HORSE-VISION-SYSTEM-BASED SCENE ANALYSIS AND SINGLE-VIEW SCENE UNDERSTANDING FROM OMNIDIRECTIONAL IMAGE

by

Hanchao Jia

Doctor of Engineer

Graduate School of Engineering

Tottori University

TABLE OF CONTENTS

LIST OF FIGURES ..... **IV**

LIST OF TABLES ..... **VII**

ABSTRACT ..... **VIII**

ACKNOWLEDGEMENTS ..... **X**

CHAPTER 1 INTRODUCTION **1**

CHAPTER 2 HORSE-VISION-SYSTEM-BASED SCENE ANALYSIS **5**

2.1 Introduction 5

2.2 Related Research 8

2.3 Realization of the HVS 8

2.3.1 Omnidirectional Monocular Vision by the HVS 8

2.3.2 Binocular Vision by the HVS 10

2.3.2.1 Determination of the CFOVRs 10

2.3.2.2 Rectification of the CFOVRs 14

2.3.2.3 Acquisition of dense disparity map 16

2.3.3 The whole vision of HVS 19

2.4 Scene Analysis by the HVS 20

2.4.1 Experiment 1: Position Estimation of One Person. 22

2.4.2 Experiment 2: Position Estimation of two Persons 24

2.5 Conclusions 26

CHAPTER 3 SINGLE-VIEW SCENE UNDERSTANDING **28**

3.1 Introduction 28
3.2 Related Research 33
3.2.1 Single-View Geometry Estimation using Perspective Images 33
3.2.2 Single-View Geometry Estimation using Omnidirectional Images 34
3.3 A Model for Indoor Scene 35
3.3.1 Sphere Camera Model 35
3.3.2 Indoor World Model 37
3.4 Estimating the Structure of Rooms from a Single Fisheye Image 38
3.4.1 Preliminary Spatial Layout Estimation 39
3.4.2 Computing the Orientation Map 43
3.4.3 Refining the Spatial Layout 44
3.4.4 Experiments 45
3.5 Indoor Scene Understanding from a Single Full-View Image 48
3.5.1 Close Geometry 48
3.5.2 Estimating Spatial Layout 50
3.5.2.1 Preliminary work 51
3.5.2.2 Generation of corners 51
3.5.2.3 Problem formulation 53
3.5.2.4 Algorithm of estimating the spatial layout 56
3.5.3 Experiments and Results 61
3.5.3.1 Using full-view images from different sensors 61

3.5.3.2 Close geometry vs. open geometry 64

3.5.3.3 Proposed method vs. conventional method 66

3.6 Conclusion 70

CHAPTER 4 FAST GENERATION OF PERSPECTIVE DISPLAY FROM FULL-VIEW IMAGE 72

4.1 Introduction 72

4.2 Related Research 74

4.2.1 Research about Bubbles 74

4.2.2 Research about SCVT 76

4.3 Generating Perspective Display from SCVT Map 82

4.3.1 Generation of Perspective Display by Using Neighboring Relation 82

4.3.2 Generation of Perspective Display by Using Pyramidal Data Structure of Spherical Bubble 83

4.3.3 The Process of Generation of Perspective Display 87

4.4 Experiment 88

4.4.1 Performance of Computational Speed 88

4.4.2 Performance of Image Quality 93

4.5 Conclusion 100

CHAPTER 5 CONCLUSION 101

REFERENCES 103

LIST OF PUBLICATION 108

LIST OF FIGURES

Fig 2.1 A sketch for the function of horse vision. 7

Fig 2.2 The horse vision system (HVS) proposed in this thesis. 7

Fig 2.3 The integrated spherical image from a pair of fisheye images in Fig 2.2(b)……9

Fig 2.4 The flatten image of Fig 2.3. 10

Fig 2.5 The common fields of view of two cameras in our system. 11

Fig 2.6 The determined common fields of view. 14

Fig 2.7 The sketch of the rectification.. 15

Fig 2.8 The rectified views of Fig 2.6. 16

Fig 2.9 The rectified views with epipolar lines..... 18

Fig 2.10 The dense disparity map computed using the longitude-latitude image. 19

Fig 2.11 The whole vision by the VHS..... 20

Fig 2.12 The bird-eye view of estimated moving track. 23

Fig 2.13 Horizontal projection of estimated moving track in the coordination of HVS. 23

Fig 2.14 The error distribution against θ variation in Fig 2.13 24

Fig 2.15 Motion detection of one pair frames. 25

Fig 2.16 The results of position estimation. 26

Fig 3.1 Different fields of view of images captured by a conventional camera and a fisheye camera. 29

Fig 3.2 The images of two types of geometry.. 31

Fig 3.3 Examples of full-view images. 32

Fig 3.4 The sphere model for full-view images. 37

Fig 3.5 A structure hypothesis.. 40

Fig 3.6 Process of preliminary spatial layout estimation. 42

Fig 3.7 The result of the refined spatial layout.. 45

Fig 3.8 An example of experiments. 47

Fig 3.9 The criterions of close geometry. 50

Fig 3.10 Three cases of corners generated from vertical lines and one case generated from horizontal lines. 52

Fig 3.11 Corners are ordered clockwise according to their projections in X-Z plane. 53

Fig 3.12 A special case that some segments of floor-wall boundaries are vertical lines. 54

Fig 3.13 The procedure of computing the most feasible boundary.. 60

Fig 3.14 An example of experimental result from a fisheye camera. 62

Fig 3.15 Some cases of experimental result from RICOH THETA.. 64

Fig 3.16 The comparison between close geometry and open geometry. 66

Fig 3.17 The comparison between the proposed method and the method of Lee et al. [29]. 68

Fig 3.18 The percentage of the correct orientation for the results in Fig 3.17..... 69

Fig 4.1 Perspective display based on the successive spherical model. 73

Fig 4.2 Sampling rate of cubic environment map for the directions. 75

Fig 4.3 Process of the geodesic division of an icosahedron. 78

Fig 4.4 2D array of SCVT image..... 80

Fig 4.5 The arrays of cell points for 0-level and 1-level subdivisions corresponding Fig 4.3(a) and (b). 84

Fig 4.6 The sketch of down-sampling of the SCVT image. 85

Fig 4.7 The sketch of up-sampling of the SCVT image. 86

Fig 4.8 The spherical image used in the experiment. 89

Fig 4.9 The pyramidal data structure of the SCVT image generated referring to 8^{th}, 7^{th}, 6^{th}, 5^{th}, 4^{th}level of subdivision of spherical bubble. 90

Fig 4.10 Perspective display based on the discrete spherical model. 91

Fig 4.11 The original image used in the experiment in section 4.4.2. 94

Fig 4.12 The SCVT image generated from Fig 4.11. 95

Fig 4.13 The perspective displays corresponds to up-sampling. 97

Fig 4.14 The perspective displays corresponds to down-sampling.. 99

LIST OF TABLES

Table 3.1 Percentage of pixels with correct orientation..... 65

Table 4.1 Properties of the SCVT map. 77

Table 4.2 Comparison of computational time under the condition of the fixed image size……..... 92

Table 4.3 Comparison of computational time under the condition of the fixed perspective angle. 93

Table 4.4 Comparison of image quality with reference image. 98

ABSTRACT

There are two challenging tasks in computer vision. One is to construct a vision system to execute scene analysis in the special environment, for the case which humans are not able to do, such as an endoscope for visual examination in hospitals, an omnidirectional camera for surveillance with a large field of view. On the other hand, we want computers to understand our world the way we do, make scene understanding more human-centric, referring to scene understanding.

This thesis focuses on the application of omnidirectional cameras in scene analysis and scene understanding. At first, a method of scene analysis based on a horse vision system is proposed. A horse vision system (HVS) consists of a pair of fisheye cameras which have a hemispherical field of view, respectively, and are laid to overlap each other partially. The characteristics of the HVS result in a representation which enables a wide omnidirectional monocular vision and a limited-field-of-view binocular vision simultaneously. The method for the realization of the proposed HVS and the preliminary experimental results of scene analysis based on the HVS are presented.

Secondly, the problem of recovering the structure of an indoor scene from a single image is studied. A novel method of estimating the spatial layout of rooms from a single fisheye image is introduced. However, fisheye images involve just partial scene, which result in visually open boundary condition, called *open geometry*. A full-view image

results in a visually close boundary condition, called *close geometry*. The characteristics of close geometry are employed to explore indoor scene understanding from a single full-view image.

Additionally in scene analysis, it is often necessary to transform the unfamiliar images captured by special cameras into the view similar to that of humans' vision. For example, the street view of city is shown and changed smoothly from panoramic images in Google Street View on the web. As a basic processing operation of omnidirectional images, a method of quickly generating the perspective display from a full view image according to users' view direction and zoom-in/out operation is proposed.

ACKNOWLEDGEMENTS

First, I would like to acknowledge my supervisor, Shigang Li, who introduced me to the field of computer vision, and guided me from my Master course. Under his supervision, I have learned doing research and also gained invaluable experiences in other aspects.

I also thank my thesis committee members Dr. Nakanishi and Dr. Itou for their nice advisement. To the members and friends in my lab, I am thankful for having the chance to work closely with them.

It is a nice experience to study in Tottori University. I would like to acknowledge the other people, who help me no matter in my life or for my research. I believe the life in the past few years in Tottori will become an unforgettable memory.

Finally, I thank my family for their unconditional love, support and encouragement.

CHAPTER 1

Introduction

This thesis focuses on the application of omnidirectional cameras in scene analysis and scene understanding. It can be organized in three parts: Horse-Vision-System-based scene analysis, single-view scene understanding from omnidirectional images and fast generation of perspective display from full-view images. A brief introduction for each is given in the following.

Vision-based scene analysis is a basic research topic in machine vision, and many approaches have been developed until now for this task. What kind of approaches should be used depends on the concrete task of scene analysis. For example, for motion detection by a stationary single camera, the method of background subtraction is popular and effective; for the construction of 3D structure of environment, a stereo method is usually used. While the processing of background subtraction by a single camera is simple, the construction of 3D structure of environment by a stereo method not only needs multiple cameras, but also costs a large amount of computation relatively. The scene analysis using a single camera is referred to as monocular vision while that using two cameras is referred to as binocular vision.

When we construct a vision system for a general scene analysis task, biological vision systems often give us a hint. Human beings have a binocular vision system; however, the scene in the rear cannot be observed. To cope with a dynamic environment

in real time, a wide field of view (FOV) brings definite merits to a vision system. In the first part of the thesis, a pair of fisheye cameras is used to imitate horses' eyes, constructing a vision system, called *horse vision system* (HVS), with two modes of vision, monocular vision and binocular vision. A mobile robot can detect obstacles by using the HVS with the narrow FOV binocular vision, and meanwhile, monitor the surrounding environment with the wide omnidirectional monocular vision.

On the other hand, though the research in computer vision has developed rapidly, scene understand is still a challenging task for computers. From a single image, humans can immediately grasp the spatial layout of the scene and understand what the image tells. However it is difficult for computers. How to make computers consider images like humans? As images are reflections of the reality, we cannot understand them without prior knowledge on the real world. A person may have nothing about a picture of the one he has not known, for example, a view of atom. Compared to complex natural condition, a familiar structured man-made environment is easier to be modeled.

Recent years have seen a growing interest in indoor scene understanding from a single image. Compared to a full dense reconstruction, this technique is more efficient and more robust for indoor environments with less texture. Moreover, it may provide strong indicators to the structure of rooms to easily distinguish objects from background in scene interpretation task. However, existing approaches typically rely on the pinhole camera geometry. These conventional cameras have a relative small field of view; on the other hand, omnidirectional systems that can provide a wide field of view are gaining

popularity. Obviously, the wider the field of view is, the more information we can gather from the environment. Therefore, why not employ an omnidirectional camera to carry out scene understanding task, which enable computers to predict more valuable information from indoor scene, as well as recognize the whole room by only one or two images.

In the second part of this thesis, scene understanding for a fisheye image is first explored. A novel method is given to estimate the spatial layout of rooms only from a collection of line segments. Then, we pay our attention to full-view images. Nowadays, 360-degree panorama display becomes easily obtained and widely used in various aspects. An approach of indoor scene understanding from a single full-view image is proposed.

In the third part, generation of perspective display from full-view images is studied. Though omnidirectional cameras enable robots to gain more valuable information from the environment, the acquired images are not friendly to humans because of large distortion. In order to overcome the deficiency, perspective display, which is much similar to that of humans' vision, needs to be generated frequently. It benefits the application such as robot control, humanoid robot, especially for constructing the interfaces of robot that focused on providing users with the most current information as if they can see by themselves. In addition, some algorithms based on conventional cameras may be adapted directly to the perspective images. It is appropriate to say that generating perspective display is a basic operation for the research of omnidirectional vision.

In general, this processing is time-consuming because of mass non-linear calculation. In this thesis, omnidirectional images are represented by SCVT (Spherical Centroidal Voronoi Tessellation) images which are called spherical bubbles, and a method to generate perspective display swiftly more than before is proposed.

In the remainder of the thesis, the detailed descriptions of these three parts are given.

CHAPTER 2

Horse-Vision-System-based Scene Analysis

2.1 Introduction

A horse has two big eyes, which are located on either side of its head. On one hand, like people, it can see the same scene with both eyes at once, resulting in better depth perception and a more concentrated field of vision. On the other hand, a horse can also use each eye to see separate scene, which enables it to get a view of surroundings on both sides. It is greatly important for horses to detect stalking carnivorous animals sneaking up from behind in order to avoid danger. So different from people, a horse holds a wide, circular view with a range of vision of more than 350°. Approximately 65° among that is binocular vision, while the remaining 285° is monocular vision [1], [2], as shown in Fig 2.1. A pair of fisheye cameras is used to imitate a horse's eyes, as shown in Fig 2.2(a). Each of them has a hemispherical FOV. The pair of fisheye cameras is mounted on a rig with the overlapping region of observation fields so as to acquire the two modes of vision, monocular vision and binocular vision, simultaneously.

There are following characteristics in the HVS:

- The pair of fisheye cameras point to different directions, respectively. Meanwhile, between them there are overlapping regions, i.e., the common FOV region (CFOVR), in the fisheye images for the realization of binocular vision. Obviously, this setup is different from that of the conventional canonical stereo.

- The two visual modes of the HVS, omnidirectional monocular vision and binocular vision result in reasonable distribution of limited computation power of a computer. Using the HVS, a robot can monitor the surrounding environment by the omnidirectional monocular vision mode, and measure the 3D information by the binocular vision mode only for a part of environment which is necessary for detailed investigation.

The main contributions of this research are as follows.

- Construct a biologically-inspired vision system, the HVS, like horses' eyes. According to our best knowledge, it may be the first system to try to imitate the visual function of a horse's eye.
- Present an algorithm of identifying the CFOVR and rectifying the CFOVRs for the binocular vision of the HVS.
- Present the preliminary experimental results of the scene analysis based on the proposed HVS.

The remainder of this part is organized as follows: Related research is introduced in the next section. In Section 2.3, the realization of the HVS is described. An application of scene analysis by the HVS is presented in Section 2.4. Finally, conclusions are given in Section 2.5.

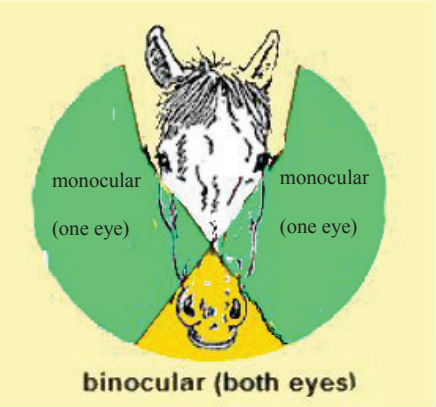
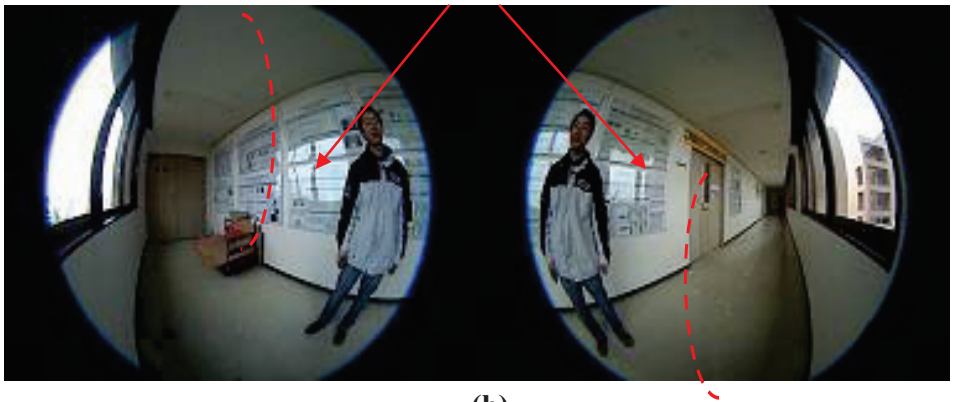


Fig 2.1 A sketch for the function of horse vision.

(a)

common fields of view region



(b)

Fig 2.2 The horse vision system (HVS) proposed in this thesis. (a) The horse vision system. (b) A pair of sample fisheye images captured by the HVS.

2.2 Related Research

Omnidirectional image sensors are widely used for visual surveillance and vision-based robot navigation. An omnidirectional image can be obtained by a catadioptric image sensor [3], a fisheye camera [4], or a camera cluster [5]. Thanks to the wide FOV of an omnidirectional camera, it is very effective to detect the motion from the surrounding environment speedily. However, it is difficult to measure the 3D information of the surrounding environment from a single omnidirectional camera.

To measure 3D information of the surrounding environment, multiple omnidirectional cameras are usually used by modifying the conventional stereo method [6], [7]. In order to obtain the 3D information of the surrounding environment as much as possible, the FOV of the multiple omnidirectional cameras are laid to overlay each other as largely as possible. In comparison with the motion detection by the background subtraction, the stereo method costs a lot of computation.

2.3 Realization of the HVS

In this section, we first describe the method of realizing the HVS, using a pair of fisheye cameras. Then we express the whole vision by combing the two visual modes, omnidirectional monocular vision and binocular vision.

2.3.1 Omnidirectional Monocular Vision by the HVS

The intrinsic parameters of the two fisheye cameras used in the VHS are calibrated using the method in [4] beforehand. Given the correspondence of features

between the pair of fisheye images, the relative pose between the pair of fisheye cameras can be computed using the method of [8]. Using the intrinsic and extrinsic parameters, the pair of fisheye images can be integrated into a wider omnidirectional image or a spherical image.

Fig 2.3 shows the spherical image obtained by mapping the pair of fisheye images onto a sphere. The duplication of the scenes appears in the CFOVR. Fig 2.4 shows the image obtained by extending the spherical image along the longitudinal and latitudinal directions.

(a) **(b)**

Fig 2.3 The integrated spherical image from a pair of fisheye images in Fig 2.2(b). (a) The front. (b) The rear.

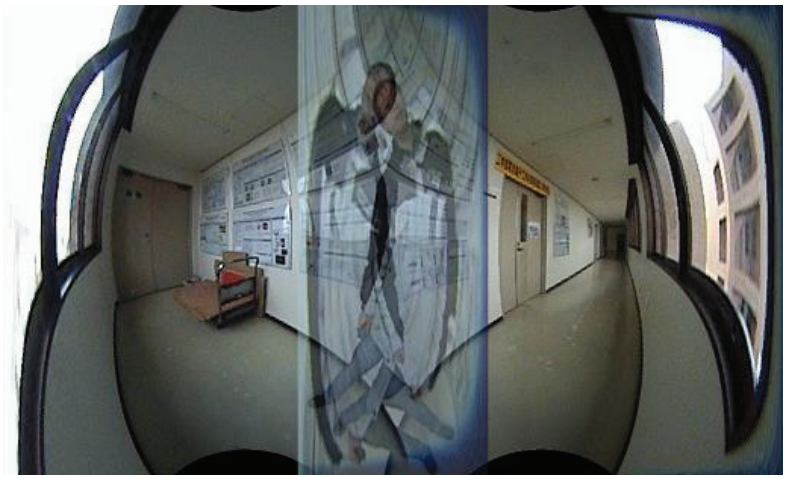


Fig 2.4 The flatten image of Fig 2.3.

2.3.2 Binocular Vision by the HVS

Here, we focus on the explanation of the difference for the rectification of the HVS in comparison with the conventional canonical stereo.

2.3.2.1 Determination of the CFOVRs

First, we explain the idea of determining the boundaries of the CFOVRs. Fig 2.5 shows two hemispherical images which correspond to the pair of fisheye images of the HVS. The boundary of the hemispherical images is the great circle on the XY plane of the camera coordinate system, respectively. The CFOVRs to be determined are indicated in blue color. Obviously, one part of the boundaries of the CFOVRs is the boundary of the hemispherical images. Thus, only the other part needs to be determined.

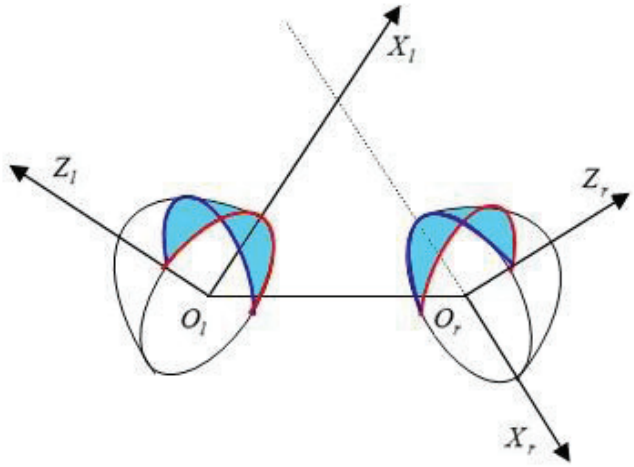


Fig 2.5 The common fields of view of two cameras in our system.

It is estimated by mapping the boundary of one hemispherical image to the other in terms of the relative orientation between the two fisheye cameras. Note that the rays from a scene point which has a big distance from the HVS points to the two cameras are almost the same. It implies that the displacement between the pair of fisheye cameras can be ignored. We also give the detail of the algorithm of determining the boundaries of the CFOVRs.

The boundary of a hemisphere refers to the great circle of the sphere on the x-y plane in the camera coordinate O-XYZ. Assume that p (u, v, s) is a point in spherical space, then the boundary can be represented as

$$\begin{cases} u^2 + v^2 + s^2 = r^2 \\ s = 0 \end{cases} \tag{2.1}$$

Where, r is the radius of the sphere. Similarly, for corresponding points p_l (u_l, v_l, s_l) and p_r (u_r, v_r, s_r) on the circumference of the great circle of the hemisphere respectively, we have:

$$\begin{cases} u_l^2 + v_l^2 + s_l^2 = r^2 \\ s_l = 0 \end{cases}, \quad \begin{cases} u_r^2 + v_r^2 + s_r^2 = r^2 \\ s_r = 0 \end{cases} \tag{2.2}$$

Using the relative pose (R_{rl}, t) between the pair of fisheye cameras computed in Section 2.3.1, we have the following equation:

$$p_l = R_{rl} p_r + t \tag{2.3}$$

By ignoring the displacement t between the pair of fisheye cameras, the equation above is represented as follows:

$$p_l = R_{rl} p_r \tag{2.4}$$

Then, the boundaries of the left CFOVR can be presented in two parts. One is the same as the original image (indicated in red color),

$$\begin{cases} u_l^2 + v_l^2 + s_l^2 = r^2 \\ s_l = 0, u_l > 0 \end{cases} \tag{2.5}$$

and the other is the projection of the half circumference of the great circle of the hemisphere in the right image (indicated in dark blue color). Using the equation (2.4), it can be presented as follows:

$$\begin{cases} (R_{rl}^{-1}p_l)_u^2 + (R_{rl}^{-1}p_l)_v^2 + (R_{rl}^{-1}p_l)_s^2 = r^2 \\ (R_{rl}^{-1}p_l)_s = 0, (R_{rl}^{-1}p_l)_u < 0 \end{cases} \quad \textbf{(2.6)}$$

Where, $(R_{rl}^{-1}p_l)_u$, $(R_{rl}^{-1}p_l)_v$, $(R_{rl}^{-1}p_l)_s$ are the values of $(R_{rl}^{-1}p_l)$ in each coordinate X, Y, Z, respectively. We use the same notation for others.

Similarly, the right CFOVR can be determined, presented as follows:

$$\begin{cases} u_r^{\,2} + v_r^{\,2} + s_r^{\,2} = r^2, s_r = 0, u_r < 0 \\ (R_{rl}p_r)_u^2 + (R_{rl}p_r)_v^2 + (R_{rl}p_r)_s^2 = r^2, (R_{rl}p_r)_s = 0, (R_{rl}p_r)_u > 0 \end{cases} \quad \textbf{(2.7)}$$

Finally, we have the equations of the boundaries of two CFOVRs. To test the method, we map the determined boundaries of the CFOVRs onto the fisheye image of the HVS by the known intrinsic parameters, as shown in Fig 2.6.

Fig 2.6 The determined common fields of view.

2.3.2.2 Rectification of the CFOVRs

Suppose that a scene point P(X,Y,Z) is projected onto the pair of the original hemispherical image of the HVS as p_{slo} and p_{sro}, respectively. Using the relative pose (R_{rl}, t) between the pair of fisheye cameras of the HVS, we have the following equations:

$$p_{slo} \cong [A \quad 0]\widetilde{P}, \; p_{sro} \cong [R_{rl}^{-1} \quad -R_{rl}^{-1}t]\widetilde{P} \tag{2.8}$$

where $\widetilde{P}$, $\widetilde{P} = (P \quad 1)^T$ is the homogeneous coordinate of P. The symbol $\cong$ means that p_{slo} and p_{sro} are equal to P multiplied by a scale factor.

To rectify the original CFOVRs, the original hemispherical images must be rotated so that the axes of the two new camera coordinate systems of the rectified CFOVRs are parallel to each other. Suppose that, after the rectification, the rotation matrix R_{rl} is changed to $R' = [r_1 \; r_2 \; r_3]^T$. The projections of scene point P onto the two rectified spherical images p_{sln} and p_{srn} can be represented as follows in term of (2.8):

$$p_{s\ln} \cong [R' \quad 0]\widetilde{P}\,,\ p_{srn} \cong [R' \quad -R't]\widetilde{P} \tag{2.9}$$

According to the characteristics of the HVS, R' is determined as follows (see Fig 2.7).

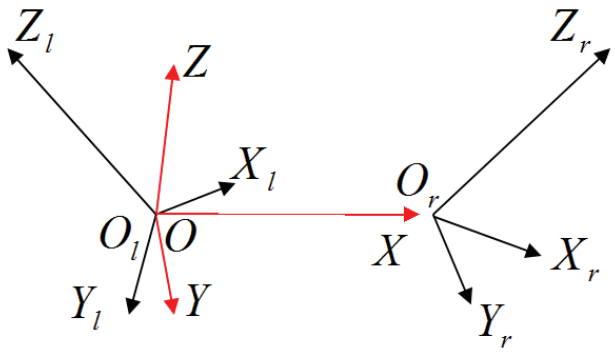


Fig 2.7 The sketch of the rectification. O-XYZ indicates the rectified camera coordinate system.

1) Let X-axis be parallel to the line joining the two centers of images. Thus, we have:

$$r_1 = \frac{t}{|t|} \tag{2.10}$$

2) Let the vector sum of the unit vectors z_1 and z_2 of Z-axis of two original images be k, and r_2 corresponding to Y-axis is presented as follows:

$$r_2 = k \times r_1 \tag{2.11}$$

Where

$$k = \frac{(z_1 + z_2)}{\|z_1 + z_2\|} \tag{2.12}$$

3) r_3 corresponding to Z-axis is determined based on r_1 and r_2, which is shown as follows:

$$r_3 = r_1 \times r_2 \tag{2.13}$$

According to the method of [7], for (2.8), (2.9) we have:

$$p_{s\ln} = R' p_{slo}\,,\ p_{srn} = R' R_{rl} p_{sro} \tag{2.14}$$

Finally, we can get the rectified image such as Fig 2.8.

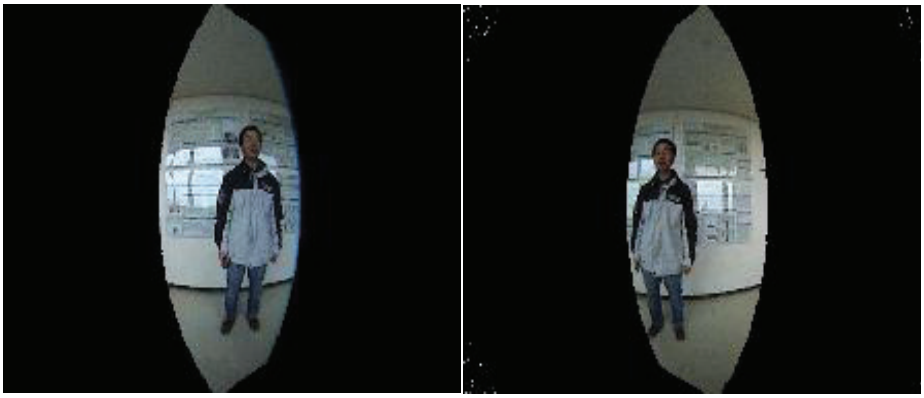
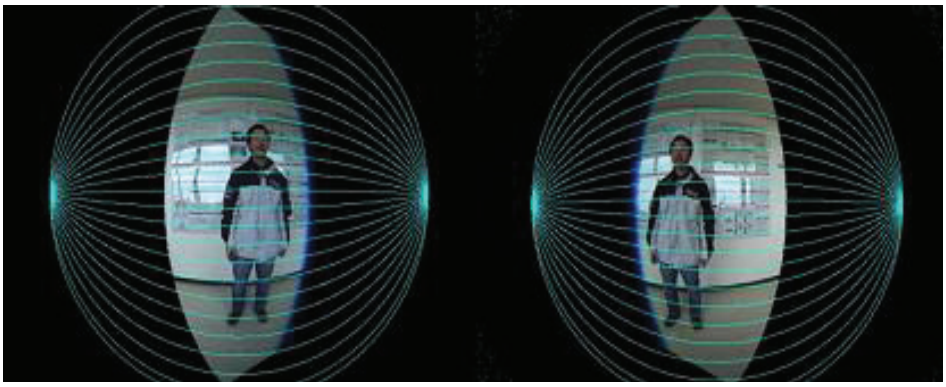


Fig 2.8 The rectified views of Fig 2.6.

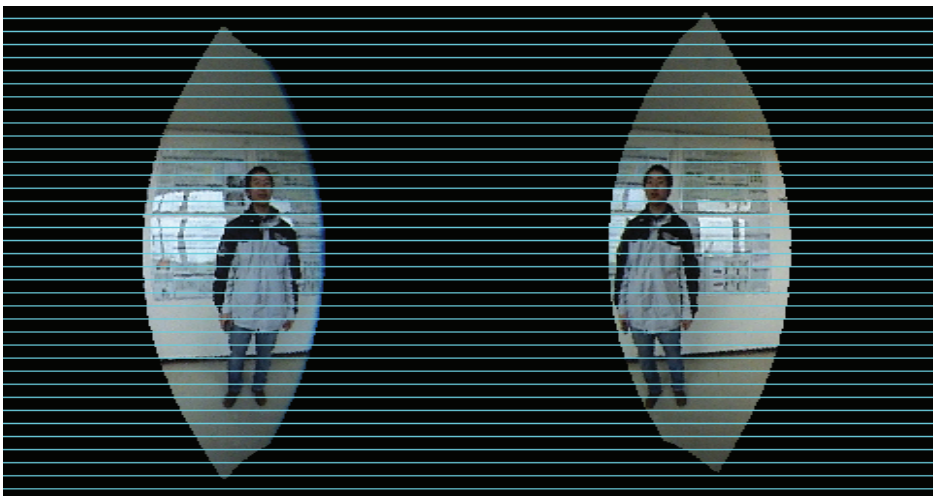
2.3.2.3 Acquisition of dense disparity map

After rectification, the epipolar lines are parallel to each other on the sphere model, like Fig 2.9(a). However, it has high cost of calculating the dense disparity because the point matching is troublesome on the sphere. Here, we transform the rectified

CFOVRs to longitude-latitude representation so that the epipolar lines of the CFOVRs are parallel to the rows of images, as shown in Fig 2.9(b). Then, the fast conventional area-correlation-based method is used to compute the disparity of the CFOVRs, as shown in Fig 2.10. The brighter pixel has a smaller distance from the HVS. Using the disparity values, the 3D information of environments can be computed further according to the disparity of spherical stereo as defined in [7].



(a)



(b)

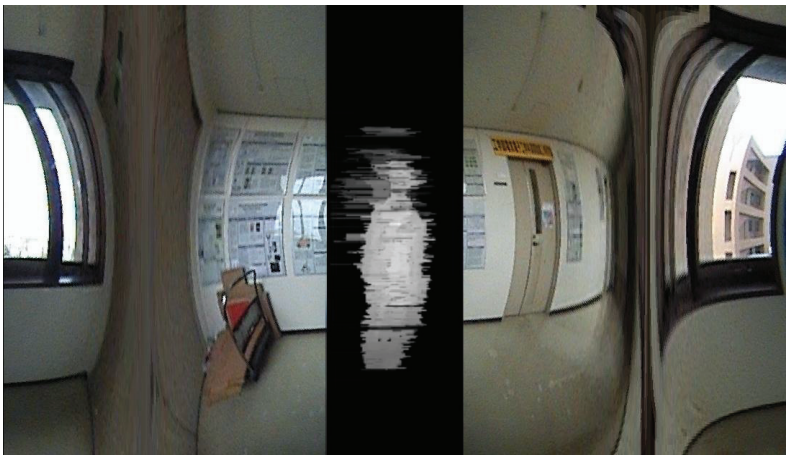
Fig 2.9 The rectified views with epipolar lines. (a) The fisheye views with epipolar line. (b) The longitude-latitude image.

Fig 2.10 The dense disparity map computed using the longitude-latitude image.

2.3.3 The whole vision of HVS

Combining the binocular vision and the monocular vision, we construct the whole vision of our system. Two kinds of display are introduced: a panoramic view with the rectified binocular vision and a display with the dense disparity map. The results are shown below in Fig 2.11.

(a)



(b)

Fig 2.11 The whole vision by the VHS. (a) By using the rectified binocular view. (b) By using the dense disparity map.

2.4 Scene Analysis by the HVS

The configuration of the proposed HVS has been introduced in Section 2.1 as shown in Fig 2.2(a). The pair of hemispherical FOV fisheye images is acquired by the

video cameras, Sony DCR-HC30, mounted with a fisheye conversion lens, Olympus FCON-02. The size of the captured fish-eye images is 640×480 pixels. A pair of sample images captured by the HVS is shown in Fig 2.2(b).

In this section, we present the preliminary experiments of scene analysis by the HVS as an application and also test the effectiveness of the proposed method. The main characteristic of the experiment by the HVS system is that it can both perform motion detection by the whole vision for near full FOV and motion estimation by binocular vision for a relative narrow FOV.

The processing is as follows:

1) Calibrate the intrinsic and extrinsic parameters of the pair of fisheye cameras in the HVS by the method in Section 2.3.1.

2) Do motion detection in the field of the whole vision for the captured video using the background subtraction algorithm [9].

3) Generate the corresponding dense disparity map in the field of the binocular vision, for every frame of the detected motion video above.

4) Compute the depth map from each disparity map until the end of the video. Then, indicate the estimated location of the movement. Finally, we can get the whole depth video of detected moving object's motion.

Here, two experiments based on the processing above are carried out. We describe them respectively in next two subsections.

2.4.1 **Experiment 1: Position Estimation of One Person.**

One person walks along a fixed route, which is set in the overlapped view of binocular vision. A dense disparity map is generated from each frame, and then the 3D information of the person is calculated. The bird's-eye view in Fig 2.12 shows the estimated trajectory indicated in pink, which is projected from 3D position onto the ground. The fixed person route is marked in blue. In addition, the white line illustrates the border of the measurement field. More accurately, we demonstrate the results by the HVS coordinate in Fig 2.13, where θ refers to the angle of the polar coordinate.

In Fig 2.14, the estimation error of the position on the ground is represented based on the polar coordinate system. We can see that the error in the central part is small, and is increasing far from the center. This tendency is consistent with the error analysis of a spherical stereo method reported in the reference [7]. However, the accuracy of the binocular vision still needs to be improved for real applications, and it will be our future work.

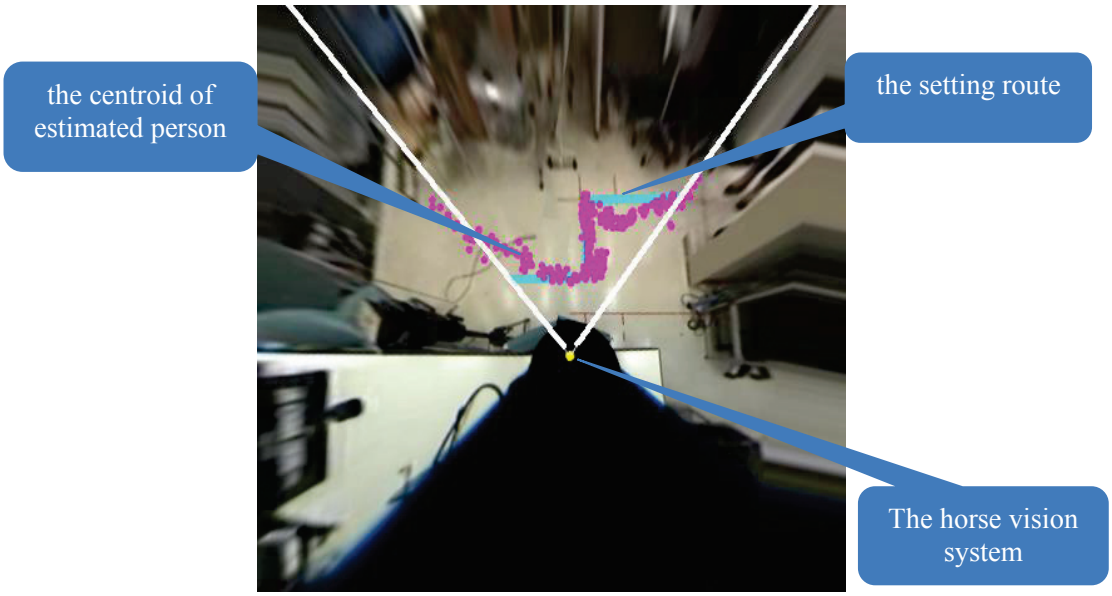


Fig 2.32 The bird-eye view of estimated moving track.

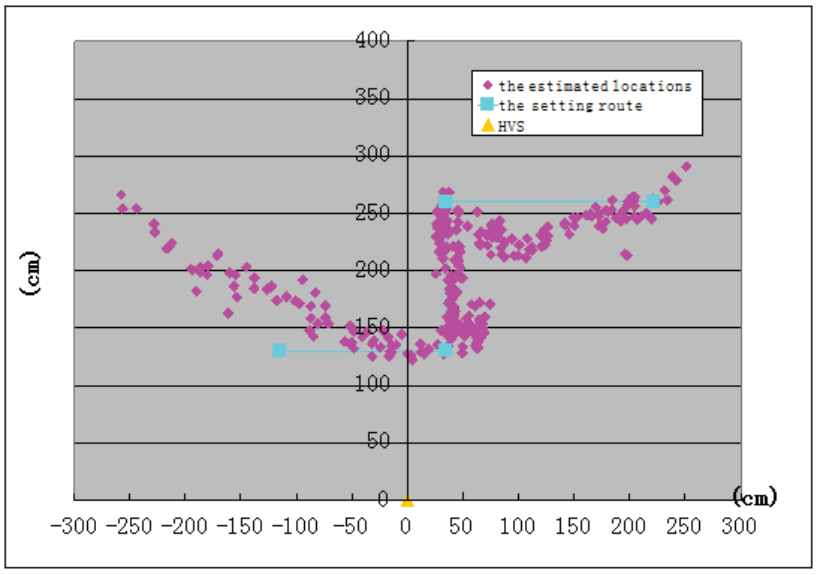


Fig 2.13 Horizontal projection of estimated moving track in the coordination of HVS.

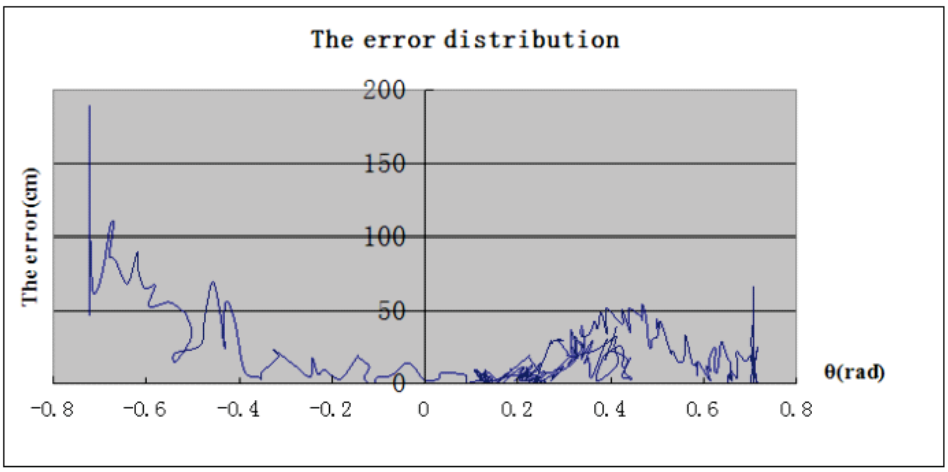
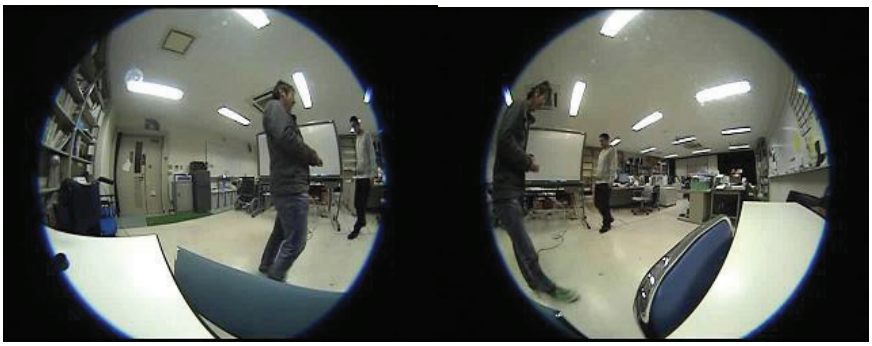


Fig 2.14 The error distribution against θ variation in Fig 2.13.

2.4.2 Experiment 2: Position Estimation of two Persons

Two persons walking from opposite direction in front of the VHS are captured as moving objects in our experiment. We also use the methods described above to deal with the captured video. Fig 2.15 illustrates one pair of original frames and the detected persons. Afterwards the disparity map is computed as shown in Fig 2.16(a). As the result of motion estimation, Fig 2.16(b) shows the area of detected persons in a depth map. The centers of gravity and the location of the VHS are also indicated.

The results indicate that our system is effective and useful for environment analysis. The experiments also demonstrate our method for imitating the horse vision performs well.



(a)

(b)

Fig 2.15 Motion detection of one pair frames. (a) Original images captured by our system. (b) The result of motion detection.

(a) (b)

Fig 2.16 The results of position estimation. (a) The computed dense disparity map. (b) The result of the depth map for motion estimation. Both the location of the VHS (green) and the centroid of the detected persons (pink) are shown in the map.

2.5 Conclusions

In this part, a biologically-inspired vision system, the VHS, like horses' eyes is constructed. It consists of a pair of fisheye cameras which have a hemispherical field of view, respectively, and are laid to overlap each other partially. These characteristics enable the HVS to implement a wide omnidirectional monocular vision and a limited-field-of-view binocular vision simultaneously. We also present an algorithm of identifying the CFOVR and rectifying the CFOVRs for the binocular vision. Finally, the preliminary experimental results of scene analysis based on the HVS are presented to show effectiveness of the proposed method.

For future work, we propose several ideas that extend our current framework.

- Improve calculation accuracy of the stereo disparity in the overlapped field, corresponding to the binocular vision. Furthermore, improve the measurement accuracy of the three-dimensional position in the environment.
- Develop active binocular vision system to achieve object tracking, when detecting an object of interest in the wide field of monocular vision.
- We notice that movement detection method used in the experiment cannot be applied in the case that a person remains stationary within the scene. In order to cope with this situation, we plan to first recognize the candidate regions of a person by the approach of pattern recognition from the monocular vision, and then calculate the 3D position of the person in the field of the binocular vision.

CHAPTER 3

Single-View Scene Understanding

3.1 Introduction

Some approaches [10, 11, 12] are presented to describe the structure of 3D rooms; Reference [13] pays much attention to recovering the objects or furniture, and in [14], an explicit volumetric representation of objects in 3D is incorporated. Furthermore, extending the humans thinking to computers, some work [15] reviews the typical definition of understanding. Implicitly, they exploit the physical interactions between human actions and scene geometry. In addition, the interest in this domain has led to some other related applications, such as precise reasoning about free space [16], an indoor navigating robot [17], real-time indoor scene understanding [18].

On the other hand, omnidirectional vision systems that can provide a wide field of view are gaining popularity (see Fig 3.1). Recent works [19, 20, 21] have showed omnidirectional image sensors can perform well in robot navigation, visual odometry, surveillance and so on. Though the aforementioned approaches of single-view interpretation may be able to be modified to adapt to an omnidirectional camera model potentially, it needs plenty of hard work from down to top, and the practical performance is really suspicious. Because such approaches are designed for the images captured by conventional cameras, which is limited in that it does not use the additional information conveyed by a larger view, for instance, longer line segments and the structure symmetry. It cannot take advantage of all the available cues for estimating the spatial layout.

Another to be emphasized, omnidirectional images, e.g. fisheye images, are prone to be affected by light; it often has less texture or lower resolution; the volume of view information is changeable towards direction and hard to extract. Thus, an omnidirectional image has its own vulnerable point in respect of describing details. The recent series of methods that attempt to model structured scene, estimate the parameters by learning or classing methods, often combined with CRF inference [22], structured SVM [23]. They may fail to be applied to omnidirectional images with less local information.

Fig 3.1 Different fields of view of images captured by a conventional camera and a fisheye camera.

Inspired by good performance of omnidirectional vision in computer vision, some researchers began to consider dealing with the images captured by omnidirectional cameras, which hold a wide field of view, such as catadioptric sensors [24], [25]. However, catadioptric images always lose the ceiling. It is quite different from fisheye

images, which can cover from ceiling to floor in the vertical direction. To the best of our knowledge, such research based on fisheye camera models has been rarely reported up to now. In this chapter, we first explore scene understanding from a fisheye image. We impose the structure by introducing a symmetrical rule which describes geometric constraints. A novel method is given to estimate the spatial layout of rooms only from a collection of line segments. It is much different from the existing approaches, which often evaluate the structure hypotheses to find the best fitting one. We define a main structure as a basic spatial layout. By our method, a preliminary structure is constructed, and then optimized to a main structure.

In essence, the model of all the aforementioned approaches either using conventional perspective images or omnidirectional images obey the similar geometric constraints, and the spatial layouts recovered refer to incomplete structures. Here, we describe these cases as *open geometry*, which implies that some parts of the space are lost (see Fig 3.2(a)). If we want to take advantage of all the available cues in the environment, is there any other more specific and comprehensive model, which enables computers to predict the entire structure? It seems what we have to do is to break through the limitation of open geometry.

We pay our attention to full-view images. Some examples are given in Fig 3.3. One of well-known applications in practice refers to Google Street View. Users are allowed to visit cities on Internet with an immersed sense. The feature of these full-view images can be included as that you are able to enjoy the entire scene at the camera

location without losing any information. In other words, the spatial layout recovered from the images is complete, composing a “close” space. In this thesis, we call the geometric constraints of this model *close geometry* in contrast with the conventional *open geometry* mentioned above (see Fig 3.2(b)). We also employ the characteristics of close geometry to explore indoor scene understanding from a single full-view image. The proposed close geometry is tested in comparison with the conventional open geometry.

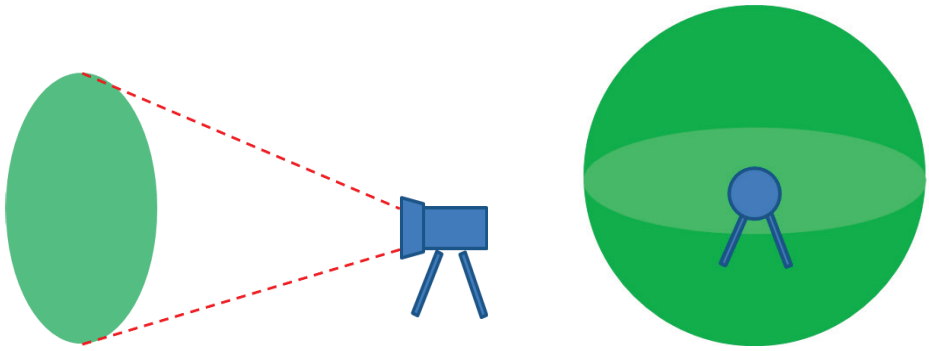


Fig 3.2 The images of two types of geometry. (a) Open geometry refers to partial scene. (b) Close geometry refers to complete scene.

(a)

(b)

Fig 3.3 Examples of full-view images. (a) Two half scenes captured by fisheye cameras. (b) Longitudinal-latitudinal representation. (c) An intuitionistic spherical display by CG.

This chapter is organized as follows. Section 3.2 introduces the related research. Section 3.3 describes the model used in our approach. We explore scene understanding for a fisheye image in Section 3.4. Then, we explain the procedure to estimate structure from a single full-view image. Finally, we draw a conclusion.

3.2 Related Research

3.2.1 Single-View Geometry Estimation using Perspective Images

The basic problem for indoor understanding is prediction of the room layout given a single image. Over the past few years many approaches have been developed to tackle the problem under the Manhattan world assumption [26]. One of the common techniques is a framework of basic geometric analysis. As summarized by Tretyak et al. [27], the bottom-up steps involve a composition of geometric primitives spanning different layers from low level (edges) over mid-level (line segments, lines and vanishing points) to high level (zenith and horizon). Beyond this pipeline paradigm, most recent approaches [22], [23], [28] address this challenging problem associated with a large set of local features, such as color, texture, location. However, we notice that an image with wide field of view is prone to be affected by light, and it often has less texture or lower revolution in respect of describing details. These methods may fail to be applied to omnidirectional images with less local information.

On the other hand, some work focuses on geometric reasoning. Lee et al. [29] demonstrates that structure recovery from line segments is comparable with the methods

using full image appearance. Flint et al. [30] develop a dynamic programming to efficiently search all feasible indoor models. In order to choose the best hypothesis, both of two methods generate a map of orientations with different planar regions.

3.2.2 **Single-View Geometry Estimation using Omnidirectional Images**

Omnidirectional cameras refer to the vision sensors that can observe a wide field of view. Though the use of omnidirectional cameras has increased among the community of computer vision considerably, not so much attention is paid to scene understanding. The study has mainly involved the detection of geometric primitives in man-made environments [31], [32], [33]. As they argue in their different works, omnidirectional vision could be privileged in estimation of the room layout, because it provides two important properties. First, the wide field of view allows minimizing the possibility of fatal occlusions and partial views, benefiting the extracting lines. Second, the vanishing points usually lie inside omnidirectional images. It may bring much convenience to the structure recovery processing.

Inspired by David C. Lee et al.'s method [29], Ozisk et al. [24] show a similar processing. Omedes et al. [25] suggest a quicker way to recover the spatial layout of a scene. Both of their work depends on a large set of accurate detected lines. Sometimes it is hard to carry out extraction step because of low resolution or bad light condition. They cannot cope with complex environments.

To be emphasized, all the methods mentioned above are designed for the problem of images with limited field of view, corresponding to parts of indoor environment, which result in visually open boundary condition, referring as *open geometry*.

3.3 A Model for Indoor Scene

Our goal is to extract the main bare structure of indoor environment ignoring all objects within rooms. That is to say, we aim to recover boundaries rather than facilities or furniture, unless they are large enough to be regarded as an inseparable part of rooms. We also estimate surface layout, describing the orientations of every point in the scene. Each image pixel is classified as belonging to floor surface, ceiling surface, or wall surface. To summarize, geometry estimation is referred as a problem of structure estimation and surface labeling.

3.3.1 Sphere Camera Model

Generally, it is convenient to turn the images captured by various types of cameras with a single viewpoint into an equivalent sphere. Considering surrounded walls without order, we employ sphere model to handle omnidirectional images, which describes a panorama view with no need to concern about specific visual sensors. Under this model, there are some important projection properties, as depicted in Fig 3.4 (left). A point $M(X, Y, Z)$ in 3D space is projected as a spherical point $m(x, y, z)$ onto a unitary sphere. The projection of a line segment L in 3D is converted to a part of great circle l,

represented by a unit normal vector $\vec{n}$. Several great circles associated with a pencil of 3D parallel lines intersect at two antipodal points, which correspond to the vanishing points.

With any challenging problem, assumptions are generally imposed to add constraints that make the problem tractable. One such assumption is that most planes are aligned with three main world axes, called Manhattan world assumption introduced by Coughlan and Yuille [26]. It states that the scene is built on a Cartesian rig. Thus, we can identify three mutually orthogonal vanishing points in the sphere aligned with the three dominant directions in the world. For convenience of description, we assume that the vanishing point in absolute y-coordinate, denoted as v_v, corresponds to the vertical direction. The other two vanishing points v_r and v_l are computed from horizontal lines of rooms, as depicted in Fig 3.4 (right). If vertical lines in the world do not appear vertical in the image, we can rectify them by easily rotating the sphere.

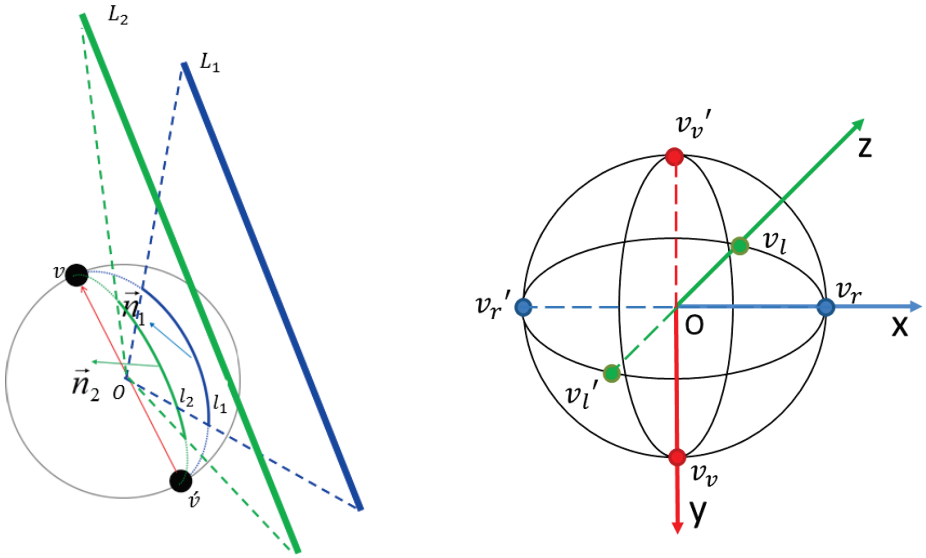


Fig 3.4 The sphere model for full-view images. Left: spherical projection. Right: vanishing points and World Coordinate.

3.3.2 Indoor World Model

Under the Manhattan world assumption, most planes can be labeled in terms of $\{r, l, v\}$ corresponding to three vanishing points. Indoor environments usually have a single floor plane and a single ceiling plane. Successive walls are situated between ceiling and floor. Each wall goes along one of two horizontal orientations. We classify indoor scene into a set of $\{floor, wall(o), ceiling\}$, referring as regions. $wall(o)$ denotes the wall with orientation o, $o \in \{r, l\}$.

The layout of rooms is represented as the combination of floor-wall boundaries, ceiling-wall boundaries and wall-wall boundaries, referring to line segments. Sometimes one can estimate the surface labels given the most likely spatial layout candidate; and

sometimes the surface labels, in turn, allow robust layout estimation. Therefore, the best structure of rooms must have both accurately estimated spatial layout and labelled surface. Combining these two types of models, we propose the “indoor world model” as a useful approximation for indoor scenes.

3.4 Estimating the Structure of Rooms from a Single Fisheye Image

In this section, we focus on the problem of estimating the spatial layout of rooms from a single fisheye image. Considering the wide field of view of fisheye cameras, we introduce a structure symmetrical rule which describes geometric constraints. A method is given to estimate and recover the preliminary spatial layout of room only from a collection of line segments extracted from a fisheye image. Then, an orientation map of structure is generated. Finally, we refine the spatial layout to obtain the main structure. The experiments demonstrate that our approach based on geometric reasoning can be used to estimate the structure of indoor scene from a single fisheye image.

The most desirable property of our model is symmetry. We extend the symmetry concept argued by David C. Lee et al. [29]. Man-made buildings always have symmetric floor and ceiling shape. As depicted in Fig 3.1, compared to traditional cameras, which may not contain both floor and ceiling plane in a single image, a complete scene structure with a floor, a ceiling and walls is always visible in a fisheye image. It gives us a hint that we can recover a wall-floor boundary from the corresponding wall-ceiling boundary under the symmetry criterion or vice versa. The researchers so far attempt to distinguish segments of the layout and segments of clutter in cluttered rooms. However, they ignore a

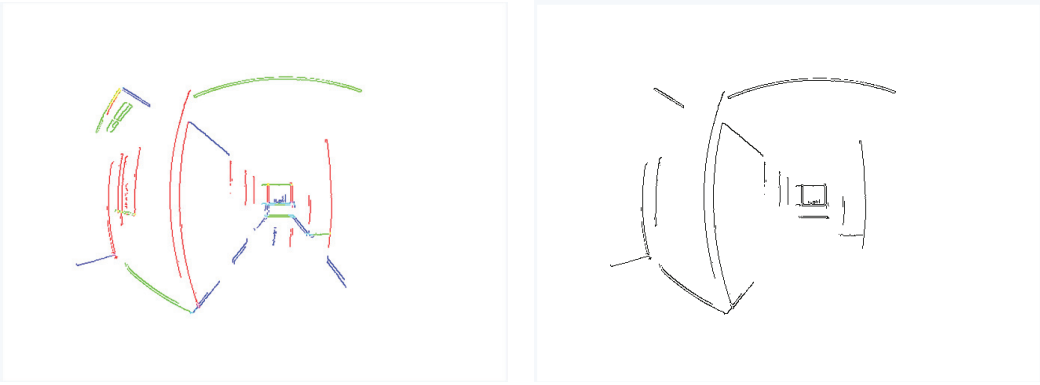
serious problem that segments attained from an edge map, are easy to be missed, or be occluded by objects. The imperfect line detection becomes a crucial problem of such processing. In our case, the layout segments are visible even when some parts are occluded. What is more, we can use the geometric symmetry to infer the location of wall-floor boundaries by wall-ceiling boundaries, which are rarely occluded.

3.4.1 **Preliminary Spatial Layout Estimation**

Walls between ceiling and floor are crucial parts of the whole room. Most of them can be divided into different wall towards orthogonal orientations by vertical lines. Therefore, we focus on vertical lines to generate the structure. A vertical line and a horizontal line next to each other are likely to come from the same wall surface in the 3D space. Since we know the corresponding vanishing point for each line, we can represent the wall plane by a pair of vertical line and horizontal line. Walls are successive in an image from left to right. It implies that if there is a set of wall constructed by a series of successive line pairs, which contains wall-floor, or wall-ceiling boundaries, we can build the whole structure of a room.

Fig 3.5 (a) depicts line segments extracted from the fisheye image in Fig 3.1. We start by picking up the first line pair from left, referring to the first wall. Then, we search for line pairs that share the same line with the ones already found and add them to our wall set. If the processing is broken up by no shared lines because of line missing in the extraction step, we extend the horizontal line segment of current rightmost pair to right

towards its vanishing orientation until it meets another pair. By repeatedly attaching more pairs, we create a structure hypothesis (see Fig 3.5 (b)).



(a) **(b)**

Fig 3.5 A structure hypothesis. (a) Line segments. (b) A structure hypothesis constructed by a series of successive line pairs.

Note that the structure hypothesis may have some lines which are not the boundaries of wall-ceiling or wall-floor. In addition, as we present before, missing lines often take place. So we need a way to remove inappropriate pairs and recover the complete successive boundaries across the image. We implement our method by applying the geometric symmetry criterion. The process is illustrated in Fig 3.6.

1) Connect the horizontal lines next to each other. We focus on horizontal lines only; complement them in order to obtain a series of successive wall. Extend either of the two horizontal lines next to each other decided by their orientations if they do not intersect. In the case of missing boundary, we connect the adjacent breakpoints instead of

extending lines. Then insert the vertical lines starting from the junction to its corresponding point. It obeys the rule that walls are defined by boundaries of wall-ceiling (wall-floor) and vertical lines.

2) Make up the horizontal lines. We check if the lines above the horizon have the corresponding ones below, and recover them if not. Do the same for the ones below the horizon. The result is shown in Fig 3.6 (b).

3) Check the vertical line. We turn our attention to the vertical lines in the hypothesis. Remove the ones that have no intersection with the horizontal lines, since it is much likely to belong to objections other than the layout. In addition, if there is one that does not intersect the horizontal lines at line endings, it tells us a boundary may be missing at the endings. We need to make up the boundary by searching for the nearest horizontal line in the collection of line segment (Fig 3.6 (c)).

4) Deal with the left and right borders. We hope the boundaries across the whole image. Unfortunately, blank space often exists between the leftmost horizontal line and the left border of an image, so as the right side. We extend the lines following the way in 1). Then, make up the horizontal lines using the method in 2). To be noticed, the extending lines near the borders cannot cross each other when the symmetric criterion is used. This rule holds because there is only one boundary of the same wall and ceiling (floor) all the time. If it happens, keep the outside one from the crossing.

Finally, we are able to obtain the preliminary spatial layout, showed in Fig 3.6 (d). Though it still has some lines which are not the boundaries, it provides enough information for us to compute an orientation map.

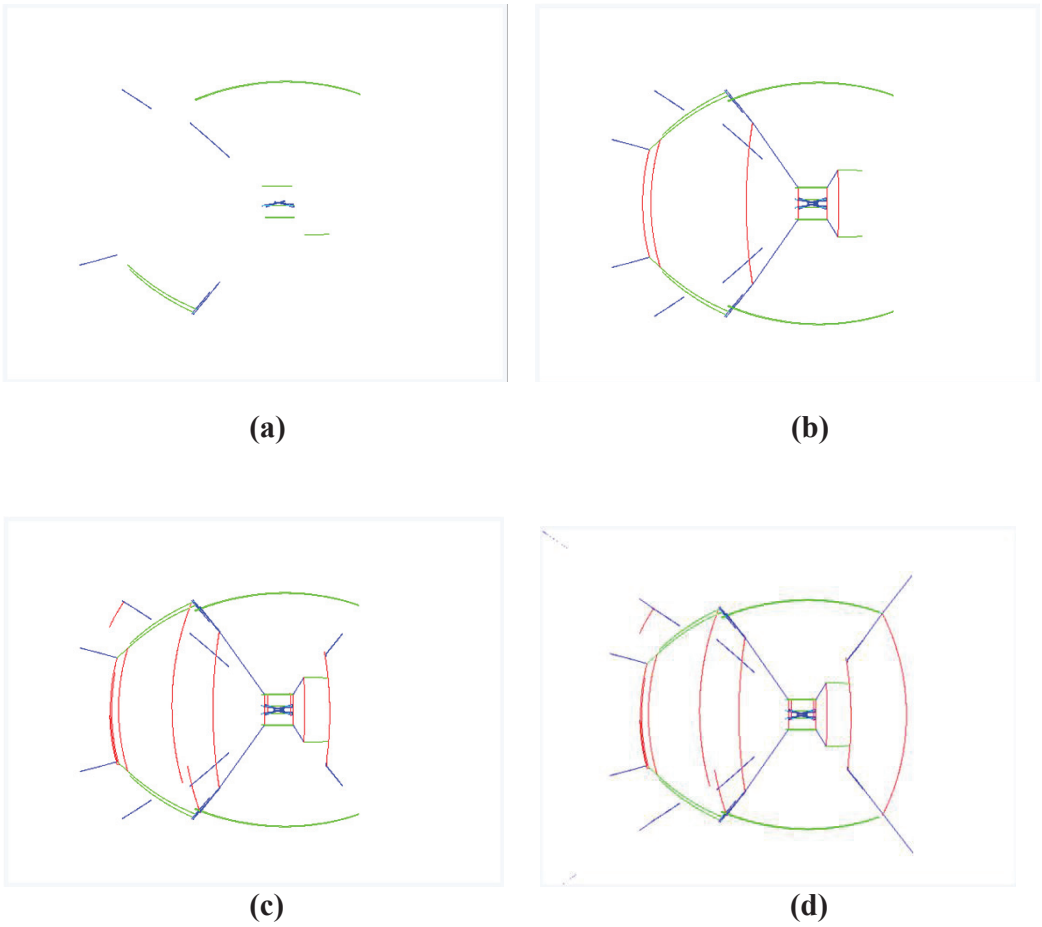


Fig 3.6 Process of preliminary spatial layout estimation. (a) The horizontal lines of the structure hypothesis in Fig 3.5(b). (b) Recover the horizontal lines. (c) The result of Step 3. (d) The preliminary spatial layout.

3.4.2 **Computing the Orientation Map**

An orientation map is referred as the local belief of region orientation. As interpreted in [29], a pixel is supported by two line segments having different orientations, and the pixel orientation is perpendicular to the plane of the two lines. Thus, the whole regions are classified to surfaces towards three possible orientations. Compared with computing from a set of line segments, it is much easier to do it from a spatial layout in our case. The process is described as follows.

Assume a small region s is surrounded by a set of line segments $Ls(ls_{1,x}, ls_{2,x}, \cdots, ls_{n,x})$, where $x \in (r, l, v)$, denotes three orientations. Let $p_s(x|Ls)$ be the likelihood of s belonging to the orientation x.

$$p_s(x|Ls) \propto p(P_y|Ls), \quad y \in (r, l, v), \quad y \neq x \tag{3.1}$$

Where $p(P_y|Ls)$ is the probability of the plane defined by the lines perpendicular to the orientation x. It can be computed by:

$$p(P_y|Ls) = \sum_{i=1}^{n} (\alpha N(l_{i,y}|Ls) - \beta N(l_{i,x}|Ls)) \tag{3.2}$$

Where $N(l_{i,y}|Ls) = 1$, if l_i belongs to the plane of orientation y; and $N(l_{i,y}|Ls) = 0$ otherwise. So as $N(l_{i,x}|Ls)$. α and β are the constants.

For every small region, we compute the likelihood of three different orientations, and decide the region orientation by choosing the maximal one. Fig 3.7(a) shows the orientation map with regions colored in red, green, and blue.

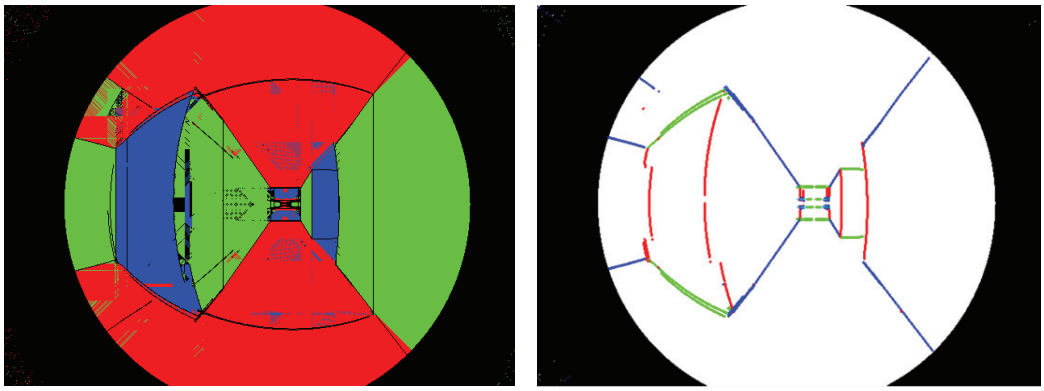
3.4.3 **Refining the Spatial Layout**

Once the orientation map is generated, it is not difficult to evaluate the scene structure. Here, the main structure is interpreted as the boundaries that can divide regions upon their orientations.

Take a vertical line segment in our preliminary spatial layout estimation as an example. Investigate the areas the vertical line is dividing. To make orientation noise robust, we define the main orientation of a region as Z , which satisfies the constraint.

$$\frac{Area(z)_{\max}}{Area(r)+Area(l)+Area(v)} > \delta \tag{3.3}$$

Where $z \in (r, l, v)$, and $Area(z)_{\max}$ donates the maximal area with orientation z in the region. δ is a threshold. If the left region has the same main orientation with the right region, it is much possible that the two regions need to be merged as one and the vertical line is not a real boundary. So we remove it from our structure. We check every line segments of preliminary spatial layout to decide if they are real boundaries. The final refined main structure is shown in Fig 3.7 (b).



(b) (b)

Fig 3.7 The result of the refined spatial layout. (a) Orientation map. The regions are colored according to their orientation. (b) The main structure after refining.

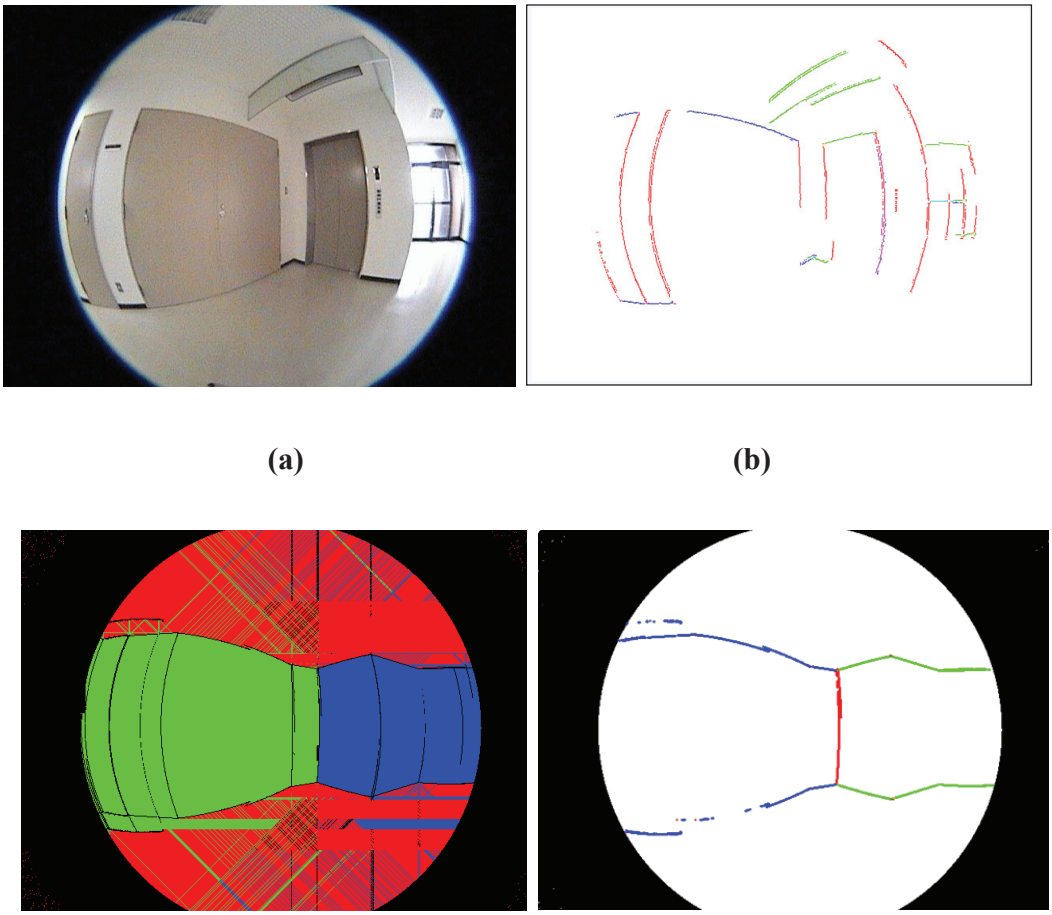
3.4.4 Experiments

An experimental result concerning the proposed method is given. Fisheye images are acquired by the video cameras, and the intrinsic parameters are calibrated beforehand. We use the following strategy for extracting line segments. First, the vanishing point detection of [21] is employed to obtain the preliminary orientations of the Manhattan world. Next, we modify line detection method of [19]. The split and merge algorithm is applied to only extract the line segments with the Manhattan orientations approximately. More formally, this constraint can be expressed as follows

$$\overrightarrow{n_l} \cdot \overrightarrow{v_i} < \lambda \tag{3.4}$$

Where $\overrightarrow{n_l}$ is the normal vector of the great circle corresponding to line l, and $\overrightarrow{v_i}$ denotes the normal vector of the vanishing point v_i. λ is a threshold. Then, refine the vanishing points using the line segments. After line extraction, we test our approach. Fig 3.8 illustrates one of the examples.

The results indicate that the presented method is able to estimate structure of indoor scene. In particular, some missing segments, or occluded boundaries could be recovered according to the symmetry.



(a) **(b)**

(c) **(d)**

Fig 3.8 An example of experiments. (a) The fisheye image. (b) Line segments. (c) Orientation map. (d) The main structure.

3.5 Indoor Scene Understanding from a Single Full-View Image

So far nearly all existing methods either using perspective images or omnidirectional images involve just partial scene, which leads to the recovered spatial layouts referring to incomplete structures. These cases are summarized as *open geometry*. On the other hand, a full-view image results in a visually close boundary condition, called *close geometry*. In this section, we advocate a new model based on close geometric constraints to explore indoor scene understanding from a single full-view image. In our system, we optimize the score function of the structure model to a linear presentation and develop a novel algorithm without necessity of implementing any orientation estimates beforehand. Furthermore, only fewer corners are required, which may enable our algorithm to run efficiently.

3.5.1 Close Geometry

Though a full-view image can also be segmented as some independent images, which follow open geometry like conventional ones, our model obeys more strict close geometry. It is an appropriate representation to approximate a real room. The main additional differences are, first, ceiling-wall or floor-wall boundaries are completely visible as long as we see by ourselves at the camera location. Walls go around making a circle, no longer left-to-right. That means walls are equivalent without a start or an end, avoiding the trouble of poor estimation on the edge in conventional images. Second, the wide field of view allows minimizing the possibility of fatal estimation caused by occlusions and partial views. Suppose an extreme case, when an object covers the whole

conventional image (if it is close enough to the camera), it is impossible to carry out any estimation. However, we may still be able to reconstruct the room regarding the conventional image just as a part of our full-view image. Third, man-made buildings always have symmetric floor and ceiling shape with constant ceiling height. Here, the planar homology used in [30] is adapted to sphere model, which describe the mapping H between the image locations of ceiling points and the corresponding ones in floor plane. We use arc length instead of Euclidean length due to our sphere model. Once H is obtained, we can recover a floor-wall boundary from the corresponding ceiling-wall boundary, or vice versa.

Another criterion is addressed among "wall" labels. Walls are situated between a ceiling and a floor successively. The vanishing points of horizontal lines must be located at the walls with different orientation (see Fig 3.9), which can be described as follows:

$$v_r \in wall(l), \quad v_l \in wall(r) \tag{3.5}$$

This constraint holds because of the definition of vanishing points as the geometric place where parallel lines appear to converge. Obviously, for the conventional images, this close geometry is not always guaranteed because of limited field of view.

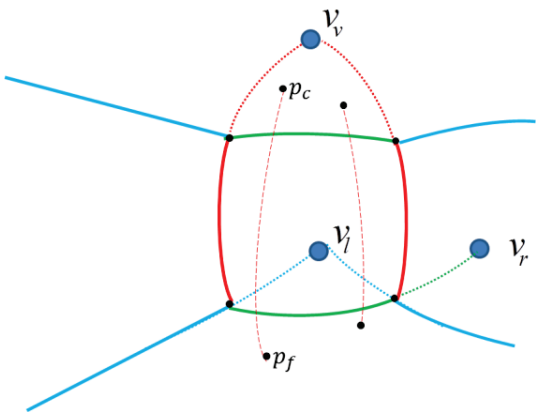


Fig 3.9 The criterions of close geometry. The pair (p_c, p_f) refers to corresponding points in the ceiling and floor plane. Vanishing point v_l lies in the wall with orientation towards v_r.

3.5.2 Estimating Spatial Layout

Here, we introduce our algorithm based on close geometric constraints to estimate spatial layout from a single full-view image. A set of corners is first generated and the expression of structure is given. Then, we find the layout which fits line segments best by exploring the solution to the maximization of the structure formulation. Finally, we recover the spatial layout and label the surface of indoor rooms.

3.5.2.1 Preliminary work

We start from a collection of line segments and vanishing points under the Manhattan world assumption. Any method capable of extracting line segments and optimizing the vanishing points could be applied.

Due to close geometry, in particular of ceiling-floor symmetry, either a point in ceiling-wall boundary or in the floor-wall boundary is sufficient to specify the other corresponding one. Without loss of generality we choose to present two corresponding boundaries only by the floor-wall boundary. In a full-view image, the longest horizontal line always lies in boundaries, if it is not occluded too heavily to be visible. According to this assumption, we may obtain a preliminary structure containing the longest horizontal line. The purpose of this step is to enhance the possibility of layout estimation associated with the process afterwards. To be emphasized, it is not vitally necessary. We set a safe threshold for checking the length of lines to ensure what we choose really belongs to the floor-wall boundaries. In the case of failing check, always caused by occluded objects, this step will be omitted and it will have no impact on spatial layout estimation later.

3.5.2.2 Generation of corners

We are now ready to generate corners. We agree with Omedes et al. [25] that the detection of vertical lines is more robust and less susceptible to noise than horizontal lines. Moreover, most of walls towards orthogonal orientations can be divided by vertical lines. Therefore, we extract corners according to vertical lines. We assign a rule that one vertical line can contain only one corner. Corners are recognized in three cases in order,

as illustrated in Fig 3.10. First, define the intersection of a horizontal and a vertical line at their crossing point as a corner. Second, extend vertical line, and detect the intersection if it crosses over a horizontal line. Otherwise, extract the ending point of vertical line itself. However, in practice, not all the vertical boundaries of walls can be extracted. For this reason, we also regard the ending points of horizontal lines as corners, except the ones lying in X-Y plane or Y-Z plane since they are against the close geometry in (3.5).

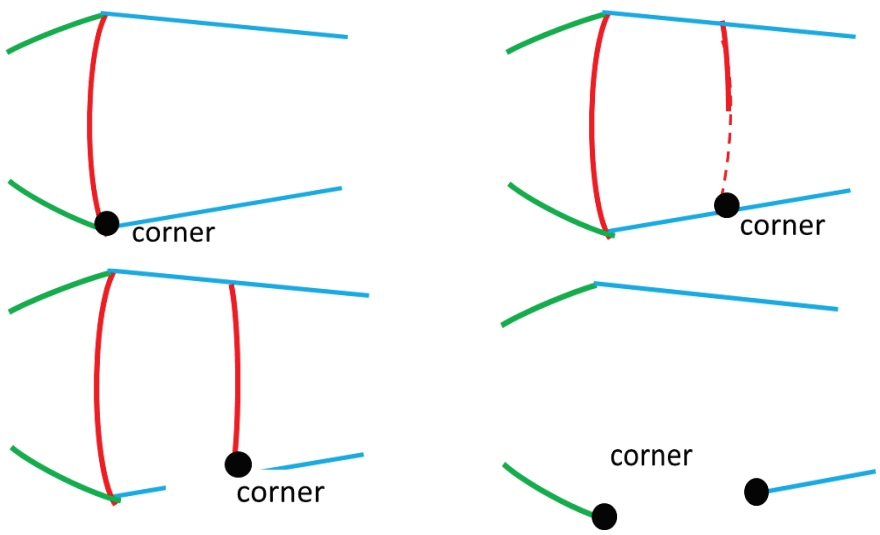


Fig 3.10 Three cases of corners generated from vertical lines and one case generated from horizontal lines.

In order to express explicitly, we project these corner into X-Z plane of the sphere model, and order them clockwise starting from the absolute x-coordinate, as shown in Fig 3.11. We denote these corners as a set $Q(q_1, q_2, \cdots, q_n)$. Note that all these operations are

complemented against floor-wall boundaries, so we only focus on the low ending points of vertical lines and the horizontal lines below horizon. The corners of ceiling-wall boundaries can be achieved easily by the similar way if necessary.

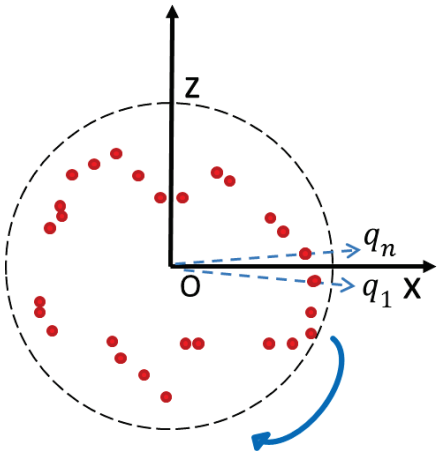


Fig 3.11 Corners are ordered clockwise according to their projections in X-Z plane.

3.5.2.3 Problem formulation

In indoor images, floor-wall or ceiling-wall boundaries often go towards two orthogonal horizontal vanishing points. However, a special case occurs when one wall is in front of another but appears to be adjacent in the image, as illustrated in Fig 3.12. Thus segments of the ceiling-wall or ceiling-wall boundaries in fact are vertical lines. In summary, boundaries have three possible orientations towards mutually orthogonal

vanishing points. Suppose that some of the corners are the points really in floor-wall boundary, constituting a set of key points $P(p_1, p_2, \cdots, p_m)$. As we state above, ceiling-wall boundaries can be computed from floor-wall boundaries. Hence, a floor-wall boundary $W(p_i, p_j, o)$ is determined by two key points p_i , p_j from P comprising orientation o , $o \in (r, l, v)$ (Fig 3.12). It also means that the wall is specified. We represent one indoor structure hypothesis of layout $G(W_1, W_2, \cdots, W_t)$, as a close circled sequence of floor-wall boundaries $W_1, W_2, \cdots, W_t$.

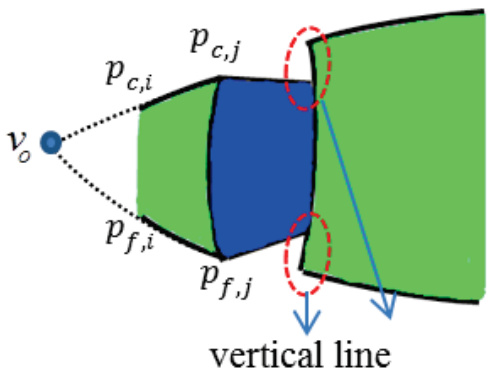


Fig 3.12 A special case that some segments of floor-wall boundaries are vertical lines. A wall is determined by two key points $p_{f,i}$, $p_{f,j}$ and orientation o.

Different sets P may define different structure hypotheses. Obviously, not all hypotheses are physically realizable, so we need a method to identify the correctness and evaluate the probability. In brief, the basic idea is to evaluate globe structures by applying

the local information. Other than conventional orientation map, we come up with a novel scheme to test hypotheses only by a collection of line segments extracted at the beginning, since line segments coming from edge map provide us desirable local geometric reasoning. Considering the representation of indoor structure, we draw a conclusion that a hypothesis is feasible if all of its key points are feasible, and the feasibility is dependent on the line segments.

Given a structure $G(W)$ and a set of line segments $Ls(ls)$, a score function $C_{ls}(ls_i, G)$ indicates the fitness of the structure G to segment ls_i. It can be computed by the sum of pixel score $C_x(x_j, G)$ over all the pixels in ls_i ,

$$C_{ls}(ls_i, G) = \sum_{x_{i,n} \in ls_i} C_x(x_{i,n}, G) \tag{3.6}$$

where $x_{i,n}$ is a pixel in ls_i. For efficiency, we parameterize $C_x(x_{i,n}, G)$ as a simple binary model,

$$C_x(x_{i,n}, G) = \begin{cases} 1, & if\ x_{i,n} \epsilon G \\ 0, & otherwise \end{cases} \tag{3.7}$$

The final score of a structure hypothesis is thus given by investigating all the segments,

$$C(G) = \sum_{ls_i \in Ls} \lambda C_{ls}(ls_i, G) \tag{3.8}$$

where λ is a constant, regulating the weight of scores. Since the structure is constituted by a series of walls, we can rewrite the final score as follows:

$$C(G) = \sum_{j=1}^{t} \sum_{ls_i \in Ls} \lambda C_{ls}(ls_i, W_j) \quad \textbf{(3.9)}$$

Here, $C_{ls}(ls_i, W_j)$ is similar to the $C_{ls}(ls_i, G)$ specifically measuring wall W_j instead of the structure G.

The process of searching for the hypothesis best fitting the structure is converted into a maximization problem. That is to maximize (3.9) to obtain a structure with greatest score. More formally, we have

$$G^* = argmax(C(G)) \quad \textbf{(3.10)}$$

3.5.2.4 Algorithm of estimating the spatial layout

It is impossible to generate a hypothesis unless we know a prior of key point sets in advance. So how to extract key points from a corner set becomes crucial. We cope with this problem by combining the process of deciding key points and pursuing the maximal score together, which makes our algorithm much more efficient and robust.

As preliminary work mentioned in Section 3.5.2.1, we carry out a test to find if there exists a long enough horizontal line in the scene. If so, we can obtain a preliminary structure even containing one or two line segments. The next step depends on the detected result. This is because, in practice, we always see two types of images: one is the edges of either floor-wall or ceiling-wall are distinct; the other is the edges of both floor-wall and ceiling-wall are obscure. For the latter case, it is not often seen but sometimes leads to fatal failure of estimation. In that situation, long horizontal line may not be detected. So that λ is assigned the same for every line segment.

On the other hand, in full-view images, most times there are a few long boundary edges distinct enough to be detected more or less, which correspond to the former case. That allows us to take advantage of the extracted preliminary structure to enhance evaluation process, increasing the reliability of our final result. For the score function, since additional information about structure is obtained, we represent structure hypotheses using boundaries based on the preliminary structure. If the extracted long boundary belongs to ceiling-wall, we express $G(W)$ by ceiling-wall boundaries, otherwise we apply floor-wall ones as we expressed before. As a result, (3.9) is rewritten as follows:

$$\mathrm{C_{ls}}(\mathrm{ls_i},\mathrm{G}) = \sum_{\mathrm{j}=1}^{\mathrm{t_1}} \sum_{\mathrm{ls_i}\in\mathrm{Ls}} \lambda_1 \mathrm{C_{ls}}\left(\mathrm{ls_i},\mathrm{W_{1,j}}\right) + \sum_{\mathrm{j}=1}^{\mathrm{t_2}} \sum_{\mathrm{ls_i}\in\mathrm{Ls}} \lambda_2 \mathrm{C_{ls}}\left(\mathrm{ls_i},\mathrm{W_{2,j}}\right) \quad \textbf{(3.11)}$$

where $W_{1,j}$ denotes the boundary hypothesis represented by the corner set Q_1 which is generated from the preliminary structure, while $W_{2,j}$ denotes the boundary hypothesis represented by other corners from set Q_2. We believe that the corners of the preliminary structure are more likely to lie in boundaries, essentially referring to key points. Thus, we give them a large weight, $\lambda_1 > \lambda_2$.

One simple way to solve this problem is just starting from (3.10) directly. Enumerate over all possible structure hypotheses by picking up corners as key points randomly. Then check the scores using the set of line segments by (3.11). However, that will be time-consuming and the result in practice is really suspicious. If we view (3.10) from a different perspective, our goal is to find out a series of closed boundaries which

pass through as many as possible segments. The corners are known and their locations are unchangeable. It holds the rule that a corner is independent from others adhering to only one wall. Flint et. al [30] also demonstrate that the placement of each wall is "conditionally independent" of the other walls given its left and right neighbors. If we can divide all the corners in Q_1 and Q_2 into some subsets $Q_{1,s_1}, Q_{1,s_2}, \cdots Q_{1,s_m}$ and $Q_{2,s_1}, Q_{2,s_2}, \cdots Q_{2,s_n}$ appropriately, with that each subset corresponds to one boundary segment, our problem is rewritten as follows:

$$\begin{cases} C\left(Q_{1,s_j}\right) = \sum_{ls_i \in Ls} \lambda_1 C_{ls}\left(ls_i, W_{1,j}\right) \\ W_{1,j}^* = argmax(C\left(Q_{1,s_j}\right)) \end{cases}$$

$$\begin{cases} C\left(Q_{2,s_j}\right) = \sum_{ls_i \in Ls} \lambda_2 C_{ls}\left(ls_i, W_{2,j}\right) \\ W_{2,j}^* = argmax\left(C\left(Q_{2,s_j}\right)\right) \end{cases} \quad \textbf{(3.12)}$$

Finally, the resultant structure is

$$G^* = \sum_{j=1}^{m} W_{1,j}^* + \sum_{j=1}^{n} W_{2,j}^* \quad \textbf{(3.13)}$$

We see that it is decomposed into a series of sub-evaluation of walls. So the next step is to build boundary hypotheses from subsets of corners and find the key points.

Let us turn our attention to the criterion in term of (3.5). It describes the relation of vanishing points and the walls. In another words, it can be interpreted as that parallel planes have to be one at each side of the imaginary line formed by joining their two

corresponding vanishing points. This constraint gives us a hint that one floor-wall boundary is restricted to be situated within two adjacent horizontal vanishing points. To be noticed, under this assumption, though a long boundary is allowed to be regarded as two with the same orientation, it enables us to classify the corners in the limited region into subsets and specify the key points to generate the most possible boundaries. As we states before, boundaries have three possible orientations towards mutually orthogonal vanishing points. A line segment in 3D is converted to a part of great circle under sphere model. It is appropriate to identify a great circle in a unitary sphere just from two points. Thus, boundaries can be decided by a corner and one of vanishing point.

All the corners and two pairs of horizontal vanishing points are ordered clockwise starting from the absolute x-coordinate. First, compute the scores of two boundary hypotheses with different horizontal orientations starting from any corner to the next corresponding vanishing point by (3.12). We recommend to pick up a corner from Q_1, because it is more likely to be a key point. If there is no preliminary structure detected, RANSAC is applied to find a line with passing through most corners and one of corners is picked up as the starting point. Then, do the same for the vertical hypothesis. The one with higher score is chosen as the resultant boundary, and the last corner belonging to this boundary is considered as a key point. Fig 3.13 shows an example of this procedure. Assuming that the corner q_i is a key point, and next key point is required. We generate three boundary hypotheses from q_i towards three vanishing points, respectively. Compute total scores of these hypotheses by summing up every score of segments belonging to the

boundaries. We can see that the one towards vanishing point v_l passes through more pixels in segments than other two boundaries. Therefore, the last corner q_{i+4} belonging to this boundary is recognized as a key point, while $W(q_i, q_{i+4}, l)$ is regarded as a part of the most feasible structure during this region. Repeat the process until all the boundaries are extracted. Finally, we obtain the entire structure by (3.13).

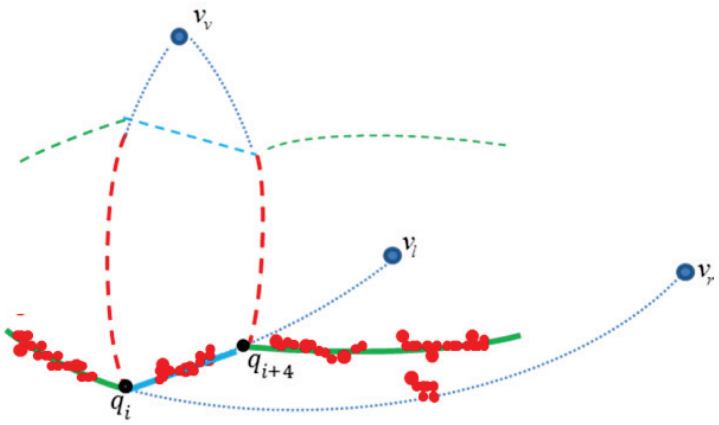


Fig 3.13 The procedure of computing the most feasible boundary. Red points indicate the pixels of line segments.

Due to our scheme, the boundary hypotheses in a region are evaluated at most three times corresponding to three vanishing points for each unique subset of corners. Moreover, there is no need to carry on iterative computation for the former determined boundaries. The overall complexity of our algorithm depends on the total number of corners coming from line segments, with a little relation to the indoor structure. So

different from other state-of-the-art methods, we present an approach that deterministically finds the global solution and exhibits computational complexity linear in scene complexity and the amount of segments.

3.5.3 Experiments and Results

Experimental results concerning the proposed method are presented. Line segments and vanishing points of scene are required in our method. Here, we modify the well-known Canny edge detection algorithm to make it suitable for sphere model, and then adopt the strategy in Section 3.4.4 to extract line segments and calculate vanishing points.

3.5.3.1 Using full-view images from different sensors

Real images with two camera sensors in different indoor environments are presented to evaluate the performance. First, we employ a sensor with a pair of fisheye cameras applied in [4] to generate full-view images. The intrinsic parameters are calibrated beforehand. Fig 3.14 shows experimental results of two examples. In order to see them easily, we project the full-view images in sphere model onto two opposite planes, much like fisheye images. In the left case, we observe that just a few segments belonging to ceiling-wall boundaries are detected. However, the left missing boundaries are completely recovered according to the ceiling-floor mapping H. For the next example, though the scene is relatively simple, there are a lot of segments on the walls corresponding to exhibition panels, which may confuse structure generation. We achieve

a good approximation of its structure. In particular, the right wall colored in blue in the right image is well reconstructed.

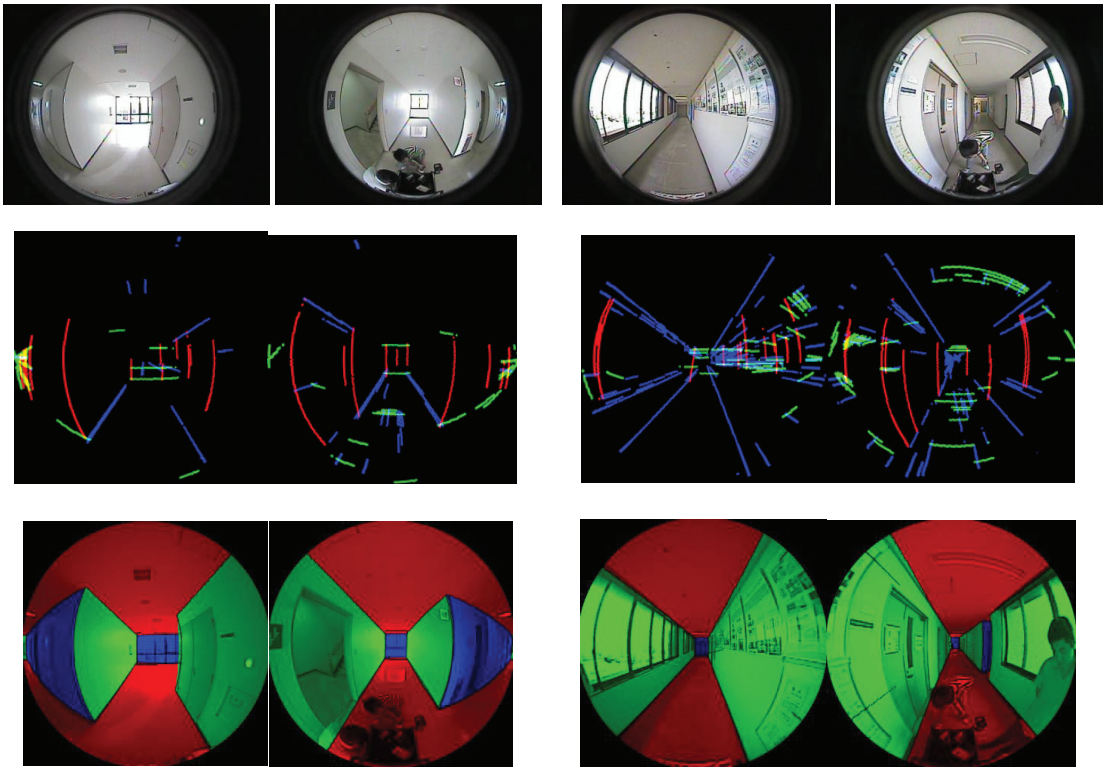


Fig 3.14 An example of experimental result from a fisheye camera. Top row: Two half scenes captured by fisheye camera. Middle row: Line segments. Last row: Estimated structure.

Another sensor used in experiments is called RICOH THETA[1], produced by RICOH Company. This one-click 360º camera provides us an easy way to capture full-view images. We mainly use it in the following experiments. Some results are given in Fig 3.15. For a quick and easy view, we display full-view images as panoramas based on sphere model, much like the captured images. In these cases, scenes become complicated. Though there are easily confused or occluded areas, which make it hard to define boundaries of structure exactly, we still obtain acceptable results. Take the case of a bath room in the second row as an example. Left part of the scene shows a washstand and a washing machine, which take up so much space that we cannot observe any floor-wall boundary on this area (see the original panorama and the extraction of line segments). Obviously it brings troubles to recover the occluded structure, especially for the conventional images with a small field of view of only this scene. However, we notice that the opposite side of room is comparatively simple corresponding to the right side of the panorama. The mapping relation can be obtained, which allows us to infer the occluded floor-wall boundaries from the corresponding ceiling-wall ones.

[1] https://theta360.com/en/.

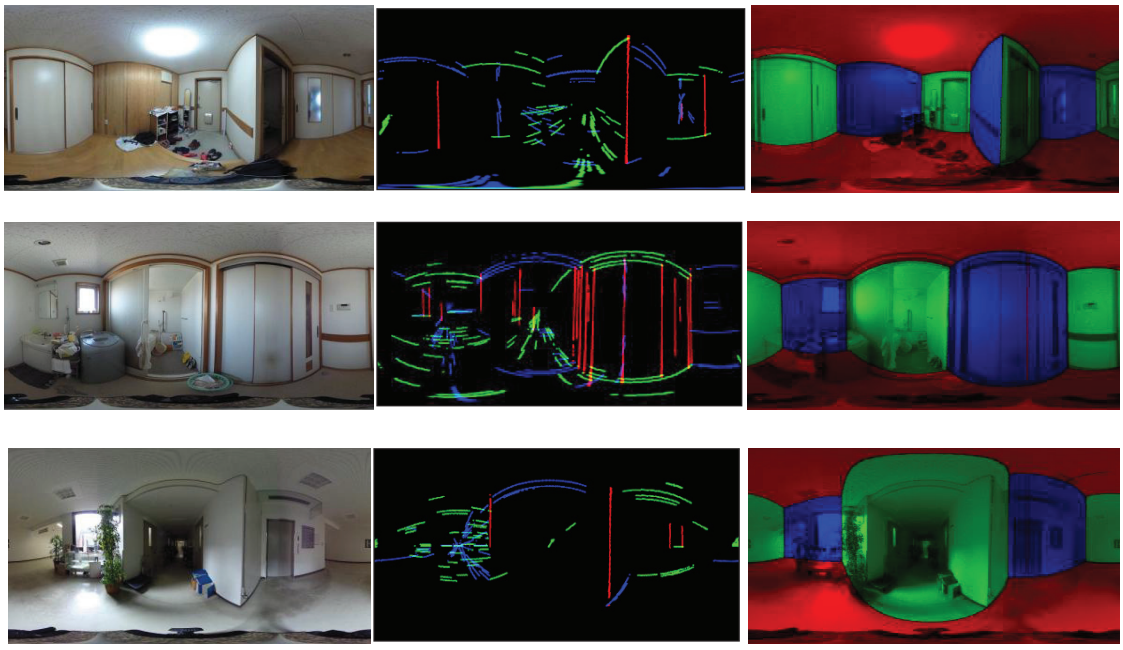


Fig 3.15 Some cases of experimental result from RICOH THETA. First column: The scene captured by RICOH THETA. Second column: Line segments. Third column: Estimated structure.

3.5.3.2 Close geometry vs. open geometry

Here, we compare the two different geometric constraints. A full-view image is separated into two images with each one covers a half of panoramic view. We regard them as independent ones obeying open geometry as the conventional ways do in [29, 30], and then execute our program.

To contrast easily, the original images and estimated structures based on close geometry are given in the first and second column of Fig 3.16. The results of two

separated views following open geometry are integrated as one panorama, as illustrated in the third column. In that column, we can see that some regions of rooms are recovered, while some failed, in particular near the edges areas of images. It occurs because the disconnected wall boundaries often cause poor estimation at the breaking points under the constraints in an open space. In the case of occluded bedroom, it is troublesome to obtain a ceiling-floor mapping in left separated image. Thus, the result turns out to be bad. In addition, the recovered layout is not connected to each other, since two images are used independently as two different scenes. More accurately, we also manually label the ground truth orientation for every pixel, ignoring the occluding objects. The percentage of pixels with the correct orientation for each image pair is reported in Table 3.1.

Table 3.1 Percentage of pixels with correct orientation.

Percentage	Image 1	Image2	Image3
Method on close geometry	0.93	0.86	0.95
Method on open geometry	0.48	0.40	0.37

As shown in the above experimental results, the proposed method based on close geometry outperforms the one based on open geometry significantly.

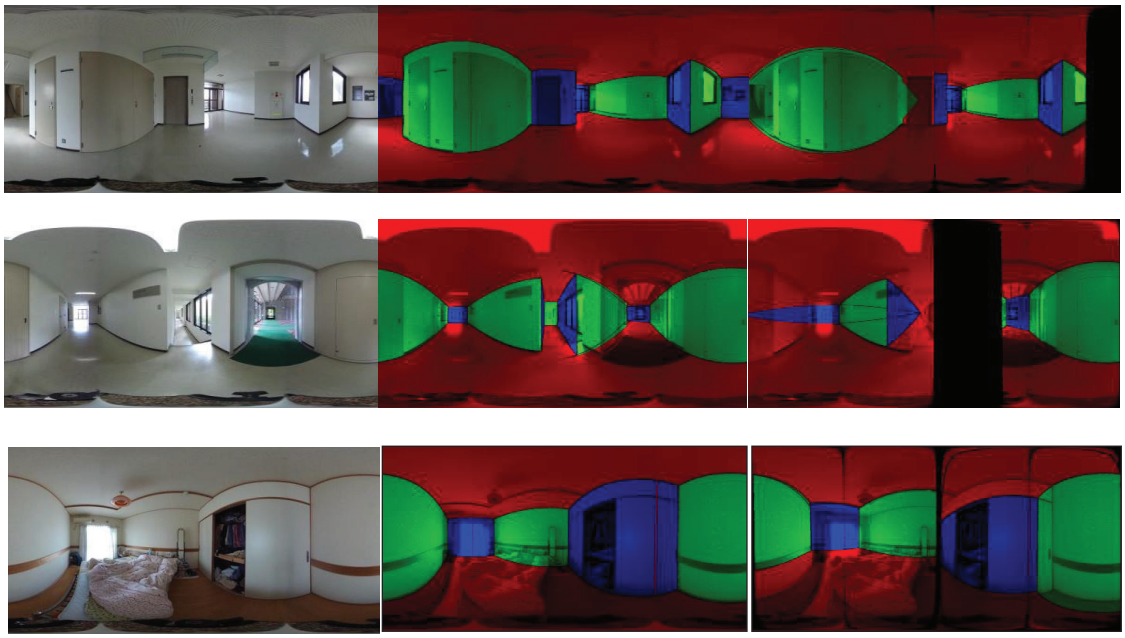


Fig 3.16 The comparison between close geometry and open geometry. First column: The original image. Second column: The result of estimated structures based on close geometry. Third column: The estimated structures of two separated views following open geometry are integrated as one panorama.

3.5.3.3 Proposed method vs. conventional method

To show the advantages of the proposed method more clearly, a comparative experiment is also carried out between the proposed method and one of the state-of-the-art methods of Lee et al. [29], which source code is available on the web. The input perspective images used in the method of [29] are generated from the full-view images, by projecting a full-view image towards the different directions on X-Z plane to generate projected images.

We compare their results to the labeled projected images from our estimated full-view image. Fig 3.17 shows some examples coming from the second and the third scene in Fig 3.16. A building structure of one image failed to be estimated. The percentage of the correct orientation for each image pair is given in Fig 3.18. We can see that our approach perform pretty well than the conventional method using an image with a narrow field of view.

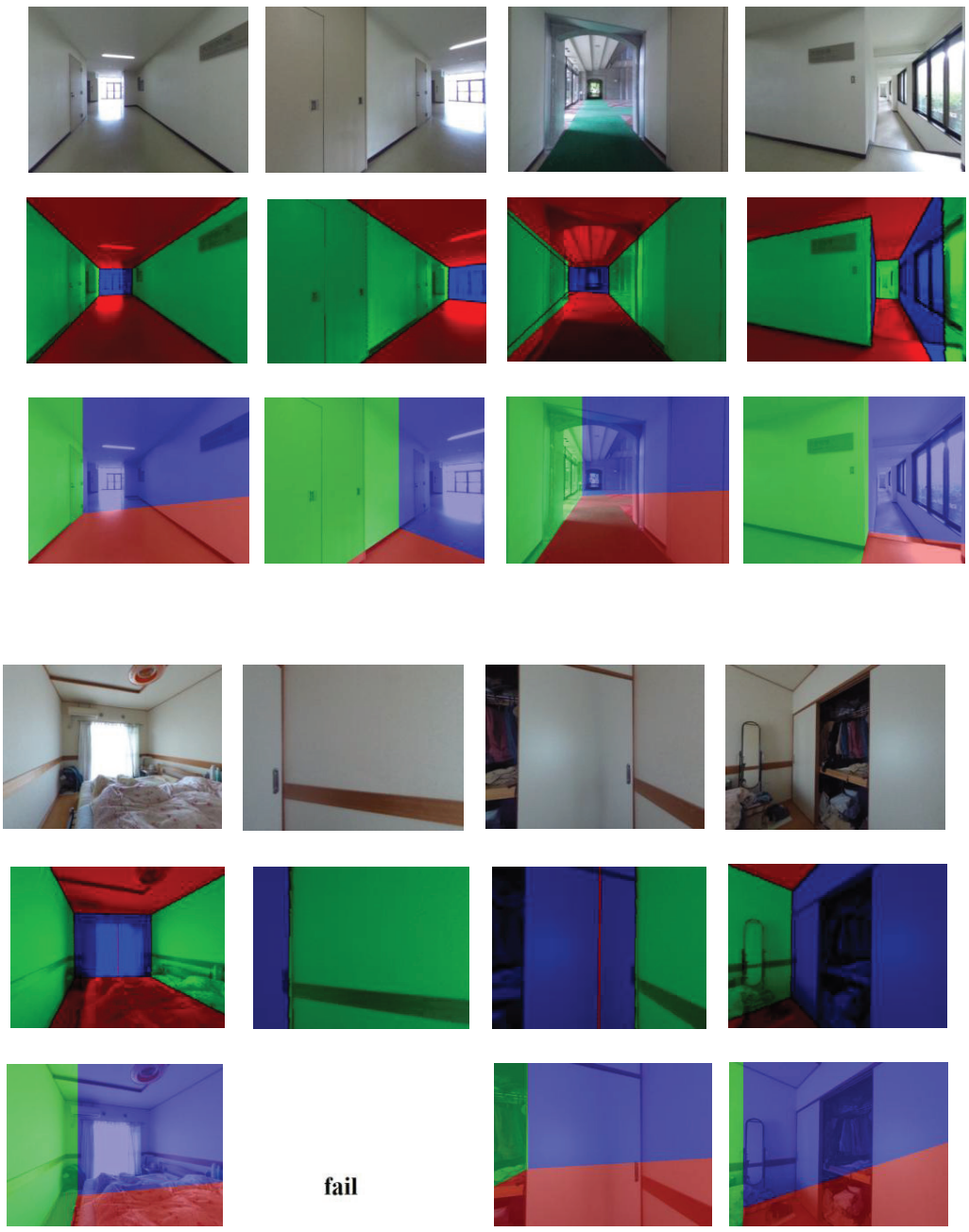


Fig 3.17 The comparison between the proposed method and the method of Lee et al. [29]. First row: The projected original image. Second row: The projected result by the proposed method. Third row: The result by conventional method using a projected view as an input.

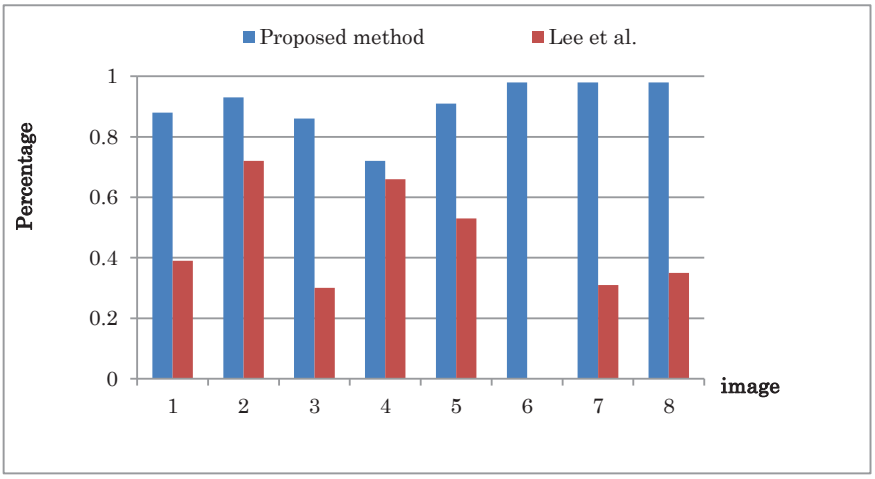


Fig 3.4 The percentage of the correct orientation for the results in Fig 3.17.

All of the experimental results shown above imply that the proposed close geometry is a powerful constraint and plays an important role in scene interpretation of full-view images. It provides more available cues, enabling computers to predict entire structure. We also believe that it is appropriate to consider an indoor model with a full-view image rather than several images with partial view.

3.6 Conclusion

In this part, we first investigate the problem of estimating structure of rooms from a single fisheye image. A symmetrical rule which describes geometric constraints in fisheye images is introduced. Then we estimate the spatial layout of rooms starting from a collection of line segments. A novel method is given to refine a preliminary structure to obtain the final main structure. The experiments demonstrate that our approach based on geometric reasoning can be used to estimate the spatial layout of indoor scene.

However, we notice that the field of view is enlarged to a hemisphere, the recovered structure still involves incomplete scene, referring as *open geometry*. In order to obtain the entire structure of rooms, we pay attention to full-view images and impose a totally different description to present geometric constraints, called *close geometry*. A novel method is given to explore indoor scene understanding by searching for the structure which fits the extracted line segments best. We combine the process of deciding boundaries and pursuing the maximal score together to make our algorithm much efficient and robust. As shown in the experimental results including the comparative experiments, the proposed method of interpreting full-view images based on close geometry outperforms the conventional methods which use images with limited field of view based on open geometry.

As future work, we plan to test more images of various cluttered rooms and improve the success rate by applying a more reliable algorithm of extracting line segments. In addition, we believe our research is very useful in many fields. Another

work could be devoted to the application of indoor scene estimation from a full-view image, such as the detection of obstacles in a room or analyzing human activities within a surveillance scene.

CHAPTER 4

Fast Generation of Perspective Display from Full-View Image

4.1 Introduction

When we do scene analysis and scene understanding using omnidirectional cameras, we find that perspective display needs to be generated frequently. For the case of the comparative experiment in the former part, we also project a full-view image towards the different directions on X-Z plane to generate projected images, for the purpose of comparing with the state-of-the-art method based on perspective images.

In general, an omnidirectional image is based on the successive spherical model (Fig 4.1). However, it has high cost of mapping a spherical point to the omnidirectional image pixel because of mass non-linear calculation. It takes a lot of time to generate perspective display. That is why so far such operation is usually done by hardware in practice for efficiency. Is there any approach that can accelerate this processing to make a perspective display easily and rapidly?

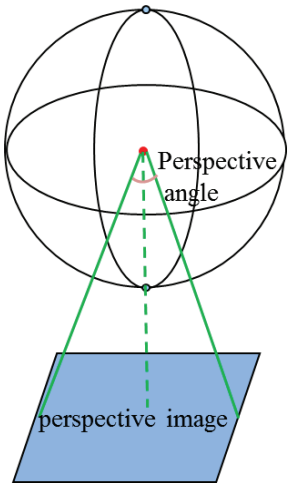


Fig 4.1 Perspective display based on the successive spherical model.

In the research of computer graphics, an omnidirectional image can be regarded as bubbles, which mean a 360-degree panorama without considering the specific visual sensors. Thus, the problem of getting perspective display from omnidirectional cameras can be redefined as getting that from bubbles. Considering the isotropy of bubbles, a natural representation of bubbles is spherical map, or spherical image. Moreover, the cells, i.e., pixels, of spherical bubbles should be as uniform as possible so that the isotropy of the sphere around any cell point is preserved as well as possible.

Here, we represent bubbles, by SCVT (Spherical Centroidal Voronoi Tessellation) images which are called spherical bubbles in our research and propose a method to generate perspective display swiftly more than before.

The rest of this part is organized as follows. The related research is introduced in the next section. The method of fast generation of perspective displays is described in Section 4.3. The experimental result is presented in Section 4.4. Finally, we draw a conclusion.

4.2 Related Research

4.2.1 Research about Bubbles

Bubbles are widely used in many fields. A famous one refers to the street-level images in Google Street View. Adding street-level images to traditional line-segment maps enhances the reality of on-line maps dramatically. Users are able to visit cities on the internet by navigating between bubbles.

Bubbles are also regarded as environment maps in computer graphics. There are some forms of bubbles with different features and different applications. A common one is called a cubic environment map [34], which is stitched from the images captured by multiple cameras or lenses with overlapping field of view [4]. A cubic environment map consists of six perspective images which correspond to the six planes of a cube with the view point at the center of the cube [35], as shown in Fig 4.2.

Let the distance of the center of the cube from the square plane be 1. Then, the distance from the center to the corner point becomes $\sqrt{3}$. The sampling rates for the directions, which is defined as the ratio of the solid angle between the maximum (the

central pixel of the square plane) and the minimum (the pixel at the corner of the square plane), differ from a factor of $3\sqrt{3}$.

Besides the cubic map, spherical environment map (which is different from the SCVT map) [36], paraboloid map [37] and latitude-longitude map [38] have been proposed.

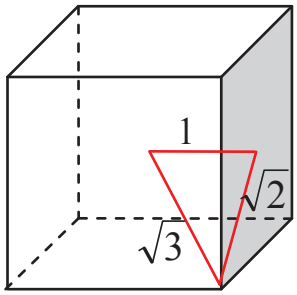


Fig 4.2 Sampling rate of cubic environment map for the directions.

The spherical environment map [36] is based on the simple analogy of a small, perfectly mirroring ball; the image that an orthographic camera sees when looking at this ball is the environment map. However, for the spherical environment map, the sampling rate of this map is maximal for directions opposing the viewing direction, and goes towards zero for directions close to the viewing direction. Moreover, there is a singularity in the viewing direction because all points where the viewing vector is tangential to the sphere show the same point of the environment.

The paraboloid map consists of two paraboloids [37]. Although the paraboloid map can be reused for any given viewing direction, that is, it is view-independent, the sampling rate for the directions is still as great as 4. The latitude-longitude map, which is generated by dividing a sphere along the latitude and longitude [38], is proposed. Since the latitude-longitude map is severely over-sampled around the poles, the sampling rates for the directions differ greatly.

4.2.2 **Research about SCVT**

A SCVT image is known as its quasi-uniform property [39], which can be obtained by subdividing the icosahedrons iteratively, as shown in Fig 4.3. Table 4.1 shows the corresponding sampling rate (the ratio of largest cell to smallest cell) of the SCVT maps for the directions. Although the sampling rate for the directions varies with the subdivision levels, it has a limit value, about 1.36. Therefore, spherical bubbles are a better representation for the isotropy in respect of direction, compared with the conventional environmental maps.

Table 4.1 Properties of the SCVT map.

Subdivision level	Number of cells	Sampling rate for directions
1	42	1.13
2	162	1.29
3	642	1.31
4	2562	1.35
5	10242	1.36
6	40962	1.36

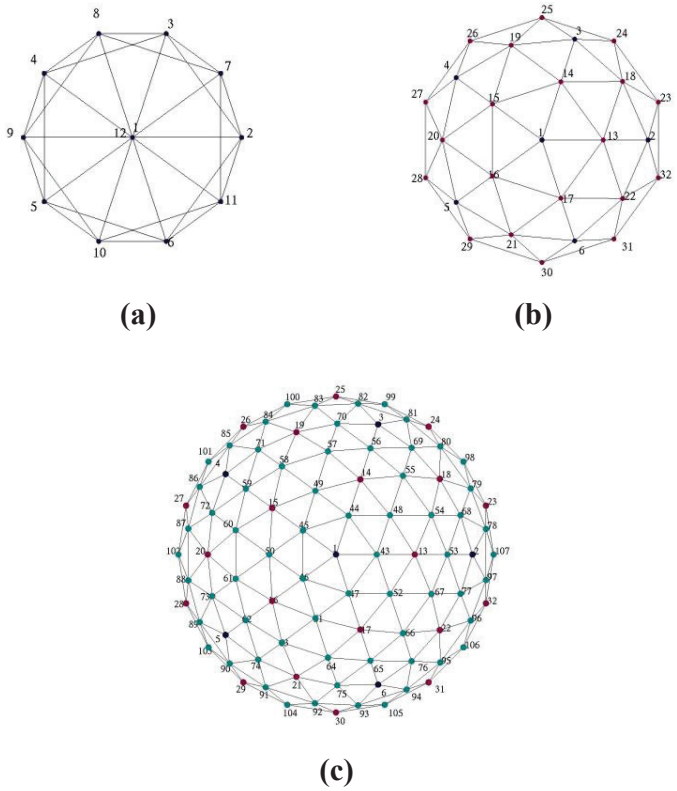


Fig 4.3 Process of the geodesic division of an icosahedron. (a) The initial icosahedrons. (b) 1-level subdivision. (c) 2-level subdivision.

Perspective displays are generated from spherical bubbles according to users' view direction and zoom-in/out operation. SCVT image can be represented as 2D array $S(i_s, j_s)$, in computer [39], [40], as shown in Fig 4.4(a). To generate perspective displays, the pixel of perspective displays $P(x_p, y_p, f_p)$, must be determined from that of the SCVT images. While the mapping from pixel (i_s, j_s), of SCVT images to spherical coordinate,

(θ,ϕ), can be carried out quickly by using a look up table, $T((i_s,j_s),(\theta_i,\phi_j))$, the inverse mapping is troublesome. Fig 4.4(b) shows the angle change of the pixels in a row of the SCVT image array of Fig 4.4(a), where the curves correspond to the rows of SCVT image array. It indicates that the mapping relation between spherical angle (θ,ϕ) and array number (i_s,j_s) is nonlinear. Therefore, search is necessary in the inverse map as follows.

$$\begin{bmatrix} x_p \\ y_p \\ f_p \end{bmatrix} \longrightarrow \begin{bmatrix} \theta \\ \phi \end{bmatrix} \xrightarrow{Search} \begin{bmatrix} i_s(\theta_i) \\ j_s(\phi_j) \end{bmatrix} \tag{4.1}$$

In the related research [39] and [40], given a spherical polar coordinate (θ,ϕ), an initial position is first estimated in 2D array $S(i_s,j_s)$; then, the corresponding cell point is found by iteratively searching for a local maximum among the estimated cell and its neighboring cells. That is to say, finding the corresponding pixel of perspective displays from spherical bubbles according to the spherical coordinates involves two search processes. One is to find the approximate location according to the average intervals of azimuth angle and polar angle between neighboring pixels, called *average search*, and the other is to find the nearest pixels according to the neighboring relations among pixels, called *neighboring search*.

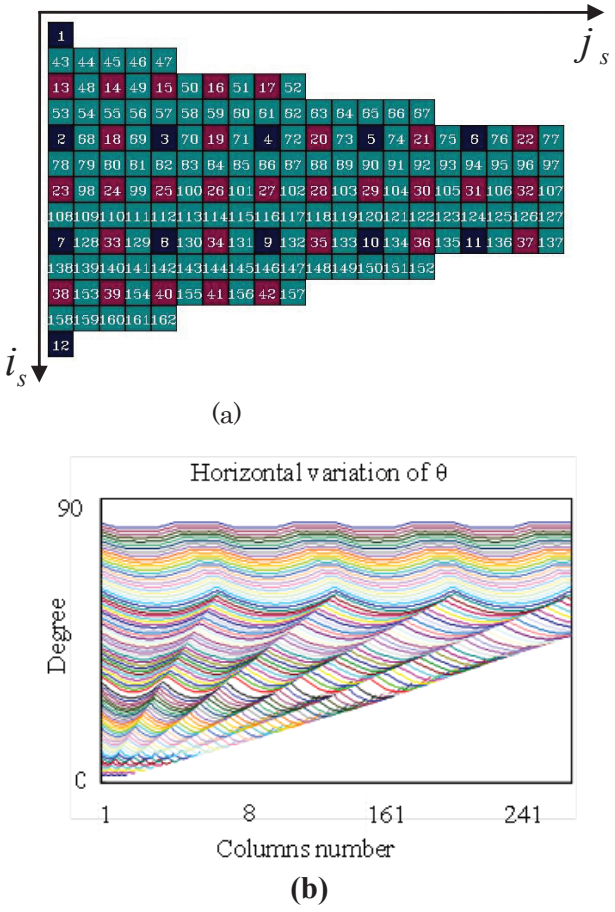


Fig 4.4 2D array of SCVT image: (a) 2D array of cell points for 2-level subdivision of Fig 4.3(c). (b) Angle change of the pixels in a row of the SCVT image array.

Since users want an instant response when cameras move with a robot, the generation of perspective displays should be carried out as speedily as possible. We accelerate this processing by employing the adjacent cues between neighboring cells and

the pyramid data structure of spherical bubble. This will be much helpful that users can get different sequent views from the camera in a moving robot by means of transformation between bubbles.

The SCVT map has the distinguished advantages over the conventional bubbles. Although SCVT maps have been proposed in related research [39] and [40], these studies focus on the algorithm of finding neighboring pixels. In comparison with them, we use SCVT map to represent bubbles. This research has the following characteristics.

- Generate perspective view from spherical bubble by employing the neighboring relation among pixels. As mentioned above, finding the corresponding pixel of perspective display from spherical bubble according to the spherical coordinates involves two search processes: average search and neighboring search. If the resolution of the perspective display and that of the spherical bubble are approximately the same, the neighbors of a pixel in the perspective display should correspond to the neighbors of the corresponding pixel in the spherical bubble. Thus, in this case, to generate the perspective display, we can omit the average search, and only carry out neighboring search except for the first one.
- Use the pyramidal data structure of spherical bubble to cope with the change of resolution of perspective display. To generate a spherical bubble with approximately the same resolution as the perspective display, the pyramidal data structure of SCVT image from the original spherical bubble

is used. To generate perspective display, its resolution is first computed. Then, the corresponding layer of the SCVT image is selected from the pyramidal data structure.

Using the above techniques, perspective display can be generated from spherical bubble with lower computation cost.

4.3 Generating Perspective Display from SCVT Map

4.3.1 Generation of Perspective Display by Using Neighboring Relation

Assume that the resolution of the generated perspective display is approximately the same as that of the spherical bubble. Thus, if two pixels are adjacent to each other in perspective displays, they should also be adjacent in SCVT maps. That is, neighboring relation between pixels is preserved for both perspective display and spherical bubble. In this case, the perspective display can be generated simply from spherical bubbles as follows.

1) For the first pixel at the top-left corner of the perspective display, compute its corresponding nearest pixels in the spherical bubble by average search and neighboring search.

2) For the next pixel to be generated, compute its corresponding nearest pixels in spherical bubble by starting from the known neighboring location merely with neighboring search.

Thus, the average search is necessarily carried out only for the first pixel.

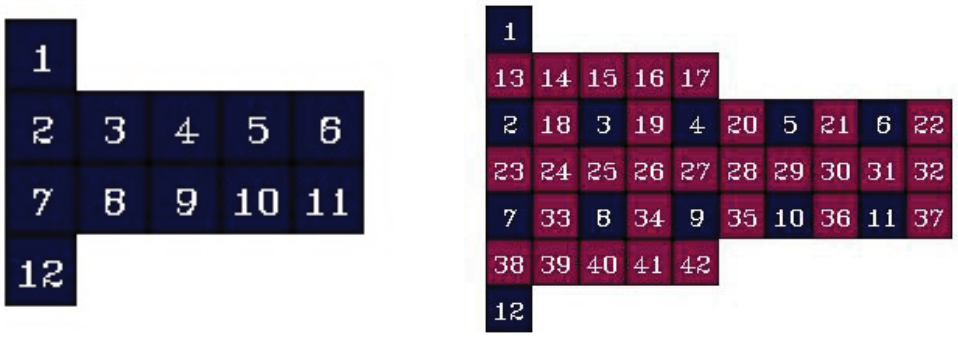
4.3.2 **Generation of Perspective Display by Using Pyramidal Data Structure of Spherical Bubble**

In practice, the resolution of perspective display is changeable with users' zoom-in/out operation. This means that the resolution of perspective display may be different from the original spherical bubble obtained from the captured images. Here, the pyramidal data structure of spherical bubble is used to cope with this problem so that the above proposed method of generating the perspective display using neighboring relation can be applied. In addition, the quality of perspective image is also promoted during the process of up-sampling. The detail information is given below.

The pyramidal data structure of spherical bubble corresponds to the subdivision of the initial icosahedron. The arrays of cell points for 0-level and 1-level subdivisions corresponding to Fig 4.3(a) and (b) are shown in Fig 4.5(a) and (b), respectively. The array of cell points for 2-level subdivision corresponding to Fig 4.3(c) is shown in Fig 4.4(a). The pyramidal data structure of spherical bubble is generated by the down-sampling or up-sampling of the original spherical bubble.

Suppose that the original SCVT image corresponds to the L th-level subdivision array $S_L(i_L, j_L)$, where i_L and j_L correspond to the row number and column number of SCVT array, respectively. Let the down-sampling SCVT image corresponding to the $(L-1)$ th-level SCVT array be $S_{L-1}(i_{L-1}, j_{L-1})$. We have

$$i_L = 2\ i_{L-1}, j_L = 2\ j_{L-1} \tag{4.2}$$



(a) **(b)**

Fig 4.5 The arrays of cell points for 0-level and 1-level subdivisions corresponding Fig 4.3(a) and (b).

To avoid the aliasing problem, the down-sampling is carried out by averaging the neighboring cells. Here, the neighboring cell search is carried out by the algorithm [40].

Fig 4.6 shows the sketch of the down-sampling with averaging the neighboring cells. The hexagonal cells with red lines indicate those of $S_{L-1}(i_{L-1}, j_{L-1})$ while the cells with black line indicate those of $S_L(i_L, j_L)$. Each cell of $S_{L-1}(i_{L-1}, j_{L-1})$ contains the entire corresponding cell of $S_L(2i_{L-1}, 2j_{L-1})$ computed in terms of (4.2) and about half of the neighboring cells of $S_L(2i_{L-1}, 2j_{L-1})$. Thus, the pixel value of $S_{L-1}(i_{L-1}, j_{L-1})$ can be computed as follows.

$$V^{L-1} = \frac{\frac{1}{2}\sum_{j=1}^{N} V_j^L + V^L}{\frac{N}{2}+1} \tag{4.3}$$

Where V^{L-1} and V^L are the pixel value of $S_{L-1}(i_{L-1}, j_{L-1})$ and $S_L(2i_{L-1}, 2j_{L-1})$, respectively. V_j^L indicates the pixel value of the neighboring cells of $S_L(2i_{L-1}, 2j_{L-1})$. N is the number of the neighboring cells. $N = 6$, except for the twelve cell points of the icosahedron, where $N = 5$.

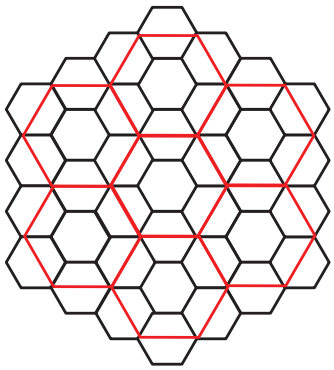


Fig 4.6 The sketch of down-sampling of a SCVT image.

To get high resolution, up-sampling is necessary. Considering the relation between the L th-level and the $(L-1)$ th-level SCVT array shown in (4.2), we use four neighboring cells of the pixel value of $S_{L-1}(i_{L-1}, j_{L-1})$ to do up-sampling. Considering the

case in Fig 4.7, let the pixel value of the $(L-1)$ th-level vertex A_j^{L-1} be V_j^{L-1} . The new pixel value V^L of the L th-level vertex A^L, is determined as follows.

$$V^L = \frac{\sqrt{3}V_1^{L-1} + \sqrt{3}V_2^{L-1} + V_3^{L-1} + V_4^{L-1}}{2(1+\sqrt{3})} \tag{4.4}$$

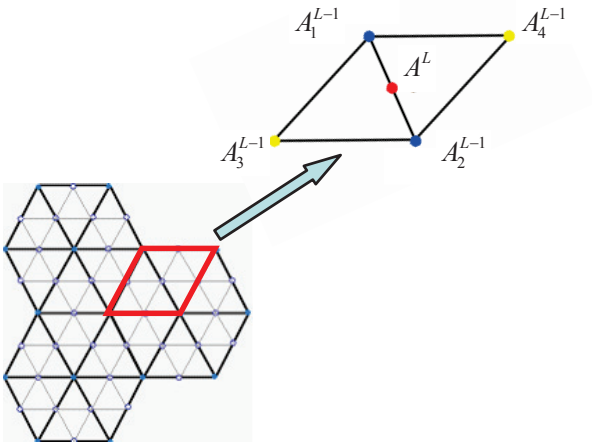


Fig 4.7 The sketch of up-sampling of the SCVT image.

Here, we use four neighboring cells to generate one cell in the next subdivision allocating different adaptable weights. It implies that the interpolation is implemented in a certain way. The size of SCVT image is enlarged with more pixels, because more points in the sphere are added. As a result, perspective display can be generated from a denser spherical bubble, which makes the perspective image smooth with better quality.

However, we still need a way to carry out basic interpolation for either up-sampling or down-sampling. We employ the method called tri-linear interpolation proposed in [41] if necessary.

Note that (4.3) can be regarded as a filter only considering itself and the nearest neighboring cells, neglecting the ones far away. In some way, the function is much like a low pass Gaussian filter. Though it can smooth images to some extent, it may cause some loss of quality for generating a perspective image. If the quality is required, we can just use the pyramidal data structure of spherical bubble with the original subdivision of icosahedron and up-sampling. In that case, not only the computational cost of generating a perspective image is cut down but also the quality is improved. Moreover, it is still much faster than the conventional method. In the experimental section, we discuss more about this.

4.3.3 **The Process of Generation of Perspective Display**

Step1. Compute the resolution of the perspective display to be generated. The resolution of the perspective display is measured as the pixels per unit solid angle by mapping the perspective display to a unit sphere. Then compute the resolution of the same definition for every SCVT image in the pyramidal structure.

Step2. Select the level whose resolution is closest to the perspective display, from the pyramidal data structure of spherical bubble.

Step3. Generate the perspective display by using neighboring relation between pixels, as mentioned in Section 4.2.2.

4.4 Experiment

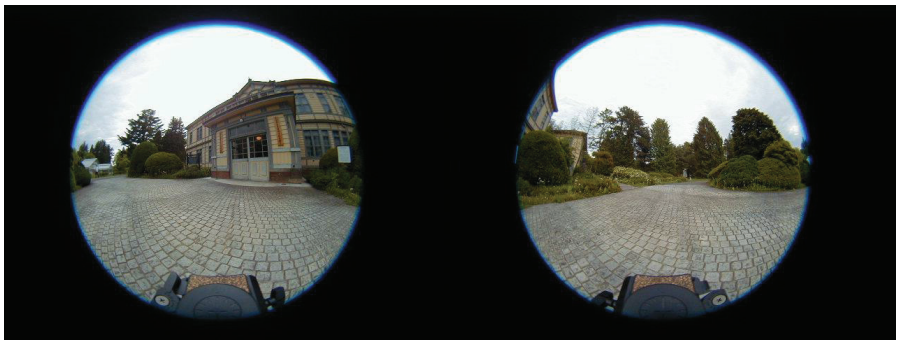
While the mapping from the pixel of perspective display to the closest pixels of spherical bubble is carried out by both the average search and neighboring search in the conventional method in [39], [40], the proposed method employs the neighboring relation combined with the pyramidal data structure of spherical bubble so that the mapping can be achieved merely by neighboring search. In this section, we present the experimental results to show the performance of the proposed method in comparison with the conventional method. In our experiments, all the perspective images generated by both two methods are using closest pixel without any interpolation, unless it is expressly stated.

4.4.1 Performance of Computational Speed

We use the spherical image in Fig 4.8(a) to test our method for computational processing speed. The spherical image is represented by a compact rectangular 2D array, SCVT as described in the research [40], which is generated from a pair of fisheye image as shown in Fig 4.8 (b). The detailed information on the format can be found in the reference [40].

The original SCVT image is 640x256 pixels, corresponding to 7th-level subdivision of an icosahedron. The pyramidal data structure of the SCVT image generated by the proposed down-sampling and up-sampling algorithm is shown in Fig 4.9.

(a)



(b)

Fig 4.8 The spherical image used in the experiment: (a) The SCVT image. (b) The raw full-view image captured by a fisheye camera.

At first the average of the solid angle per pixel, which stands for the resolution of image, is computed. Then, the corresponding level of the pyramidal data structure of the SCVT image is selected according to the computed average of the solid angle per pixel. Finally, the perspective display is generated from the selected level of the pyramidal data structure of the SCVT image.

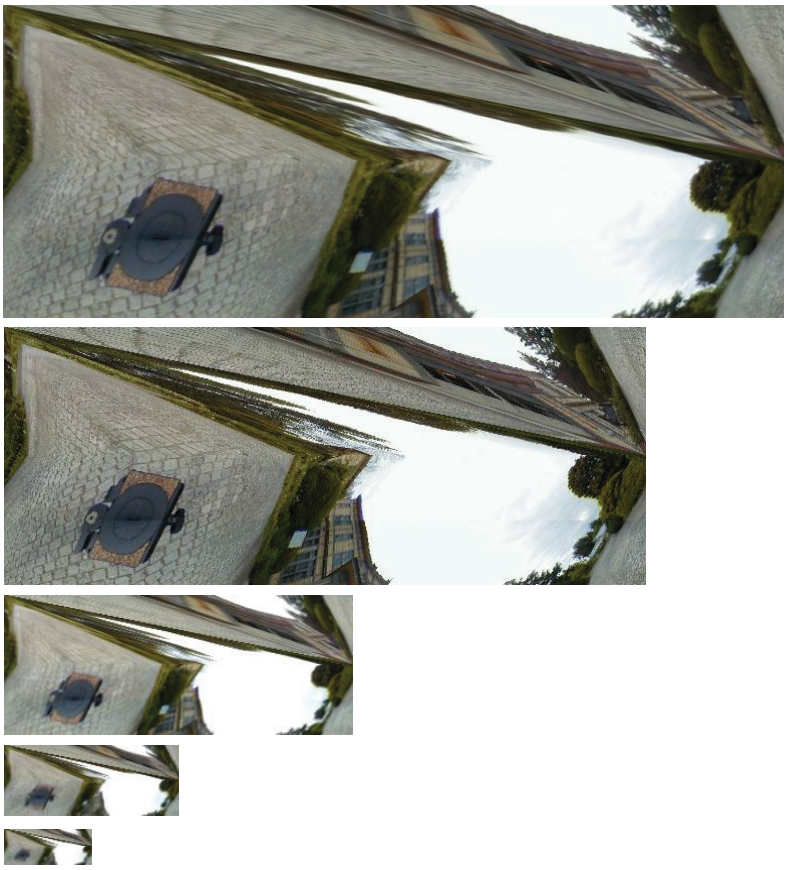


Fig 4.9 The pyramidal data structure of the SCVT image generated referring to 8^{th}, 7^{th}, 6^{th}, 5^{th}, 4^{th}level of subdivision of spherical bubble.

Since the resolution of perspective display is determined by the view size and the view field (see Fig 4.10), we can carry out the experiments with the both conditions varying, respectively.

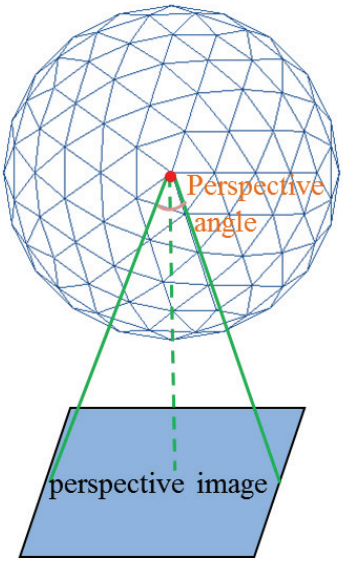


Fig 4.10 Perspective display based on the discrete spherical model.

First, the image size of the perspective display to be generated is fixed as 100×100 pixels. The field of view of the perspective display (perspective angle) is changed. Table 4.2 shows the computational time of some cases, which correspond to different levels of subdivision of spherical bubble by the proposed method. Though only 7th-level subdivision of the original SCVT data structure is used in the conventional method, the speed varies heavily because average search is carried on for every pixel, of which the cost of computation is sensitive to the perspective angle. However, the proposed method performs well. The computational time is approximately shortened to

the half. What is more, it holds a good property of computational stability, since the cost of doing neighboring search is almost the same for each cell. We also list the processing time of generating perspective image mapping from fisheye image directly in the reference [4] based on the widely used successive spherical model. It needs to solve the non-linear equations, and executes much slowly.

Table 4.2 Comparison of computational time under the condition of the fixed image size.

perspective angle	level of pyramidal data structure	conventional method	proposed method	method based on successive spherical model
40°	8^{th} Level	63 ms	32 ms	187ms
90°	7^{th} Level	78 ms	31 ms	188ms
145°	6^{th} Level	62 ms	31 ms	203ms

Then, we do another experiment by changing the image size with the perspective angle fixed to 40°. The computational time of some cases are shown in Table 4.3. As the image size is enlarged, computational cost of either method increases. However, our method is superior to other two algorithms.

Table 4.3 Comparison of computational time under the condition of the fixed perspective angle.

image size	level of pyramidal data structure	conventional method	proposed method	method based on successive spherical model
300×300	8^{th} Level	608ms	312 ms	2388ms
200×200	7^{th} Level	255 ms	125 ms	748ms
100×100	6^{th} Level	62 ms	31 ms	203ms

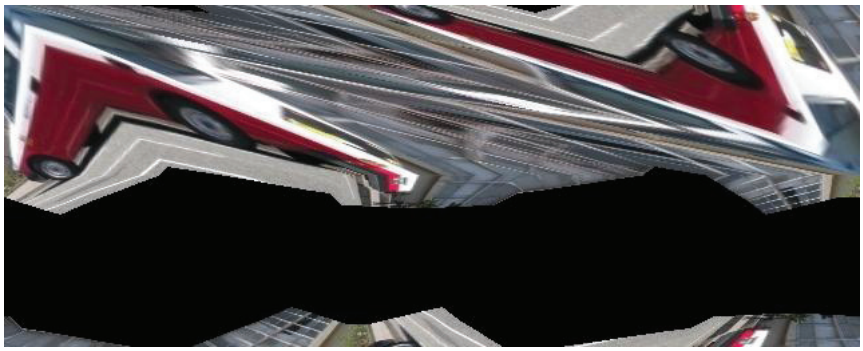
Obviously, the results shows that the algorithms based on discrete spherical model, SCVT perform greatly better than that based on successive spherical model. Compared with the conventional method, the computational time of the proposed method is much shorter, less than the half at best. Therefore, it can generate perspective images fast coping with changing resolution. In other words, it means we can get perspective display instantly from spherical bubble according to users' view direction and zoom-in/out operation.

4.4.2 Performance of Image Quality

In order to test our algorithm in respect of image quality, an ordinary image in Fig 4.11 is employed. We map it back to discrete spherical model to obtain a SCVT image, which is indicated in Fig 4.12(a). Fig 4.12(b) gives us an intuitionistic spherical view of the SCVT format by CG.

Fig 4.11 The original image used in the experiment in section 4.4.2.

Let Fig 4.12(a) correspond to 7th-level subdivision of an icosahedron. Then, we generate the pyramidal data structure by the proposed method, and resize the original image with interpolation to make the resolution correspond to different level of the pyramidal data structure, with perspective angles fixed. Regarding the resized images as the reference ones, we can evaluate the qualities of perspective displays generated by the conventional method and our new method. Note that it may bring in some absolute deviation between the reference images and the SCVT images because of resized operation and the process of discrete division. However, it is still meaningful to compare two methods based on the same standard.



(a)

(b)

Fig 4.12 The SCVT image generated from Fig 4.11: (a) The SCVT format. (b) The corresponding intuitionistic spherical view by CG.

The well-known MSE (Mean Square Error) and SNR (Signal to Noise Ratio) are used as evaluation criterion. SNR is expressed in the way of logarithmic decibel scale, which is defined in the reference [42] as follow:

$$SNR = 10\log_{10}\frac{S}{N} \tag{4.5}$$

Here, S is the square sum of all pixels' value in the perspective image generated, calculated by the formula:

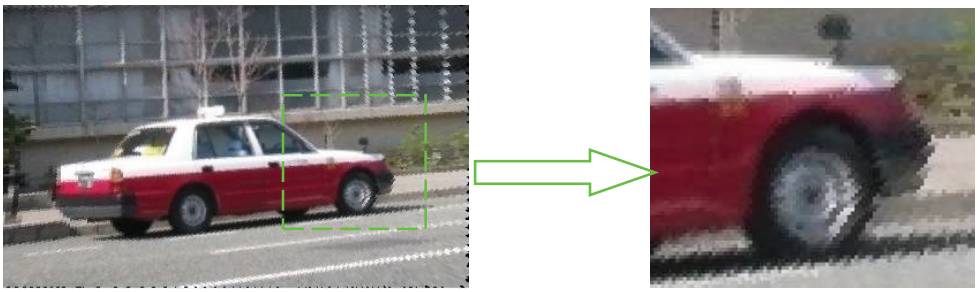
$$S = \sum_{i=0}^{w-1}\sum_{j=0}^{h-1}[r(i,j)]^2 \tag{4.6}$$

Where $r(i,j)$ is the value of pixel (i,j). And N is the square err of all the corresponding pixels' value between the generated image and the reference one.

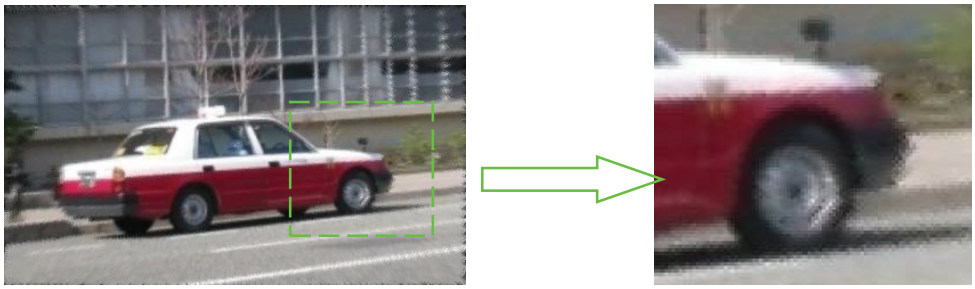
$$N = \sum_{i=0}^{w-1}\sum_{j=0}^{h-1}[r(i,j)-t(i,j)]^2 \tag{4.7}$$

Where $t(i,j)$ refers to the value of pixel (i,j) in the reference one. We also give the SSIM results, which are able to reflect more subjective judgment of humans.

First, we calculate the resolution of the resized reference image, and test 8^{th} level of subdivision of spherical bubble for our proposed method, referring to up-sampling. Fig 4.13 shows the results of perspective displays generated by the two algorithms. Obviously, the quality of (b) has better performance than (a). Table 4.4 lists the MSE, SNR and SSIM in comparison with the resized reference one. R, G, B in the second line indicates the RGB channels of image, respectively. From the table, we can see the quality of perspective display is improved apparently because of up-sampling operation.



(a)



(b)

Fig 4.13 The perspective displays corresponds to up-sampling. (a) Conventional algorithm. (b) Proposed algorithm with 8th level of subdivision.

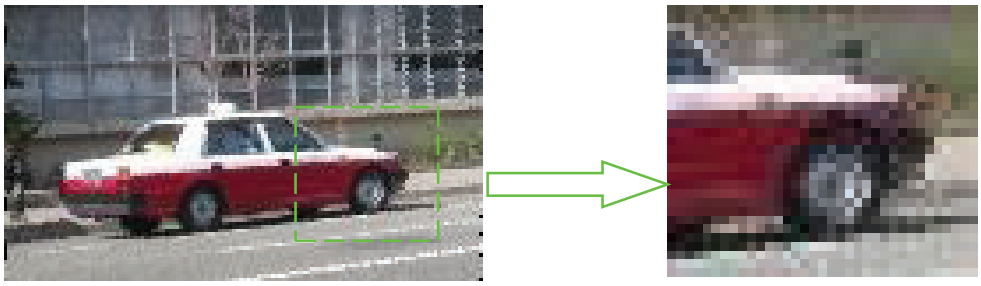
Then, we turn to the case of 6th level of subdivision of the spherical bubble, referring to down-sampling. Though the computational time by our method is 33ms, greatly reduced from 78ms by the conventional method, the perspective image gets deteriorated as shown in Fig 4.14. In such a case, if the quality is required, we can just use the original subdivision of icosahedron 7-th level instead of 6-th level. Then the image with the same quality as Fig 4.14(a) can be obtained. The processing time is

sacrificed comparing to 6-th level, however, it is still much faster than the conventional method, shortened from 78ms to 38ms. A good balance between the processing speed and the image quality can be made by adopting the pyramidal data structure skillfully according to our requirement.

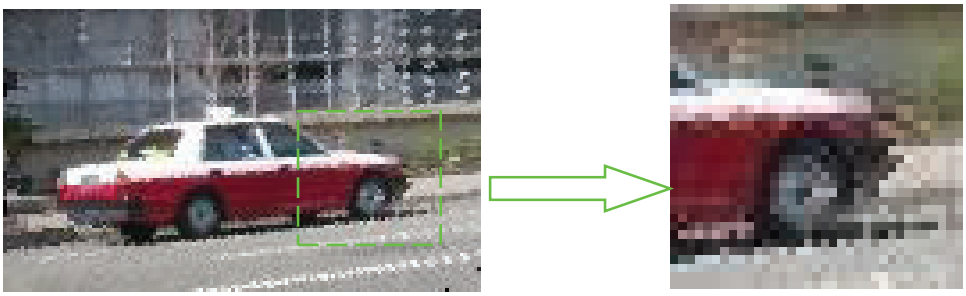
Table 4.4 Comparison of image quality with reference image.

Method	MSE			SNR(dB)			SSIM		
	B	G	R	B	G	R	B	G	R
Conventional Method	180.73	180.17	185.76	19.63	19.59	19.72	83.3 %	83.8 %	83.5 %
Proposed Method	120.90	121.54	129.01	21.38	21.30	21.30	86.1 %	86.5 %	86.2 %

The results of experiments above show that our method performs well. The computational cost of generating perspective image is cut down and the quality is improved by up-sampling operation. For the pyramidal data structure of SCVT image, even though it is possible to do more discrete division in theory, we usually take it as far as 9-th level division in practice. The size of 9-th level SCVT image is 2560×1024, and it has enough resolution for the full view sensor often used.



(a)



(b)

Fig 4.14 The perspective displays corresponds to down-sampling. (a) Conventional algorithm. (b) Proposed algorithm with 6th level of subdivision.

4.5 Conclusion

Spherical bubble, represented as a SCVT image, is quasi-uniform, and has distinctive advantage in the sampling rate for the direction over other maps. Omnidirectional images can be regarded as spherical bubbles for processing. When omnidirectional cameras are used in robotic systems, it is necessary to frequently generate perspective display with changeable resolution from spherical bubble according to the users' view direction and zoom-in/out operations.

In this part, the adjacent cues among neighboring pixels combined with the pyramidal data structure of spherical bubble are employed to cut down the computational cost of generating perspective display. In addition, image quality is also improved when up-sampling is carried on. The experimental results are presented to show the effectiveness of the proposed methods. In the future, we will incorporate more outstanding interpolation schemes instead of linear interpolations during the process of generating pyramidal structure. Besides that, we also intend to use spherical bubble for real-time tasks in a mobile robot.

CHAPTER 5

Conclusion

Scene analysis and scene understanding are two ultimate goals in computer vision and remain tremendous challenges for vision researchers. Scene analysis connects vision sensors and human's eyes. We hope visual display can offer friendly and useful information for users. On the other hand, Scene understanding also link computer to our brains, we would like computers to be able to think like humans with artificial intelligence.

There is no doubt that the interest for omnidirectional vision in the robotics community is getting more and more pragmatic. Omnidirectional vision has an incomparable advantage of holding a large field of view. It is appropriate to carry out a complete representation of environment which can be used to answer virtually any question about our world, ranging from map building and motion estimation to object categorization.

This thesis deliberately presents some of the issues in scene analysis and scene understanding using omnidirectional sensors, as follows:

1) A pair of fisheye cameras is used to develop a horse vision system, by imitating a horse's eyes for scene analysis.

2) A novel method of interpreting indoor scenes from a single omnidirectional image, either a fisheye image or a full-view image is proposed.

3) As a basic processing operation of omnidirectional vison, perspective display plays a critical role for the application in scene analysis. A method of fast generation of perspective display from omnidirectional images is also given.

This dissertation covers some of application for omnidirectional vision towards two major challenging tasks in computer vison. We construct a vision system for environment analysis, represent a framework for structure estimation, and develop a novel algorithm to obtain perspective display. The methods presented in this thesis should not be thought of as final algorithms. We believe they can be extended and adapted for other different tasks.

Our work is just one small step towards the problems in omnidirectional vision. So many things remain to be done, and broader utilization is expected in the future. It may combine visual information with artificial intelligence, which links the virtual world to the real world, by improving interaction with humans.

REFERENCES

[1] B. Timney, T. Macuda, "Vision and hearing in horses" , *Journal of the American Veterinary Medical Association*, Vol. 218, No. 10, pp. 1567-1574, May. 2001.

[2] G. Waring, "Horse Behavior", Norwich, NY: William Andrew Press, pp.18-25, 2007.

[3] J. C. Bazin, et al, "Automatic scene structure and camera motion using a catadioptic system,"*Computer Vision and Image Understanding*, Vol.109, pp.186-203, Feb. 2008.

[4] S. Li, "Monitoring around a vehicle by a spherical image sensor", *IEEE Trans. on ITS*, Vol.7, No.4, pp.541-550, 2006.

[5] C. Chen, et al, "Cooperative mapping of multiple PTZ cameras in automated surveillance systems," In Proc. CVPR, pp. 1078-1084, 2009.

[6] J. Heller, T. Pajdla, "Stereoraphic rectification of omnidirectional stereo pairs," In Proc. CVPR, pp. 1414-1421, 2009.

[7] S. Li, "Binocular spherical stereo," *IEEE Trans. Intell. Transp. Syst.*, Vol. 9, No. 4, pp. 589-600, Dec. 2008.

[8] S. Li, K. Fukumori, "Spherical stereo for the construction of immersive VR environments," In Proc. IEEE VR Conf., pp. 217-222, 2005.

[9] A. Elgammal, et al, "Background and Foreground Modeling Using Nonparametric Kernel Density for Visual Surveillance," *Proceedings of the IEEE*, Vol. 90, No.7, pp. 1151-1163, 2002.

[10] L. Pero, J. Guan, E. Brau, J. Schlecht, and K. Barnard, “Sampling Bedrooms,” In Proc. CVPR, pp. 2009-2016, 2011.

[11] V. Hedau, D. Hoiem, and D. Forsyth, “Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry,” In Proc. ECCV, pp. 224-237, 2010.

[12] V. Hedau, D. Hoiem, and D. Forsyth, “Recovering the Spatial Layout of Cluttered Rooms,” In Proc. ICCV, pp. 1849-1856, 2009.

[13] H. Wang, S. Gould, and D. Koller, “Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding,” In Proc. ECCV, pp. 435-449, 2010.

[14] D. C. Lee, A. Gupta, M. Hebert, and T. Kanade, “Estimating Spatial Layout of Rooms using Volumetric Reasoning about Objects and Surfaces,” In Advances in Neural Information Processing Systems (NIPS), Vol. 24, November, 2010.

[15] D. Fouhey, V. Delaitre, A. Gupta, A. Efros, I. Laptev, and J. Sivic, “People Watching: Human Actions as a Cue for Single-View Geometry,” In Proc. ECCV, pp. 732-745, 2012.

[16] V.Hedau, D.Hoiem and D.Forsyth, “Recovering Free Space of Indoor Scenes from a Single Image,” In Proc. CVPR, pp. 2807-2814, 2012.

[17] G. Tsai and B. Kuipers, “Dynamic visual understanding of the local environment for an indoor navigating robot,” In Proc. IROS, pp. 4695-4701, 2012.

[18] G.Tsai, C. Xu, J. Liu and B. Kuipers, “Real-time indoor scene understanding using Bayesian filtering with motion cues,” In Proc. ICCV, pp. 121 - 128, 2011.

[19] J.C. Bazin, C. Demonceaux, P. Vasseur, I.S. Kweon, "Motion Estimation by Decoupling Rotation and Translation in Catadioptric Vision," *Computer Vision and Image Understanding*, Vol. 114, Issue 2, pp. 254-273, 2010.

[20] D. Gutierrez-Gomez, L. Puig and J. J. Guerrero, "Full Scaled 3D Visual Odometry from a Single Wearable Omnidirectional Camera," In Proc. IROS, pp. 4276 - 4281, 2012.

[21] Shigang Li, Ying Hai, "Easy Calibration of a Blind-Spot-Free Fisheye Camera System Using a Scene of a Parking Space," *IEEE Transactions on ITS, 12(1)*: pp. 232-242, 2011.

[22] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3-d scene structure from a single still image," In *PAMI*, Volume. 31, Issue. 5, pp. 824-840, 2008.

[23] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," In *JMLR*, Vol. 6, pp.1453-1484, 2005.

[24] Ozisik. N.D., López-Nicolás. G., and Guerrero. J.J, "Scene structure recovery from a single omnidirectional image," In ICCV Workshops, pp. 359–366, 2011.

[25] Jason Omedes, Gonzalo López-Nicolás, and José Jesús Guerrero, "Omnidirectional Vision for Indoor Spatial Layout Recovery," Frontiers of Intelligent Autonomous Systems 2013, pp. 95-104, 2013.

[26] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," In Proc. ICCV, pp. 941 - 947, Vol.2, 1999.

[27] E. Tretyak, O. Barinova, P. Kohli, and V. Lempitsky, "Geometric Image Parsing in Man-Made Environments," In *IJCV*, Vol. 97, Issue. 3, pp. 305-321, 2012.

[28] S. Satkin, J. Lin, and M. Hebert, “Data-Driven Scene Understanding from 3D Models,” In Proc. BMVC, 2012.

[29] D. C. Lee, M. Hebert, and T. Kanade, “Geometric Reasoning for Single Image Structure Recovery,” In Proc. CVPR, pp. 2136–2143, 2009.

[30] Alex Flint, Christopher Mei, David W. Murray, and Ian D. Reid, “A Dynamic Programming Approach to Reconstructing Building Interiors,” In Proc. ECCV, pp. 394--407, 2010.

[31] J.C. Bazin, C. Demonceaux, P. Vasseur, I.S. Kweon, “Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment,” In *International Journal of Robotics Research*, Vol 31, Issue 1, pp. 63-81, 2012.

[32] L. Puig, J. Bermudez-Cameo and J. J. Guerrero, “Self-orientation of a hand-held catadioptric system in man-made environments,” In Proc. ICRA, pp. 2549 - 2555, 2010.

[33] S. Li and H. Jia, “Vanishing point estimation by spherical gradient”, In Proc. ICPR, pp. 902-905, 2012.

[34] J. Kopf, B. Chen, R. Szeliski and M. Cohen, “Street slide: browsing street level imagery”, In Proc. SIGGRAPH ’2010, Vol.29, Issue.4, No.96, 2010.

[35] D. Voorhies and J. Foran, “Reflection vector shading hardware”, In Proc. SIGGRAPH’94, pp.163-166, 1994.

[36] R.S. Wright, N. Haemel, G. Sellers and B. Lipchak, “OpenGL Supper Bible”, 5th Edition, Addison-Wesley Professional, 2010.

[37] W. Heidrich and H.P. Seidel, “View-independent environment map”, In Proc. ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware, pp.39-45, 1998.

[38] Pixar, “The RenderMan Interface”, Version 31.1. Pixar, San Rafal, CA, 1989.

[39] C.H. Chen and A.C. Kak, “A robot vision system for recognizing 3-Dobjects in low-order polynomial time”, *IEEE Trans. Systems, Man, and Cybernetics*, Vol.19, No.6, pp.1535-1563, 1989.

[40] S. Li and Y. Hai, “A full-view spherical image format”, In Proc. ICPR, pp.2337-2340, 2010.

[41] S. Li, H. Jia and I. Nakanishi, “Interpolation of discrete spherical image”, 5th International Congress on Image and Signal Processing, pp.713-717, 2012.

[42] R.C. Gonzalez and R.E. Woods, “Digital Image Processing”, 3rd Edition, Prentice Hall, pp.354, 2008.

LIST OF PUBLICATION

Journals:

[1] Hanchao Jia and Shigang Li, "Realization of Horse Vision System Based on Fisheye Cameras", *IEEJ Transactions on Electronics Information and Systems*, Vol.132, No.12, pp.1992-1998, 2012 (in Japanese).
Corresponding to Chapter 2.

[2] Hanchao Jia and Shigang Li, "Fast Generation of Perspective Display from Spherical Bubble", *Journal of Signal Processing*, Vol. 18, No. 2, pp. 111-119, 2014.
Corresponding to Chapter 4.

International Conferences and Workshops:

[1] Hanchao Jia and Shigang Li, "Scene Analysis based on Horse Vision System", 12th IAPR Conference on Machine Vision Applications (MVA), pp.267- 270, Jun, 2011.
Corresponding to Chapter 2.

[2] Shigang Li, Hanchao Jia and Isao Nakanishi, "Interpolation of discrete spherical image", 5th International Congress on Image and Signal Processing (CISP), pp.602-606, Oct, 2012.

[3] Shigang Li and Hanchao Jia, "Vanishing point estimation by spherical gradient", 21st International Conference Pattern Recognition (ICPR), pp.902-905, Nov, 2012.

[4] Shigang Li, Hanchao Jia and Isao Nakanishi, "Line detection by spherical gradient", International Conference on Image Analysis and Recognition, pp.318-325, Jun, 2013.

[5] Hanchao Jia and Shigang Li, “Estimating the Structure of Rooms from a Single Fisheye Image”, Recent Advances in Computer Vision and Pattern Recognition in Conjunction with ACPR 2013, pp. 818 – 822, Nov, 2013.
Corresponding to Chapter 3.

[6] Shigang Li, Hanchao Jia and Isao Nakanishi, “Computing optical flow from bio-inspired spherical retina”, IEEE International Conference on Mechatronics and Automation (ICMA), pp. 547 – 552, Aug. 2014.

Non-Refereed International Conferences and Workshops:

[1] Hanchao Jia and Shigang Li, “ Estimating Structure of Rooms from Full-view Image”, Scene Understanding Workshop in conjunction with CVPR 2014, Jun, 2014.
Corresponding to Chapter 3.