo matching in the next chapter.

Chapter 5

Multi-Baseline Stereo

With the foundation of accurate calibration result and rectification method of epipolar curve, stereo matching is obtainable now. Stereo matching is one the most active research areas in computer vision, a large number of algorithms for stereo correspondence have been developed. According to the taxonomy and the test bed of Middlebury in [8], there are more than 100 algorithms related to this topic.

Due to the special feature: more than one baseline of our camera system, multi-baseline stereo [4] is appropriate, so we first derive the mathematical analysis of multi-baseline stereo in our camera system. Then a general approach is present. Finally we learn the idea from [5], which is the top performer in the Middlebury benchmark now (May 2012), to implement the multi-baseline stereo in disparity space, which is much faster and more flexible. What's more, the disparity space approach holds the merit of free from distortion.

5.1 General Approach

In [4], the author derives the mathematical analysis of multi-baseline stereo in parallel stereo. However our camera system is not parallel stereo camera system so here we re-derive the mathematical properties of multi-baseline stereo in our cases and then provides a practical way to implement it.

5.1.1 The SSD function

In our camera system, there are five equivalent cameras at different position, so let us suppose their position as P_1, P_2, P_3, P_4, P_5 according to the index defined in Fig. 3.1.

Let $f_5(x)$ and $f_i(x)$ $i = 1, 2, 3, 4$ be the image pair taken by the center real camera and the i^{th} subcamera respectively. Imagine a scene point Z whose true depth and coordinate in center image is z^r and (u, v) respectively. Then point Z 's disparity d for the image pair taken from P_5 and P_i is

$$d_{(u,v)}^r(i) = (u, v)^T - m \left(Ab^i \cdot z^r \cdot \frac{(u, v, f(u, v))^T}{\|(u, v, f(u, v))\|} \right) \quad (5.1)$$

where Ab^i is the transformation from center real camera to subcamera i (shown in Fig. 3.1). $m(\dots)$ means the projection from 3D space to image plane (see detail in Section 2.1).

The image intensity function $f_5(x)$ and $f_i(x)$ near the matching position for Z could be expressed as

$$f_5(u, v) = f(u, v) + n_5(u, v) \quad (5.2)$$

$$f_i(u, v) = f \left((u, v) - d_{(u,v)}^r(i) \right) + n_i(u, v) \quad (5.3)$$

Assuming that $f(u, v)$ is true intensity of the image without noise and the white noise $n_5(u, v)$, $n_i(u, v)$ obey such distribution independently:

$$n_5(u, v), n_i(u, v) \sim N(0, \sigma_n^2) \quad (5.4)$$

The SSD value $e_{d(i)}$ over a window W at a pixel position of (u, v) of image $f_5(u, v)$ for the candidate disparity $d_{(u,v)}(i)$ is defined as:

$$e_{d(i)}((u, v), d_{(i)}) \equiv \sum_{j \in W} \left(f_5((u, v) + j) - f_i((u, v) + d_{(u,v)}(i) + j) \right)^2 \quad (5.5)$$

where the $\sum_{j \in W}(\dots)$ means summation over the window.

The $d_{(u,v)}(i)$ that gives a minimum of $e_{d(i)}((u, v), d_{(u,v)}(i))$ is determined as the estimation of the disparity at (u, v) . Since the SSD measurement $e_{d(i)}((u, v), d_{(u,v)}(i))$ is a random variable, its expected value could be computed:

$$\begin{aligned} E \left[e_{d(i)}((u, v), d_{(u,v)}(i)) \right] \\ = \sum_{j \in W} \left(f((u, v) + j) - f((u, v) + d_{(u,v)}(i) - d_{(u,v)}^r(i) + j) \right)^2 \\ + 2N_w \sigma_n^2 \end{aligned} \quad (5.6)$$

here N_w is the number of pixels inside the window.

By assuming the true disparity $d_{(u,v)}^r(i)$ is constant over the window, (5.6) says that naturally the SSD function $e_{d(i)}((u, v), d_{(u,v)}(i))$ is expected to take a minimum when $d_{(u,v)}(i) = d_{(u,v)}^r(i)$, i.e. at the right disparity or depth.

5.1.2 Elimination of Ambiguity by SSSD function

First let us show how the SSD function $e_{d(i)}((u, v), d_{(u,v)}(i))$ fails when there is ambiguity in the underlying intensity function. Suppose the intensity signal $f(u, v)$ has the same pattern around pixel positions (u, v) and $(u, v) + a$ in a window W

$$f((u, v) + j) = f((u, v) + a + j), \quad j \in W \quad (5.7)$$

where $a \neq 0$ is a constant. Then from equation (5.6)

$$E \left[e_{d(i)}((u, v), d_{(u,v)}(i)) \right] = E \left[e_{d(i)}((u, v), d_{(u,v)}(i) + a) \right] = 2N_w \sigma_n^2 \quad (5.8)$$

This means that the ambiguity is expected in matching in terms of position of minimum SSD value, so SSD function fails in stereo matching when ambiguity arises.

Now we rewrite equation (5.1) into a function form:

$$d_{(u,v)}^r(i) = D((u, v), Ab^i, z^r) \quad (5.9)$$

Similarly:

$$d_{(u,v)}(i) = D((u, v), Ab^i, z) \quad (5.10)$$

Where z^r and z are the true and candidate depth, respectively. We write the SSD with respect to the depth by substituting (5.10) into (5.5):

$$e_{z(i)}((u, v), z) \equiv \sum_{j \in W} \left(f_5((u, v) + j) - f_i((u, v) + D((u, v), Ab^i, z) + j) \right)^2 \quad (5.11)$$

At position (u, v) for a candidate depth z , its expected value is

$$\begin{aligned}
& E[e_{z(i)}((u, v), z)] \\
&= \sum_{j \in W} \left(f((u, v) + j) - f\left((u, v) + D((u, v), Ab^i, z - z^r) + j\right) \right)^2 \\
&+ 2N_w \sigma_n^2
\end{aligned} \tag{5.12}$$

Finally, we define a the SSSD evaluation function $e_{z(12...n)}((u, v), z)$, the sum of SSD functions with respect to depth z for multiple pairs, which is obtained by adding the SSD function $e_{z(i)}((u, v), z)$ for individual stereo pairs:

$$e_{z(12...n)}((u, v), z) = \sum_{i=1}^n e_{z(i)}((u, v), z) \tag{5.13}$$

The expected value for (5.13) is:

$$\begin{aligned}
& E[e_{z(12...n)}((u, v), z)] \\
&= \sum_{i=1}^n E[e_{z(i)}((u, v), z)] \\
&= \sum_{i=1}^n \sum_{j \in W} \left(f((u, v) + j) \right. \\
&\quad \left. - f\left((u, v) + D((u, v), Ab^i, z - z^r) + j\right) \right)^2 + 2nN_w \sigma_n^2
\end{aligned} \tag{5.14}$$

Now let us show how SSSD function is now capable of eliminating the ambiguity.

As done before, suppose the underlying intensity pattern $f(u, v)$ has the same pattern around (u, v) and $(u, v) + a$ (equation (5.7)).

$$E[e_{z(i)}((u, v), z^r)] = E[e_{z(i)}((u, v), z^f)] = 2N_w\sigma_n^2 \quad (5.15)$$

where the false depth z^f satisfies:

$$f((u, v) + j) = f\left((u, v) + D\left((u, v), Ab^i, z^f - z^r\right) + j\right), j \in W \quad (5.16)$$

Ambiguity still exist now since a minimum is expected at a false depth z^f . However, an important point to be observed here is that this minimum for the false depth z^f changes its position as the transformation Ab^i changes, while the minimum for the correct depth z^r does not.

This is the property makes new SSSD evaluation function eliminates the ambiguity. For example, in our camera system, we have four subcameras thus we have four transformation matrixes (i.e. four baselines): Ab^i $i = 1, 2, 3, 4$. ($Ab^1 \neq Ab^2 \neq Ab^3 \neq Ab^4$). From equation (5.14)

$$\begin{aligned} & E[e_{z(1234)}((u, v), z)] \\ &= \sum_{i=1}^n E[e_{z(i)}((u, v), z)] \\ &= \sum_{j \in W} \left(f((u, v) + j) - f\left((u, v) + D((u, v), Ab^1, z - z^r) + j\right) \right)^2 \\ &+ \sum_{j \in W} \left(f((u, v) + j) - f\left((u, v) + D((u, v), Ab^2, z - z^r) + j\right) \right)^2 \\ &+ \sum_{j \in W} \left(f((u, v) + j) - f\left((u, v) + D((u, v), Ab^3, z - z^r) + j\right) \right)^2 \\ &+ \sum_{j \in W} \left(f((u, v) + j) - f\left((u, v) + D((u, v), Ab^4, z - z^r) + j\right) \right)^2 \\ &+ 8N_w\sigma_n^2 \end{aligned}$$

(5.17)

It could be proved that

$$E[e_{z(1234)}((u, v), z)] > 8N_w\sigma_n^2 = E[e_{z(1234)}((u, v), z^r)] \text{ for } z \neq z^r \quad (5.18)$$

In conclusion, $e_{z(1234)}((u, v), z)$ is expected to have the smallest value only at the correct z^r , in other words, the ambiguity is now eliminated by the SSSD function.

5.1.3 Implementation in our camera system with epipolar curve

After the mathematical analysis of the properties of multi-baseline stereo, here we show a practical way to implement it.

The side view of our camera system is shown in Fig. 5.1 and Fig. 5.2, for a pixel (u, v) in center camera, according to the calibration result of intrinsic parameters, a vector P in 3D space is available. Then the candidate depth z_k in equation (4.3) could be computed, those possible depth points were projected into all the four subcameras, as explained in Section 4.2, four epipolar curves $\ell_i, i = 1, 2, 3, 4$ is obtained now.

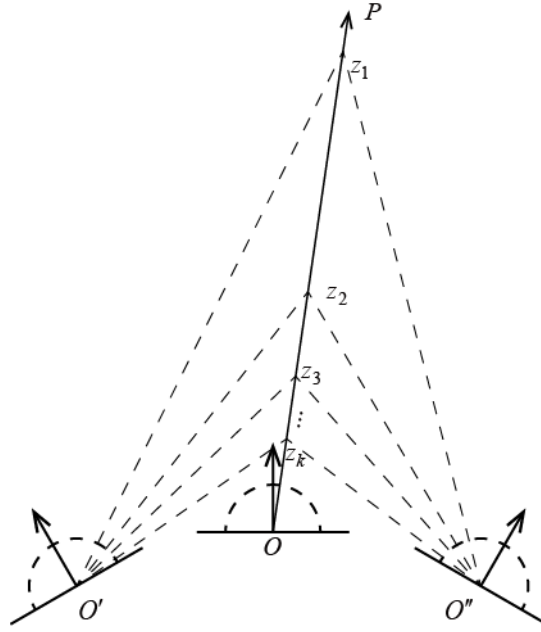


Fig. 5.1 The Process of depth searching in slide of the system

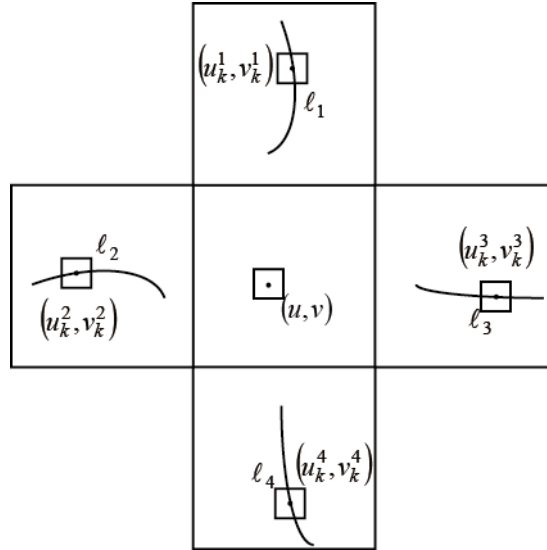


Fig. 5.2 Epipolar Curves and matching window in image plane

At each candidate depth z_k , its projected points in four sub-images are denoted by (u_k^i, v_k^i) , $i = 1, 2, 3, 4$. Then four SSD values (5.11) between point (u, v) and

(u_k^i, v_k^i) are computed, then the summation of these four SSD values is the SSSD value (5.13).

In computing the SSD functions by window, the ‘mirror effect’ (Fig. 1.2) should be taken care of. Fig.5.3 below describes the proper correspondence between center image and sub-image within a window:

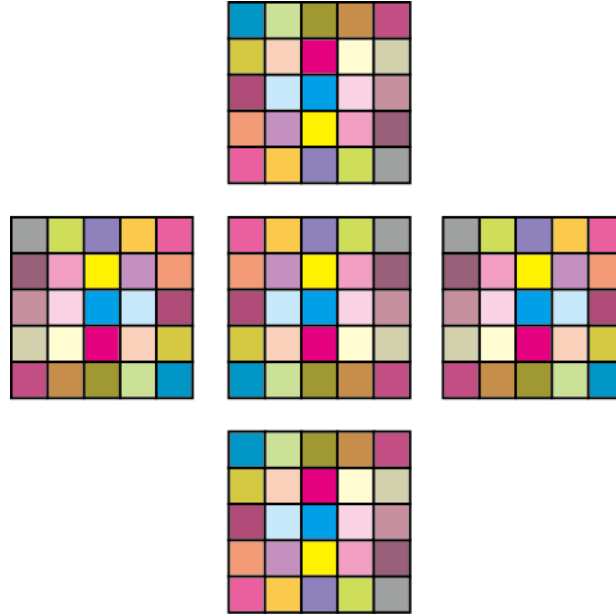


Fig. 5.3 The window correspondence under Mirror Effect

By finding the minimum SSSD value among all the candidate depth z_k , the estimated depth is computed. Finally, by traverse all the pixels in the center image, repeating previous steps each time, the depth map is calculated.

5.2 Fast Implementation in Disparity Space

Though the general approach of multi-baseline (Section 5.1.3) is clear and direct to implement, it holds such defects:

- Suffer from both perspective distortion and lens distortion;
- Low computation efficiency caused by redundant computation;
 - With the size of matching window increase, the computation time increase;
 - Difficult to realized by parallel computation;

To make the camera system moving means we have to find an efficient algorithm to get the depth map around real time, to this end, a parallel algorithm in stereo matching, which is implemented in disparity space is presented.

5.2.1 The Formation of Disparity Space

In general, the disparity space (Fig. 5.4) is a 3 dimension space with two dimensions of position in images and one dimension in disparity. In each cell $C(p, d)$, the difference in intensity of pixel p at disparity d is stored.

As shown in (5.10), under a certain transformation from center camera to i^{th} subcamera, the disparity at pixel (u, v) is a function of depth. So similar to (5.5) before aggregation within the window:

$$C_i(p, d) = f_5(u, v) - f_i\left((u, v) + D\left((u, v), Ab^i, z\right)\right) \quad (5.19)$$

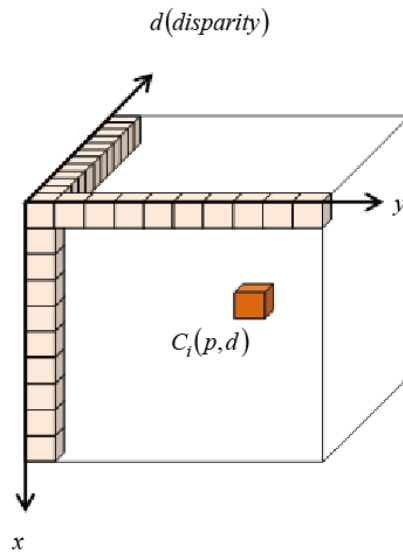


Fig. 5.4 The Disparity Space

5.2.2 Multi-Baseline Stereo in Disparity Space

Once the disparity space is generated, the OII technique in [5] and [6] could be simplified to get the SSSD value as described in section 5.1.2.

As shown in Fig.5.5 (a), we need to aggregate the local aggregation region $U_d(p)$: the square window in red. Let $H_d(p)$ and $V_d(p)$ be the horizontal and vertical support region of pixel p , respectively. Note the fact that the aggregation of the whole support region is equivalent to aggregating the support region horizontally followed by aggregating vertically (Fig.5.5 (b), (c)).

$$U_d(p) = \bigcup_{q \in V_d(p)} H_d(q) \quad (5.20)$$

In details,

$$E_d(p, d) = \sum_{s \in U_d(p)} e_d(s, d) = \sum_{q \in V_d(p)} \left(\sum_{s \in H_d(q)} C_i(s, d) \right) = \sum_{q \in V_d(p)} E_d^H(q, d) \quad (5.21)$$

where $E_d^H(q, d)$ represents the result after the horizontal integration step.

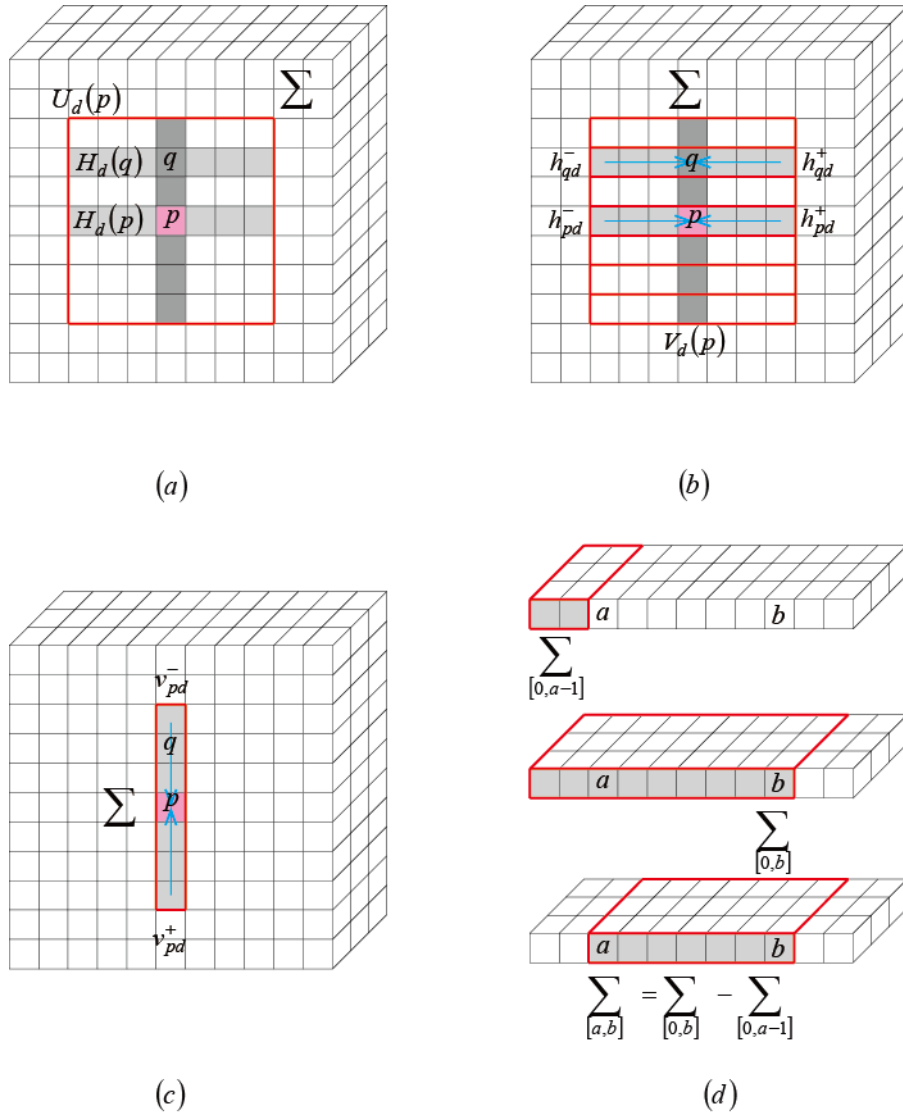


Fig. 5.5 Illustration of the OII technique

Assume a local window has the size of $(2 \cdot lw + 1) \times (2 \cdot lw + 1)$, then the overall OII technique achieves this aggregation goal in four steps:

Step 1: Given the pixelwise raw matching cost $C_i((x, y), d)$, we first build a horizontal integral image $S^H((x, y), d)$, storing the cumulative row sum as

$$S^H((x, y), d) = \sum_{0 \leq m \leq x} C_i((m, y), d) = S^H((x - 1, y), d) + C_i((x, y), d) \quad (5.22)$$

Here an efficient way to get $S^H((x, y), d)$ is by iteratively computation from $S^H((x - 1, y), d)$. In real practice, $S^H((x, y), d)$ could be formulated by one loop horizontally. Note that when $x = 0$, $S^H((-1, y), d) = 0$.

Step 2: For each pixel $q = (x_q, y_q)$ on the vertical region of p , we then compute the horizontal integral $E_d^H(q, d)$ in (5.21), using the horizontal integral image $S^H((x, y), d)$ obtained in step 1:

$$\begin{aligned} E_d^H(q, d) &= S^H((x_q + h_{qd}^+, y_q), d) - S^H((x_q - h_{qd}^- - 1, y_q), d) \\ &= S^H((x_q + lw, y_q), d) - S^H((x_q - lw - 1, y_q), d) \end{aligned} \quad (5.23)$$

As shown in Fig. 5.5 (d), h_{qd}^- and h_{qd}^+ are the left and right length of the horizontal region, since in our approach, a size fixed window is applied, so $h_{qd}^- = h_{qd}^+ = lw$.

Step 3: Taking the computed horizontal matching cost $E_d^H(p, d)$ as the new input, similar to what has done in step 1, a vertical integral image $S^V(p, d)$, which stores the cumulative column sum, could be obtained as:

$$S^V((x, y), d) = \sum_{0 \leq n \leq y} E_d^H((x, n), d) = S^V((x, y - 1), d) + E_d^H((x, y), d) \quad (5.24)$$

As explained in step 1, an iteratively computation is efficient and also when $y = 0$, $S^V((x, -1), d) = 0$

Step 4: Based on the vertical aggregated image $S^V(p, d)$, we finally derive the fully aggregated matching cost $E_d(p, d)$ for pixel $p = (x_p, y_p)$ with similar subtraction as Step2:

$$\begin{aligned} E_d(p, d) &= S^V\left((x_p, y_p + v_{pd}^+), d\right) - S^V\left((x_p, y_p - v_{pd}^- - 1), d\right) \\ &= S^V\left((x_p, y_p + lw), d\right) - S^V\left((x_p, y_p - lw - 1), d\right) \end{aligned} \quad (5.25)$$

v_{pd}^+ and v_{pd}^- are up and bottom length of the vertical region, since a square window is applied, so $v_{pd}^+ = v_{pd}^- = lw$.

Up to now, the SSD value in disparity space is obtained. With four pairs of images, four aggregated disparity spaces E_d^i , $i = 1, 2, 3, 4$ could be calculated. Because all these E_d^i are computed with the same base: the center image and have the same dimension (shown in Fig. 5.6), it is nature to sum them up to get the SSSD value:

$$E_d^{1234} = \sum_{i=1}^4 E_d^i \quad (5.26)$$

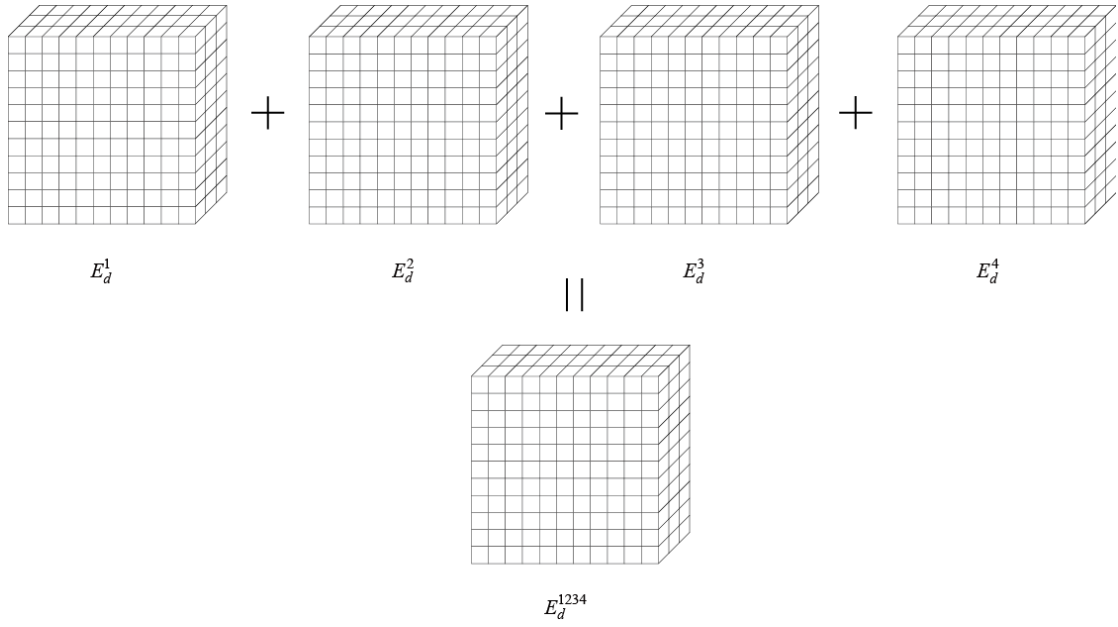


Fig. 5.6 Getting SSSD values from SSD values in aggregated disparity space

The final step is to find the minimum SSSD value in disparity dimension for each pixel to get the proper disparity; meanwhile, the depth is obtained. Applying this final step to all pixels in E_d^{1234} would a depth map be formed.

5.2.3 Merits of the Disparity Space Approach

Following the above steps in section 5.2.1 and 5.2.2, the multi-baseline stereo is implemented in disparity space. In general, this disparity space approach holds such two merits:

- Fast rapid speed in computation:

Firstly, let us notice the fact that all the disparity values $D((u, v), Ab^i, z)$ in (5.10), which are equivalent to the epipolar curve ℓ_i (Fig. 5.2), need to be computed only

once. Although with the moving of camera, the image f_i it captured would change, while the relative transformation between subcameras would not. So a strategy of computing and storing all these epipolar curves could be applied to cut off the computation time.

Secondly, as stated in [5], this algorithm is parallel in nature (as shown in Fig. 5.5 and Fig. 5.6), so it could be applied on GPU with CUDA implementations. What's more, even with CPU computation, it also outperforms the general approach with its compact data structure.

Last but not least, this algorithm holds an interesting property in computation that with the increase in window size, the computation is almost the same; however the general approach would show notable increase in computation time with increased size of window.

- Free from distortion

Fig. 5.7 shows how disparity space approach is naturally free from distortion. The object has the depth z at pixel (u, v) in center camera O . Those two blue lines around (u, v) refer to a fixed sized window. z^- and z^+ are the depth points of the object related to the window around (u, v) respectively. (u', v') and (u'', v'') are the projection of point z in camera O' and O'' . Points z^- and z^+ are also projected into these cameras individually in green lines.

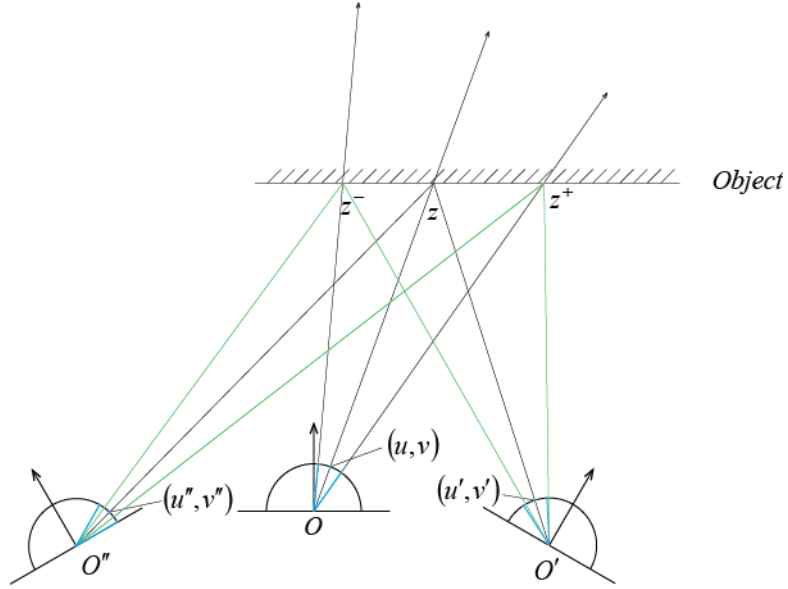


Fig. 5.7 Free from distortion by Disparity Space approach

By referring to the difference of blue window and green projected points, in general approach, these two point sets are usually different, so the general approach suffers from the problem of distortion.

When it comes to the disparity approach, the actual windows aggregated in subcameras are those projected points from z^- and z^+ . (in epipolar curves), rather than a fixed sized window around (u', v') and (u'', v'') . Assume the depth of object does not change, then we are comparing the true projections of point z, z^- and z^+ , so the aggregated SSD value is free from distortion, in both perspective distortion and lens distortion.

With the property of free from distortion, together with the property that the window size has no influence on the computation cost, a window with large size is suggested, which may conclude more local details and help eliminating the mismatching.

Chapter 6

Experimental Results

In this section, we show the experimental results of calibration, rectification and multi-baseline stereo, respectively. In the calibration part, both synthetic data and real system experiments are performed to test the validation of our proposed method. We show the improvement in proposed linear calibration by comparison with the baseline method directly derived from Davide’s Ocam-calib Toolbox [2], [3]. Both rectification results of perspective reprojection and epipolar curves are shown not only to provide foundation for stereo-matching but also to prove the accuracy of calibration in another aspect. Finally, the 3D reconstruction is done by multi-baseline stereo in disparity space. We show that the depth map with high quality could be obtained in seconds. With the depth map available, the well-known ICP algorithm [20] is applied to align those depth points to do the 3D reconstruction with a moving camera.

6.1 Calibration Results

6.1.1 Simulation Experiment

To test the performance of our proposed calibration method, we set up the Simulation Experiment with 5 cameras with 6 shots with the extrinsic parameters shown in Fig. 6.1. As a natural extension, we set different intrinsic parameters to model and calibration results of real omnidirectional cameras according to Table 6.1. We set a chessboard with $6 \times 8 = 48$ corners with the size 24mm in each corner.

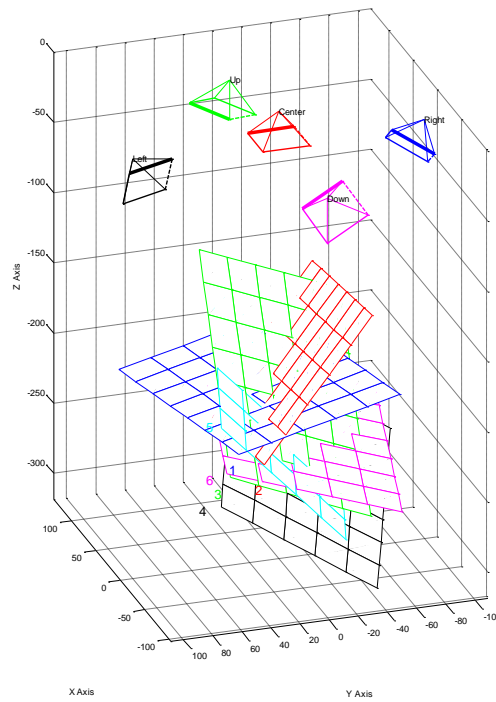


Fig. 6.1 A picture of our simulation experiment to show the setting of extrinsic parameters

Camera 1	Camera 2	Camera 3	Camera 4	Camera 5
				

Table 6.1 The setting of intrinsic parameters is according to the calibration results of these real omnidirectional camera

The robustness of our calibration technique, in case of inaccuracy in detection the calibration points, could be studied by adding Gaussian noise (with deviation σ) to

the true projected points. Then our calibration method is performed on those corrupted point to show the validation and robustness of the algorithm.

Fig. 6.2 plots the reprojection error vs. different noise level σ . The noise level is varied from $\sigma = 0$ pixels to $\sigma = 3.0$ pixels with $\Delta\sigma = 0.1$ pixels in increase. As usual, we define the reprojection error as the distance, in pixels, between back-projected 3D points and corrupted image point. As shown in the figure, the average error increases linearly with the noise level increase in both results of linear initialization and nonlinear refinement. Observe that the nonlinear refinement result is always better than that in the linear method.

Usually $\sigma = 1.0$ pixel is the maximum noise in practical calibration, which means our proposed would obtain around 0.4 pixels in accuracy.

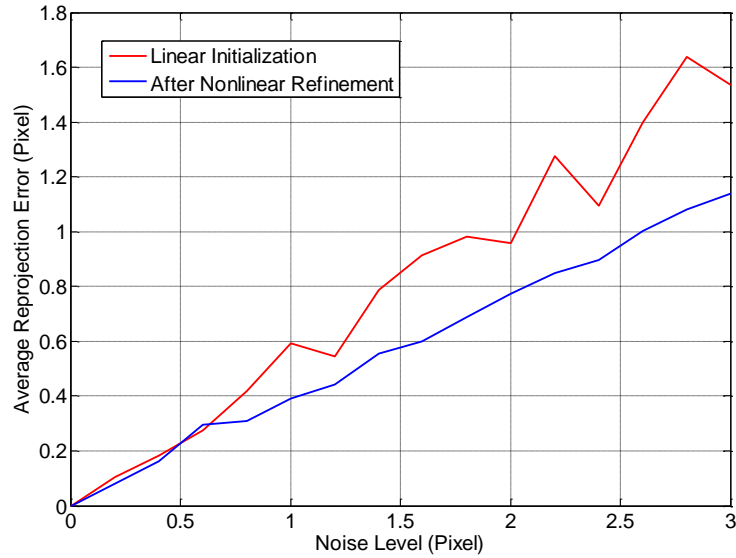


Fig. 6.2 The reprojection error vs. the noise level with both linear and nonlinear methods

In figure 6.3, we test the accuracy of estimation of translation from center camera to subcameras, i.e., the absolute error in vector \mathbf{b} (equation 3.3), after nonlinear refinement. The absolute error is very small because even with the noise level

$\sigma = 3.0$, it is still less than 0.5 mm. What's more, the errors in $\mathbf{A}, \mathbf{R}, \mathbf{t}$ are also very small which show the validation of our calibration method.

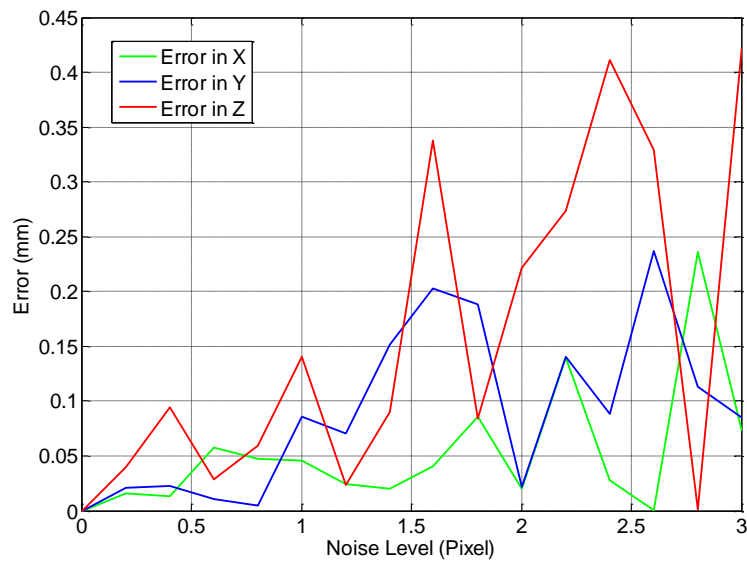


Fig.6.3 Accuracy of the extrinsic parameters of translation from center camera to subcameras in average

6.1.2 Real Experiment

Following the steps mentioned in Chapter 3, to make the calibration process easy and convenient, a Matlab Toolbox was developed, which implements the calibration method for our Monocular Multi-view Camera system.

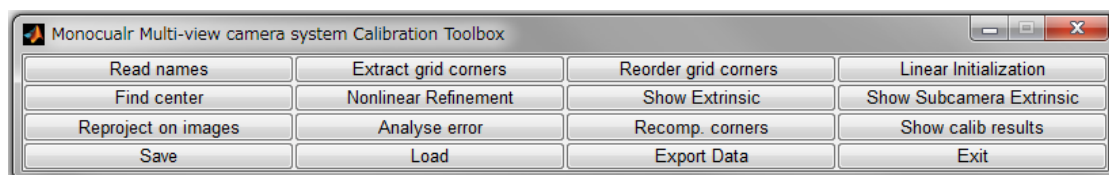


Fig. 6.4 The Monocular Multi-view camera system Calibration Toolbox

This toolbox (Fig. 6.4) is tested on our real camera system, and the camera has the resolution of 1600×1200 . We took 6 shots, so after separation, 30 images are

used in calibration. The total 48 corner points are detected by Harris corner detector having sub-pixel accuracy.

A. The advantage of free from misleading

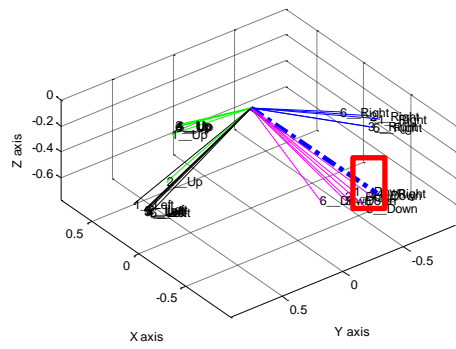
In our previous work, we simply extended Jiang’s calibration method [1] from using single shot to multiple shots by taking the average of the normal vector of mirrors and the distance between real camera and mirrors. Then we utilized the nonlinear refinement by Bundle Adjustment mentioned in previous section. This previous method shows that single shot usually does not provide reasonable precision. However, it suffers from the inconvenience of manually pick out the misleading.

In detail, we used Davide’s Toolbox [3], [4] to calibrate the intrinsic parameters of the camera and the position of chessboard (extrinsic parameters), and utilize the position of chessboard to calculate the position of mirrors then get the relative space transformation between subcameras [1]. However, because Davide’s Toolbox only focuses the intrinsic parameters (the position of chessboard is a by-product) and the size of chessboard in our images are relative small (we have to put five chessboards in one shot), so there might be some misleading in estimate the position of each chessboard. Those misleading may corrupt the initial estimation of the relative space transformation between subcamera, and then lead to local-minimum in Bundle Adjustment, which is not desirable in calibration.

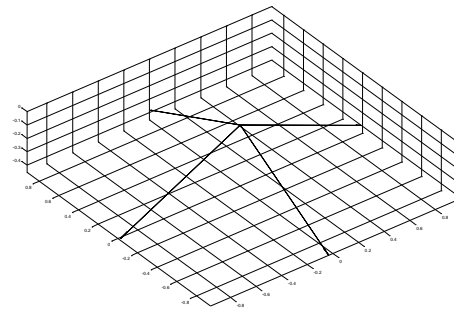
In our proposed method, the iteration process in solving the rotation matrixes naturally prevents such misleading: the relative space transformation between subcameras are considered in all shots, meanwhile the position of chessboard in each shot is evaluated by all images taken by all subcameras. The key improvement is that we utilize those well estimated rotation matrixes by mean rotation at beginning instead of estimate those rotation matrixes in all shots separately then take the mean of them at last.

As shown in Fig. 6.5 (a), we separately calculate and draw the normal vector of 4 mirrors (in different colors) in all images, the difference between each mirrors' vectors shows the necessary of using multiple images during calibration. A wrong estimation of the right subcamera (see the blue camera in Fig. 6.5 c) is caused by one misleading vector: one blue thick dashed vector differs much from its group (highlighted by a red rectangle).

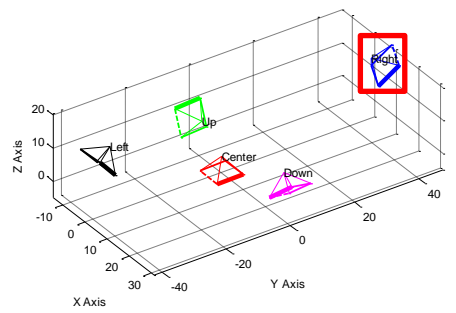
However, our proposed method does not suffer from such problem since with the same input data since the normal vectors of mirrors are obtained correctly (Fig. 6.5 b). The final result (Fig. 6.5 d) of the linear calibration outperforms the previous result (Fig. 6.5 c). After Bundle Adjustment, the proper initialized case would obtain 0.62 pixels in average reprojection error while the bad initialized one gets 0.73 pixels in average reprojection error. Fig.6.5 (e) and (f) shows the changing of the relative position of subcameras in each iteration, which shows that our proposed method provides a better initialization, especially those rotation matrixes.



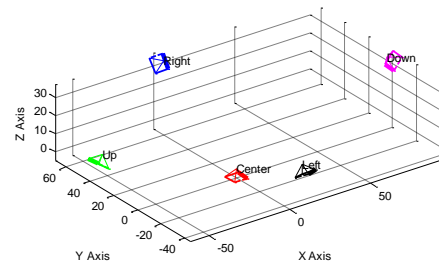
(a)



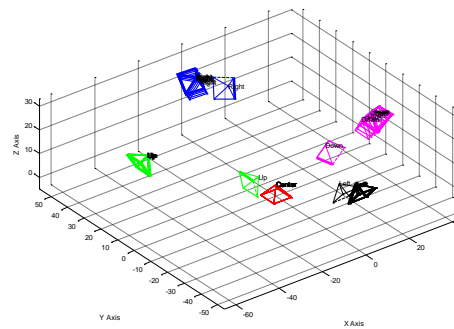
(b)



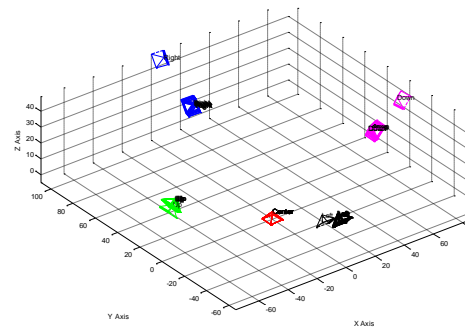
(c)



(d)



(e)



(f)

Fig. 6.5 Comparison of the proposed method vs. previous method

Fig. 6.6 plots the average reprojection errors in each iteration, from which we could observe that the calibration precession with previous initialization in 10th iteration is just the same as the one with proposed initialization in 1st iteration.

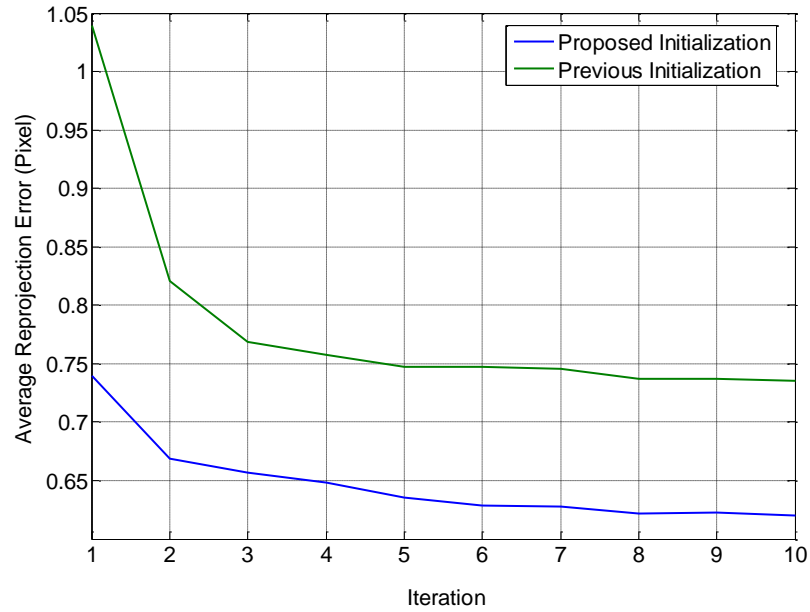


Fig. 6.6 The reprojection error vs. the iteration time with different initialization: Blue and green lines represent the proposed method and previous method respectively

B. The Overall evaluation of calibration

Let us first evaluate the overall performance of our calibration technique in case of inaccuracy in detecting the calibration points. To this end, in Fig. 6.7 we show the 3D points of a chessboard back-projected onto the image. Those blue circles represent the projected points and red crosses are obtained by Harris corner detector. In fact, those projected points match the image better than those detected points sometimes.

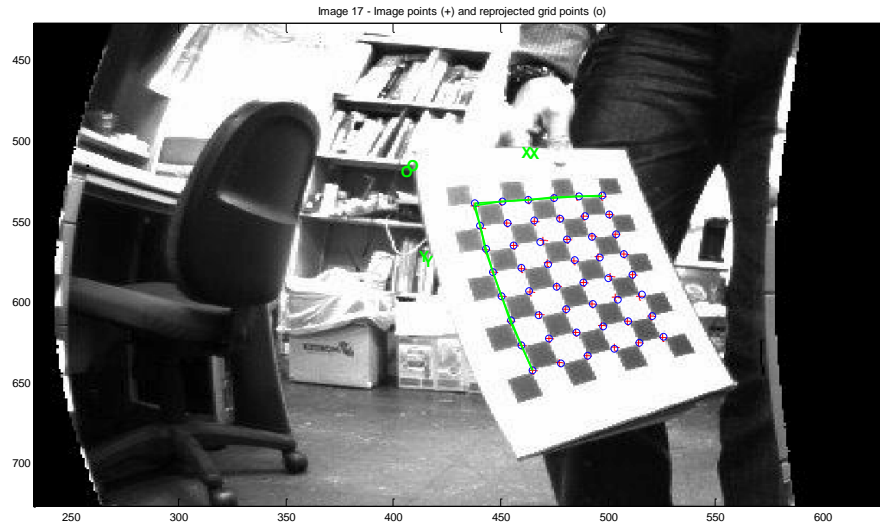


Fig. 6.7

Fig. 6.7 Reprojection Points in Real Image

Then in Fig. 6.8 is shown the distribution of average reprojection error in each shot. From which majority of the reprojection errors are sub-pixel and those large reprojection errors happen in few images since they are in the same color.

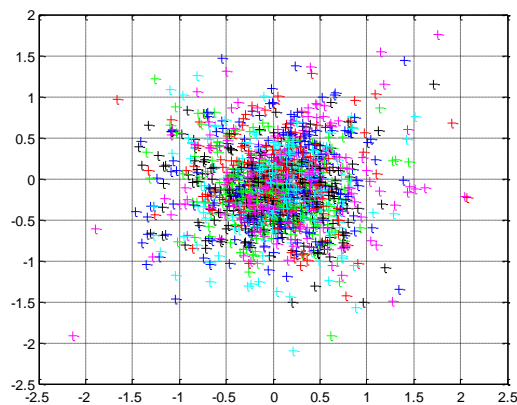


Fig. 6.8

Fig. 6.8 The distribution of average reprojection error in each shot

6.2 Rectification Results

Although only the epipolar curve is used in future study, here we show both results in perspective reprojection and epipolar curve.

6.2.1 Perspective Reprojection Result

As described in Section 4.1, with the calibration results, origin image taken from the camera system Fig. 6.9 is rectified and the rectification result is shown in Fig. 6.10. With the perspective reprojection, the distortion from lens is rectified. And now the camera system is equivalent to parallel stereo with an anteroposterior offset, which means the direct epipolar constraint (equation 4.4) does not hold (shown as the disagreement in matching in the red horizontal or vertical lines). So epipolar lines or fundamental matrix are needed in addition.

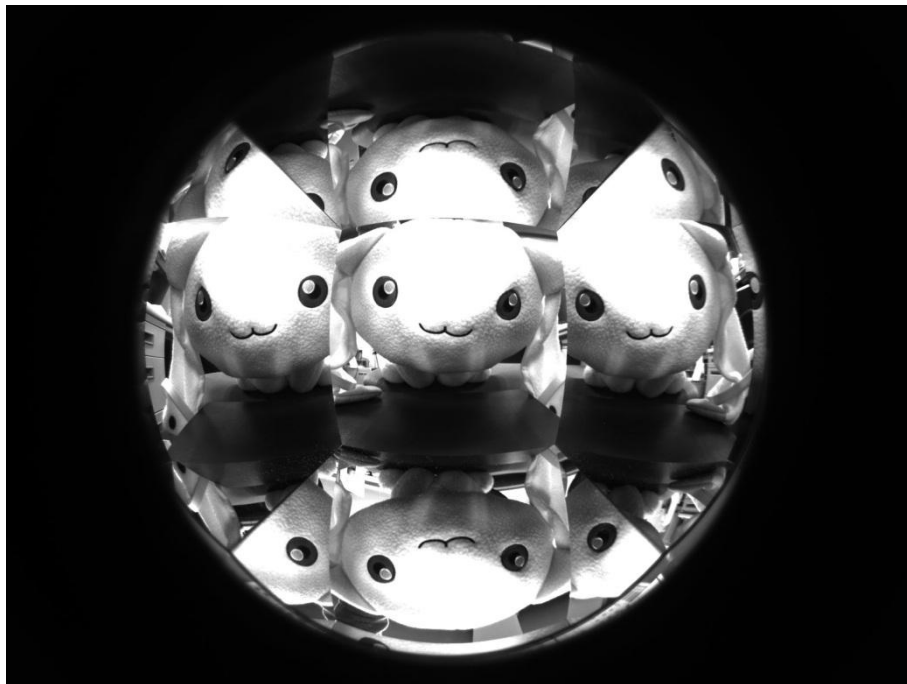


Fig. 6.9 The origin image taken by the monocular multi-view camera system

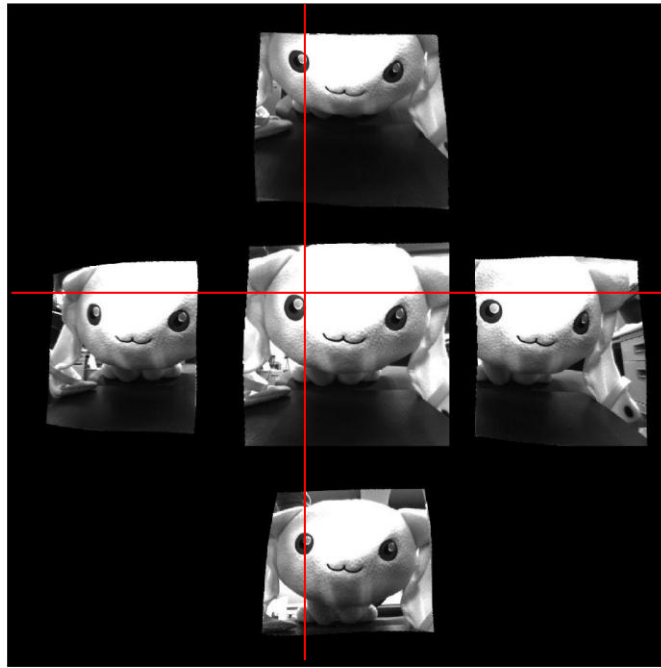


Fig. 6.10 The rectification result of perspective reprojection

6.2.2 Epipolar Curve Result

Fig. 6.11 shows the result of epipolar curve constraint. These four epipolar curves correspond to the red cross in center image. For one thing, these epipolar curves provide foundation for stereo matching in future study, for another, they prove the effectiveness of calibration from side aspect.

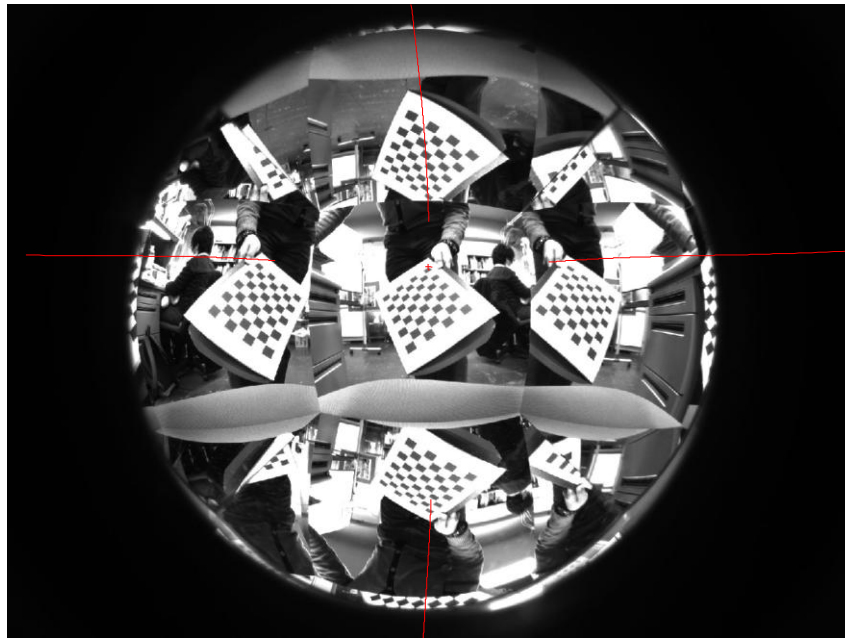


Fig. 6.11 The Epipolar curves

6.3 Multi-Baseline Stereo Results

Finally, multi-baseline stereo is applied to get the depth map with single shot. We make comparison between the general approach and the disparity space approach, mainly to illustrate the effectiveness of the disparity space approach.

6.3.1 General Approach

Following the procedures described in Section 5.1.3, we develop the depth map from the input image (Fig. 6.12) with the resolution of 1600X1200. With different size of local window, the result of depth maps are shown in Fig. 6.13.

From the different results of depth map, it could be concluded that with a larger local window, the depth map would be smoother (with fewer outliers), however, some details (such as the hands behind the bear) might be lost.

Another property of this general approach is the computation time, from Fig. 6.14, with the increase in size of local window, the computation time increases significantly. What's more, even with a small local window, it still need about 3 minutes to generate a depth map, which is unacceptable slow in application of the moving camera.

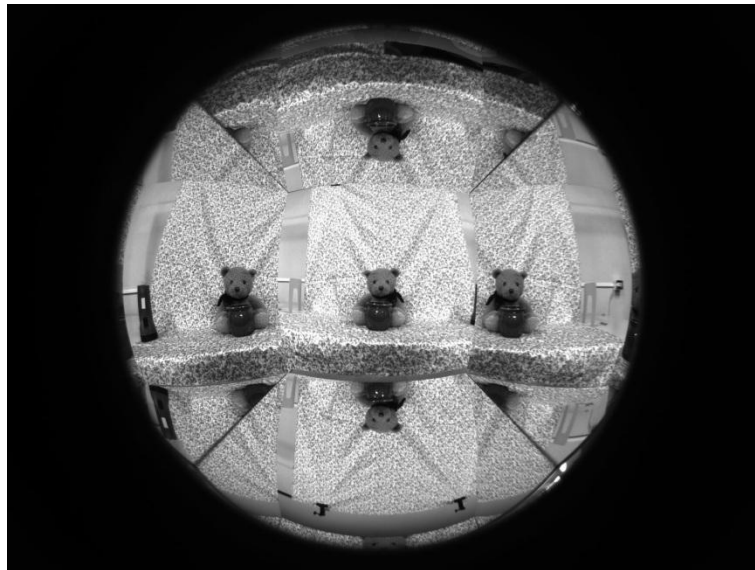


Fig. 6.12 The input Image

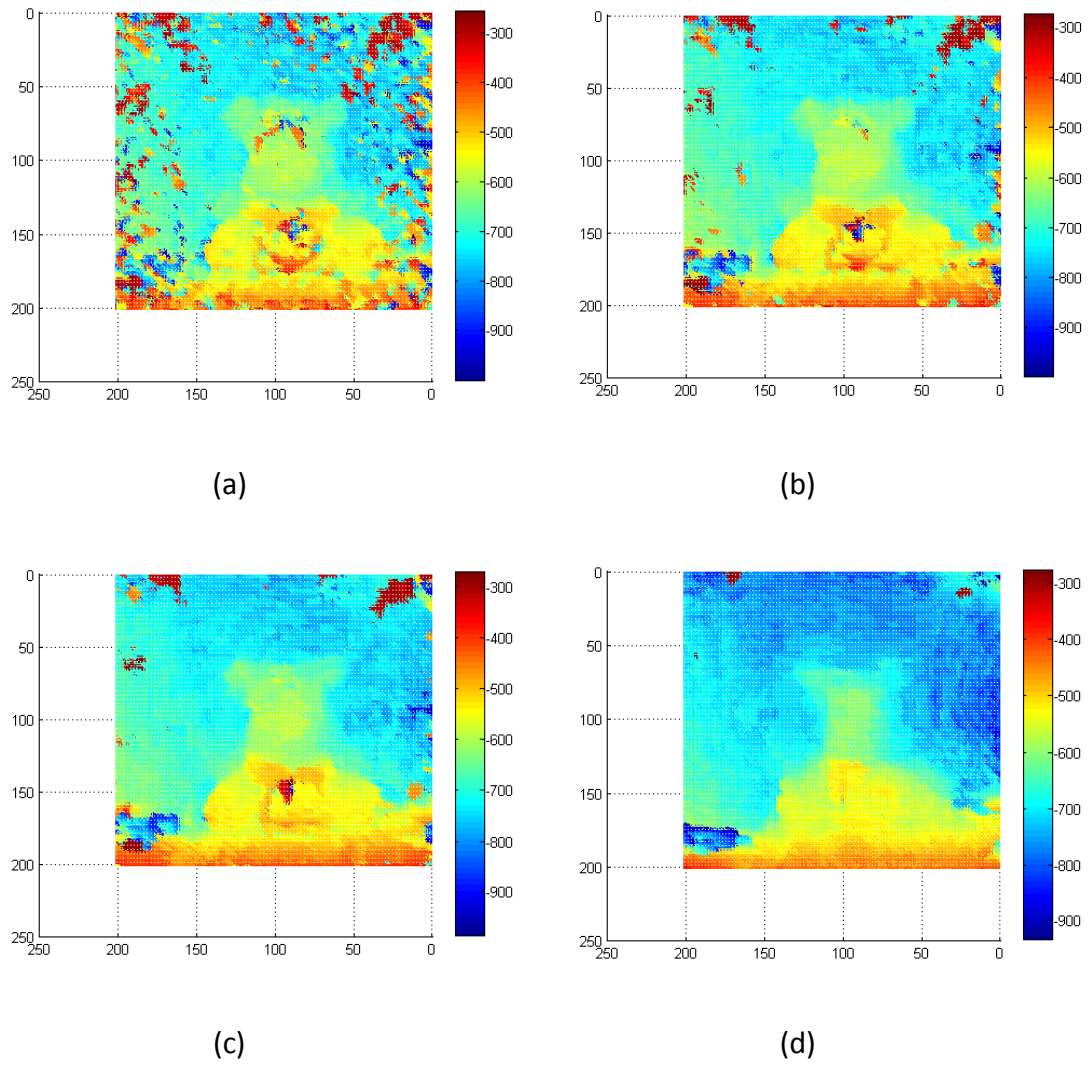


Fig. 6.13 The output depth map with different window size of 7X7, 11X11, 15X15 and 31X31, respectively

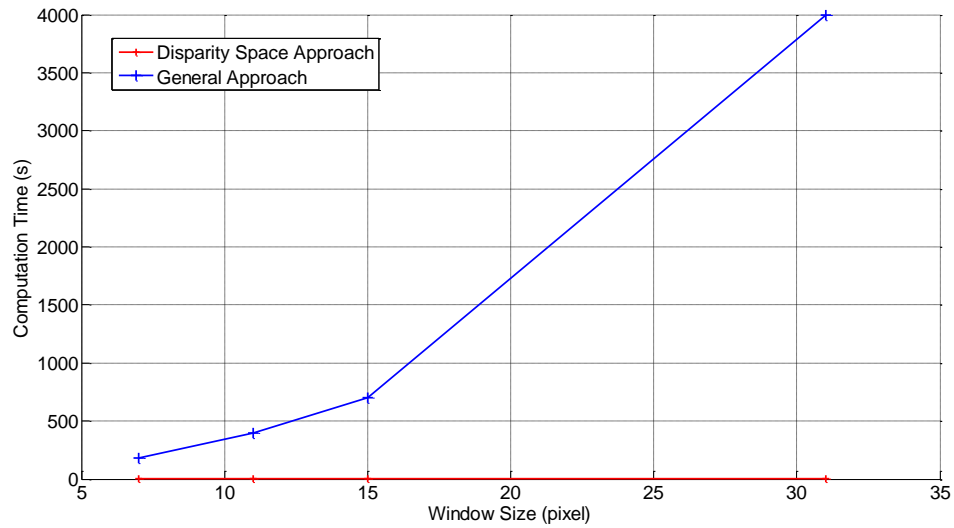


Fig. 6.14 Computation Time vs. Window Size

6.3.2 Fast Implementation Results

The disparity space approach in Section 5.2.2, however, is much faster and more efficient.

With the same input image Fig. 6.12 and with different window size, the depth map (Fig. 6.15) now could be obtained by our un-optimized Matlab codes in around 5 seconds (as shown in Fig. 6.14), which is at least 100 times faster than the general approach, and holding reasonable precision.

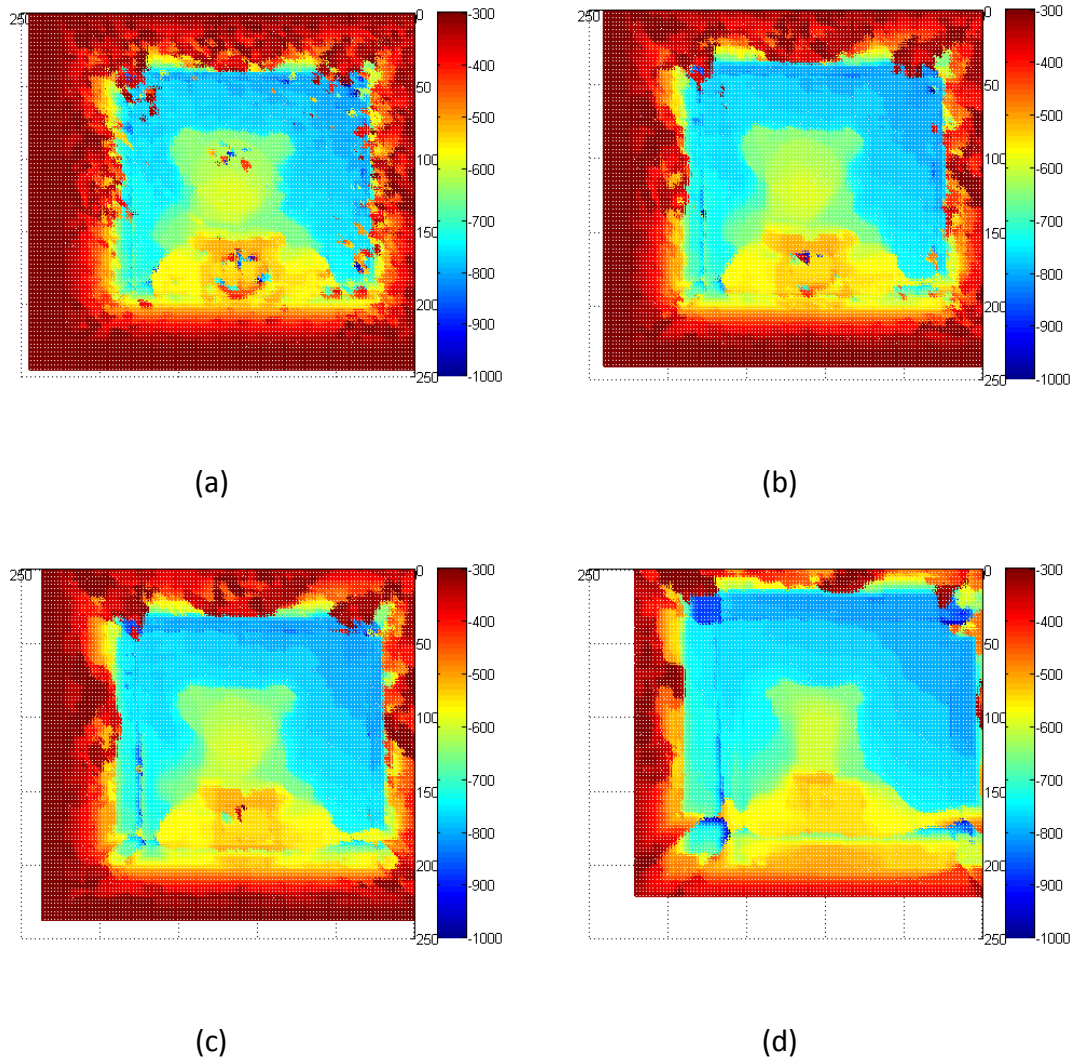


Fig. 6.15 Depth map obtained by Disparity Space approach with different window size of 7X7, 11X11, 15X15 and 31X31, respectively

Note the fact that the square region with small depth around the depth map is caused by the limitation of field of view with four mirrors. While the general approach's result does not show such region, illustrating that the general approach has mismatching in that entire region because of the limitation of distortion.

When compared the result of general approach (Fig. 6.13) with the disparity space approach (Fig. 6.15). It could be concluded that with the same size of local window,

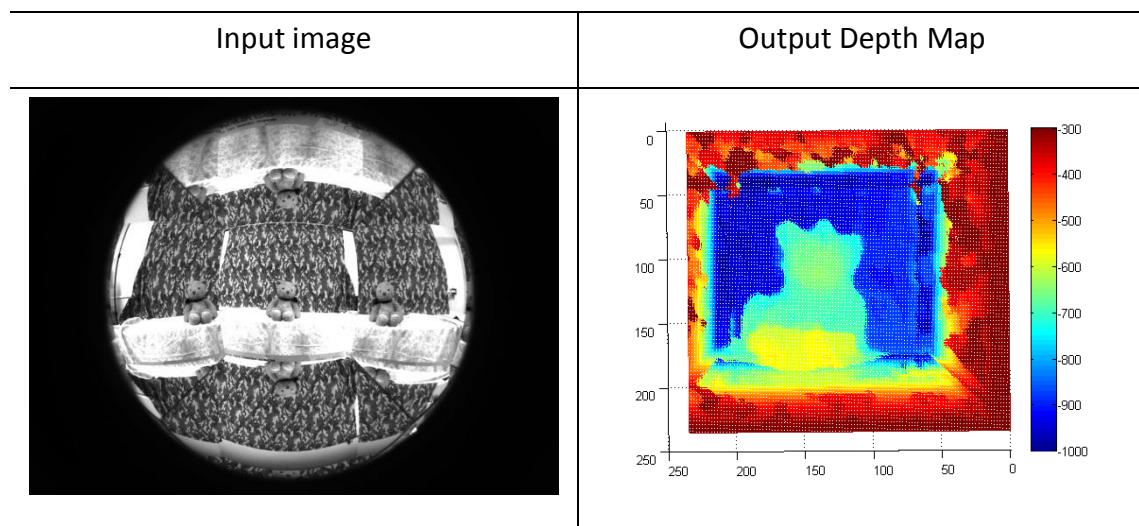
the disparity space approach always outperforms the general one, which demonstrates the advantage of ‘free from distortion’ of the general approach again.

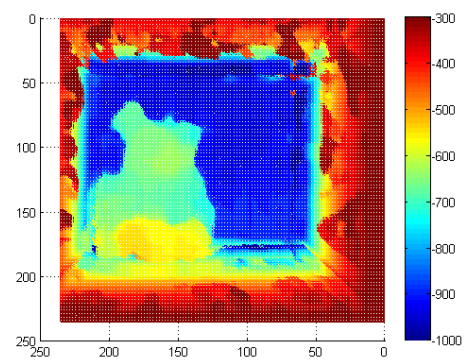
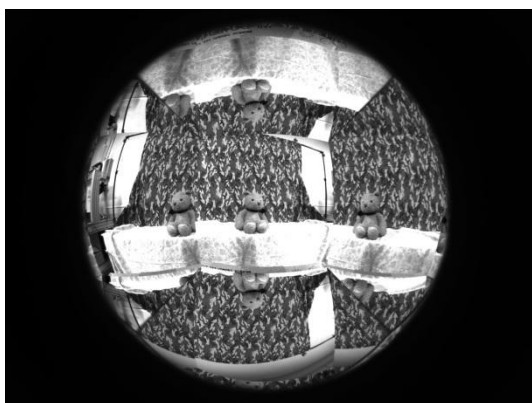
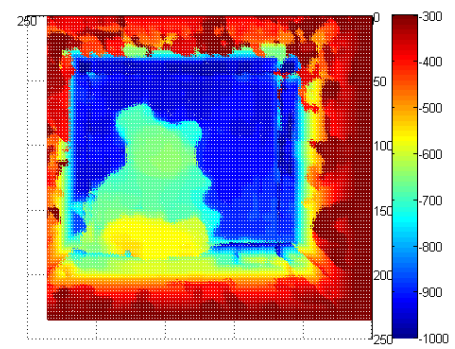
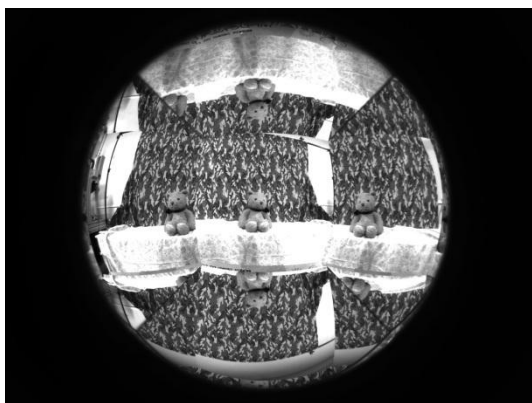
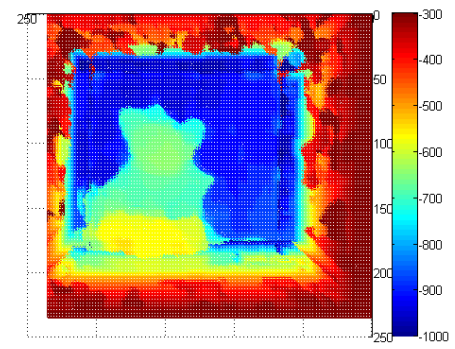
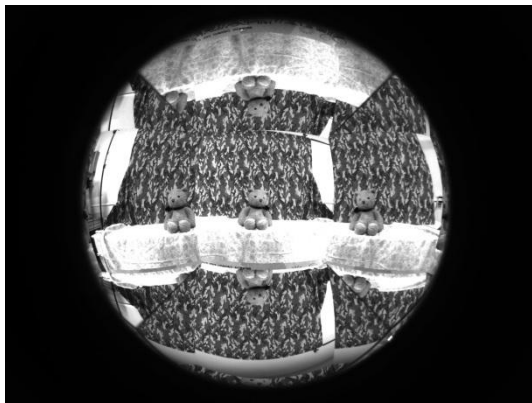
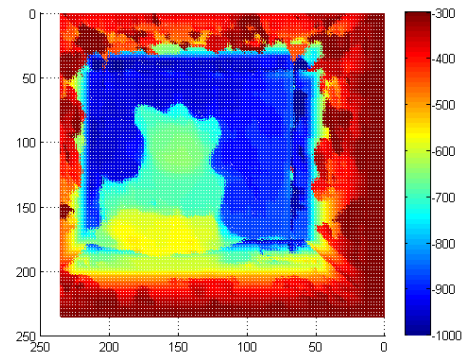
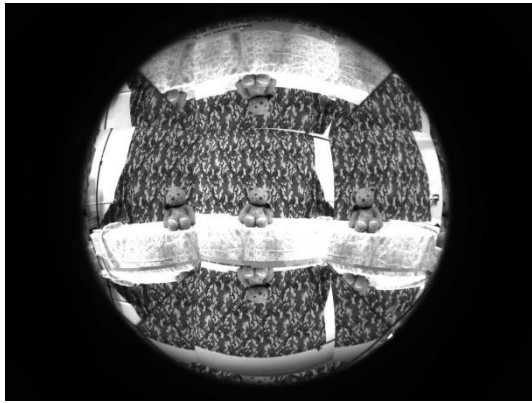
All the results shown above prove the effectiveness and accuracy of calibration, without which no such depth maps could be generated.

6.4 3D Reconstruction Results

Finally we show how 3D reconstruction is done with our camera system.

The camera system is moving around the object and the depth map of each shot is computed. As shown in Fig. 6.16, eight images are obtained as input, and eight depth maps are computed according to each shot with the window size 17X17.





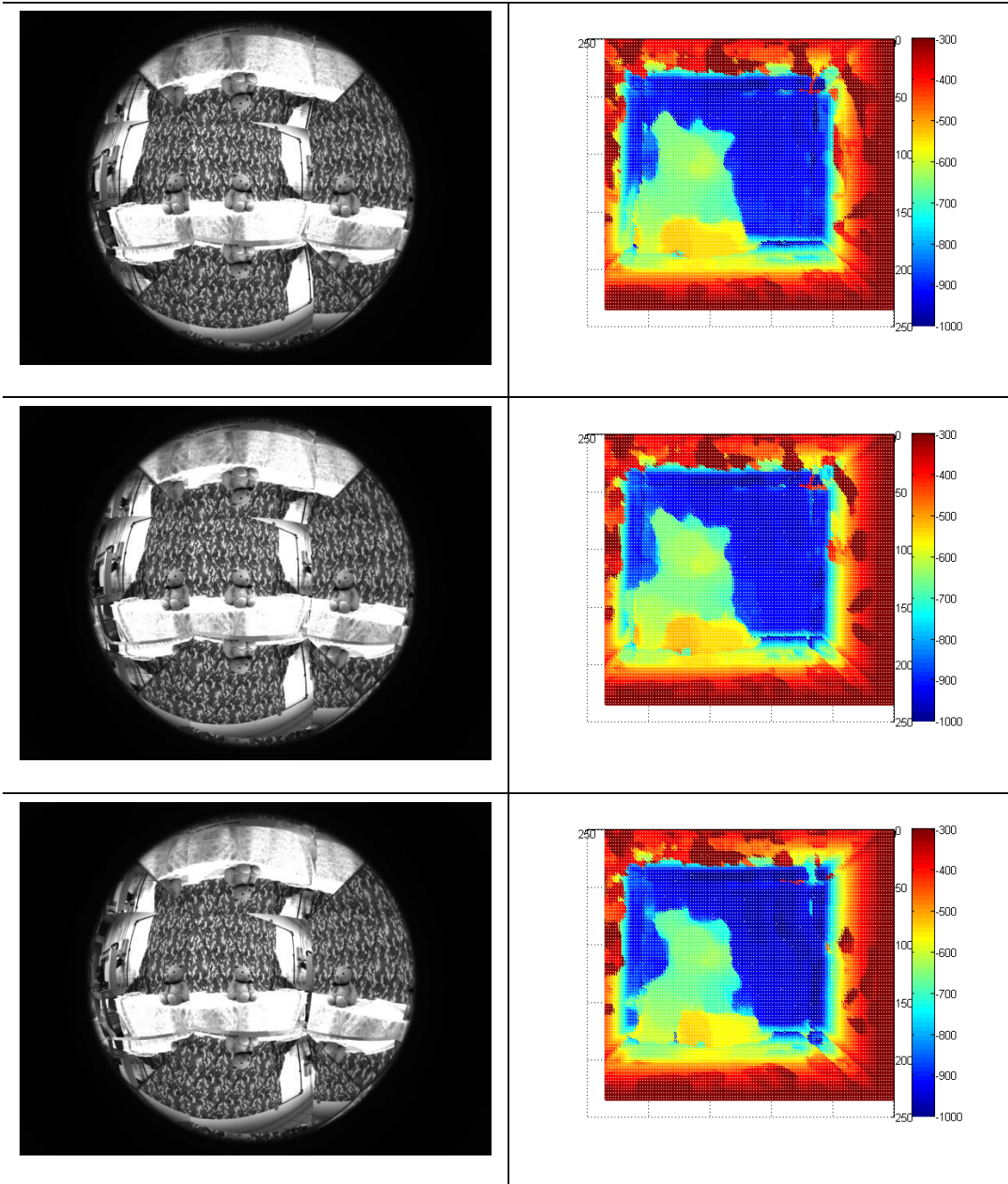
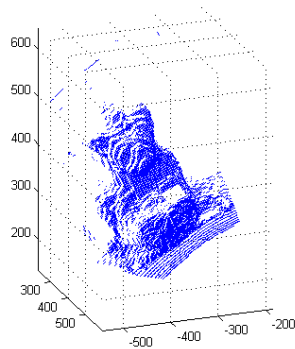
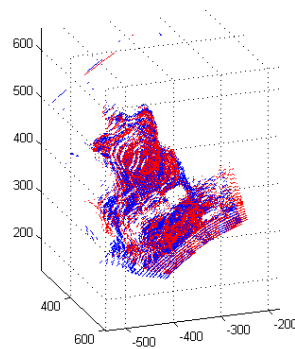
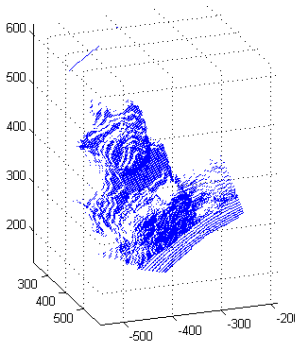
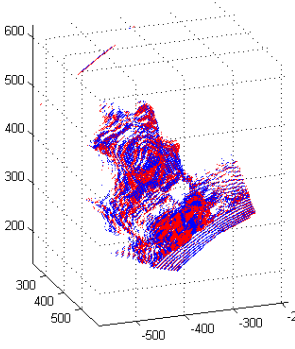
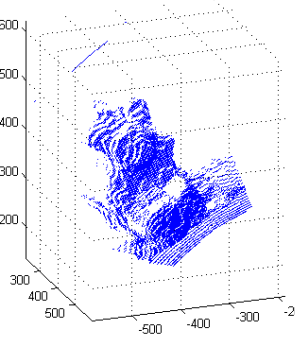
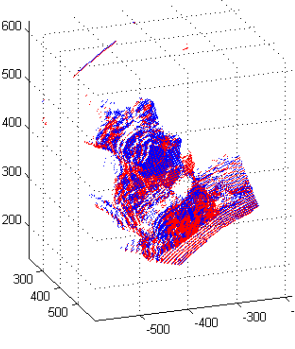
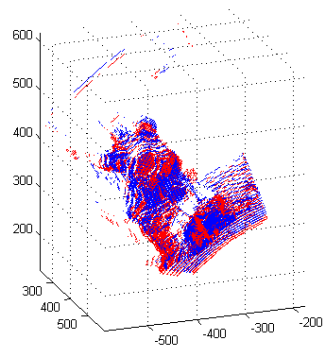
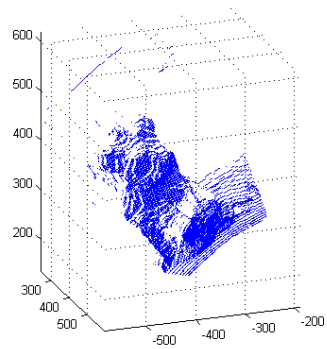
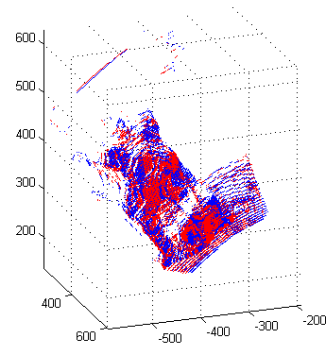
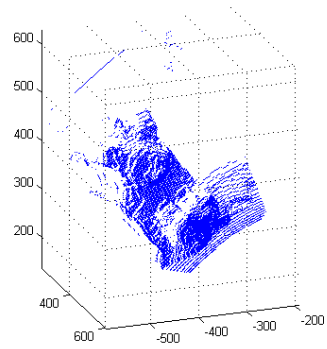
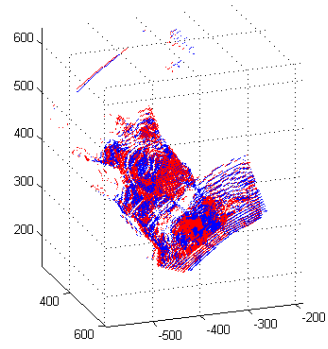
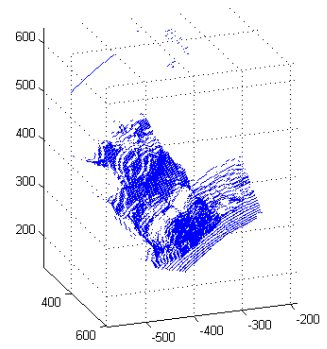
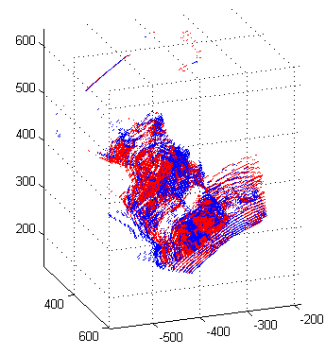
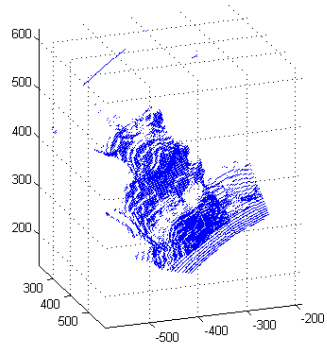


Fig. 6.16 The input image sequence and its depth map sequence

With the depth map and calibrated intrinsic parameters of the camera, the 3D point cloud could be obtained, as shown left column in Fig. 6.17, here we filter out such depth information as background, mirror box. The ICP algorithm [20] is applied to align adjacent point clouds. In detail, we utilize the ICP function [21] in Matlab, and the results are shown in the right column in Fig. 6.17.

3D Point Cloud	ICP Alignment between Adjacent shots
	
	
	



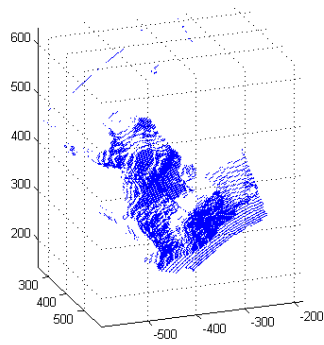


Fig. 6.17 ICP Alignment between two adjacent point clouds

Finally, we align all these 8 shots in one figure, as shown in Fig. 6.18

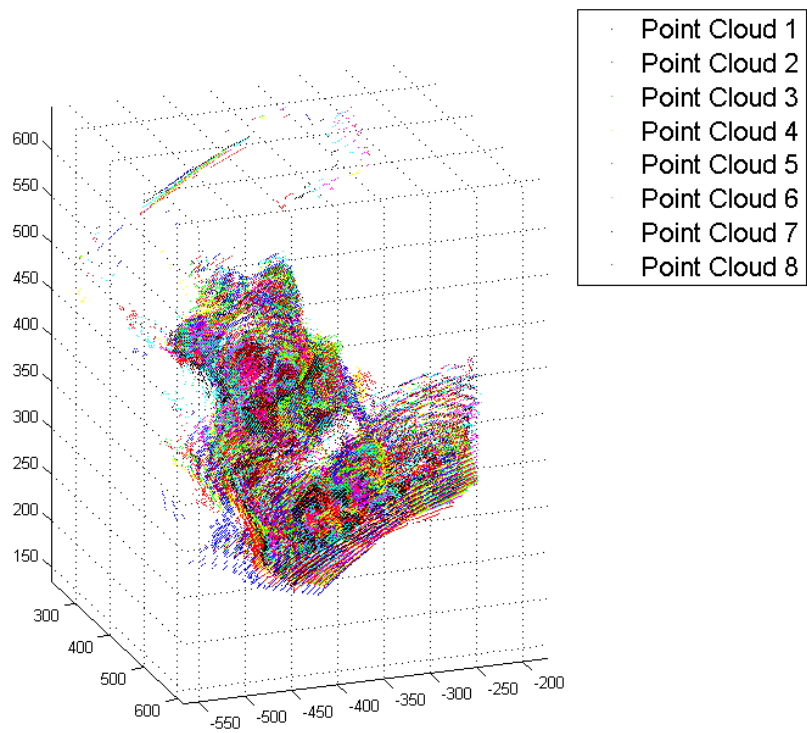


Fig. 6.18 Point Cloud Alignment with all input images

From the results shown in Fig. 6.18, side details of the bear is now available, and with more inputs, more details of the object would be obtained.

So the final goal of 3D reconstruction with the moving camera system is now accomplished.

Chapter 7

Conclusion and Future Work

In this thesis, we have shown the whole procedure how to efficiently generate the depth map by the monocular multi-view camera system. The main contribution of this work lies in two aspects: Calibration of the camera system and Multi-Baseline Stereo in disparity space.

To our knowledge, the proposed method in calibration is the first one that systematically calibrates the monocular camera system and its extension to general multiple omnidirectional camera system is also novel. In detail, the proposed linear calibration procedure is free from misleading compared to the baseline method, which would lead to a better final result after bundle adjustment. The merit of auto center detection from ‘OcamCalib’ by D. Scaramuzza is also inherited. The whole calibration method is implemented by a Matlab Toolbox, which makes the calibration procedure convenient and highly automatic.

Furthermore, the implementation of Multi-Baseline Stereo in disparity space outperforms the general approach in rapid computation speed. The disparity space approach is compact in data structure and flexible for application. With the OII technique, the aggregation in local window over the whole image could be done only with four one-dimension loops, the computation maintains even with the increased window size. Disparity space approach is also free from distortion in nature.

As the results shown, with single shot, the depth map could be obtained in seconds with reasonable precision. Simply moving the camera would generate different depth map with the same object from different views, then the well-known ICP algorithm is applied to align those point clouds with each shot, the experimental results have shown the validation of this approach of 3D reconstruction.

However, there are still some works needs to be done in the future.

Real Time Performance

It is interesting to note that the multi-baseline stereo matching in disparity space is highly parallel. And as done in [5], one important future direction is to implement the algorithm on Graphics Processing Units (GPU) by the programming environment of Nvidia/CUDA.

Structure from motion

Because of the property (getting the depth map with single shot) of our camera system, we do not focus on the motion effect very much. A future work is to explore the possibility and advantage of applying structure from motion with this camera system.

New hardware implementation

The monocular multi-view camera system now suffers from the disadvantages of low resolution of the fisheye camera and limited view angle. So by replacing the fisheye camera to updated fisheye lens would solve the problem of low resolution. We also plane to remove the side mirrors to get a larger view angle in horizontal and to explore the new properties of such camera system.

Reference

- [1] W. Jiang, M. Shimizu and M. Okutomi. Single-Camera Multi-Baseline Stereo using Fish-Eye Lens and Mirrors. *In Proc. ACCV 2009, pages MP1-34,1-12, 2009*
- [2] D. Scaramuzza, A. Martinelli and R. Siegwart. A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion. *In Proceedings of IEEE International Conference on Computer Vision Systems (ICVS'06), New York, January 2006.*
- [3] D. Scaramuzza, A. Martinelli and R. Siegwart. A Toolbox for Easily Calibrating Omnidirectional Cameras. *In IEEE International Conference on Intelligent Robots and Systems (IROS 2006). (2006)*
- [4] M. Okutomi and T. Kanade. A Multiple-Baseline Stereo. *In IEEE Trans. PAMI, 15(4): 353-363*
- [5] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang and X. Zhang. On Building an Accurate Stereo Matching System on Graphics Hardware. *In ICCV workshops 2011*
- [6] K. Zhang, J. Lu and G. Lafruit. Cross-Based Local Stereo Matching Using Orthogonal Integral Images. *In IEEE Trans. Cir. and Sys. For Video Technol., vol. 19, no. 7, pp. 1073-1079, 2009*
- [7] K. Zhang, J. Lu, G. Lafruit, R. Lauwereins and L. V. Gool. Real-Time Accurate Stereo with Bitwise Fast Voting on CUDA. *In ICCV Workshops 2009*
- [8] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *In International Journal of Computer Vision 47(1/2/3), 7-42, 2002*
- [9] M. Moakher. Means and Averaging in the Group of Rotations. *In SIAM Journal on Matrix Analysis and Applications, 2002*
- [10] T. Svoboda, D. Martinec and T. Pajdla. A Convenient Multi-Camera Self-Calibration for Virtual Environments. *In Teleoperators and Virtual Environments 14, 4 (August), 407-422.*
- [11] C. Geyer and K. Daniilidis. Catadioptric Projective Geometry. *In International Journal of Computer Vision 45, Number 3, December 2001, page 223-243.*

- [12] J. P. Barreto and K. Daniilidis. Wide Area Multiple Camera Calibration and Estimation of Radial Distortion. *In Omnivis-2004, ECCV-2004 workshop, 2004.*
- [13] J. Gluckman and S. K. Nayar. Catadioptric Stereo Using Planar Mirror. *In International Journal of Computer Vision 44(1), 65-70, 2001.*
- [14] J. Gluckman and S. K. Nayar. Planar Catadioptric Stereo: Geometry and Calibration. *In Proceedings of the 1999 Conference of Computer Vision and Pattern Recognition, 1999.*
- [15] J. J. More. The Levenberg-Marquardt algorithm: Implementation and theory. *In Watson GA (ed): Numerical Analysis. Lecture Notes in Mathematics 630. Berlin: Springer-Verlag, pp. 105-116*
- [16] B. Micusik, T. Pajdla. Autocalibration & 3D Reconstruction with Non-central Catadioptric Camera. *In CVPR 2004, Washington US, June 2004*
- [17] J. Kumler and M. Bauer. Fisheye lens designs and their relative performance.
- [18] B. Micusik, D. Martinec and T. Pajdla. 3D Metric Reconstruction from Uncalibrated Omnidirectional Images. *In ACCV 2004, Korea January 2004.*
- [19] T. Svoboda and T. Pajdla. Epipolar Geometry for Central Catadioptric Cameras. *In IJCV, 49(1), pp. 23-37, Kluwer August 2002.*
- [20] P. J. Besl and N. D. McKay. A Method of Registration of 3-D Shapes. *In IEEE Transactions on Pattern Analysis and Machine Intelligence - Special issue on interpretation of 3-D scenes—part II, Volume 14 Issue 2, February 1992.*
- [21] J. Wilm. Iterative Closest Point algorithm in Matlab. *In <http://www.mathworks.com/matlabcentral/fileexchange/27804-iterative-closest-point>*