PARAMETER EXTRACTION AND IMAGE ENHANCEMENT FOR CATADIOPTRIC OMNIDIRECTIONAL CAMERAS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF INFORMATICS

OF

THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

YALIN BAŞTANLAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF INFORMATION SYSTEMS

APRIL 2005

Approval of the Graduate School of Informatics

_____

Assoc. Prof. Dr. Nazife BAYKAL

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. Onur DEMİRÖRS

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. Yasemin YARDIMCI

Supervisor

Examining Committee Members

Prof. Dr. A. Enis ÇETİN (Bilkent Univ., EE) _____

Assoc. Prof. Dr. Yasemin YARDIMCI (METU, II) _____

Assoc. Prof. Dr. Aydın ALATAN (METU, EE) _____

Asst. Prof. Dr. Erkan MUMCUOĞLU (METU, II) _____

Dr. Adem MÜLAYİM (Bilkent Univ., EE) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

First Name, Surname: YALIN BAŞTANLAR

Signature:

ABSTRACT

PARAMETER EXTRACTION AND IMAGE ENHANCEMENT FOR CATADIOPTRIC OMNIDIRECTIONAL CAMERAS

BAŞTANLAR, Yalın

M.Sc., Department of Information Systems

Supervisor: Assoc. Prof. Dr. Yasemin YARDIMCI

April 2005, 114 pages

In this thesis, catadioptric omnidirectional imaging systems are analyzed in detail. Omnidirectional image (ODI) formation characteristics of different camera-mirror configurations are examined and geometrical relations for panoramic and perspective image generation with common mirror types are summarized.

A method is developed to determine the unknown parameters of a hyperboloidal-mirrored system using the world coordinates of a set of points and their corresponding image points on the ODI. A linear relation between the parameters of the hyperboloidal mirror is determined as well. Conducted research and findings are instrumental for calibration of such imaging systems.

The resolution problem due to the up-sampling while transferring the pixels from ODI to the panoramic image is defined. Enhancing effects of standard interpolation methods on the panoramic images are analyzed and edge detection-based techniques are developed for improving the resolutional quality of the panoramic images. Also, the projection surface alternatives of generating panoramic images are evaluated.

Keywords: Omnidirectional camera, catadioptric camera, panoramic vision, camera calibration, interpolation

ÖZ

KATADİOPTRİK TÜMYÖNLÜ KAMERALAR İÇİN PARAMETRE ÇIKARIMI VE İMGE İYİLEŞTİRMESİ

BAŞTANLAR, Yalın

Yüksek Lisans, Bilişim Sistemleri Bölümü

Tez Yöneticisi: Doç. Dr. Yasemin YARDIMCI

Nisan 2005, 114 sayfa

Bu tezde, katadioptrik tümyönlü görüntüleme sistemleri detaylı olarak incelenmiştir. Farklı kamera-ayna konfigürasyonları için tümyönlü imge oluşum özellikleri ve panoramik ve perspektif görüntü oluşumu için geometrik bağıntılar özetlenmiştir.

Bir hiperbolik aynalı görüntüleme sisteminin bilinmeyen parametrelerinin çıkarımı için bir nokta setinin gerçek dünya koordinatlarını ve tümyönlü imge üzerindeki eşleniklerini kullanan bir yöntem geliştirilmiştir. Ayrıca, hiperbolik ayna parametreleri arasında doğrusal bir ilişki tespit edilmiştir. Yapılan araştırma ve buluntular bu tür görüntüleme sistemlerinin kalibrasyonu için önem taşımaktadır.

Tümyönlü görüntüden panoramik görüntüye geçiş sırasında yaşanan üstörneklemeden kaynaklanan çözünürlük problemi tanımlanmıştır. Standart aradeğerleme yöntemlerinin panoramik imgeler üzerindeki iyileştirme etkileri incelenmiş ve çözünürlük kalitesini artırmak amacıyla kenar sezimi tabanlı teknikler geliştirilmiştir. Ayrıca, panoramik görüntü oluşumu için projeksiyon yüzeyi alternatifleri değerlendirilmiştir.

Anahtar Kelimeler: Tümyönlü kamera, panoramik kamera, katadioptrik kamera, kamera kalibrasyonu, aradeğerleme

ACKNOWLEDGMENTS

I express very cordial appreciation to Assoc. Prof. Dr. Yasemin Yardımcı for her valuable supervision throughout the research. With her precious knowledge in different study areas, she oriented me so well that I have been able to complete this thesis.

I wish to thank Dr. Kazumasa Yamazawa, Dr. Simon Baker, Prof. Dr. Enis Çetin, Dr. Adem Mülayim and Radu Orghidan for their advises concerning the research held in this thesis. I also thank Assoc. Prof. Dr. Aydın Alatan and Asst. Prof. Dr. Erkan Mumcuoğlu for evaluating this work.

This thesis work is supported by BAP-2004-07-04-04: “Determination of omnidirectional camera system parameters using the information obtained from the images” and DPT 2004K120720: “Multimedia Archive System for E-Government” projects.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

LIST OF ABBREVATIONS

ODI	Omnidirectional Image
SVP	Single-viewpoint
RMSe	Root Mean Square Error
MAD	Mean Absolute Deviation

CHAPTER 1

Introduction

1.1 Panoramic Vision Background

Enlarged field of view has always been an important goal for imaging. The word ‘panorama’ is a combination of two Greek words; ‘pan’ meaning ‘all’ and ‘horama’ meaning ‘sight’. First attempts to generate a panoramic view were in the form of paintings in the late 1700’s. Sketches are transferred on round canvas to create realistic wide images of landscapes. Attempts for photographic panorama were made at late 1800’s, in which a number of photographs taken at different directions are attached each other in order to create a full panorama. More historical information is given in Benosman [1].

Technological development put forward modern ways to achieve the large fields of view such as using special lenses (ex: fisheye lens) or special mirrors together with the cameras. The method of capturing multiple images and stitching them together to create panoramas is still used [2 - 6] and it possesses the advantage of providing high resolution in some applications, however considerable amount of post-processing is needed for real-time applications.

Today, the term ‘omnidirectional’ is used for the cameras that are capable of sensing the light rays coming from all directions. Fish-eye lenses also have extended field of views up to a hemisphere and are used for omnidirectional viewing [7]. However, they usually fail to satisfy a fixed viewpoint, which makes the construction of

perspective images difficult. In addition, to capture a hemispherical view, the fish-eye lens must be quite complex and large and hence expensive.

Most of the omnidirectional cameras are catadioptric systems which means they use combinations of mirrors and lenses. The term ‘catadioptrics’ comprises ‘catoptrics’; the science of reflecting surfaces (mirrors) and ‘dioptrics’; the science of refracting elements (lenses). Rees [8] is the first to patent a catadioptric omnidirectional capturing system using a hyperboloidal mirror and a normal perspective camera in 1970. Since then, geometric properties of the mirrors and image formation with varying mirror-lens combinations are extensively studied [9-15].

1.2 Rationale of the Study

In the last few years, number of applications of omnidirectional viewing significantly increased due to the increase in the speed of processing and ease of device production. Application fields include surveillance, teleconferencing, advertising in real estate, tourism, robot navigation and virtual environment construction.

An omnidirectional viewing device ideally has the capability of viewing 360º in all directions. It is not practical to produce a ‘true’ omnidirectional sensor, therefore manufactured cameras usually provide 360º horizontal view and a sufficient field of vertical view. Nowadays, considerable amount of effort is spent on the design of mirrors with enlarged vertical field of views, increased resolution and decreased imaging errors [16 - 26]. In parallel, calibration work is performed in order to increase the accuracy of the applications and broaden the usage of catadioptric sensors [27 - 36].

Catadioptric omnidirectional cameras are widely used for research purposes as well. Study fields, which have already been using normal-view cameras extended their scope in a way to involve these cameras. Example studies are in the fields of object detection [37], tele-presence [38 - 40], robot navigation [41 - 43], stereo systems [44], super-resolution [45 - 47], optical flow estimation [48, 49], reconstruction and modeling [50-55].

Within this context, to have the best output from the omnidirectional sensors study on the geometries and the calibration parameters of the mirror-lens combinations has undeniable importance. Achieving more realistic representations of the real world would increase the effectiveness of these devices.

1.3 Approach

In this thesis, omnidirectional images obtained through catadioptric camera systems are inspected in detail. Omnidirectional image formation characteristics of different camera-mirror configurations are examined and generated panoramic images are analyzed. Succeeding work is based on the observed problems and held in three main topics (study areas); inspection of mirror reflections and extraction of imaging system parameters using omnidirectional images, panoramic projection alternatives and enhancement of generated panoramas.

1.4 Road Map

In Chapter 2, catadioptric omnidirectional image formation background is given in order to introduce the reader to the geometrical properties of catadioptric sensors.

Chapter 3 provides, firstly the previous work on the calibration of catadioptric omnidirectional systems and then our study to determine the parameters of hyperboloidal mirrors from omnidirectional images.

Chapter 4 presents the work held in this thesis concerning the analysis of resolutional properties of omnidirectional mirror types and panoramic projection alternatives.

Chapter 5 presents the work held in this thesis in order to enhance the regions of panoramic images having low quality due to the resolutional deficiency.

Chapter 6 concludes the thesis by summarizing the overall study, evaluating the results and discussing further work possible in this area.

CHAPTER 2

Catadioptric Omnidirectional Image Formation

2.1 Image Formation Basics

Omnidirectional image formation in a catadioptric system is visualized in Figure 2.1. Light rays coming from the world point, P(X,Y,Z), are reflected at the mirror surface and go through the lens of the camera in order to take place in the image plane. A typical omnidirectional sensor (mirror-lens combination) is shown in Figure 2.2.

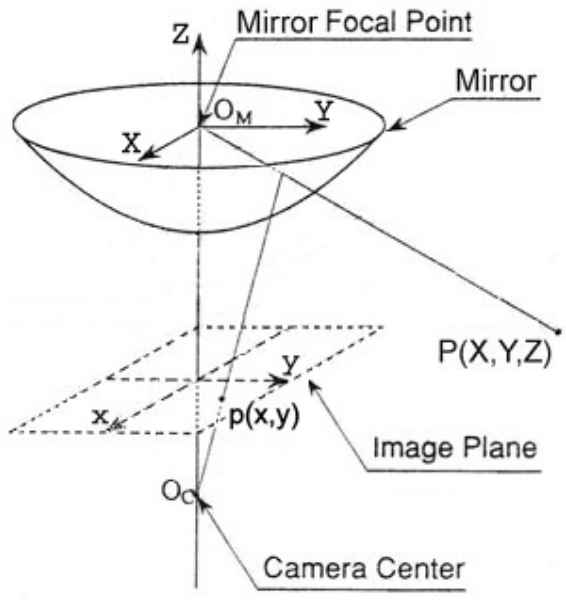

Figure 2.1: Omnidirectional image formation with a catadioptric omnidirectional imaging system

Figure 2.2: A typical omnidirectional sensor (www.remotereality.com)

The result is an omnidirectional image (Figures 2.3 and 2.5). Then, panoramic image (Figure 2.4) and perspective projection (Figure 2.6) are obtained by applying geometric conversion to the omnidirectional image (ODI).

Figure 2.3: Omnidirectional image (ODI), obtained through a hyperboloidal mirror generated on the image plane of a regular camera (courtesy of Czech Technical Univ.)

Figure 2.4: Panoramic image generated from the omnidirectional image (ODI) in Figure 2.3

Figure 2.5: An omnidirectional image obtained through a paraboloidal mirror [22]

Figure 2.6: Perspective image generated from Figure 2.5

2.2 Omnidirectional Mirror Design

2.2.1 Single-viewpoint Phenomenon

Catadioptric omnidirectional sensors are mainly divided into two categories: 'Single-viewpoint' and 'Non single-viewpoint'. 'Single-viewpoint' means that the omnidirectional image is formed by the light rays that are targeting a single point. For instance, in the mirror shown in Figure 2.7, light rays coming from the world points A, B and C and targeting the focal point (single viewpoint) of the hyperboloidal mirror are reflected on the mirror surface so that they will pass through the pinhole (camera center). This single viewpoint acts a virtual pinhole through which the scene is viewed as it occurs in regular cameras. However, for the system shown in Figure 2.8, directions of the light rays that are used for image formation do not intersect at a certain point as in the single-viewpoint case, therefore a single point through which the scene can be viewed cannot be defined. Single viewpoint constraint provides quick conversion of geometrically correct panoramic and perspective images because they are generated as seen from the mentioned viewpoint.

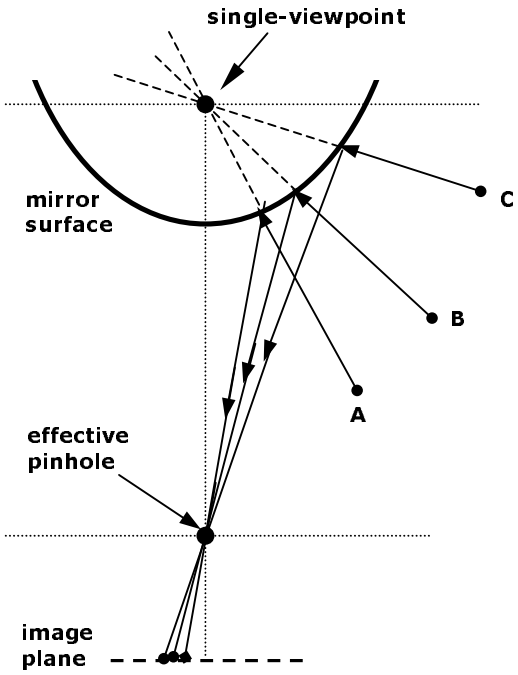

Figure 2.7: Directions of light rays in a single-viewpoint system

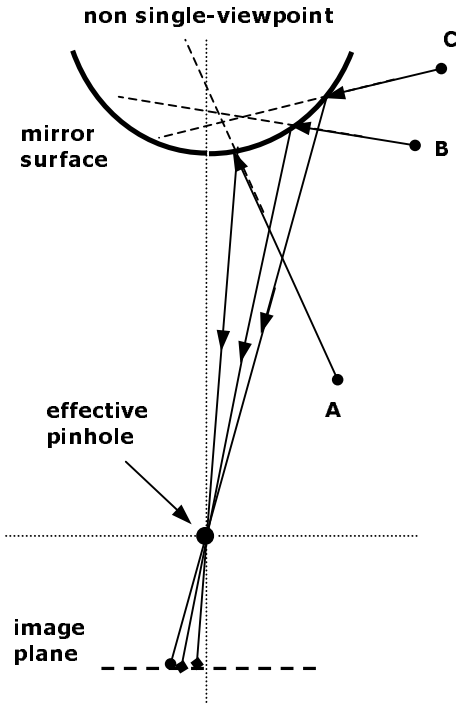

Figure 2.8: Directions of light rays in a non single-viewpoint system

Projection surfaces are defined in a way that this viewpoint will be the virtual pinhole of a regular camera. Doing so, perspective images can be generated using the standard camera-viewpoint geometry.

2.2.2 Single-viewpoint Analysis of Mirror Types

Baker and Nayar [10] inspected in detail the geometries of single viewpoint mirrors. Since the cross sectional profiles of the mirrors of omnidirectional sensors does not change around the rotational axis for 360°, the cross sections of the mirror types of catadioptric sensors are examined in this paper.

Planar mirrors have single viewpoint (Figure 2.9). Although a planar mirror cannot be an omnidirectional viewer by itself, several planar mirrors (e.g. 4 mirrors for a square-based pyramid system), and a camera for each, can come together to constitute a single viewpoint omnidirectional viewer [56].

In conical mirrors, the pinhole of the camera must be located at the apex of the cone to be a single viewpoint (Figure 2.10). This means that only the rays that graze the mirror surface enter the pinhole of the mirror, which is a problem that cannot be solved practically. Therefore, the conical mirror is not proper to be used as a single viewpoint catadioptric sensor.

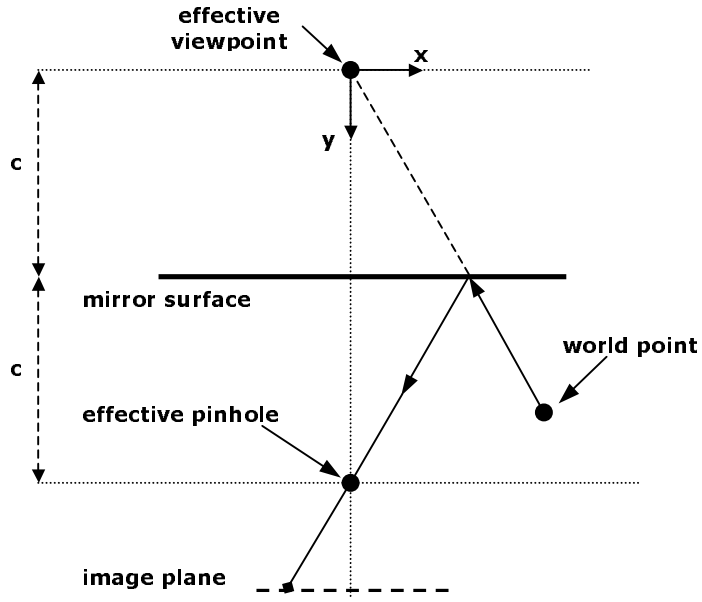

Figure 2.9: Planar mirror geometry

In spherical mirrors, the center of the sphere can be considered as viewpoint. However, all light rays targeting this viewpoint would be reflected back and diverge. Therefore, an effective pinhole does not occur outside the mirror making the sphere a non single-viewpoint mirror (Figure 2.11).

Ellipsoidal mirrors have single viewpoint property (Figure 2.12). Ellipse is a geometrical shape that has two focal points. Light rays coming from the world point and passing through one of the focal points (effective viewpoint) are reflected at the inner ellipse surface so that they will pass through the other focal point (effective pinhole). However, ellipsoidal mirrors should be used in concave form to behave as a single viewpoint system. This causes the field of view to be restricted to one hemisphere, which is not very desirable. Objects should be placed below the horizontal level of the camera to be seen in the omnidirectional image.

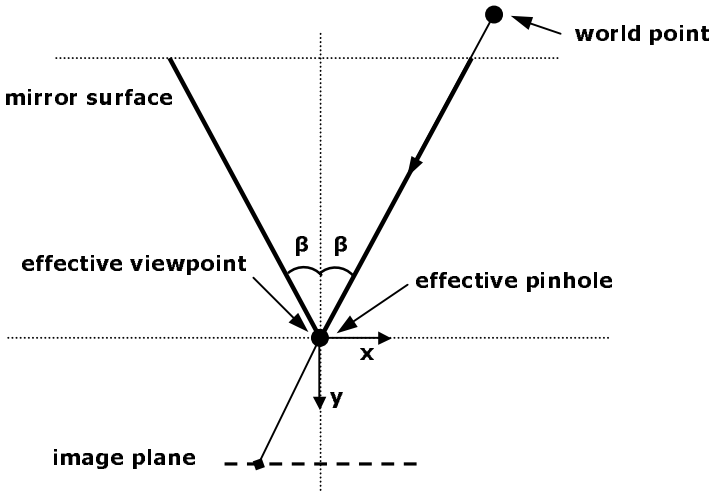

Figure 2.10: Conical mirror geometry

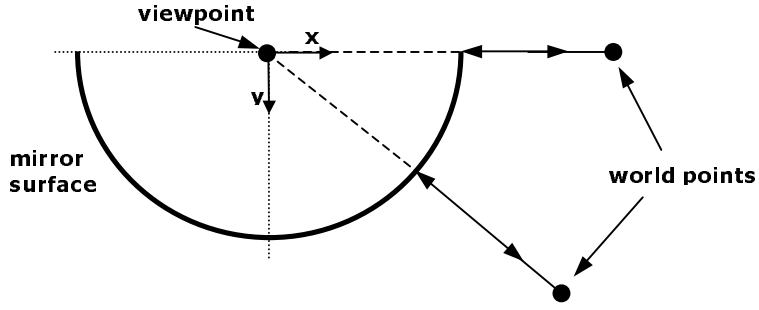

Figure 2.11: Spherical mirror geometry

Hyperboloidal mirrors have single viewpoint and the rays targeting that viewpoint are reflected so that they pass through an effective pinhole (Figure 2.13). The mentioned viewpoint and pinhole are located at the two focal points of the hyperbola, which is the cross sectional profile of the hyperboloidal mirror.

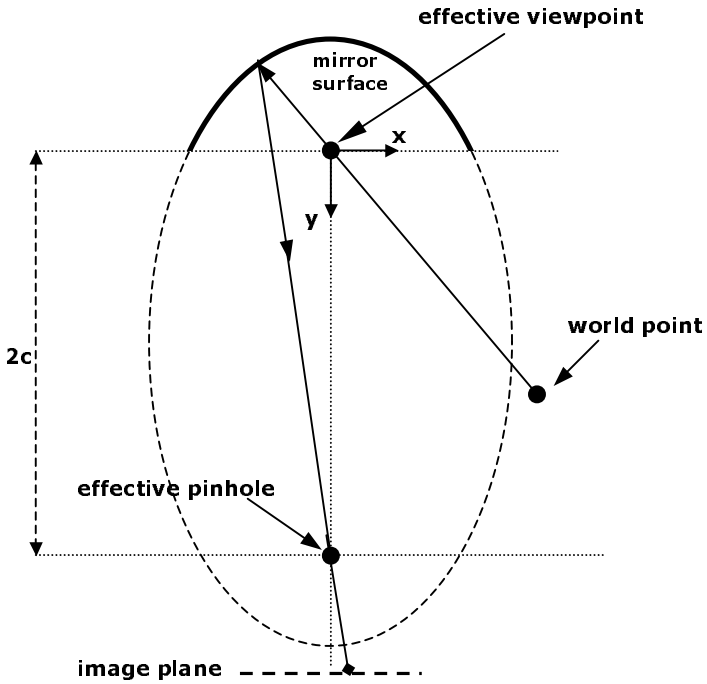

Figure 2.12: Ellipsoidal mirror image formation geometry

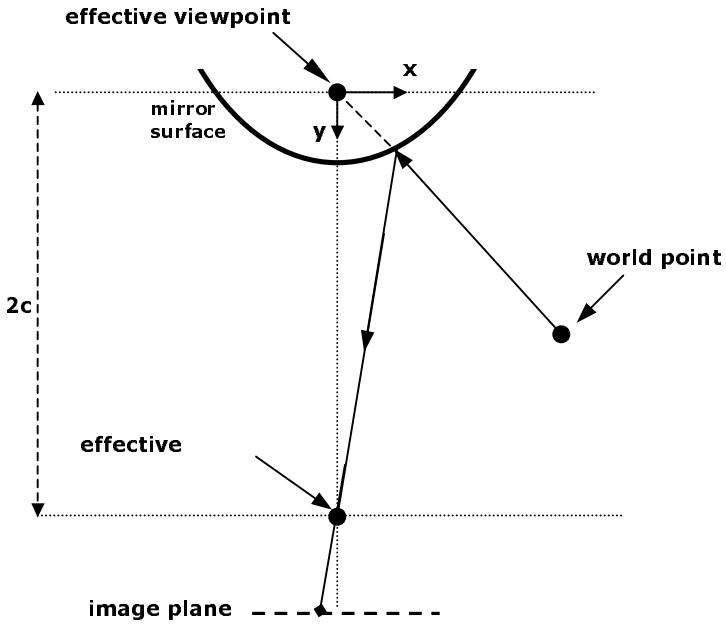

Figure 2.13: Hyperboloidal mirror image formation geometry

Paraboloidal mirrors also have single viewpoint (the focal point of parabola), but the rays targeting that viewpoint are reflected orthogonally, which requires the use of a telecentric lens to collect the parallel rays (Figure 2.14).

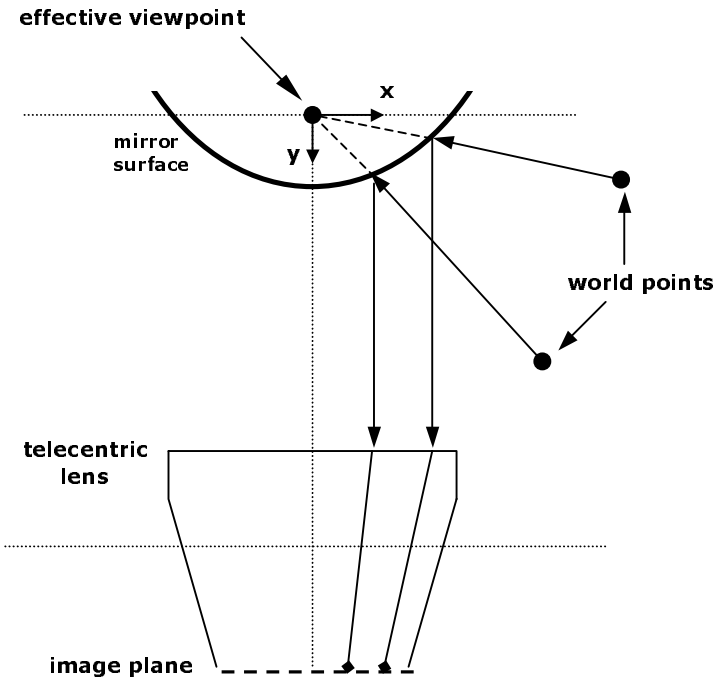

Figure 2.14: Paraboloidal mirror image formation geometry

2.2.3 Use of Non Single-viewpoint Mirrors

Although single-viewpoint mirrors are desirable for efficient projection generation, non single-viewpoint mirrors may be preferred to achieve uniform resolution and wider field of view and/or ease of manufacturing. Use of spherical and conical mirrors is shown in Figure 2.15.

In addition to the mirrors that do not satisfy single-viewpoint constraint, also an incorrectly aligned system could cause non single-viewpoint. When an imaging system does not maintain a single viewpoint, a locus of viewpoints is formed, called a caustic. Each pixel in the image maps to a point on the caustic surface (Figure 2.16). Swaminathan et al. [13] examined these viewpoint loci with regard to field of view, resolution and other geometric properties.

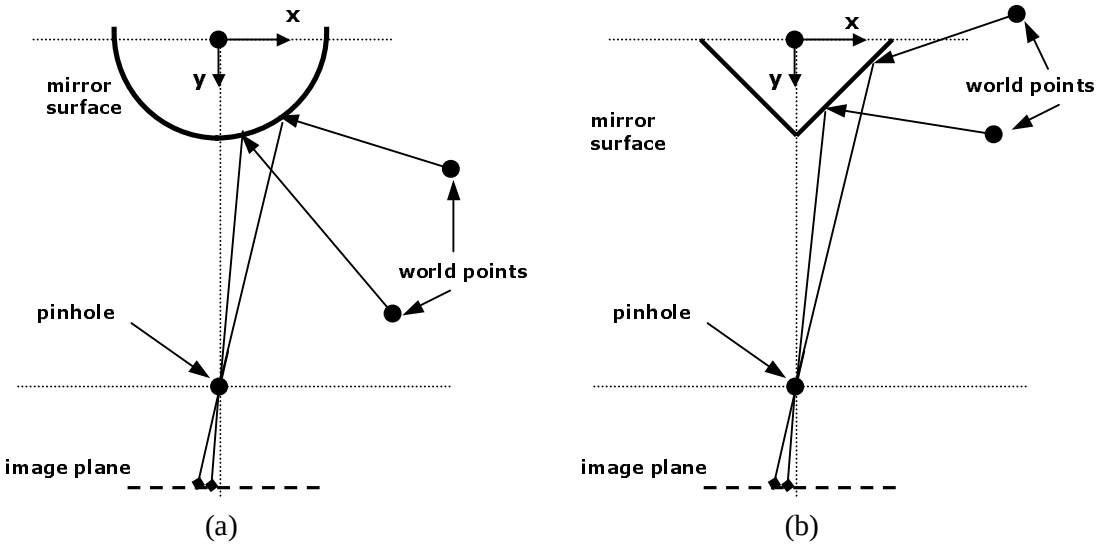

Figure 2.15: Usage geometry of non single-viewpoint mirrors; a) spherical b) conical

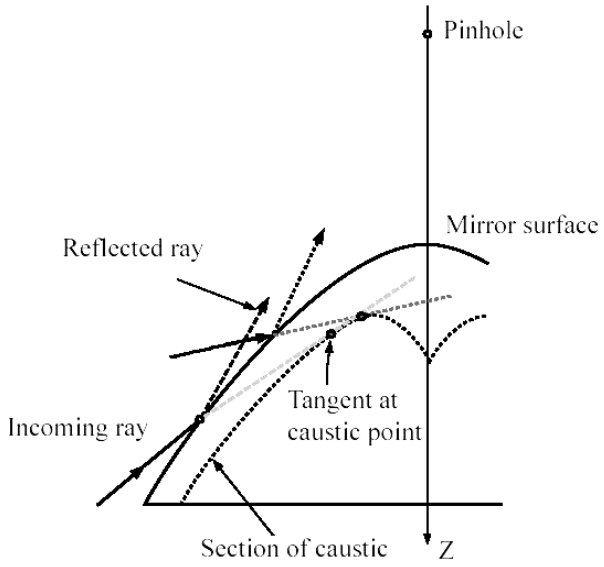

Figure 2.16: Caustic formation in non single-viewpoint mirrors

Ishiguro [18] approached the issue from practical point of view and evaluated the most common mirror types (spherical, conical, hyperboloidal and paraboloidal) including economic and practical advantages and disadvantages. Hyperboloidal and paraboloidal mirrors have the advantage of being single viewpoint, however, manufacturing cost of spherical and conical mirrors are very low. Spherical mirrors are suitable for acquiring images of objects under the omnidirectional camera system and focusing on a spherical mirror by a viewer camera is easier when compared to

other mirror types because its bottom part typically has a larger radius of curvature. Considering the material that the mirror is made of, it is stated that the standard process of making mirrors from glass and aluminum coating is difficult and costly. In case machines are available for precise finishing of metals, brass and chrome can be used instead of glass and aluminum respectively. He also suggests using a black needle along the mirror-camera axis in order to prevent the internal reflections of the transparent cylinder that holds the mirror.

2.2.4 Complex Mirror Systems

Current studies on omnidirectional mirror design aims to create different mirror surfaces and reflective systems that consist of more than one mirror.

Nayar and Peri [21] worked on folded mirror systems that use multiple mirrors in order to obtain smaller omnidirectional devices with wider views. Two mirrored systems that are geometrically equivalent to single mirror systems are summarized. They consist of a combination of planar, paraboloidal, hyperboloidal and elliptic mirrors. It is mentioned that although geometrically equivalent with the single mirror systems, folded systems may have reduced optical aberrations since a convex and a concave mirror compensate the curvature of each other.

Nayar [22] examined field of view of the paraboloidal mirror system. A hemisphere is visible when the paraboloid is cut at the level of focal point of the mirror. In order to obtain a true omnidirectional (360° x 360°) viewer, two omnidirectional viewers can be placed back-to-back such that their focal points coincide. If only a single mirror is used, it is possible to increase the vertical viewing angle by cutting the paraboloid at a higher level.

Single mirrors have non-uniform resolutions. Due to the curvature of the mirror, regions in the omnidirectional mirror do not cover equal amount of area in the scene. Conroy and Moore [25] derived mirror surfaces that are resolution invariant vertically, so that adjacent pixels in omnidirectional image correspond to the real world points that are vertically equi-distant from each other. In this paper, stereo omnidirectional systems are also introduced. They are constructed by two coaxial,

axially symmetric mirror profiles. Proposed mirror has two lobes with different sizes and these two mirror lobes are attached in a way that smaller mirror will be at the top (vertex) of the larger one.

Swaminathan et al. [26] presented a linear method to determine the mirror shape that meets a designer's requirements. The requirements are specified as a map from pixels to scene points which they call 'image to scene map'.

Hicks and Bajcsy [17] exhibited a mirror design that views wide horizontal area under the mirror and reflects an undistorted (perspective) omnidirectional image of this area. However, resultant mirror design is not proper for vertical regions.

Gaspar et al. [20] summarizes the constant horizontal, vertical and angular resolution issues that are studied earlier and combines these definitions in a mirror. The surface of the mirror is manufactured so that it will achieve uniform resolution when used with a specific log-polar camera.

2.3 Panoramic and Perspective Image Generation

In this section, image generation with most commonly used single-viewpoint mirrors; paraboloidal and hyperboloidal mirror is explained in detail. Also, projection geometry of the spherical mirror (a commonly used non single-viewpoint mirror type) is explained.

2.3.1 Paraboloidal Mirrors

Parabola is a geometrical shape formed by points, which are equi-distant from a line (*l*) and a point (*p*, focal point) as shown in Figure 2.17.

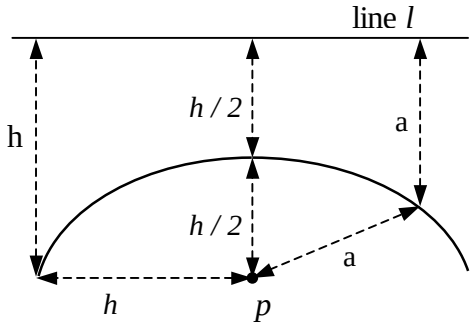

Figure 2.17: Demonstration of parabola

Paraboloid is a quadratic surface whose cross section is a parabola. Paraboloidal mirror has one parameter that defines the surface profile, *h*, the horizontal distance from focal point to the mirror surface:

$$z = \frac{h^2 - r^2}{2h}. \tag{2.1}$$

In this expression, *z* is the vertical coordinate and *r* is the horizontal coordinate. In a paraboloid, $r = \sqrt{x^2 + y^2}$ and for a radially symmetric paraboloid as in mirrors, *x*=*y*.

Peri and Nayar [57] summarized panoramic and perspective image generation in systems with paraboloidal mirrors. Light rays coming from the world points and targeting the mirror focal point are reflected orthogonal on the mirror surface. Then, these rays go through the telecentric lens to form the omnidirectional image. For the sake of simplicity, in Figure 2.18, omnidirectional image plane is represented as a circle having the same diameter with the paraboloidal mirror. This image plane (and the telecentric lens) is independent of projection surface and can be at any elevation since the light rays are reflected orthogonally.

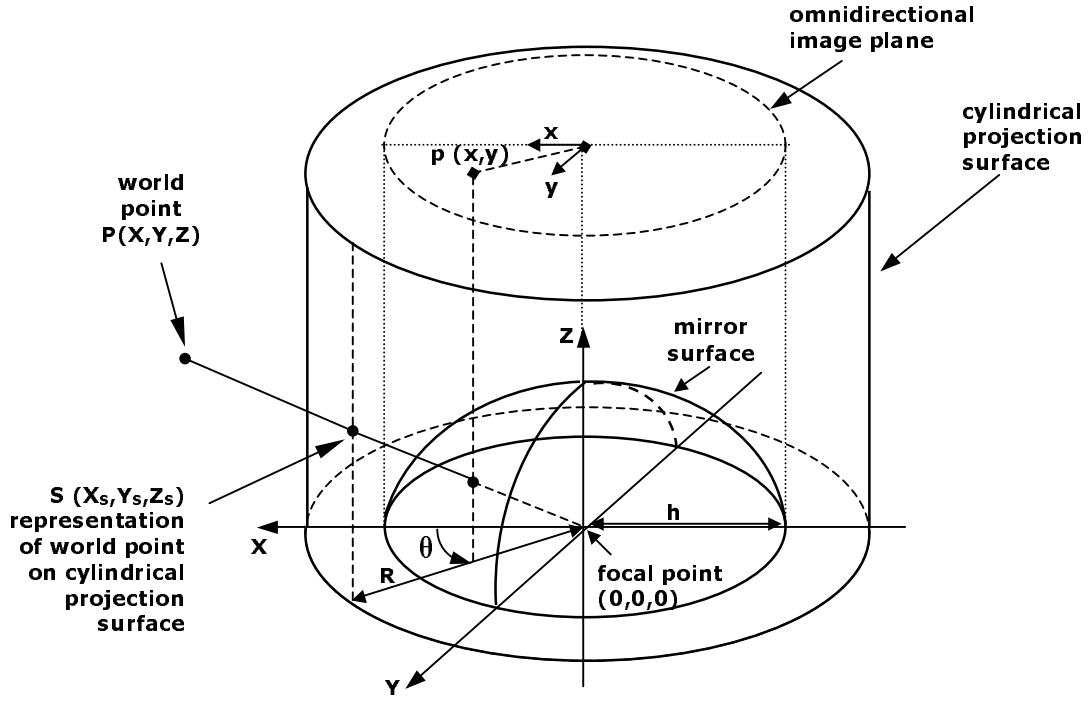

Figure 2.18: Cylindrical (panoramic) projection with paraboloidal mirror

Panoramic image can be defined as the scene projected to a focal point-centered cylinder around the mirror (Figure 2.18), therefore 'cylindrical projection' term is widely used.

To find the intensity value of a pixel in the cylindrical (panoramic) projection, corresponding pixel coordinates in the omnidirectional image (*x*, *y*) are:

$$x = \frac{h}{Z_S + \sqrt{{X_S}^2 + {Y_S}^2 + {Z_S}^2}} \cdot X_S \qquad y = \frac{h}{Z_S + \sqrt{{X_S}^2 + {Y_S}^2 + {Z_S}^2}} \cdot Y_S \qquad (2.2)$$

where, *h* is the parameter of the paraboloid and X_S,Y_S,Z_S are the 3D coordinates of the pixel in the projection. For a pixel of panoramic image at m^{th} row and n^{th} column:

$$Z_S = m, \qquad X_S = R \cdot \sin\theta, \qquad Y_S = R \cdot \cos\theta, \qquad \theta = \frac{n \cdot 2\pi}{L}$$

R is the radius of the cylinder, *L* is the width of the panoramic image and θ is the radial angle (azimuth) around the rotational axis of the mirror.

Perspective (rectilinear) image is formed when a planar projection surface is defined (Figure 2.19). In perspective view, all straight lines in the scene are rendered straight and all forms are represented distortion-free as in the regular camera images. To find the intensity value of a pixel in the perspective projection, corresponding pixel coordinates in the omnidirectional image (*x*, *y*) are:

$$x = \frac{h}{Z_S + \sqrt{m^2 + n^2 + D^2}} \cdot X_S \qquad y = \frac{h}{Z_S + \sqrt{m^2 + n^2 + D^2}} \cdot Y_S \qquad (2.3)$$

where, *h* is the parameter of the paraboloid, *D* is the distance from the focal point to the horizontal center of the projection surface *and* X_S,Y_S,Z_S are the 3D coordinates of the pixel in the projection. For a pixel of perspective image at m^{th} row and n^{th} column:

$$Z_S = m, \qquad X_S = \sqrt{D^2 + n^2} \cdot \sin\theta, \qquad Y_S = \sqrt{D^2 + n^2} \cdot \cos\theta$$

and θ is the radial angle (azimuth) around the rotational axis of the mirror.

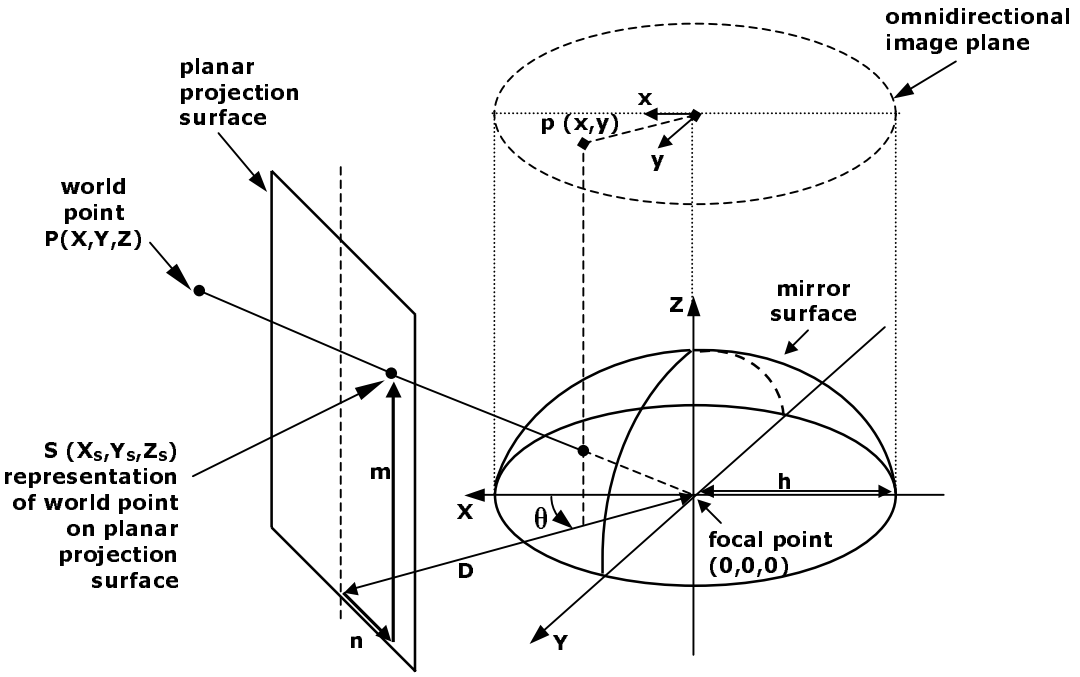

Figure 2.19: Perspective projection with paraboloidal mirror

For both panoramic and perspective projection geometries explained above, the projection surface is assumed to start at the elevation of the mirror focal point, otherwise Z_s will not be equal to m.

2.3.2 Hyperboloidal Mirrors

Hyperbola is a geometrical shape formed by points such that the difference of whose distances from the two focal points (foci) is constant (Equation 2.4). Hyperboloidal mirror has two parameters that define the surface profile, which are *c* (half the distance between two focal points) and *b* (half the distance between two arcs of hyperbola as shown in Figure 2.20. Also, *c* is the hypotenuse ($c^2 = a^2 + b^2$) of such a right triangle that prolongation of the hypotenuse forms an asymptote for the hyperbolic arc.

$$\frac{r^2}{a^2} - \frac{z^2}{b^2} = -1 \tag{2.4}$$

In this expression, *z* is the vertical coordinate and *r* is the horizontal coordinate. In a hyperboloid, $r = \sqrt{x^2 + y^2}$ and for radially symmetric hyperboloids as mirrors, *x*=*y*.

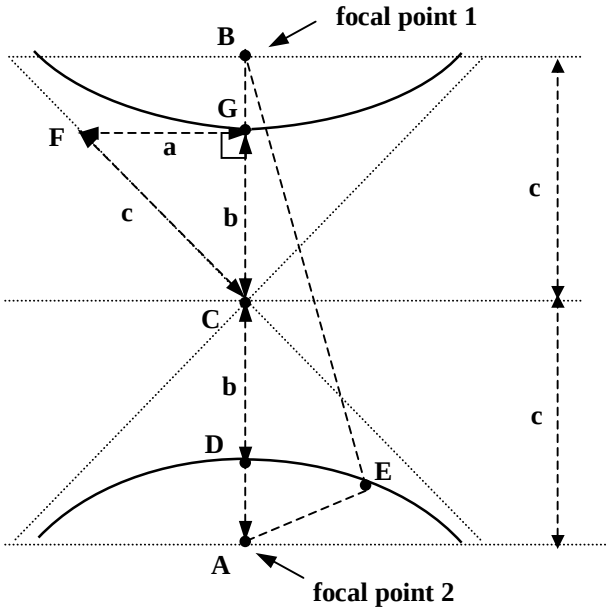

Figure 2.20: Demonstration of hyperbola: |DB|-|DA| = |EB|-|EA|,
C is the center of the hyperbola, c is the hypotenuse of the right triangle
CGF and lies on one of the two asymptotes of the hyperbola

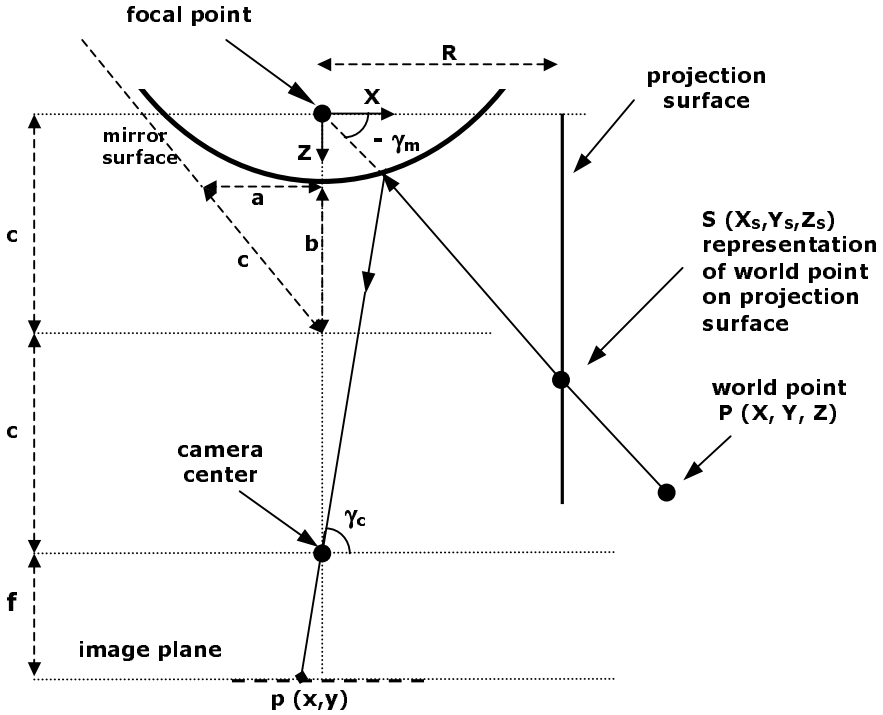

Figure 2.21: Hyperboloidal mirror projection geometry

Yamazawa et al. [15] used an omnidirectional viewing system with a hyperboloidal mirror and Onoe et al. [38] described panoramic image generation in that system (Figure 2.21). Perspective (rectilinear) projections can be performed using the same projection geometry.

Light rays coming from the world points and targeting the mirror focal point are reflected so that they pass through the other focal point of the hyperbola. If θ is the radial angle (azimuth) around the rotational axis of the mirror, the pixel coordinates of the projected point (x, y) in the omnidirectional image are given by:

$$x = \frac{f}{\tan\gamma_c} \cdot \sin\theta \qquad y = \frac{f}{\tan\gamma_c} \cdot \cos\theta \tag{2.5}$$

$$\gamma_c = \tan^{-1}\left[\frac{(b^2 + c^2)\cdot\sin\gamma_m - 2bc}{(b^2 - c^2)\cdot\cos\gamma_m}\right] \qquad -\gamma_m = \tan^{-1}\left[\frac{Z_S}{R}\right]$$

where, b and c are the parameters of the paraboloidal mirror, R is the distance from the focal point to the center of projection surface, f is the focal distance of the camera, X_S,Y_S,Z_S are the 3D coordinates of the pixel in the projection surface, γ_m is the angle between horizontal axis of the mirror and light ray coming from the world point, γ_c is the angle between image plane and light ray coming in the camera.

2.3.3 Spherical Mirrors

Spherical mirrors are not single-viewpoint systems as discussed earlier, i.e. there is not a single point according to which conversion angles and light rays can be modeled. However, knowing the size of the mirror and calculating the locations on the mirror surface where each light ray is reflected, panoramic and perspective image generation can still be performed (Figure 2.22).

At the outer boundary of the omnidirectional image, the light ray is tangent to the spherical mirror surface and the angle α gets its maximum value:

$$\alpha_{\max} = \sin^{-1}\left(\frac{g}{d}\right) \tag{2.6}$$

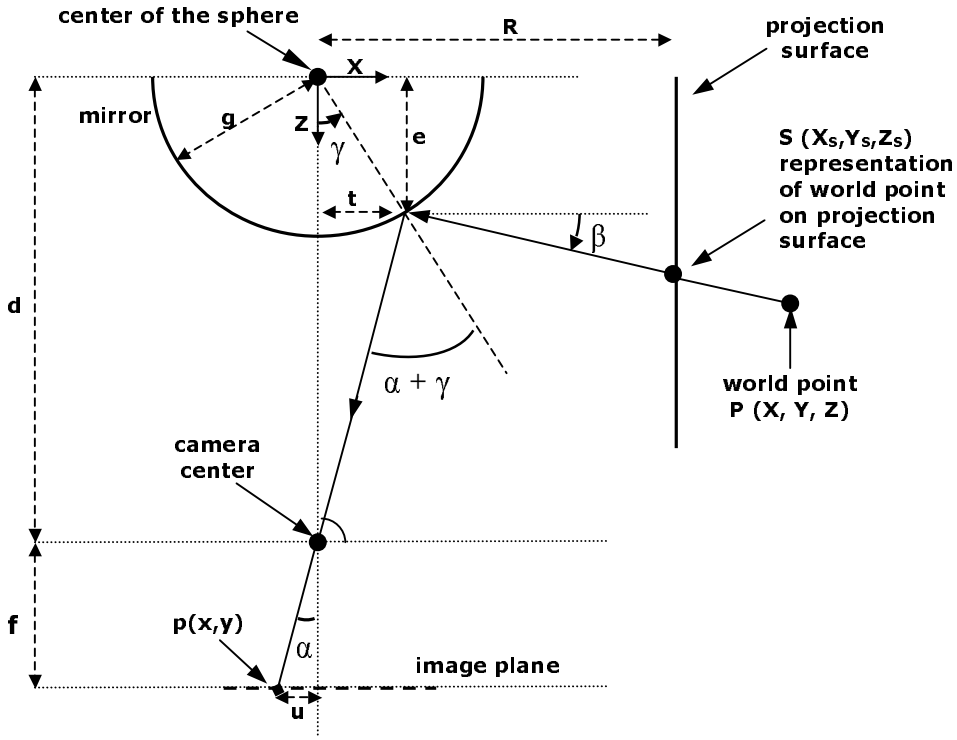

Figure 2.22: Spherical mirror projection geometry

Other geometrical relations in a spherical mirror projection are:

$$\gamma = \sin^{-1}(\frac{d}{g} \cdot \sin\alpha) - \alpha\,, \qquad \beta = \frac{\pi}{2} - 2\gamma - \alpha \tag{2.7}$$

Elevation of the point on the projection surface can be written as:

$$Z_S = e + \tan\beta \cdot (\sqrt{{X_S}^2 + {Y_S}^2} - t) \tag{2.8}$$

where, $e = g \cdot \cos\gamma$ and $t = g \cdot \sin\gamma$.

From Equation 2.8 we obtain

$$\beta = \tan^{-1}\left(\frac{Z_S - e}{\sqrt{{X_S}^2 + {Y_S}^2} - t}\right). \tag{2.9}$$

To find the location of the point in the omnidirectional image we should determine α. We can find α by solving the Equation 2.10 for the [0, α_{max}] interval. Equation 2.10 is the combination of Equations 2.7 and 2.9 both of which are equal to β and is written in terms of α only.

$$\tan^{-1}\left(\frac{Z_S - g\cdot\cos\left(\sin^{-1}\left(\frac{d}{g}\cdot\sin\alpha\right)-\alpha\right)}{\sqrt{X_S{}^2+Y_S{}^2}-g\cdot\sin\left(\sin^{-1}\left(\frac{d}{g}\cdot\sin\alpha\right)-\alpha\right)}\right)-\frac{\pi}{2}-\alpha-2\cdot\sin^{-1}\left(\frac{d}{g}\cdot\sin\alpha\right)=0 \qquad (2.10)$$

Then, the location of the image point is:

$$x = u\cdot\sin\theta \qquad y = u\cdot\cos\theta \qquad (2.11)$$

Where, $u = f\cdot\tan\alpha$ and θ is the azimuth angle.

CHAPTER 3

Calibration of Catadioptric Omnidirectional Cameras

3.1 Camera Calibration Overview

Camera calibration is the task for determining the parameters of an imaging system which relate the coordinates of the points in the scene to the locations of the pixels in the image [58]. Solving a calibration problem requires a set of conjugate pairs; matches of feature points between views and scene. These matches must be correct when true parameters are employed. When a single camera is considered, the calibration problems can be discussed in two main categories; exterior calibration and interior calibration.

Exterior calibration is to determine the relationship between image plane coordinates (x,y) and world (scene) coordinates (X,Y,Z). The parameters used for this relation (translation and rotation) are called extrinsic parameters. Each image point (x,y) defines not a point in the world but a ray starting from the center of projection. In 3D coordinate system of camera, image points actually have coordinates (x,y,f), where f is the distance from image plane to the center of projection (camera constant).

Interior calibration is to determine the internal geometry of the camera. The parameters aimed to be determined are called intrinsic parameters, which are;

- *camera constant*: the distance from image plane to the center of projection
- *principal point*: the location of the origin of the image plane coordinate system
- *lens distortion coefficients* corresponding to changes in image plane coordinates due to the optical imperfections of the camera

- *aspect ratio* is given by the ratio of center-to-center distances between pixels in x and y directions (this may be obtained from the specifications of the camera)
- *scale factors*: distortions in the aspect ratio caused by timing problems in digital electronics

For the basic method of camera calibration, exterior calibration problem is solved first. We can use initial approximations for intrinsic parameters to get a mapping. Once we know (approximate) where the projected points should be, we can use projected locations (x,y) and measured locations (x',y') to correct the intrinsic parameters.

Another method (affine method) for camera calibration combines the exterior and interior calibration problems. Uncorrected image coordinates to true image coordinates are modeled by an affine transformation in the image plane. This affine transformation accounts for camera constant error, translation, rotation, skew error (due to non-orthogonality of camera sensor) and differential scaling (unequal spacing of sensor rows and columns) errors. It does not include lens distortions.

For increased accuracy more equations than unknowns can be used and an overdetermined set of equations is solved using non-linear regression, which is called non-linear method for camera calibration.

3.2 Calibration of Catadioptric Omnidirectional Systems

Calibration techniques of catadioptric omnidirectional cameras are similar to the calibration techniques of regular cameras with the difference that the light rays are reflected from a mirror, which brings complexity to the conversion of world coordinates to camera coordinates. When a single viewpoint mirror is used, because the world points are modeled as seen from this single viewpoint, the directions of the light rays coming into the camera can easily be calculated. If the catadioptric system is non single-viewpoint, then the modeling is cumbersome.

Because catadioptric cameras include reflective elements for image formation, when compared to regular cameras the number of parameters that have to be determined

during a calibration problem increases. The curvature parameters of the mirror and the distance between the mirror and the camera are additional parameters. If we have a catadioptric system specially produced for our work, we know the mirror-lens parameters and the only reason for calibration may be to determine the possible deviations of these parameters. If we use a catadioptric system sold on the market, we may not even know the parameters because most of the suppliers do not provide them with the product. They usually provide a software which is able to convert the omnidirectional images to cylindrical and normal perspective images. But the provided conversion software performs only predefined tasks. For instance, an omnidirectional camera system produced for teleconferencing provides projections of certain regions in order to show the faces of the people sitting around a table. If you want to use the same system for tele-presence or virtual navigation then you need to know the system parameters. This fact increases the importance of the study on calibration of catadioptric systems.

The following paragraphs introduce the reader to the previous work on catadioptric camera calibration. A summary is given in Table 3.1 as well.

Geyer and Daniilidis [27] used user-supplied points along straight lines (or a dot calibration pattern) for catadioptric camera calibration. Their experiments are held with a paraboloidal-mirrored camera. The only requisite is the detection of two sets of parallel lines in one image. If the pixel aspect ratio is equal to 1, the reflections of these parallel lines on the omnidirectional image are circles. If not, elliptical arcs are obtained instead of circular arcs and the ratio of principal axes of the ellipse directly reveals the aspect ratio. The vanishing points of these circles are adequate to reveal image principal point and the parabola parameter (h).

Geyer and Daniilidis [28] improved their calibration method for paraboloidal mirrors, which uses straight lines in a way that detecting 3 straight lines in the image is adequate to apply the calibration algorithm. They propose a closed-form solution for parabola parameter (h), principal point and aspect ratio for skewless cameras and a polynomial root solution in the presence of skew.

Geyer and Daniilidis [29] stated that calibration of a catadioptric projection is possible with only two line images if the mirror is not paraboloidal (three line images are needed for paraboloidal case as stated in [28]). Foci of ellipses that line images form reveal the foci of the hyperbola. Techniques are not shown experimentally.

They also mentioned that every single-viewpoint catadioptric system is equivalent to central projection to a sphere followed by projection from a point on the sphere to the main projection surface. They added that the catadioptric projections have duals and the mapping between a projection and its dual returns the foci of a conic shape of line image. Details of these propositions can be found in [29].

On the other hand, Kang [30] proposed two techniques for paraboloidal mirrors requiring no calibration pattern. First one assumes the maximum vertical viewing angle of the mirror is known *a priori*. He matches the bounding circle of the ODI with the maximum vertical viewing angle. The output is the paraboloidal parameter; h. Second technique is based on the consistency of points tracked across an image sequence. The unknowns which are the paraboloidal parameter (h), aspect ratio, image skew and the location of the principal point are estimated by using the least-median error metric.

Orghidan et al. [31] presented a technique for calibration of a hyperboloidal mirror-camera system. In this technique, camera intrinsic parameters and mirror parameters are found by non-linear minimization based on Levenberg-Marquard and a set of known object points. This set of 3D coordinates of object points is obtained via a laser emitter and another hyperbolic mirror which reflects the laser rays to the view area of the hyperbolic mirror of the camera whose parameters are to be determined.

The proposed design for locating coordinates of the points in the scene is not easy to establish. Precise alignment of the system is required and locations of mirrors and parameters of the mirror of the laser projector are assumed to be known. Errors in positions will affect the accuracy of the determined object coordinates. The intrinsic parameters included in the minimization are aspect ratio and principal point. Aspect ratio may usually be determined from the specifications of the camera used and

principal point can also be determined from geometrical relationships of the reflected laser rays if laser projection system is established precise enough to be used for locating 3D coordinates.

Cauchois et al. [32] proposed a calibration technique for a system with a conical mirror. Proposed technique consists of two stages. First, using a 2D calibration pattern located on the top of the cone, intrinsic parameters (scale factors, camera constant and principal point location) of the camera and the distance from the cone to the camera coordinate system is estimated according to the position error values of reflected calibration points on the image. In the second stage, two perpendicular planes in the scene are used as calibration patterns. Intrinsic parameters estimated in the first stage are used as initial values in the second stage. Extrinsic parameters (parameter of the cone, rotation matrix, translation vector) are obtained and intrinsic parameter estimates are improved.

Derrien and Konolige [33] developed a method to approximate a single-viewpoint for spherical mirrors. This method assigns a single viewpoint and performs the conversion as if all the incoming rays are targeting this point. Inevitably, resultant images are distorted but the authors show that this distortion is about 1-2%.

Fiala and Basu [34], performs calibration and feature extraction for a stereo non single-viewpoint catadioptric sensor. Mirror of the proposed imaging system comprises two lobes in order to provide a stereo panoramic view. The main axes of both mirror lobes are collinear with the camera's optical axis. Thus, using the common points in the two (one inside the other) omnidirectional images of the same scene, they obtain the focal length of the camera, distances to the two mirror centers and the radii of the mirrors. Since, the mirror type used is spherical, there is no additional mirror parameter. Usage of a non single-viewpoint mirror type causes not being able to determine a projection center, which makes the task of locating the scene points harder. However, authors have overcome this handicap by using only horizontal and vertical lines in the scene while comparing two omnidirectional images. Horizontal line segments are extracted using the Panoramic Hough Transform.

Goncalves and Araujo [35] studied the recovery of a mirror shape described by the conic expression: $x_1^2+x_2^2+Ax_3^2+Bx_3=C$, when the intrinsic parameters of the camera and 3D coordinates of 3 world points are given. Also a good initial value of the unknown parameters is required. Solving the differential equation, parameters in the mirror surface equation are estimated with errors below 2%.

Fabrizio et al [36] worked on a calibration technique, which can be generalized to all catadioptric mirrors. It is assumed that surface parameters of the mirror are known. Given these parameters, the proposed method does not require any calibration pattern and it uses only one image. First of all, outer and inner boundaries of the omnidirectional image are determined using an edge detection technique. It is mentioned that these boundaries may not be circles if camera and mirror are not perfectly aligned. Using relationship between image pixels and the coordinates of the reflectance points on the mirror surface, camera focal point is determined. Also, CCD camera orientation (exterior calibration) parameters are estimated by minimizing the least square error.

Epipolar geometry of stereo catadioptric omnidirectional systems is extensively described by Svoboda et al [9].

Table 3.1: The previous work on catadioptric camera calibration

Paper	**Mirror type**	**Required information**	**Technique used**	**Output**
Geyer and Dani.[27]	Parab.	2 sets of parallel lines in the scene	Geometric relationships between the images of given lines	aspect ratio, principal point, parabola parameter
Geyer and Dani.[28]	Parab.	3 straight lines in the scene	Geometric relationships (Technique described in [27], shown experimentally)	parabola parameter, principal point and aspect ratio for skewless cameras
Geyer and Dani.[29]	All	2 lines in the scene (except paraboloidal)	Geometric relationships (Not shown experimentally)	principal point, mirror parameters
Kang [30] - 1st	Parab.	Bounding circle of ODI, vertical viewing angle	Already known relations	parabola parameter
Kang [30] - 2nd	Parab.	Consistency of points tracked across an image sequence	Least-median error metric with the error between image pairs	parabola parameter, aspect ratio, image skew, principal point

Table 3.1: The previous work on catadioptric camera calibration (cont.)

Paper	Mirror type	Required information	Technique used	Output
Orghidan et al. [31]	Hyperb.	Additional laser camera-hyperb. mirror system	Object coordinates are determined by locating the laser ray points in the ODI. Non-linear minimization based on Levenberg-Marquard for conjugate pairs	scale factors, principal point, translation vector between camera and mirror system, hyperbola parameters
Cauchois et al. [32]	Conical	A 2D and a 3D calibration pattern for two stages	To estimate intrinsic parameters, mean value error and 2D calib. pattern. To find extrinsic parameters and to finalize intrinsic parameters 3D calib. pattern	scale factor, camera constant, principal point, translation vector, rotation matrix, mirror parameters
Fiala and Basu [34]	Spher. with two lobes	Stereo panoramic image	Locating the image of points in the both omnidirectional images	camera constant, radii of the mirrors, distance between camera and mirror
Goncalv. and Araujo [35]	All	Intrinsic parameters, coordinates of 3 world points	A good initial value is determined, DAE is solved	parameters in the general equation of conic reflectors: $x_1^2+x_2^2+Ax_3^2+Bx_3=C$
Fabrizio et al. [36]	All	Parameters of the mirror, a set of conjuate pairs	Using mirror parameters camera constant is determined. Extrinsic parameters are obtained by error minimization between conjugate pairs.	camera constant, extrinsic parameters

3.3 Inspecting the Reflections of Different Mirror Types

3.3.1 Reflection Simulations

We performed MATLAB simulations to observe the effects of mirror shapes and their parameters in the omnidirectional images. We simulated a catadioptric omnidirectional camera placed in the middle of the ceiling of a virtual room with certain objects located inside (Figure 3.1). We generated the reflection of paraboloidal, hyperboloidal and spherical mirror surfaces and compared the omnidirectional images gathered through these mirror types (Figure 3.2).

In Figure 3.2, mirror parameters are selected so that the outer circles of the omnidirectional images (image of the ceiling boundary) will coincide with each other. This approach revealed the effects of curvature differences between mirror types.

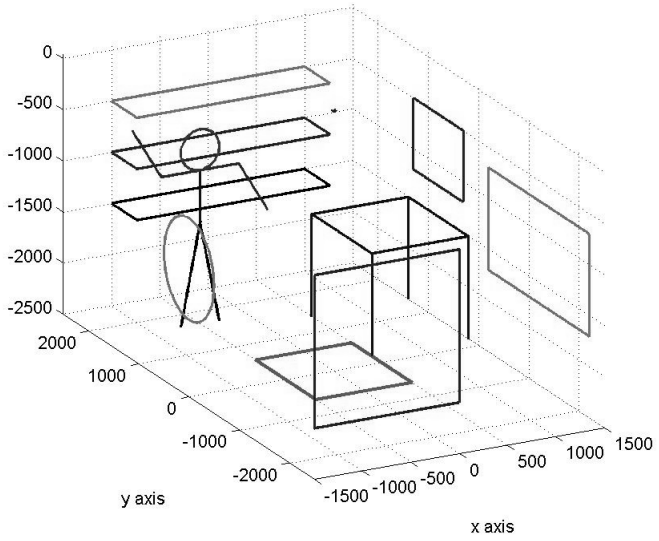

Figure 3.1: Virtual room created by MATLAB

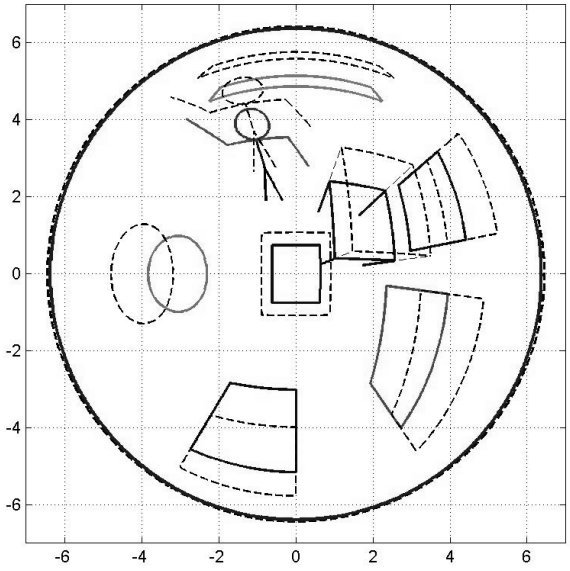

Figure 3.2: Reflection simulations of paraboloidal, hyperboloidal and spherical mirror surfaces (solid line: paraboloidal and hyperboloidal, dashed line: spherical)

Reflections of the hyperboloidal and paraboloidal mirrors are not identical but they can be made visually very close if their parameters are selected appropriately. Reflection of the spherical mirror however, is significantly different; the region under the mirror (when mirror surface looks down) covers a wider area in the omnidirectional image (ODI). The region in the three dimensional space at the same elevation with the mirror is represented by a narrower area in the ODI.

To visualize the small difference between the reflections of paraboloidal and hyperboloidal mirror in Figure 3.2, two dimensional reflection simulations are generated (Figures 3.3 and 3.4) with the same parameters.

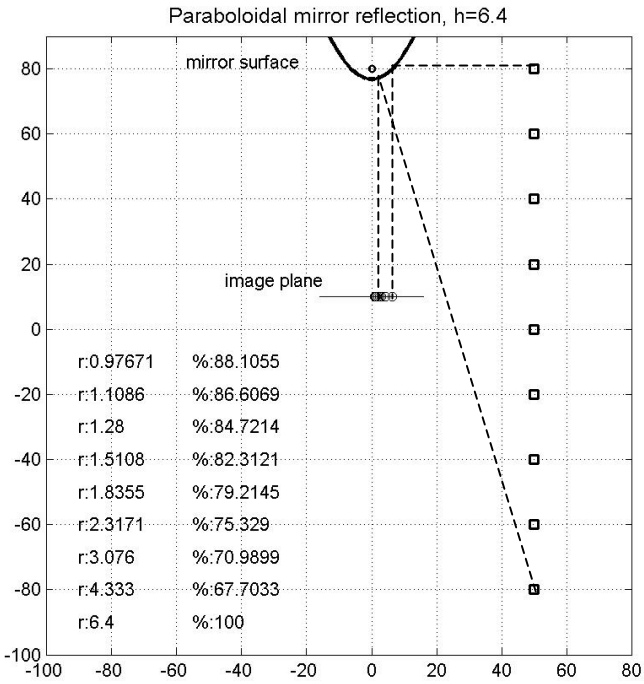

Figure 3.3: A reflection example by paraboloidal mirror (to be compared with 3.4)

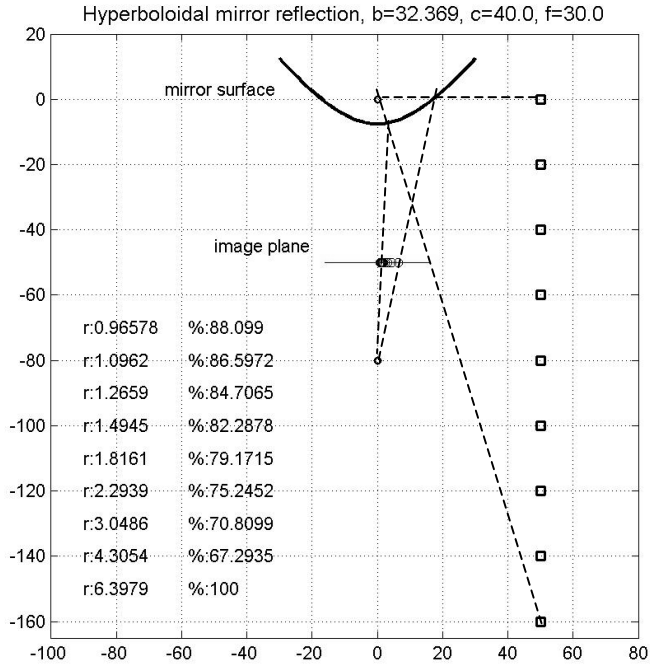

Figure 3.4: A reflection example by hyperboloidal mirror (to be compared with 3.3)

World points are equi-distant and are located at a certain distance from the optical axis of the catadioptric system and corresponding image points are located at (r) from the optical axis. Images of same size (diameter) are obtained by both mirror types by changing their parameters. Although the size is same, the locations of internal points differ. The distance ratio of each point with respect to its neighbor (%) is obtained as well. These ratios are not same due to the different curvatures of mirrors, concluding that these two reflections do not resemble each other by a scale factor.

To observe the effect of mirror parameters on the omnidirectional images, two dimensional reflection simulations are generated for different paraboloidal, hyperboloidal and spherical mirror profiles. Figures 3.5-3.7 represent the locations of image points of objects that are equi-distant in the scene.

In Figure 3.5, two paraboloidal mirror reflections with different paraboloid parameters (h) are compared. Although the locations of pixels in the ODI are different, the ratio of these locations in the two reflections remains same for every point.

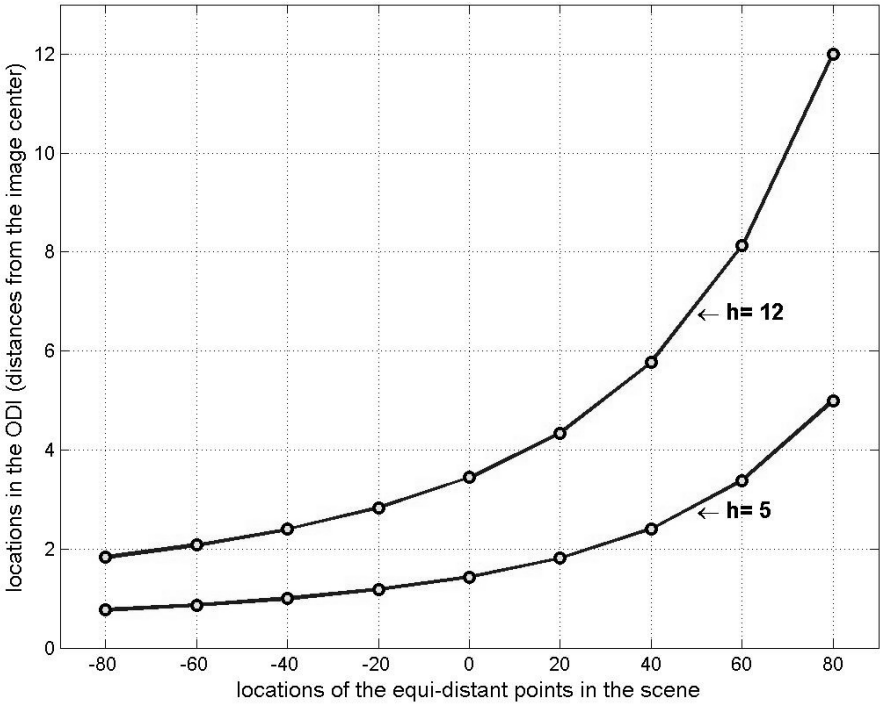

Figure 3.5: Effect of paraboloidal mirror parameter (h) on the ODI

In other words, for the two ODIs obtained through two paraboloidal mirrors having h=5 and h=12, location of a pixel in one image is 5/12 of the location in the other image. The location of pixels and the mentioned ratio is given in Table 3.2. Thus, we conclude that the two ODIs resemble each other by a scale factor.

Table 3.2: The ratio of pixel locations in the two ODIs

Locations in the scene	-80	-60	-40	-20	0	20	40	60	80
Pixel location in the ODI, h=5 units	0.763	0.866	1	1.18	1.434	1.81	2.403	3.385	5
Pixel location in the ODI, h=12 units	1.831	2.079	2.4	2.833	3.442	4.345	5.767	8.124	12
Ratio (for h=5/h=12)	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417	0.417

In Figure 3.6, reflections of hyperboloidal mirrors with different parameters are compared. Effect of mirror parameters (b and c) and camera focal distance (f) can be observed from this figure. Trying with different parameter sets we saw that the same reflection can be obtained by more than one parameter sets. These reflections do not resemble each other by a scale factor since the mirror curvature changes. Only camera focal distance (f) has a linear relationship, i.e. if the mirror parameters remain same, change in the f value creates a scale difference.

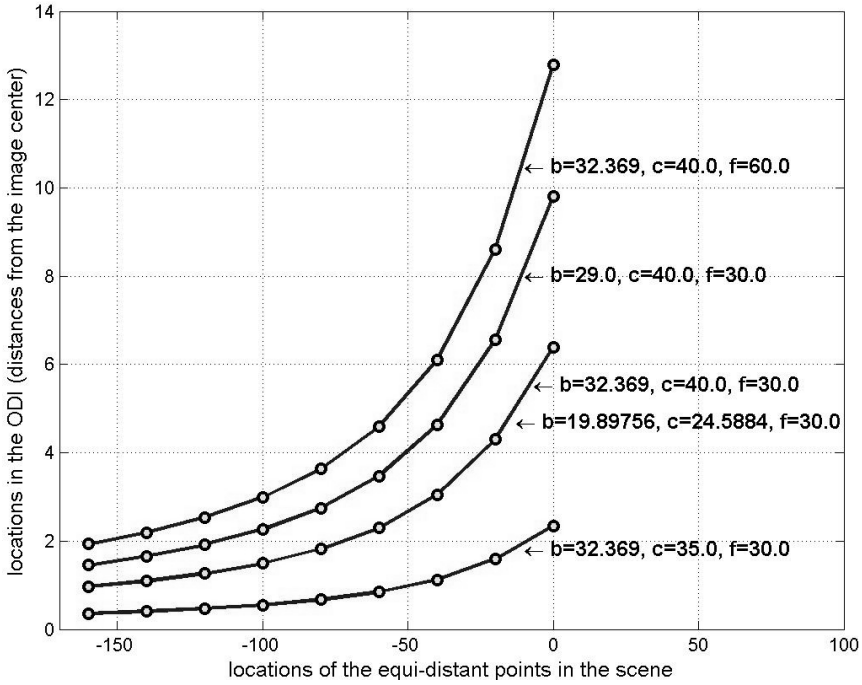

Figure 3.6: Effect of hyperboloidal mirror parameters (b,c) and focal distance (f) on the ODI

In Figure 3.7, reflections of two spherical mirrors with different radii (g) are compared. The distance between the camera and the mirror (d) and the focal distance (f) are included in the comparison as well. The mirror profile has one parameter (radius), however the distance between the camera and the mirror affects the resultant image as well, because the light rays are not reflected orthogonally as in the paraboloidal case. Therefore, these reflections do not resemble each other by a scale factor. Only f has a linear relationship as in the hyperboloidal case, i.e. if other parameters remain same, change in the f value creates a scale difference.

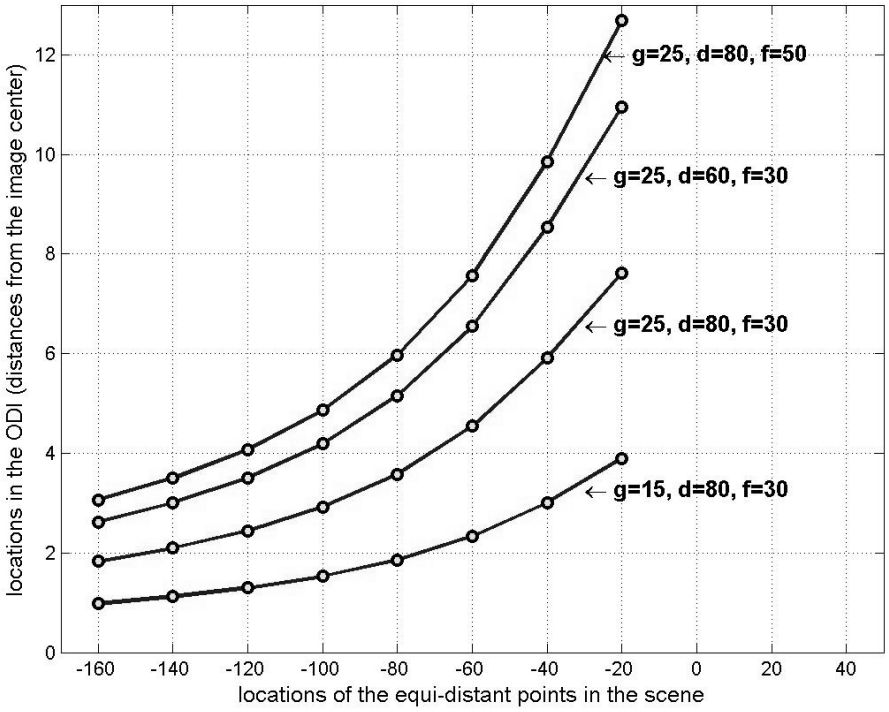

Figure 3.7: Effect of spherical mirror radius (g), camera-mirror distance (d) and camera focal distance (f) on the ODI

3.3.2 Conclusions

- The curvatures of paraboloidal and hyperboloidal mirrors are different from each other, thus the reflections are not the same. However, when the sizes of the resultant omnidirectional images are adjusted so that they are equal and the mirror parameters are selected appropriately, the difference between the images are not large enough to be noticed visually.

- Paraboloidal mirror's surface equation has only one parameter. Because the light rays that form the image are reflected orthogonally from the mirror and are collected by a telecentric lens, the distance between the mirror and the camera does not affect the omnidirectional image. These two facts conclude that, two ODIs taken with different size of paraboloidal mirrors resemble each other by a scale factor.

- Hyperboloidal mirror's surface equation has two parameters. Thus, unlike the paraboloidal mirror both scale and curvature may change. By employing different parameter sets we saw that the same reflection can be obtained by more than one parameter set. The relation between these two surface parameters is shown experimentally in section 3.4.1 and mathematically in section 3.4.3.

- In systems with hyperboloidal mirrors, the distance between the camera and the mirror is important. However, because it is related to the one of the surface parameters, this distance value is not taken into account as a third parameter of the mirror-lens system.

- Spherical mirror's surface equation has one parameter (the radius). However, the reflections do not resemble each other by a scale factor as in the paraboloidal mirror, since the distance between the camera and the mirror affects the resultant image and has to be taken into account as a second parameter.

- We did not study the conical mirror case, but we expect it to be similar to the spherical mirror case. This is because the conical mirror also has one surface parameter (the slope) and the distance between the camera and the mirror affects the resultant image.

3.4 Determining the Parameters of Hyperboloidal Mirror from ODIs

In order to determine the parameters of the hyperboloidal mirror, the real world coordinates of the objects and the pixels in the omnidirectional image are matched. In this way, mirror parameters (b and c) and focal distance (f) in the equation below

are sought for. Equation 3.1 is a modified version of Equation 2.5 in a way that X,Y,Z coordinates of world points are used instead of projection angles γ_m and γ_c :

$$x = \frac{f \cdot (b^2 - c^2) \cdot X}{(b^2 + c^2) \cdot Z - 2bc\sqrt{X^2 + Y^2 + Z^2}}, \quad y = \frac{f \cdot (b^2 - c^2) \cdot Y}{(b^2 + c^2) \cdot Z - 2bc\sqrt{X^2 + Y^2 + Z^2}} \tag{3.1}$$

where, *x* and *y* are the x axis coordinate and y axis coordinate of the point in the image plane respectively.

3.4.1 Error Calculation for Different Mirror Parameter Sets

Assuming the focal distance of the camera and world coordinates are known, we should be able to match the points in the image and real world coordinates (conjugate pairs). For a given pair set of x values and world coordinates, total squared error for n pairs will be:

$$\sum_{i=1}^{n} E_i = \sum_{i=1}^{n} \left(x_i - \frac{f \cdot (b^2 - c^2) \cdot X_i}{Z_i \cdot (b^2 + c^2) - 2bc\sqrt{X_i^2 + Y_i^2 + Z_i^2}} \right)^2 \tag{3.2}$$

Error relationship for y-axis can be written in the same way between *y* values and world coordinates. While calculating the total error, effective viewpoint of the mirror is taken as the origin of the coordinate system.

The virtual room was simulated by a hyperboloidal mirror with parameters: b=32.369 units and c=40 units whose focal point is located at the origin. Camera focal distance (f) is taken as 30 units. A square in this virtual room (*e.g.* a painting on the wall) is created with dimensions 800x800 units. If the unit is taken as one millimeter, the created square will be 80cm.x80cm in the real world. 160 points are selected from the square created, 40 from each edge, so that the distances between them will be equal. The locations of the image points of these scene points in the omnidirectional image are calculated using the parameters given above. As a result, matching the 3D scene coordinates of the points and their 2D image coordinates we obtained 160 conjugate pairs. Using this conjugate pair set, we have calculated the total squared error value (Equation 3.2) in terms of the mirror parameters (b,c).

Figure 3.8 is the total squared error value graph obtained for a range of hyperboloidal mirror parameters (b & c). This graph has a valley shape enlarging towards the origin. Original (b,c) pair is represented by a star. The white line in the graph represents the solution set, which means that more than one pair of (b,c) makes the error minimum as will be discussed in Section 3.4.2. Figure 3.9 is a 3D view of this error surface for a smaller (b,c) range.

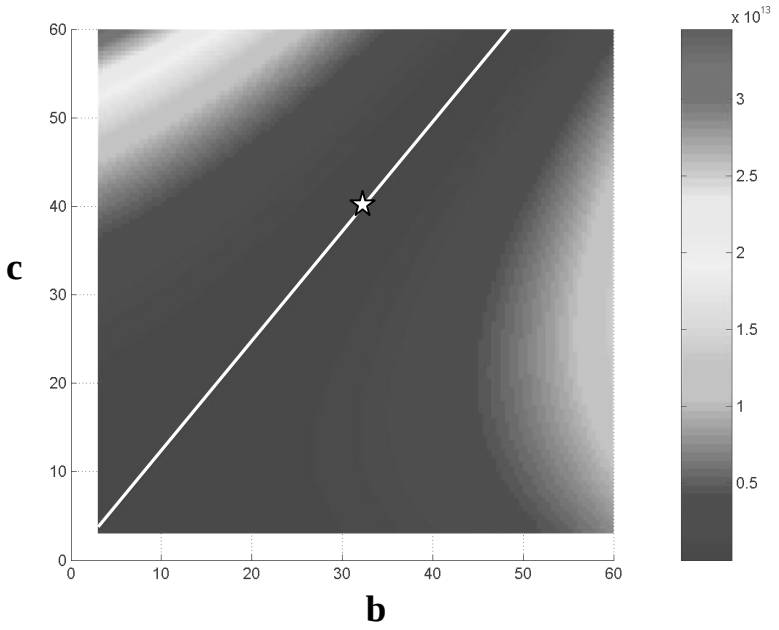

Figure 3.8: Total error values for b[3,60]-c[3,60] range,
star: the original (b,c) pair (32.369,40), line: the solution set

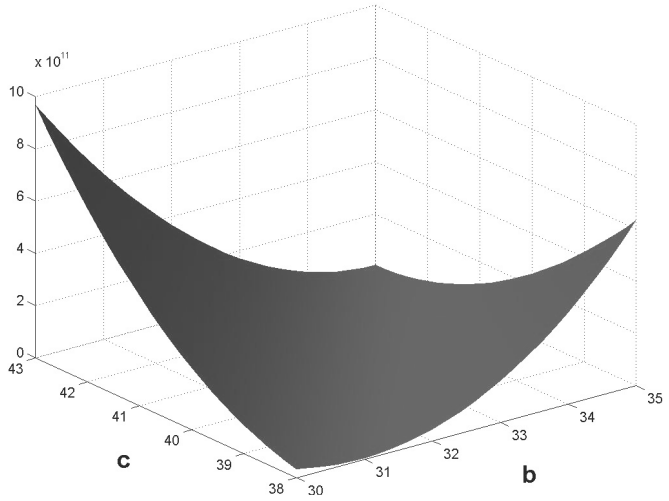

Figure 3.9: 3D view of Figure 3.8 for b=[30,35] - c=[38,43] range

3.4.2 Optimization to Determine the Mirror Parameters (b,c)

An optimization algorithm is used in order to find a proper (b,c) pair. To obtain b and c, Equation 3.2 is minimized by the 'steepest descent' method as explained in [59]. At each step, (b,c) pair is increased in the direction of gradient. Error value is checked and scale factor is employed to regulate the step size.

The optimization algorithm mentioned above is run with the conjugate pair set used in Section 3.4.1, which is composed of 160 points calculated while generating the ODI. Optimization steps are indicated in Figure 3.10. Each different initial guess of the optimization algorithm gave us a different (b,c) pair and in all those pairs the ratio b/c is the same (slope of the white line). We generated 2D reflection simulations using the estimated parameters to see whether they give the same reflection with the original parameter set. Figure 3.11 is the reflection generated with the parameters found by the optimization shown in Figure 3.10 and gives exactly the same result with Figure 3.4 which was generated by the original (b,c) pair.

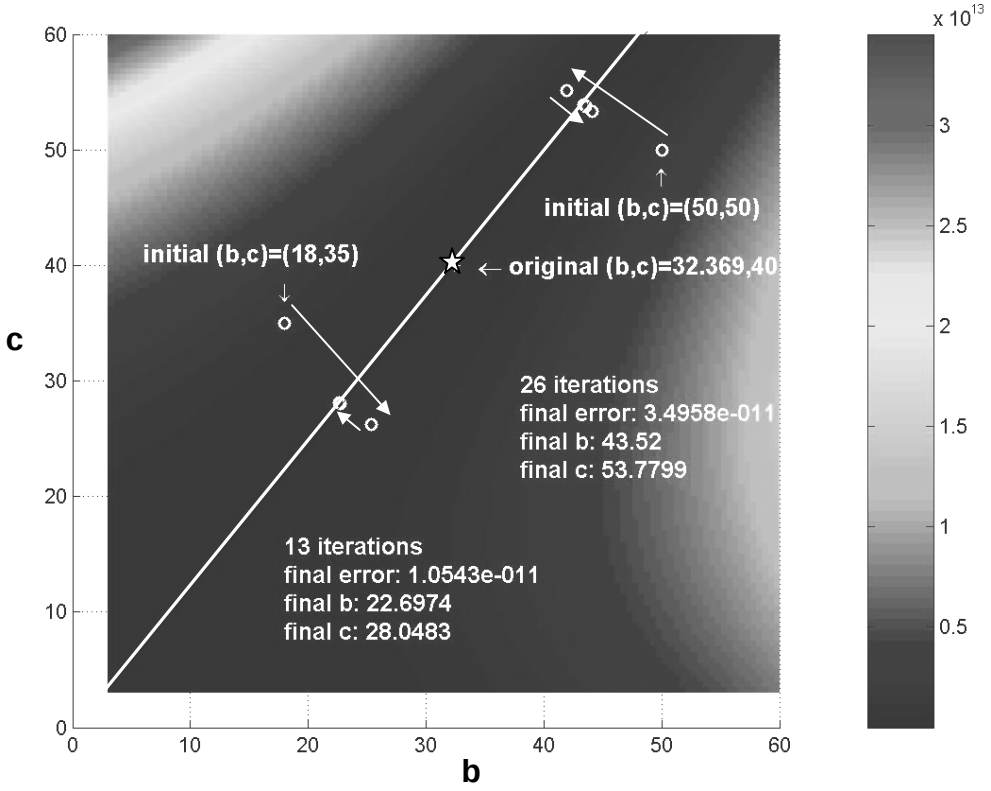

Figure 3.10: Visualization of the optimization, star: the original (b,c) pair (32.369,40), circles: the (b,c) pairs given initially and found at each iteration, line: the solution set

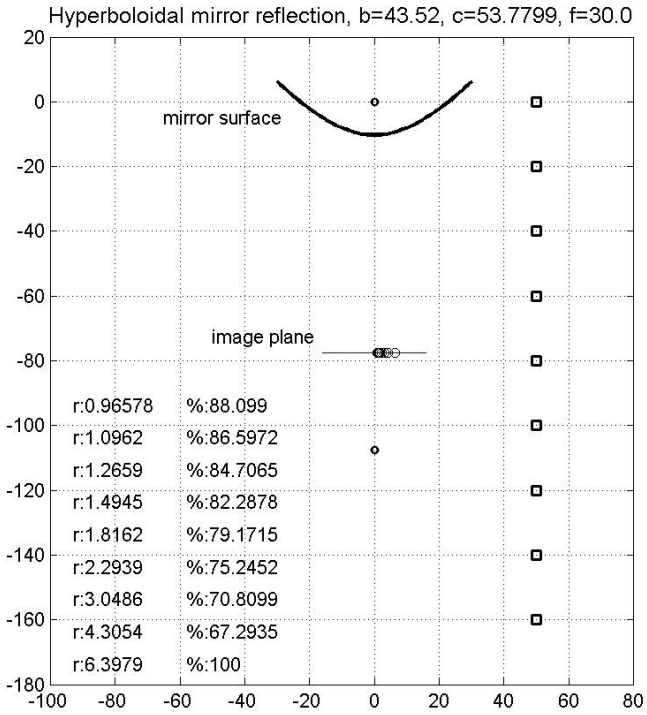

Figure 3.11: Hyperboloidal mirror reflection generated with the parameters found by optimization, to be compared with original reflection (Figure 3.4)

3.4.3 Linear Relationship between Mirror Parameters

From Equation 3.1, it is inferred that a linear relationship exists between hyperboloidal mirror parameters b and c. In fact,

$$b = c \cdot \frac{-x \cdot \sqrt{X^2 + Y^2 + Z^2} \pm \sqrt{x^2 \cdot (X^2 + Y^2 + Z^2) + f^2 \cdot X^2 - Z^2 \cdot x^2}}{f \cdot X - Z \cdot x} \tag{3.3}$$

Derivation of Equation 3.3 from Equation 3.1 is given in Appendix A. When one of the two parameters is given, the other can be determined using Equation 3.3. This explains the line in Figure 3.10.

3.4.4 Using Mirror Parameter Ratio (b/c) as One Parameter

We included the camera focal distance, f, in the unknown parameter set in addition to b and c. However, we realized that the optimization can not be efficiently performed if we want to determine the parameters more precisely. Since there is an infinite number of optimum (b,c) pairs, the estimated values of b and c tend to decrease to make the squared error equation smaller. This also prevents the progress of f value,

because error decreases as b and c decreases regardless of the value of f. Therefore, we modified our total error function so that it will include the relationship between b and c (shown in Section 3.4.2) and take the b/c ratio as one parameter, k:

$$\sum_{i=1}^{n} E_i = \sum_{i=1}^{n} \left(x_i - \frac{f \cdot \left(k^2 - 1\right) \cdot X_i}{Z_i \cdot \left(k^2 + 1\right) - 2k \cdot \sqrt{X_i^{\,2} + Y_i^{\,2} + Z_i^{\,2}}} \right)^2 \tag{3.4}$$

Figure 3.12 is the total error value graph obtained for a range of new unknown parameters (k & f). This graph has also a valley shape, but total error is not equal to zero at the origin.

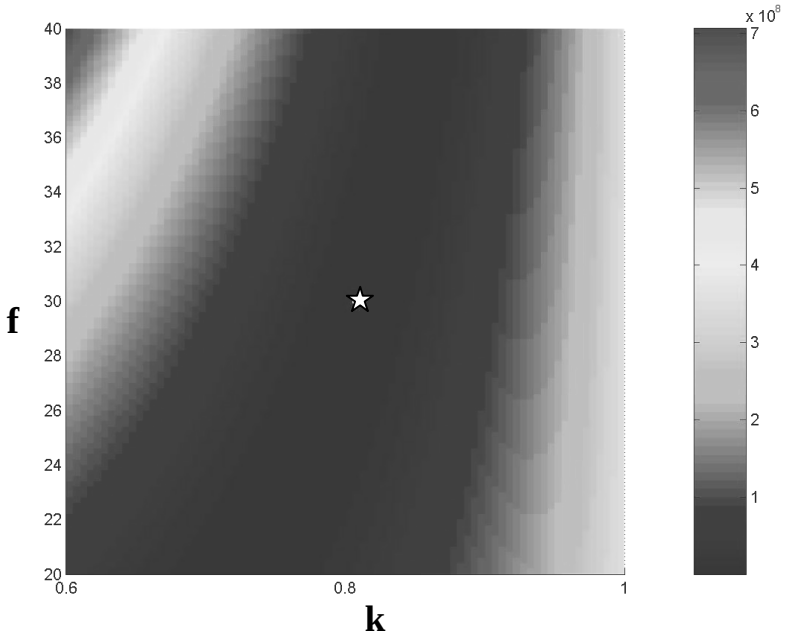

Figure 3.12: Total error values for k[0.6,1]-f[20,40] range,
star: the original (k,f) values (0.809225,30)

For the same virtual room, four different sets of point conjugate pairs are created. Estimation of b and c are repeated for the cases below:

A) 100 points are randomly selected so that their 3D coordinates will be inside the virtual room. The 2D coordinates on the ODI corresponding to the selected 3D points are computed exactly.

B) 100 points are randomly selected so that their 3D coordinates will be inside the virtual room, but this time the corresponding 2D coordinates on the ODI are perturbed to mimic measurement errors. The perturbations are modeled as

samples from a Gaussian distribution with mean zero and standard deviation 0.0125 along x and y directions. The value 0.0125 in the scale of Figure 3.2 corresponds to 0.5 pixel standard deviation in a 512x512 pixel image. The explained procedure of creating 100 conjugate pairs is repeated 100 times and average values for k and f are recorded.

C) The procedure in Case A is repeated for 20 selected points.

D) The procedure in Case B is repeated for 20 selected points. Again, 100 trials are performed and average values for k and f are recorded.

This time, optimization is performed using Gauss-Newton least square minimization algorithm as explained in [59]. It is more powerful than the steepest descent algorithm used previously. For each iteration, error value is checked and scale factor is employed to regulate the step size. Norm of the current step is selected as the stopping condition. Calculation functions of total squared error, residual vector and jacobian matrix and Gauss-Newton optimization subroutine written in MATLAB are given in Appendix B. Optimization is performed for each of the above mentioned four cases with the same initial values. Results are given in Table 3.3. In this table, k_o and f_o indicate the original values of k and f, $\bar{k}$ and $\bar{f}$ indicate the average values of k and f estimated over 100 trials.

Table 3.3: Results of the optimization for k (b/c ratio) and f (focal distance)

	Original	**Initial**	**Case A** (100 points, exact)	**Case B** (100 points, noisy)	**Case C** (20 points, exact)	**Case D** (20 points, noisy)
$\bar{k}$	0.809225	0.72	0.809225	0.809438	0.809225	0.811254
$\bar{f}$	30	37	30.0000	30.1461	30.0000	30.9551
RMSe in x	-	5.483	0	0.0072	0	0.0153
RMSe in y	-	5.483	0	0.0072	0	0.0153
MAD for k	-	0.089225	0	0.00669	0	0.02087
MAD for f	-	7.0	0	1.1829	0	4.3598
# iterations	-	-	276	181.5	286	206.5
Error @ $\bar{k}$, $\bar{f}$	-	-	4.3E-20	1.0E-6	3.3E-20	7.5E-6
Error @ k_o , f_o	-	-	6.8E-26	1.3E-6	2.1E-26	8.6E-3

Mean absolute deviations of estimated k and f values compared to original values are given in the table as MAD for k and f. To compare the success of the optimization for each case we simulated the ODIs with the estimated parameters. In this simulation, image points of 4 checkerboards (100 points totally) on each wall of the virtual room are compared with the reflection of the original parameters. The 'root mean square error' (RMSe) between the locations of the simulated image points and the original image point locations are included in the table. Also, average number of iterations and total square errors with estimated and original k,f values are given.

As expected, the optimization with calculated points gave perfect results. Increasing the number of points used has a positive impact when the perturbed feature points are used as in Cases B and D. The reflections with results of Case A and Case C are the same with the original pair's reflection, whereas in Case D the RMSe is 0.0153 which corresponds to 0.60 pixel for a 512x512 pixel image and in Case B the RMSe is 0.0072 which corresponds to 0.28 pixel for a 512x512 pixel image. However, we see that these errors are not distinguishable by eye when we look at the omnidirectional images.

3.4.5 Determining Parameters Using Calibration Pattern

We tried to determine the parameters when the world coordinates of the objects are not known but the relationship between the points in a calibration pattern are known. For the calibration pattern, we used a square meshed into slices with equal thickness as shown in Figure 3.13. The square having dimensions of 200x200 cm. and divided into 64 equal squares is located at one of the walls of the virtual room (Figure 3.14).

In the previous optimization performed with known object 3D coordinates, there were two unknowns (k, f). In the optimization using calibration pattern we have three additional unknowns: the 3D coordinates of the calibration pattern's reference corner (X_c, Y_c and Z_c). Since the relative distances between the corner points are known, all 3D coordinates are written in terms of X_c, Y_c and Z_c (Figure 3.13).

Similar to the optimization performed with known 3D coordinates, four different sets of point conjugate pairs are selected to compare the optimization performances:

A) 81 corner points in Figure 3.13 are selected and the 2D coordinates on the ODI corresponding to the selected 3D corner points are computed exactly.

B) 81 corner points in Figure 3.13 are selected but this time the corresponding 2D coordinates on the ODI are perturbed to mimic the measurement errors. The perturbations are modeled as samples from a Gaussian distribution with mean zero and standard deviation 0.0125 along x and y directions. The value 0.0125 in the scale of Figure 3.14 corresponds to 0.5 pixel standard deviation a 512x512 pixel image.

C) The procedure in Case A is repeated for the 25 circled corners in Figure 3.13.

D) The procedure in Case B is repeated for the 25 circled corners in Figure 3.13.

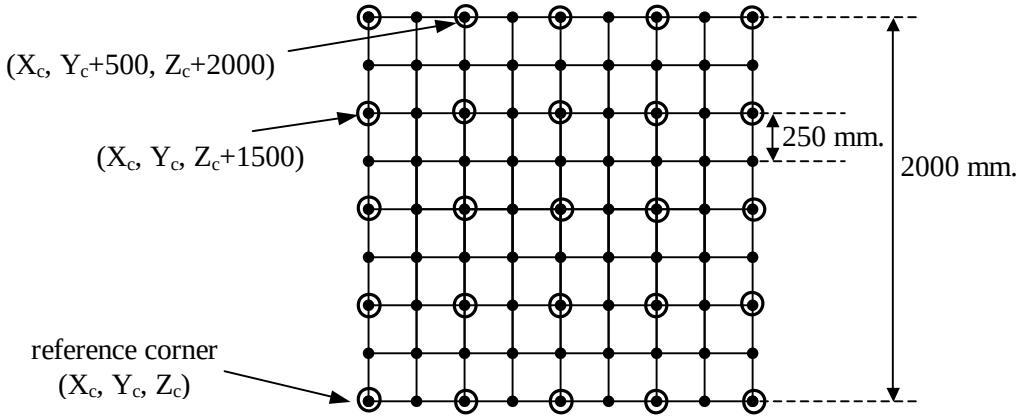

Figure 3.13: Calibration pattern used for parameter optimization:
All 81 corner points are used for optimization Cases A and B,
25 circled corner points are used for optimization Cases C and D.

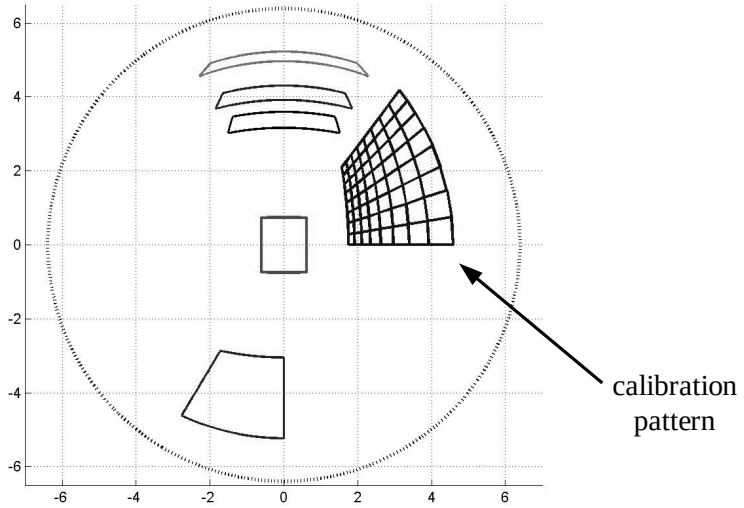

Figure 3.14: ODI obtained from the virtual room including calibration pattern

Table 3.4: Results of the optimization with the calibration pattern

	Original	**Initial**	**Case A** (81 points, exact)	**Case B** (81 points, noisy)	**Case C** (25 points, exact)	**Case D** (25 points, noisy)
$\bar{k}$	0.809225	0.78	0.809225	0.796219	0.809225	0.784088
$\bar{f}$	30	28	30.0000	28.6300	30.0000	26.5850
$\bar{X}$	1500	1480	1500.00	1501.34	1500.00	1503.01
$\bar{Y}$	0	-30	0.00	-0.39	0.00	0.85
$\bar{Z}$	-2500	-2520	-2500.00	-2504.80	-2500.00	-2502.95
RMSe in x	-	0.0600	0	0.0098	0	0.0101
RMSe in y	-	0.0600	0	0.0098	0	0.0101
# iterations	-	-	6611	2593	6613	1848
Error @ $\bar{k}$, $\bar{f}$	-	-	1.0E-27	7.9E-7	1.0E-27	4.4E-6
Error @ k_o , f_o			1.0E-32	2.0E-6	1.0E-32	9.7E-3

Results are shown in Table 3.4. The 'root mean square error' (RMSe) value for Case B (81 corners, noised) is 0.0098 which corresponds to 0.39 pixel error in a 512x512 pixel image. In Case D (25 corners, noised) RMSe is 0.0101 corresponding to 0.40 pixel error in a 512x512 pixel image. According to these values it can be said that there is not much efficiency difference between 81 noisy points and 25 noisy points, however when the differences between the original parameters (k_o, f_o) and the estimated ones ($\bar{k}$, $\bar{f}$) are considered, we see that estimated values in Case B (81 noisy points) are closer to the original values than Case D (25 noisy points).

We observed that inital values affect the performance of the optimization when noisy coordinates are used. For the same data, if we start from another point different results may be obtained. Thus, we conclude that optimization should also be tested with different calibration patterns having different size, shape and location.

3.4.6 Conclusions

We found out that for every 'b', there is such a 'c' that gives the same reflection with another (b,c) pair. As a result, if we are given an omnidirectional image taken with a hyperboloidal mirror and if we know the real world locations of the objects seen in

the image, we are able to find the b/c ratio of the hyperboloidal mirror with a performance depending on the method of measurement of the locations of points in the omnidirectional image and the number of points used for the optimization.

Parameter extraction method that is used can be evaluated as an alternative to the work of Orghidan et al [31]. Their proposed design for locating coordinates of the points in the scene depends on a laser emitter and requires precise alignment of components of the imaging system.

We also tried our parameter extraction method with a calibration pattern. The 3D coordinates of a corner of this calibration pattern is estimated. Since, we assumed that the pattern is located on the wall in our virtual room, its alignment is known and coordinates of all points in the pattern can be written in terms of the corner point. An enhancement to this optimization may be using a calibration pattern of which the alignment is not known.

We also observed that the size, shape and location of the calibration pattern may affect the results. A calibration pattern especially for omnidirectional cameras can be designed.

The work presented in this chapter is performed with simulated reflections from a hyperboloidal mirror. The developed method can be tested with a real omnidirectional system in the future. In a real system, additional parameters such as sensor aspect ratio, image skew and lens distortions should be determined.

CHAPTER 4

Resolution of Panoramic Images

In this chapter, resolutional properties of the panoramic images (panoramas) that are generated from the omnidirectional images are examined. In Section 4.1, the blurring problem encountered in panoramic projections is defined and the effect of hyperboloidal, paraboloidal and spherical mirror profiles on this problem is analyzed. In Section 4.2, different projection surface alternatives for the panoramic projections are evaluated regarding their advantages and disadvantages for the problem defined in Section 4.1.

4.1 Resolution Analysis

4.1.1 Previous Work on Mirror Analysis

An expression for the resolution of the single-viewpoint catadioptric sensors is obtained by Baker and Nayar [10]. This expression is used to compare the resolutional properties of a mirror type with respect to the planar mirror. It is showed that resolution decreases as we move from the vertex to the edge of the mirror. Also, 'defocus blurring function' for a lens and mirror combination is derived and effects of ellipsoidal, hyperboloidal and paraboloidal mirrors are analyzed.

Nayar [22] found out that the resolution from the vertex (the bottom of the mirror) to the fringe (point having the same elevation with the focal point) of a paraboloidal mirror increases by a factor of 4. He advises using non-uniform resolution sensors to compensate this effect or to use larger detector array.

4.1.2 The Resolution Problem

In the panoramic images converted from the omnidirectional images (ODI), we observe lower image quality at the bottom or top of the panoramas depending on the location of the camera. This is because the number of pixels in the ODI feeding panoramic image is less as we move to the inner parts of ODI. Thus, the bottom (or top) of the panoramic image has lower sampling frequency and image quality decreases resulting in blurry regions.

Figure 4.1: An ODI sample [22]

To visualize the problem, pixels in the ODI (Figure 4.1) are placed at the panoramic image (Figure 4.2) by forward mapping. The white pixels in Figure 4.2 do not have a match in the ODI. In normal panoramic projection generation (backward mapping), these white pixels will be filled by interpolation of the neighboring pixels.

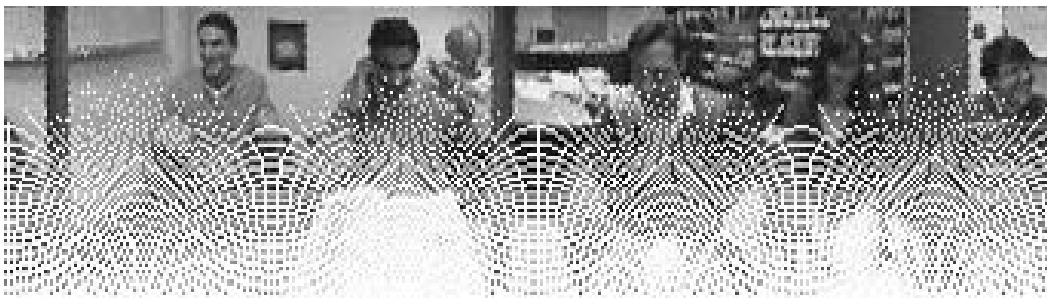

Figure 4.2: Panorama generated from Figure 4.1 by forward mapping

4.1.3 Analysis of Data Distribution during Panorama Generation

4.1.3.1 Spherical Mirrors

In Figure 4.3, an omnidirectional photo obtained through a spherical mirror is given. The panorama generated from this ODI is shown in Figure 4.4, which is converted by forward mapping in order to visualize the missing information. The pixels at the outer boundary of the ODI are assumed to be images of objects which are at the same horizontal level with center of the mirror's sphere.

As indicated in the previous section, we observe missing information at the bottom of the panorama due to the decrease in size in the ODI towards center.

Moreover, the upper parts of the panorama also have missing information but this time the reason is not the smaller area in the ODI but the increasing curvature of the spherical mirror towards the peripheral section.

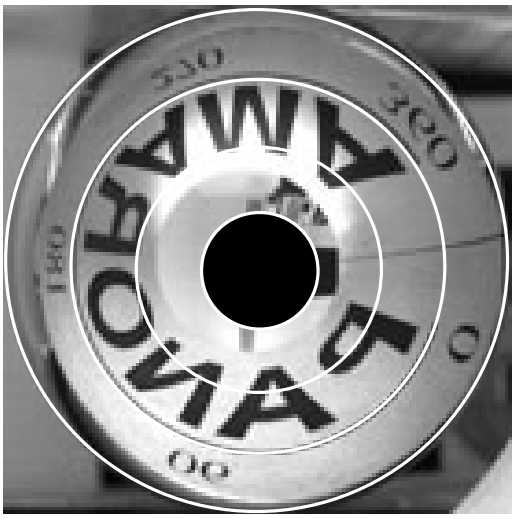

Figure 4.3: An ODI obtained through spherical mirror (Courtesy of Czech Technical Univ.)

Figure 4.4: Panorama generated from Figure 4.3 by forward mapping

The ODI in Figure 4.3 is divided into three concentric circles with equal thickness and the white lines in Figure 4.4 represent the borders of these circles. MATLAB simulations are performed to observe the scattering ratios of the peripheral and central sections of the spherical mirror. In Figure 4.5, three sections in different gray tones from top to bottom represent the outer, middle and inner one-third regions in the ODI respectively. The result of a 2D analysis concerning spherical mirror reflection is shown at Figure 4.6. The small squares in the image plane represent the pixels in the ODI along the radius and they are selected to be equidistant from each other. Elevations of the corresponding points in the panoramic image are represented as circles. We observe that as we move up on the spherical mirror (go away from the center in the ODI), representation of the 3D scene gets better. However, when we reach the elevation of the mirror the angles between the light rays increase. This

results in lower resolution. Thus, we can conclude that spherical mirror is not suitable for objects located higher than itself.

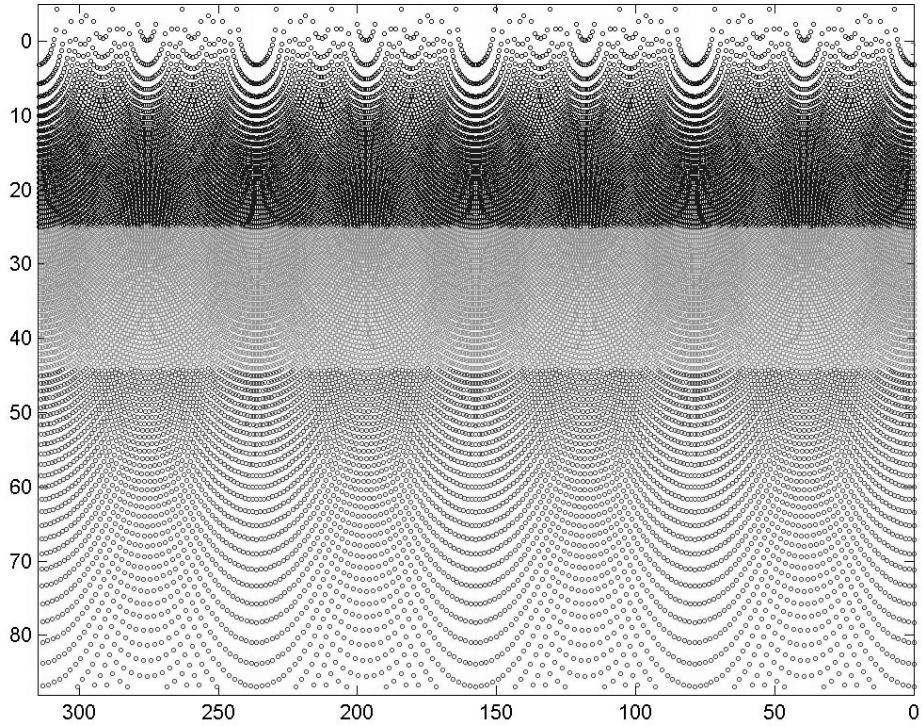

Figure 4.5: MATLAB simulation shows the scattering ratios of the peripheral (top) and central (bottom) sections of the spherical mirror

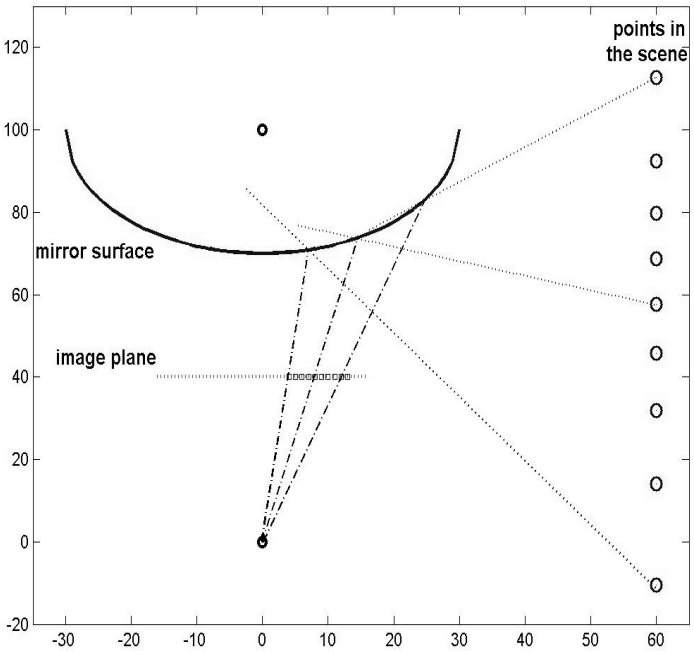

Figure 4.6: Spherical mirror reflection simulation

4.1.3.2 Paraboloidal Mirrors

In Figure 4.7, an ODI obtained through a paraboloidal mirror is given. The panorama generated from this ODI is shown in Figure 4.8, which is converted by forward mapping in order to visualize the missing information. We observe missing information at the bottom of the panorama due to the decrease in size in the ODI towards the center as expected. However, we do not observe the problem at the upper part as in the spherical mirror case.

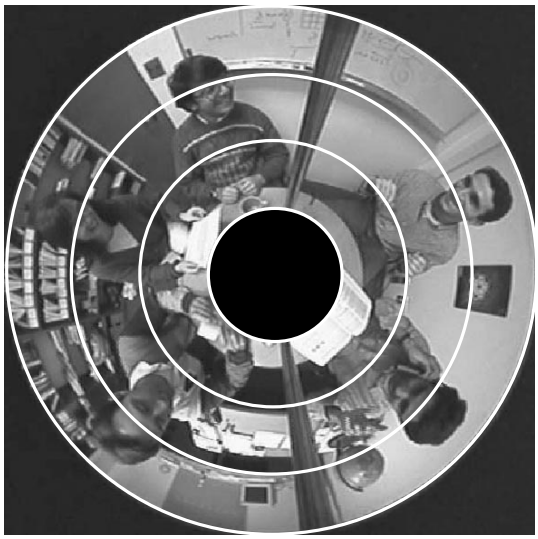

Figure 4.7: An ODI obtained by a paraboloidal mirror [22]

The curvature of the paraboloidal mirror is different from the spherical one in a way that the slope of the mirror surface at the periphery is always finite. Thus, as we move up the angles between the light rays do not increase as in the spherical mirror.

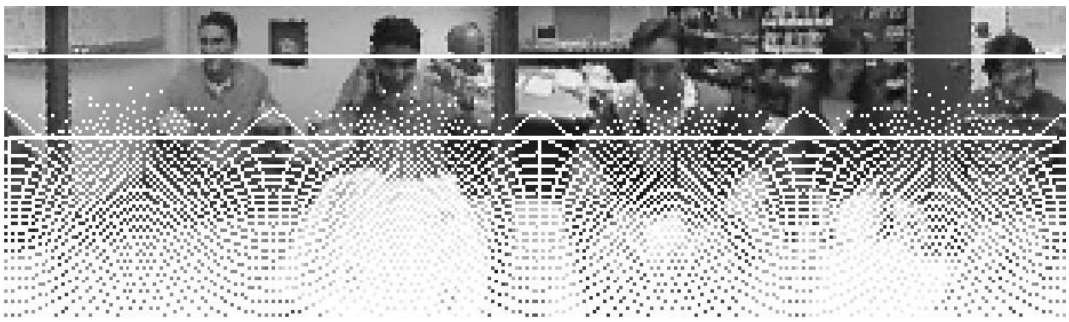

Figure 4.8: Panorama generated from Figure 4.7 by forward mapping

The ODI (Figure 4.7) is divided into three concentric circles with equal thickness and the white lines in Figure 4.8 represent the borders of these circles. The scattering ratios of the peripheral and central sections are shown in Figure 4.9. In this figure, three sections in different gray tones from top to bottom represent the outer, middle and inner one-third regions in the ODI respectively. The result of a 2D analysis concerning paraboloidal mirror reflection is shown at Figure 4.10. We observe that as we move up (go away from the center in the ODI), the angles between the light

rays always decrease. Thus, we can conclude that paraboloidal mirror is more suitable for visualizing objects higher than itself when compared to spherical mirror.

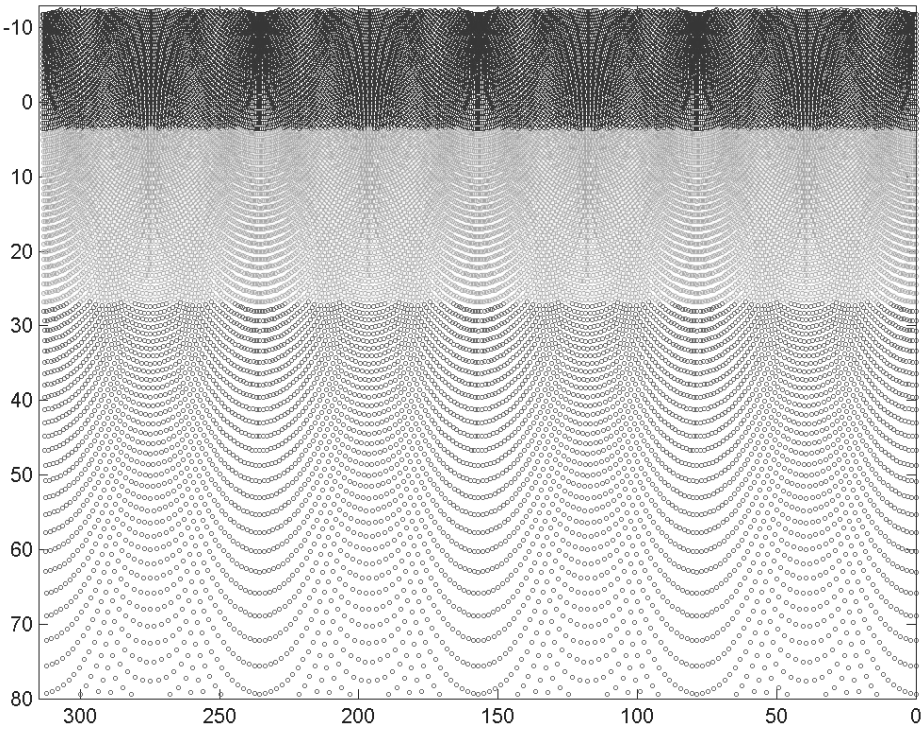

Figure 4.9: MATLAB simulation shows the scattering ratios of the peripheral (top) and central (bottom) sections of the paraboloidal mirror

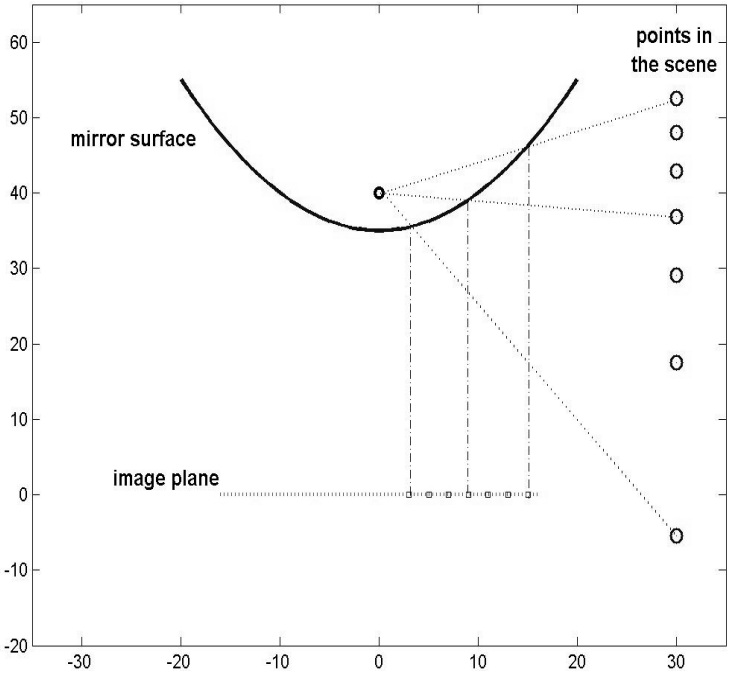

Figure 4.10: Paraboloidal mirror reflection simulation

4.1.3.3 Hyperboloidal Mirrors

In Figure 4.11, an ODI obtained through a hyperboloidal mirror is given. The panorama generated from this ODI is shown in Figure 4.12, which is converted by forward mapping in order to visualize the missing information. We observe missing information only at the bottom of the panorama due to the decrease in size in the ODI towards center as in the paraboloidal case.

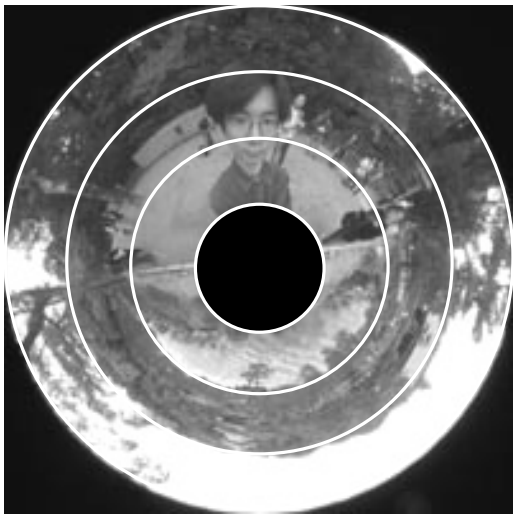

Figure 4.11: An ODI obtained by a hyperboloidal mirror (Courtesy of Kazumasa Yamazawa)

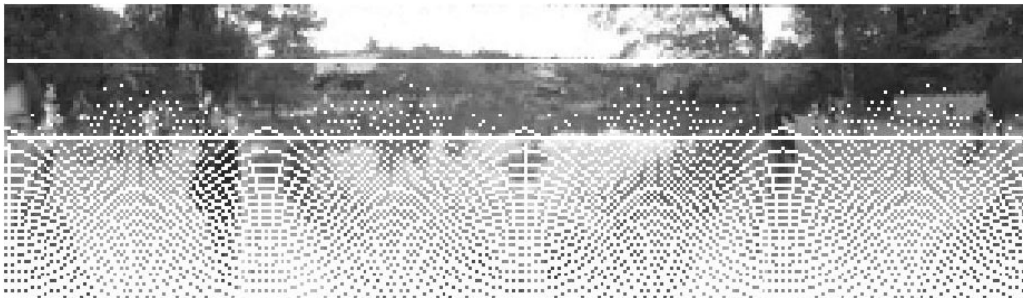

Figure 4.12: Panorama generated from Figure 4.11 by forward mapping

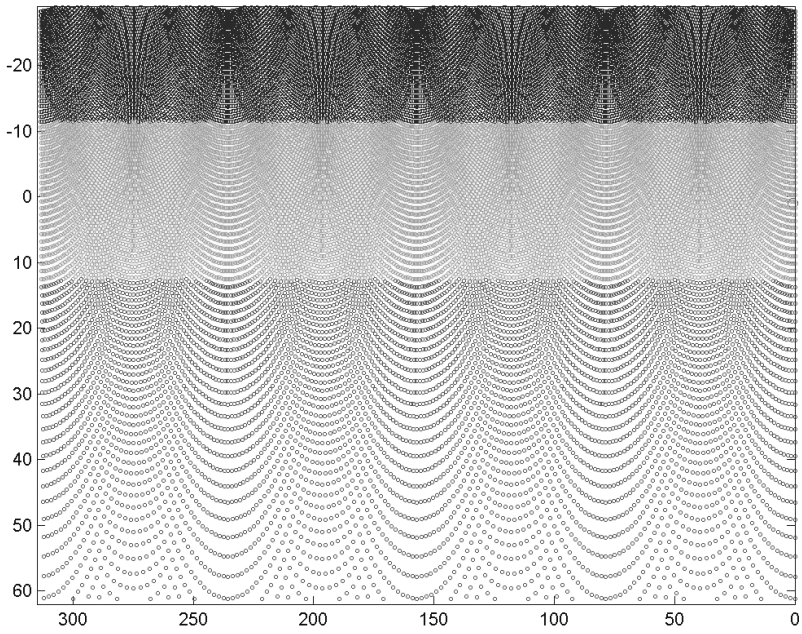

Figure 4.13: MATLAB simulation shows the scattering ratios of the peripheral (top) and central (bottom) sections of the hyperboloidal mirror

The scattering ratios of the peripheral and central sections of the hyperboloidal mirror are shown in Figures 4.13 and 4.14. Results of the simulations and conclusions are similar to the paraboloidal mirror case.

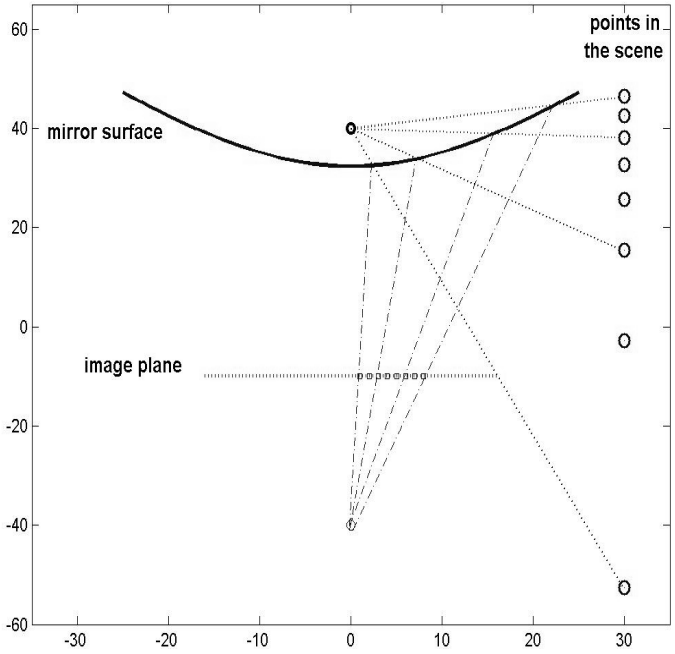

Figure 4.14: Hyperboloidal mirror reflection simulation

4.1.4 'Vertical Field of View' Analysis

4.1.4.1 Paraboloidal Mirrors

The slope of the surface of a paraboloid increases as we move away from the center (along the radius). The relationship between the radius and the slope is:

$$z = \frac{r^2 - h^2}{2h} \tag{4.1}$$

$$\text{Slope; } \frac{dz}{dr} = \frac{r}{h} \tag{4.2}$$

where z is the vertical coordinate axis, r is the horizontal coordinate axis of the mirror surface and h is the parameter of parabola. Although, it is observed that the slope goes to infinity as r goes to infinity, in real life applications the dimensions of a mirror should be restricted. We analyzed how big mirrors we need in order to observe the objects in higher elevations.

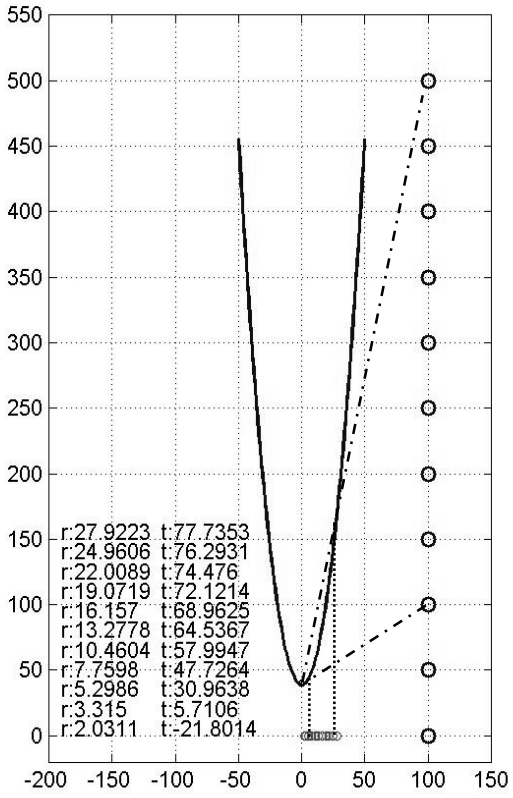

Figure 4.15: Paraboloidal reflection (h=3cm.)

In Figure 4.15, we see the reflection simulation of a paraboloidal mirror with the parameter h=3 cm. The positions of the regularly spaced points in the scene (circles) and corresponding pixels in the omnidirectional image are shown. The position of a pixel in the image along the radius is represented by 'r' and the maximum vertical viewing angle (positive above horizontal) is represented by 't'. We see that when the maximum viewing angle is 48 degrees above horizontal the corresponding mirror radius is around 7.8 cm, which makes the diameter 15.6 cm. This size can be considered as the maximum size to be carried, used and protected practically. To view 75 degrees above horizontal we would need a mirror with 45 cm diameter and 160 cm height which is impractical. Another important conclusion is that the inner one-third of the ODI will contain the information of the objects below the elevation of the mirror. This is the area that contains least amount of information and partly covered by the image of camera itself. Thus, such a mirror will not be useful to observe below the horizontal level of the mirror only. To view objects below the mirror level, paraboloidal mirrors with higher values of 'h' should be selected.

4.1.4.2 Hyperboloidal Mirrors

The maximum slope of the surface of a hyperboloid is the slope of the asymptotes of the hyperbola (b/a) as shown in Equation 4.4. This relationship is derived from the formula of the hyperbola (Equation 4.3). The slope defined in Equation 4.4 is shown graphically in Figure 4.16.

$$z = b \cdot \sqrt{\frac{r^2}{a^2} + 1} \qquad \text{where,} \quad a = c^2 - b^2 \tag{4.3}$$

$$\text{Slope; } \frac{dz}{dr} = \frac{b}{a} \tag{4.4}$$

Where, *z* is the vertical coordinate and *r* is the horizontal coordinate of the mirror surface and *a* and *b* are the parameters of the hyperbola. For a hyperboloidal mirror (c = 25cm., b = 24cm., a = 7cm.), the positions of the regularly spaced points in the scene (circles) and corresponding pixels in the image are shown in Figure 4.17. The position of a pixel in the image along the radius is represented by 'r' and the maximum vertical viewing angle (positive above horizontal) is represented by 't'. We see that when the maximum viewing angle is 70 degrees above horizontal the corresponding mirror diameter is about 55 cm, which is impractical. When we consider a mirror having a maximum vertical viewing angle of 50 degrees, corresponding mirror size will be around 15 cm, which can be considered as maximum size to be used.

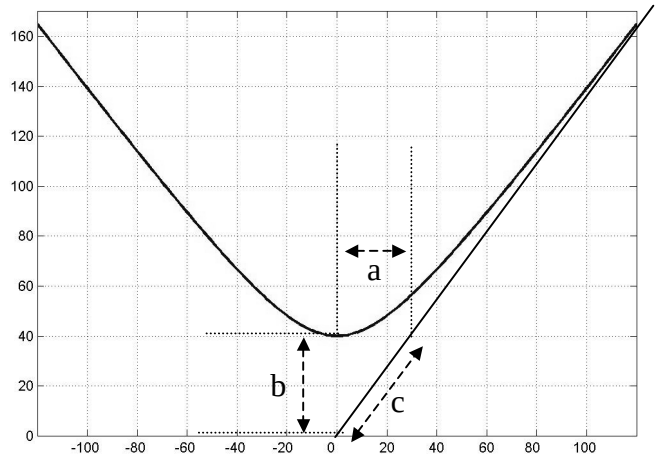

Figure 4.16: The maximum slope of the hyperbola (b/a = 4/3)

Similar to the paraboloidal case, the inner one-third of the ODI will contain the information of the objects below the elevation of the mirror. Thus, such a hyperboloidal mirror will not be useful to observe the objects below the horizontal level of the mirror. To view objects below the mirror level, hyperboloidal mirrors with lower values of ‘b/a’ should be selected.

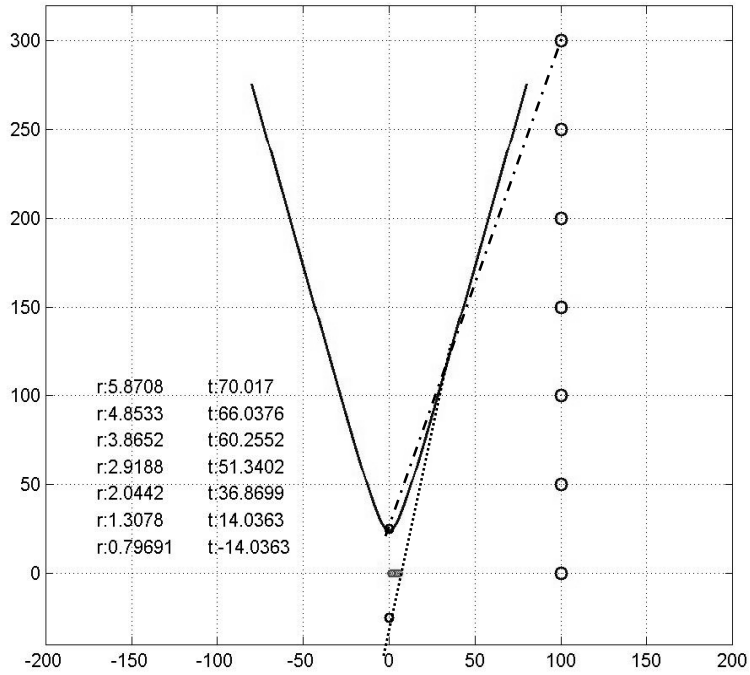

Figure 4.17: Hyperboloidal mirror reflection, c = 25cm., b = 24cm., a = 7cm.

4.2 Projection Surface Alternatives

The panoramic images generated by the single viewpoint catadioptric omnidirectional systems can be defined as projections, viewpoint of which is the focal point of the mirror. All the objects in the scene are viewed as seen from this viewpoint. However, the dimensions of the objects viewed in a panoramic image depend on the projection surface used to convert the omnidirectional image to the panoramic projection. We have examined four projection methods (surfaces); cylindrical, conical, spherical, mixed by experiments held for a virtual bathroom shown at Figure 4.18. In Figure 4.19, the omnidirectional image used for projections is shown. We also generated the panoramic images of the ODI in Figure 4.1 using the mentioned projection surface alternatives to evaluate the results on a real image.

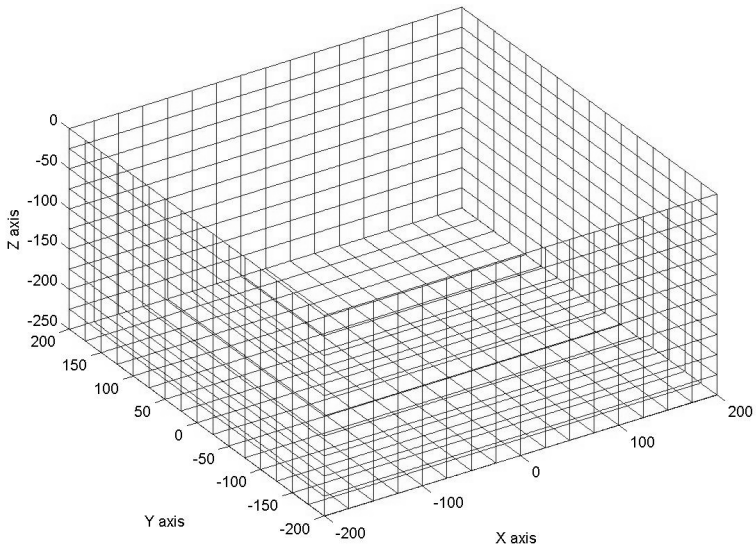

Figure 4.18: Virtual bathroom (tiled with 25x25 faiences) used for projection analysis

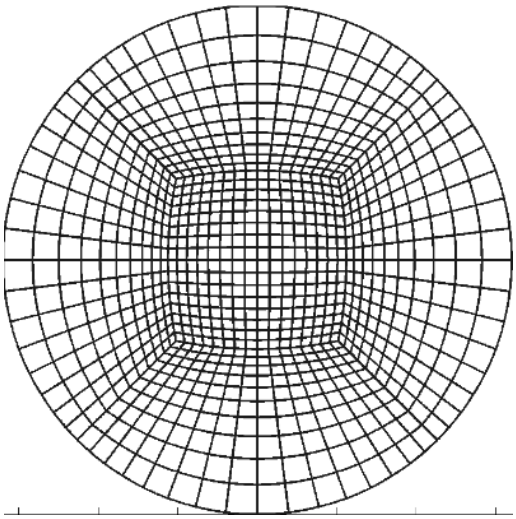

Figure 4.19: The ODI used for projections

4.2.1 Cylindrical Projection

In this method the projection surface is represented as the surface of a cylinder where the viewpoint is located at the central axis of the cylinder (Figure 4.20). As we move up on the surface of the cylinder, same amount (length) of the surface have narrower angles ($x_1=x_2$, $\beta<\alpha$). Normal cameras see x_1 smaller than x_2 since their projections are angle-based as our eyes. However, in cylindrical projection equal distances on the cylinder surface are represented equally on the projection.

In Figure 4.22, thickness of faience rows remain same as we move down on the wall, however our eyes would see the faiences near the bottom smaller compared to the ones at the top.

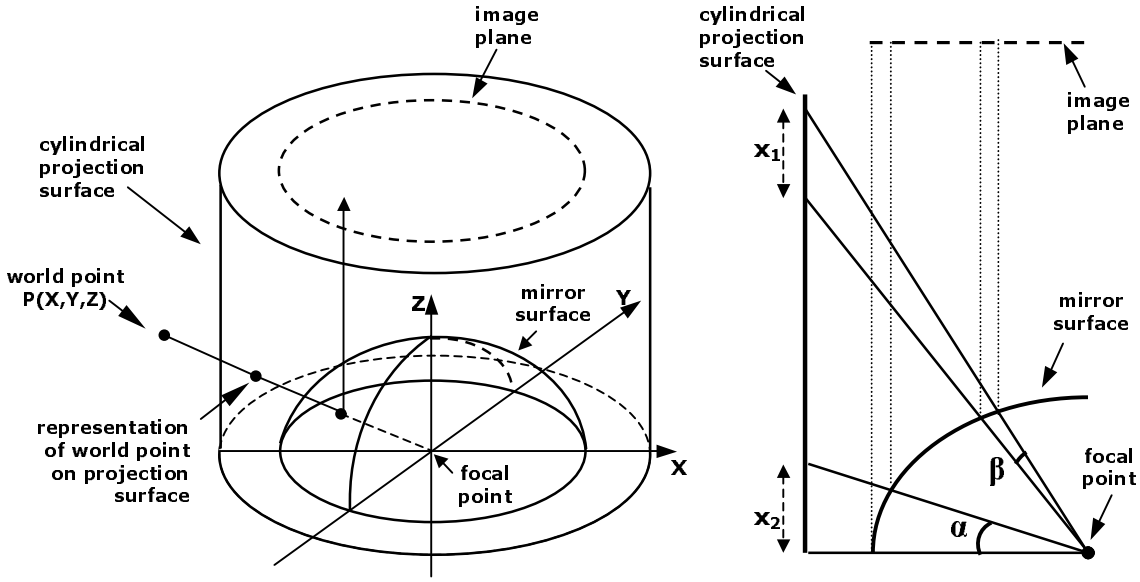

Figure 4.20: Cylindrical projection

Another problem in the cylindrical projection occurs when the real world does not have a wall-like shape as the cylindrical surface has. Problem is visualized in Figure 4.21 and can be observed in Figure 4.22, the floor faiences covers wider area in the image. From the two equal distances in the cylindrical projection surface (x_1=x_2), x_1 is represented by 3 faiences, whereas x_2 is represented by 1.5 faiences.

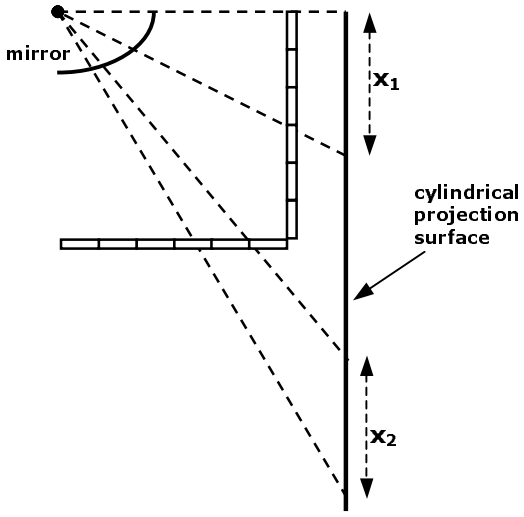

Figure 4.21: Projection of faiences in the bathroom

In Figure 4.23, panoramic image generated from Figure 4.1 by cylindrical projection is shown. As can be observed, the table occupies a wide area in the image because it is close to the mirror.

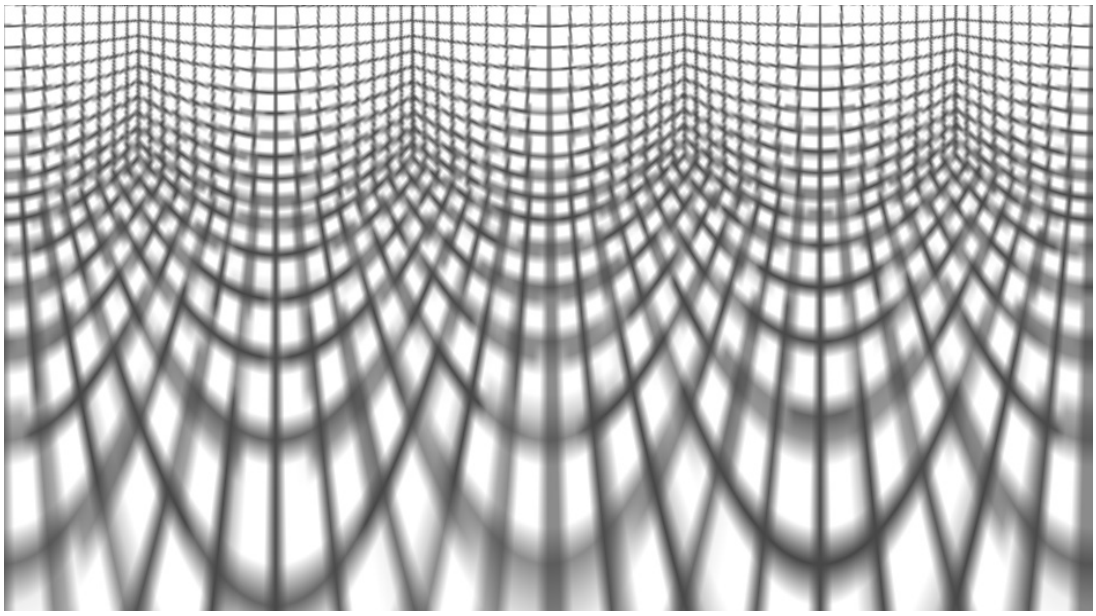

Figure 4.22: Panorama of the bathroom generated by cylindrical projection (540x942)

Figure 4.23: Panorama of the Figure 4.1 generated by cylindrical projection (540x942)

4.2.2 Conical Projection

In this method, the projection surface is a cone, where the viewpoint is located at the center of the bottom circle of the cone (Figure 4.24). As we move up on the conical surface, the angle corresponding to the equal distances on the surface becomes greater first, but then it starts to get smaller ($x_1=x_2=x_3$, $\alpha<\beta>\gamma$). Same lengths in the conical surface will have the same height at the resultant projection.

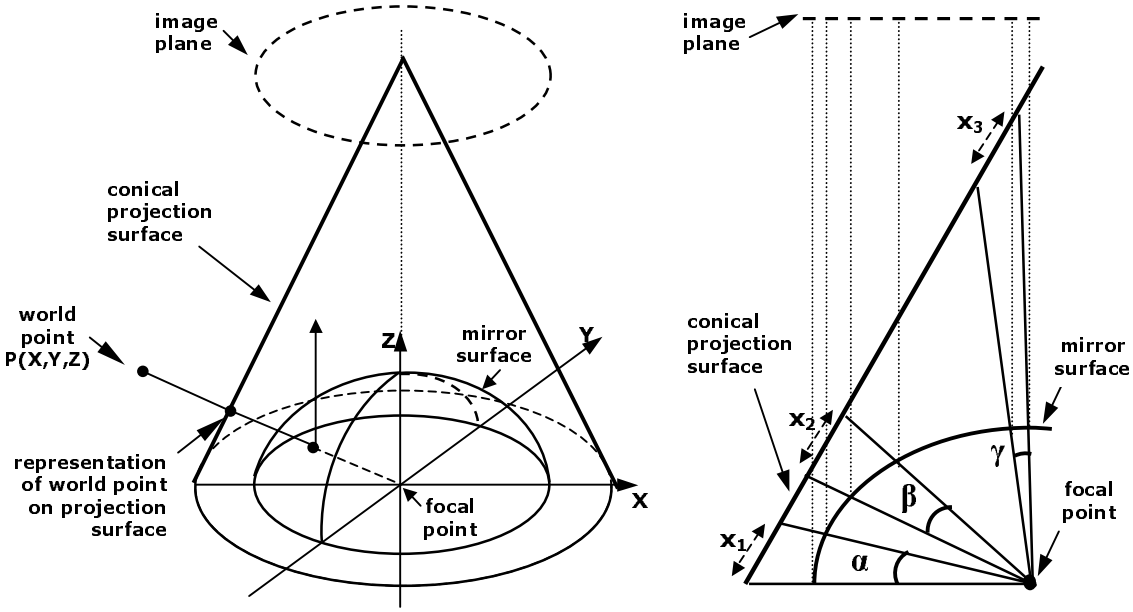

Figure 4.24: Conical projection

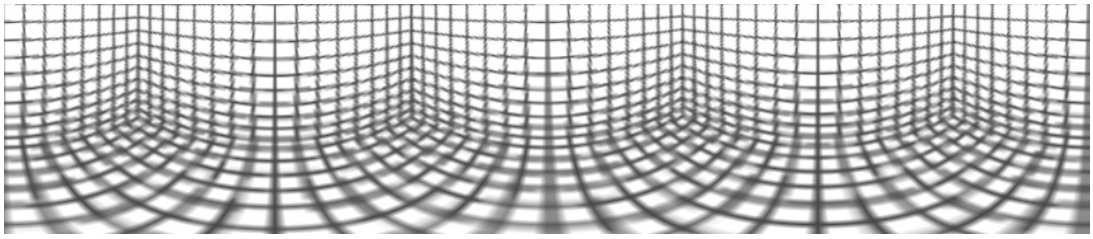

Figure 4.25: Panorama of the bathroom generated by conical projection (181x942)

Figure 4.26: Panorama of the Figure 4.1 generated by conical projection (181x942)

Panoramic image of the bathroom generated by conical projection is shown in Figure 4.25. Although it is not angle-based the resultant image is closer to what our eyes see. Faiences on the wall cover smaller areas as we move down. In Figure 4.26, panoramic image generated from Figure 4.1 by conical projection is shown. The area in the image occupied by the table is reasonable.

4.2.3 Spherical Projection

In this method the projection surface is represented as the surface of a sphere where the viewpoint is located at the center of the sphere (Figure 4.27). Same length of the surface has same projection angles ($x_1=x_2$, $\alpha=\beta$) as regular cameras and our eyes see. Thus, in spherical projection, vertical distribution of the pixels is angle based. Panoramic images of the virtual bathroom and Figure 4.1 generated by spherical projection are shown in Figures 4.28 and 4.29 respectively.

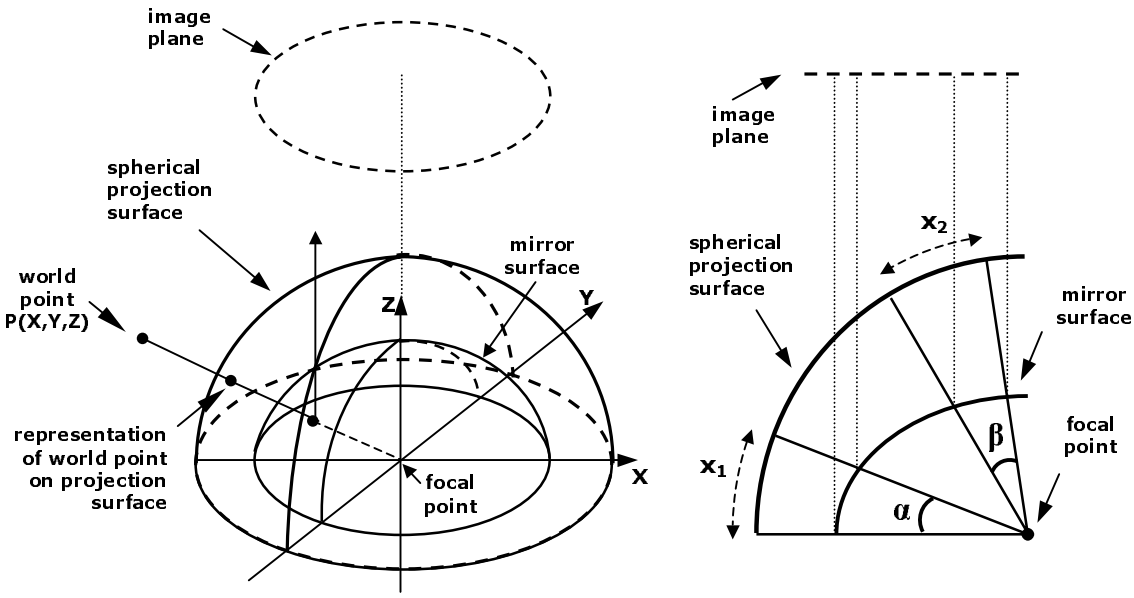

Figure 4.27: Spherical projection

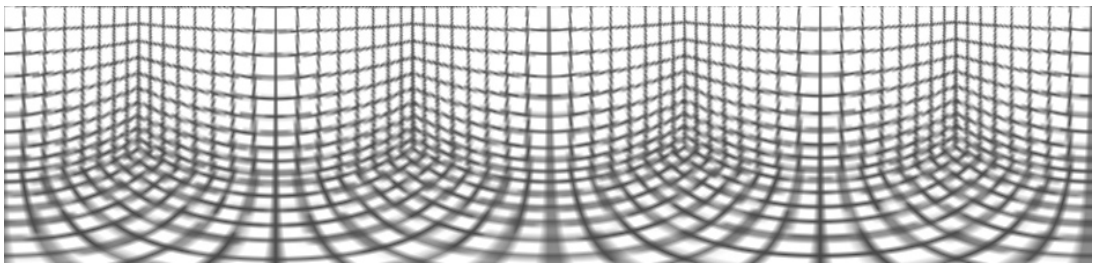

Figure 4.28: Panorama of the bathroom generated by spherical projection (196x942)

Figure 4.29: Panorama of the Figure 4.1 generated by spherical projection (196x942)

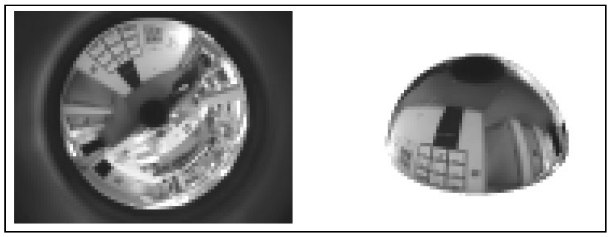

Figure 4.30: Photographic representation of the spherical projection surface (on the right) generated from an ODI (on the left), [48]

Although not directly related to the issue of projection surface analysis, Daniilidis et al. [48] suggested using a sphere as the underlying domain of the image processing for catadioptric omnidirectional images. According to their work, applying operators (e.g. gaussian filter) on a spherical surface and using samples from the omnidirectional image produces better results than applying operators directly on the omnidirectional image. Photographic representation of the spherical projection surface is given in Figure 4.30.

4.2.4 Mixed Method for Projection

In this method, projection surface is not defined as a certain geometric shape but designed to be exactly parallel to the shape of the environmental real world. It aims to represent the equal lengths in the real world equally in the projection. In Figure 4.31, the real world is assumed to be the bathroom, midpoint of which coincides with the focal point of the mirror.

Because the method does not work angle-based, distances on the projection surface does have a direct relationship with the projection angles ($x_1=x_2$, $\alpha\neq\beta$). Thus, the resultant projection is different from what human eye and regular cameras see.

Panoramic image of the bathroom generated by mixed projection method is shown in Figure 4.32. We could not generate the panorama from Figure 4.1 by mixed projection since we do not know the size of the room and the locations of the objects viewed.

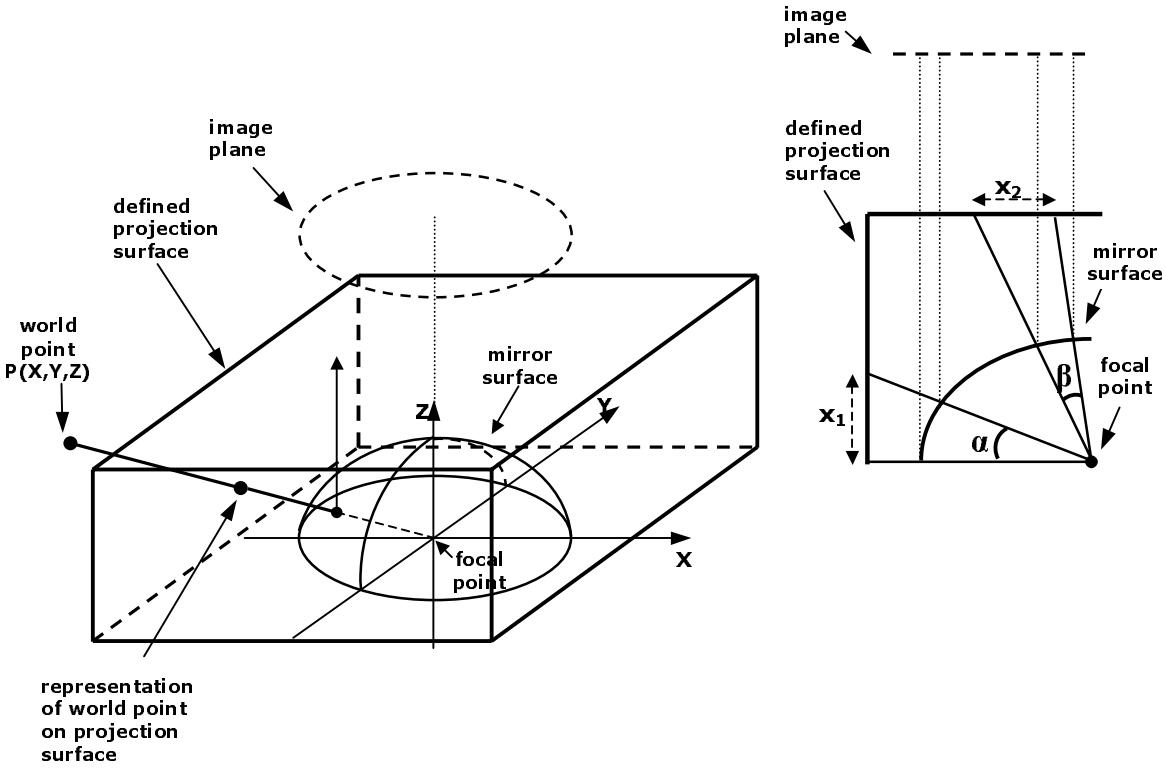

Figure 4.31: Mixed method for projection

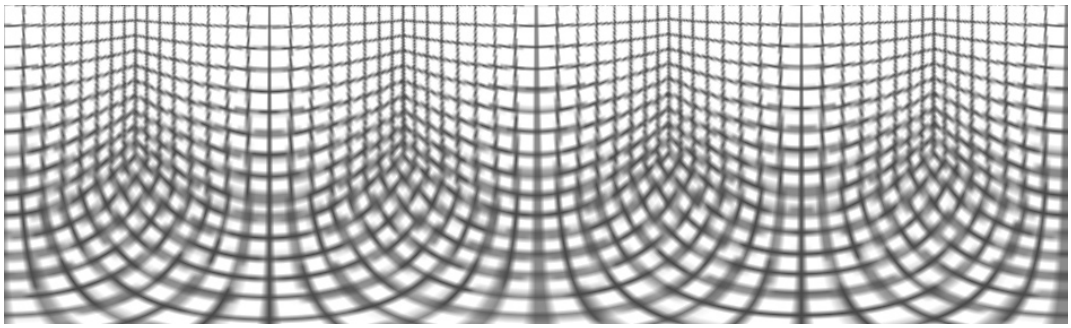

Figure 4.32: Panorama of the bathroom projected mixed method (282x942)

4.2.5 Conclusions

By modifying the projection surface we can manipulate the panoramic images. Cylindrical projection constitutes an unappealing view for the objects imaged towards the center of the ODI. Mixed method is not successful as well because it creates an unrealistic representation, i.e. it conflicts what our eyes see. Conical and spherical projections create appealing panoramic images since they generate realistic vertical representations. Unfortunately, horizontal widening for the regions towards the center of the ODI is a problem that cannot be solved because in these regions smaller circumferences have to be represented as whole rows of the panoramic image.

It can be said that, spherical projection should be preferred since it has an angle-based vertical representation like regular cameras; therefore it suits best to our eyes best. However, most of the current panoramic imaging applications use cylindrical projection. It requires less computational cost and when a high cylindrical representation is not especially desired the low-quality regions can be cut from the panorama.

CHAPTER 5

Panoramic Image Enhancement

In Chapter 4, resolutional properties of omnidirectional images were examined and the blurring problem was defined. This occurs in the panoramic images due to the lack of information while transferring pixels from ODI to the panoramic image. Several techniques that we have used and developed to solve this blurring problem are presented in this chapter.

5.1 Standard Interpolation Methods

While generating new images, to guarantee that a value is generated for every pixel in the output image, we must use the inverse transformation to determine the position in the input image from which a value must be sampled. This is known as the 'backward mapping'. In the following paragraphs widely used interpolation methods are described using the backward mapping.

5.1.1 Nearest-neighbor (zero-order) Interpolation

In the nearest-neighbor method, for each pixel in the image to be generated, the corresponding location in the input image is determined. The intensity value of the pixel that is nearest to the determined location is assigned as the intensity value of the pixel in the output image. This is the simplest method we used to obtain an intensity value.

5.1.2 Bilinear (first-order) Interpolation

In the bilinear interpolation method, the intensity value of the pixel in the output image is filled with the four pixels in the surrounding of the corresponding location in the input image. Calculation of intensity values with bilinear interpolation is given in Equation 5.1 and geometry for omnidirectional case in shown in Figure 5.1.

The intensity value of point to be estimated (x,y) can be defined in [60] as:

$$I(x,y) = I(x_0,y_0) + [I(x_1, y_0) - I(x_0, y_0)] \cdot \Delta x + [I(x_0, y_1) - I(x_0, y_0)] \cdot \Delta y$$
$$+ [I(x_1, y_1) + I(x_0,y_0) - I(x_0, y_1) - I(x_1, y_0)] \cdot \Delta x \cdot \Delta y \quad (5.1)$$

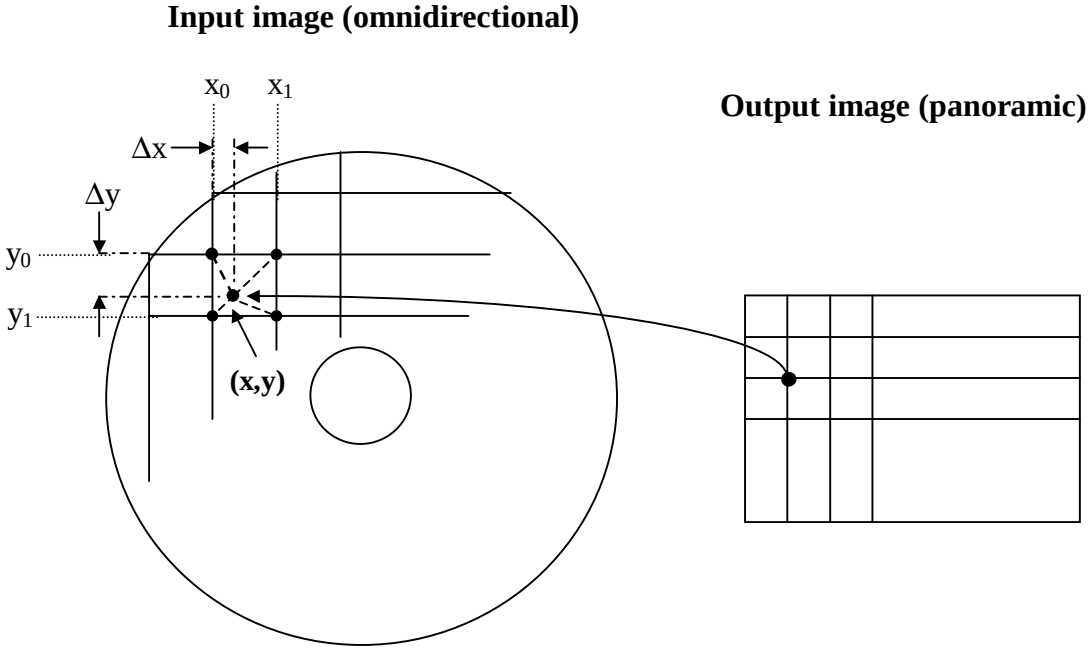

Figure 5.1: Image generation from omnidirectional image with bilinear interpolation

5.1.3 Bicubic (third-order) Interpolation

In the bicubic interpolation method, the intensity value of the pixel in the output image is filled with the 16 neighbor pixels of the corresponding location in the input image. Bicubic interpolation technique, as described by Bourke [61], is given in Figure 5.2 and Equation 5.2.

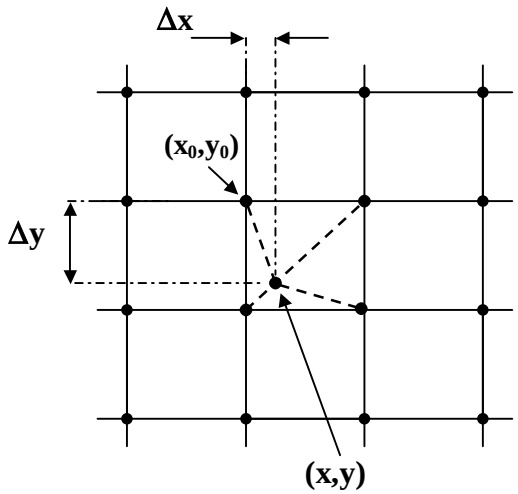

Figure 5.2: Pixel neighborhood in input image used for bicubic interpolation

The intensity value of point to be estimated (x,y) can be defined as:

$$I(x,y) = \sum_{m=-1}^{2}\sum_{n=-1}^{2} I(x_0+m, y_0+n)\cdot R(m-\Delta x)\cdot R(\Delta y - n) \tag{5.2}$$

where, m and n is the 4x4 grid span around (x,y) and R is the cubic weighting function given below:

$$R(a) = \frac{1}{6}\left[P(a+2)^3 - 4\cdot P(a+1)^3 + 6\cdot P(a)^3 - 4\cdot P(a-1)^3\right]$$

$$P(a) = \begin{cases} a & a > 0 \\ 0 & a \le 0 \end{cases}$$

5.2 Image Enhancement Effects of Standard Interpolation Methods

To overcome the problem described in section 4.1.2, we first analyzed the enhancement effects of nearest-neighbor, bilinear, bicubic and cubic spline interpolation methods. Performances of these four interpolation methods were compared by a series of computations performed on checkerboard test patterns.

Nearest neighbor is the simplest interpolation method to be used for geometric conversions, but not the most effective. Bilinear interpolation is a popular method among many application fields. Bicubic interpolation is also very effective, but not

preferred often because of its higher computational cost. Cubic spline is also a well-known method, usually used for approximation problems. Employment of spline interpolation in our tests is performed by using 'csape' function of the MATLAB. This function is defined as 'cubic spline interpolation with end conditions', in which first and second degree continuity is enforced.

5.2.1 Bilinear and Bicubic Interpolation Comparison with Noisy Images

First, enhancement performances of bilinear and bicubic are compared with each other. Interpolation performed on checkerboards, having different cell size and contrast. These images are exposed to different amount of gaussian or salt & pepper noise. Gaussian noise amount is manipulated by its variance. In salt & pepper noise, the intensity values of some pixels (amount is defined as percentage and pixels selected randomly) are changed to 0 (black) or 255 (white). In Figures 5.3 and 5.4 two examples of generated checkerboards are given.

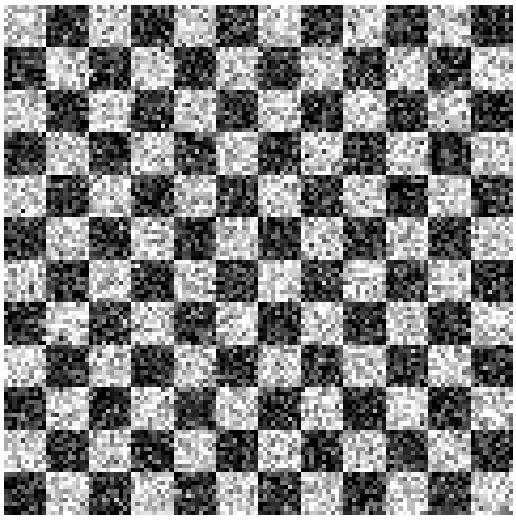

Figure 5.3: Noised checkerboard 10x10, initial intensity values: [50,200], Gaussian noise (mean=0, variance=0.04)

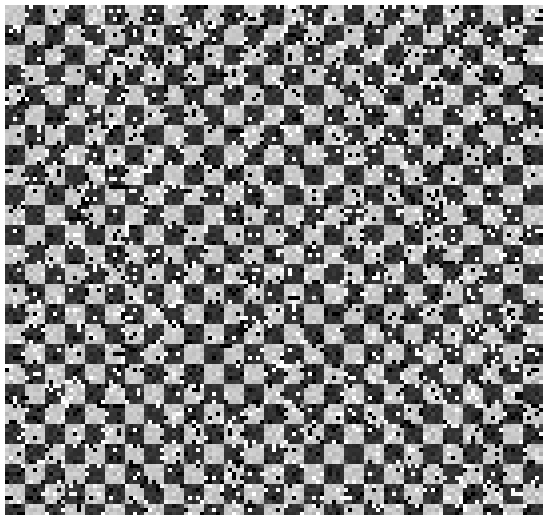

Figure 5.4: Noised checkerboard 5x5, initial intensity values: [50,200], Salt-pepper noise (%15)

Comparison of enhancement performance is done on the basis of intensity value differences between original and interpolated images. In other words, the most effective interpolation method is the one that gives the minimum intensity difference between the original image and interpolated image. For an m *x* n image, intensity difference is defined as:

$$\varepsilon = \sum_{i=1}^{m}\sum_{j=1}^{n}\left|X(i,j) - X_0(i,j)\right| \tag{5.3}$$

X and X_0 are the original and noisy image respectively, (i,j) is the location of the pixel of which the intensity value is measured. Resultant differences of the test are given in Table 5.1. Enhancement effects in flat surfaces and edges are separated in Table 5.2.

Intensity values along a typical row of the checkerboard are graphed in order to visualize the enhancement effects. Sample graphs are given in Figures 5.5 - 5.11. Gaussian variances 0.01 vs. 0.04 graphs and contrast dif. 50-200 vs. 80-170 graphs are also obtained.

Table 5.1: Total intensity differences between original and interpolated images

Contrast and noise information	**Bilinear interpolation**	**Bicubic interpolation**	**Noisy image**
Gaussian noise	**ε**	**ε**	**ε**
Contrast; [50-200] Variance; 0.01	262893	255488	287953
Contrast; [80-170] Variance; 0.01	210639	197651	288152
Contrast; [100-150] Variance; 0.01	177258	160011	286547
Contrast; [50-200] Variance; 0.04	364780	342129	519920
Contrast; [80-170] Variance; 0.04	338164	303402	560086
Contrast; [100-150] Variance; 0.04	322654	282094	575617
Salt & pepper noise	**ε**	**ε**	**ε**
Contrast; [50-200] Percentage; 0.05	210803	229872	93845
Contrast; [80-170] Percentage; 0.05	151360	159589	87055
Contrast; [50-200] Percentage; 0.15	346686	348393	267495
Contrast; [80-170] Percentage; 0.15	287821	278998	267255
Contrast; [50-200] Percentage; 0.25	476247	462282	456240
Contrast; [80-170] Percentage; 0.25	403304	377769	458140
Contrast; [50-200] Percentage; 0.35	589788	560205	633050
Contrast; [80-170] Percentage; 0.35	499420	458589	627065

* All computations done with 10x10 checkerboard 119x119 pixel images
* Pixel intensity value range: [0,255]

Table 5.2: Total intensity differences separated into flat regions and edges

Contrast and noise information	**Bilinear interpolation**		**Bicubic interpolation**		**Noisy image**	
Gaussian noise	**ε** flat	**ε** edge	**ε** flat	**ε** edge	**ε** flat	**ε** edge
Contr; [50-200] Variance; 0.01 Decrease of error after interpol.	102839 45%	156514 -64%	89057 52%	163085 -72%	187544	95386
Contr; [80-170] Variance; 0.01 Decrease of error after interpol.	104498 45%	103188 -8%	90071 52%	104933 -10%	188322	95535
Contr; [100-150] Variance; 0.01 Decrease of error after interpol.	102325 46%	72199 23%	88510 53%	69090 27%	188141	93964
Contr; [50-200] Variance; 0.04 Decrease of error after interpol.	180140 46%	179690 -3%	155713 54%	182093 -4%	335476	175577
Contr; [80-170] Variance; 0.04 Decrease of error after interpol.	199147 46%	133410 28%	171665 54%	127245 31%	366050	184566
Contr; [100-150] Variance; 0.04 Decrease of error after interpol.	204491 46%	113015 40%	175895 54%	101552 46%	377381	188742

* Size of the flat surface is 8x8 out of 10x10 checkerboard.
* Minus sign in 'decrease of error' means error is increased.
* All computations done with 10x10 checkerboard 119x119 pixel images.
* Pixel intensity value range: [0,255]

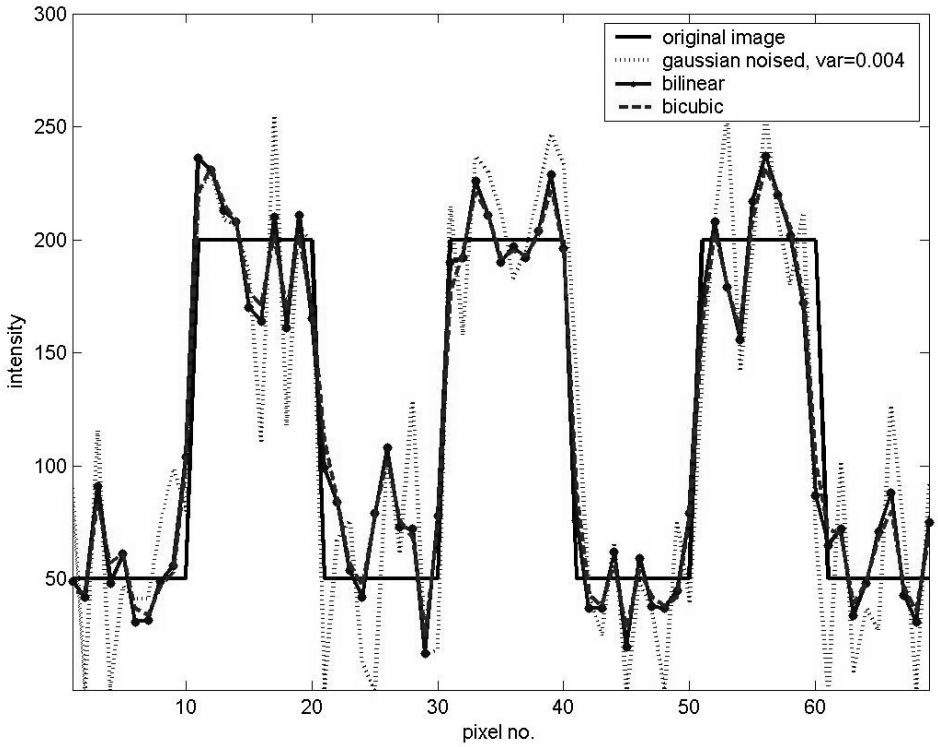

Figure 5.5: Intensity values along a horizontal line segment of an interpolated checkerboard, Square size:10x10, Contrast: [80,170], Gaussian var: 0.04

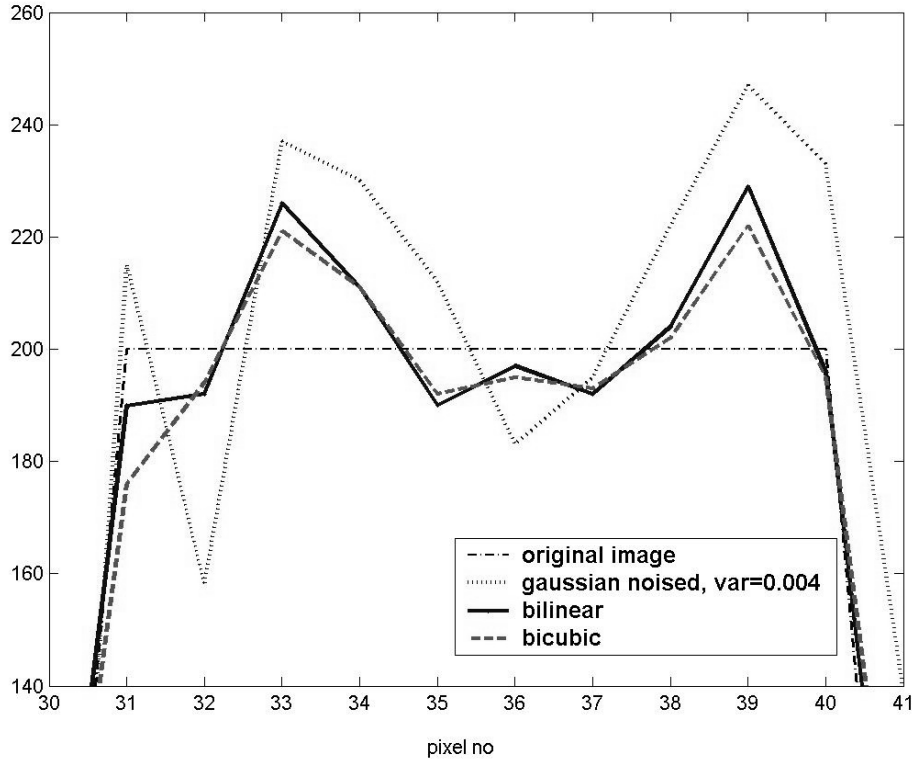

Figure 5.6: Local intensity values taken from Figure 5.5

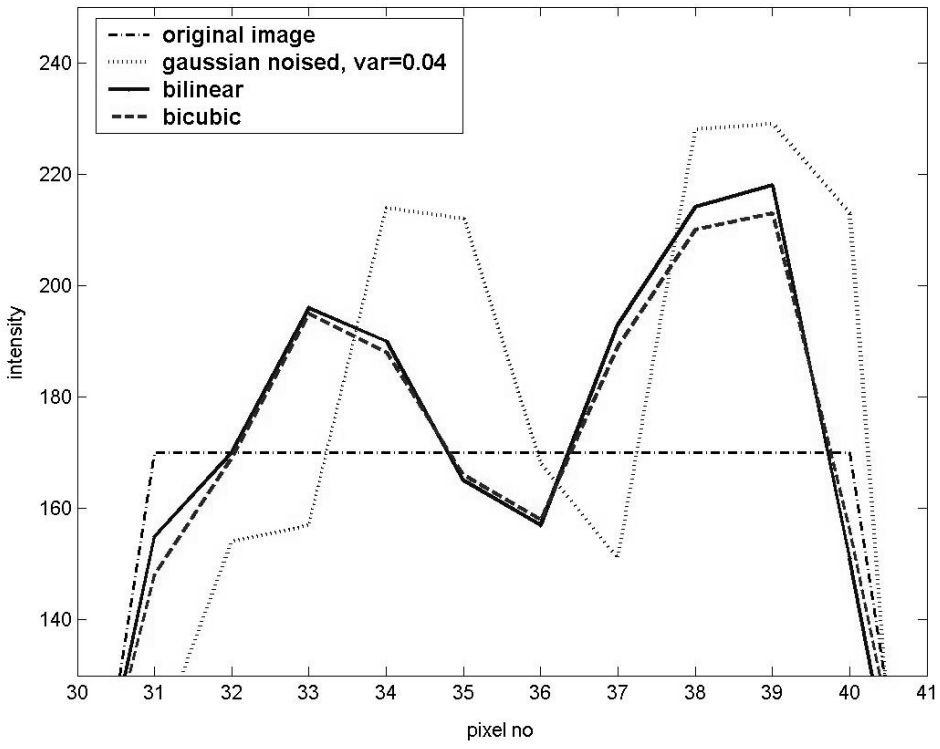

Figure 5.7: Local intensity values taken from Figure 5.5

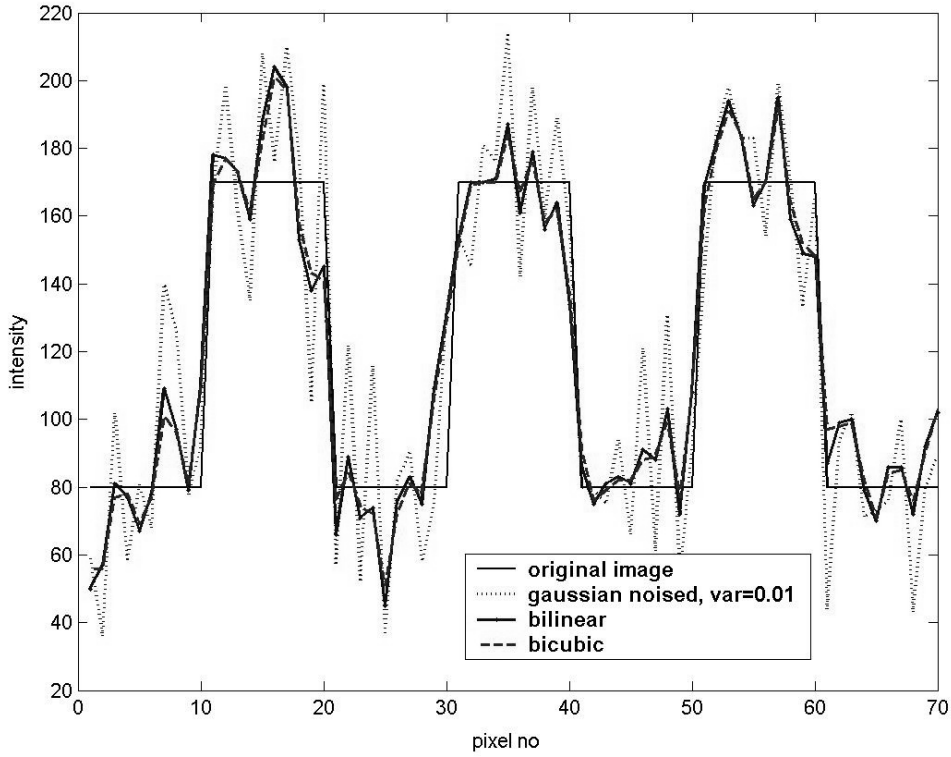

Figure 5.8: Intensity values along a horizontal line segment of an interpolated checkerboard, Square size:10x10, Contrast: [80,170], Gaussian var: 0.01

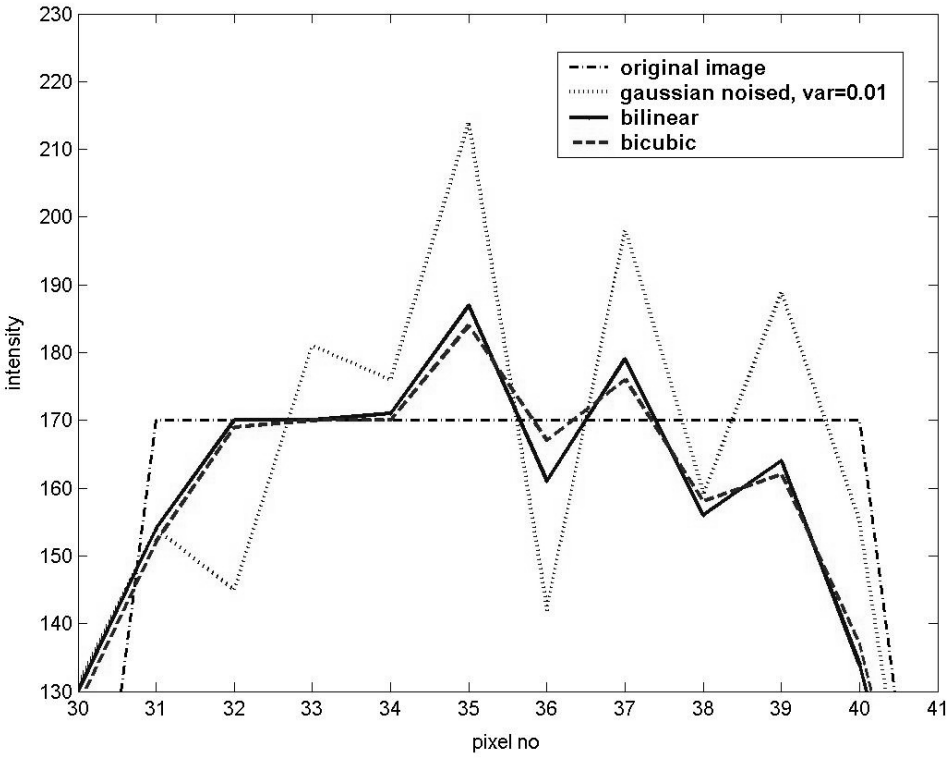

Figure 5.9: Local intensity values taken from Figure 5.8

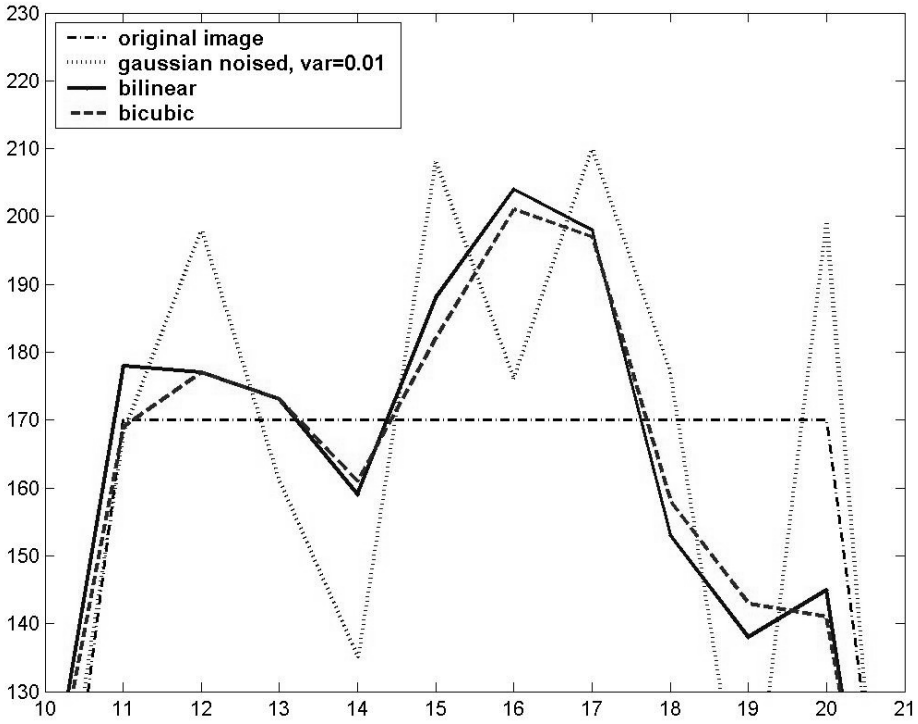

Figure 5.10: Local intensity values taken from Figure 5.8

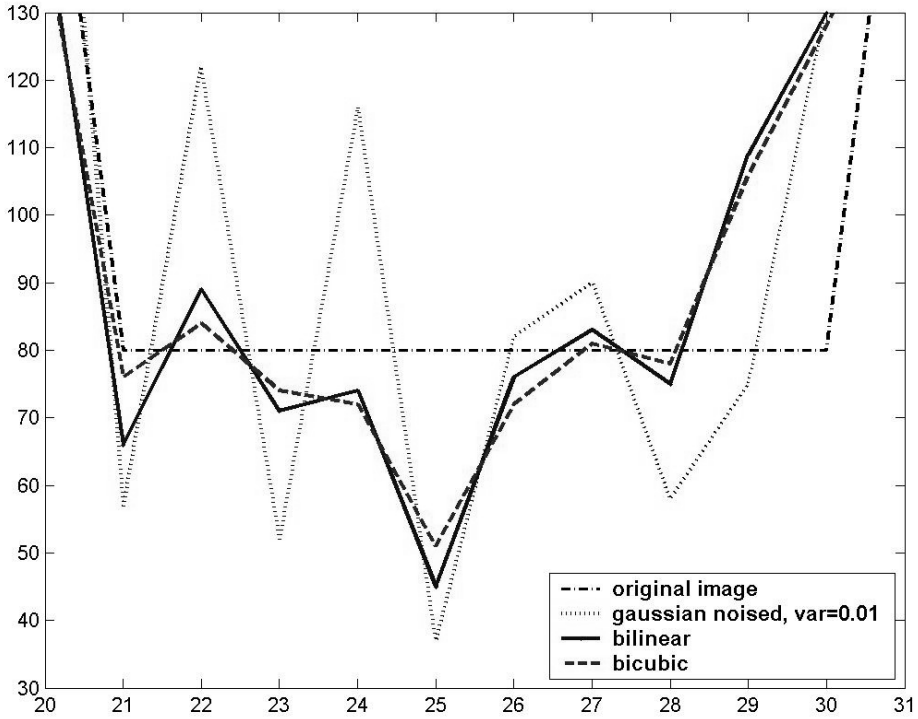

Figure 5.11: Local intensity values taken from Figure 5.8

5.2.2 Further Analysis Including Cubic Spline and Nearest Neighbor Methods

When the results of bilinear and bicubic interpolation for real omnidirectional images are compared, it is not possible to detect the difference between the images visually. An omnidirectional image (ODI) obtained by a hyperboloidal mirror is given in Figure 5.12. Figures 5.13 and 5.14 are the panoramic images generated from this ODI by bilinear and bicubic interpolation respectively.

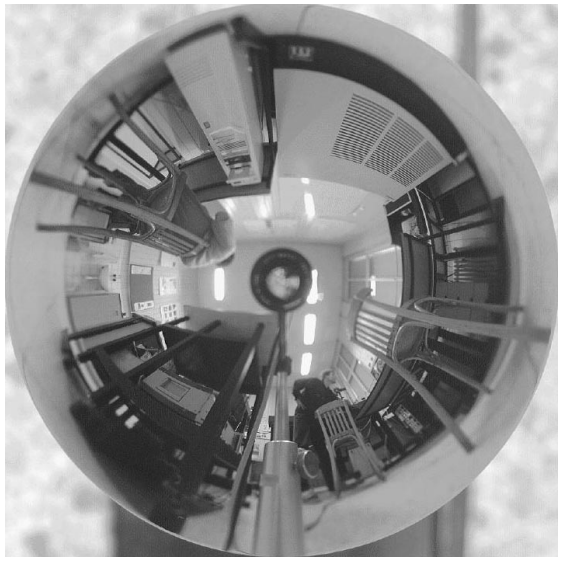

Figure 5.12: Omnidirectional image obtained by a hyperboloidal mirror

Figure 5.13: Panoramic image generated from the ODI in Fig. 5.12 by bilinear interpolation

Figure 5.14: Panoramic image generated from the ODI in Fig. 5.12 by bicubic interpolation

Concluding that the bilinear and bicubic interpolation methods do not produce ‘visually different’ results, ‘nearest neighbor’ and ‘cubic spline’ methods are included in the inspection and examined both visually and by pixel intensity analysis. Cubic spline method is employed by using ‘csape’ function of the MATLAB.

Panoramic images generated from the ODI in Figure 5.12 by cubic spline and nearest neighbor methods are given in Figures 5.15 and 5.16 respectively. The resultant image created by cubic spline is visually the same with the resultant images of bilinear and bicubic interpolation. However, the panoramic image created by nearest neighbor method is obviously different. Quadrangles occurred in the upper part of the image instead of blurring occurred when the other methods are employed.

Intensity value graphs showing the results of all four methods employed are generated. Sample graphs are given in Figures 5.17 and 5.18, original checkerboard and noisy image intensity values are not depicted for the sake of simplicity. Intensity value of 1 corresponds to 200 for [0,255] gray scale range.

Figure 5.15: Panoramic image generated from the ODI in Fig. 5.12 by cubic spline method

Figure 5.16: Panoramic image generated from the ODI in Fig. 5.12 by nearest neigh. method

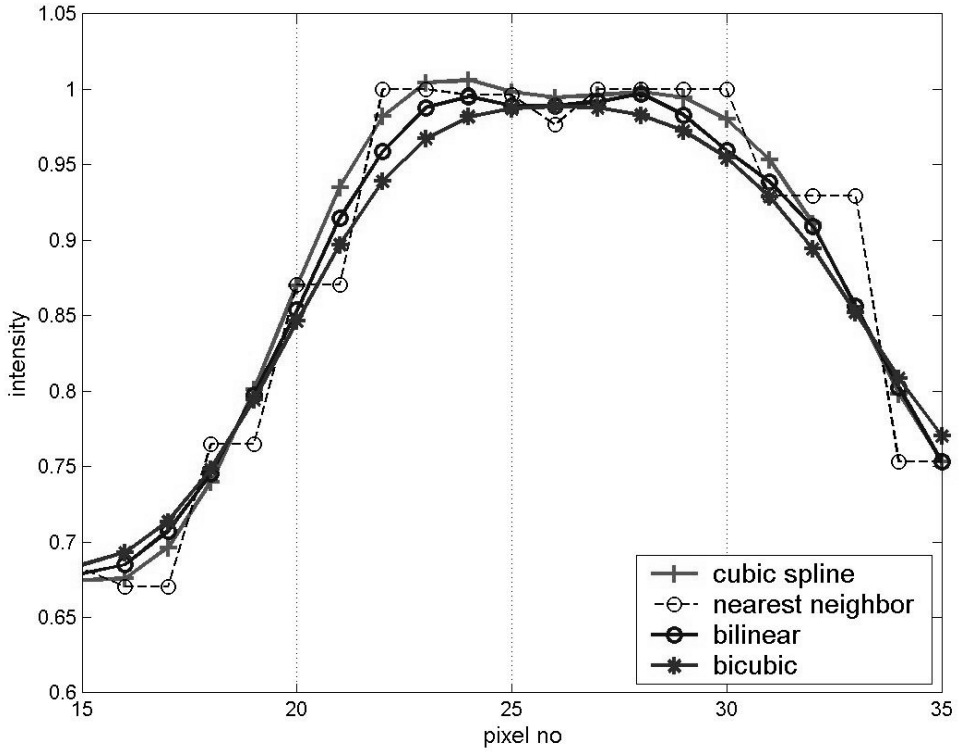

Figure 5:17: Intensity values of a noised checkerboard image interpolated by cubic spline, nearest neighbor, bilinear and bicubic interpolation methods. Sample local view.

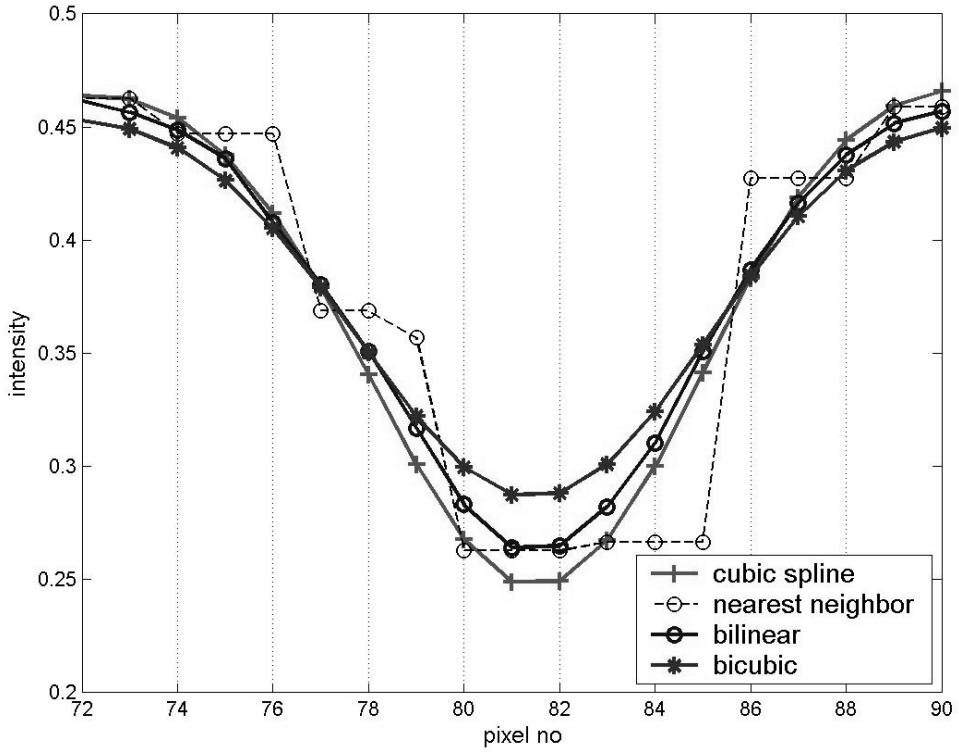

Figure 5:18: Intensity values of noised checkerboard image interpolated by cubic spline, nearest neighbor, bilinear and bicubic interpolation methods. Sample local view.

5.2.3 Effects of Interpolation Methods for Upsampled Images

To implement the upsampling occurring while transferring pixels from the ODI to the panoramic image and results in blurring in panoramic images (explained in Section 4.1.2), interpolation tests are held with enlarged checkerboards as well. In Figures 5.19 - 5.22, checkerboards of which cell size are enlarged from 3x3 to 15x15 with different interpolation methods. Intensity values along a row of these upsampled checkerboards are transferred into graphs in order to visualize the enhancement effects. Sample graphs are given in Figures 5.27 - 5.30.

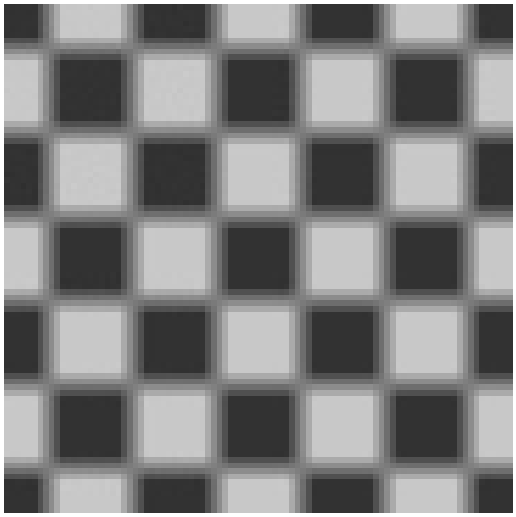

Figure 5.19: Checkerboard upsampled by bilinear interpolation

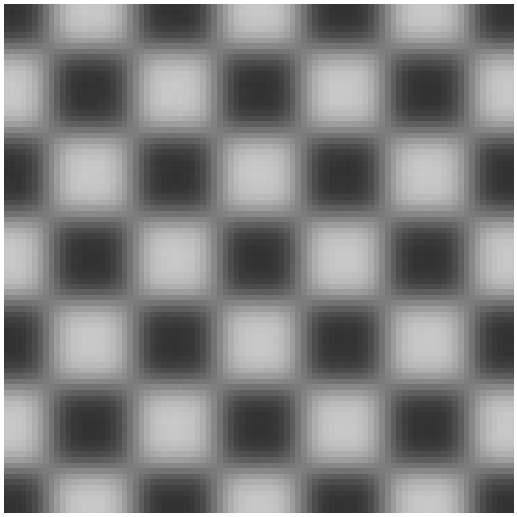

Figure 5.20: Checkerboard upsampled by bicubic interpolation

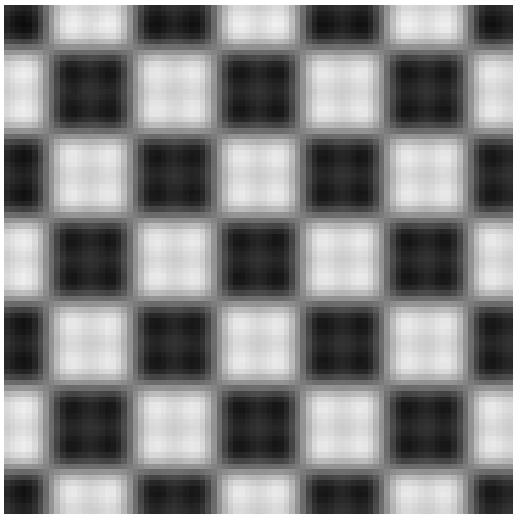

Figure 5.21: Checkerboard upsampled by cubic spline method

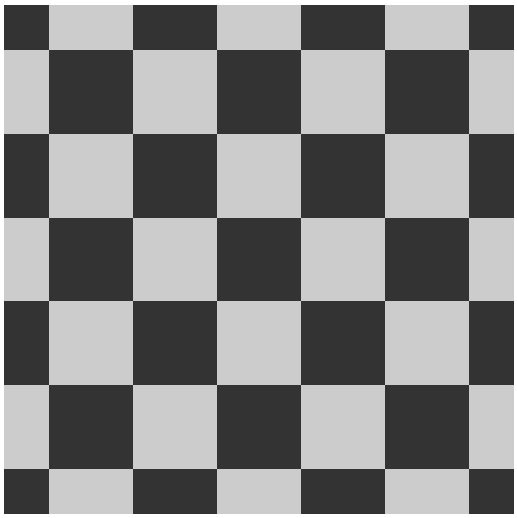

Figure 5.22: Checkerboard upsampled by nearest neighbor method

Upsampling is also performed for 45º rotated checkerboards in order to observe the effects of change in edge alignment. Figures 5.23 - 5.26, checkerboards of which cell size are enlarged from 12x12 to 60x60 with different interpolation methods. Visual effects may differ according to the edge alignment as in the case of nearest neighbor method. Although result of this method is seem to be good for an orthogonal checkerboard (Figure 5.22), for a rotated checkerboard pixel-wide jumps are enlarged and the result is not appealing (Figure 5.26). Moreover, the result of bicubic interpolation (Figure 5.24) is better than the result of bilinear interpolation (Figure 5.23) since it reduces better the one pixel-wide jump effect.

Figure 5.23: 45º Rotated checkerboard upsampled by bilinear interpolation

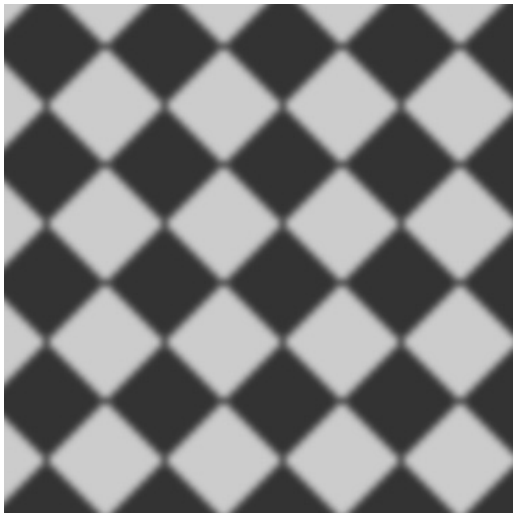

Figure 5.24: 45º Rotated checkerboard upsampled by bicubic interpolation

Figure 5.25: 45º Rotated checkerboard upsampled cubic spline method

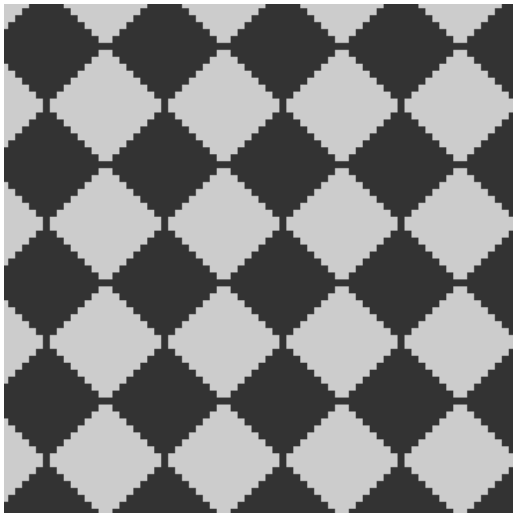

Figure 5.26: 45º Rotated checkerboard upsampled by nearest neighbor method

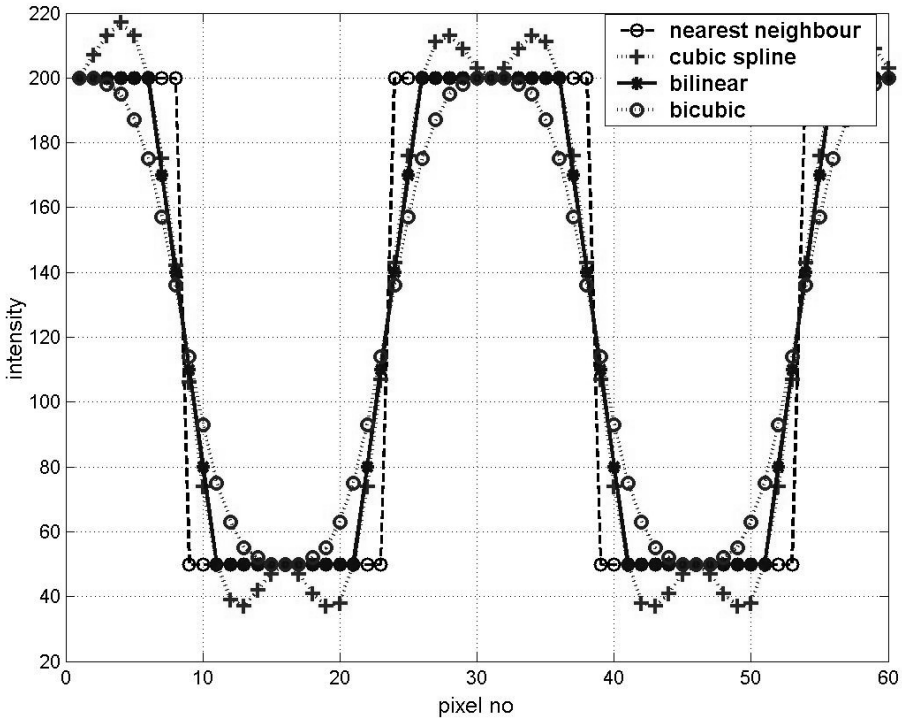

Figure 5.27: Intensity values of a checkerboard upsampled 5 times by nearest neighbor, cubic spline, bilinear and bicubic interpolation methods (cell size is enlarged from 3x3 to 15x15)

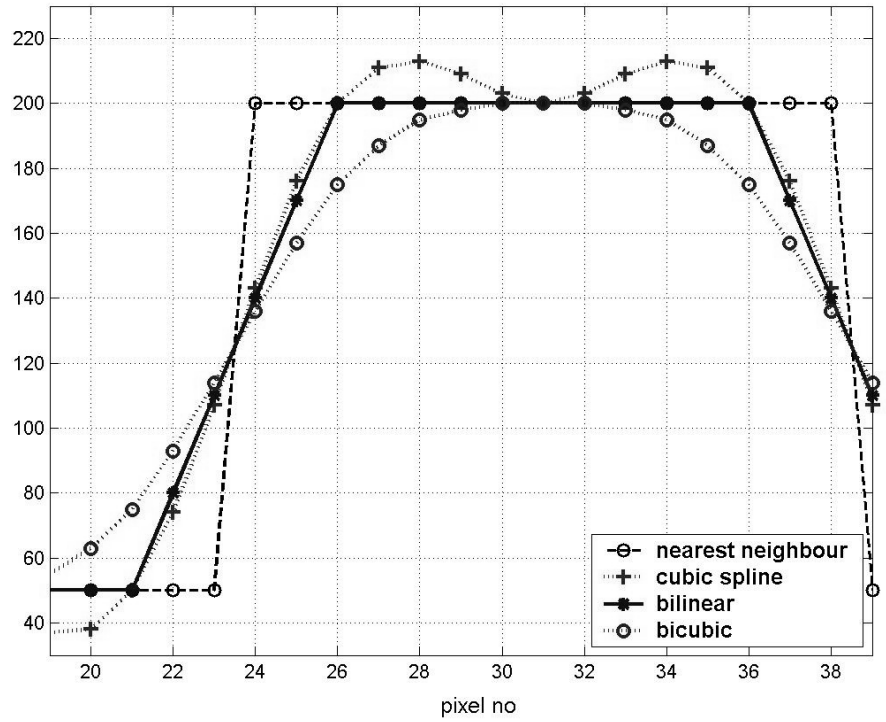

Figure 5.28: Local intensity values taken from Figure 5.27

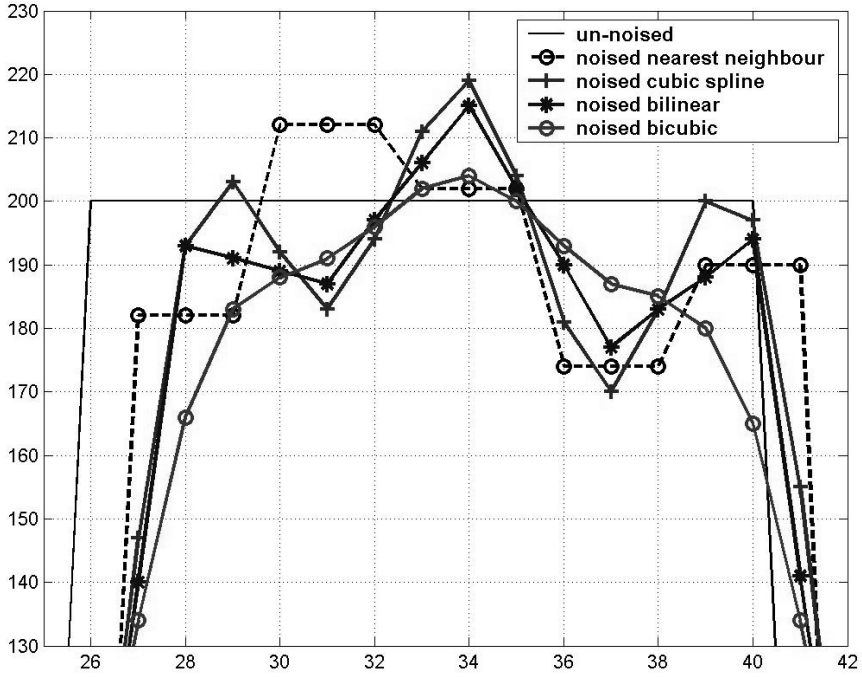

Figure 5.29: Intensity values of a noised checkerboard upsampled 3 times by nearest neighbor, cubic spline, bilinear, bicubic interpolation methods, sample local view (cell size is enlarged from 5x5 to 15x15 and added Gaussian noised variance is 0.01)

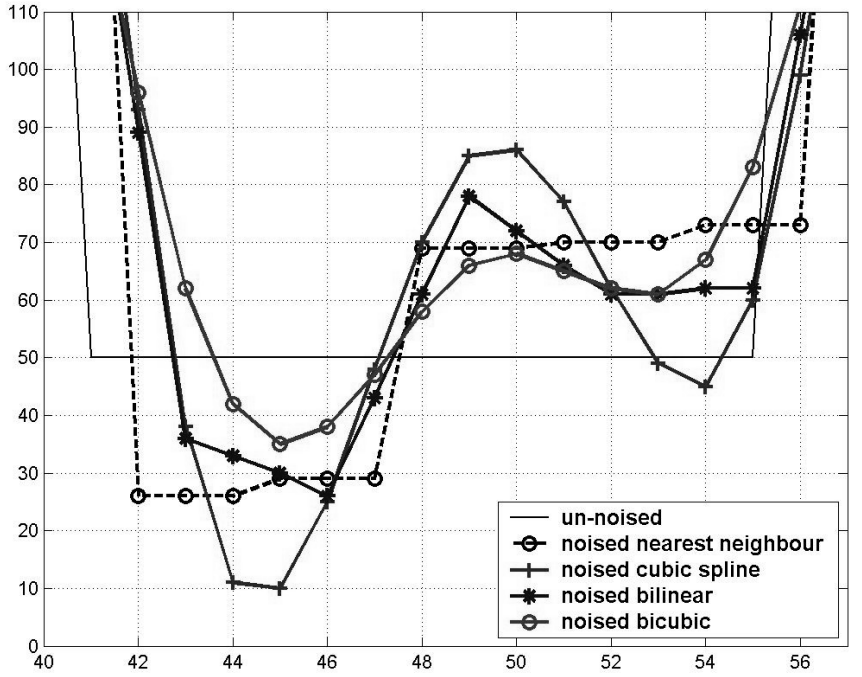

Figure 5.30: Intensity values of a noised checkerboard upsampled 3 times by nearest neighbor, cubic spline, bilinear, bicubic interpolation methods, sample local view (cell size is enlarged from 5x5 to 15x15 and added Gaussian noised variance is 0.01)

5.2.4 Results

The bilinear interpolation, bicubic interpolation, nearest neighbor and cubic spline methods are compared by a series of computations performed on checkerboards, having different cell size and contrast. These images are exposed to gaussian and salt & pepper noises and enlarged to implement the upsampling effect. Panoramic images generated by these methods are also created from omnidirectional images to examine the visual differences.

Gaussian noise:

- Both bilinear and bicubic interpolation improved the gaussian noised images. Total error in image was decreased by both methods.
- Images embedded in Gaussian noise with high variance improved more when compared to the ones with low variance. In other words, error decrease is greater.
- Gaussian noised images having low contrast improved more when compared to the ones having high contrast. High contrast results in overshoots in the edge regions.
- Decrease of error, after bicubic interpolation is slightly larger than the one after bilinear interpolation. Therefore, it can be said that overall performance of bicubic interpolation is higher.
- In images, having high contrast and noised with a low variance gaussian noise, the improvement of interpolation occurs only at the flat surfaces. For these images, bicubic interpolation smoothes the high intensity differences at edges more than the bilinear interpolation does, which results in edge blurring.
- When the contrast gets lower and variance of noise increases both flat surfaces and edges are improved. For those images, the performance of bicubic interpolation is higher both at flat surfaces and edges.
- The computational complexity of bicubic interpolation is significantly higher than bilinear interpolation. Increase in performance does not seem to be worth that computational complexity.

Salt & pepper noise:

- Neither of the methods is good in small amounts of salt & pepper noise. Error increases after interpolations, because while smoothening noise, interpolation also affects the uncorrupted samples. Nonlinear interpolation techniques could perform better.
- In large amount of salt & pepper noise (>= 25%), both interpolations improved the noisy images. Total error in image was decreased by both methods.
- When interpolations have enhancing effect, i.e. noise is >=25%, overall performance of bicubic interpolation is higher. However, its disadvantages (high computational effort and edge smoothening) stand still.
- Noisy images with low contrast improved more when compared to the ones having high contrast.

Image upsampling:

- Bicubic interpolation smoothes the sharp edges and it does not provide any advantage in flat surfaces when compared to bilinear interpolation. If the smoothing effect visually enhances the image as in the rotated checkerboard case, bicubic interpolation may be used. Otherwise, bilinear interpolation should be preferred to preserve sharp edges.
- In noisy image upsampling, bicubic interpolation has the advantage of smoothing the noises in flat surfaces, however, disadvantage of blurring edges remains.
- Nearest neighbor may only be preferred for synthetic images with sharp orthogonal edges, which is a very rare case. It causes the 'ladder' effect (step by step blurring) in photographic images and does not provide an appealing view.
- Cubic spline method performs good job when passing the edges, however it causes undulation on the flat surfaces for a period after passing the edge.

Overall conclusion would be that there is not a considerable visual difference between images obtained by bilinear, bicubic and spline methods and none of them could provide a direct solution to the blurring problem defined in 4.1.2. When their overall efficiencies and computational efforts are considered, bilinear interpolation should be preferred and another solution to the blurring problem should be searched.

5.3 Feature-Based Image Enhancement Techniques

Observing the effects of standard interpolation methods we have concluded that the feature-based techniques have higher potential to overcome the blurring problem encountered in the panoramic images.

5.3.1 Detecting High Intensity Difference and Sharpening the Edge Pass

In this method, the panoramic image is scanned horizontally and vertically and the pixel intensity values are recorded for a predefined array length (*e.g.* 12 pixels). If the intensity difference along this array exceeds a certain threshold (*e.g.* 64 gray levels), then the pixel values are reorganized so that the edge pass length will be shorter. Process is visualized in the below Figure 5.31. The algorithm implemented using Java is given in Appendix C.

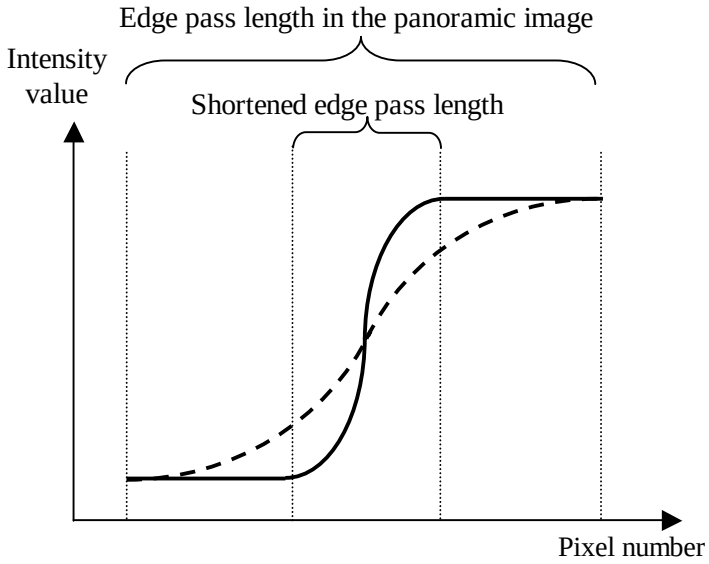

Figure 5.31: Pixel intensity values in the panoramic image along the edge pass
– – – ; in initial panoramic image ——— ; in sharpened image

For a cylindrical panorama, the output of the algorithm is compared with the original panoramic image (Figures 5.32 and 5.33). The result is not satisfactory. It is hard to define an array length and a threshold value that works for the whole image. We included some modifications of the main algorithm to overcome this problem. In

particular, we changed the buffer size as we move in the panorama in the vertical direction. As the sampling rate gets lower the buffer size is increased to account for increased edge-pass length in the panorama. We also checked the intensity values in the buffer and modified the edge-pass shortening if these values do not increase or decrease regularly (for example if the intensity increases rapidly first and then goes same within the defined buffer size).

Another observed problem is the following: the decision for shortening the edge pass is given at each row independently causing edge shifts. As a result, the boundaries of the features in the image are sometimes ribbed. This effect increases when the boundaries of the features in the panoramic image are not straight or distorted during the conversion from the omnidirectional image.

Algorithm is also applied to the spherical panorama (Figures 5.34 and 5.35) of the same omnidirectional image, which was used for Figures 5.32 and 5.33. As explained in Section 4.2.3, in spherical panoramas vertical distribution of the pixels is angle based similar to the human eye. Thus panoramas generated by spherical model are more condensed and appealing than the cylindrical model. On the other hand, horizontal widening for the regions towards the center of the ODI is a problem that is not yet solved. In fact, it can be said that blurring in spherical panoramas is caused by horizontal upsampling and for spherical panoramas only horizontal sharpening has importance. However, as can be observed in Figures 5.34 and 5.35, the problems mentioned in the previous paragraph remain.

We also tried sharpening the ODI before generating the panoramic image. However, because the edge pass length in the ODI is already about two-three pixels sharpening the ODI was not effective.

Figure 5.32: A flu region of a cylindrical panorama (bilinear interpolation)

Figure 5.33: Sharpening algorithm is applied to the cylindrical panorama in Figure 5.32 vertical buffer size=8, horizontal buffersize=8; vertical threshold=60; horizontal threshold=70

Figure 5.34: A flu region of a spherical panorama (bilinear interpolation)

Figure 5.35: Sharpening algorithm is applied to the spherical panorama in Figure 5.34 vertical buffer size=6, horizontal buffersize=8; vertical threshold=60; horizontal threshold=70

5.3.2 Edge Detection-Based Interpolation

5.3.2.1 Previous Work on Feature-Based Interpolation Methods

An orientation adaptive technique is explained by Gerek and Cetin [62]. In this technique, for every 4-pixel neighborhood first the intensity value differences in horizontal, vertical and diagonal directions are calculated and the direction with the least intensity difference is determined. Then the image is upsampled and the intensity values of the new pixels are defined as the average of the pixel values in the direction found in the previous step.

In another technique proposed by Hong et al. [63], again the horizontal, vertical and diagonal edge prototypes are employed and pixels used for the calculation of the new pixel values are selected in this manner as in [62]. However, this time the values of

new pixels are not determined by arithmetic mean but by utilizing bilinear interpolation. In the 4-pixel neighborhood, for the intensity values of the non-selected pixels, values of the selected pixels are re-employed to fulfill the requirement of bilinear interpolation method. Finally, Gaussian low pass filters are applied to eliminate the blocking artifacts and discontinuities in the image.

Jiang and Moloney [64] proposed another direction adaptive interpolation method in which the weights of the neighboring pixels in the calculation of new pixel's intensity are defined dependent to the gradient angle. When the intensity gradient is high in a certain direction interpolation of pixels in that particular direction is avoided, whereas pixels in the orthogonal direction are more heavily interpolated.

In another feature-based method described in Ramanarayanan et al. [65] after the edges are detected they are represented by cubic splines and the image is divided into regions according to this information. Then, bilinear interpolation is employed using only the pixels in the same region.

5.3.2.2 Developed and Tested Edge Detection-Based Methods

We have developed a technique similar to [65] with the difference that detected edges are not combined with splines. Our implementation is based on Canny edge detection algorithm, which compares the resultant gradient vector with a threshold to define an edge. It also uses a lower threshold value. Gradients greater than this lower threshold are defined as edges only if they are neighbors of the edges of primary threshold. The directions of the detected edges are recorded as well.

In this technique, standard bilinear interpolation is used unless an edge is detected in the selected 4-pixel neighborhood. In the mentioned gradient-based interpolation techniques [62-64] however, the developed algorithm is used for the whole image regardless of the occurrence of an edge. For the edge detected pixel neighborhoods, the four corners used for the interpolation are selected or unselected according to the directions in which edges are located, which gives us 16 different edge schemes. Some examples to the edge schemes are given in Figure 5.36. To fulfill the requirement of bilinear interpolation method, values of the selected pixels are re-employed instead of the intensity values of the non-selected pixels in the 4-pixel

neighborhood. The developed algorithm is implemented in Java and works well on synthetic images (Figure 5.37).

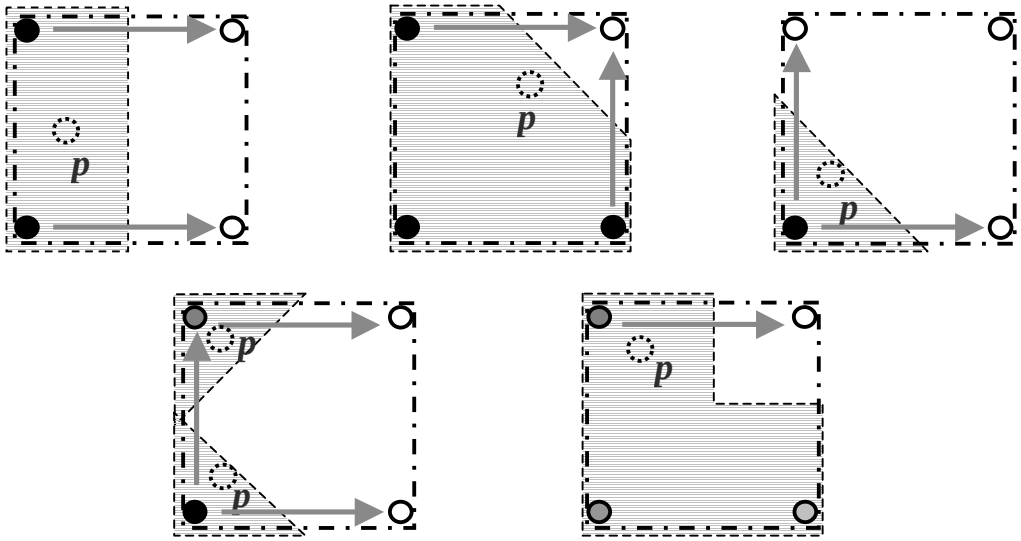

Figure 5.36: Example schemes for selecting the pixels for interpolation, shaded areas: the regions that are interpolated using pixels only in the area, arrows: the edges (high intensity differences), ***p***: the location of which the intensity value to be determined

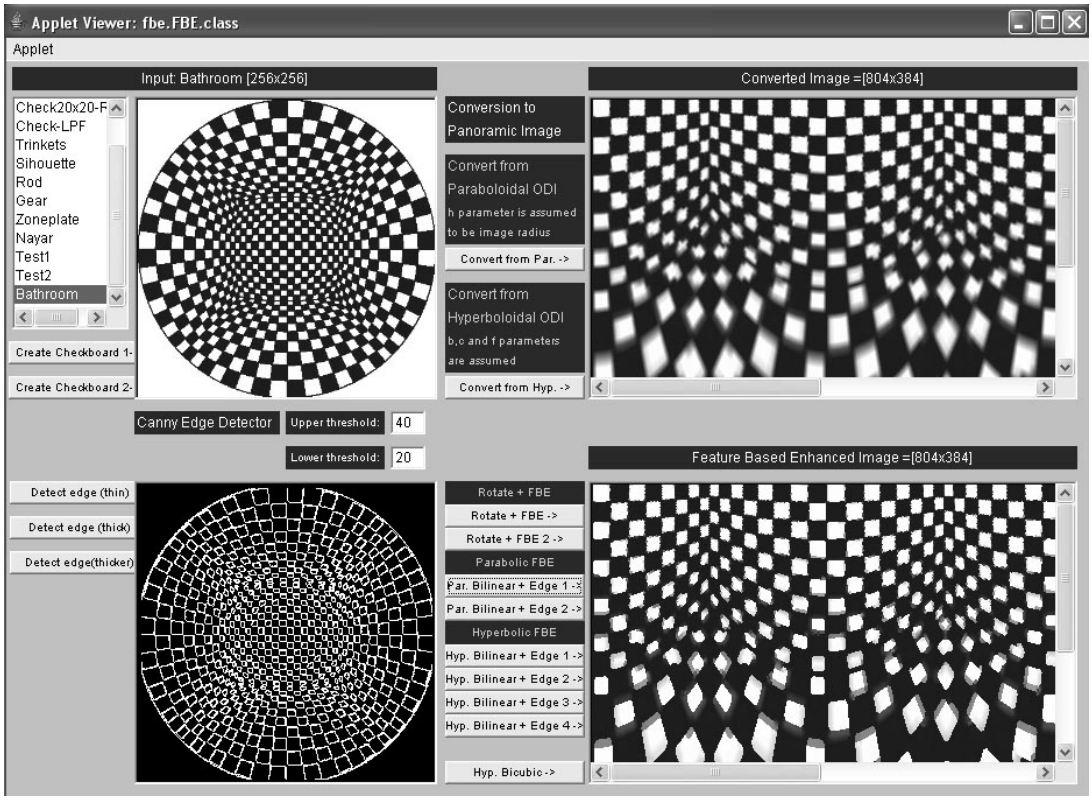

Figure 5.37: A snapshot of the Java application constructed to implement the edge detection-based enhancement algorithm (up-left: input ODI, up-right: bilinear, down-left: detected edges, down-right: enhanced)

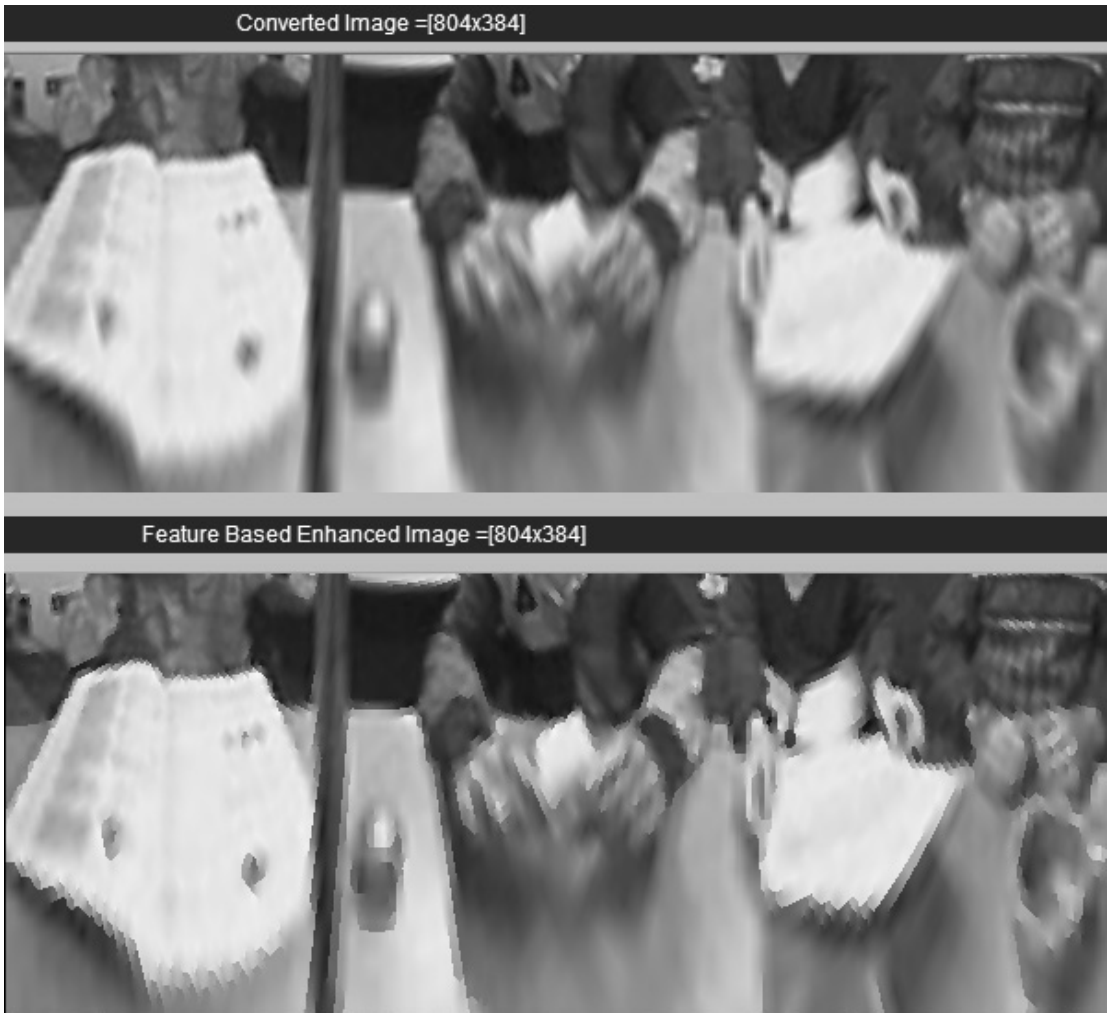

Figure 5.38: Visual result of the edge detection-based enhancement algorithm
(top: panoramic image generated by standard bilinear interpolation,
bottom: panoramic image generated by feature-based method)

However, for photographic omnidirectional images the results are not satisfactory (Figure 5.38). The main reasons are:

1) When the detected edges are not orthogonal or 45º inclined, because of the pixel leaps along the edge, the zig-zags are observed in the resultant image (Figure 5.39). When the images are upsampled these zig-zags get larger. In the Ramanarayanan's method, simple synthetic images are used and edges are modeled with splines. However, it is difficult to generate spline edges in photographic images.

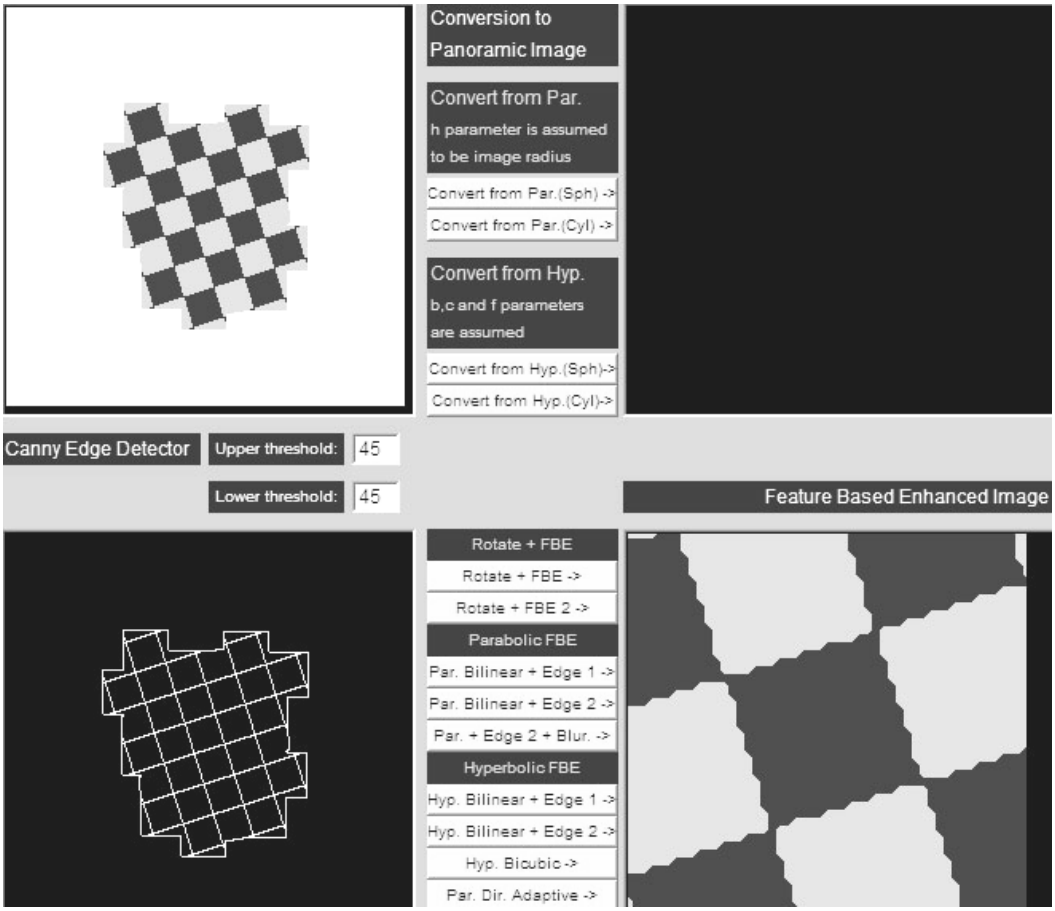

Figure 5.39: A snapshot from the edge detection-based enhancement Java application. 18º inclined checkerboard is upsampled to test the algorithm (up-left: inclined checkerboard, down-left: detected edges, down-right: upsampled & enhanced checkerboard)

2) When the edge pass is not sharp (such as different gray levels/tones) the determined edge thickness may not be one pixel (Figure 5.40). Sometimes, in order to catch the intensity differences in all directions we need 'thick' edges. As a result, such multi-layer edges result in thick layers in the resultant image and a thick and multi-layer transition occurs.

In Figure 5.41, there exist thin lines of gray pixels at the intersections of white and black regions in the input image (up-left). Some edges detected by edge detection algorithm are double-layered (down-left). This is the reason of the gray areas in some regions of the output (panoramic) image (down-right). In Figure 5.38, effects of this multi-layer phenomenon on the photographic panoramic images can be observed. Thick-layered transitions are observed because of the detected multi-layer edges.

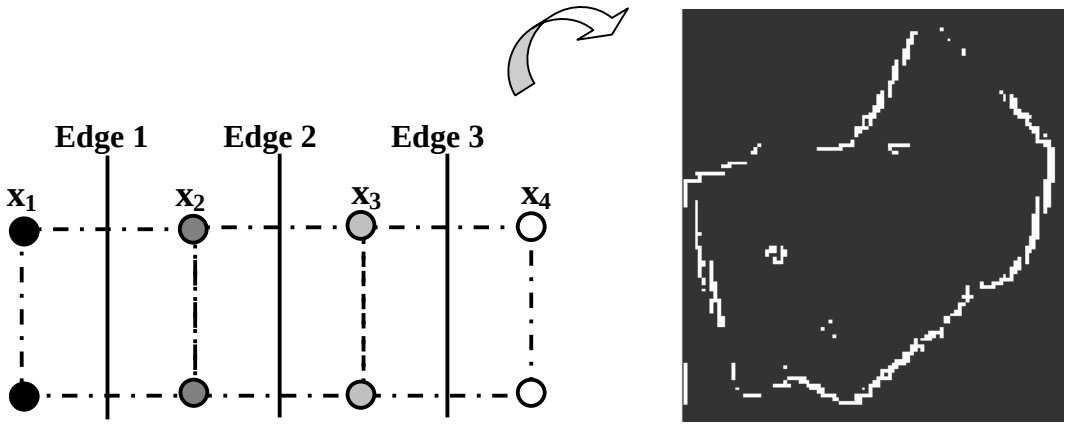

Figure 5.40: Multi-layer edge detection phenomenon, left: different gray levels neighboring in the ODI are represented, right: detected multi-layer edges

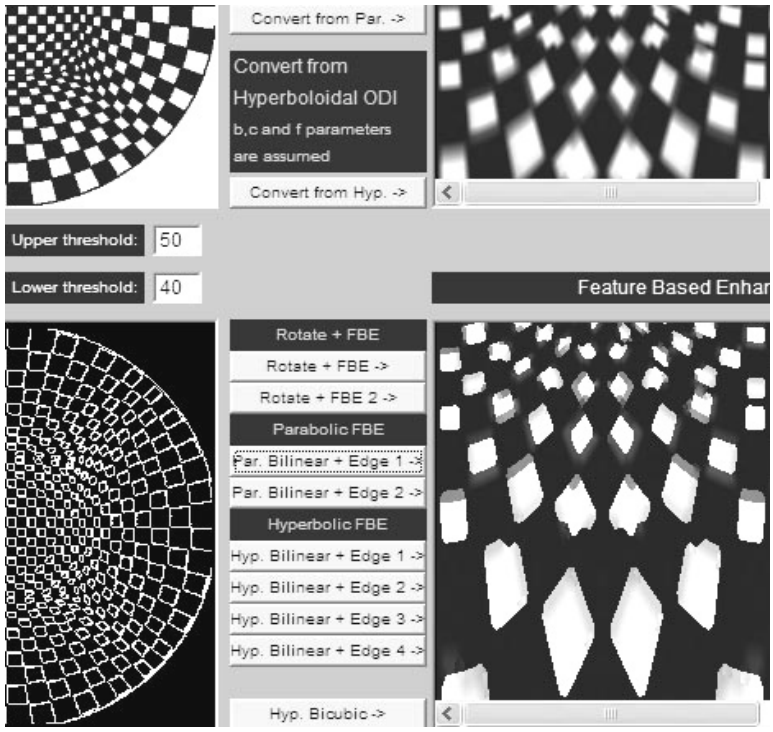

Figure 5.41: A snapshot from the edge detection-based enhancement Java application. A synthetic ODI is input. (up-left: input ODI, down-left: detected multiplayer edges, down-right: enhanced panoramic image)

Even though we decrease the detected edge pass length to one pixel in the edge detection algorithm, because the actual edge will be thicker the rest of the area will remain blurred. As an example, in Figure 5.42 the detected edges (down-left) are thin when compared to the edge detection used for Figure 5.38. For this reason, in the panoramic image at down-right of Figure 5.42 sharpened partially. Sharpening at the edge of the book is not continuous.

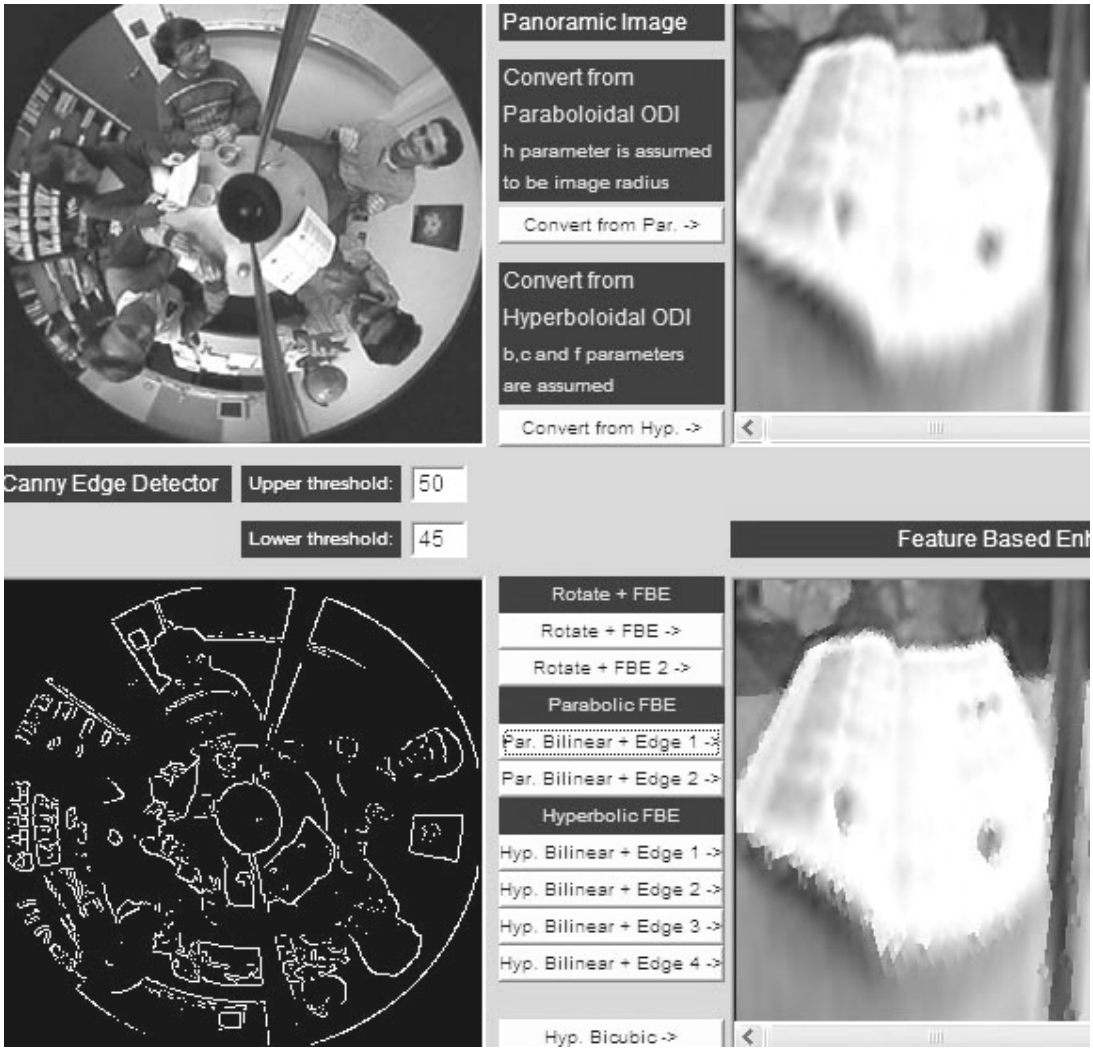

Figure 5.42: A snapshot from the edge detection-based enhancement Java application. (up-left: input ODI, up-right: panoramic image generated by standard bilinear interpolation, down-left: detected edges, down-right: panoramic image generated by feature-based method)

We added blurring to the enhancement algorithm and obtain a more appealing view (Figure 5.43) when compared with the result in Figure 5.38.

We also tried our algorithm with a spherical panoramic image (Figure 5.44). As the vertical distribution is more condensed in the spherical panoramas, enhancement is needed mainly in the horizontal direction. Blurring added resultant image is given in Figure 5.45.

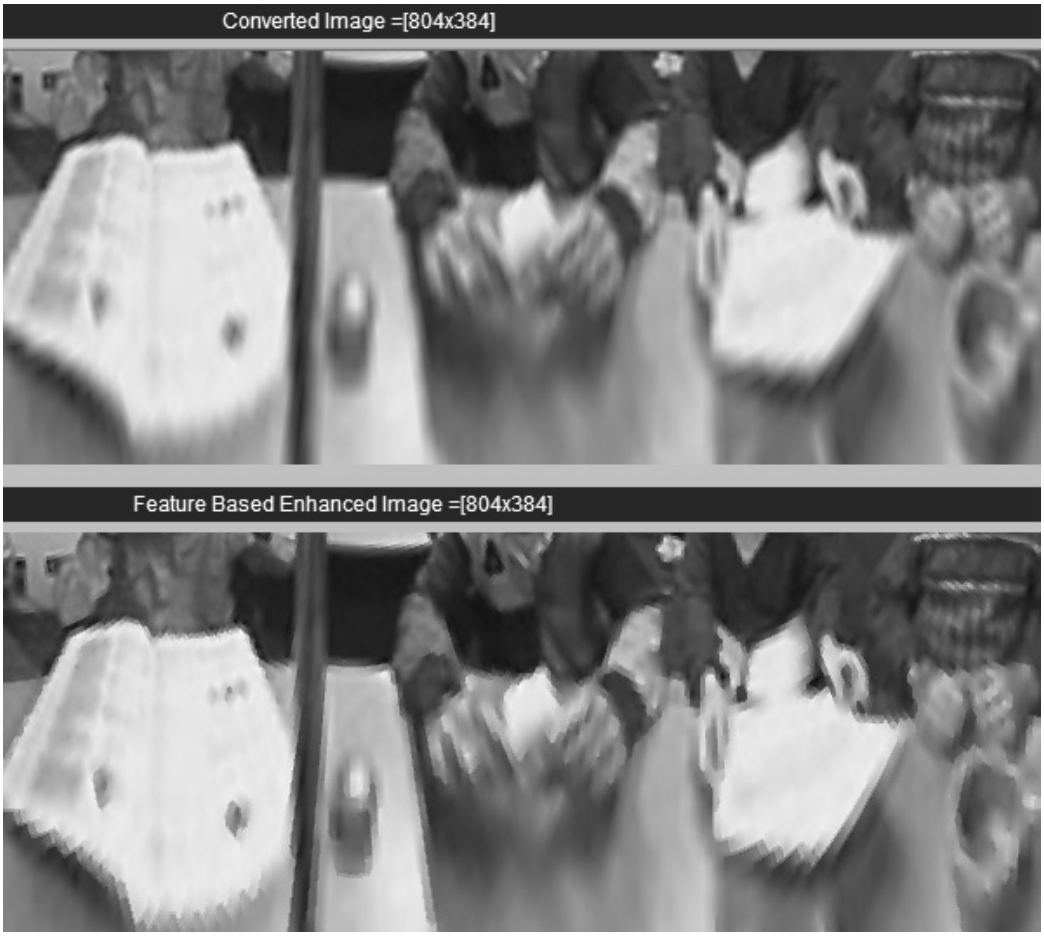

Figure 5.43: Blurring added to the enhancement algorithm,
to be compared with the image at the bottom of Figure 5.38

We have tested the method described in [62] for our problem. In this method, for every 4-pixel neighborhood, gradient vectors in horizontal, vertical and diagonal directions are calculated and direction in which the least intensity difference occurred is recorded. Intensity values of the pixels in the generated panoramic image are defined as the average of the pixels on the same side of the possible edge (Figure 5.46), i.e. pixels in the recorded direction. When we look at the resultant image (Figure 5.47), we observe diamond-like boxes as if it is crystallized. The boundaries of these boxes are very sharp. Because the intensity value of the generated pixel is given as the arithmetic mean of the two neighboring pixels, the calculated intensity value does not change until the next neighborhood where the intensity value changes suddenly. Together with the upsampling, this phenomenon shows itself as constant-intensity value regions.

Figure 5.44: Result of the edge detection-based enhancement for spherical panorama
(top: spherical panoramic image generated by standard bilinear interpolation,
bottom: spherical panoramic image generated by feature-based method)

Figure 5.45: Blurring added to the enhancement algorithm,
to be compared with the image at the bottom of Figure 5.44

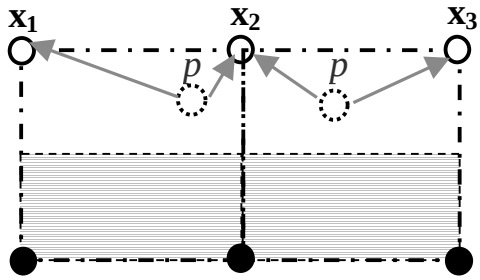

Figure 5.46: Interpolation scheme of the method described in [62], $I[p] = (I[x_1]+I[x_2]) / 2$,
p: the location of which the intensity value to be determined,
shaded area: the region that is interpolated using black pixels

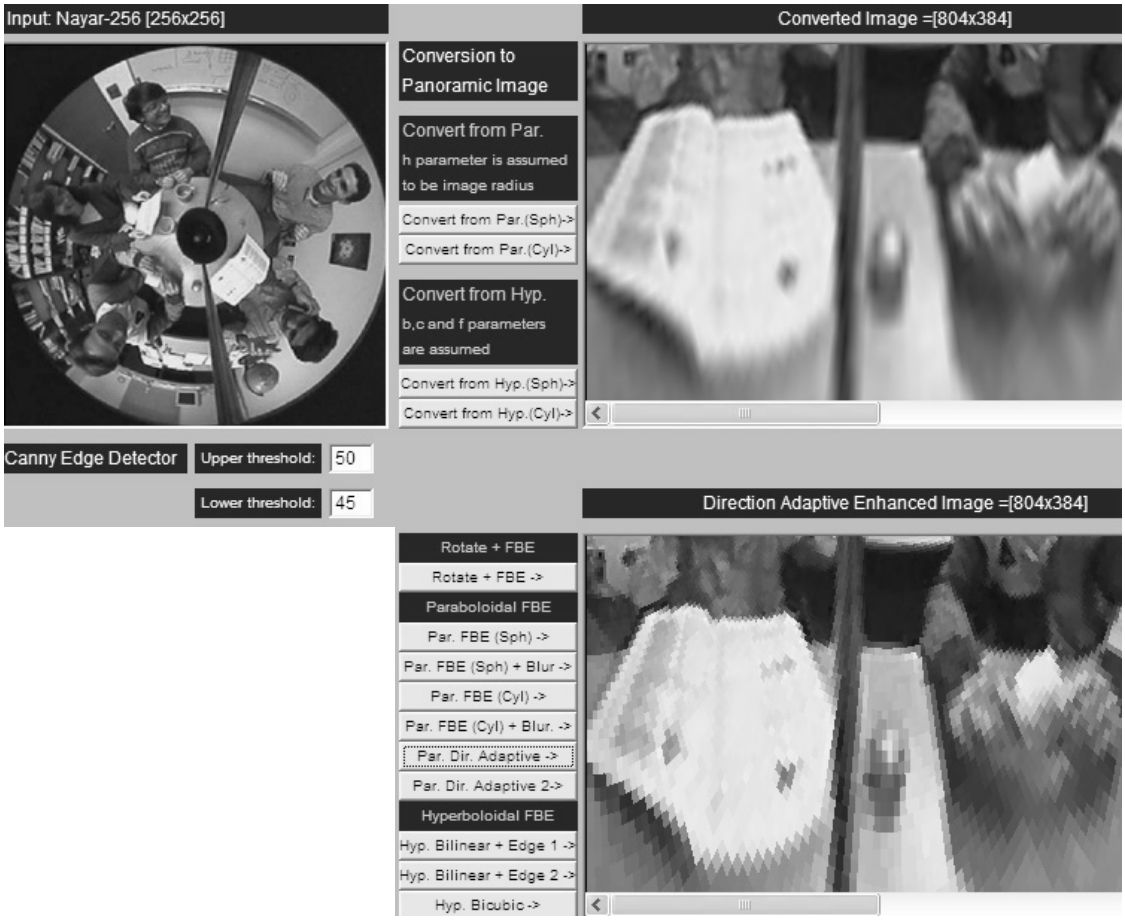

Figure 5.47: A snapshot of the Java application constructed to test the direction-adaptive interpolation technique described in [62]

To overcome the sudden pixel value changes, after determining edge orientations we used a modified bilinear interpolation as explained in [63]. In bilinear interpolation the distances to the corners of the interpolation mask are included in the calculation, therefore sudden pixel value changes are decreased. To fulfill the requirement of bilinear interpolation method, values of the selected pixels are re-employed instead of the intensity values of the non-selected pixels in the 4-pixel neighborhood. A small amount of blurring is also added to the algorithm by smoothing the transitions between different calculation regions within the same 4-pixel neighborhood.

The result is given in Figure 5.48. Sharpness between the different intensity value areas is decreased but could not be totally alleviated because there still occurs considerable intensity changes between the adjacent interpolation areas and the

different calculation regions within one area. Result of the same algorithm on the spherical panoramic image is given in Figure 5.49.

More successful results are not expected with this algorithm since only four types of edges (vertical, horizontal and diagonals) are employed. The algorithm that we presented previously is able to define more edge schemes and only needs to check the edge direction in the regions of detected edges.

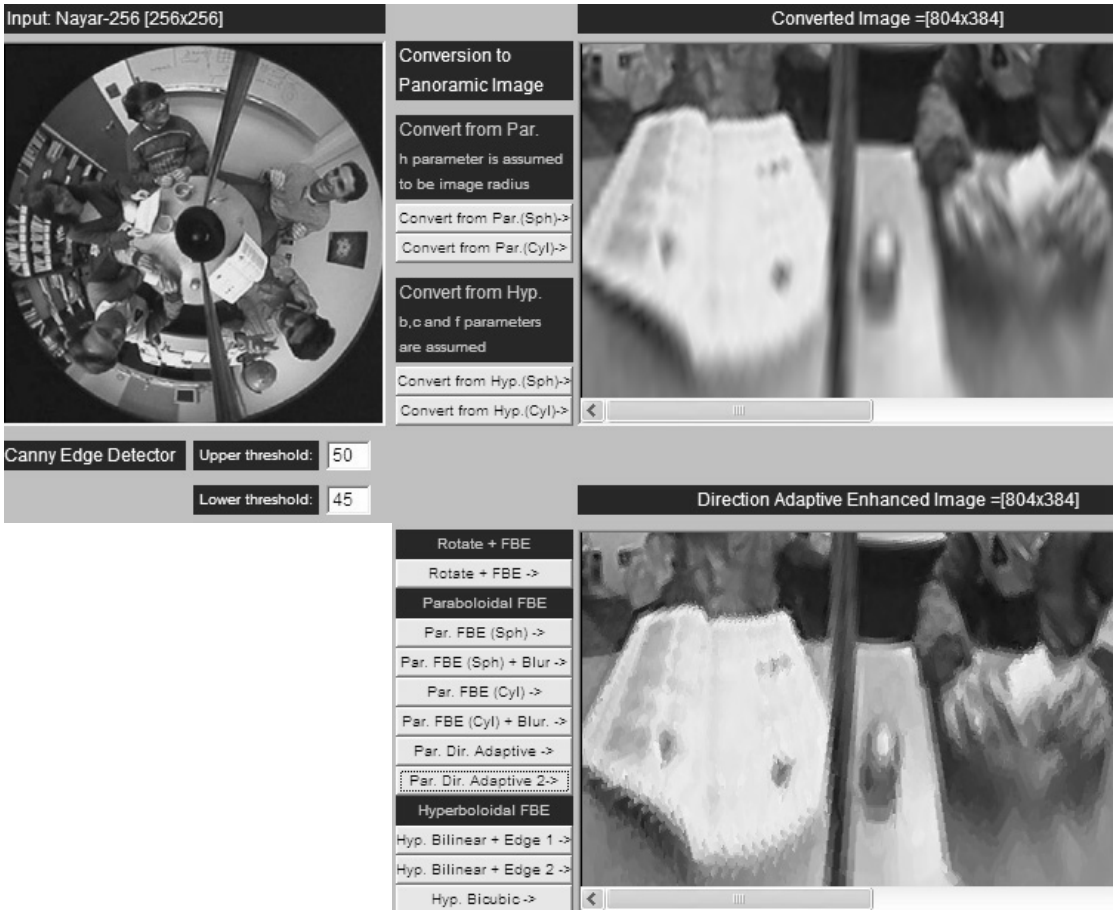

Figure 5.48: A snapshot of the Java application constructed to test a direction-adaptive interpolation similar to the one explained in [63]

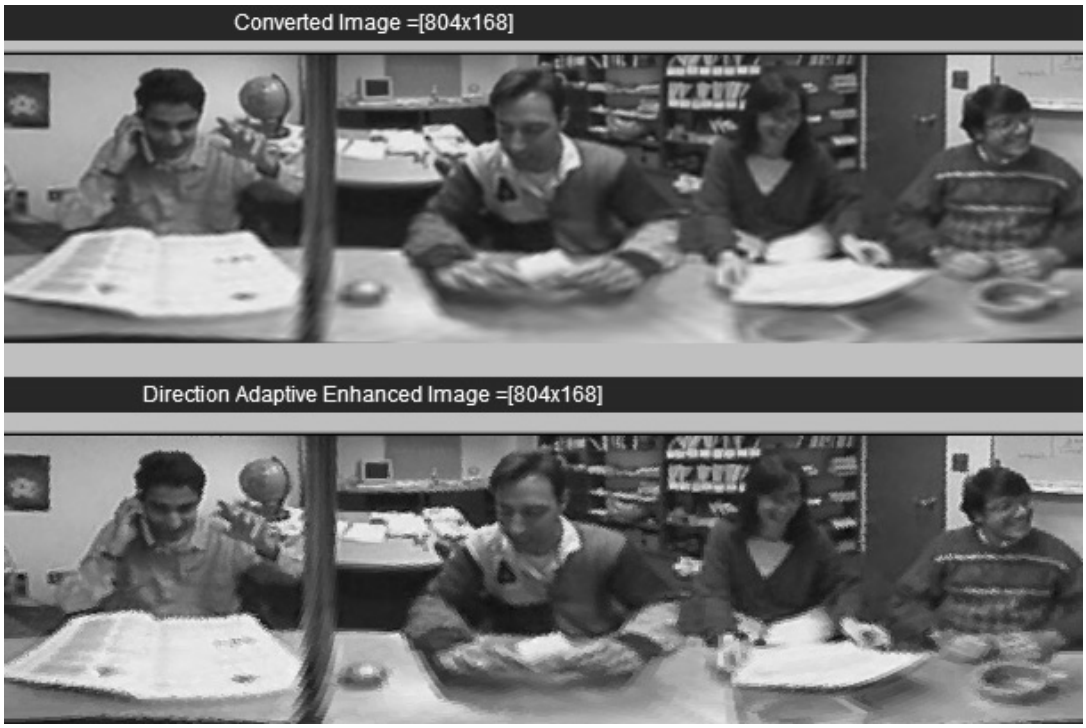

Figure 5.49: Result of the direction-adaptive interpolation for spherical panorama, to be compared with Figure 5.48

5.3.2.3 Conclusions

We observe thick layers in the resultant panoramic image due to the upsampling applied during image generation. This problem could be solved by considering the edges globally rather than locally. When the edges are described only by local information in a neighborhood of four pixels the edge orientations and locations are quantized to very few levels. If we were to consider a wider area, better representation of the edge would be obtained at the expense of higher computational load.

At the lower parts of the panoramic image considerable amount of upsampling (approx. 8-10 times) occurs. In the previous work examined, research is made with images upsampled 2 to 5 times. Therefore, these artifacts are not as annoying as in our experiments.

When the detected edges are not orthogonal or 45º inclined, the zig-zags are observed in the resultant image due to the pixel leaps along the edge because edge

can not be defined as a perfect line. Together with the upsampling these zig-zags are amplified. Using splines while determining the edges would be a solution to this problem as explained in [65]. However, it is difficult to keep track of control points in large photographic images, as there will inevitably be multiple edges. The examples presented in [65] were synthetic images.

Lastly, edge transitions typically do not occur in single pixels in photographic images. They rather manifest themselves as a continuum of gray levels, resulting in multiple layers in the upsampled image. Thinning the edges to a one-pixel width would not solve the problem because the neighborhood of the edge will remain blurred.

Using spherical panoramic images instead of cylindrical ones as input improves the result automatically since the spherical panoramas are more condensed in vertical direction and only horizontal direction needs improvement.

As a conclusion, we developed and tested several techniques for our blurring problem. Because of the high upsampling rate and information loss around the center and it becomes very difficult to determine where the actual edge is located. We believe that more modifications in the mentioned techniques will not be able to improve the result further. Hybrid methods combining different techniques (including the mentioned ones) or knowledge-based methods like using the knowledge of edges in the image such as tables and shelves may be able to enhance the images further.

CHAPTER 6

Conclusions

In this thesis, catadioptric omnidirectional imaging systems are analyzed from different points of view. Application areas and studies on such imaging systems are investigated and state-of-the-art is summarized.

Geometrical properties of common mirror types used in catadioptric systems and image formation through these systems are examined in Chapter 2. Geometrical relations for panoramic and perspective image generation with different mirror types are summarized. One can easily implement viewing applications of described systems using the provided information.

6.1 Findings and Application Areas

Reflections of paraboloidal, hyperboloidal and spherical mirrors are simulated and the omnidirectional images gathered through these mirror types are compared. Paraboloidal mirror's surface equation has one parameter and light rays targeting the mirror focal point are reflected orthogonally and collected by a telecentric lens. These two facts conclude that ODIs taken with different paraboloidal mirrors resemble each other by a scale factor. Hyperboloidal mirror's surface equation has two parameters. Thus, unlike the paraboloidal mirror both scale and curvature may change by manipulating these two parameters. Spherical mirror has one parameter (the radius), which changes the size of the mirror. But ODIs taken with different spherical mirrors do not resemble each other by a scale factor because the distance between the mirror and the camera affects the resultant image as well.

A method is developed to determine the unknown parameters of a system with a hyperboloidal mirror using the world coordinates of a set of objects and their corresponding image coordinates on the ODI. A linear relation between the parameters of the hyperboloidal mirror (b and c) is determined. It is observed that as long as the b/c ratio is conserved, changing b or c does not affect the reflection. This ratio is used as one parameter for the optimizations. We are able to estimate the b/c ratio and camera focal length (f) with a performance depending on the noise of image point readings and the number of points used for the optimization.

We also tried our parameter extraction method with a calibration pattern. The 3D coordinates of a corner of this calibration pattern are estimated. We assumed that the alignment of the pattern is already known to express the coordinates of all points in the pattern in terms of the reference corner point.

As explained in the first chapter, applications of catadioptric omnidirectional cameras are spreading and systems with more than one mirror or multi-camera systems are already being used. Using such systems requires a well-settled knowledge on the calibration issues. Conducted research and findings in this thesis are instrumental for calibration of such imaging systems.

In Chapter 4, resolutional properties of the panoramic images (panoramas) that are generated from the omnidirectional images are examined. In Section 4.1, the blurring problem encountered in panoramic projections is defined and the effect of hyperboloidal, paraboloidal and spherical mirror profiles on this problem is analyzed.

Different projection surface alternatives for the panoramic projections are evaluated as well. By modifying the projection surface we can manipulate the panoramic images. Cylindrical projection gives an unappealing view for the objects imaged towards the center of the ODI. Conical and spherical projections create appealing panoramic images since they generate realistic vertical representations. It can be said that, spherical projection should be preferred since it has an angle-based vertical representation like regular cameras; therefore it suits our eyes best. However, most of the current panoramic imaging applications use cylindrical projection. It has lower

computational load and when a high cylindrical representation is not especially desired the low-quality regions can be cut from the panorama.

Concerning the enhancement of panoramic images, the effects of standard interpolation methods are evaluated first. The bilinear interpolation, bicubic interpolation, nearest neighbor and cubic spline methods are compared by a series of computations performed on checkerboards, having different cell size and contrast. These images are exposed to gaussian and salt & pepper noises and enlarged to implement the upsampling effect. Panoramic images generated by these methods are also created from omnidirectional images to examine the visual differences. Results showed that there is not a considerable visual difference between images obtained by bilinear, bicubic and spline methods and none of them could provide a direct solution to the 'blurring problem'. When their overall efficiencies and computational efforts are considered, bilinear interpolation should be preferred and another solution to the blurring problem should be searched.

To enhance the panoramic images blurred due to the high upsampling rate, several feature-based techniques are developed and tested. One major problem observed is the zig-zags due to the pixel leaps along the edge because edge can not be defined as a perfect line. Using splines while determining the edges would be a solution to this problem, however, it is difficult to keep track of control points in large photographic images, as there will inevitably be multiple edges. The second major problem can be defined as the edge transitions typically do not occur in a single pixel in photographic images. They rather manifest themselves as a continuum of gray levels, resulting in multiple layers in the upsampled image.

Using spherical panoramic images instead of cylindrical ones as input improves the result automatically since the spherical panoramas are more condensed in vertical direction and only horizontal direction needs improvement.

Enhancing the panoramic images can be useful for application areas based on presentation and display purposes. Appealing panoramas could be obtained with fewer images and/or less computational cost.

6.2 Limitations and Future Work

The parameter extraction method for the systems with a hyperboloidal mirror is developed with simulated reflections from a hyperboloidal mirror. It does not include all parameters of a real imaging system. In a real system, additional parameters such as sensor aspect ratio, image skew and lens distortions should be determined. Therefore the presented method may not work as expected and testing it for a real omnidirectional system is an important part of future work.

In the part where optimization is performed with a calibration pattern, it is assumed that the alignment of the pattern is known. An extention to this may be using a calibration pattern alignment of which is not known. We also observed that the size, shape and location of the calibration pattern may affect the results. A calibration pattern especially for omnidirectional cameras can be designed in the future.

Developed and evaluated edge-detection based techniques for panoramic image enhancement are not very successful, but they can constitute a base for future work. Because of the information loss due to the high upsampling rate in the panoramic images, it becomes very difficult to determine where the actual edge is located. We believe that more modifications in the mentioned techniques will not be able to improve the results significantly. Hybrid methods combining different techniques (including the mentioned ones) or knowledge-based methods like using the knowledge of edges in the image such as tables and shelves may be able to enhance the images further.

Catadioptric imaging is a new area and standard image processing techniques are often inadequate for these systems. Specialized algorithms have to be developed for catadioptric omnidirectional imaging systems.

REFERENCES

[1] Benosman, R., Kang, S.B. (2001). Panoramic Vision: Sensors, Theory and Applications, Springer-Verlag, New York.

[2] Fermüller, C., Aloimonos, Y., Baker, P., Pless, R., Neumann, J., Stuart, B. (2000). Multi-Camera Networks: Eyes from Eyes, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2000.

[3] Chen, S.E. (1995). QuickTime VR - An Image-Based Approach to Virtual Environment Navigation, Computer Graphics: Proc. of SIGGRAPH 95, 29-38.

[4] Foote, J., Kimber, D. (2000). The Flycam System: Practical Panoramic Video and Automatic Camera Control, Proc. of IEEE Int. Conf. on Multimedia and Expo (ICME) 2000.

[5] Ikeda, S. (1999). Telepresence Using an Omnidirectional Movie, Proc. of the First Int. Symposium on Mixed Reality.

[6] Kim, S., Chang, E.Y., Ahn, C.H., Woo, W. (2003). Image-Based Panoramic 3D Virtual Environment Using Rotating Two Multi-view Cameras. Proc. of International Conference on Image Processing (ICIP) 2003, 917-920.

[7] Rees, D.W. (1970). Panoramic Television Viewing System, US Patent No. 3505465.

[8] Oh, S.J., Ahn, D.J., Hall, E.L. (1988). A Wide Angle Vision System for Mobile Robotics, Proc. of IEEE Int. Sym. on Intelligent Control 1988.

[9] Svoboda, T., Pajdla, T., Hlavac, V. (1998). Epipolar Geometry for Panoramic Cameras, Proc. of European Conference on Computer Vision (ECCV) 1998.

[10] Baker, S., Nayar, S.K. (1999). A Theory of Single-Viewpoint Catadioptric Image Formation, Int. J. of Computer Vision, 35(2), 175–196.

[11] Bruckstein, A.M., Richardson, T.J. (1996). Omniview Cameras with Curved Surface Mirrors, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2000.

[12] Bogner, S.L. (1995). An Introduction to Panoramic Imaging, Proc. of IEEE Conference on Systems, Man and Cybernetics 1995, vol.4, 3099-3106.

[13] Swaminathan, R., Grossberg, M.D., Nayar, S.K. (2001). Non Single-viewpoint Catadioptric Cameras, Technical Report, Department of Computer Science, Columbia University.

[14] Yagi, Y., Kawato, S., Tsuji, S. (1994). Realtime Omnidirectional Image Sensor (COPIS) for Vision-Guided Navigation, IEEE Transactions on Robotics and Automation, vol.10, no:1.

[15] Yamazawa, K., Yagi, Y., Yachida, M. (1993). Omnidirectional Imaging with Hyperboloidal Projection, Proc. of IEEE/JRS Int. Conf. on Intelligent Robots and Systems.

[16] Marchese, F.M., Sorrenti, D.G. (2002). Mirror Design of a Prescribed Accuracy Omni-directional Vision System, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2002.

[17] Hicks, R.A., Bajcsy, R. (2000). Catadioptric Sensors that Approximate Wide-Angle Perspective Projections, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2000.

[18] Ishiguro, H. (1998). Development of Low-cost Compact Omnidirectional Vision Sensors, Proc. of the International Conference on Information Systems, Analysis and Synthesis 1998, 433–439.

[19] Hicks, R.A., Perline, R.K. (2002). Equi-areal Catadioptric Sensors, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2002.

[20] Gaspar, J., Deccó, C., Okamoto J., Santos-Victor, J. (2002). Constant Resolution Omnidirectional Cameras, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2002.

[21] Nayar, S.K., Peri, V. (1999). Folded Catadioptric Cameras, Technical Report, Dept. of Computer Science, Columbia University.

[22] Nayar, S.K. (1997). Catadioptric Omnidirectional Camera, Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 1997.

[23] Srinivasan, M.V. (2003). New Class of Mirrors for Wide-Angle Imaging, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2003.

[24] Benosman, R., Deforas, E., Devars, J. (2000). A New Catadioptric Sensor for the Panoramic Vision of Mobile Robots, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2000.

[25] Conroy, T., Moore, J. (1999). Resolution Invariant Surfaces for Panoramic Vision Systems, Proc. of Int. Conf. on Computer Vision (ICCV) 1999.

[26] Swaminathan, R., Grossberg, M.D., Nayar, S.K. (2004). Designing Mirrors for Catadioptric Systems that Minimize Image Errors, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2004.

[27] Geyer, C., Daniilidis, K. (1999). Catadioptric Camera Calibration, Proc. of Int. Conf. on Computer Vision (ICCV) 1999.

[28] Geyer, C., Daniilidis, K. (2002). Paracatadioptric Camera Calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, May 2002.

[29] Geyer, C., Daniilidis, K. (2000). Equivalence of Catadioptric Projections and Mappings of the Sphere, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2000.

[30] Kang, S. (2000). Catadioptric Self-calibration, Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2000.

[31] Orghidan, R., Salvi, J., Mouaddib, M. (2003). Calibration of a Structured Light-Based Stereo Catadioptric Sensor, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2003.

[32] Cauchois, C., Brassart, E., Drocourt, C. (1999). Calibration of the Omnidirectional Vision Sensor: SYCLOP, Proc. of Int. Conf. on Robotics and Automation 1999.

[33] Derrien, S., Konolige, K. (2000). Approximating a Single Viewpoint in Panoramic Imaging Devices, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2000.

[34] Fiala, M., Basu, A. (2002). Feature Extraction and Calibration for Stereo Reconstruction Using Non-SVP Optics in a Panoramic Stereo-vision Sensor, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2002.

[35] Goncalves, N., Araujo, H. (2003). Mirror Shape Recovery from Image Curves and Intrinsic Parameters, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2003.

[36] Fabrizio, J., Tarel, J.P., Benosman, R. (2002). Calibration of Panoramic Catadioptric Sensors Made Easier, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2002.

[37] Yamazawa, K., Yagi, Y., Yachida, M. (1995). Obstacle Detection with Omnidirectional Image Sensor: HyperomniVision, Proc. of Int. Conf. on Robotics and Automation 1995.

[38] Onoe, Y., Yamazawa, K., Takemura, H., Yokoya, N. (1998), Telepresence by Real-time View-Dependent Image Generation from Omnidirectional Video Streams, Journal of Computer Vision and Image Understanding, v.71.

[39] Nagahara, H., Yagi Y., Yachida, M. (2003). Wide Field of View Head Mounted Display for Tele-presence with an Omnidirectional Image Sensor, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2003.

[40] Pintaric, T., Neumann, U., Rizzo, A. (2000). Immersive Panoramic Video, Proc. of the Eighth ACM International Conference on Multimedia 2000.

[41] Chahl, J.S., Srinivasan, M.V. (2000). A Complete Panoramic Vision System: Incorporating Imaging, Ranging, and Three Dimensional Navigation, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2000.

[42] Hong, J., Tan, X., Pinette, B., Weiss, R., Riseman, E.M. (1991). Image-Based Homing, Proc. of Int. Conf. on Robotics and Automation 1991.

[43] Murphy, J.R. (1995). Application of Panoramic Imaging to a Tele-operated Lunar Rover, Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics 1995.

[44] Nene, S.A., Nayar, S.K. (1998). Stereo with Mirrors, Proc. of Int. Conf. on Computer Vision (ICCV) 1998.

[45] Nagahara, H., Yagi Y., Yachida, M. (2000). Super-resolution from an Omnidirectional Image Sequence, Proc. of Int. Conf. on Industrial Electronics, Control, Instrumentation 2000.

[46] Nagahara, H., Yagi Y., Yachida, M. (2001). Resolution Improving Method from Multifocal Omnidirectional Images, Proc. of International Conference on Image Processing (ICIP) 2001, 654-657.

[47] Nagahara, H., Yagi Y., Yachida, M. (2003). Super-resolution Modeling Using an Omnidirectional Image Sensor, IEEE Transactions on Systems, Man and Cybernetics- Part B: Cybernetics, vol.33, no:4, August 2003.

[48] Daniilidis, K., Makadia, A., Bulow, T. (2002). Image Processing in Catadioptric Planes: Spatiotemporal Derivatives and Optical Flow Computation, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2002.

[49] Stratmann, I. (2002). Omnidirectional Imaging and Optical Flow, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2002.

[50] Taylor, C.J. (2000). VideoPlus, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2000.

[51] Doubek, P., Svoboda, T. (2002) Reliable 3D Reconstruction from a Few Catadioptric Images, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2002.

[52] Shimamura, J., Yokoya, N., Takemura, H., Yamazawa, K. (2000). Construction of an Immersive Mixed Environment Using an Omnidirectional Stereo Image Sensor, Proc. of IEEE Workshop on Omnidirectional Vision (OmniVis) 2000.

[53] Ishiguro, H., Ng, K.C., Capella, R., Triverdi, M.M. (2003). Omnidirectional Image Based Modeling, Machine Vision and Applications, v.14.

[54] Gaspar, J., Grossman, E., Santos-Victor, J. (2001). Interactive Reconstruction from an Omnidirectional Image, Proc. of Int. Sym. on Intelligent Robotic Systems 2001.

[55] Biber, P., Andreasson, H., Duckett, T., Schilling, A. (2004). 3D Modeling of Indoor Environments by a Mobile Robot with a Laser Scanner and Panoramic Camera, Proc. of Int. Conference on Intelligent Robots and Systems 2004.

[56] http://www.fullview.com

[57] Peri, V., Nayar, S.K. (1997). Catadioptric Omnidirectional Camera, Proc. of DARPA Image Understanding Workshop, New Orleans, May 1997.

[58] Jain, R., Kasturi, R., Schunck, B.G. (1995). Machine Vision, McGraw-Hill, New York.

[59] Dennis, J. E., Schnabel, R. B. (1983). Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, New Jersey, Chapter 6.

[60] Efford, N. (2000). First-order Interpolation, Digital Image Processing: A Practical Introduction Using Java, Pearson Education Limited, Essex, England, Chapter 9.5.2, 240-241.

[61] Bourke, T. (2001). Bicubic Interpolation for Image Scaling. http://astronomy.swin.edu.au/~pbourke/colour/bicubic/

[62] Gerek, O., Cetin, A. E. (2004). A 2D Orientation Adaptive Prediction Filter in Lifting Structures for Image Coding, Accepted for publication in IEEE Trans. on Image Processing.

[63] Hong, K.P., Paiky, J.K., Kimz, H.J., Leez, C.H., (1996). An Edge-Preserving Image Interpolation System for a Digital Camcorder, Proc. of Int. Conf. on Consumer Electronics, June 1996.

[64] Jiang, H., Moloney, C., (2002). A New Direction Adaptive Scheme for Image Interpolation, Proc. of Int. Conference on Image Processing (ICIP) 2002.

[65] Ramanarayanan, G., Bala, K., Walter, B. (2004). Feature-Based Textures, Proc. of Eurographics Symposium on Rendering 2004.

APPENDICES

APPENDIX A

Derivation of Linear Relationship between Mirror Parameters

We modified Equation 3.1 by writing one of these parameters (b) in terms of the other parameter (c);

$$x = \frac{f_c \cdot (b^2 - c^2) \cdot X}{(b^2 + c^2) \cdot Z - 2bc\sqrt{X^2 + Y^2 + Z^2}} \tag{3.1}$$

$$f_c \cdot (b^2 - c^2) \cdot X - x \cdot ((b^2 + c^2) \cdot Z - 2bc\sqrt{X^2 + Y^2 + Z^2}) = 0$$

$$(f_c \cdot X - z \cdot x) \cdot b^2 + 2xbc\sqrt{X^2 + Y^2 + Z^2} - (f_c \cdot X + z \cdot x) \cdot c^2 = 0$$

$$b = \frac{-2x \cdot c \cdot \sqrt{X^2 + Y^2 + Z^2} \pm \sqrt{4x^2 \cdot c^2 (X^2 + Y^2 + Z^2) + 4c^2 (f_c{}^2 \cdot X^2 - Z^2 \cdot x^2)}}{2(f_c \cdot X - Z \cdot x)}$$

$$b = c \cdot \frac{-x \cdot \sqrt{X^2 + Y^2 + Z^2} \pm \sqrt{x^2 \cdot (X^2 + Y^2 + Z^2) + f_c{}^2 \cdot X^2 - Z^2 \cdot x^2}}{f_c \cdot X - Z \cdot x} \tag{3.3}$$

APPENDIX B

Optimization Algorithm for Parameters k and f

Below subroutine is written in MATLAB and used for the determination of the hyperboloidal mirror parameter k (which is equal to b/c as defined in Section 3.4.4) and camera focal length (f) when a number conjugate pairs (world coordinates X,Y,Z and corresponding image coordinates x,y) are provided. It uses Gauss-Newton method for optimization. Also total squared error value is checked at each step, if error is not decreased step size is made smaller by a scale factor. Calculation functions of total squared error, residual vector and jacobian matrix are given after the optimization subroutine.

gaussnewton.m

```
------------------
% This subroutine is to determine the hyperboloidal mirrored omnidirectional system
% parameters: k and f.
% It uses three other subroutines: totalerror.m, residual.m and jacobian.m

function u = gaussnewton(k,f,X,Y,Z,x,y)
% here, k and f are the initial guesses of unknown parameters
% X,Y,Z,x,y are the real world and image coordinate values of conjugate pairs

u=[k; f];                 % initial values to be iterated
e(1)=totalerror(u(1),u(2),X,Y,Z,x,y);    % error calculation for initial values
eps=1;                    % this variable is used for stop value of iteration loop
i=1;                      % counter for error array
format long;
r=residual(u(1),u(2),X,Y,Z,x,y);     % residual vector for initial values
j=jacobian(u(1),u(2),X,Y,Z,x,y);     % jacobian matrix for initial values

while eps > 10^(-20)          % when step size decreases to this value iteration stops

  i=i+1;                          % iteration is incremented
  m=0.5;                          % scale factor used for regulating the step size
```

```
  s=(-1)*(j'*j)^(-1)*(j'*r');% direction of the step (Gauss-Newton)

  while eps > 10^(-20)      % if error is not decreased step size is made smaller
    step = s*m;             % step is calculated
    unew=u+snew;  % pair is modified and new error is calculated
    e(i)=totalerror(unew(1),unew(2),X,Y,Z,x,y);  % error with current values
    if e(i) < e(i-1)
       break                % if error is decreased this step is completed
     else
       m=m/2;               % if error is not decreased size of step is halved
       eps=sqrt(snew(1)^2 + snew(2)^2);   % norm of the step is calculated
     end;
  end;

  u=unew;                   % new (b,c) pair at the end of iteration step

  r=residual(u(1),u(2),X,Y,Z,x,y);    % residual vector is modified
  j=jacobian(u(1),u(2),X,Y,Z,x,y);    % jacobian matrix is modified

end;
----------------------
```

totalerror.m

```
----------------------
% This function is to determine the total error for given parameters k and f
% according to a number of given conjugate pair coordinates (X,Y,Z,x,y)

function e = totalerror(k,f,X,Y,Z,x,y)
[piksv,piksh] = size(x);       % size of the conjugate pair set is determined

for a=1:piksh

   errx(a)= ( x(a) - ( f*X(a)*(k^2-1) / (Z(a)*(k^2+1) -
               2*k*sqrt(X(a)^2+Y(a)^2+Z(a)^2)) ) );
   erry(a)= ( y(a) - ( f*Y(a)*(k^2-1) / (Z(a)*(k^2+1) -
               2*k*sqrt(X(a)^2+Y(a)^2+Z(a)^2)) ) );
   evx=errx(a)^2;           % square of the error for x coordinates is calculated
   evy=erry(a)^2;           % square of the error for y coordinates is calculated

end;

e=(sum(evx)+sum(evy))/(2*piksh);        % mean of error is calculated
----------------------
```

residual.m

```
------------------
% This function is to construct the residual vector

function r = residual(k,f,X,Y,Z,x,y)
[piksv,piksh] = size(x);       % size of the conjugate pair set is determined

for a=1:piksh
   rx(a) = ( x(a) - ( f*X(a)*(k^2-1) / (Z(a)*(k^2+1) -
                        2*k*sqrt(X(a)^2+Y(a)^2+Z(a)^2)) ) );
   ry(a) = ( y(a) - ( f*Y(a)*(k^2-1) / (Z(a)*(k^2+1) -
                        2*k*sqrt(X(a)^2+Y(a)^2+Z(a)^2)) ) );
end;

r=[rx ry];              % residual vector is constructed
---------------------
```

jacobian.m

```
------------------
% This function is to construct the jacobian matrix

function j = jacobian(k,f,X,Y,Z,x,y)
[piksv,piksh] = size(x);       % size of the conjugate pair set is determined

for a=1:piksh

      % dr/dk component; ath row 1st column
      jx(a,1)= (-f*X(a)*2*k) / (Z(a)*(k^2+1) -
               2*k*sqrt(X(a)^2+Y(a)^2+Z(a)^2)) + (f*X(a)*(k^2-
               1)*(Z(a)*2*k-2*sqrt(X(a)^2+Y(a)^2+Z(a)^2))) /
               (Z(a)*(k^2+1) - 2*k*sqrt(X(a)^2+Y(a)^2+Z(a)^2))^2 ;
      jy(a,1)= (-f*Y(a)*2*k) / (Z(a)*(k^2+1) -
               2*k*sqrt(X(a)^2+Y(a)^2+Z(a)^2)) + (f*Y(a)*(k^2-
               1)*(Z(a)*2*k-2*sqrt(X(a)^2+Y(a)^2+Z(a)^2))) /
               (Z(a)*(k^2+1) - 2*k*sqrt(X(a)^2+Y(a)^2+Z(a)^2))^2 ;

      % dr/df component; ath row 2nd column
      jx(a,2)= (-X(a)*(k^2-1)) / (Z(a)*(k^2+1) -
               2*k*sqrt(X(a)^2+Y(a)^2+Z(a)^2));
      jy(a,2)= (-Y(a)*(k^2-1)) / (Z(a)*(k^2+1) -
               2*k*sqrt(X(a)^2+Y(a)^2+Z(a)^2));

end;

j=[jx; jy];             % jacobian matrix is constructed
---------------------
```

APPENDIX C

Algorithm for Sharpening the Edge Pass

In this algorithm, the panoramic image is scanned horizontally and vertically and the pixel intensity values are recorded for a predefined array length (buffer). If the intensity difference along this array exceeds a certain threshold, then the pixel values are reorganized so that the edge pass length will be shorter. This method is implemented by using the Java algorithm given below.

Sharpen.java

```
-----------------------------------
import javax.swing.JOptionPane;

public class Sharpen {

    int verBuffersize;          // these variables are assumed to be created elsewhere
    int horBuffersize;
    int[][] verBuffer;
    int[][] horBuffer;
    int verThres;
    int horThres;
    IntRGBImage rgbOut5;

    public Sharpen(IntRGBImage rgbIn) {

        verBuffersize=Integer.parseInt(Sharpening.txtVerBuffersize.getText());
        horBuffersize=Integer.parseInt(Sharpening.txtHorBuffersize.getText());
        verThres=Integer.parseInt(Sharpening.txtVerThreshold.getText());
        horThres=Integer.parseInt(Sharpening.txtHorThreshold.getText());

        // initialize the vertical buffer
        verBuffer=new int[verBuffersize][3];
        for (int a=0;a<3;a++)
            for (int b=0;b<verBuffersize;b++)
                verBuffer[b][a]=0;
```

```
    // initialize the horizontal buffer
    horBuffer=new int[horBuffersize][3];
    for (int a=0;a<3;a++)
        for (int b=0;b<horBuffersize;b++)
            horBuffer[b][a]=0;
}

//this method is to sharpen the lower part of a panoramic image
public IntRGBImage sharpenLow(){

    int[] verAverage=new int[verBuffersize];
    int[] horAverage=new int[horBuffersize];
    int counter=0;
    int loopSize=0;
    int instantOrder=0;
    boolean giderekBuyuyor;
    boolean giderekKuculuyor;

    // Vertical sharpening
    for (int i=0;i<rgbOut5.Width;i++)
        for (int j=0;j<rgbOut5.Height;j++){

        // transfer of buffer elements
        for (int a=0;a<3;a++)
            for (int b=(verBuffersize-1);b>0;b--)
                  verBuffer[b][a]=verBuffer[b-1][a];

        verBuffer[0][0]=(rgbOut5.Data[j*rgbOut5.Width+i] >> 16) & 0xff;
        verBuffer[0][1]=(rgbOut5.Data[j*rgbOut5.Width+i] >> 8) & 0xff;
        verBuffer[0][2]=(rgbOut5.Data[j*rgbOut5.Width+i]) & 0xff;

        // initialization of average array
        for (int k=0;k<verBuffersize;k++)
            verAverage[k]=(verBuffer[k][0]+verBuffer[k][1]+verBuffer[k][2])/3;

        counter=counter+1;
        loopSize=verBuffersize-1;
        giderekBuyuyor=true;
        giderekKuculuyor=true;

        while (loopSize>0){
            if ((Math.abs(verAverage[0]-verAverage[loopSize])>verThres) &&
```

```
            (j>loopSize)&&(counter>(loopSize))&&
            (j>(0.35*rgbOut5.Height+0.6*rgbOut5.Height*
            (loopSize/verBuffersize))) ){

                for (int k=0;k<loopSize;k++){
                    if (verAverage[k]>=verAverage[k+1]){
                        continue;
                    }else{
                        giderekBuyuyor=false;
                    }
                }
                for (int k=0;k<loopSize;k++){
                    if (verAverage[k]<=verAverage[k+1]){
                        continue;
                    }else{
                        giderekKuculuyor=false;
                    }
                }
                if (giderekBuyuyor || giderekKuculuyor){
                    for (int k=1;k<=loopSize;k++){
                        if (k<((loopSize+1)/2-1)){
                                instantOrder=(int)(k*0.4);
                        } else if (k>((loopSize+1)/2)){
                                instantOrder=(int)(k+(loopSize-k)*0.6);
                        } else {
                                instantOrder=(k);
                        }
                        rgbOut5.Data[(j-k)*rgbOut5.Width+i]=0xFF000000
                                | (verAverage[instantOrder]<< 16) |
                                (verAverage[instantOrder]<< 8) |
                                (verAverage[instantOrder]);
                    }
                }
        }
        loopSize=loopSize-1;
    }
}


// Horizontal sharpening
counter=0;
instantOrder=0;
```

```
for (int j=0;j<rgbOut5.Height;j++)
    for (int i=0;i<rgbOut5.Width;i++){

    //transfer of buffer elements
    for (int a=0;a<3;a++)
                for (int b=(horBuffersize-1);b>0;b--)
                        horBuffer[b][a]=horBuffer[b-1][a];

     horBuffer[0][0]=(rgbOut5.Data[j*rgbOut5.Width+i] >> 16) & 0xff;
     horBuffer[0][1]=(rgbOut5.Data[j*rgbOut5.Width+i] >> 8) & 0xff;
     horBuffer[0][2]=(rgbOut5.Data[j*rgbOut5.Width+i]) & 0xff;

     //initialization of average array
     for (int k=0;k<horBuffersize;k++)
        horAverage[k]=(horBuffer[k][0]+horBuffer[k][1]+horBuffer[k][2])/3;

     counter=counter+1;
     loopSize=horBuffersize-1;
     giderekBuyuyor=true;
     giderekKuculuyor=true;

     while (loopSize>0){
        if ((Math.abs(horAverage[0]-horAverage[loopSize])>horThres)&&
            (j>loopSize)&&(counter>(loopSize))&&
            (j>(0.35*rgbOut5.Height+0.6*rgbOut5.Height*
            (loopSize/horBuffersize))) ){

                for (int k=0;k<loopSize;k++){
                        if (horAverage[k]>=horAverage[k+1]){
                                continue;
                        }else{
                                giderekBuyuyor=false;
                        }
                }
                for (int k=0;k<loopSize;k++){
                        if (horAverage[k]<=horAverage[k+1]){
                                continue;
                        }else{
                                giderekKuculuyor=false;
                        }
                }
                if (giderekBuyuyor || giderekKuculuyor){
                    for (int k=1;k<=loopSize;k++){
```

```
                        if (k<((loopSize+1)/2-1)){
                                instantOrder=(int)(k*0.4);
                        } else if (k>((loopSize+1)/2)){
                                instantOrder=(int)(k+(loopSize-k)*0.6);
                        } else {
                                instantOrder=(k);
                        }
                        rgbOut5.Data[j*rgbOut5.Width+(i-k)]=0xFF000000
                                | (horAverage[instantOrder]<< 16) |
                                (horAverage[instantOrder]<< 8) |
                                (horAverage[instantOrder]);
                    }
                }
            }
            loopSize=loopSize-1;
        }
    }

    return rgbOut5;
}
```